

IRIX® Admin:
System Configuration and Operation
(日本語版)

ドキュメント番号 007-2859-020JP

編集協力者

執筆 Arthur Evans, Jeffrey B. Zurschmeide

改訂 Pam Sogard, Helen Vanderberg, Bob Bernard, Terry Schultz および Julie Boney

編集 Christina Cary, Cindy Kleinfeld, および Susan Wilkening

制作 Amy Swenson, Glen Traefald, および Karen Jacobson

技術協力 Scott Henry, Carlin Otto, Kam Kashani, Chris Wagner, Paul Mielke, Robert Stephens, Joe Yetter, Gretchen Helms, John Schimmel, Robert Mende, Vernon Schryver, Michael Nelson および Landon Noll

イラスト Dany Galgani

著作権

© 1992-2002 Silicon Graphics, Inc. All rights reserved この文書の一部は、別途記載される通り、他社により著作権が保護されている場合があります。この電子文書の内容の一部または全体を、Silicon Graphics, Inc. の書面による事前の許可なく複写、配布したり、派生製品を作成することは、いかなる形態においても禁じられています。

LIMITED AND RESTRICTED RIGHTS LEGEND

The electronic (software) version of this document was developed at private expense; if acquired under an agreement with the USA government or any contractor thereto, it is acquired as "commercial computer software" subject to the provisions of its applicable license agreement, as specified in (a) 48 CFR 12.212 of the FAR; or, if acquired for Department of Defense units, (b) 48 CFR 227-7202 of the DoD FAR Supplement; or sections succeeding thereto. Contractor/manufacturer is Silicon Graphics, Inc., 1600 Amphitheatre Pkwy 2E, Mountain View, CA 94043-1351, USA.

商標とその所有者

Silicon Graphics, SGI, SGI ロゴ、Challenge, Indigo, IRIS, IRIX, Octane, Onyx, および Origin は Silicon Graphics, Inc. の登録商標であり、Crimson, Indigo2, IRIS FailSafe, IRIS InSight, IRIS WorkSpace, IRIX Networker, NUMAlink, Performance Co-Pilot, Power Challenge, Power Indigo2, Power Onyx, および XFS は Silicon Graphics, Inc. の商標です。Indy は米国においてライセンス供与されている登録商標で、その他の国では Silicon Graphics, Inc. が所有する登録商標です。

Centronics は GENICOM, L.L.C の商標です。Cray は Cray, Inc. の登録商標です。Documenter's Workbench は Novell, Inc. の商標です。FrameMaker, Illustrator, および PostScript は Adobe Systems, Incorporated の登録商標です。IBM は International Business Machines Corporation の登録商標です。LFS は Platform Computing Corporation の商標です。R4000 と R8000 は MIPS Technologies, Inc. の登録商標です。RPC と Sun は Sun Microsystems, Inc. の商標です。Tektronix は Tektronix, Inc. の商標です。Versatec は Versatec Corporation の商標です。UNIX は 米国およびその他の国々における Open Group の登録商標です。Xylogics は Xylogics, Inc. の商標です。

表紙デザイン Sarah Bolles (Sarah Bolles Design), Dany Galgani (SGI Technical Publications)

このマニュアルの変更点

このガイドでは、IRIX 6.5.18 リリースに合わせて以下の情報が追加または変更されています。

IRIX 6.5.18 リリースに合わせてこのガイドに加えられた変更は、以下のとおりです。

- 233 ページの「一般的なパラメータ」の `nbuf` パラメータに関して、値の式の説明を修正しました。
- 245 ページの「リソース・リミット・パラメータ」に、リソース・リミットの値 (`rlim_t`) に関する情報を追加しました。
- 283 ページの「スイッチ・パラメータ」の `reboot_on_panic` パラメータに関して、説明を変更しました。

また編集・レイアウト上の変更も行われています。

変更内容

バージョン	説明
007	1999年2月 IRIX 6.5.3 リリースの情報の追加
008	1999年5月 IRIX 6.5.4 リリースの情報の追加
009	1999年7月 IRIX 6.5.5 リリースの情報の追加
010	2000年1月 IRIX 6.5.7 リリースの情報の追加
011	2000年4月 IRIX 6.5.8 リリースの情報の追加
012	2000年7月 IRIX 6.5.9 リリースの情報の追加
013	2000年10月 IRIX 6.5.10 リリースの情報の追加
014	2001年1月 IRIX 6.5.11 リリースの情報の追加
015	2001年4月 IRIX 6.5.12 リリースの情報の追加

- 016 2001 年 8 月
 IRIX 6.5.13 リリースの情報の追加
- 017 2001 年 11 月
 IRIX 6.5.14 リリースの情報の追加
- 018 2002 年 2 月
 IRIX 6.5.15 リリースの情報の追加
- 019 2002 年 5 月
 IRIX 6.5.16 リリースの情報の追加
- 020 2002 年 11 月
 IRIX 6.5.18 リリースの情報の追加

目次

このマニュアルの変更点	iii
変更内容	v
図一覧	xxxii
表一覧	xxxiii
例一覧	xxxv
手順	xxxvii
このマニュアルについて	xxxix
このマニュアルの内容	xxxix
このマニュアルの対象読者	xli
関連文献	xlii
文献の入手方法	xliv
参考文献	xliv
IRIX マン・ページ	xliv
リリース・ノート	xlv
IRIX ヘルプ・システム	xlv
SGI 社の WWW (World Wide Web) サイト	xlvi
表記上の決まり	xlvi
読者からのフィードバック	xlvii

1. システム構成と操作の概要	1
システム管理の基本原則	2
アカウントのパスワード	2
特権ユーザ (root)・アカウントへのアクセスの制限	2
ユーザ・プライバシー	3
定期的なパスワード・ファイルのチェック	3
ハードウェアの変更のチェック	4
ソフトウェアのバージョンアップのチェック	4
システム停止の通知	4
不正行為に対する方針	5
システム・ログ・ブックの保守	6
トラブルの通知	7
システム管理者の仕事	7
管理ツールの概要	9
2. IRIX システムの機能の活用	11
IRIX シェル	11
正規表現とメタキャラクタの使用法	12
C シェルのショートカット	14
<i>tcsh</i> シェル	17
Bourne シェル	17
Korn シェルのショートカット	17
リモート・ワークステーションでのウィンドウの表示	18
カスタム・シェル・ウィンドウの作成	19
ファイルの自動検索と操作	21
<i>find</i> コマンドによるファイルの検索	21
<i>find</i> と <i>cpio</i> を使用したファイルの検索とコピー	24
<i>sed</i> エディタ	24
再帰コマンド	24

at、batch、cron コマンドによるタスクのスケジューリング	25
at コマンドによるタスクのスケジューリング	25
batch コマンドによるタスクのスケジューリング	26
cron コマンドによるタスクのスケジューリング	27
/etc/nologin ファイルによるログインの禁止	27
マウスのショートカット	28
マウスによるテキストのコピー・アンド・ペースト	29
マウスによる新しいシェル・ウィンドウの作成	30
マン・ページの作成	31
システム監視ツール	32
savecore によるクラッシュ・ダンプの保存	33
icrash システム・クラッシュ分析ユーティリティ	33
fru ハードウェア・エラー分析ツール	35
システム・ログを表示する sysmon	37
システム使用率を監視する availmon	38
availmon の登録	39
availmon サイト・ログ・ファイルの設定	40
availmon の管理	41
自動レポートによる availmon の使用	41
セキュリティ上の制限があるサイトで内部レポートを送信する場合の availmon の使用	42
セキュリティ上の制限があるサイトでレポートを送信しない場合の availmon の使用	43
availmon レポート	43
amsend による availmon レポートの送信	44
amreport による availmon レポートの表示	45
3. システムの起動、停止および実行レベル	47
システムの起動	47
マルチユーザ・モードでのシステムの停止	48
シングルユーザ・モードでのシステムの停止	50

IRIX オペレーティング・システムの実行レベル (システム状態)	. 50
<i>init</i> によるシステム状態の制御	. 52
システムの停止からマルチユーザ状態への移行	. 54
<i>init</i> プロセス: 初期化	. 55
<i>init</i> プロセス: 実行レベルの準備	. 55
<i>init</i> プロセス: <i>getty</i> .	. 56
システムの実行レベルの変更	. 56
実行レベルのディレクトリ	. 57
実行レベル・ファイルの変更	. 58
<i>shutdown</i> コマンドによるマルチユーザ・モードからシングルユーザ・モードへの変更	. 58
/etc/inittab によるシステムの電源切断	. 59
4. IRIX オペレーティング・システムの構成	. 61
システム構成の確認	. 61
<i>hinv</i> によるインストール済みハードウェアの確認	. 61
/hw ディレクトリ内のインストール済みハードウェアの確認	. 67
IP ネットワーク・インタフェースのハードウェア・デバイスへの割当て方	. 67
<i>versions</i> によるインストール済みソフトウェアの確認	. 68
<i>gfxinfo</i> によるインストール済グラフィックス・ハードウェアの確認	. 69
<i>uname</i> による基本システムの確認	. 70
<i>lpstat</i> によるプリンタの状態の確認	. 70
<i>chkconfig</i> によるソフトウェア構成オプションの確認	. 70

ソフトウェアの構成	73
<i>chkconfig</i> によるソフトウェア構成オプションの設定	73
システム・デフォルトの設定と変更	75
システム・ディスプレイの設定	75
マルチプロセッサ・システムでのプロセッサ割当ての変更	76
実行中の作業からの CPU の除外	78
システム名の変更	78
ネットワーク・アドレスの設定	79
デフォルト・プリンタの設定	80
時間帯の設定	80
システムの日付と時刻の変更	83
ファイルおよびディレクトリのアクセス権の制御	84
ディレクトリ・パーミッション	86
ファイル・パーミッション	86
パーミッションの変更	87
<i>umask</i> によるパーミッションの設定	88
アクセス・コントロール・リスト (ACL: Access Control Lists) と機能	89
システムのパーティション分割	89
パーティション分割の利点	91
パーティション分割の欠点	91
パーティション分割ソフトウェアのインストール	91
パーティション間のネットワークの設定	91
システム・コンソールのコントローラへの接続	93
パーティションの設定	93
<i>mkpart</i> パーティション設定コマンド	93
PROM からのパーティション分割	95
<i>partmgr</i> パーティション構成ツール	96
サポートされる構成	101
パーティション分割のガイドライン	102

PCI ホットプラグのサポート	103
PCI バス速度の制約	103
挿入手順	104
取外しとクエリ操作	105
5. マルチユーザ環境でのシステム管理	107
ユーザ・アカウント管理	107
ユーザ ID	107
グループ ID	108
プロジェクト ID 番号	109
シェル・コマンドによるユーザ・アカウントの追加	111
ユーザ・アカウント追加のための <code>/etc/passwd</code> の編集	111
ユーザ追加のための <code>/etc/group</code> の編集	112
新しいユーザのためのホーム・ディレクトリの設定	113
新しいアカウントの確認	115
シェル・コマンドによるグループの追加	116
ユーザ・グループの変更	116
システムからのユーザの削除	117
システムからのグループの削除	118
ユーザ・アカウントのロック	118
<i>newgrp</i> および <i>multgrps</i> によるユーザ・グループの変更	119
ユーザ情報の変更	119
ユーザのログイン名の変更	120
passwd コマンドによるユーザのパスワードの変更	121
ユーザのログイン ID 番号の変更	122
ユーザのデフォルト・グループの変更	123
ユーザのコメント・フィールドの変更	123
ユーザのデフォルト・ホーム・ディレクトリの変更	124
ユーザのデフォルト・シェルの変更	125

ユーザ環境	.125
ログイン・シェル	.126
C シェルの設定ファイル	.127
Bourne シェルと Korn シェルの設定ファイル	.129
シェル環境変数の設定	.131
シェル環境の表示	.131
デフォルト環境変数	.132
新しい環境変数の定義	.132
IRIX のプロンプトの変更	.133
デフォルトのファイル・パーミッション (<i>umask</i>)	.133
<i>umask</i> によるデフォルトのファイル・パーミッションの変更	.134
特別なログイン・シェル	.135
メッセージの送信	.136
電子メール・プログラム	.136
その日のメッセージ (<i>motd</i>) 機能	.136
リモート・ログイン・メッセージ	.138
<i>news</i> コマンドによるメッセージの送信	.138
<i>write</i> コマンドによるメッセージの送信	.139
<i>wall</i> コマンドによるメッセージの送信	.140
6. ディスクおよびスワップ領域の設定	.143
ディスク使用量に関するコマンド	.143
<i>du</i> (ディスク使用量) コマンド	.143
<i>df</i> (フリー・ディスク・ブロック) コマンド	.144
<i>quot</i> (各ユーザのディスク使用量) コマンド	.144
<i>diskusg</i> (ディスク・アカウント) コマンド	.144

ディスク領域の管理	144
ファイルの圧縮とアーカイブ	144
<i>quotas</i> サブシステムによるディスク領域の管理	145
ネットワーク・ファイル・システム (NFS: Network File System) によるディスク領域の管理	147
ディスク・パーティションによるディスク領域の管理	147
無駄なディスク領域	148
スワップ領域	148
ページングとスワップ領域の監視	149
仮想スワップ領域の追加	150
<i>swap -l</i> コマンドによるスワップ領域のリスト	150
<i>swap -s</i> コマンドによるスワップの動作のチェック	150
負のスワップ領域	151
仮想スワッピング	152
単一ディスク・システムでのスワップ領域の拡張	153
マルチディスク・システムでのスワップ領域の拡張	155
7. ユーザ・プロセスの管理	157
ユーザ・プロセスの監視	157
<i>top</i> によるプロセス監視	158
<i>Array</i> のプロセス管理	158
<i>osview</i> によるプロセス監視	158
<i>sar</i> によるプロセス監視	159
<i>ps</i> によるプロセスの監視	159
プロセスの優先順位	160
<i>nice</i> によるプロセスの優先順位付け	160
<i>npri</i> によるプロセスの優先順位付け	161
実行中プロセスの優先順位の変更	162
プロセスの終了	162
<i>kill</i> コマンドによるプロセスの終了	163
<i>killall</i> コマンドによるプロセスの強制終了	163

Cpusetの定義と管理	.164
Checkpoint and Restart.	.165
NQE.	.165
Share II	.165
Performance Co-Pilot	.166
8. ファイル変更モニタ (fam: File Alteration Monitor) の使用	.167
構成ファイル	.167
<i>fam</i> の基礎的なトラブルシューティング	.169
Sun NIS マスターを使用している場合の <i>fam</i> のトラブルシューティング	.172
9. コマンド (PROM) モニタの使用	.175
PROM モニタ	.176
コマンド (PROM) モニタの起動方法	.176
コマンド・モニタのコマンド・リスト	.179
コマンド・モニタでのヘルプ情報の入手方法	.181
コマンド・モニタでのコマンドの使用法	.181
コマンド・モニタでのコマンド行エディタの使用法	.181
コマンド・モニタでのコマンドの構文	.182
コマンド・モニタのファイル名の構文	.182
コマンド・モニタのデバイス名	.183
ARCS PROM のパス名の構文	.184
コマンド・モニタの実行方法	.185
コマンド・モニタによるプロセッサの再初期化	.185
PROM パスワードの設定	.185
コマンド・モニタの環境変数	.186
現在のコマンド・モニタ環境変数の表示	.190
コマンド・モニタ環境変数の変更	.191
キーボード変数の設定	.191
環境変数の削除	.192

コマンド・モニタでのプログラムの起動方法	192
<i>auto</i> コマンドによるデフォルト・ファイルの起動	193
<i>boot</i> コマンドによる固有プログラムの起動	193
スタンドアロン・シェル (<i>sash</i>)	194
スタンドアロン・シェルの起動	194
<i>bootp</i> プロトコル	196
<i>bootp</i> によるネットワークを経由した起動	196
大規模なネットワークを経由した起動	198
ディスクまたはほかのデバイスからの起動	200
10. システム・パフォーマンスの調整	201
システム・パフォーマンスの調整	202
カーネルの調整に使用するファイル	202
カーネルのチューニング・パラメータの概要	203
大きなシステム構成のチューニング・パラメータ	204

オペレーティング・システムの監視206
カーネルからのメッセージの受信とテーブル・サイズの調整206
<i>timex</i> , <i>sar</i> , <i>par</i>207
<i>timex</i> の使用208
<i>sar</i> の使用208
時間間隔を指定した <i>sar</i> の連続使用方法209
ユーザ制御アクティビティの前後における <i>sar</i> の使用方法209
コマンド実行中の <i>sar</i> と <i>timex</i> の使用方法210
<i>par</i> の使用210
<i>sar</i> , <i>par</i> , および <i>timex</i> の要約212
ディスク I/O のパフォーマンス212
ディスク I/O のチェック212
論理ボリュームを使用したディスク I/O の改善213
パーティションと追加ディスクを使用したディスク I/O の改善214
ディスク・ハードウェアの追加によるディスク I/O の改善216
ページングとスワッピング216
過剰なページングとスワッピングのチェック217
スワップ I/O 問題の対処方法219
CPU アクティビティとメモリの割当て220
CPU のチェック220
CPU パフォーマンスの向上220
使用可能メモリのチェック221
システム・メモリの容量の確認222
メモリの極大化222

オペレーティング・システムの調整	223
オペレーティング・システムの調整手順	223
オペレーティング・システムの調整: パラメータの値の確認	224
オペレーティング・システムの調整: パラメータの変更とシステムの再構築	224
システムのバックアップ	224
カーネルのコピー	225
パラメータの変更	225
autoconfig を使用した新しいカーネルの作成と起動	226
起動できないカーネルからの回復	228
複数のページ・サイズ	229
推奨ページ・サイズ	229
合体のチューニング・パラメータ	229
大きなページの確保	230

A. IRIX カーネルのチューニング・パラメータ	.231
一般的なパラメータ	.233
<i>cachefs_readahead</i>	.234
<i>cachefs_max_threads</i>	.234
<i>nbuf</i>	.235
<i>min_bufmem</i>	.236
<i>callout_himark</i>	.236
<i>ncallout</i>	.236
<i>reserve_ncallout</i>	.237
<i>ncsize</i>	.237
<i>ndquot</i>	.237
<i>nproc</i>	.238
<i>maxpmem</i>	.239
<i>syssegsz</i>	.239
<i>maxdmasz</i>	.239
<i>mbmaxpages</i>	.240
<i>ecc_recover_enable</i>	.241
<i>utrace_bufsize</i>	.241
<i>dump_level</i>	.241
<i>buf_relse_age</i>	.242
システム・リミット・パラメータ	.242
<i>maxup</i>	.243
<i>ngroups_max</i>	.243
<i>maxwatchpoints</i>	.244
<i>nprofile</i>	.244
<i>maxsymlinks</i>	.244

リソース・リミット・パラメータ	245
<i>ncargs</i>	246
<i>rlimit_core_cur</i>	247
<i>rlimit_core_max</i>	247
<i>rlimit_cpu_cur</i>	248
<i>rlimit_cpu_max</i>	248
<i>rlimit_data_cur</i>	248
<i>rlimit_data_max</i>	248
<i>rlimit_fsize_cur</i>	249
<i>rlimit_fsize_max</i>	249
<i>rlimit_nofile_cur</i>	249
<i>rlimit_nofile_max</i>	250
<i>rlimit_pthread_cur</i>	250
<i>rlimit_pthread_max</i>	250
<i>rlimit_rss_cur</i>	250
<i>rlimit_rss_max</i>	251
<i>rlimit_stack_cur</i>	251
<i>rlimit_stack_max</i>	251
<i>rlimit_vmem_cur</i>	252
<i>rlimit_vmem_max</i>	252
<i>rsshogfrac</i>	253
<i>rsshogslop</i>	254
<i>shlbmax</i>	254
<i>cpulimit_gracetime</i>	254

ページング・パラメータ	.255
<i>bdflushr</i>	.257
<i>gpgsmask</i>	.258
<i>gpgshi</i>	.258
<i>gpgslo</i>	.259
<i>maxlkmem</i>	.259
<i>maxfc</i>	.260
<i>maxsc</i>	.260
<i>maxdc</i>	.261
<i>minarmem</i>	.261
<i>minasmem</i>	.261
<i>numa_pagin_node_freemem_low_threshold</i>	.262
<i>scache_pool_size</i>	.262
<i>tlbdrop</i>	.262
<i>vfs_syncr</i>	.263
<i>maxpglst</i>	.263
<i>zone_accum_limit</i>	.263
<i>percent_totalmem_64k_pages</i>	.263
<i>nlpages_64k</i>	.264
プロセス間通信 (IPC: interprocess communication) パラメータ	.265
プロセス間通信 (IPC: interprocess communication) メッセージ・パラメータ	.266
<i>msgmax</i>	.267
<i>msgmnb</i>	.267
<i>msgmni</i>	.267
<i>msgseg</i>	.268
<i>msgssz</i>	.268
<i>msgtql</i>	.269

プロセス間通信 (IPC: interprocess communication) セマフォ・パラメータ	269
<i>semnmi</i>	270
<i>semmsl</i>	270
<i>semopm</i>	270
<i>semvmx</i>	271
<i>semaem</i>	271
プロセス間通信 (IPC: interprocess communication) 共有メモリ・パラメータ	271
<i>shmmax</i>	272
<i>shmmni</i>	273
<i>shmmin</i>	272
<i>sshmseg</i>	273
ストリーム・パラメータ	274
<i>nstrpush</i>	274
<i>nstrintr</i>	274
<i>strctlsz</i>	274
<i>strmsgsz</i>	275
<i>strholdtime</i>	275
<i>strpmonmax</i>	275
シグナル・パラメータ	276
<i>maxsigq</i>	276
ディスパッチ・パラメータ	276
<i>memaff_sched</i>	277
<i>slice_size</i>	277
<i>rtcpus</i>	278

ファイルシステム・パラメータ278
efs_inline279
cwcluster279
bdflush_interval279
dwcluster280
min_file_pages280
min_free_pages280
autoup280
ロード可能ドライバ・パラメータ281
bdevsw_extra281
cdevsw_extra282
fmodsw_extra282
vfswsw_extra283
munlddelay283
CPU 動作パラメータ283

スイッチ・パラメータ	283
panic_on_sbe	285
sbe_log_errors	285
sbe_mfr_override	286
sbe_report_cons.	286
corepluspid	286
r4k_div_patch	287
mload_auto_rtsyms.	287
xpg4_sticky_dir	287
tty_auto_strhold	287
reset_limits_on_exec	288
ip26_allow_ucmem	288
restrict_fastprof	288
reboot_on_panic	289
svr3pipe	290
nosuidshells.	290
posix_tty_default	290
restricted_chown	291
use_old_serialnum	291
subnetsarelocal	292
process_core_cpu_logging	292
タイマ・パラメータ	292
fasthz	293
itimer_on_clkcpu	293
timetrim	294

ネットワーク・ファイル・システム (NFS: Network File System) パラメータ	.294
portmap_timeout	.295
sm_timeout.	.295
GraceWaitTime.	.295
first_retry	.295
normal_retry	.296
lockd_grace_period	.296
lock_share_requests	.296
lockd_blocking_thresh.	.296
nfs_portmon	.297
svc_maxdupreqs	.297
ソケット・パラメータ	.297
unpst_sendspace	.299
unpst_recvspace	.299
unpdg_sendspace	.300
unpdg_recvspace	.301
udp_hashtablesz	.301
tcp_sendspace	.302
tcp_recvspace	.302
tcp_hashtablesz	.303
tcp_sack.	.304
VINO パラメータ	.304
vino_mtune_dmrpages	.304
ラージ・ページ・パラメータ	.305
nlpages_256K	.305
nlpages_1m.	.306
nlpages_4m.	.306
nlpages_16m	.306

拡張アカウンティング・パラメータ	306
do_procacct	307
do_extpacct	307
do_sessacct	307
min_local_paggid	307
max_local_paggid	308
asmachid	308
dfltprid	308

NUMA パラメータ308
numa_migr_default_mode310
numa_migr_default_threshold311
numa_migr_threshold_reference311
numa_migr_min_maxradius312
numa_migr_min_distance.312
numa_zone_radius_max312
numa_migr_vehicle312
numa_refcnt_default_mode313
numa_refcnt_overflow_threshold.313
numa_migr_memory_low_threshold.314
numa_migr_memory_low_enabled314
numa_migr_freeze_enabled314
numa_migr_freeze_threshold.314
numa_migr_melt_enabled.314
numa_migr_melt_threshold315
numa_migr_bounce_control_interval.315
numa_migr_dampening_enabled.315
numa_migr_dampening_factor315
mem_tick_enabled316
mem_tick_base_period316
numa_migr_unpegging_control_enabled316
numa_migr_unpegging_control_interval.316
numa_migr_unpegging_control_threshold317
numa_migr_traffic_control_enabled317
numa_migr_traffic_control_interval317
numa_migr_traffic_control_threshold317
ページ複製パラメータ317
numa_kernel_replication_ratio318

移行メモリ・キュー・パラメータ	318
numa_migr_coaldmigr_mech	318
numa_migr_user_migr_mech	319
numa_migr_auto_migr_mech	319
XVM パラメータ	319
xvm_max_revive_threads	320
xvm_max_revive_rsc	320
CXFS パラメータ	320
コア・ファイル・パラメータ	321
coremask	321
B. システム・エラー・メッセージを使用したシステム構成に関するトラブルシューティング	323
ディスク領域に関するメッセージ	323

一般的なシステム・メッセージ325
ファイルのパーミッションの問題325
IP (ネットワーク) アドレスの問題326
デフォルトのインターネット・アドレス326
IP アドレスの重複326
イーサネット・ケーブルの問題327
root ファイルシステムが見つからない場合327
ログインおよび <i>su</i> の問題328
ログイン・メッセージ328
<i>su</i> メッセージ328
ネットワークの起動の問題328
オペレーティング・システム再構築の問題329
電力障害の検出329
予備電力供給ユニット障害の検出330
SCSI コントローラのリセット330
<i>syslogd</i> デーモンの問題331
システム・クロックと日付の問題331
システムの電源切断のメッセージ331
時刻と日付のメッセージ331
タイム・サーバ・デーモンのメッセージ332
システムの再起動の情報332
保持または無視されたトラップ333
メモリとスワップのメッセージ333
<i>growreg</i> の不足しているメモリ333
ページ解放のパニック334
物理メモリの問題334
回復可能なメモリ・エラー335
<i>savecore</i> I/O エラー336
スワッピングとページングのメッセージ336
その他のメモリ・メッセージ339

システム・パニック・メッセージ	340
C. アプリケーションの調整	341
<i>timex</i> によるアプリケーション・パフォーマンスの確認	341
アプリケーション調整	342
User 時間短縮のためのガイドライン	342
過剰ページングの減少のためのガイドライン	343
I/O スループット向上のためのガイドライン	343
アプリケーションの検証と再配置	344
<i>prof</i> を使用したプログラム動作の分析	344
<i>pixie</i> を使用したプログラムの再配置	345
市販のアプリケーションの調整	346
D. IRIX ディレクトリとファイル	349
IRIX root ディレクトリ	349
重要な IRIX システム・ディレクトリ	350
重要な IRIX システム・ファイル	351
IRIX デバイス特殊ファイル	354
ASCII 変換テーブル	358
E. EPSF (Encapsulated PostScript File) バージョン 3.0 と PostScript ファイル形式	361
F. 参考文献	363
索引	365

図一覧

図 2-1	シェル・ウィンドウのポップアップ・メニュー	30
図 2-2	シェル・ウィンドウの「クローン(Clone)」サブメニュー	31
図 2-3	sysmon システム・ログ・ブラウザ	37
図 4-1	パーティション分割されたシステム	90
図 4-2	パーティション間の通信	92
図 4-3	サポートされている構成	102
図 9-1	ARCS システム起動メッセージ	177

表一覧

表 i	マン・ページの構成のアウトライン xlv
表 2-1	IRIX のメタキャラクタ 12
表 2-2	sysmon の優先度 38
表 3-1	システム状態 52
表 4-1	北米の時間帯 80
表 4-2	欧州の時間帯 81
表 4-3	アジアの時間帯 82
表 4-4	中東の時間帯 82
表 4-5	南米の時間帯 82
表 4-6	豪州とニュージーランドの時間帯 83
表 7-1	ps -ef コマンドの出力例. 159
表 9-1	コマンド・モニタのコマンド・リスト 179
表 9-2	コマンド行エディタのコマンド 181
表 9-3	コマンド・モニタ・コマンドでのデバイス名 183
表 9-4	ARCS パス名 184
表 9-5	不揮発性 RAM に格納されている環境変数 187
表 9-6	IRIX オペレーティング・システムに影響を及ぼす環境変数 189
表 9-7	ARCS PROM 環境変数 190
表 9-8	国際キーボード用の <i>keybd</i> 変数 191
表 10-1	調整に使用するファイルとディレクトリ 202
表 10-2	大きなシステム構成のチューニング・パラメータ 204
表 10-3	システム・コール・エラーと関連するパラメータ 206
表 10-4	I/O の負荷が高いシステムを示す出力結果 213
表 10-5	アプリケーションのディスク・アクセス 214

表 10-6	過剰なスワッピングとページングの有無を検査するためのフィールド	218
表 10-7	CPU の負荷が高いシステムの指標	220
表 A-1	システム・コール・エラーと IPC 調整パラメータ	265
表 D-1	ASCII から 8 進数へのマッピング	358
表 D-2	ASCII から 16 進数へのマッピング	359
表 D-3	ASCII から 10 進数へのマッピング	360

例一覧

例 2-1	<i>fru</i> コマンド出力	36
例 3-1	<i>/etc/inittab</i> ファイル	52
例 4-1	<i>hinv</i> コマンド出力	62
例 4-2	<i>versions</i> コマンド出力	69
例 4-3	<i>gfxinfo</i> コマンド出力	69
例 4-4	<i>chkconfig</i> コマンド出力	70
例 4-5	Projects ディレクトリ	87
例 5-1	<i>pwck</i> コマンド出力115
例 5-2	C シェル環境131
例 5-3	Welcome-to-the-System メッセージ137
例 7-1	Cpuset 設定ファイル164
例 8-1	デフォルトの <i>fam.conf</i> ファイル167

手順

手順 2-1	テキストのコピー・アンド・ペースト	29
手順 2-2	新しいシェル・ウィンドウの作成	30
手順 2-3	サイト・ログ・ファイルの作成	40
手順 3-1	システムの起動	47
手順 3-2	マルチユーザ・モードでのシステムの停止	48
手順 3-3	シングルユーザ・モードでのシステムの停止	50
手順 3-4	マルチユーザ・モードからシングルユーザ・モードへの変更	58
手順 4-1	ワークステーションの名前の変更	79
手順 4-2	ネットワーク・アドレスの設定	79
手順 4-3	if_cl ドライバの設定	92
手順 4-4	パーティションに分かれたシステムの設定	95
手順 4-5	パーティションの結合	96
手順 4-6	pciconfig コマンドを使用した PCI カードのインストール	104
手順 5-1	ユーザ・アカウントの追加	111
手順 5-2	/etc/passwd ファイルの編集	111
手順 5-3	/etc/group ファイルの編集	113
手順 5-4	ホーム・ディレクトリの設定	113
手順 5-5	システムへのグループの追加	116
手順 5-6	ユーザ・グループの変更	116
手順 5-7	ユーザ・アカウントの削除	117
手順 5-8	システムからのグループの削除	118
手順 5-9	ユーザ・アカウントのロック	118
手順 5-10	ユーザのログイン名の変更	120
手順 5-11	新しいパスワードの割当て	121

手順 5-12	ユーザのログイン ID 番号の変更	122
手順 5-13	ユーザのデフォルト・ホーム・ディレクトリの変更	124
手順 5-14	ユーザのデフォルト・シェルの変更	125
手順 6-1	ディスク割当て実施のガイドライン	145
手順 6-2	専用スワップ領域の追加	155
手順 8-1	fam のトラブルシューティング	169
手順 9-1	コマンド・モニタの使用	176
手順 9-2	PROM パスワードの設定	185
手順 9-3	sash を使った起動	194
手順 9-4	bootp を使った起動	196
手順 9-5	ゲートウェイの構成	198
手順 9-6	automount を使った起動	199
手順 10-1	システムの調整手順	202
手順 10-2	システムの調整	223
手順 10-3	システムの再構築	224
手順 10-4	新しいカーネルの作成	226
手順 10-5	起動できないカーネルからの回復	228
手順 B-1	ディスク使用率の通知	325
手順 B-2	SIMM のチェックリスト	334
手順 B-3	メモリやスワップ領域が不足している場合のチェックリスト	338
手順 C-1	基本ブロックのカウント	345
手順 C-2	プログラムの再配置	346

このマニュアルについて

このマニュアルでは、SGI ワークステーションやサーバを使用して、IRIX オペレーティング・システムでシステムの構成や操作を行う方法を説明します。システムの起動からオペレーティング・システム・カーネルの調整に至るまで、広範囲に渡る作業について説明します。

グラフィックス機能を持つワークステーションでは、「システム・マネージャ (System Manager)」を使用すると便利です。詳細については、『Personal System Administration Guide』を参照してください。このマニュアルは、グラフィックス機能を持つワークステーションを管理する上で最初に必要な情報を提供しています。「システム・マネージャ (System Manager)」の代わりに、IRIX のコマンド行インタフェースを使用することもできます。どちらも結果は同じです。「システム・マネージャ (System Manager)」は、IRIS WorkSpace のようなアプリケーションとは異なり、システム上に新規ファイルを作成しません。

このマニュアルおよびその他の IRIX Admin のマニュアルには、サーバのシステム管理を行う上での基本的な情報が書かれています。特に、「システム・マネージャ (System Manager)」を使用できないグラフィックス機能がないサーバを使用している場合は、これらのマニュアルをお読みください。このマニュアルでは、従来のシェル・コマンドを使用して IRIX オペレーティング・システムを管理する方法を説明します。

このマニュアルの内容

このマニュアルは、オンラインの IRIS InSight または Web ブラウザを使用して参照資料として利用されることを前提に構成されています。そのため、最初から順に読む必要はありません。各章や各節には、その項目に関する必要な情報がすべて用意されています。まえがきと第 1 章のほかに事前に資料を読む必要はありません。

『IRIX Admin: System Configuration and Operation』は、次の章で構成されています。

第1章 「システム構成と操作の概要」

この章では、管理者が使用できるさまざまなツールについて説明し、管理者向けのさまざまな情報を提供します。

第2章 「IRIX システムの機能の活用」

管理者にとって有用な IRIX 機能と、オペレーティング環境によっては使用できない IRIX 機能について説明します。

第3章 「システムの起動、停止および実行レベル」

システムの起動や停止に関する簡略な説明を提供します。

第4章 「IRIX オペレーティング・システムの構成」

新しいシステムの構成、またはシステム構成の変更に必要な作業やプロセスについて説明します。

第5章 「マルチユーザ環境でのシステム管理」

ユーザ・アカウントやユーザ・グループの追加または削除、ユーザ環境の操作、およびユーザとの通信のプロセスについて説明します。

第6章 「ディスクおよびスワップ領域の設定」

ディスク領域の簡単な管理方法について説明します。適切なディスクの使用量を保持するためのディスク領域をチェックする手順、およびユーザのディスク使用量の割当てを設定する手順を説明します。また、システムのスワップ領域を管理する方法も提供します。この章では、ディスクを追加するプロセス、ファイルシステムを作成および保持するプロセスについては説明しません。これらのプロセスについては、『IRIX Admin: Disks and Filesystem』を参照してください。

第7章 「ユーザ・プロセスの管理」

ユーザの CPU 使用量の監視方法、プロセスの優先順位の設定方法、およびプロセスの終了方法について説明します。

第8章 「ファイル変更モニタ (fam: File Alteration Monitor) の使用」

famd 変更モニタ・デーモンに関する情報を提供します。このプログラムは、複数のプログラムによって同時に使用されたファイルに対する変更をアプリケーションに通知します。

第9章「コマンド (PROM) モニタの使用」

システムの構成やテストに必要な起動レベルのユーティリティについて説明します。また、ワークステーションの起動の環境、およびコマンド・モニタの各コマンドについても説明します。

第10章「システム・パフォーマンスの調整」

システム・パフォーマンスの分析方法、およびシステム・パフォーマンスを改善するためにシステム・パラメータを調節する方法について説明します。

付録 A 「IRIX カーネルのチューニング・パラメータ」

システム・パフォーマンスの調整に必要なさまざまなパラメータについて説明します。

付録 B 「システム・エラー・メッセージを使用したシステム構成に関するトラブルシューティング」

一般的なシステム・エラー・メッセージに関するトラブルシューティングの要点を提供します。

付録 C 「アプリケーションの調整」

システムのリソースの限界を考慮に入れたアプリケーションの調整方法について説明します。

付録 D 「IRIX ディレクトリとファイル」

IRIX を管理する上で重要なディレクトリやファイルのリストを提供します。

付録 E 「EPSF (Encapsulated PostScript File) バージョン 3.0 と PostScript ファイル形式」

IRIX システムで使用される 2 種類の一般的な PostScript ファイル形式について説明します。

付録 F 「参考文献」

システム管理者にとって有用な参考文献のリストを示します。

このマニュアルの対象読者

このマニュアルは、ホーム・ディレクトリ構成と作業用ディレクトリの管理という一般ユーザの作業範囲を超えて、複数のシステムを管理するシステム管理者を対象としています。このマニュアルと付属のマニュアルには、IRIX コマンド、システム構成、およびネットワーク構成に関する情報など、システムの保守を行う管理者に必要な情報が含まれています。

一般のエンド・ユーザが実際には高度なシステム管理作業を行っている場合があります。このマニュアルは、初心者から経験豊かな管理者までが IRIX システムを正しく構成できるように、必要な操作について説明します。このマニュアルを使用することによって、今までエンド・ユーザと認識していたユーザも高度なシステム管理作業を行う自信と経験を身につけることができます。

関連文献

以下の文書には、有用な追加情報が記載されています。

- 『IRIX Admin: Software Installation and Licensing』—IRIX オペレーティング・システム (UNIX オペレーティング・システムの SGI 実装) で実行するソフトウェアのインストール方法とライセンス管理方法について説明しています。 *inst* コマンドを使ってミニルート・インストールやライブ・インストールを実行する方法が記載されています。また、IRIX で実行する特定のアプリケーションへのアクセスを制御するライセンス管理製品とそのマニュアルも紹介します。
- 『IRIX Admin: Disks and Filesystems』— ディスク、ファイルシステム、および論理ボリュームの各概念について説明しています。また、SCSI ディスク、XFS と EFS の各ファイルシステム、XLV 論理ボリューム、および帯域保証 I/O のシステム管理手順も紹介します。
- 『IRIX Admin: Networking and Mail』— *sendmail*、UUCP、SLIP、および PPP などに触れ、ネットワーク・システムやメール・システムの計画、設定、使用、および管理について説明しています。
- 『IRIX Admin: Backup, Security, and Accounting』— ファイルをバックアップし、リストアする方法、システムとネットワークのセキュリティを保護する方法、ユーザごとにシステムの利用を記録する方法について説明しています。
- 『IRIX Admin: Resource Administration』— システム・リソース管理の手引きです。IRIX プロセス制約や IRIX ジョブ制約、Miser Batch Processing System、Cpuset System、Comprehensive System Accounting (CSA)、IRIX のメモリ使用量、Array Services など、さまざまな IRIX リソース管理機能の使用法および管理方法について説明しています。
- 『IRIX Admin: Peripheral Devices』— 端末、モデム、プリンタ、CD-ROM、テープ・ドライブなど、さまざまな周辺デバイス用のソフトウェアを設定と管理方法について説明しています。また、周辺デバイスに接続されるケーブルの仕様も記載されています。

- 『IRIX Admin: Selected Reference Pages』(InSight では入手できません)— システムのダウン時に必要となるコマンドの使用方法を扱ったマン・ページについて概説しています。通常、各マン・ページは1つのコマンドについて説明しますが、密接に関連した複数のコマンドを扱っているものもあります。マン・ページは、*man(1)* コマンドを使ってオンラインで入手できます。
- 『Desktop User's Guide』は、ファイルの印刷や検索、アプリケーションの実行などの重要な操作を完了する手順についてステップごとに説明しています。また、さまざまな手法やショートカットを紹介し、コマンドとメニューについて総合的に説明します。
- 『Getting Started With Array Systems』は、Array 用プログラムの使用、構成、管理、および作成について説明しています。Array とは、高速ネットワークと Array ソフトウェア (IRIX オペレーティング・システムの Array セッション機能と Array Service 製品) と関連性のある IRIX ノードの集合です。
- 『IRIX Checkpoint and Restart Operation Guide』は、IRIX Checkpoint and Restart (CPR) の使用方法と管理方法、安全にチェックポイントおよび起動を行えるアプリケーションの開発方法について説明しています。
- 『MIPSpro Compiling and Performance Tuning Guide』は、MIPSpro コンパイラ・システム、その他のプログラミング用ツールとインタフェース、プログラムのパフォーマンスを向上させる方法について説明しています。
- 『NIS Administrator's Guide』は、NFS などほかのネットワーク・サービスにネットワーク・エンティティの位置情報を伝えるネットワーク情報サービス NIS の SGI 実装について説明しています。
- 『Personal System Administration Guide』は、SGI ワークステーションのシステム管理者が行う必要がある作業について説明し、システム管理者が使用できるさまざまなツールとユーティリティについて詳しく説明しています。
- 『NQE Administration』は、Network Queuing Environment (NQE) の構成、監視、および制御を行う方法について説明しています。
- 『Performance Co-Pilot User's and Administrator's Guide』は、Performance Co-Pilot (PCP) ソフトウェア・パッケージを管理する方法について説明しています。
- 『Share II for IRIX Administrator's Guide』は、Share II の設定方法と管理方法について説明しています。

文献の入手方法

SGI 文書は、SGI Technical Publications Library
<http://techpubs.sgi.com> で入手できます。

参考文献

以下は、システムで提供されている文献やリソースのリストとそれぞれの内容の説明です。

IRIX マン・ページ

IRIX マニュアル・ページ (マン・ページ) では、IRIX コマンドの使用法、サブルーチン、および IRIX オペレーティング・システムを構成するその他の要素に関する簡潔な情報を提供します。これらのマン・ページは、管理者にとって最も重要な参照資料の 1 つです。通常、ページごとに 1 つのコマンドを説明していますが、関連のある複数のコマンドをまとめて説明したマン・ページもあります。

IRIX マン・ページは、*man* コマンドを使用してオンラインで参照できます (ただし、マン・ページがインストールされているか、またはマウントされている場合)。マン・ページを参照するには、シェル・プロンプトで *man* コマンドを実行します。たとえば、*diff* のマン・ページを参照するには、次のコマンドを実行します。

man diff

よく参照するマン・ページや主な管理操作を行う前に必要なマン・ページは印刷し、ノートなどに保管しておいてください。

コマンド、システム・ファイル、ほかのシステム・オブジェクトについては、それぞれ別々のページで説明しています。マン・ページは表 i に示すように 7 つのセクションに分かれています。このマニュアルでマン・ページを表す場合、標準の UNIX 表記法に従い、コマンド名の後にかっこで囲んだセクション番号を付けます。たとえば、*cc(1)* は、セクション 1 の *cc* マン・ページを表しています。

表 i に、マン・ページのセクション番号とその内容を示します。

表 i マン・ページの構成のアウトライン

マン・ページの種類	セクション番号
一般的なコマンド	(1)
Administrator Commands	(1M)
システム・コールおよびエラー番号	(2)
ライブラリ・サブルーチン	(3)
ファイル形式	(4)
その他	(5)
デモおよびゲーム	(6)
特殊ファイル	(7)

IRIS InSight のオンライン・マニュアルでは、参照先としてコマンド・マン・ページが表示されます。コマンド名の後には、かっこで囲んだセクション番号が表示され、コマンド名とかっこ内の番号は実際のマン・ページとリンクしています。たとえば、`man(1)` をクリックすると、`man` コマンドのマン・ページが表示されます。

リリース・ノート

製品のリリース・ノートでは、現在のバージョンに関する情報が提供されています。また、管理マニュアルの内容が当てはまらない例外についての情報も提供されています。リリース・ノート をオンラインで参照するには、`relnotes` コマンドを使用してください。オプションの製品やアプリケーションには、それぞれ別個にリリース・ノートが付属しています。システムにインストールされているすべての製品のリリース・ノートをグラフィカルに表示するには、`grelnotes` コマンドを使用してください。

IRIX ヘルプ・システム

IRIX システムにはヘルプ・システムが用意されています。基本的なシステムのセットアップや使用方法に関する一般的な質問に対するヘルプ情報を提供します。ヘルプを参照するには、`desktophelp` コマンドを使用してください。

SGI 社の WWW (World Wide Web) サイト

SGI 社の WWW (World Wide Web) では、最新の情報を SGI 社の顧客に提供しています。下記の URL アドレスに Web ブラウザからアクセスして、インターネットで閲覧できます。

<http://www.sgi.com>—SGI 社の WWW サイトのホーム・ページ

<http://www.aw.sgi.com>—Alias | Wavefront の WWW サイトのホーム・ページ

<http://techpubs.sgi.com>—SGI 社のテクニカル文書に関するライブラリ

これらのサイトから *IRIX Admin* マニュアル・セットを含む SGI 社の公開されているインターネット情報すべてを参照できます。

表記上の決まり

このマニュアルでは、次の表記法を用いています。

『』	ほかのマニュアルのタイトルを表します。
「」	本書のほかの章や節のタイトルを表します。 また、メニュー名やボタン名などの UI (User Interface) を表します。
->	プルダウン・メニューの階層構造を表します。
<>	キーボードのジェネリック・キー (Ctrl, Shift, Alt など) を表します。 キーの操作方法として、次に例を示します。
<Enter>	<Enter> キーを押します。
<Alt>+h	<Alt> キーを押しながら h キーを押します。
<Alt>+h c	<Alt> キーを押しながら h キーを押した後、 すぐに c キーだけを押します。
<Shift>+<Ctrl>+n	<Shift> キーを押しながら <Ctrl> キーと n キーを同時に押します。
<Ctrl>+x <Ctrl>+c	<Ctrl> キーを押しながら x キーを押した後、すぐに <Ctrl> キーを押しながら c キーを押します。

ほかのマニュアルへのリンクや、アプリケーションなどの実行可能な語句は赤く表示されます。

本書のほかの章、節、または図などへのリンクは青く表示されます。

読者からのフィードバック

この文書の技術面での正確性、内容、または編成についてご意見をお聞かせください。ご意見とともに、マニュアルのタイトルと文書番号を忘れずにお知らせください。(オンライン文書の文書番号は、文書の開始ページに記載されています。印刷文書の文書番号は、裏表紙に記載されています。)

弊社には次のいずれかの方法でご連絡いただけます。

- SGI Technical Publications Library。
techpubs@sgi.com
- 次の URL にあるテクニカル文書に関するライブラリの WWW ページでフィードバックオプションをお使いください。
<http://techpubs.sgi.com>
- カスタマサービス担当者に連絡し、ご意見を SGI 社のフィードバック管理システムに記録するようにご依頼ください。
- 次の宛先に手紙をお送りください。

Technical Publications
SGI
1600 Amphitheatre Pkwy., M/S 535
Mountain View, California, USA 94043-1351, USA

- Technical Publications までファックスをお送りください (1 650 932 0801)。

ご意見をいただいた場合は迅速に対応いたします。読者の皆様からの貴重なご意見をお待ちしております。

システム構成と操作の概要

システム管理者は、まずシステムを既設のネットワークに接続するか、スタンドアロンで起動した後、使用環境に合わせてシステムを構築します。システムの構築には通常、必要なソフトウェアとハードウェアのインストール、システム・ネームやネットワーク・アドレスの設定、ユーザ・アカウントの作成などが含まれ、何も設定されていない状態から、システム管理者やユーザの要求に合わせてカスタマイズを行います。

必要なハードウェアのインストールについては、ハードウェアのマニュアルを参照してください。ソフトウェアのインストールについては、『IRIX Admin: Software Installation and Licensing』を参照してください。このマニュアルでは、システムに電源を入れた後、出荷時の状態から、実際に使用可能になるまでの作業手順を説明します。

このマニュアルでは、システム管理者がシステムを構築する手順について説明します。また、その手順を行う理由とその機能についても説明します。ここでは、システムが導入されたとき、所有権が変更されたとき、ハードウェアが大幅にバージョンアップされたときなど、大きな変更を加える場合にのみ実行される典型的な作業を説明します。また、進行中の作業や、インストールされたシステムを標準的に利用しているときに生じる作業についても説明します。

なお、システム管理者は、「システム・マネージャ (System Manager)」で利用できるグラフィカル・インタフェース・ツールについて理解している必要があります。このツールを使用することで、一般的な管理作業を簡単に実行できるようになります。このマニュアルでは、システム・マネージャについては説明しません。コマンド行およびファイル・インタフェースを使用してシステム管理用の機能を実行する方法について説明します。

この章では、IRIX のシステム管理の概要を述べます。このマニュアルの付録 F には、システム管理に役立つ書籍のリストがあります。これらの書籍は、オンラインのブックストアからも入手できます。SGI 社のシステムは、このような文献で述べている内容と一致する部分と異なる部分がありますが、正しいシステム管理に関する原則は同じです。

システム管理の基本原則

以降の節では、システム管理における基本原則の概要について説明します。管理者は、各自のサイトが最適に動作するように、個々に判断する必要があります。ここでは、一般に安全性が高いと考えられている原則について述べます。

アカウントのパスワード

サイトのセキュリティを確保するため、各ユーザには、独自のユーザ ID 番号とパスワードを持つアカウントが必要です。ユーザは絶対にパスワードを他人に教えないでください。パスワードとシステム・セキュリティについての詳細は、『IRIX Admin: Backup, Security, and Accounting』を参照してください。アカウントについての詳細は、『IRIX Admin: Resource Administration』を参照してください。

特権ユーザ (root) ・ アカウントへのアクセスの制限

通常システム管理は、システム管理者がルート (特権ユーザ) ・アカウントでログインして実施します。ルート・アカウントは一般のユーザ・アカウントとは異なり、すべてのファイルシステムにアクセスでき、ファイルやディレクトリ、プログラムへのアクセスを制御する一般のパーミッション体系から制約されることもありません。ルート・アカウントは、システム管理者がユーザ・ファイルのプライバシーやシステム・ファイルの整合性を保持しながら、システム管理に必要なすべての作業を行うためのアカウントです。ユーザを区別しないほかのオペレーティング・システムには、ユーザ・ファイルのプライバシーやシステム・ファイルの機密性を保持する手段はほとんどありません。UNIX システムでは、ルート・アカウントを使用した場合にかぎり、システムのパーミッションを無効にし、システム・ファイルを変更できます。

サイト管理者は、日常的な作業を実施するために一般ユーザのアカウントを使用する必要があります。ルート・アカウントは、システム管理に必要な作業を行う場合にのみ使用します。

マルチユーザ・システムのセキュリティを確保するには、ルート・アカウントへのアクセスを制限する必要があります。各ワークステーションの主要ユーザは各自のマシンのルート・アカウントを使用できますが、別のワークステーションのルート・アカウントにはアクセスできません。

原則として、`root` のパスワードは必要最小限のユーザだけに与えます。パスワードの機密性を保持しながら非常時には使用できるように、`root` のパスワードをロックできるファイル・キャビネットに保存する方法もあります。

ユーザ・プライバシー

マルチユーザ・システムでは、ユーザが他人のファイルにアクセスする場合があります。`chmod(1)` コマンドを使用してファイルのパーミッションを設定すると、このようなアクセスを制御できます。デフォルトのパーミッションは、シェル・パラメータの `umask` によって制御されます。`umask` の設定についての詳細は、133 ページの「デフォルトのファイル・パーミッション (`umask`)」を参照してください。

デフォルトでは、すべてのユーザにファイルの読み込み権が与えられているので、ユーザは簡単にデータを交換できます。ユーザは、各自のファイルに対してこのデフォルト設定を変更できます。ただし、`umasks` を設定しないユーザが多く、個人ファイルへのアクセス権の変更を忘れがちです。ユーザがファイルのパーミッションの重要性を認識し、他人のファイルへのアクセスに関する方針を理解することが大切です。この方針は、必要に応じて緩やかにすることも厳しくすることもできます。

定期的なパスワード・ファイルのチェック

少なくとも 1 週間に一度は、`pwck(1M)` コマンドと `grpck(1M)` コマンドを実行し、`/etc/passwd` ファイルと `/etc/group` ファイルにエラーがないかどうかをチェックしてください。この手順は、`cron(1M)` コマンドを使用して自動化できます。また、`cron` を使用すると、ユーザ・アカウントに結果をメールで送信できます。`cron` を使用してルーチン・タスクを自動化する方法については、25 ページの「`at`、`batch`、`cron` コマンドによるタスクのスケジューリング」を参照してください。

`pwck` コマンドと `grpck` コマンドはパスワード・ファイルとグループ・ファイルを読み取り、不正なエントリや整合性のないエントリを通知します。通知されるのは、IRIX の通常の操作に支障のあるエラーです。たとえば、`/etc/passwd` のエントリで 2 人のユーザ名に同じユーザ識別 (UID) 番号が与えられている場合は、`pwck` によってエラーが通知されます。`grpck` は、`/etc/group` ファイルに対して同じチェックを行います。システムが提供する標準の `passwd` ファイルには、エラーが含まれている場合があります。

ハードウェアの変更のチェック

ハードウェアの構成を変更すると、単純に思われる変更でもシステム・ソフトウェアに影響することがあります。システムに変更を加えた場合は、問題が発生したときのために、その変更箇所を知っておく必要があります。

ソフトウェアのバージョンアップのチェック

小規模なアプリケーションのバージョンアップでさえシステムに影響する場合があります。ソフトウェアには、インストール時にカスタマイズされた設定ファイルを上書きするものもあります。指定されたユーティリティが所定のディレクトリに使用されていた場合は、ソフトウェアのバージョンアップでユーティリティが移動する可能性もあります。また、ソフトウェアの新しいバージョンが古いバージョンと同じように動作しない可能性もあります。

システム・ソフトウェアの構成を変更した場合は、必ずユーザに通知してください。直接関係のないソフトウェアが突然動作しなくなったときに、すぐに調査できる体制を整えておく必要があります。システムの変更でユーザに問題が起きたときは、確実に支援できるようにしておきます。

ソフトウェアをバージョンアップするときは、現在のソフトウェアのどの機能が使用されているのか、またはバージョンアップによって不都合が生じないかどうかを事前に検討する必要があります。通常、ユーザは新しいバージョンのソフトウェアに慣れるまでに時間がかかります。

古いバージョンのソフトウェアを完全に削除すると、ユーザは戸惑うかもしれません。すべてのユーザが新しいバージョンに慣れるまで、古いバージョンのソフトウェアをシステムに残しておいてください。

システム停止の通知

システムに影響する作業が必要な場合は、通常、ユーザにその旨をできるだけ詳細に伝えておく必要があります。また、システムのサービスを中断する場合は、システムの回復予定時期をユーザに通知してください。ハードウェア、ソフトウェア、方針、手順などの変更に関する情報は、ユーザ通知メッセージ・ファイルである `/etc/motd` を使用して、ユーザに通知します。

管理作業を行う場合は、通常、システムを停止し、マルチユーザ・モード以外の実行レベルにする必要があります。これにより一般ユーザはシステムにアクセスできなくなります。マルチユー

ザ・モードを終了する前に、システムを使用しているすべてのユーザにログオフするように通知してください。このような作業は、ユーザの作業に影響しない時間に行ってください。

ユーザに通知する余裕もないまま、システムを停止しなければならないこともあります。できるだけ5～15分前には通知してください。

マルチユーザ・モードを終了する必要がある場合、必要に応じて以下に従ってください。

- サービスに影響するような作業は、できるだけシステムの使用率が低い時間帯に実施してください。事前に予定している作業については、ユーザ通知メッセージ (/etc/motd) を使用し、それらの作業日程をユーザに通知します。
- ログイン中のユーザに影響のある作業は、ログインしているユーザ名を確認した上で実施してください。システムにログインしているユーザに関する情報は、/etc/whodo、/bin/who、または /usr/bsd/w などのコマンドを使用して確認できます。また、ps -ef を実行して、大規模なバックグラウンド・タスク（コンパイル作業など）が行われているかどうかを確認してください。
- システムが使用中の場合は、システム状態の変化や保留中の保守に関する詳しい情報を事前にユーザへ提供してください。すぐに作業を実施する場合は、/etc/wall コマンドを使用し、システムを停止する時間を通知するブロードキャスト・メッセージを送信します。システムの停止は、現在の作業を終了してログオフするために十分な猶予時間（5～15分程度）をユーザに与えた上で実施してください。

不正行為に対する方針

サイト管理者は、次に示すような不正行為に対する方針を設定します。

- 故意にシステムをクラッシュさせる行為
- パスワードを推測したり盗むためのプログラムなどを使用して他人のアカウントに侵入する行為
- ほかのユーザからの電子メールを偽造する行為
- ワームやウイルスなどの有害なプログラムを作成してばらまく行為

このような行為がシステム全体に大きな損害を与える危険性があることをユーザに十分認識させてください。不正行為には、常に一貫した厳しい罰則を適用する必要があります。

不正行為による損害を防止する上で最も重要なことは、`root` パスワードへのアクセスを制限することです。

システム・ログ・ブックの保守

システム管理者は、管理対象の各システムについて完全な記録を保持する必要があります。一時的な問題を解決する場合、または一定期間にわたってシステムの運用特性を調査する場合は、システム・ログ・ブックを使用すると効果的です。また、システムの起動時にトラブルが発生するとオンライン・ログを参照できなくなるので、ログ・ブックのハードコピーを保持しておくことも重要です。

管理対象のシステムのログ・ブックに入力する必要がある情報は以下の通りです。

- 保守作業の記録（日付と作業内容）
- エラー・メッセージや診断メッセージのプリントアウト
- 各部のシリアル番号などを含む、設備やシステム構成の変更（日付と作業内容）（適用可能な場合）
- 重要な設定ファイルのコピー
- システムの各ディスクに関する `prtvtoc(1M)` の出力リスト
- `/etc/passwd` ファイル
- `/etc/group` ファイル
- `/etc/fstab` ファイル
- `/etc/exports` ファイル

システム・ログの形式とログに記載される各項目の種類は、論理構造に従う必要があります。システム・ログを定期的に更新する日記であると考えてください。システムをどのように使用するかによって、システム・ログの形式と保守の重要性は大きく変わります。

システム・ログのほかに、ユーザ・トラブル・ログを保存しておくると便利です。発生した問題をいくつかのパターンに分類し、問題をどのように解決したかを記録しておけば、同じ問題が再発したときに白紙の状態から始める必要はありません。また、ユーザ・トラブル・ログは、新しい管理者にローカル・システムの特性や、発生しやすい問題を教える際に役立ちます。

トラブルの通知

トラブルを通知するための簡単な手段をユーザに提供することも大切です。たとえば、**trouble** というメール・エリアスを作成し、問題が発生した場合、ユーザはそこにメールを送信して支援を受けられるようにします。メール・エリアスについての詳細は、『IRIX Admin: Networking and Mail』を参照してください。

システム管理者の仕事

システム管理者は、システム・セキュリティ上またはその他の理由で、エンド・ユーザが取扱えないすべての作業を実施する責任を負います。そのため、システム管理者は、このマニュアルで説明する高度なプログラムを使用します。

システム管理者は、多くの管理作業を実施する責任があります。このマニュアルで紹介されている最も一般的な管理作業は以下の通りです。

オペレーション

オマシンの正常な動作の確認、非稼働時に行う保守作業の計画、新規ユーザの登録、新規ソフトウェアのインストール、`/etc/motd` ファイルと `/etc/issue` ファイルの更新など。第2章、「IRIX システムの機能の活用」、第3章、「システムの起動、停止および実行レベル」、および第4章「IRIX オペレーティング・システムの構成」を参照してください。また、第5章「マルチユーザ環境でのシステム管理」と第9章、「コマンド (PROM) モニタの使用」も参照してください。

故障の分析

システム・ログを参照し、過去の事例を照合することによるトラブルシューティング。第1章「システム構成と操作の概要」を参照してください。

ディスク領域の使用計画

必要に応じたシステム・リソースの追加。通常システム使用率を調べることによって判断します。第6章「ディスクおよびスワップ領域の設定」、第7章、「ユーザ・プロセスの管理」、および第10章「システム・パフォーマンスの調整」を参照してください。

システムの調整

最適なパフォーマンスを引出すためのカーネルとユーザ・プロセスの優先順位の調整。第 10 章, “ システム・パフォーマンスの調整 ” と付録 A, “ ” を参照してください。

アプリケーションの調整

システムのリソースに見合ったアプリケーションの調整。付録 C, “ ” を参照してください。

リソース管理

プロセス・アカウントやディスク・アカウント、その他のリソースの共有に関する計画。『IRIX Admin: Backup, Security, and Accounting』と『IRIX Admin: Resource Administration』を参照してください。

ネットワーキング

マシン、モデム、プリンタの相互接続。『IRIX Admin: Networking and Mail』を参照してください。

セキュリティ

外部からの侵入に対する十分なセキュリティの確保と、組織内のプライバシーやシステム整合性の保持。『IRIX Admin: Backup, Security, and Accounting』を参照してください。

ユーザの移行支援

ユーザがサイトのすべてのワークステーションを利用できるように支援すること。『IRIX Admin: Networking and Mail』を参照してください。

ユーザの教育

ユーザがシステムを正しく使用できるように教えること。第 5 章, “ マルチユーザ環境でのシステム管理 ” と第 8 章「ファイル変更モニタ (fam: File Alteration Monitor) の使用」を参照してください。

バックアップ

システム・バックアップの作成と保守。『IRIX Admin: Backup, Security, and Accounting』を参照してください。

配列サービス (Array Services) 製品を使用している場合には、追加設定が必要になります。『Getting Started With Array Systems』を参照してください。

管理ツールの概要

システム管理を実施するには、システムの構成に応じて、次の管理ツールを利用できます。

システム・マネージャ

グラフィックス・ワークステーションで利用できるツールであり、システム管理機能へのアクセスが容易に行えます。このツールを使用するだけでほとんどのシステム管理作業が迅速かつ簡単に行えます。ただし、グラフィックス機能を持たないシステムでは、このツールを利用できません。

コマンド行ツール

IRIX システムには、コマンド行インタフェースを持つ豊富なシステム管理ツールが用意されています。これらのツールは、シェル・スクリプトを使用してシステムを自動的に構成したり、別のシステムからリモート・ログインするような特殊な使用状況下でシステムを修理する場合に特に有効です。

たとえば、サイト管理者は、コマンド行ツールを使用して、指定のタイミングでシステムを自動的に変更できます。たとえば、設定ファイルを定期的に配布することができます。これらのコマンド行ツールは、すべての IRIX システムに用意されています。

『IRIX Admin』マニュアル・セットは、基本的にコマンド行インタフェースとシステム・ファイルを直接扱う方法について説明しています。GUI 手法でのシステム管理作業については、『Personal System Administration Guide』を参照してください。

IRIX システムの機能の活用

この章では、システム管理者にとって便利な IRIX オペレーティング・システムの機能について説明します。これらの機能についてはほかの章でも触れますが、見落とすことがないようにここでまとめて説明します。はじめて UNIX 環境のシステムを使用する管理者にとって、この章は管理作業に必要な時間を節約する上で役に立ちます。また、システム管理者以外のユーザも新しいヒントや機能を知ることができます。

この章では、以下について説明します。

- 11 ページの「IRIX シェル」
- 18 ページの「リモート・ワークステーションでのウィンドウの表示」
- 19 ページの「カスタム・シェル・ウィンドウの作成」
- 21 ページの「ファイルの自動検索と操作」
- 24 ページの「再帰コマンド」
- 25 ページの「at、batch、cron コマンドによるタスクのスケジューリング」
- 27 ページの「/etc/nologin ファイルによるログインの禁止」
- 28 ページの「マウスのショートカット」
- 31 ページの「マン・ページの作成」
- 32 ページの「システム監視ツール」
- 38 ページの「システム使用率を監視する availmon」

IRIX シェル

IRIX シェルは、システムに対してコマンド行インタフェースを提供しています。以下で説明する機能は、IRIX コマンド・シェルの一部として提供されています。

正規表現とメタキャラクタの使用方法

多数のファイルやディレクトリを参照するためにコマンドで使用するショートカットを正規表現といいます。正規表現は、英数字および IRIX シェルに対して特別な意味を持つ一連の句読文字を組合わせたものです。このような句読文字（メタキャラクタ）といい、シェル・コマンドと共に特別な意味を持つ場合に使用します。

ショートカットを使うと、キーストロークを少なくできます。長くて複雑なコマンド行を繰り返して入力する管理者は、この機能を使用して時間を節約できます。

表 2-1 に IRIX のメタキャラクタを示します。

表 2-1 IRIX のメタキャラクタ

メタキャラクタ	意味
*	ワイルドカード
?	単一文字のワイルドカード
[]	定義設定のマーク

アスタリスク (*) は汎用的なワイルドカードです。シェルは、この文字をすべてのファイルを表すものと解釈します。たとえば、次のコマンドは、ディレクトリ内のすべてのファイルをアルファベット順に連結するようにシェルに指示します。

```
cat *
```

また、次のコマンドは、ディレクトリ内のすべてのファイルを削除するようにシェルに指示します（注意が必要）。

```
rm *
```

上記のコマンドではファイルだけが削除されます。ディレクトリを削除する場合は、`rm(1)` という別のコマンドを使用します。アスタリスク文字は常にファイル名の全体に対応するとはかぎらず、ファイル名の一部に対応させることもできる点に注意してください。たとえば、次のコマンドは、`.old` というサフィックスを持つファイルをすべて削除します。

```
rm *.old
```

疑問符 (?) は単一文字に対応するワイルドカードです。このメタキャラクタは、単一文字を表します。たとえば、ディレクトリ内に次のようなファイルがあるとします。

```
file1
file2
file3
file.different
```

file.different 以外のすべてのファイル (file1、file2、file3) を削除したい場合は、次のコマンドを入力します。

```
rm file?
```

疑問符の代わりにアスタリスクを指定すると、すべてのファイルが削除されますが、疑問符は単一文字を表すワイルドカードなので、file.different は削除されません。

角かっこ ([]) は集合の要素を表します。たとえば、上記の単一文字ワイルドカードの例で、file1 と file2 を削除し、file3 と file.different を残す場合は、次のコマンドを入力します。

```
rm file[12]
```

上記のコマンドは、先頭が file で始まり、その後に 1 または 2 が続き、その後には何も文字が続かない名前のファイルを削除します。かっこ内の文字は 1 文字ずつ別個に解釈されます。したがって、上記の例のディレクトリに file12 という名前のファイルがあっても、そのファイルは上記のコマンドでは削除されません。

また、角かっこの中ではダッシュ (-) を使用して文字の範囲を表すこともできます。たとえば、file1、file2、file3 を削除する場合は、次のコマンドを入力します。

```
rm file[1-3]
```

英文字についてもアルファベット順で範囲を指定できます。シェルでは大文字と小文字が区別されるので、角かっこ内ですべての英文字を指定する場合は、次の構文を使用します。

```
[a-z,A-Z]
```

角かっこはほかのメタキャラクターと合わせて使用できます。たとえば、次のコマンドでは、2 または 3 で終わる名前のファイルがすべて削除されますが、file1 や file.different は削除されません。

```
rm *[23]
```

C シェルのショートカット

IRIX C シェル (/sbin/csh) には、日常的なタスクを実行するためにキーストロークを少なくする機能がいくつかあります。これらの機能や csh のほかの機能についての詳細は、csh(1) マンページを参照してください。以下に、C シェルで使用可能な主な機能を示します。

ファイル名の補完機能

この機能は次のコマンドで起動されます。

```
set filec
```

ファイル名の補完機能とは、コマンド名やファイル名の最初の 1～2 文字を入力して <Esc> キーを押すと、シェルによって完全な名前が生成される機能です。さまざまなサフィックスを持つ長いファイル名を入力する場合は、この機能を利用すると便利です。入力した文字に一致するファイル、ディレクトリ、コマンドが複数ある場合、シェルはできるかぎり名前を補完させようとした後、ピープ音を鳴らしてさらに多くの情報を入力するように要求します。このとき <Ctrl>+D キーを押すと、入力した文字に一致するすべてのファイルやディレクトリを表示できます。

シェル・スクリプト

シェル・スクリプトを利用すると、シェルによって実行されるプログラムを作成できます。この機能は、構文と命令の集合を使用する点でプログラミング言語と似ていますが、コンパイラは不要であり、オブジェクト・ファイルも作成されません。つまり、シェル・スクリプトはシェルによって直接実行されます。ほとんどのシステム管理者は、この機能を利用して使用計画と複雑な手順を必要とする日常の管理作業を実行します。このような管理作業の例として、非常にサイズが大きなファイルを検索し、そのようなファイルをシステムに長期間保持できない旨をファイルの所有者に知らせることができます。シェル・スクリプトのプログラミング規則についての詳細は、csh(1) マンページを参照してください。

I/O リダイレクト

この機能を利用すると、コマンドの出力をファイルに格納したり、別のコマンドの入力として使用できます。また、コマンドの入力をファイルから取込むこともできます。通常、この機能はシェル・スクリプトの一部として使用されますが、コマンド行では複数のコマンドを連結するための一般的な方法です。次のコマンド行を見てください。

```
ps -ef | grep commandname
```

パイプ文字 (|) は、ps コマンドの出力を grep コマンドの入力として使用するようシェルに指示します。この結果、プロセス・リストの行のうち、コマンド *commandname* を含む行が画面に表示されます。これにより、システム管理者がプロセス・リストを検索する手間が省けます。出力を画面に表示するのではなく、ファイルに保存するには、次のコマンドを入力します。

```
ps -ef | grep commandname > filename
```

ジョブ制御

この機能を利用すると、同時に実行される複数のプログラムを単一の画面（またはシェル・ウィンドウ）を使用して管理できます。この機能は、単一の文字ベースの端末からシステムを操作する管理者にとって特に便利です。

コマンドのエリアシング

この機能を利用すると、よく使用するコマンド文字列のエリアスを作成し、キーストロークを少なくできます。たとえば、次のコマンドをよく使用する場合は考えます。

```
ls -CF | more
```

このコマンド行は、-CF オプション付きの ls コマンドを実行し、出力が一画面に収まらない場合に画面を一時停止して表示します。この場合、この形式のディレクトリ・リストを表示するたびに上記のコマンド行全体を入力するのは面倒なので、エリアスを作成すると便利です。任意のキーストロークを上記のコマンド行の別名として定義できます。上記のコマンド行のエリアスとして "ls" を使用し、ls コマンドの標準的な動作を変更することもできます。

ただし、別名を作成する場合は、引数を必要とするコマンドや、ls のように引数の必要性が問われないコマンドについては、それらの引数を表す記号をエリアスの定義に含める必要があります。エリアスでこのような引数を表す標準的な記号は、次の正規表現です。

\!*

先頭のバックスラッシュは、感嘆符がシェルに対して持つ本来の意味を無視し、感嘆符をそのままコマンド行に渡すことを意味します。シェルは、別名のコマンド行で指定される引数として感嘆符を解釈します。この正規表現の中のアスタリスクは、引数として入力されたすべての文字がシェルに渡されることを意味します。たとえば、このエリアスを定義するには、次の行を `.cshrc` ファイルに挿入します。

```
alias ls 'ls -CF \!* | more'
```

シェル・プロンプトで次のコマンドを入力します。

```
ls filename
```

実際には、次のコマンドが実行されます。

```
ls -CF filename | more
```

シェル・スクリプト内部でも、ファイル名の補完機能、正規表現、I/O のリダイレクトをエリアスと自由に組合わせて使用できます。

コマンド履歴

シェルは、ログイン・セッション中に入力されたコマンドを保存します。こうしたコマンドを編集したり再実行することによって、キーストロークを少なくできます。`history` コマンドを使用すると、コマンド履歴が番号を付けて順番に表示されます。ログイン・セッションで最初に入力されたコマンドは 1 番、2 番目に入力されたコマンドは 2 番と表示されます。シェルに記録されるコマンドの数は、`.cshrc` ファイル内で設定できます。最も最近のコマンドを実行するには、次のように入力します。

!!

文字 `q` で始まる一番新しいコマンドを実行するには、次のように入力します。

!q

履歴リスト内の番号を指定してコマンドを実行するには、次のように入力します。

!n

n は、再実行するコマンドの番号です。

tcsh シェル

`/usr/bin/tcsh` プログラムは、C シェルを改良したバージョンです。C シェル機能のほかに、`tcsh` には `tcsh(1)` マン・ページで説明される多くの機能があります。次に、システム管理者にとって最も便利な機能の一部を示します。

- `emacs` と `vi` のキー・コマンドを使用した便利なコマンド行編集
- 各コマンドのタイム・スタンプを記録する高機能の履歴機構
- 特定の時間や間隔でコマンドを実行するための組込み機構、ディレクトリの内容を表示するための組込み機構

Bourne シェル

Bourne シェル (`/sbin/bsh`) は C シェルに比べて機能の数は少ないですが、インタフェースに関する制限や中間層はほとんどありません。たとえば、Bourne シェルではシェル・スクリプト・プログラムをシェル・プロンプトから直接作成できます。I/O リダイレクトとコマンドのエリアシングはサポートされていますが、コマンド履歴、ジョブ制御、ファイル名の補完機能は使用できません。Bourne シェルとその機能についての詳細は、`sh(1)` マン・ページを参照してください。

Korn シェルのショートカット

Korn シェルは、C シェルと Bourne シェルの両方の機能を提供するために開発されました。`/sbin/ksh` プログラムには、Bourne シェルの簡単なシェル・プログラミング機能のほか、C シェルのジョブ制御、履歴機構、ファイル名の補完機能などがあります。Korn シェルではこれらの機能が異なる方式で実装され、コマンド行の編集機能も改良されています。詳細については、`ksh(1)` マン・ページを参照してください。次に、`ksh` の便利な機能をいくつか紹介します。

`emacs` による編集

この編集モードを起動するには、`emacs` オプションまたは `gmacs` オプションをオンにします。コマンド行を編集するには、カーソルを編集対象箇所へ移動した後、`emacs` でテキスト・ファイルを編集する場合と同じ方法で、コマンド行に対して文字や単語の挿入、または削除を行います。コマンド行の先頭の文字を除く任意の位置で、`emacs` の編集コマンドのすべてを使用できます。

vi による編集

この編集モードを起動するには、**vi** オプションをオンにします。このオプションには 2 種類のタイピング・モードがあります。コマンドを入力するときは、**vi** の入力モードを使用しています。編集する場合は、<Esc> キーを押して制御モードを起動し、**vi** でテキスト・ファイルを編集する場合と同じ方法で、コマンド行に対して文字や単語の挿入、または削除を行います。

ジョブ制御

指定されたプロセス（ジョブ）に関する情報を表示します。ジョブを指定する引数がない場合は、すべてのアクティブ・プロセスに関する情報を表示します。**-l** フラグを指定すると、通常の情報に加にプロセス ID 番号も表示されます。**-n** フラグを指定すると、最後の通知以降に停止または終了したジョブだけが表示されます。**-p** フラグを指定すると、プロセス・グループだけが表示されます。ジョブ制御の引数についての詳細は、`ksh(1)` マン・ページを参照してください。

bg コマンドは、指定されたプロセスをバックグラウンドに移動します。ジョブが指定されていない場合、カレント・プロセスはバックグラウンドに移動されます。

fg コマンドは、指定されたプロセスをフォアグラウンドに移動します。プロセスが指定されていない場合、カレント・プロセスはフォアグラウンドに移動されます。

リモート・ワークステーションでのウィンドウの表示

ネットワークに接続されたリモート・ワークステーション上のグラフィカル・ユーティリティやアプリケーションを起動して、ローカルのワークステーションでそのウィンドウやすべての入出力を表示できます。これは自分の席から、リモート・ワークステーションの保守を行う場合に非常に便利です。呼出されたプログラムはリモート・ワークステーションで起動し、ウィンドウは指定したディスプレイ・ワークステーションに表示されます。

まず、リモート・システムが自分のディスプレイにアクセスできるようにする必要があります。これには、ディスプレイ・ワークステーションで `xhost` コマンドを使用します。

xhost +remote_workstation

次に、`rsh(1C)`、`rlogin(1C)`、または `telnet(1C)` を使用して、そのシステムの保守を行うために必要な特権レベルでリモート・ワークステーションにログインします。これは、`guest` アカウントで十分な場合もあれば、管理者がシステムにユーザ・アカウントを持っている場合もあります。場合によっては、`root` のアクセス権が必要です。作業に適したアクセス・レベルを選択してください。続けて、`csch` ユーザと `tcsh` ユーザの場合は次のコマンドを入力します。

```
setenv DISPLAY local_workstation:0
```

または、`bsh` ユーザと `ksh` ユーザの場合は次のコマンドを入力します。

```
DISPLAY=local_workstation:0 ; export DISPLAY
```

ウィンドウを表示するワークステーションの名前を `local_workstation` に指定します。ローカル・ワークステーション名は、プログラム実行中のリモート・システムの `/etc/hosts` ファイルに記述されている必要があります。

リモート・システムで任意のユーティリティやアプリケーションを起動すると、ローカル・ワークステーションにウィンドウが表示されます。すべての入出力は、ローカル・ワークステーションを通じて行われます。ネットワークの通信容量に制限があるため、プログラムの応答時間が通常より非常に遅くなることがあります。

入出力の完了後、表示プログラムを通常どおり終了し、リモート・ワークステーションをログアウトします。

カスタム・シェル・ウィンドウの作成

IRIX システムでは、グラフィックス・ワークステーションのパレットから希望のカラーを選択し、シェル・ウィンドウを作成できます。また、システムにあるフォント・セットから希望のフォントを選択することもできます。`xwsh` コマンドでシェル・ウィンドウを作成し、さらにオプションを使用してさまざまなフォントやカラーなどの機能を制御できます。ウィンドウで `f` コマンド・シェルは、デフォルトで、`/etc/passwd` ファイルのエントリが使用されます。または、`xwsh(1G)` マン・ページの指示に従って、コマンド行で指定することもできます。

`xwsh` で利用できる機能のリストについては、`xwsh(1G)` マン・ページを参照してください。次は、最もよく使用される機能の例です。

たとえば、シェル・ウィンドウの背景を暗いグレーにし、テキストを黄色にするには、次のコマンドを実行します。

```
xwsh -fg yellow -bg gray40 &
```

このコマンドによって、新しいウィンドウが作成され、指定されたカラーの新しいシェルが生成されます。フォントとウィンドウ・サイズは、コマンドで指定されていないのでデフォルト値が使用されます。シェルはコマンドを入力する位置にウィンドウを固定しないので、シェルを作成したコマンドはバックグラウンドで実行されます。コマンド行の末尾にアンパサンド記号 (&) を付けると、コマンドを常にバックグラウンドで実行できます。プロセスをバックグラウンドで実行する方法については、`cs(1)` マン・ページを参照してください。

使用できるグレーは 100 種類あります。Gray0 は最も暗いグレーで、黒とほぼ同じです。Gray100 が最も明るく、白とほぼ同じです。前景 (テキスト) に黄色、背景に Gray40 を使用すると、グレーの黒板に黄色いチョークで字を書いたように見えます。使用できるカラーのリストを表示するには、`colorview` コマンドを使用してください。このコマンドを実行すると、カラーのスクロール・リストと、現在選択されているカラーの見本を示すウィンドウが表示されます。

次の例では、空色の背景上のテキストを黒に変更し、ウィンドウのサイズを指定します。このとき、背景と前景のコントラストを強くすると、画面が読みやすくなります。

```
xwsh -fg black -bg skyblue -geometry 80x40 &
```

`geometry` オプションの最初の数字 (80) は、新しいウィンドウの幅を 80 文字 (デフォルト値) に指定しています。2 番目の数字 (40) は、画面の行数を指定します。コマンド行の末尾にアンパサンド (&) があるので、この `xwsh` コマンドもバックグラウンドで処理されます。

`xwsh` コマンドに `-iconic` フラグを付けると、新しいシェル・ウィンドウをデスクトップのアイコンとして起動できます。

デフォルト以外のフォントを使用する場合は、オンスクリーンのフォント選択ユーティリティ (`xfontsel`) を使用するか、コマンド行でフォントを指定できます。コマンド行ではフォントの属性をいくつも指定する必要があるため、ユーティリティを使用する方が簡単です。また、フォント、字体や行間隔 (標準体またはボールド体、および標準の間隔または狭い間隔の指定など)、フォント・サイズなどには多くの選択項目があるため、選択に時間がかかります。オンスクリーンのフォント選択ユーティリティを使用すれば、何を選択するとどのような表示になるかをウィンドウ上で確認できます。

選択項目の指定が完了したら、フォントの選択情報や `xwsh` コマンドのオプションなどをシェル・スクリプトにコピー・アンド・ペーストし、後で再利用できます。たとえば、次の `xwsh` コマンドの例では、IRIS 固有のフォントである *haebfix* を標準体で通常間隔、高さを 15 ピクセルで使用することを指定します。その他の情報は、フォント選択ユーティリティによって自動的に生成されます。

```
xwsh -iconic -fg yellow -bg grey40 -geometry 80x40 -fn \  
-sgi-haebfix-medium-r-normal--15-150-72-72-m-90-iso8859-1 &
```

このコマンドや、頻繁に使用する別の IRIX コマンドに対してエリアスを作成することができます。詳細については、14 ページの「C シェルのショートカット」、またはシェルに関するマニュアルを参照してください。

上記のコマンドはシェル・スクリプト内に一行で表示されます。この例では、書式上の制約からコマンドが 2 行に分割されています。

フォント選択ユーティリティの使用方法については、`xwsh(1G)` と `xfontsel(1)` のマン・ページを参照してください。

ファイルの自動検索と操作

IRIX システムには、多数のファイルを迅速に操作するためのツールがいくつか用意されています。以下は最もよく使用されるツールです。

- ファイルを検索し、それらのファイルを操作するための別のコマンドを起動させる `find(1)` ユーティリティ。
- 事前に定義済みのコマンドを使用し、ファイルを編集する `sed(1)` プログラム。
- 多くのプログラムで使用できる再帰オプション。再帰オプションを使用すると、さまざまなレベルのサブディレクトリにあるファイルを迅速に操作できます。

`find` コマンドによるファイルの検索

`find` コマンドを使用してファイルを検索し、検索したファイルに対してさらに別のコマンドを実行することができます。`find` コマンドは、指定されたディレクトリとその下のすべてのサブ

ディレクトリを検索し、指定されたファイルを見つけます。find の基本的なコマンド行は次のような形式になります。

```
find . -name filename -print
```

このコマンドは、カレント・ディレクトリとそのすべてのサブディレクトリを検索し、*filename* に一致するすべてのファイル名を見つけ、画面以上にそのパス名を表示します。**-name** オプションは、次に指定する引数が検索対象のファイル名であることを指示します。**-print** オプションは、ファイルが検出されたときにそのファイル名を表示することを指示します。

検索では、正規表現を使用することもできます。12 ページの「正規表現とメタキャラクタの使用方法」を参照してください。

次のコマンド行は、`/tmp/file` の最新更新日付より後に変更されたファイルを検索します。`touch` (`touch(1)` を参照) を使用して古い日付のファイルを作成すると、その日付以降に変更されたファイルを検索できます。

```
find / -local -newer /tmp/file -print
```

次に、ユーザのディレクトリの中で `missingfile` という名前のファイルを検索する例を示します。

```
find /usr/people/trixie -name missingfile -print
```

find コマンドでは、検出されたファイルに対して別のコマンドを実行することもできます。次に、カレント・ディレクトリとすべてのサブディレクトリにあるファイルのパーミッションを変更する例を示します。

```
find . -name '*' -local -exec chmod 644 {} \;
```

find コマンドの後にあるピリオド (.) は、カレント・ディレクトリから検索を開始し、その下にあるすべてのサブディレクトリを検索対象とすることを示します。次の **-name** フラグは、検索されるファイル名を示します。この例では、アスタリスク (*) が指定されているので、ディレクトリ構造内のすべてのファイルが選択されます。正規表現とメタキャラクタについての詳細は、12 ページの「正規表現とメタキャラクタの使用方法」を参照してください。

-local オプションは、ディレクトリ構造内に物理的に存在するファイルだけを検索するように find に指示します。ネットワーク・ファイル・システム (NFS: Network File System) によってマウントされているファイルやディレクトリは検索対象にはなりません。**-exec** オプションは、次

の引数をコマンド（この例では `chmod 644`）として `find` に実行させます。中かっこ (`{}`) は `find` が現在処理しているファイルを表します。

コマンド行の最後の 2 文字は、`chmod` コマンドの一部であり、検索パラメータと一致するすべてのファイルに対して `-exec` オプションで実行されます。バックスラッシュは、その後のセミコロンがシェルによって解釈されないようにするために必要です。このセミコロン (`;`) は `chmod` コマンド内のキャリッジ・リターンを表すもので、シェルではなく `chmod` コマンドに渡されます。

`find` コマンドにはたくさんの便利なオプションがあります。

-inum *n* ファイル名の代わりに *i* ノード番号 (*n*) によってファイルを検索します。

-mtime *n* 指定した時間内 (*n*) で変更されていないファイルを示します。

-perm [*-*]l|onum

onum (8 進数) に一致するパーミッションを持つファイルを示します。*onum* はファイルのパーミッションを表す 8 進数です。`chmod(1)` マン・ページを参照してください。マイナス符号 (`-`) を指定しない場合は、*onum* に完全に一致するパーミッションを持つファイルだけが示れます。

onum の前にマイナス符号を付けると、*onum* の中に実際に設定されているビットだけがファイルのパーミッションと比較されます。

-type *x* *x* で指定されるタイプによってファイルを示します。ファイル・タイプとして、*b* (ブロック型特殊ファイル)、*c* (キャラクタ型特殊ファイル)、*d* (ディレクトリ)、*l* (シンボリック・リンク)、*p* (FIFO: 名前付きパイプ)、*f* (普通のファイル)、*s* (ソケット) を指定できます。

-links *n* *n* 個のリンクを持つファイルを示します。

-user *uname* *uname* (ユーザ名) に属しているファイルを示します。*uname* が数値で、`/etc/passwd` ファイルにログイン名として登録されていない場合は、*uname* はユーザ ID とみなされます。

-group *gname* *gname* (グループ名) に属しているファイルを示します。*gname* が数値で、`/etc/group` ファイルに登録されていない場合は、*gname* はグループ ID とみなされます。

-size *n* [*c*] ファイルのサイズが *n* ブロック (1 ブロック = 512 バイト) であることを示します。*n* の後に *c* を付けると、サイズはバイト単位になります。

`-ok cmd` `-exec` と同じ機能です。ただし、検索されたファイルに対してコマンド (*cmd*) を実行するかどうかをたずねる疑問符 (?) が出力されます。この機能は、ファイルを選択して削除する操作などを行う場合に便利です。

find と cpio を使用したファイルの検索とコピー

`find` コマンドと `cpio` コマンドを使用すると、各ディレクトリに対するパーミッションがあればディレクトリやディレクトリ階層を安全かつ簡単にコピーできます。ディレクトリとその中のすべてのファイル、ディレクトリ階層全体とその中のすべてのファイルをコピーするには、次のコマンドを使用します。

```
mkdir new_directory_name
cd the_directory_you_want_to_copy
find . -print | cpio -pdlmv full_path_including_new_directory_name
```

この順序でコマンドを実行すると、ファイルシステムの境界を透過的に横断されるだけでなく、シンボリック・リンクが新しいディレクトリ内に保持されます。

詳しい説明については、`find(1)` マン・ページを参照してください。

sed エディタ

ストリーム・エディタの `sed` を使用すると、ファイルの編集作業を自動化できます。`sed` は、編集スクリプトで定義した変更内容に従って、ファイル内のテキストを編集します。`sed` はファイルに対して編集スクリプトで定義した変更を加え、編集済みのファイルを標準出力に送信します。このコマンドについての詳細は、`sed(1)` マン・ページおよび標準の UNIX マニュアルを参照してください。

再帰コマンド

再帰コマンドを使用すると操作時間を大幅に節約できます。たとえば、ディレクトリ内のすべてのファイルやディレクトリの所有者、およびその下のサブディレクトリ内のすべてのファイルやディレクトリの所有者を再帰的に変更する場合は、`chown(1)` 再帰オプションを使用できます。

```
chown -R username directory
```

次に、再帰オプションを持つ IRIX システムのコマンドを示します。

```
ls -R
```

```
rm -r
```

```
chgrp -R
```

再帰オプションを持たないコマンドを再帰的に実行する場合は、find を使用してそのコマンドを実行します。詳細については、21 ページの「find コマンドによるファイルの検索」を参照してください。

再帰オプションを使用すると、ファイルやファイルシステムが確認されずに自動的に変更されてしまうので、非常に危険です。chgrp コマンドは、ファイルシステムのツリー内を下方向だけでなく上方向にも再帰的に操作します。再帰コマンドを無条件で実行できると確信する場合を除いて、個別に操作を実行することをお勧めします。また、再帰的コマンドとメタキャラクタ（12 ページの「正規表現とメタキャラクタの使用法」を参照）を組合わせて使用することは避けてください。rm コマンドの **-i** オプションを使用すると、ファイルを削除する前に確認のプロンプトを表示できます。

at、batch、cron コマンドによるタスクのスケジューリング

at、batch、cron の各ユーティリティを使用すると、管理者やユーザが行う多くの日常作業を自動化できます。ただし、ユーザの cron パーミッションと at パーミッションが有効であることが必要です。詳細は、crontab(1) を参照してください。これらのユーティリティの機能は似ています。どのユーティリティも、コマンドを後で実行します。ただし、at は特定の時刻に 1 回だけ指定されたコマンドを実行し、cron はスケジュールを設定した上でそのスケジュールに従ってコマンドを指定された回数だけ実行し、batch はシステムの負荷が低くなったときにコマンドを実行する点がそれぞれ異なります。次に、各ユーティリティの使い方の例を示します。詳細については、at(1)、batch(1)、cron(1M) マン・ページを参照してください。

at コマンドによるタスクのスケジューリング

特定時刻に一度だけタスクを実行する場合は、at コマンドを使用します。たとえば、ある日の真夜中に共有ディレクトリを使用禁止にする必要がある場合、次のコマンドを入力すれば、その時刻にいる必要はありません。

```
at 2400
chmod 000 /usr/public
<Ctrl>+D キーを押します。
```

at コマンドと、コマンドを実行する日付と時刻は同一の行に入力する必要があります。この行を入力した後に **<Enter>** キーを押すと、プロンプトは表示されず、at は入力待ち状態になります。次に、シェル・プロンプトで入力する場合と同じ方法で、実行予定のコマンドを入力します。コマンドを入力した後、再度 **<Enter>** キーを押し、それ以上のコマンドがないことを at に知らせるために **<Ctrl>+D** キーを押します。単一の at コマンドを使用し、複数のコマンドを特定時刻に実行させることもできます。たとえば、共有ディレクトリを7月14日の正午に使用禁止にすると同時にその旨をユーザに知らせるためにユーザ通知メッセージを変更する場合は、新しいユーザ通知メッセージを含む /tmp/newmesg というファイルを作成した後、次のようなコマンドを入力します。

```
at 1200 July 14
chmod 000 /usr/public
mv /etc/motd /etc/oldmotd
mv /tmp/newmesg /etc/motd
<Ctrl>+D キーを押します。
```

デフォルトでは、at を使用して実行したコマンドの出力は、システムの電子メールで起動側のユーザに送信されます。コマンドの出力結果を別の場所に送信する場合は、パイプ (|) やファイル・リダイレクト (>) などによって標準出力をリダイレクトします。標準出力のリダイレクトについての詳細は、コマンド・シェルの説明を参照してください。

at コマンドについての詳細は、at (1) マン・ページを参照してください。

batch コマンドによるタスクのスケジューリング

batch コマンドは at と同じ動作を行いますが、コマンドの実行予定時間を事前に指定しません。コマンドは、システムの全体的な負荷が十分低くなり実行可能状態になった時点で実行されます。cron サブシステムのほかのコマンドと同じように、特に指定しない場合は、コマンドの出力は起動したユーザに電子メールで送信されます。システムの遅延やピーク時にパフォーマンスを低下させる CPU 集約型の大規模なジョブについては、batch を使用すると効果的です。ピーク時

の終了まで実行を待てる場合は、このようなジョブを batch の待ち行列に入れてシステムに実行を任せることをお勧めします。batch コマンドについての詳細は、batch(1) マン・ページを参照してください。

cron コマンドによるタスクのスケジューリング

コマンドをスケジュールに従って定期的に行いたい場合は、cron コマンドとそのサブシステムが提供する精密なジョブ・スケジューリング機構を使用できます。at コマンドと batch コマンドは技術的に cron サブシステムの一部であり、cron を使用してその機能を実現しています。cron は、同じサブシステム内で最も柔軟な設定が可能なコマンドです。

cron は crontab ファイルを設定することによって使用できます。このファイルには、実行するコマンドとそれらのスケジュールを記述します。crontab ファイル設定についての詳細は、cron(1M) および crontab(1) マン・ページを参照してください。

ネットワーク・バックアップ作業のスケジューリング、パスワード・ファイルの整合性のチェック、システムとの対話を必要としないほかの定期的なタスクについては、cron を使用すると効果的です。cron を使用して実行したコマンドの出力結果は、デフォルトで crontabs ファイルを送信したユーザに電子メールで送信されます。したがって、cron によって pwck(1M) などのコマンドをスケジュールした場合は、必要に応じてメールで送信されたテスト結果を検証できます。

cron コマンドまたは at コマンドを使用して crontabs ファイルを変更した場合は、cron を再起動しないかぎり、その変更内容は有効になりません。

システム管理者は、/etc/cron.d 内のファイル cron.allow、cron.deny、at.allow、および at.deny を使用して、at コマンドと batch コマンドへのアクセスを制御できます。また、特定のユーザやすべてのユーザがこれらのコマンドを使用できないように設定できます。詳細については、at(1) マン・ページと crontab(1) マン・ページを参照してください。

/etc/nologin ファイルによるログインの禁止

/etc/nologin ファイルは、一般ユーザのログインを禁止します。login(1) プログラムのこの機能は、システム管理者がシステムを完全なマルチユーザ・モードに保持したまま、一般ユー

ザのログインを禁止できるように設計されたものです。システムの完全なバックアップを作成する場合、またはオペレーティング・システムを予告なく停止させるようなシステム・テストを実施する場合などに効率的です。このような作業はシステムの使用率が低い時間帯に実施してください。

`/etc` ディレクトリに `nologin` というファイルを作成するだけで、ログインを禁止できます。`/etc` にこのファイルを作成する場合は、`root` でログインする必要があります。ログインを禁止するほかに、`login` プログラムはユーザのアクセスを拒否する場合に `/etc/nologin` の内容を表示します。`/etc/nologin` はコンソール・ウィンドウでは適用できないので、注意してください。

`/etc/nologin` に登録するメッセージの例を以下に示します。

```
The system is unavailable for a few moments while we perform some
routine maintenance. We will be done shortly and regret any
inconvenience this may cause you. -Norton
```

ログインを再び許可するには、`/etc/nologin` ファイルを削除します。システムがクラッシュした場合には、`tty` 回線またはコンソールに接続する疑似 `tty`（ただし、ネットワーク経路を除く）を使用して `root` でシステム・コンソールにログインし、`/etc/nologin` を削除することができます。

マウスのショートカット

グラフィックス・ワークステーション（および一部の X 端末）では、ショートカット機能を使用できます。文字ベースの端末だけを使用して管理作業を行うサーバ管理者は、ショートカット機能を使用できない場合があります。システムのグラフィックス・コンソールでは、プルダウン・メニューやポップアップ・メニューを使用せずにウィンドウ間でカット・アンド・ペーストを行います。ポップアップ・メニューを使用すると、これらのすべての操作をウィンドウで実行できます。

マウス・ボタンの動作は変更することができます。この節の例では、マウス・ボタンの動作はデフォルト設定であることを前提とします。マウス・ボタンの動作を変更している場合は、その変更を考慮した上で次の例を参照してください。

ポップアップ・ウィンドウの使い方の詳細については、オンラインの『Desktop User's Guide』を参照してください。

マウスによるテキストのコピー・アンド・ペースト

手順 2-1 は、マウスのショートカットを使用する場合の最も一般的な方法です。ここでは、画面上の異なるウィンドウ間でテキストをカットまたはコピーしてペーストする方法を説明します。

手順 2-1 テキストのコピー・アンド・ペースト

テキストをカットまたはコピーして異なるウィンドウ間でペーストするには、以下の手順に従ってください。

1. マウスで制御するカーソルを画面上で見つけてください。カーソルの形状は、ウィンドウの作業領域内では小さい矢印、またはウィンドウ枠の上やアプリケーションのウィンドウの作業領域内では別の図形になります。カーソルがすぐに見つからない場合は、マウスを少し動かして画面の様子を観察してください。カーソルは簡単に見つかります。
2. ウィンドウ間でペーストするテキストの先頭にカーソルを置き、マウスの一番左側のボタンを押します。マウス・ボタンを押したまま、ペースト対象のテキストの末尾までカーソルを移動してください。選択されたテキストは強調表示されます。テキスト内の広い領域を選択する場合は、その全領域上でカーソルを移動する必要はありません。選択するテキストの末尾にカーソルを直接移動すると、その間にあるテキストが選択されます。テキストの列や不連続な複数のテキストは同時に選択できません。選択するテキストの末尾にカーソルが移動したら、マウス・ボタンを放してください。テキストは強調表示されたままになります。
3. テキストのペースト先とするウィンドウにカーソルを移動し、そのウィンドウがテキストを取込める状態であることを確認します。たとえば、長いコマンド行をペーストする場合は、待機状態のシェル・プロンプトがあり、そこに別のコマンドが入力されていないことを確認します。ファイルにテキストをペーストする場合は、エディタ上でファイルが開いていること、およびエディタがテキスト挿入可能なモードになっていることを確認してください。
4. テキストをペーストするには、ペースト先のウィンドウ内にカーソルを移動し、マウスの中央のボタンを1回押します。中央のボタンを押すたびに、テキストがウィンドウにペーストされます。テキストが表示されるまでに時間がかかる場合もあります。テキストが表示される前にボタンを何度も押すと、押した回数だけのテキストがペーストされます。

5. 選択したテキストは、ウィンドウ (テキストを選択した元のウィンドウも含む) の下部にペーストすることもできます。カーソルがそのウィンドウ内にある場合にマウスの一番右側のボタンを押し、表示されるポップアップ・メニューから「SEND (Send)」オプションを選択してください。

選択したテキストは、新たに別のテキストを選択するか、またはカーソルを元のウィンドウに戻してマウスの一番左側のボタンを1回クリックするまで記憶されます。

マウスによる新しいシェル・ウィンドウの作成

手順 2-2 で、マウスを使って新しいシェル・ウィンドウを作成する方法を説明します。

手順 2-2 新しいシェル・ウィンドウの作成

新しいシェル・ウィンドウを作成するには以下の手順に従ってください。

1. カーソルをシェル・ウィンドウに移動し、マウスの一番右側のボタンを押して、そのままにします。図 2-1 のようなポップアップ・メニューが表示されます。



図 2-1 シェル・ウィンドウのポップアップ・メニュー

2. マウス・ボタンを押したままでマウスを下方向に移動し、「クローン (Clone)」オプションを選択すると、サブメニューが表示されます。このサブメニューには、新しいウィンドウのための各種オプションがあります。これらのオプションを選択すると、現在使用中のシェル・ウィンドウと同じ機能を持つ別のシェル・ウィンドウを作成できます。この機能をクローン・オプションと呼びます。カレント・ウィンドウのテキストと背景の色は新しいウィンドウにも継承されます。新しいウィンドウ内の行の数は、サブメニューの選択項目で指定します。図 2-2 のように、この行の数としては、カレント・ウィンドウと同じ数、24、40、60 のいずれかを選択できます。

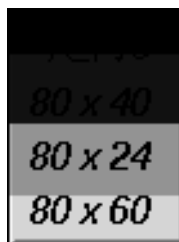


図 2-2 シェル・ウィンドウの「クローン(Clone)」サブメニュー

3. マウスを下方向に移動して各オプションを強調表示し、希望するオプションの上でマウス・ボタンを放して適切なサイズを選択します。新しいウィンドウが現在の画面上に表示されず。任意のシェル・ウィンドウでこの手順を何度でも実行できます。

マン・ページの作成

マン・ページは、オンラインで参照できる IRIX コマンドの説明です。IRIX の標準のプログラム、ユーティリティ、ファイルについては完全なマン・ページがオンラインで参照可能です。これらのマン・ページは、`man(1)` コマンドを使用して参照します。また、次の手順に従って、独自のマン・ページを作成することもできます。一般ユーザ用にスクリプト、ユーティリティ、プログラム、アプリケーションを作成した場合はマン・ページを作成してください。ユーザは新しいツールの使用方法を簡単に理解でき、将来的な保守作業も簡単になります。

サイトの中にはオプションの Documenter's Workbench が導入されていない場合もありますが、テキスト・エディタがある場合は、マン・ページのコピーを作成できます。詳細については、次を参照してください。

Documentor's Workbench を使用せずに(テキストをフォーマットする `nroff` コマンドを使用せずに) 純粋テキスト形式のマン・ページを作成する場合は、`vi` (または任意のエディタ) を使用し、標準的なマン・ページの形式に従ってスクリプトやプログラムを文書化します。マン・ページのファイルには、スクリプト名やプログラム名の後 (名前の末尾) に `.1` というサフィックスを付け、ローカルなマン・ページであることを示します。

メモ: サフィックスは数字の "1" ではなく英文字の "1" です。

作成したマン・ページは、man コマンドによって認識されるように /usr/man ディレクトリ階層に格納します。これらのマン・ページは、/usr/man/man1 などのローカル・ディレクトリに格納してください。ローカルなマン・ページであることを示す文字の 1 を使用します。ディレクトリがない場合は、次のコマンドで新しいディレクトリを作成してください (root でログインする必要があります)。

```
mkdir /usr/man/man1
```

長いマン・ページは圧縮してディスク領域を節約する必要があります。pack コマンドを使用し、テキスト・ファイルを簡潔な形式に圧縮してください。たとえば、“program” という名前のユーザ・スクリプトで作成したマン・ページを圧縮するには、次のコマンドを入力します。

```
pack program.1  
mv program.1.z /usr/man/man1/program.1.z
```

メモ : man コマンドは自動的に圧縮を解除してマン・ページを読取ります。

マン・ページをテストするには、次のコマンドを入力します。

```
man 1 program
```

指定したマン・ページが表示されます。ただし、/usr/man が \$MANPATH 環境変数で設定されていることが前提です。詳細については、man(1) マン・ページを参照してください。

システム監視ツール

IRIX システムでは、システムで生じる可能性がある問題に対し、管理者がデバッグするための各種プログラム・セットが提供されています。ここでは、このソフトウェアについて簡単に説明します。詳細については、関連するマン・ページ、ヘルプ・システム・ファイル、およびソフトウェアとともに配付されるリリース・ノートを参照してください。savecore、icrash、fru、および sysmon ソフトウェアを組合せて使用することにより、オペレーティング・システムのクラッシュで何が発生したかを見ることができます。

savecore によるクラッシュ・ダンプの保存

savecore の chkconfig の設定によっては、システムがクラッシュしたときに、システム・メモリのイメージを自動的に保存することができます（可能な場合）。savecore は次の 2 つの設定が可能です。

savecore on

システムは、システム・クラッシュ時に、システム・メモリのイメージを自動的に保存しようとしています。イメージは /var/adm/crash/vmcore.N に保存されます。ここで N は、最新のコア・ファイルに割当てられた通し番号です。ダンプが正しく作成された場合、以下に示す icrash ユーティリティでこれを調べることができます。

savecore off

システムはコア・ダンプを保存しません。ただし、icrash は /unix および /dev/swap で実行され、システムがクラッシュした場合に何が起きたのかを通知します。この場合、コア・ダンプはディスク領域上に保存されません。

savecore オプションの現在の状態を調べるには、次のコマンドを入力します。

```
chkconfig | grep savecore
```

デフォルトでは savecore は on です。

詳細については、savecore(1M) マン・ページと「icrash システム・クラッシュ分析ユーティリティ」(33 ページ) で icrash についての説明を参照してください。

icrash システム・クラッシュ分析ユーティリティ

icrash(1M) ユーティリティは、詳細なカーネル情報を読みやすい形式で対話式に作成します。icrash コマンドは、savecore が生成したシステム・クラッシュ・ダンプについてのレポートを作成します。

icrash を実行すると、2 つのレポート・ファイルが作成されます。次の N は、コア・ダンプの境界番号です。

`analysis.N`

コア・ダンプの分析には、`putbuf` ダンプ、`fru` 情報、スタック・トレースなどの有用な情報があります。このアイテムは、システムがクラッシュした場合に何が起きたのかを詳しく説明し、ハードウェアまたはソフトウェアを変更する前に、システムの予備分析を実施するために使用されます。詳細については、`icrash(1M)` マン・ページを参照してください。

`summary.N`

サマリ・レポートは、パニックの付帯情報、クラッシュの時間、および `fru` 情報から構成される `availmon` 用のファイルです。詳細については、`availmon(5)` マン・ページを参照してください。

システム・クラッシュ・ダンプのタイプに応じて、`icrash` はシステムがクラッシュした場合に何が起きたのかを知らせる独自のレポートを作成します。`icrash` コマンドは、動作中のシステムでも、ネームリスト・ファイルと `core` ファイルをコマンド行から指定しても実行できます。ネームリスト・ファイルには、調査の対象となるシステム・メモリ・イメージへのシンボリック・アクセスに必要なシンボル・テーブル情報があります。

`icrash` の各バージョンは、付属する OS のリリースに固有なので、ほかのリリースの OS では動作しません。OS のバージョンが、パッチのレベルも含めてまったく同じでないかぎり、`icrash` をほかの IRIX システムにコピーしないでください。`icrash` を動作中のシステムで実行すると、表示される情報は刻々と変化するので、表示されたときに無意味な結果が出力されることがあります。次に、`icrash` が提供する機能のいくつかを示します。

- システムがパニックを起こした場合に生成されるシステム・クラッシュ・レポート
- それぞれのクラッシュに対して適切なハードウェアで提供されるフィールド交換ユニット (FRU: Field replacement unit) 情報
- カーネル構造のボード情報へのダイレクト・アクセス
- カーネル機能の解放
- コマンドの説明機能 (`icrash` の `help` システムを参照)
- コマンド行の編集と履歴

fru ハードウェア・エラー分析ツール

fru (Field Replacement Unit) コマンドは、Challenge L と XL、Onyx L と XL、および Power Challenge と Power Onyx システムでのみ、フィールド交換ユニットの分析を表示します。このコマンドについては、fru(1M) マン・ページを参照してください。プログラムは、エラー状況にあるハードウェアを分析し、エラーがハードウェアから生じたのかどうかを確認します。この分析は、カーネル・クラッシュ・ダンプで生成されるハードウェアのエラー状態に基づいて実行されます。ハードウェアのエラー状態がダンプされていない場合は、fru の分析は表示されません。

個々のボードが、ハードウェアのエラー状態に基づいて、別々に分析されます。分析が終了すると、信頼性が最も高いレベルのボードが表示されます。現在、分析されるボードとして、IO4、MC3、IP19、および IP21 があります。後のバージョンでは、異なる分析を報告することがあるので、fru のバージョンをその出力からチェックする必要があります。

表示される信頼レベルは fru アナライザが付けた信頼度に基づいています。問題が生じたボードについては fru アナライザがリストします。信頼レベルは数段階しかないことに注意してください。それぞれの確率の意味を十分認識してください。

- 10 % ボードは、ハードウェアのエラー状態であり、分析対象となります。
- 30 % ボードには、低い確率でエラーの可能性がります。
- 40 % ボードには、中程度の確率でエラーの可能性がります。
- 70 % ボードには、高い確率でエラーの可能性がります。
- 90 % ボードには、明らかに問題がります。
- 95 % ボードには、明らかに問題があり、エラーのタイプと一致しています。

複数のボードについて通知される可能性があるため、フィールド・エンジニアは、ボードの交換を慎重に行ってください。たとえば、2つのボードが 10% であると通知された場合は、リストされたこれらのボードが原因であるとはかぎりません。70% またはそれ以上の確率を持つボードがある場合は、そのボードに問題がある確率が高いとみなし、交換する必要があります。30 ~ 40% のボードには疑問があるので、再調査する必要があります。再調査は、同じスロットに挿入された特定のボードのシステム・クラッシュ間の誤動作の頻度に基づいて行います。

fru の目的は、実際のハードウェア上の問題を明らかにすることで、むやみにボードを交換することではありません。適切なシステムは、それぞれのカーネル・コア・ダンプ用の icrash レ

ポート内に fru 分析があります。ボードを交換する前に、フィールド・エンジニアはこの分析を再調査してください。

例 2-1 では、各 fru コマンド出力の例は、1 つのコア・ダンプによってもたらされます。出力内容はさまざまです。

例 2-1 fru コマンド出力

```
>> fru
FRU ANALYZER (2.0.1):
++ PROCESSOR BOARD
++ IP21 board in slot 2: 40% confidence.
++ END OF ANALYSIS

>> fru
FRU ANALYZER (1.6.5):
++ MEMORY BOARD
++ MC3 board in slot 1: 70% confidence.
++ END OF ANALYSIS

>> fru
FRU ANALYZER (1.6.5):
++ CPU slice 3 (CC CHIP)
++ and/or Integral component (A CHIP)
++ on the IP19 board in slot 5: 40% confidence.
++ CPU slice 3 (CC CHIP)
++ and/or Integral component (A CHIP)
++ on the IP19 board in slot 7: 40% confidence.
++ CPU slice 2 (CC CHIP)
++ and/or Integral component (A CHIP)
++ on the IP19 board in slot 9: 40% confidence.
++ CPU slice 3 (CC CHIP)
++ and/or Integral component (A CHIP)
++ on the IP19 board in slot 11: 40% confidence.
++ END OF ANALYSIS

>> fru
FRU ANALYZER (2.0.1): No errors found.
```

システム・ログを表示する sysmon

sysmon ユーティリティを使用することにより、システム・ログ・ファイル (/var/adm/SYSLOG) を表示することができます。8 段階の SYSLOG 優先順位が 4 段階の優先順位レベルに簡略化されています。syslog (3C) マン・ページを参照してください。

sysmon ユーティリティは、デスクトップ・システム・モニタの一部です。このユーティリティは、「ツールチェスト (Toolchest)」の「システム (System)」->「システム・マネージャ (System Manager)」->「システム・パフォーマンス (System Performance)」->「システム・ログの表示 (View System Log)」を選択して実行できます。図 2-3 ような図が表示されます

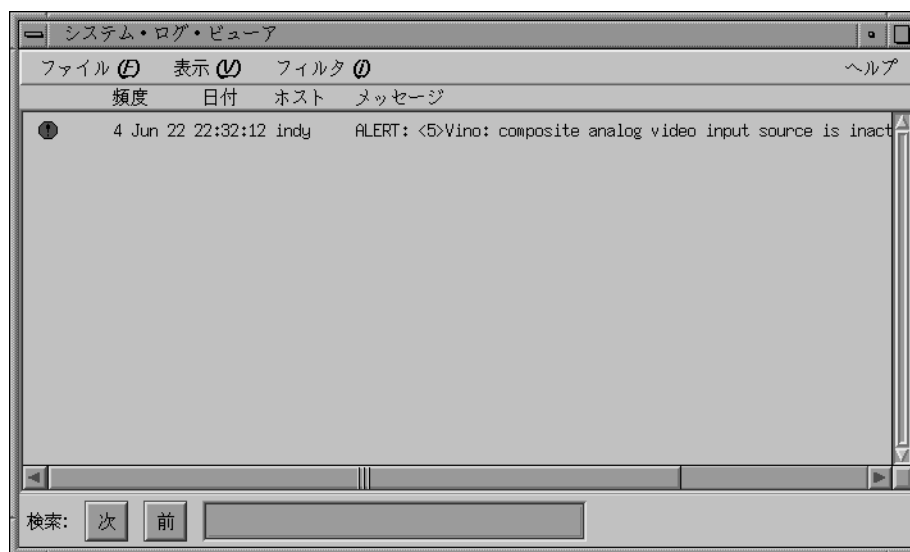


図 2-3 sysmon システム・ログ・ブラウザ

ウィンドウのプルダウン・メニューから、「ファイル」、「表示」および「フィルタ」オプションを選択できます。選択の結果は、\$HOME/.sysmonrc ファイルに保存されます。これらのオプションに関して、詳しい情報が必要な場合は、このウィンドウで参照できるオンライン・ヘルプ、または sysmon のリリース・ノートを参照してください。

次の表 2-2 は、SYSLOG の優先度と簡略化された `sysmon` の優先度の対応表を示します。

表 2-2 `sysmon` の優先度

<code>sysmon</code> の優先度	SYSLOG の優先度	優先順位
CRITICAL	LOG_EMERG	0
CRITICAL	LOG_ALERT	1
ERROR	LOG_CRIT	2
ERROR	LOG_ERR	3
WARNING	LOG_WARNING	4
WARNING	LOG_NOTICE	5
INFO	LOG_INFO	6
INFO	LOG_DEBUG	7

システム使用率を監視する `availmon`

使用率モニタは、`icrash` と FRU アナライザが含まれたソフトウェアパッケージであり、システムの使用率と診断データの収集と配付のための技術基盤を提供します。詳細については、`availmon(5)` マン・ページを参照してください。

`availmon` システムは、システム使用率の情報とクラッシュの診断情報を収集します。使用率の情報は、システムの信頼性と使用率の評価に使用できます。クラッシュ診断情報は、そのままデバッグに役立ちます。

`availmon` ソフトウェアは、システムの起動処理と停止処理に組み込まれています。このソフトウェアは、通常の停止、システム・パニック、システム・ハング、瞬間的電力切断、電力損失を区別できます。システムの稼動時間は、デーモン処理によって概算されます。診断情報は、`icrash(1M)`、`/usr/adm/SYSLOG`、`sysmon(1M)`、`hinv(1M)`、`versions(1M)`、および `gfixinfo(1G)` から収集されます。`availmon` 動作はすべて設定可能です。

提供されているディストリビューション・ソフトウェアの `availmon` は、古いリリースの IRIX では動作しない可能性があります。

管理者は、システム使用率のデータベースを使用することもできます。このデータベースは、SGI のサポートを信頼度の高いサービスにするためのものです。システムにおけるすべての使用率と診断情報は、SGI のデータベースに保持されます。このデータベースにアクセスすると、一般的に信頼できるデータ入手できます。また、接続している特定のシステム固有の履歴データも入手できます。これは、availmon の基本的な機能です。

availmon の登録

SGI Availmon Database にシステムを登録するには、`amregister` コマンドを使用して `availmon` の環境設定を行い、`autoemail` をオンにしてください。

管理者のシステムを登録するには、`root` でログインして次のコマンドを入力します。

```
/usr/etc/amregister -r
```

システムのタイプによっては、システムのシリアル番号を入力する必要があります。詳細については、`amregister(1M)` マン・ページを参照してください。

availmon ソフトウェアは、`chkconfig(1M)` コマンドを使用して有効にできます。このコマンドについては、「`chkconfig` によるソフトウェア構成オプションの確認」で説明します。フラグについては次のとおりです。

availmon ソフトウェア全体のオン / オフを制御します。デフォルトでは、このオプションはオンです。

その他の設定フラグは、`amconfig` ユーティリティで設定します。`amconfig` は `chkconfig` に似ていますが、異なるレコード・ファイルを使用しています。次に 4 つのフラグを示します。

autoemail レポートの自動配付を有効にします。デフォルトでは、このオプションはオフですが、`amregister` によってオンにできます。

hinvupdate デーモンを有効にします。デーモンは、`hinv` および `gfxinfo` で通知される変更をチェックします。デフォルトでは、大きなシステムの場合はオンになり、ほかのシステムではオフになります。

shutdownreason

各システム停止の理由について、システムが特権ユーザに問い合わせるようにします。デフォルトでは、大きなシステムの場合はオンになり、ほかのシステムではオフになります。

tickerd

システム稼動を監視するデーモンを有効にします。デフォルトでは、大きなシステムの場合はオンになり、その他のシステムではオフになります。

また、電子メール・リストの環境ファイルとして

`/var/adm/avail/config/autoemail.list` があります。これは、`availmon` レポートに対して、レポート・タイプ、電子メールの形式、および電子メール・アドレスを制御するために使用されます。電子メール・リストは、`amconfig` コマンドによって編集、および保守されます。デフォルトでは、診断レポートを SGI 社に送信するように設定されています。

複数のシステムが関与しているサイトでは、共通の電子メール環境ファイル

(`/var/adm/avail/autoemail.list`) を設定するために、`amconfig` コマンドを1つのシステムで実行し、関与しているすべてのシステムにそのファイルをコピーできます。その後、各システムで `amregister -r` を実行します。

availmon サイト・ログ・ファイルの設定

1つまたは複数のシステムのサイト・ログ・ファイルが必要な場合は、擬似電子メールのエリアスを作成できます。このエリアスは、利用率レポートを `amreceive` に通知します。`amreceive` の出力は、サイト・ログ・ファイルに追加されます。ここでの手順では、すべてのシステムを登録する前に実行しなければなりません。初期状態での利用率レポートが登録時に送信されるためです。

`availmon` をセットアップし、`amreport` を実行すると、システム全体の利用率の統計が表示されたり、各システムの利用率レポートが表示されます。サイト・ログ・ファイルを入力して実行すると、すべてのシステムに対して全体的な使用可能率の統計が表示されたり、各システムの利用率レポートが表示されます。

手順2-3で、サイト・ログ・ファイルを作成する方法を説明します。

手順 2-3 サイト・ログ・ファイルの作成

サイト・ログ・ファイルを作成するには以下の手順に従ってください。

1. 単一のシステムで電子メールのエリアスを作成し、すべての使用率レポートを `amreceive` に通知します。たとえば、サイト・ログ・ファイルが `/disk/amrlog` の場合は、次の行をメール・サーバ・システムの `/etc/aliases` ファイルに追加します。

```
amrlog: "| /var/adm/avail/amreceive >> /disk/amrlog"
```

次に、`newaliases` コマンドを実行し、この電子メールのエリアスを設定します。

2. 次に、メール・サーバ・システムで、`amconfig` コマンドを実行し、標準的な電子メール・リストを構成します。上記の例では、次の項目を追加します。

```
availability(text): amrlog
```

このシステムの `/var/adm/avail/config/autoemail.list` をサイトの残りのシステムにコピーします。

3. 39 ページの「`availmon` の登録」の説明に従って、`amregister` を実行し、すべてのシステムに登録します。
4. 次のコマンドを実行すると、関与しているすべてのシステムに関する、全体的な統計や各システムの統計、および使用率レポートを表示できます。

```
amreport -s /disk/amrlog
```

availmon の管理

`availmon` の管理に関する 3 つの例を説明します。1 つは、一般のユーザがレポートを自動的に送信する場合、残りの 2 つはセキュリティ上の制限があるサイトで内部レポートを送信する場合としない場合を説明します。

自動レポートによる `availmon` の使用

`availmon` が単一のシステムにインストールされている場合は、インストール後にシステムを再起動します。次に、電子メール・リストの登録、設定のための引数を付けずに `amregister` を実行します。この結果、`autoemail` がオンになり、終了時にすべての設定アドレスに登録レポートが送信されます。システムに IP19、IP21、IP22、または IP25 プロセッサがない場合は、システムのシリアル番号を入力するように指示されます。

`shutdownreason` と `tickerd` は、オン / オフの切替えが可能です。デフォルトの `autoemail.list` を次に示します。

```
availability (compressed, encrypted) :  
availability (compressed) :  
availability (text) :  
diagnosis (compressed, encrypted) : availmon@csd.sgi.com  
diagnosis (compressed) :  
diagnosis (text) :
```

次のエントリを追加することもできます。

```
availability (text) : local_sysadmin  
diagnosis (compressed, encrypted) : local_support
```

上記のオプションな入力部分では、*local_sysadmin* と *local_support* をそれぞれシステム管理者と日本シリコングラフィックス株式会社サポート担当者の電子メール・アドレスに置換えてください。電子メールに暗号データを入れることが法律で禁止されているサイトの場合は、“(compressed,encrypted)” 中のアドレスを“(compressed)” にします。

セキュリティ上の制限があるサイトで内部レポートを送信する場合の `availmon` の使用

セキュリティ上の制限があるサイトでは、以下の手順に従って `availmon` をセットアップした後で使用します。セットアップの手順は、39 ページの「`availmon` の登録」と同じです。ただし、サイト外のアドレスについては削除します。

システム管理者は `availmon` レポートを受信すると、再起動したシステムの診断レポートの `/var/adm/crash/diagreport` をチェックし、問題がある可能性のデータを除去し、`amsend` を使用してフィルタをかけたレポートを `availmon@csd.sgi.com` および日本シリコングラフィックス株式会社のサポート・アドレスに送ります。診断レポートに `icrash`、`syslog`、`hinv`、`versions`、または `gfxinfo` のデータのいずれかが含まれる場合には、次のコマンドを実行してレポートを送ります。

```
amsend -i -z -x availmon@csd.sgi.com ...
```

レポートにこのようなデータが含まれない場合には、次のコマンドを実行します。

```
amsend -d -z -x availmon@csd.sgi.com ...
```

電子メールに暗号データを入れることが法律で禁止されているサイトでは、コマンド・ラインから `-x` を削除します。

セキュリティ上の制限があるサイトでレポートを送信しない場合の availmon の使用

外部へのレポート送信ができないサイトでは、availmon を使用する上で特別な操作を必要としません。ただし、IP19、IP21、IP22、または IP25 プロセッサを使用しないプラットフォームでは、amregister を動作させ、**autoemail** をオフにします。この結果、これらのシステムで作成されたレポートは、自動的に送信されません。**shutdownreason** と **tickerd** の 2 つのオプションは、ユーザの選択によって、オン / オフを切替えることができます。

システムが再起動されても外部レポートは送信されないの、システムが停止した場合に、システム管理者はレポート・ファイルをチェックし、理由を確認する必要があります。チェックする前にシステム・クラッシュが起こった場合は、古いレポートは新しいレポートによって上書きされます。ただし、コア・ダンプと icrash レポートは、削除しないかぎり残ります。この結果、セキュリティ上の制限があるサイトでは、内部レポートの送信をお勧めします。

amsend を使用して、診断レポートを SGI 社に送信できます。42 ページの「セキュリティ上の制限があるサイトで内部レポートを送信する場合の availmon の使用」を参照してください。その他の手段として、amconfig を実行し、標準的な電子メール・リストを構成する方法があります。この方法では、レポートを送信する必要がある場合に、amnotify を使用してそのリストに従ってレポートを送信できます。

availmon レポート

availmon が生成するレポートには、使用率と診断に関する 2 種類のレポートがあります。

使用率レポートの内容は、開始時間、停止時間、停止理由、稼動時間、再起動時間、およびシステム・クラッシュの原因と思われる事項の要約（該当事項がある場合）です。次に、標準的な使用率レポートを示します。

```
----- whizkid -----
Report Version ..... 0.1
Internet Address ..... whizkid
Reason for Shutdown .... Hang
Started at ..... Mon Oct  3 16:56:08 1994
Stopped at ..... Unknown instant
Uptime ..... 4304 minutes (2 days 23 hrs 44 mins)
-----
Press <enter> to display summary ...
```

<Enter> キーを押すと、次のような情報が表示されます。

```
===== SUMMARY for whizkid =====
Controlled Shutdowns ... 0
Hangs ..... 1
Panics ..... 0
Average Uptime ..... 2189 minutes (1 day 12 hrs 29 mins)
Least Uptime ..... 74 minutes (1 hr 14 mins) (*)
Most Uptime ..... 4304 minutes (2 days 23 hrs 44 mins)
Availability ..... 0.7870%
Logging started at .... Mon Oct  3 16:56:08 1994
System has been up for . 74 minutes (1 hr 14 mins)
Last boot at ..... Tue Oct 24 23:03:44 1995
=====
```

診断レポートには、使用率レポートとしての情報も含まれます。また、FRU アナライザの結果を含む icrash 分析レポート、重要な syslog メッセージ、およびシステムのハードウェアおよびソフトウェアの構成とバージョン情報が追加されています。

使用率の情報は、/var/adm/avail/availlog に永久に保存されます。/var/adm/availにあるファイルは、availmon によって保持されているので、削除、修正、移動などの操作は行わないでください。最新の使用率および診断レポートは、/var/adm/crash/availreport と /var/adm/crash/diagreport に保存されるので、コア・ダンプと同じように取扱ってください。

amsend による availmon レポートの送信

availmon レポートの送信を設定するには、自動と手動の 2 つの方法があります。

自動送信を選択すると、各タイプのレポートに対して、任意の数の受信者を設定できます。診断レポートは、SGI availmon database と日本シリコングラフィックス株式会社のサポート担当者に送信し、使用率レポートは、ローカル・システムの管理グループに送信するように設定しておくことをお勧めします。すべてのレポートのコピーをローカル・ログ・アカウントに送信することも可能です。

手動送信を選択した場合は、/var/adm/crash ディレクトリに 2 種類のレポートが作成されます。このレポートを編集したり、フィルタリングしてから、amsend コマンドを使用して正式なレポートとして送信できます。

availmon ソフトウェアは、データを圧縮し、エンコードするように設定できます。受信者は、amreceive コマンドを使用して、デコードおよび圧縮解除を行い SGI のデータベースにデータを保存します。可能であれば、サイトでデータを暗号化することをお勧めします。

amreport による availmon レポートの表示

amreport (1M) ユーティリティを使用して、availmon レポートをレビューし、使用率に関する統計情報を得ることができます。このプログラムは、ローカル・システムの使用率ログ・ファイルや、別のシステムの使用率の統計レポート（サイト・ログ・ファイルなど）を処理できます。

amreport ユーティリティは、統計レポートと使用率レポートを階層的に表示します。すべてのシステムに関する全体的な統計、すべてのシステムの統計表、各システムの統計、各システムのすべての再起動時の情報、および各システムの使用率レポートを表示します。入力がローカルなログ・ファイルである場合は、すべてのシステムの統計は表示できません。このユーティリティについての詳細は、amreport (1M) マン・ページを参照してください。

システムの起動、停止および実行レベル

この章では、システムの起動と停止の操作手順、デフォルトの実行レベルの設定方法、および個別の環境設定の方法について説明します。主な内容は次のとおりです。

- 47 ページの「システムの起動」
- 48 ページの「マルチユーザ・モードでのシステムの停止」
- 50 ページの「シングルユーザ・モードでのシステムの停止」
- 50 ページの「IRIX オペレーティング・システムの実行レベル (システム状態)」
- 59 ページの「/etc/inittab によるシステムの電源切断」

システムの起動

IRIX システムを起動するには手順 3-1 に従ってください。

手順 3-1 システムの起動

1. すべてのケーブル（電源、ディスプレイ・モニタ、キーボードなど）が正しく接続されていることを確認します。ワークステーションやサーバのケーブル接続についての詳細は、システムのユーザ・ガイドやハードウェア・マニュアルを参照してください。
2. ディスプレイ・モニタ（またはコンソール端末）とコンピュータの電源を入れます。

システムを初めて起動する場合、アカウント（root、ゲスト、その他）に NULL パスワードが割当てられていると、「Would you like to secure this machine now? (今すぐこのマシンのセキュリティを保護しますか?)」というメッセージが表示されます。「No (いいえ)」を選択すると、変更は加えられません。「Yes (はい)」を選択すると、簡単なプロンプトが表示されます。

コンピュータは電源投入時に診断を実行し、著作権の情報とシステムの起動に関する情報を表示します。これらのメッセージは、コンソール画面またはサーバの診断端末（第1シリアル・ポートに接続されたASCII端末）の画面に表示されます。また、見逃した場合に備えて /var/adm/SYSLOG ファイルにも書込まれます。

電源が切れたり、予期しない停止が起こったためにシステムを再起動する場合は、SCSIバスやほかのハードウェアに関するエラー・メッセージが表示されることがあります。これは、ディスクが稼働速度に達するのに時間がかかったり、SCSIバスがリセットするのに時間が必要なために一時的に表示されるメッセージです。30秒間待ってから、オペレーティング・システムを再起動してください。

オペレーティング・システムによってファイルシステムのチェックが必要であると判断された場合は、*fsck* コマンドを使用してチェックが行われます（EFSのみ）。*fsck* コマンドは、オペレーティング・システムがファイルシステムをマウントする前に問題を検出し、それを修正します。*fsck* は、停電などでシステムが正常に停止されなかった場合に実行されます。*fsck* の使用方法についての詳細は、『IRIX Admin: Disks and Filesystems』および *fsck(1M)* マン・ページを参照してください。*fsck* は予防的な処置であり、これが実行されたとしても必ず問題があるわけではありません。

システムはマルチユーザ・モードになり、ログインできる状態になります。システムは常に実行させておいてください。IRIX オペレーティング・システムは、常に実行できる状態にしておくとも最も効率的です。こうすることにより、予定しておいた操作や、ハウスキーピング・プロセスをスケジュールどおりに実行できます。

マルチユーザ・モードでのシステムの停止

マルチユーザ・モードでシステムを停止するには手順3-2に従ってください。

手順 3-2 マルチユーザ・モードでのシステムの停止

1. *who* (1) コマンドを使用して、オペレーティング・システムにログインしているユーザがいるかどうかを調べます。

who

2. *wall* (1M) コマンドを実行し、システムを停止する旨をすべてのユーザに通知します。

wall

次の例のようなメッセージを入力します。

ビルの電源システムに問題が発生しました。
10 分後にシステムを停止しますので、ログオフするように
してください。ご迷惑をおかけしますが、よろしくお願いします。
システム管理

メッセージの入力が完了したら、<Ctrl>+D キーを押します。メッセージはシステムを使用
しているすべてのユーザに送信されます。ユーザは次のようなメッセージを受信します。

Indy3 10月17日 17時02分27秒 上の root ttyq3 からのブロードキャスト・メッセージで
す。

ビルの電源システムに問題が発生しました。
10 分後にシステムを停止しますので、ログオフするように
してください。ご迷惑をおかけしますが、よろしくお願いします。
システム管理

3. /etc/shutdown コマンドを入力します。

```
/etc/shutdown -y -i0 -g600
```

上記のコマンドは、ユーザがクリーンアップとログオフを行うために 10 分 (600 秒) のグ
レース・ピリオドを与えています。その他のフラグは、システムを完全に停止すること (-i0)、
システムの停止に関するプロンプトにはすべてユーザが yes (-y) と答えると想定することを
示しています。この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。

シャットダウン開始。1998 年 5 月 8 日 (金曜日) 17 時 10 分 59 秒 JST
hostname day time year 上の root ttyq3 からのブロードキャスト・メッセージです。
システムは 600 秒後にシャットダウンします。すぐにログオフしてください。

10 分経過すると、次のメッセージが表示されます。

```
The system is shutting down. Please wait.  
Okay to power off the system now. Press any key to restart.
```

コマンド・モニタ・プロンプトまたは「System Maintenance」メニューが表示されます。
ハードディスクを破損する危険性があるので、ワークステーションの電源は必ずコマンド・
モニタ・プロンプトまたは「System Maintenance」メニューが表示されてから切つてくだ
さい。

4. ワークステーションの電源を切ります。

システム停止についての詳細は、halt(1M) および shutdown(1M) マン・ページを参照して
ください。システムの停止は、障害がある場合、またはソフトウェアやハードウェアの変更が必
要な場合にのみ行ってください。IRIX システムは、ユーザがログインしていないとき、またはシ
ステムが使用されていないときも常に継続して動作するように設計されています。

シングルユーザ・モードでのシステムの停止

システムがシングルユーザ・モードになっている場合は、手順 3-3 に従ってください。

手順 3-3 シングルユーザ・モードでのシステムの停止

1. `shutdown` コマンドを使用してシステムの電源を切り、ファイルシステムの整合性を保持します。`root` でログインし、次のコマンドを入力します。

```
shutdown -y -i0 -g0
```

`-y` はすべての質問に `yes` と答えることを示し、`-i0` は状態 0 (「System Maintenance」メニュー) に移行することを示し、`-g0` はグレース・ピリオドがないこと (0 秒) を示しています。

次のようなコメントが表示されます。

```
シャットダウン開始。1998年 5月 8日 (金曜日) 17時 10分 59秒 JST  
hostname day time year 上の root ttyq3 からのブロードキャスト・メッセージです。  
システムは 600 秒後にシャットダウンします。すぐにログオフしてください。
```

システムはすべてのサービスを停止し、コマンド・モニタ・プロンプトまたは「System Maintenance」メニューが表示されます。

ハードディスクを破損する危険性があるので、ワークステーションの電源は必ずコマンド・モニタ・プロンプトまたは「System Maintenance」メニュー、またはシステムの電源を切ってもよいというメッセージが表示されてから切ってください。

2. ワークステーションの電源を切ります。

IRIX オペレーティング・システムの実行レベル (システム状態)

IRIX システムは、シングルユーザ・モードまたはマルチユーザ・モードで実行できます。シングルユーザ・モードでは、システム上でごく少数のプロセスしか実行できず、グラフィックスは使用できません。また、単一のログインのみが許可されます。マルチユーザ・モードでは、複数のログイン・セッション、複数のファイルの同時オープン、多くのバックグラウンド・デーモンを含む複数のプロセスの実行などが可能です。

システムがシングルユーザ・モードになるかマルチユーザ・モードになるかは、`init` プログラムによって制御されます。システムの状態にはそれぞれラベル (数値または文字) が割り当てられます。たとえば、停止状態は状態 0、シングルユーザ・モードは状態 `s` です。

マルチユーザ・モードのラベルの種類は非常に複雑です。マルチユーザ・モードには、ログインの数に制限のない状態や制限がある状態など、さまざまな状態があるためです。それぞれの状態には、すべて異なる番号が割当てられます。

システムの状態は、`/etc/inittab` ファイルによって制御されます。このファイルには、システムの状態とそれに対応するラベルがリストされています。

システムを標準的なマルチユーザ・モード (init 状態 2) にすると、次の設定が行われます。

- ファイルシステムがマウントされます。
- スケジュール済みのタスクを実行する cron デーモンが起動されます。
- ネットワーク・サービスが起動されます (オンに設定されている場合のみ)。
- uucp シリアル回線ネットワーク機能が利用できるようになります。
- lp パッケージがインストールされている場合は、そのスプーリング機能とスケジューリング機能が利用できるようになります。
- ユーザがログインできる状態になります。

マルチユーザ・モードであっても、すべてのシステム機能が実行できるわけではありません。たとえば、ソフトウェアのインストールやファイルシステムのチェックなどの作業はミニルートを必要とするので、シングルユーザ・モードで実行する必要があります。

システムの状態を表す同義語には、次のように多くの種類があります。

- init 状態
- 実行状態
- 実行レベル
- 実行モード
- システム状態

同様に、それぞれのシステム状態を表す用語も複数あります。たとえば、シングルユーザ・モードを表す用語には次の 3 つがあります。

- シングルユーザ

- シングルユーザ・モード
- 実行レベル 1

に、出荷時のオペレーティング・システムで使用できる実行レベルを示します。ユーザが独自の実行レベルを定義することもできます。

表 3-1 システム状態

実行レベル	説明
0	電源 OFF 状態
1, s, または S	シングルユーザ・モード。ソフトウェア・ユーティリティのインストールと削除、ファイルシステムのバックアップ作成とリストア、およびファイルシステムのチェックに使用される。この状態では root 以外はすべてアンマウントされ、コンソール関連以外のユーザ・プロセスは無効となる。
2	マルチユーザ・モード。システムの通常の動作モード。デフォルトでは、root (/) と user(/usr) のファイルシステムがマウントされている。システムの電源を入れるとマルチユーザ・モードになる。
6	再起動モード。システムを一度終了し、再起動させる。システム構成パラメータを変更するときに使用する。

init によるシステム状態の制御

init は、起動時にシステムによって最初に生成される一般的なプロセスです。このプロセスは、各実行レベルに存在するプロセスを詳細に定義した `/etc/inittab` ファイルを読取ります。

マルチユーザ・モード (実行レベル 2) では、*init* は `/etc/inittab` から現在の実行レベルに一致する 2 というタグを持つエントリをスキャンし、そのタグを含む行の 3 番目のコロン (:) 以降のコマンドを実行します。詳細については、`inittab(4)` マン・ページを参照してください。

例 3-1 に、`/etc/inittab` ファイルの例を示します。

例 3-1 `/etc/inittab` ファイル

```
is:2:initdefault:
fs::sysinit:/etc/bcheckrc </dev/console >/dev/console 2>&1
```

```

mt::sysinit:/etc/brc </dev/console >/dev/console 2>&1
s0:06s:wait:/etc/rc0 >/dev/console 2>&1 </dev/console
s1:1:wait:/etc/shutdown -y -iS -g0 >/dev/console 2>&1 </dev/console
s2:23:wait:/etc/rc2 >/dev/console 2>&1 </dev/console
s3:3:wait:/etc/rc3 >/dev/console 2>&1 </dev/console
or:06:wait:/etc/umount -ak -b /proc,/debug > /dev/console 2>&1
of:0:wait:/etc/uadmin 2 0 >/dev/console 2>&1 </dev/console
RB:6:wait:echo "The system is being restarted." >/dev/console
rb:6:wait:/etc/uadmin 2 1 >/dev/console 2>&1 </dev/console
#

```

実際の `/etc/inittab` ファイルは非常に長いため、上記の例は編集して簡略にしてあります。`/etc/inittab` が誤って削除され、システムの停止時に検索されない場合、`init` はシングルユーザ状態 (`init s`) になります。シングルユーザ状態である間、`/usr` ファイルシステムはマウントされたままになり、`init` を使用せずに起動されたプロセスもそのまま実行されます。このような場合には、実行状態を変更する前に `/etc/inittab` を置換えてください。`inittab` の各行の形式は以下のとおりです。

id:level:action:process

- *id* は各エントリを一意に識別する 1 個または 2 個の文字で構成されます。
- *level* は 0 以上の数字 (0 ~ 6) と文字 (a, b, c) から構成されるタグで、*action* が実行されるレベルを指定します。*level* が指定されていないと、*action* はすべてのレベルで実行できます。
- *action* には次のいずれかを指定できます。

– **sysinit**

`init` がシステム・コンソール (Console Login) にメッセージを送信する前に、*process* を実行します。

– **bootwait**

システムの起動後に `init` がはじめてシングルユーザ状態からマルチユーザ状態に移行したときに、*process* を起動します。**initdefault** が 2 に設定されている場合は、システムの起動の直後にプロセスが実行されます。`init` はプロセスを起動し、プロセスの終了を待ちます。プロセスの終了後に再起動はされません。

– **wait**

level に到達したときに、*process* を起動し、それが終了するのを待ちます。

- **initdefault**

init が起動されると、*level* になります。この *action* の *process* フィールドは意味を持ちません。

- **once**

process を一度だけ実行します。終了した後は再び実行しません。

- **powerfail**

コンピュータの電源が直接切断された場合に *process* を実行します。

- **respawn**

process が実行されていない場合にそれを起動して終了するのを待った後、別のプロセスを起動します。

- **ondemand**

respawn と同じ意味ですが、*level a*、*b*、*c* に対してのみ使用します。

- **off**

level 中で、*process* を強制終了 (kill) するか無視します。

- *process* は任意の実行可能プログラム (シェル・スクリプトも含む) です。
- # を使用して、行の終わりにコメントを追加できます。*init* は、# で始まる行をすべて無視します。

実行レベルを変更するとき、*init* はそのレベルで指定されていないプロセスをすべて強制終了します。

システムの停止からマルチユーザ状態への移行

通常システムが起動され、実行されるとマルチユーザ・モードになります。ユーザが IRIX システムの全機能を利用できるのは、システムがマルチユーザ・モードの場合だけです。

デフォルトでは、電源を入れるとシステムはマルチユーザ・モードになります。また、*inittab* ファイル内の **initdefault** 行を変更して、デフォルト設定を変更することもできます。実際には次の順序でマルチユーザ状態になります。表 3-1 も参照してください。

1. オペレーティング・システムがロードされ、`init` によってシステムの初期化が開始されません。
2. `/etc/rc2` の手順に従って実行レベルの変更準備が行われます。
3. 使用可能な端末回線上で `getty` プロセスを起動して、一般ユーザがシステムを利用できる状態にします。ネットワークに接続されたシステムの場合は、ネットワークを介してアクセスできる状態にします。

`init` プロセス：初期化

PROM ハードウェアに常駐する特殊な起動プログラムによってオペレーティング・システムが物理メモリに最初にロードされると、`init` プロセスが生成されます。`init` は `/etc/inittab` をスキャンし、**sysinit** タイプのエントリを検索します。

```
fs::sysinit:/etc/bcheckrc </dev/console >/dev/console 2>&1
mt::sysinit:/etc/brc </dev/console >/dev/console 2>&1
```

これらのエントリは順番に実行され、システムの初期化に必要な作業を行います。各エントリは、`/dev/console` を標準 I/O として使用するの注意してください。この結果、システムがマルチユーザ状態になる前に、システム・コンソールと交信します。

`init` プロセス：実行レベルの準備

次に、システムは特定の実行レベルに移行します。まず、`init` は `/etc/inittab` をスキャンし、**initdefault** タイプのアクションを指定するエントリを検索します。このようなエントリが検索された場合は、次に実行するエントリを選択するためのタグとして、そのエントリの実行レベルを使用します。次に示す `/etc/inittab` の例では、**initdefault** エントリの実行レベルは 2 (マルチユーザ状態) です。

```
is:2:initdefault:
s2:23:wait:/etc/rc2 >/dev/console 2>&1 </dev/console
co:23:respawn:/etc/gl/conslog
t1:23:respawn:/etc/getty -s console ttyd1 co_9600 #altconsole
t2:23:off:/etc/getty ttyd2 co_9600 # port 2
t3:23:off:/etc/getty ttyd3 co_9600 # port 3
t4:23:off:/etc/getty ttyd4 co_9600 # port 4
```

上記の例に示したほかのエントリは、システムをマルチユーザの実行レベルにするために必要なアクションを指定しています。まず `/etc/rc2` が実行され、`/etc/rc2.d` ディレクトリ内の

Sで始まるすべてのファイルが実行されます。詳細については、57ページの「実行レベルのディレクトリ」を参照してください。/etc/rc2が実行されると、主に次のような操作が実行されません。

- ファイルシステムの設定とマウントを行います。
- cron デーモンを起動します。
- uucp を利用可能にします。
- ライン・プリンタ (lp) システムがインストールされている場合は、それを利用可能にします。
- アカウンティングがインストールされオンに設定されている場合は、それを起動します。
- ネットワークがインストールされオンに設定されている場合は、それを起動します。
- sar がインストールされオンに設定されている場合は、それを起動します。
- メール・デーモンを起動します。
- システム・モニタを起動します。

init プロセス: getty

マルチユーザ・モードへ移行する最終手順は、getty プロセスの実行です。

init はコンソール用の getty を起動後、指定したポートに接続された端末回線上で getty を起動します。

上記の処理によって完全なマルチユーザ環境が確立され、一般ユーザがログインできる状態になります。

システムの実行レベルの変更

システム管理者は、telinit(1M) コマンドを使用して init に /etc/inittab のエントリを実行させ、実行レベルを変更できます。この結果、主要なプロシージャ (shutdown、/etc/rc0、/etc/rc2 など) が実行され、新しい実行レベルが初期化されます。telinit コマンドは、single、multi、reboot のスクリプトで実行されます。

たとえば、シングルユーザ・モードからマルチユーザ・モードに移行するには、次のコマンドを入力します。

multi

詳細については、`multi(1M)`、`single(1M)`、および `reboot(1M)` マン・ページを参照してください。

シングルユーザ・モードなどのように下位の実行レベルに変更するには、58 ページの「shutdown コマンドによるマルチユーザ・モードからシングルユーザ・モードへの変更」で説明されている方法を使用することをお勧めします。これで新しい実行レベルができました。実行レベルが 1 (シングルユーザ・モード) の場合、システム管理者は操作を続行できます。

実行レベルのディレクトリ

実行レベル 0、2、3 にはそれぞれ `rc0.d`、`rc2.d`、`rc3.d` というディレクトリがあり、各ディレクトリには別のレベルへ移行するときの実行するファイルが含まれています。これらのディレクトリ内のファイルはすべて `/etc/init.d` ディレクトリのファイルにリンクされています。実行レベルのファイル名は次のどちらかの形式になります。

sNNname

kNNname

ファイル名は 3 つの部分で構成されます。

S または **K** 最初の文字は、新しい実行レベルでプロセスを起動 (**start**) するか、または強制終了 (**kill**) するかを示します。

NN 次の 2 つの文字は 00 から 99 までの数字です。これらの数字は、プロセスが起動される順序 (**S00**、**S01**、**S02** など) または強制終了される順序 (**K00**、**K01**、**K02** など) を示します。

name ファイル名の残りの部分は、該当するファイルがリンクされる `/etc/init.d` のファイル名です。

たとえば、`init.d` の `cron` ファイルは、`rc2.d` の `S75cron` および `rc0.d` の `K15cron` ファイルにリンクされています。`S75cron` は、実行レベル 2 (**init 2**) になると **start** オプション

付きで実行され (`sh S75cron start`)、`K75cron` は、実行レベル 0 (`init 0`) になると `stop` オプション付きで実行されます (`sh K70cron stop`)。このシェル・スクリプトは、`start` オプション付きで実行される場合は `/sbin/cron` を起動し、`stop` オプション付きで実行される場合は `cron` プロセスを強制終了します。

実行レベル・ファイルの変更

実行レベル・ファイルはシェル・スクリプトであるため、内容を参照してその機能を確認できます。これらのファイルはユーザが変更することもできますが、将来のリリースではスクリプトの内容が変更される可能性もあるので、独自のファイルを新規に作成しておくことをお勧めします。独自のスクリプトを作成する場合は、次の規則に従います。

- スクリプトは `/etc/init.d` に格納します。
- 上記の命名規則に従って、適切な実行レベルのディレクトリ内のファイルと、スクリプトとのシンボリック・リンクを確立します。つまり、`/etc/init.d` 内のスクリプト・ファイルと、スクリプトを起動および停止する実行レベルに該当するディレクトリ内の `SNNfile` および `KNNfile` とのシンボリック・リンクを確立します。通常は、`/etc/rc2.d` と `/etc/rc0.d` がこれに相当します。
- スクリプトが `start` オプションまたは `stop` オプション、あるいはその両方を認識し、起動と停止を実行できることを確認します。

適切なディレクトリから既存のスクリプトをコピーし、必要な変更を加える方が簡単な場合もあります。スクリプトの記述方法については、`/etc/init.d`、`/etc/rc0.d`、および `/etc/rc2.d` 内のスクリプトとリンクを参照してください。

`shutdown` コマンドによるマルチユーザ・モードからシングルユーザ・モードへの変更

シングルユーザ・モードでの `root` ファイルシステムのバックアップなど、管理機能の実行が必要な場合は、手順 3-4 のように `shutdown` コマンドを使用します。

手順 3-4 マルチユーザ・モードからシングルユーザ・モードへの変更

シングルユーザ・モードでシステムを停止するには、次の 3 つの方法があります。

1. `shutdown -i1` コマンドを入力します (推奨する方法)。最初の警告メッセージから最後の警告メッセージまでのグレース・ピリオドを指定するには、`-g` オプションを使用します。
2. `single` コマンドを入力します。このコマンドは、シングルユーザ・モードに切替え、`getty` プロセスをオフにするシェル・スクリプトを実行します。
3. `init 1` コマンドを入力します。このコマンドは、`init` プロセスにテーブルをスキャンさせます。テーブルで最初に見つかった `s1` エントリに従って、`init` は停止プロセスを開始します。

このプロセスは、`/etc/rc0` プロシージャを呼出し、`/etc/rc0.d` ディレクトリ内のすべてのファイルを実行します。次に、`shutdown` コマンドの機能を示します。

- 開いているすべてのファイルを閉じ、すべてのユーザ・プロセスを停止します。
- すべてのデーモンとサービスを停止します。
- すべてのシステム・バッファをディスクに書込みます。
- `root` 以外のすべてのファイルシステムをアンマウントします。

次の行は、`/etc/inittab` ファイルにあるシングルユーザ処理用のエントリの例です。

```
s1:1:wait:/etc/shutdown -y -iS -g0 >/dev/console 2>&1 </dev/console
```

システムがシングルユーザ・モード環境の場合は、必要な管理作業を実行できます。

/etc/inittab によるシステムの電源切断

システムの電源は、`/etc/inittab` の次のエントリによって切断されます。

```
s0:06s:wait:/etc/rc0 >/dev/console 2>&1 </dev/console
of:0:wait:/etc/uadmin 2 0 >/dev/console 2>&1 </dev/console
```

システムの停止は余裕を持って行ってください。`powerdown` コマンドか `init 0` コマンドを使用するか、直接 `/etc/shutdown -i0` コマンドを起動して停止できます。

いずれの場合も、`/etc/shutdown` と `/etc/rc0` が呼出され、すべてのユーザ・プロセス、デーモン、およびその他のサービスのクリーンアップと停止処理が行われ、ファイルシステムがアンマウントされます。最後に `/sbin/uadmin` コマンドが呼出され、最後の手順であるシステムの電源の物理的な切断がファームウェアの制御下にあることが示されます。

IRIX オペレーティング・システムの構成

この章では、システムを使用するためにカスタマイズが必要な設定、および編集の必要があるファイルについて説明します。ここでリストされているすべての設定や名前を変更する必要はありません。使用する目的に応じて、システムをカスタマイズすることができます。この章では、以下について説明します。

- 61 ページの「システム構成の確認」では、どのハードウェアとソフトウェアがインストールされ、アクティブであるのか、および現在のシステム・ソフトウェアの設定に関する情報を提供します。
- 「ソフトウェアの構成」(73 ページ) では、基本的なシステムの設定とオプションの変更についての情報を提供します。
- 「システムのパーティション分割」(89 ページ) では、SGI 3000 Series システムの場合、そのシステム内を複数に分割するための情報を提供します。

システム構成の確認

IRIX システムでは、システムのハードウェアおよびソフトウェアの構成を確認するための 2 つのコマンドが用意されています。hinv(1M) コマンドはハードウェアの接続情報を、versions(1M) コマンドはソフトウェアのインストール情報をそれぞれ表示します。このほか、グラフィックス・ハードウェア、システム名、設定されているプリンタ、および基本的なシステム設定について報告するコマンドがあります。

hinv によるインストール済みハードウェアの確認

hinv コマンドは、システムのハードウェア接続情報を起動時に表示します。このコマンドは、コマンド・モニタ (PROM) またはシステムのシェル・プロンプトから実行できます。表示される情報は、プロセッサの種類、メイン・メモリのサイズ、ディスク、テープ・ドライブおよびその他のデバイスなどに関するものです。

例 4-1 に、IRIX システムのシェル・プロンプトから実行した `hinv` コマンドの出力を示します。詳細については、「`/hw` ディレクトリ内のインストール済みハードウェアの確認」(67 ページ) と `hwgraph(4D)` マン・ページを参照してください。

例 4-1 `hinv` コマンド出力

`-v` オプションと `-m` オプションを指定した場合、`hinv` コマンドは詳細情報を出力します。これにはボード名、パーツ番号、バーコード番号が含まれ、さらに各ボードの物理位置が `/hw` の下のパスとして表示されます。

```
% hinv -vm
```

```
Location: /hw/module/001c10/node
          IP35 Board: barcode KHD544      part 030-1604-001 rev -G
Location: /hw/module/001c10/Ibrick/xtalk/14
          IBRICK Board: barcode KJX216     part 030-1557-002 rev -F
Location: /hw/module/001c10/Ibrick/xtalk/15
          IBRICK Board: barcode KJX216     part 030-1557-002 rev -F
Location: /hw/module/001c10/Ibrick/xtalk/15/pci/4
[missing info: Not present on interface] Laser  unreadable
Location: /hw/module/001c13/node
          IP35 Board: barcode KHD546      part 030-1604-001 rev -G
Location: /hw/module/001c13/Xbrick/xtalk/14
          XBRICK Board: barcode KHP021     part 030-1565-001 rev -K
Location: /hw/module/001c13/Xbrick/xtalk/15
          XBRICK Board: barcode KHP021     part 030-1565-001 rev -K
Location: /hw/module/001c16/node
          IP35 Board: barcode KHD136      part 030-1604-001 rev -F
Location: /hw/module/001c21/node
          IP35 Board: barcode KHD152      part 030-1604-001 rev -G
Location: /hw/module/001c24/node
          IP35 Board: barcode KHD106      part 030-1604-001 rev -G
Location: /hw/module/001c29/node
          IP35 Board: barcode KHD079      part 030-1604-001 rev -G
Location: /hw/module/001c32/node
          IP35 Board: barcode KHD135      part 030-1604-001 rev -G
Location: /hw/module/001c35/node
          IP35 Board: barcode KHD191      part 030-1604-001 rev -G
Location: /hw/module/001r19/router
          ROUTER Board: barcode KRE126    part 030-1596-001 rev -F
Location: /hw/module/001r27/router
          ROUTER Board: barcode KRE084    part 030-1596-001 rev -F
32 400 MHZ IP35 Processors
CPU: MIPS R12000 Processor Chip Revision: 3.5
```

FPU: MIPS R12010 Floating Point Chip Revision: 0.0
CPU 0 at Module 001c10/Slot 0/Slice A: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 1 at Module 001c10/Slot 0/Slice B: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 2 at Module 001c10/Slot 0/Slice C: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 3 at Module 001c10/Slot 0/Slice D: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 4 at Module 001c13/Slot 0/Slice A: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 5 at Module 001c13/Slot 0/Slice B: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 6 at Module 001c13/Slot 0/Slice C: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 7 at Module 001c13/Slot 0/Slice D: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 8 at Module 001c16/Slot 0/Slice A: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 9 at Module 001c16/Slot 0/Slice B: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 10 at Module 001c16/Slot 0/Slice C: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 11 at Module 001c16/Slot 0/Slice D: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 12 at Module 001c21/Slot 0/Slice A: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 13 at Module 001c21/Slot 0/Slice B: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 14 at Module 001c21/Slot 0/Slice C: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 15 at Module 001c21/Slot 0/Slice D: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 16 at Module 001c24/Slot 0/Slice A: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 17 at Module 001c24/Slot 0/Slice B: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 18 at Module 001c24/Slot 0/Slice C: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 19 at Module 001c24/Slot 0/Slice D: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 20 at Module 001c29/Slot 0/Slice A: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz
CPU 21 at Module 001c29/Slot 0/Slice B: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 22 at Module 001c29/Slot 0/Slice C: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 23 at Module 001c29/Slot 0/Slice D: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 24 at Module 001c32/Slot 0/Slice A: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 25 at Module 001c32/Slot 0/Slice B: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 26 at Module 001c32/Slot 0/Slice C: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 27 at Module 001c32/Slot 0/Slice D: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 28 at Module 001c35/Slot 0/Slice A: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 29 at Module 001c35/Slot 0/Slice B: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 30 at Module 001c35/Slot 0/Slice C: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

CPU 31 at Module 001c35/Slot 0/Slice D: 400 Mhz MIPS R12000 Processor Chip (enabled)
Processor revision: 3.5. Secondary cache: Size 8 MB Speed 266 Mhz

Main memory size: 4096 Mbytes
Instruction cache size: 32 Kbytes
Data cache size: 32 Kbytes
Secondary unified instruction/data cache size: 8 Mbytes

Memory at Module 001c10/Slot 0: 512 MB (enabled)
Bank 0 contains 256 MB (Standard) DIMMS (enabled)
Bank 1 contains 256 MB (Standard) DIMMS (enabled)

Memory at Module 001c13/Slot 0: 512 MB (enabled)
Bank 0 contains 256 MB (Standard) DIMMS (enabled)
Bank 1 contains 256 MB (Standard) DIMMS (enabled)

Memory at Module 001c16/Slot 0: 512 MB (enabled)
Bank 0 contains 256 MB (Standard) DIMMS (enabled)
Bank 1 contains 256 MB (Standard) DIMMS (enabled)

Memory at Module 001c21/Slot 0: 512 MB (enabled)
Bank 0 contains 256 MB (Standard) DIMMS (enabled)
Bank 1 contains 256 MB (Standard) DIMMS (enabled)

Memory at Module 001c24/Slot 0: 512 MB (enabled)
Bank 0 contains 256 MB (Standard) DIMMS (enabled)
Bank 1 contains 256 MB (Standard) DIMMS (enabled)

Memory at Module 001c29/Slot 0: 512 MB (enabled)
Bank 0 contains 256 MB (Standard) DIMMS (enabled)
Bank 1 contains 256 MB (Standard) DIMMS (enabled)

Memory at Module 001c32/Slot 0: 512 MB (enabled)
Bank 0 contains 256 MB (Standard) DIMMS (enabled)
Bank 1 contains 256 MB (Standard) DIMMS (enabled)

Memory at Module 001c35/Slot 0: 512 MB (enabled)
Bank 0 contains 256 MB (Standard) DIMMS (enabled)
Bank 1 contains 256 MB (Standard) DIMMS (enabled)
ROUTER in Module 001c10/Slot 0: Revision 1: Active Ports [1,2,3,4,5,6,7,8] (enabled)
ROUTER in Module 001c24/Slot 0: Revision 1: Active Ports [1,2,3,4,5,6,7,8] (enabled)
Integral SCSI controller 4: Version Fibre Channel QL2200A
Disk drive: unit 114 on SCSI controller 4 (unit 114)
Disk drive: unit 115 on SCSI controller 4 (unit 115)
Disk drive: unit 116 on SCSI controller 4 (unit 116)
Disk drive: unit 117 on SCSI controller 4 (unit 117)
Disk drive: unit 118 on SCSI controller 4 (unit 118)
Disk drive: unit 119 on SCSI controller 4 (unit 119)
Disk drive: unit 120 on SCSI controller 4 (unit 120)
Disk drive: unit 121 on SCSI controller 4 (unit 121)
Disk drive: unit 122 on SCSI controller 4 (unit 122)
Disk drive: unit 123 on SCSI controller 4 (unit 123)
Disk drive: unit 124 on SCSI controller 4 (unit 124)
Disk drive: unit 125 on SCSI controller 4 (unit 125)
Integral SCSI controller 3: Version Fibre Channel QL2200A
Disk drive: unit 114 on SCSI controller 3 (unit 114)
Disk drive: unit 115 on SCSI controller 3 (unit 115)
Disk drive: unit 116 on SCSI controller 3 (unit 116)
Disk drive: unit 117 on SCSI controller 3 (unit 117)
Disk drive: unit 118 on SCSI controller 3 (unit 118)
Disk drive: unit 121 on SCSI controller 3 (unit 121)
Integral SCSI controller 0: Version Fibre Channel QL2200
Disk drive: unit 40 on SCSI controller 0 (unit 40)
Disk drive: unit 41 on SCSI controller 0 (unit 41)
Disk drive: unit 42 on SCSI controller 0 (unit 42)
Disk drive: unit 43 on SCSI controller 0 (unit 43)
Disk drive: unit 44 on SCSI controller 0 (unit 44)
Disk drive: unit 45 on SCSI controller 0 (unit 45)
Disk drive: unit 46 on SCSI controller 0 (unit 46)
Disk drive: unit 47 on SCSI controller 0 (unit 47)
Disk drive: unit 48 on SCSI controller 0 (unit 48)
Disk drive: unit 49 on SCSI controller 0 (unit 49)
Integral SCSI controller 1: Version Fibre Channel QL2200A
Disk drive: unit 114 on SCSI controller 1 (unit 114)
Disk drive: unit 115 on SCSI controller 1 (unit 115)
Disk drive: unit 116 on SCSI controller 1 (unit 116)
Disk drive: unit 117 on SCSI controller 1 (unit 117)
Disk drive: unit 118 on SCSI controller 1 (unit 118)
Disk drive: unit 119 on SCSI controller 1 (unit 119)
Disk drive: unit 120 on SCSI controller 1 (unit 120)
Disk drive: unit 121 on SCSI controller 1 (unit 121)

```
Disk drive: unit 122 on SCSI controller 1 (unit 122)
Disk drive: unit 123 on SCSI controller 1 (unit 123)
Disk drive: unit 124 on SCSI controller 1 (unit 124)
Disk drive: unit 125 on SCSI controller 1 (unit 125)
Integral SCSI controller 2: Version Fibre Channel QL2200A
  Disk drive: unit 1 on SCSI controller 2 (unit 1)
  Disk drive: unit 2 on SCSI controller 2 (unit 2)
Integral Fast Ethernet: ef0, version 1, module 001c10, pci 4
  PCI Adapter ID (vendor 4215, device 8704) pci slot 1
  PCI Adapter ID (vendor 4215, device 8704) pci slot 2
  PCI Adapter ID (vendor 4215, device 8704) pci slot 1
  PCI Adapter ID (vendor 4215, device 8704) pci slot 2
  PCI Adapter ID (vendor 4215, device 8704) pci slot 3
  PCI Adapter ID (vendor 4265, device 3) pci slot 4
  PCI Adapter ID (vendor 4545, device 22529) pci slot 5
  PCI Adapter ID (vendor 4172, device 32777) pci slot 6
  PCI Adapter ID (vendor 4140, device 192) pci slot 7
IO3 external interrupts: 1
HUB in Module 001c10/Slot 0: Revision 2 Speed 200.00 Mhz (enabled)
HUB in Module 001c13/Slot 0: Revision 2 Speed 200.00 Mhz (enabled)
HUB in Module 001c16/Slot 0: Revision 2 Speed 200.00 Mhz (enabled)
HUB in Module 001c21/Slot 0: Revision 2 Speed 200.00 Mhz (enabled)
HUB in Module 001c24/Slot 0: Revision 2 Speed 200.00 Mhz (enabled)
HUB in Module 001c29/Slot 0: Revision 2 Speed 200.00 Mhz (enabled)
HUB in Module 001c32/Slot 0: Revision 2 Speed 200.00 Mhz (enabled)
HUB in Module 001c35/Slot 0: Revision 2 Speed 200.00 Mhz (enabled)
IP35prom in Module 001c10/Slot n0: Revision 6.44
IP35prom in Module 001c13/Slot n0: Revision 6.44
IP35prom in Module 001c16/Slot n0: Revision 6.44
IP35prom in Module 001c21/Slot n0: Revision 6.44
IP35prom in Module 001c24/Slot n0: Revision 6.44
IP35prom in Module 001c29/Slot n0: Revision 6.44
IP35prom in Module 001c32/Slot n0: Revision 6.44
IP35prom in Module 001c35/Slot n0: Revision 6.44
USB controller: type OHCI
```

システムにインストールされている周辺デバイスが `hinv` の出力リストに表示されない場合は、ハードウェアにトラブルが発生している可能性があります。なお、`hinv` をコマンド・モニタから実行した場合は、VME バスのボードでシステムに接続している一部の周辺デバイスが認識されないことがあります。この場合は、`hinv` をシステムのシェル・プロンプトから起動します。それでも周辺デバイスが認識されない場合は、ポートやデバイスをソケットに取り付け直し、正しい SCSI アドレスが使用されていることを確認します。それでも問題が解決しない場合は、ハー

ドウェア自体が不良である可能性があります。なお、インストール後に MAKEDEV コマンドを実行しないと、デバイスは hinv によって認識されないので注意してください。

/hw ディレクトリ内のインストール済みハードウェアの確認

/hw ディレクトリ構造の内容は、起動時に /etc/brc から ioconfig コマンドを実行することによって、必要に応じて更新できます。詳細については、54 ページの「システムの停止からマルチユーザ状態への移行」と、oconfig(1M) マン・ページを参照してください。認識されるすべてのハードウェアは、/hw の下のエントリとして表示されます。また、これらのエントリの一部は、/hw 構造と /dev 構造の各部とシンボリック・リンクになっています。

たとえば、システムの root パーティションは、次のようなエントリで表示されます。

```
/hw/scsi_ctrlr/1/target/1/lun/0/disk/partition/0/block
/hw/disk/root
/dev/root
```

先頭のエントリ (/hw/scsi_ctrlr/ など) は、hwgraph(4) で説明されているようにボードの物理位置を表します。2 番目と 3 番目のパス (/hw/disk/root と /dev/root) は、長い物理パス名 (上の例の先頭のパス) へのシンボリック・リンクです。シンボリック・リンクは、デバイスを参照する必要がある各種システム・コマンドにとって便利な方法です。

IP ネットワーク・インタフェースのハードウェア・デバイスへの割当て方

reboot、shutdown、halt、init コマンドなどを実行した後、または電源を切った後でシステムを再起動すると、起動ルーチンは NUMALink 内部接続構造に接続するすべてのシステムのハードウェア・デバイスを検索します。構造内のすべてのモジュールのロットとリンクが検索されます。

このルーチンは検索した後、ハードウェア・グラフと呼ばれる階層構造のファイルシステムを生成します。ハードウェア・グラフは、検索されたすべてのハードウェアをリストします。ハードウェア・グラフの最上位は /hw として表示されます。詳細については、hwgraph(4) マン・ページを参照してください。

ハードウェア・グラフが完了した後、`ioconfig` プログラムはユニット番号を必要とする各検索済みデバイスに番号を割当てます。`hinv` およびデバイス・ドライバなど、ほかのプログラムは、割当てられた番号を読み込み、それを使用します。

初期起動時に、`ioconfig` は順番に番号を割当てます。たとえば、2つの IRIS ATM XIO ボードが検索された場合、`unit0` (ポート 0 から 3 まで) と `unit1` (ポート 4 から 7 まで) の番号が割当てられます。

以降の起動で `ioconfig` を実行すると、検索済みのハードウェアと新しいハードウェアが区別されます。検索済みのハードウェアに対しては、初期起動時に割当てたユニット番号とポート番号を割当てます。新しいハードウェアに対しては、順番に新しい番号を割当てます。`ioconfig` は、たとえデバイスが削除されても一度割当てた番号を再び別のデバイスに割当ててことはありません。

新しいハードウェアは、ハードウェア・グラフ・リスト (`/hw` に含まれるパス) の情報に基づいて以前に検索されたハードウェアと区別されます。検索済みデバイスのデータベースは、`/etc/ioconfig.conf` ファイルに保存されます。たとえば、同じ名前の代替えボードを古いボードと同じ場所にインストールした場合には、古いボード番号が割当てられます。このボードを別の場所に移動した場合には、新しいユニット番号とポート番号が割当てられます。

ハードウェア・グラフと `ioconfig` についての詳細は、`hwgraph(4)` および `ioconfig(1M)` マン・ページを参照してください。

versions によるインストール済みソフトウェアの確認

`inst` でインストールしたソフトウェア・パッケージの情報は、`versions` コマンドを使用して参照できます。ただし、このコマンドは、システム・シェル・プロンプトでのみ実行でき、コマンド・モニタからは実行できません。`inst` 以外の方法でインストールしたソフトウェアは、`versions` の出力に含まれません。この出力では、ソフトウェア製品の名称と、リリースのバージョン番号が表示されます。デフォルトでは、`versions` を使用するとすべてのソフトウェア製品とそのサブシステムが表示され、通常は数百行の長さになります。したがって、この出力をファイルにリダイレクトし、後で必要に応じて参照する方が便利です。インストールされている製品の概要 (特定のサブシステムのリストは表示されません) を参照する場合は、69 ページの例 4-2 のように `-b (brief)` フラグを指定します。

例 4-2 versions コマンド出力

versions -b コマンドの出力例を示します。完全なリストはこの例より長くなります。

```
I = インストール、R = 削除
 名前          日付      説明
I 4Dwm          07/18/96 Desktop Window Manager, 6.2 (based on
OSF/Motif 1.2.4)
I demos        07/18/96 Graphics Demonstration Programs, 6.2
I desktop_base 07/18/96 IndigoMagic Desktop Base Software,
6.2
I desktop_eoe  07/18/96 IndigoMagic Desktop, 6.2
I desktop_tools 07/18/96 Desktop Tools, 6.2
I dps_eoe      07/18/96 Display PostScript/X, 2.0.5 based on
PostScript Level 2
I eoe          07/22/96 IRIX Execution Environment, 6.2
I insight      07/18/96 IRIS InSight Viewer, 2.3.3
I nfs          07/18/96 Network File System, 6.2 with IMPACT
10000
```

gfxinfo によるインストール済グラフィックス・ハードウェアの確認

gfxinfo コマンドは、システムにインストールされているグラフィックス・ハードウェアの種類を調べるのに役立ちます。このコマンドは標準のサーチ・パスに含まれていない /usr/gfx ディレクトリにあるため、通常はフルパス名を指定しなければ実行できません。例 4-3 から明らかのように、このコマンドに引数の実行は必要ありません。

例 4-3 gfxinfo コマンド出力

次に、Indy ワークステーションでの出力例を示します。

```
% /usr/gfx/gfxinfo
Graphics board 0 is "NG1" graphics.
  Managed (":0.0") 1280x1024
  24 bitplanes, NG1 revision 3, REX3 revision B,
  VC2 revision A
  MC revision C, xmap9 revision A, cmap revision C,
  bt445 revision A
  Display 1280x1024 @ 60Hz, monitor id 12
```

hinv コマンドではなく、このコマンドを使用すると、グラフィックス・システムに関する詳細な情報が得られます。hinv コマンドは Indy 24-bit という情報だけを返します。gfxinfo の出力結果から、画面の数とピクセルの解像度、ビットプレーンの構成、構成要素のリビジョン・レベル、モニタの種類などを確認できます。現在、gfxinfo に関するマン・ページはありません。グラフィックス機能がないサーバにはこのコマンドはインストールされていません。

uname による基本システムの確認

uname コマンドは、オペレーティング・システムのバージョンやホスト名などに関する情報を返します。-a オプションを指定すると、uname の出力リストの詳細が表示されます。uname コマンドのオプションとフィールドについての詳細は、uname(1) マン・ページを参照してください。

lpstat によるプリンタの状態の確認

lpstat -a コマンドを実行すると、lp スプーリング・システム用に設定されているすべてのプリンタとその状態が表示されます。プリンタ管理についての詳細は、『IRIX Admin: Peripheral Devices』と、lpadmin(1M) および lpstat(1) マン・ページを参照してください。

chkconfig によるソフトウェア構成オプションの確認

chkconfig(1M) コマンドを使用すると、例 4-4 のようにしてワークステーションやサーバの設定を簡単に確認できます。/sbin/chkconfig コマンドは、各種のプロセス・デーモンの状態（アクティブであるかどうか）を報告します。

例 4-4 chkconfig コマンド出力

構成を確認するには chkconfig コマンドを入力します。

chkconfig

上記の例は、ネットワーク・ファイル・システム (NFS: Network File System) オプションがインストールされたネットワーク・ワークステーションでの典型的な出力例です。出力リストの左側のカラムはシステム機能を示しており、右側のカラムはそれらの機能のオン / オフ状態を示しています。

Flag	State
====	=====
acct	off
audit	off
automount	on
gated	off
lockd	on
mouted	off
named	off
network	on
nfs	on
noiconlogin	off
nsr	on
quotacheck	off
quotas	off
routed	on
rtnetd	off
rwhod	off
sar	on
snmpd	on
timed	on
timeslave	off
verbose	off
visuallogin	on
windowssystem	off
yp	on
yppmaster	off
ypserv	off

次に、各システム機能について詳しく説明します。

acct	システム・アカウントのオン / オフ。
audit	システム監査トレールのオン / オフ。
automount	NFS の automount (1M) デーモンのオン / オフ。この設定オプションは、NFS がワークステーションにインストールされている場合にのみ使用できます。
gated	複数のルーティング・プロトコルを制御する gated (1M) デーモンのオン / オフ。
lockd	NFS の lock デーモンのオン / オフ。この設定オプションは、NFS がワークステーションにインストールされている場合にのみ使用できます。
mouted	Stanford IP マルチキャスト・ルーティング・デーモンのオン / オフ。

named	インターネット・ドメイン・ネーム・サーバである <code>named(1M)</code> のオン / オフ。
network	ネットワークのオン / オフ。
nfs	NFS のオン / オフ。この設定オプションは、NFS がワークステーションにインストールされている場合のみ使用できます。
noiconlogin	ビジュアル・ログイン・プログラムの <code>pandora(1)</code> は、システムに登録されたユーザをアイコンで表示します。このオプションは <code>pandora</code> のオン / オフではなく、アイコンを表示させるかどうかを切替えます。 <code>pandora</code> のオン / オフは <code>visuallogin</code> オプションを使用して切替えます。
nsr	この設定オプションは、IRIS Networker バックアップ・ユーティリティがワークステーションにインストールされている場合のみ使用できます。
quotacheck	ディスク領域割当てチェッカのオン / オフ。
quotas	ディスク割当てサブシステムのオン / オフ。
routed	ネットワーク・ルーティング・テーブルを管理する <code>routed(1M)</code> のオン / オフ。
rtnetd	到着するネットワーク・パケットよりも優先度の高いリアルタイム・プロセスを先に処理できるようにする <code>rtnetd(1M)</code> のオン / オフ。
rwhod	<code>rwhod(1M)</code> のオン / オフ。
sar	システム・アクティビティの状態を報告する <code>sar(1)</code> のオン / オフ。
snmpd	SNMP (Simple Network Management Protocol) デーモンのオン / オフ。
timed	4.3BSD のタイム・サーバ・デーモン <code>timed(1M)</code> のオン / オフ。
timeslave	SGI のタイム・サーバ・デーモンのオン / オフ。 <code>timed</code> と同じように、このオプションはワークステーションのクロックを別のワークステーションのクロックに補正します。通常はワークステーションのグループやサイト全体で使用するマスター・タイム・サーバです。
verbose	システムの起動時や停止時に、デーモンの機能に関する情報を印刷するかどうかを制御するオプション。このオプションをオフにすると、システムの起動時や停止時に印刷される情報が少なくなります。
visuallogin	ビジュアル・ログイン・プログラムである <code>pandora(1)</code> のオン / オフ。

windowsystem	ウィンドウ・マネージャのオン / オフ。
yp	ネットワーク・インフォメーション・サービス (NIS: Network Information Service) のオン / オフ。この機能は yp (yellow page) と呼ばれています。NIS は NFS ソフトウェアに含まれています。この設定オプションは、NFS がワークステーションにインストールされている場合にのみ使用できます。
ypmaster	NIS マスター・サービスのオン / オフ。この設定オプションは、NFS がワークステーションにインストールされている場合にのみ使用できます。
ypserv	NIS サーバとそのバインド・プロセスのオン / オフ。この設定オプションは、NFS がワークステーションにインストールされている場合にのみ使用できます。

ソフトウェアの構成

ここでは、IRIX オペレーティング・システムをカスタマイズするために用意されているさまざまなオプションの設定方法について説明します。

chkconfig によるソフトウェア構成オプションの設定

chkconfig(1M) コマンドを使用すると、システム構成の一部を変更できます。chkconfig を使用してシステム構成を変更できる部分を調べるには、chkconfig コマンドを入力します。

chkconfig

このコマンドを入力すると設定オプションのリストが表示されます。設定オプションについては、70 ページの「chkconfig によるソフトウェア構成オプションの確認」で説明されています。-s オプションを指定すると、設定オプションの状態（オンまたはオフ）に従ってリストをソートすることもできます。

設定オプションを変更するには、chkconfig コマンドに 2 つの引数を指定します。1 つは変更する設定オプションの名前、もう 1 つはそのオプションの新しい設定状態（オンまたはオフ）を指定します。システム構成を変更するには、ルート特権を持っていないければなりません。

たとえば、プロセス・アカウントングをオンに設定する場合は、root またはシステム管理者としてログインし、次のように入力します。

```
chkconfig acct on
```

プロセス・アカウントングをオフにする場合は、次のように入力します。

```
chkconfig acct off
```

/etc/init.d ディレクトリにあるスタートアップ・スクリプトはシステムが起動されてマルチユーザ・モードになる場合にのみ実行されるため、一部のシステム設定はシステムを停止して再起動しないと有効になりません。スタートアップ・スクリプトは、chkconfig で設定されたファイルから情報を読み取り、どのデーモンを起動するかを決定します。

chkconfig で制御できる設定オプションの中には、chkconfig のリストに表示されないものがあります。次に、そのような設定オプションを示します。

nostickytmp /tmp ディレクトリに制限を与えます。このディレクトリに制限がある場合 (**nostickytmp** にオフを設定した場合) は、そのファイルを削除できる明示的なパーミッション (書込み権) があるか、特権ユーザの権限があるファイルを所有していなければ、このディレクトリ内のファイルを削除できません。

nostickytmp にオンを指定した場合は、ファイルの所有者でないユーザでも共有の書込み可能なディレクトリである /tmp 内のファイルを自由に削除したり置換えることができます。大きいテンポラリ・ファイルを作成するためのディスク領域が不足している場合は、このような設定が便利ですが、重要なファイルが削除されないようにディスク領域を増やすことをお勧めします。

nocleantmp システムを起動するたびに、/tmp ディレクトリ内をすべて消去するかどうかを制御します。**nocleantmp** をオンに設定すると、/tmp は消去されません。これをオフに設定すると、システムの起動時に /tmp 内のすべてのファイルが削除されます。

上記の設定オプションを chkconfig メニューのリストに表示する場合は、**-f** オプションを使用して、オプションの設定ファイルを作成します。

```
chkconfig -f nocleantmp on
```

上記の例は、/etc/config ディレクトリに nocleantmp という設定ファイルが作成され、再起動時に /tmp ディレクトリからファイルが削除されないことを示しています。

システム・デフォルトの設定と変更

次に、プログラムやシステムの機能に影響を与えるシステムの設定項目を示します。

- システム・ディスプレイ
- プロセッサの割当て
- 時間帯
- システム名
- ネットワーク・アドレス
- デフォルト・システム・プリンタ
- システムの日付と時間
- ファイルとディレクトリのアクセス権
- アクセス・コントロール・リスト (ACL: Access Control List) と機能

これらのデフォルトについては別の章で詳しく説明しますが、IRIX システムの概要を理解するためにこの章でも簡単にまとめます。

システム・ディスプレイの設定

DISPLAY 環境変数を変更することによって、システム上で動作するプログラムやユーティリティの出力を同じネットワーク内の別のシステムの画面に表示させることができます。この機能は、ネットワーク上にグラフィカル・システムと非グラフィカル・サーバがある場合に便利です。つまり、サーバが提供する情報をグラフィカルに出力させる場合は、表示先をグラフィックス・ワークステーションに切替えることができます。

たとえば、サーバのコンソールが文字ベースの端末のみの場合、`gr_osview` を実行して CPU の使用量をグラフィカルに検証するには、サーバ上で次のコマンドを実行します (csh シェルと tcsh シェルの場合)。

```
setenv DISPLAY graphics_system:0  
gr_osview
```

ksh シェルと sh シェルの場合は、次のコマンドを実行します。

```
DISPLAY=graphics_system:0; export DISPLAY
```

また、グラフィックス・システム上で `xhost` コマンドを実行することにより、サーバでそれを表示できるようにします。

```
xhost +server_system
```

サーバ上で `gr_osview` を起動すると、指定したグラフィックス・システムのウィンドウに出力が表示されます。この例では、システム名は `graphics_system` です。システム名の後の `:0` は、出力をディスプレイ・モニタ 0 (グラフィック・コンソール) に表示させるように指示しています。グラフィックス・コンソールの使用後は、サーバ上で次のコマンドを実行し、ディスプレイをリセットしなければなりません。

```
setenv DISPLAY local_server:0
```

`local_server` には、現在使用しているサーバ名を指定します。グラフィックス・システムからサーバにログインし、`DISPLAY` 変数を上記のように設定した場合、操作を終了した後そのままログアウトします。

マルチプロセッサ・システムでのプロセッサ割当ての変更

マルチプロセッサ・システムでは、`mpadmin(1)` コマンドと `Miser` コマンド・セットを使用して、プログラムをシステムの各種プロセッサに割当てる方法を設定できます。システムがマルチプロセッサかどうかを確認するには、`hinv(1M)` コマンドを使用します。マルチプロセッサ・システムの場合は、`hinv` の出力リストに次のような情報が表示されます。

```
Processor 0: 40 MHZ IP7
Processor 1: 36 MHZ IP7
Processor 2: 40 MHZ IP7
Processor 3: 40 MHZ IP7
Processor 4: 40 MHZ IP7
Processor 5: 40 MHZ IP7
Processor 6: 40 MHZ IP7
Processor 7: 40 MHZ IP7
```

また、次のような情報が表示される場合もあります。

```
8 40 MHZ IP7 Processors
```

シングルプロセッサ・システムの場合は、`hinv` の出力リストに次のような情報が表示されます。

1 100 MHZ IP22 Processor

システムにプロセッサが 1 つしかない場合でも (ほとんどのシステムでは単一のプロセッサのみ)、このコマンドは動作しますが、特に何も行われません。

mpadmin コマンドを使用すると、プロセッサをオフに切替えたり、プロセッサの各種状態を表示したり、システム・クロックなどのシステム機能を特種なプロセッサに移動することができます。詳細については、mpadmin(1M) マン・ページを参照してください。

バッチ環境でシステムをパーティション分割するには、Miser のバッチ・スケジューリング・コンポーネントを使用します。Miser は、バッチ・アプリケーション間やバッチと対話式アプリケーション間のメモリおよび CPU の割当てを細かく制御することができます。Miser についての詳細は、『IRIX Admin: Resource Administration』を参照してください。

Cpuset System は本来は作業負荷マネージャ・ツールで、システム管理者が、特定のプロセスまたは複数のプロセスのセットが使用できるプロセッサの数を制限できるようにするものです。オプションで、カーネル・メモリとユーザ・メモリの両方を制限できます。

メモリ制限機能を有効にすると、提供された CPU のリストから、それぞれが CPU のセットを含むノードのセットが計算され、該当するノードに割当てられた CPU へのメモリ割当てを制限できます。割当てを使用可能な物理メモリのみに制限したり、オーバフロー分をスワップ・ファイルにスワップするように指定できます。

システム管理者は、cpuset を使用して、大規模なシステム内に CPU の区画を作成できます。このような分割されたシステムを使用すると、プロセスのセットを特定の CPU 群に含まれるよう制御でき、それらのプロセスがシステム上の他の作業に対して及ぼす干渉や競合の量を低減できます。制限付きの cpuset の場合、cpuset にアタッチされたプロセスは、システム上の他の作業の影響を受けません。cpuset にアタッチされたプロセスのみを、その cpuset に割当てられた CPU で実行されるようスケジュールできます。オープンな cpuset は、プロセスを CPU の特定のセットに制限して、これらのプロセスがシステムの他の部分に与える影響を最小限に抑える場合に使用できます。

cpuset コマンドを使用して、これらの CPU セットを作成、破棄、および表示できます。詳細については、cpuset(1) マン・ページおよび『IRIX Admin: Resource Administration』を参照してください。

実行中の作業からの CPU の除外

`mpadmin(1)` コマンドを使用して、指定した CPU を実行中の作業から除外できます。この機能は、特定の CPU にハードウェア・エラーが発生している可能性があるが、システム全体をシャットダウンしたくないような状況で便利です。

`mpadmin -x[processor]` コマンドは、番号 `processor` で指定されるプロセッサを除外します。`mpadmin` コマンドに `-I` 引数を指定した場合と同様に、除外したプロセッサは分離されません。また、除外したプロセッサ上では、`runon(1)` コマンドおよび `MP_MUSTRUN` 引数を指定した `sysmp(2)` 機能を使用してコマンドを実行できなくなります。`processor` を指定しなかった場合、`mpadmin -x` コマンドは、除外されているプロセッサの数を標準出力に書込みます。プロセッサ上で実行されている `MUSTRUN` プロセスが存在する場合、この要求は失敗します。詳細については、`mpadmin(1)` マン・ページを参照してください。

システム名の変更

システム名は複数の場所に記録されています。システム名を変更する場合は、次のものに記録されているシステム名をすべて変更してください。変更漏れがあると、システムは正常に動作しません。

- `/etc/sys_id` ファイル
- `/etc/hosts` ファイル (ネットワーク用)
- `hostname` で参照または設定するカーネル・データ構造
- NIS マスター・サーバ上の NIS マップ (NIS を実行している場合)

メモ：起動時に実行するプログラムの多くは、ワークステーションの名前を参照しているので、稼働中のワークステーションの名前は勝手に変更しないでください。

システム名を表示するには、引数を指定せずに `hostname` コマンドを入力します。

hostname

このコマンドを入力すると、システム名が表示されます。また、`uname` コマンドを入力すると、システム名とともにほかの情報も表示されます。

手順 4-1 で、ワークステーションの名前を変更する方法を説明します。

手順 4-1 ワークステーションの名前の変更

1. `root` でログインします。
2. `/etc/sys_id` ファイルを編集します。ホスト名を新しい名前に変更し、ファイルを保存してエディタを終了します。
3. `/etc/hosts` などのネットワーク・ファイル内や、あるいは NIS マスター・サーバ上の NIS マップ内のホスト名も変更します。
4. システムを再起動します。

起動時にホスト名を参照するすべてのプログラムは、正しいホスト名を使用します。

ワークステーションのインターネット・アドレスについての詳細は、『IRIX Admin: Networking and Mail』を参照してください。また、システム名についての詳細は、`hostname(1)` および `uname(1)` マン・ページを参照してください。

ネットワーク・アドレスの設定

システムのネットワーク・アドレス (IP アドレス) についての詳細は、『IRIX Admin: Networking and Mail』を参照してください。

手順 4-2 では、ネットワーク・アドレスを設定する方法を説明します。

手順 4-2 ネットワーク・アドレスの設定

1. `/etc/hosts` ファイル内のシステム名と同じ行にネットワーク・アドレスを設定します。
たとえば `194.45.54.4 magnolia`
2. ネットワーク・インフォメーション・サービス (NIS: Network Information Service) を使用する場合は、NIS がシステムにインストールされていることを確認し、`/var/yp/ypdomain` ファイル内にドメイン名を設定します。
3. `nvramp(1M)` コマンドを使用し、`netaddr` 変数をシステムの IP アドレス番号に設定します。
`nvramp netaddr 194.45.54.4`

デフォルト・プリンタの設定

デフォルト・プリンタは、`lpadmin(1M)` コマンドを使用して設定します。次のコマンドは、デフォルト・プリンタを *laser* に設定します。

```
lpadmin -dlaser
```

この例では、正しく設定されている *laser* というプリンタがすでにあることを前提としています。プリンタの設定についての詳細は、『IRIX Admin: Peripheral Devices』を参照してください。

時間帯の設定

システムで使用する時間帯を設定するには、`/etc/TIMEZONE` ファイルを編集します。たとえば、米国の東海岸のサイトについては、このファイルは次のような内容になります。

```
# Time Zone
TZ=EST5EDT
```

`TZ=EST5EDT` の意味は次のとおりです。

- 現在の時間帯は米国東部標準時間 (EST)
- グリニッジ標準時より西方向に 5 時間の差がある (5)
- 夏時間が適用されている (EDT)

`TZ` の環境変数はシステムの起動時に `init(1)` によって読取られ、`TZ` の値はその後のすべてのプロセスに渡されます。東部標準時など、時間帯の指定は、ユーザに便利のように引数としてそのまま使用されます。指定の主な部分は、グリニッジ標準時からの偏差と、夏時間を適用しているかどうかです。表 4-1 から表 4-6 に、主要な地域（北米、欧州、アジア、中東、南米、豪州およびニュージーランド）の時間帯の情報を示します。

表 4-1 北米の時間帯

地域	標準時との差	略称
ニューファンドランド	-3:30	NST
大西洋	-4:00	AST
東部	-5:00	EST

表 4-1 北米の時間帯

地域	標準時との差	略称
中部	-6:00	CST
サスカチエワン	-6:00	CST
山地	-7:00	MST
太平洋	-8:00	PST
ユーコン	-9:00	YST
アラスカ	-10:00	AST
ハワイ	-10:00	HST
ベーリング	-11:00	BST
北バハ	-8:00	PST
南バハ	-7:00	MST
メキシコ (標準)	-6:00	CST

表 4-2 欧州の時間帯

地域	標準時との差	略称
アイルランド	0:00	BST
英国	0:00	BST
西欧	0:00	WET
アイスランド	0:00	WET
中欧	1:00	MET
ポーランド	1:00	MET
東欧	2:00	EET
トルコ	3:00	EET
西ロシア	3:00	WSU

表 4-3 アジアの時間帯

地域	標準時との差	略称
台湾	8:00	CST
香港	8:00	HKT
日本	9:00	JST
韓国	9:00	ROK
シンガポール	8:00	SST

表 4-4 中東の時間帯

地域	標準時との差	略称
イスラエル	2:00	IST
エジプト	2:00	EET

表 4-5 南米の時間帯

地域	標準時との差	略称
ブラジル / 東部	-3:00	EST
ブラジル / 西部	-4:00	WST
ブラジル / アクレ	-5:00	AST
ブラジル / デノロンハ	-2:00	FST
チリ / 大陸	-4:00	CST
チリ / イースター島	-6:00	EST

表 4-6 豪州とニュージーランドの時間帯

地域	標準時との差	略称
豪州 / タスマニア	10:00	EST
豪州 / ケイーンズランド	10:00	EST
豪州 / 北部	9:30	CST
豪州 / 西部	8:00	WST
豪州 / 南部	9:30	CST
豪州 / ビクトリア	10:00	EST
豪州 / NSW	10:00	EST
ニュージーランド	12:00	NZT

時間帯の設定についての詳細は、`timezone(4)` マン・ページを参照してください。

システムの日付と時刻の変更

日付と時刻を設定するには、`date(1)` コマンドを使用します。たとえば、日付を 1999 年 4 月 1 日、時刻を 09:00 に設定する場合は、`root` でログインして次のように入力します。

```
date 0401090099
```

実行中のシステムで日付や時刻を変更すると、予期しない結果になることがあります。ユーザや管理者は、システム・スケジューリング・ユーティリティ (`at`、`cron`、`batch`) を使用して特定の時間にコマンドを実行します。現在有効な日付や時刻を変更すると、これらのユーティリティが予定どおりの時間に実行できなくなります。同じように、システム・ユーティリティの `make(1)` を使用している場合は、`Makefile` で指定されているコマンドによって不適切な処理が行われる可能性があります。したがって、システムの日付と時刻は十分に注意して正しい設定を保持しなければなりません。これらの設定をむやみに変更すると、ユーザの作業にかなり不都合が生じたり、それらの作業を破壊する危険性があります。

`timed` をスレーブ・システムで実行している場合は、時刻の設定は `date` コマンドではなく `timed` によって行われます。詳細については、`timed(1M)` マン・ページを参照してください。

ファイルおよびディレクトリのアクセス権の制御

通常、IRIX ファイルのアクセス権は、システムのセキュリティを守りながら、複数のユーザ間でファイルを簡単に使用できることを目的に設定されます。ここでは、ファイルのアクセス権を変更し、ユーザ、グループ、一般ユーザに対して読み権、書き込み権、実行権を許可したり拒否する方法について説明します。なお、ユーザは `umask` を設定することによって、自分のファイルへのデフォルトのアクセス権を制御できることにも注意してください。詳細については、133 ページの「デフォルトのファイル・パーミッション (`umask`)」を参照してください。

IRIX オペレーティングでは、パーミッションは3つのカテゴリ、ユーザは3種類の関係に分かれます。つまり、ファイルの所有者、所有者のユーザ・グループ、および一般ユーザ関係です。ディレクトリをロング・リストで表示すると、ディレクトリ内の各ファイルのパーミッション・フィールドは次のように表示されます。

```
-rwxrwxrwx
```

パーミッション・ラインに `rwx` の文字列が3組表示されていることに注意してください。最初の `rwx` は、ファイルの所有者の権限を示し、次の組はグループ・メンバーの権限を示し、最後の組はシステム的一般ユーザの権限を示します。上記の例ではフル・パーミッションを与えています。次にパーミッションを制限した例を示します。

```
-rw-r--r--
```

パーミッションの3つのカテゴリは、読み、書き込み、実行を示します。ファイルのロング・リストでは、`r` は読みを、`w` は書き込みを、`x` は実行を表します。ファイルのロング・リストを表示するには、システム・プロンプトで次のように入力します。

```
ls -l
```

パーミッション情報とともに、`ls -l` コマンドはファイルの所有者やサイズ、最後に編集された日付を表示します。

読み権は、ファイルの内容を表示することができます。書き込み権は、ファイルを変更したり削除することができます。また、実行権は、シェル・プロンプトからコマンドとして実行することができます。

パーミッション・リストの各文字には意味があります。パーミッションは左側が先頭になり、最初の文字はダッシュです。先頭文字以外のダッシュは、パーミッションが何も与えられておらず、

そのパーミッションで指定されているアクションに対して何も操作することができないことを意味します。パーミッション・リストの先頭部分は、そこに表示される内容により、それがファイルなのか、ディレクトリなのかを示します。ディレクトリの場合には、先頭部分に `d` が表示されます。`d` 以外の文字として、パイプ、ブロックまたはキャラクタなどの特殊デバイス・ファイルを表す文字、またはほかのファイルの種類を示す文字が表示されます。詳細については、`ls(1)` マン・ページを参照してください。

ファイルのパーミッションの現在の設定状態を確認するには、`ls -l` コマンドを使用します。たとえば、`review` ファイルのパーミッションの状態を確認するには、次のコマンドを入力します。

ls -l review

```
-rw-r--r-- 1 jones engr 1015  8月 14日 16時 20分 review
```

パーミッションは、ユーザ、グループ、一般ユーザごとに読み込み (`r`)、書き込み (`w`)、実行 (`x`) で表示されます。つまり、ユーザ、グループ、一般ユーザに対して、ファイルへの読み込み権、書き込み権、実行権の組み合わせがそれぞれ指定されます。先頭の文字（上の例の場合は `dashu`）に続く次の 3 文字がユーザの読み込み権、書き込み権、実行権です。その次の 3 文字がグループの読み込み権、書き込み権、実行権です。最後の 3 文字は、一般ユーザの読み込み権、書き込み権、実行権です。したがって上記の例の場合、ユーザ `jones` は `review` ファイルへの読み込み権 (`r`) と書き込み権 (`w`) を持ち、グループは読み込み権 (`r`) だけを持ち、一般ユーザも読み込み権 (`r`) だけを持ちます。誰も実行権 (`x`) は持ちません。

特権ユーザまたはファイルの所有者は、このパーミッションの設定を変更できます。特権ユーザは、`chmod` コマンドで、一般ユーザにファイルへの書き込み権を設定できます。たとえば、`review` ファイルへの書き込み権をグループと一般ユーザに与えるには、以下に示すように `go+w` オプションを使用します。ここで、`g` はグループ、`o` は一般ユーザ、`+w` は書き込み権を表します。

chmod go+w review

アクセス権は次のようになります。

```
-rw-rw-rw- 1 jones engr 1015  8月 14日 16時 20分 review
```

このほかに、8 進数でパーミッションを制御する方法があります。この方法では、読み込み権、書き込み権、実行権 (`4+2+1`) を `7` で表します。ファイルに対する読み込み権、書き込み権、実行権をすべてのユーザに与えるには、`chmod 777` コマンドを使用します。また、所有者に読み込み権だけを与え、ほかのユーザにはいっさいのアクセス権を与えない場合は、`chmod 400` コマンドを使用します。ファイルとディレクトリに対するアクセス権の設定についての詳細は、`chmod(1)` マン・ページを参照してください。

メモ: デバイス・ファイルに対して `chmod` を使用する場合、`/etc/ioperms` ファイルを編集して変更を反映させてください。これを行わないと、再起動後にデバイス・ファイルはデフォルトのアクセス権を返します。

`/etc/ioperms` 内のエントリの形式は、次のとおりです。

device_name owner group nnn

device_name にはデバイス・ファイル名、*owner* にはファイル所有者、*group* にはグループ、*nnn* には上記および `chmod(1)` マン・ページで説明されている 8 進数のパーミッション設定を指定します。`/etc/ioperms` ファイル内のデバイス・パーミッション設定についての詳細は、`ioconfig(1M)` マン・ページを参照してください。

ディレクトリ・パーミッション

各パーミッションの意味が多少異なる点を除いて、ディレクトリとファイルのパーミッションは同じです。たとえば、ディレクトリの読み権は、そのディレクトリの内容を `ls` コマンドで表示できるという意味です。書き込み権は、そのディレクトリのファイルに対して追加、変更、削除などの操作を実行できることを意味します。ただし、そのディレクトリに対する書き込み権を持っていないければ実行することができず、またディレクトリを所有していても個々のファイルに対する書き込み権がなければファイルを変更したり削除することはできません。ディレクトリの実行権は、`cd` コマンドを使用して、そのディレクトリに移動できることを意味します。

ファイル・パーミッション

パーミッション・フィールドの先頭の文字（この場合は、ダッシュ）に続く 3 つの文字は、ファイルの所有者のパーミッションを表します。次にファイルのロング・リストの例を示します。

```
-rwx----- 1 owner grp 6680 4月 24日 16時26分 shell.script
```

このファイルはディレクトリではないため、先頭文字はダッシュになっています。`rwx` の文字は、ファイルの所有者 `owner` が、このファイルに対して読み権、書き込み権、実行権を持っていることを示します。次の 3 つの領域は、所有者のグループに対する権限を示します。次の例では、グループは `grp` です。このファイルのパーミッションは次のように設定されているとします。

```
-rwxr-x--- 1 owner grp 6680 4月 24日 16時26分 shell.script
```

この場合、グループ *grp* のすべてのメンバーが、このファイルを読込んだり実行することができますが、変更したり削除することはできません。グループ *grp* のすべてのメンバーは、個々に所有するファイルのプールを共有することができます。グループの読み込みおよび書き込みの権限に注意して使用することにより、1人が所有する1セットのドキュメント・ソースを任意のグループ・メンバーが作業できるように設定することができます。

最後の領域は、システム上のその他すべてのユーザの権限を指定し、パブリック・パーミッションといいます。システム上の任意のユーザが読み込み可能に設定されているファイルは、パブリック・ファイルといいます。

例 4-5 に、Projects ディレクトリのロング・リストの例を示します。

例 4-5 Projects ディレクトリ

総ブロック数 410

```
drw----- 1 owner grp 48879  3月 29日 18時10分 critical
-rw-r--r-- 1 owner grp  1063  3月 29日 18時10分 meeting.notes
-rw-rw-rw- 1 owner grp  2780  3月 29日 18時10分 new.deal
-rwxrwxrwx 1 owner grp  8169  6月  7日 13時41分 new.items
-rw-rw-rw- 1 owner grp  4989  3月 29日 18時10分 response
-rw----- 1 owner grp 23885  3月 29日 18時10分 project1
-rw-r----- 1 owner grp  3378  6月  7日 13時42分 saved_mail
-rw-r--r-- 1 owner grp  2570  3月 29日 18時10分 schedules
-rwxrwxr-x 1 owner grp  6680  4月 24日 16時26分 shell.script
```

このディレクトリのファイルのパーミッションの設定はさまざまです。所有者のみに権限が制限されているファイルもあれば、所有者のグループに属するメンバーのみに読み込みの権限が設定されているファイルもあります。また、すべての人に対して読み込み、書き込み、実行の権限を設定しているファイルもあります。シェル・スクリプトは、すべてのユーザが実行可能です。

パーミッションの変更

ファイルのパーミッションは、`chmod(1)` コマンドを使用して行います。`chmod` コマンドは、自分が所有するファイルに対してのみ使用できます。通常、このコマンドは、個人用または秘密にしておきたいファイル、または個人ディレクトリを保護したい場合、あるいは所有者以外の人々がファイルを使用できるようにファイルに対して権限を与える場合に使用します。ファイルまたはディレクトリに対するアクセスを所有者にのみ制限する場合には、次のように入力します。

```
chmod 600 filename
```

```
chmod 700 dirname
```

これに対して、さらにパーミッションを追加する場合には、`chmod` コマンドにパーミッションを示す文字を指定します。たとえば、一般ユーザにファイルの書き込み権限を与える場合には、次のように入力します。

```
chmod +w filename
```

ほかの例については、`chmod(1)` マン・ページを参照してください。

umask によるパーミッションの設定

ファイルのデフォルト・パーミッションを決定するには、`umask` コマンドを `.cshrc`、`.profile` または `.login` ファイルに登録します。システム全体のデフォルトの `umask` 設定は `/etc/profile` または `/etc/cshrc` ファイルにあります。`umask` の設定を変更することにより、ファイルまたはディレクトリのデフォルト・パーミッションを有効な DAC (Discretionary Access Control) パーミッションに置換えることができます。詳細については、`umask(1)` マン・ページを参照してください。

`umask` コマンドの欠点は、作成するすべてのファイルが同じパーミッションになることです。通常、作成するファイルはグループ・メンバーがアクセスできるように設定します。たとえば、あるユーザが突然いなくなり、別のユーザがその人のプロジェクトを引継がなければならない場合、ソース・ファイルにアクセスできなければ困ります。自分のホーム・ディレクトリにある個人ファイルをプライベート・ファイルとして設定してあっても、`umask` によりグループに対して読み、書き込み権限を与えている場合には、これらの個人ファイルにもアクセスすることができます。ただし、`umask` のメカニズムは、このようなアクセスを防ぐことです。たとえば、プライベート・ファイルのためのディレクトリを作成し、`chmod` コマンドで所有者のみにアクセスを制限することができます。この場合、自分以外の人はそのディレクトリにアクセスすることはできません。

IRIX ユーティリティを使用して、ホーム・ディレクトリのすべてのファイルを自動的に指定したパーミッションに変更することができます。自分のアカウントを設定することにより、ログアウトするたびに指定したファイルとディレクトリに対してこのアクションを適用することができます。たとえば、`personal`、`letters`、`budget` という3つのディレクトリがあるとします。自分のホーム・ディレクトリの `.logout` ファイルに、システムからログアウトするたびに実行するコマンドを登録します。以下に示すコマンドを `.logout` に登録すると、これら3つのディレクトリに自分以外の人アクセスできないように制限することができます。

```
chmod 700 budget personal letters
```

```
chmod 600 budget/* personal/* letters/*
```

umask コマンドは、DAC の重要な部分になります。このコマンドによりファイルのセキュリティを保護し、ファイルのアクセスを快適にすることができます。グループ・メンバーに対して読みおよび書き込み権限のみを許可する場合には、次の行を `.cshrc` あるいは `.profile` ファイルに登録します。

```
umask 006
```

これは、作成したすべてのファイルに次のようなパーミッションを設定します。

```
-rwx-rw----
```

umask を 006 に設定すると、作成したディレクトリは次のようなパーミッションになります。

```
drwxrwx---
```

つまり、所有者とそのグループはそのファイルまたはディレクトリに対してフル・パーミッションを持ちます。特権ユーザ (root) を除く一般ユーザはこのファイルにアクセスできません。

アクセス・コントロール・リスト (ACL: Access Control Lists) と機能

アクセス・コントロール・リスト (ACL: Access Control Lists) と機能に関する詳しい説明は、『IRIX Admin: Backup, Security and Accounting』を参照してください。

システムのパーティション分割

SGI Origin 3000 Series システムのパーティション分割では、大規模な構成のシステム内部が複数のシステムに分割され、図 4-1 に示したようにそれぞれの分割システムが固有のコンソール、root ファイルシステム、および IP ネットワーク・アドレスを所有するように設定されます。各パーティションで再起動、ソフトウェアのロード、電源の切断、更新などを個別に実行することが可能です。パーティションは NUMALink 接続で IP を用いて通信できます。

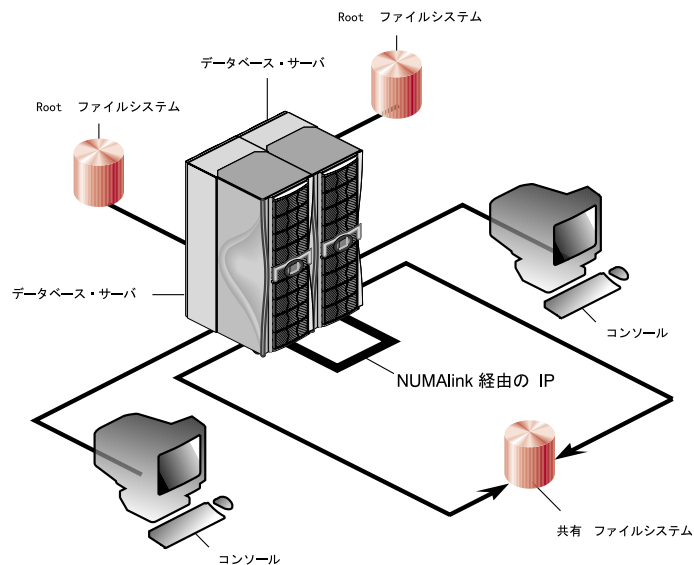


図 4-1 パーティション分割されたシステム

パーティションは異なるサイズで定義することができ、特定のシステムを複数の方法で構成することも可能です。たとえば、128 プロセッサ・システムをそれぞれ 32 プロセッサを持つ 4 つのパーティションに分割することができます。システムのパーティション分割でサポートされている構成については、101 ページの「サポートされる構成」を参照してください。

パーティション・サイズやパーティションの数は、問題の抑制やスケーラビリティに影響します。一般的に、問題の抑制やスケーラビリティを考えた場合、極端な設定を避けることをお勧めします。

問題の抑制の点からみると、パーティション・サイズを大きく設定すると、パーティションで問題が発生した場合に全パーティションがダウンしてしまうため、システム内のほとんどのデータが失われてしまいます。

スケーラビリティの点からみると、単一パーティション内でシステムは独立したシステムであるかのように構成されます。1 つのアプリケーションは、単一のパーティション内でのみ実行可能です。また、複数のパーティションに存在するアプリケーション間の通信には、ネットワーク・プロトコルが必要になります。

パーティション分割の利点

システムのパーティション分割の主な理由の1つに、問題の抑制があります。ハードウェアおよびソフトウェアで障害が発生した場合や、シャットダウンの一部の動作として単一のパーティションをダウンしても、残りのシステムに影響することはありません。ハードウェアのメモリ・バリアは任意の2つのパーティション間で設定されるため、単一のパーティションで構成されている大規模なシステムよりも有用性が高くなります。

パーティション分割の別の有用な使い方として、大きなジョブを実行する必要がある場合、夜中にシステムを1つの大規模なシステムに変え、昼間は複数のシステムに分割することができます。

パーティション分割の欠点

システムのパーティション分割の欠点として、マルチ・スレッド・アプリケーションや共有メモリ・アプリケーションなどの大容量のアプリケーションを実行できなくなったり、そのパフォーマンスが低下することがあります。さらにパーティションに分けられたシステムでは、所有するパーティション以外のファイルシステム、ディスク、ネットワーク・デバイスなどのシステム・リソースにアクセスすることができません。

パーティション分割ソフトウェアのインストール

システムのパーティション分割機能を使用するには、SGI 3000 Series システムに `eo.e.sw.partition` ソフトウェア・パッケージがインストールされている必要があります。また、パーティション分割ソフトウェア・パッケージをインストールした後に、新しいカーネルを設定して、システムを起動する必要があります。

パーティション間のネットワークの設定

パーティション間の通信は、92 ページの図 4-2 に示したように `xpc` (障害許容力メッセージング・システム) を通して行われます。各パーティションのカーネルは、ほかのすべてのカーネルを認識し、ハードウェアの BTE (ブロック転送エンジン) を使用して通信します。 `xpc` ドライバはこのハードウェアを使用します。

xpc ドライバは、複数 (32 個) の論理チャネルを提供します。論理チャネルの一部はカーネル間の通信に使用され、それ以外はロー c1 あるいはネットワーク if_cl デバイスに使用されます。

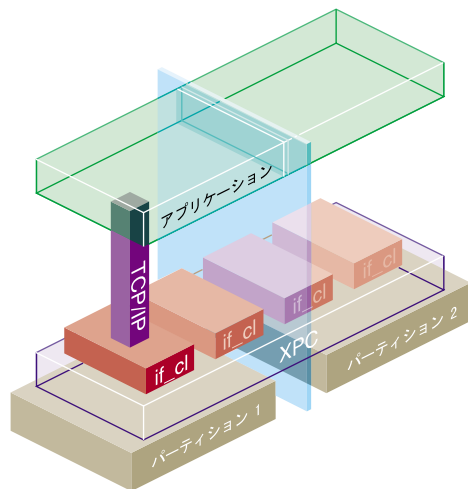


図 4-2 パーティション間の通信

各パーティションは 1 つの c1 ドライバと 1 つの if_cl ドライバを所有します。if_cl ドライバは IP インタフェースを xpc のトップで実行します。if_cl は NUMalink ネットワーク・インタフェースです。アプリケーションは BSD ソケットを使用してパーティション間の通信を実現できます。

c1 デバイスに対しては、文字デバイスのユーザがほかのパーティションと通信するために 16 の全二重論理チャネルが提供されています。相手先のパーティションが起動されると /hw ファイルシステムに各デバイスが現れます。パス名は、次のようになります。ここで *partition_number* と *device_number* は 2 桁の 16 進数の値になります。

```
/hw/xplink/raw/partition_number/device_number
```

if_cl ドライバは、ifconfig コマンドを使用して設定します。詳しい情報については ifconfig(1M) マン・ページを参照してください。if_cl ドライバをネットワーク・ドライバとして設定する手順は、基本的にイーサネット・ドライバ (ef0) の設定手順と同じです。

if_cl ドライバを設定するには手順 4-3 に従ってください。

手順 4-3 if_cl ドライバの設定

1. `root` でログインします。
2. `/etc/config/netif.options` ファイルを編集して、`c10` インタフェースを追加します。
3. `/etc/hosts` ファイルを編集して、`c10` インタフェースのネットワーク・アドレスを追加します。
4. システムを再起動してネットワークを再起動します。

システム・コンソールのコントローラへの接続

システムのパーティション分割は、管理用の機能です。システム・コンソールは、SGI 3000 Series システムのインストール時に選択した構成で指定されたコントローラに接続されています。ケーブルの再接続や詳細については、日本シリコングラフィックス株式会社のサポート部門までお問い合わせください。

パーティションの設定

パーティションの設定には `mkpart` パーティション設定コマンドを使用するか、コマンド・モニタから `setpart` コマンドを使用して実行できます。

`mkpart` パーティション設定コマンド

`mkpart` パーティション設定コマンドでは、以下のタスクを実行できます。

- 現在のパーティションをリストする
- 現在の構成を保存する
- 以下のタスクを実行してパーティションを設定する
 - 既存のシステムを複数のパーティションに分割する
 - パーティションを結合して1つの大きなシステムにする
 - 既存のパーティションを分割し、再編成する

`mkpart` コマンドは、スーパーユーザ特権を持つシステム管理者のみが実行できます。システムが `CAP_NO_SUPERUSER` として設定されている場合は、`CAP_SHUTDOWN`、`CAP_DEVICE_MGT`、および `CAP_SYSINFO_MGT` の `Capability` が有効でなければなりません。プロセスの特権を細か

く制御する **Capability** のメカニズムの詳細については、`capabilities(4)` マン・ページを参照してください。mkpart の影響を受けるすべてのパーティションで、IRIX オペレーティングシステムが実行されている必要があります。

このコマンドの詳細については、mkpart (1M) マン・ページを参照してください。

パーティション・コマンドの典型的な例を次に示します。

1. mkpart -l を実行してすべてのパーティションをリストします。
2. mkpart を再度実行して、システムのパーティション分割を実行します。
 - パーティション ID は、L1 システム・コントローラに保存されます。
 - mkpart は mkpd デーモンを使用して ID をリモート・パーティションの L1 に書込みます。
3. mkpart が正しく実行された後で、変更されたすべてのパーティションを再起動します。

mkpart コマンド行構文は次のとおりです。

```
mkpart [-p partid -m brick ... [ -m brick ... ] ... ] ... [-F]
```

```
mkpart [-i] [-F]
```

```
mkpart [-l]
```

```
mkpart [-n]
```

```
mkpart [-h]
```

mkpart コマンドは、以下のオプションをサポートしています。

`-p partid`

新しい構成のパーティション ID を指定する

`-m brick`

C ブリック（一部の文書ではブリックが使用されています）番号を示す

-i	現在アクティブなパーティションすべてを、0 のパーティション ID を使用して 1 つのシステムに結合する。0 のパーティション ID は、パーティションに分けられていないシステムを示します。
-l	現在のパーティションを表示する
-n	このパーティションのパーティション ID を印刷する
-h	使用領域に関するメッセージを印刷する

注意： -F オプションは慎重に使用してください。mkpart コマンドのユーザは、このオプションを誤って使用した場合にどのような結果になるかを理解しておく必要があります。

-F	すべての完全性チェックを強制的にオーバーライドします。指定したとおりにパーティションを設定し、システムの再起動を試行します。
----	--

PROM からのパーティション分割

パーティションの設定はコマンド・モニタから setpart コマンドを使用しても実行することができます。

setpart コマンドを使ってパーティションを設定するには手順 4-4 に従ってください。

手順 4-4 パーティションに分かれたシステムの設定

1. システムをリセットします。
2. 「System Maintenance」メニューから、コマンド・モニタを示す 5 つ目のオプションを入力します。
3. setpart コマンドを実行します。setpart -h コマンドを入力すると、すべての setpart オプションが表示されます。setpart コマンドは、パーティション番号、パーティションを含む C ブリックの C ブリック番号など、ユーザ入力を受付ける対話式的コマンドです。

4. 設定が完了した後、システムを再びリセットします。システムは2つまたは複数のパーティションで起動されます。各パーティションは、I ブリック、root ディスク、コンソールなどを所有する必要があります。

パーティションを結合して元のシステムに戻すには手順 4-5 に従ってください。

手順 4-5 パーティションの結合

1. すべてのパーティションでコマンド・モニタになります。
2. 任意のパーティションで `setpart -c` コマンドを実行します。このコマンドは、すべてのパーティション情報をクリアし、すべてのパーティションを同時にリセットします。その結果、システム全体がリセットされ、単一のパーティションに戻ります。

partmgr パーティション構成ツール

`partmgr(1M)` コマンドは、SGI Origin 3000 Series のサーバ用のパーティション分割管理ツールを起動します。このツールを使用して、以下を含むパーティション定義を定義および保存できます。

- ブリックおよびラックのパーティション ID 情報
- 関連する `prom` 環境変数

このコマンドを使用できるのは、スーパーユーザ特権を持つシステム管理者だけです。システムが `CAP_NO_SUPERUSER` として設定されている場合は、`CAP_SHUTDOWN`、`CAP_DEVICE_MGT`、および `CAP_SYSINFO_MGT` の `Capability` が有効でなければなりません。プロセスの特権を細かく制御する `Capability` のメカニズムの詳細については、`capabilities(4)` マン・ページを参照してください。

システムを新しいパーティション・スキームに変更するには、影響を受けるパーティションをすべて再起動する必要があります。

パーティション ID 0 は、オペレーティング・システムにおいて特別な意味を持ち、パーティション分割システムで使用することはできません。有効なパーティション ID は、1 ~ 63 の範囲（これらの数値を含む）の任意の値です。

この節で使用するブリックという用語は、C ブリック・モジュールを指します。基本的な C ブリック・モジュールは、4 個の MIPS CPU と最大 8GB のローカル・メモリで構成されます。

partmgr(1M) コマンドを使用するには、以下の必要条件を満たす必要があります。

- eoe.sw.partition ソフトウェア・パッケージをインストールする必要があります。
- L1 および L2 ファームウェアが最新である必要があります。
- partdb enable L1 コマンドを使用して、すべての L1 でパーティション分割管理を有効にする必要があります。
- partmgr を動作させるには、すべてのパーティションで mkpd デーモンが実行されている必要があります。すべてのパーティションで実行されていない場合は、partmgr コマンドでリモート・パーティションをシャットダウンできません。
- PROM がモード定義で定義されている環境変数を使用するには、PROM 環境変数 AutoPROMVars を 1 に設定する必要があります。

このコマンドの詳細については、partmgr(1M) マン・ページを参照してください。

partmgr コマンド行構文は、以下のとおりです。

```
partmgr [-h]
```

```
partmgr [-l]
```

```
partmgr [-a]
```

```
partmgr [-n]
```

```
partmgr [-v]
```

```
partmgr -c [-m modename]
```

```
partmgr -x modename [-F]
```

```
partmgr -s modename [-F]
```

```
partmgr -i [-F]
```

```
partmgr -d [-F] -f file
```

```
partmgr -d [-F] -m modename -u [-e var=value]...
```

```
partmgr -d [-F] -m modename [-p part -b bricks [-e var=value]...]...
```

```
partmgr -d [-F] -m modename [-p part -r racks [-e var=value]...]..
```

partmgr コマンドでは、以下のオプションがサポートされています。

-h

ヘルプ。使用方法メッセージを出力します。

-l

すべての既知のシステム・ブリックのブリック ID を標準出力にリストします。

-a

現在定義されているモードをすべてリストします。

-n

現在のパーティション ID を表示します。

-v

現在のパーティションの環境変数を表示します。

-c

モード構成を表示します。モードを指定 (-m *modename*) しなかった場合は、現在のモードが表示されます。

-x *modename*

モードを削除します。

-s *modename*

特定のモードを現在のモードに設定します。一部のパーティションを再起動しなければならない場合があります。

-i

-s *Unpartitioned* と同じです。システムをパーティション分割されてないデフォルトの状態に初期化します。

-d

新しいモードを定義します。

- `-m modename`
モード名を指定します。このオプションは、`-c` および `-d` オプションとともに使用します。
- `-u`
新しいモードを定義する際に、パーティション分割されていないシステムを指定するために使用します。
- `-p part`
パーティション ID を指定します。パーティション ID は、1 ~ 63 の範囲（両端の数値を含む）の数字です。`-p`、`-b`、`-r`、および `-e` オプションの順序は重要です。`partmgr(1M)` マン・ページを参照してください。
- `-b bricks`
ブリック ID を指定します（例：001c13）。
- `-r racks`
ラック ID を指定します（例：001）。
- `-e var=value`
環境変数の名前と値のペアを指定します。
- `-f file`
新しいモードを定義する際に、入力ファイルを指定するために使用します。
- `-F`
一部の完全性チェックを強制的にオーバーライドします。このオプションは慎重に使用してください。`partmgr` を使用するユーザは、このオプションを使用した結果を理解しておく必要があります。

注意：`partmgr` コマンドは慎重に使用してください。使い方を誤ると、システム内のすべてのパーティションが完全にフリーズしてしまい、PROM 起動もできなくなる可能性があります。

例 1

それぞれが 4 つのブリックで構成される 2 つのパーティションを持つ 8 ブリックのシステムに対してモードを定義するには、以下を実行します。

```
partmgr -d -m modename -p 1 -b 001c10,001c13,001c16,001c21 -p 2 -b
001c24,001c29,001c32,001c35
```

例 2

PROM 環境変数 *AutoBoot* が **No** に設定されているパーティション分割されていないシステムに対してモードを定義するには、以下を実行します。

```
partmgr -d -m modename -u -e AutoBoot=No
```

例 3

以下の 3 つの例は、入力ファイル (-f オプション) を使用した場合のもので、先頭のスペースは入力してもかまいませんが、必須ではありません。ここでは読みやすくするために追加してあります。

a.

```
MODE modename1

PARTITION 1
BRICKS
    001c10 001c13 001c16 001c21

ENVIRON
    AutoBoot=No

PARTITION 2
BRICKS
    002c10 002c13 002c16 002c21

ENVIRON
    AutoBoot=No
    OSLoadPartition=scsi(0)disk(1)rdisk(0)partition(0)
    SystemPartition=scsi(0)disk(1)rdisk(0)partition(8)
```

b.

```
MODE modename2

PARTITION 1
RACKS
    001 002
```

```
PARTITION 2
RACKS
003 004
```

c.

```
MODE modename3

UNPARTITIONED

ENVIRON
AutoBoot=No
OSLoadPartition=scsi(0)disk(1)rdisk(0)partition(0)
SystemPartition=scsi(0)disk(1)rdisk(0)partition(8)
netaddr=128.162.192.65
```

サポートされる構成

図 4-3 は、システムのパーティション分割でサポートされる構成です。各構成ごとに、C ブリックの合計数、各パーティション内の C ブリックの数、および各パーティション内のプロセッサの最大数が示されています。詳細については、『SGI Origin 3000 Owner's Technical Configuration Manual』を参照するか、日本シリコングラフィックス株式会社のサポート部門にお問い合わせください。

C ブリック	各パーティション内の C ブリック数	パーティション内のプロセッサの最大数
8	2 x 4 C ブリック	16
16	2 x 8 C ブリック	32
	4 x 4 C ブリック	16
32	2 x 16 C ブリック	64
	4 x 8 C ブリック	32
64	2 x 32 C ブリック	128
	4 x 16 C ブリック	64
128	2 x 64 C ブリック	256
	4 x 32 C ブリック	128

図 4-3 サポートされている構成

メタルータとは、ほかのルータとのみ接続するルータのことです。メタルータは、32 個を超える C ブリックを含むシステムで必要となります。

パーティション分割のガイドライン

システムのパーティション分割を実行する場合は、次のガイドラインに注意してください。

- 1つのパーティションは、1つまたは複数の4プロセッサCブリックで構成されている必要があります。
- 各パーティションに1～63の固有なID番号を割り当てる必要があります。
- 各パーティションは次のコンポーネントを含んでいる必要があります。
 - サイズが64Cブリック以下のシステムの場合は最低1個のCブリック。サイズが65Cブリック以上のシステムの場合は、4の倍数個のCブリック（最小限）
 - 1個のIブリック（最小限）
 - 1台のrootディスク
 - 1個のコンソール接続

4. パーティション内の最小番号の C ブリックのコンソールが、そのパーティションのコンソールになります。元のコンソールは、そのままそのコンソールがあるパーティションのコンソールとして維持されます。
5. 各パーティションは独自の PROM 環境変数のセットを持ちます。これらは *ConsolePath*、*OSLoadPartition*、*SystemPartition*、*netaddr* および *root* です。
6. パーティション内のすべてのブリックは物理的に隣接している必要があります。また、同じパーティション内の任意の 2 つのプロセッサ間のルートは、ほかのパーティションを経由せず、パーティション内に含まれている必要があります。パーティション内のブリックが隣接していない場合、システムは起動しません。

PCI ホットプラグのサポート

SGI Origin 3000 Series システムでは、システムの電源が入っているときに、認定コントローラ・カードの追加と取外しの両方を行うことができます。個々のデバイス・ドライバは、ホットプラグ挿入、ホットプラグ取外し、またはその両方をサポートしていることも、いずれもサポートしていないこともあります。ドライバがホットプラグ挿入と取外しの両方をサポートしている場合は、そのドライバでサポートされているカードをシステムの実行中に交換できます。

PCI ホットプラグ操作をサポートしているのは、認定デバイス・ドライバのみです。このような操作は、カーネルに静的にリンクされたドライバおよびロード可能ドライバで利用可能です。特定の認定デバイス・ドライバの詳細については、SGI サービス担当者までご確認ください。

メモ：認定ドライバのないカードはホットプラグ挿入または取外しできません。

PCI バス速度の制約

カードが一枚も挿入されていないバスには、66 MHz カードのみをインストールできます。すでに別のカードがインストールされているバスにインストールしたカードは、既存のカードと同じバス速度で動作します。新しいカードをインストールすることによって既存のカードのバス速度との不一致が生じる場合、ホットプラグ操作は失敗します。P ブリックにはそれぞれ 2 つのスロットを持つ 6 つのバスがあるため、33 MHz および 66 MHz で動作するカードをインストールでき

るよう、起動時に各スロットにカードをインストールしておくことができます。66 MHz カードは 33 MHz でも動作可能で、33 MHz バスに挿入できます。

挿入手順

手順 4-6 に、Qlogic Fiber Channel および SCSI コントローラ・カードなどの認定カードのインストール方法を説明します。

手順 4-6 `pciconfig` コマンドを使用した PCI カードのインストール

1. 実行中の SGI Origin 3000 Series システムの P ブリックまたは I ブリックの空き PCI スロットに、サポートされているカードを挿入します。

PCI カードを物理的に P ブリックまたは I ブリックにインストールします。まず、カードを PCI キャリアに取付け、ブリック背面のスロットに挿入します。ブリックの各スロットは、スロット最上部にあるバスおよびスロット番号ラベルで識別できます。タブを押して、カードをスロットに装着します。

PCI カードのインストールの詳細については、システムに付属の『SGI Origin 3000 Series Owner's Guide』を参照してください。

2. `pciconfig` コマンドを使用してカードをオンラインにします。

次の例では、`pciconfig` コマンドを使用して PCI スロットの電源をオンにし、インストールされているカードのタイプを判断し、適切なデバイス・ドライバを呼出してカードとソフトウェアを初期化します。

```
pciconfig -b /hw/module/001c16/Pbrick/bus/3/pci -s 1 -u  
-b オプションで /hw/module/001c16/Pbrick/bus/3/pci を P ブリックのバス 3 へのバスとして指定し、-s オプションでスロット 1 を指定します。-u オプションは、指定したスロットにカードをホットプラグ挿入することを示します。これらのオプションの詳細については、pciconfig(1M) マン・ページを参照してください。
```

3. ハードウェア・グラフで新しいデバイスのエントリをチェックします。

正常にホットプラグ挿入できると、システムはカードが起動時にインストールされていた場合と同様に動作します。通常、新しいデバイスのエントリはハードウェア・グラフに表示されます。

取外しとクエリ操作

`pciconfig` コマンドを使用して、実行中のシステムからカードを取外したり、ホットプラグ操作の前後にスロットのステータスをクエリすることもできます。これらの操作の詳細については、`pciconfig(1M)` マン・ページを参照してください。

マルチユーザ環境でのシステム管理

この章では、ユーザをサポートするために必要なシステム構成と管理方法について説明します。また、ユーザ・アカウントの管理、ユーザ環境の設定、ユーザ間の通信についても説明します。

ユーザ・アカウント管理

最も一般的なシステム管理作業は、ユーザ・アカウントの作成と削除です。サイト管理者はこの節の内容を十分理解した上でユーザ・アカウントの設定や変更を行ってください。グラフィカルな「システム・マネージャ (System Manager)」ツール (グラフィック・ワークステーションでのみ利用可能) は、これらの作業を行うために非常に便利なツールです。システム・マネージャについての詳細は、『Personal System Administration Guide』を参照してください。このマニュアルでは、コマンド行から実行する方法について説明します。

ユーザ ID

各ユーザ・アカウントにはユーザ ID 番号が割当てられます。これらの番号は各ワークステーションで一意であり、サイト全体でも一意にする必要があります。アカウントのユーザ ID は、`/etc/passwd` ファイルの 3 番目のフィールドに保持されています。

あるユーザ・アカウントをクローズしても、そのアカウントのユーザ ID 番号は再使用できません。古い ID 番号で所有されているファイルがシステム内に残っていたり、ファイルが古いバックアップ・テープからシステムにコピーされる可能性があるからです。これらのファイルは、新しいユーザを古い ID 番号に関連付けることによってアクセス可能になります。原則として、一般的に、ユーザに割当てられる ID 番号は永久的にユーザに属します。ユーザ ID 番号についての詳細は、`passwd(4)` マン・ページを参照してください。

グループ ID

各ユーザ・アカウントは、システムを構成するグループの 1 つに属しています。同じ業務に関わるユーザは 1 つのグループにできます。たとえば、出版部門のメンバーを *pub* というグループに入れます。このようなユーザ設定方法の利点は、グループ内のユーザがシステムのすべてのユーザとではなくグループ内部だけでファイルやリソースを共有できることです。

各グループにはグループ ID 番号が割当てられます。これらの番号は各ワークステーション上で一意であり、サイト全体でも一意にする必要があります。ユーザ ID と同じように、グループ ID も再利用できません。

新しいファイルを作成すると、そのファイルにはグループ ID が割当てられます。ファイルのグループ ID は、`chgrp(1)` コマンドを使用して変更できます。ファイルの所有者または実行権を持つほかのユーザは、ファイルのパーミッション・フィールドを操作することによって、グループのほかのメンバーに読み込み、書き込み、実行の特権を与えることができます。

グループに関する情報は、`/etc/group` ファイル内に保持されます。次に、このファイルのエントリの例と説明を示します。

```
raccoons::101:norton,ralph
```

各エントリは 1 行で構成され、各行には次のフィールドがあります。

- | | |
|-------|--|
| グループ名 | 任意の長さのグループ名。ただし、コマンドによっては最初の 8 文字だけが使用されます。最初の文字は英字でなければなりません。 |
| パスワード | 暗号化されたパスワード。上記の例のようにこのフィールドが空の場合は、パスワードは必要ありません。 <code>passwd(1)</code> コマンドでは、グループ・パスワードの設定や変更は行えません。グループにパスワードを設定するには、まず <code>passwd</code> コマンドを使用してパスワードを暗号化します。この目的のために作成されたテスト用のユーザ・アカウントを使用し、終了後に削除してください。次に、暗号化されたパスワードを <code>/etc/passwd</code> ファイルから <code>/etc/group</code> の保護対象のエントリにそのままコピーします。 <code>newgrp</code> コマンドを使用して、あるユーザをパスワードで保護されたグループに変更する場合、 <code>/etc/group</code> ファイルにグループ・メンバーとしてエントリされているユーザはパスワードの入力は要求されませんが、その他のユーザはパスワードを入力する必要があります。ただし、ユーザ・グループのパスワードによる保護はほとんど使用しません。 |

グループ ID 0 から 60,000 までの番号。番号にはカンマを挿入しないでください。100 以下の番号はシステム・アカウント用に予約されています。

ログイン名 カンマで区切られたグループ・メンバーのログイン名。

ユーザ・グループについての詳細は、group(4) マン・ページを参照してください。

プロジェクト ID 番号

多くのサイトでは、特定のシステムの使用量に対して個々の部門に別々に課金する必要があります。通常、これはシステムの各ユーザ ID の総使用量を該当する部門に課金することで行います。ただし、サイトによっては 2 つ以上の部門で作業するユーザがいるため、こうしたユーザの総使用量を単一の部門に課金することは適切ではありません。このようなユーザに複数のユーザ ID またはグループ ID を与えることは可能ですが、管理上難しい可能性があります。また、アカウント作業のためにオペレーティング・システムのセキュリティ機能を使用していることになってしまいます。したがって、IRIX ではプロジェクト ID という概念を採用しています。

プロジェクト ID はグループ ID と同様ですが、以下の 2 つの点が大きく異なります。

- 現在のプロジェクト ID は、個々のプロセスではなく Array セッション全体に関連付けられます。
- プロジェクト ID はアクセス権に影響を与えません。プロジェクト ID はアカウントिंगを主な目的にしており、実際に拡張アカウントING情報で報告されます（詳細については、extacct(5) を参照してください）。

ユーザは、自分のユーザ ID に関連付けられたデフォルトのプロジェクト ID を 1 つ持ちます。別のプロジェクトに課金すべき作業を行わなければならない場合は、newproj(1) コマンドを使用して、新しいプロジェクト ID に切替えることができます。newproj コマンドではユーザの UID と GID は変更されないため、アクセス権に影響はありません。newproj コマンドは、以前のシェルのバックグラウンドで実行されているプログラムが引続き以前のプロジェクト ID に従ってアカウントINGの対象となるよう、新しいシェルと Array セッションを起動します。

認証されていないプロジェクトをユーザが指定できないようにするため、newproj コマンドは、各ユーザに対して有効なプロジェクトをリストしたファイル (/etc/project) を参照します。/etc/project ファイルの形式は、/etc/passwd ファイルまたは /etc/group ファイルに似ています。詳細については、project(4) マン・ページを参照してください。プロジェクト ID は

数値なので、追加のファイル (/etc/projid, projid(4) を参照) を使用して、/etc/project で使用される ASCII プロジェクト名と数値のプロジェクト ID を対応付けます。これらのファイルでユーザが見つからない場合は、標準のデフォルト・プロジェクト ID が使用されます。systune(1) を使用して *dfltprid* 変数を変更すると、このデフォルト値を変更できます。

プロジェクト ID の詳細については、projects(5) マン・ページを参照してください。

ファイルのプロジェクト ID を変更するには、chproj(1) コマンドを使用します。

chproj の構文は、次のとおりです。

```
chproj [ -Rfh ] project file ...
```

chproj コマンドは、*files* のプロジェクト ID を *project* に変更します。プロジェクトは、プロジェクト・ファイルで見つかったプロジェクト名、または 10 進数のプロジェクト ID のいずれかになります。

-f (*force*) オプションを指定した場合、エラーは報告されません (使用法のエラーを除く)。

-R オプションを指定した場合、コマンドは引数で指定されるディレクトリを再帰的に下降し、指定した所有者またはグループを設定します。シンボリック・リンクが見つかった場合に -h オプションが指定されていないければ、シンボリック・リンク先もチェックされて、シンボリック・リンクで参照されているファイルの所有者、グループ、またはプロジェクトが変更されます。

-h オプションを指定したときにファイルがシンボリック・リンクの場合は、シンボリック・リンクの所有者、グループ、またはプロジェクトが変更されます。このオプションがない場合、シンボリック・リンクで参照されているファイルの所有者、グループ、またはプロジェクトが変更されます。

スーパーユーザ以外が chproj コマンドを起動した場合、ファイル・モードの set-user-ID および set-group-ID ビット (それぞれ 04000 および 02000) がクリアされます。

ファイルのプロジェクトを変更できるのは、スーパーユーザまたは CAP_SETUID 機能を持つユーザのみです。

chproj コマンドの詳細については、chown(1) マン・ページを参照してください。

シェル・コマンドによるユーザ・アカウントの追加

状況に応じて、「システム・マネージャ (System Manager)」などの自動化ツールを使用せずに手動でユーザ・アカウントを追加する必要があります。すべての管理者は、自動化ツールに問題が生じた場合や、サイトでのユーザ・アカウント管理用に独自の特別なスクリプトやプログラムを作成する場合の手順を理解しておく必要があります。また、作業終了後は、`pwck` コマンドを使用して作業が正しく行われたことを確認してください。

手順 5-1 で、ユーザ・アカウントを手動で追加する方法を説明します。

手順 5-1 ユーザ・アカウントの追加

1. `/etc/passwd` ファイルを編集します。
2. `/etc/group` ファイルを編集します。
3. ユーザのホーム・ディレクトリとスタートアップ・ファイルを作成します。
4. 新しいアカウントを確認します。

上記の手順を、以下の 4 つのセクションでさらに詳しく説明します。

ユーザ・アカウント追加のための `/etc/passwd` の編集

手順 5-2 で、`/etc/passwd` ファイルを編集する方法を説明します。

手順 5-2 `/etc/passwd` ファイルの編集

1. `root` でログインします。
2. 任意のテキスト・エディタを使用して、`/etc/passwd` ファイルを編集します。

`/etc/passwd` ファイルには、システムの各アカウントごとに 1 行のエントリがあります。各エントリは、コロンで区切られた 7 つのフィールドで構成されます。次にエントリの例を示します。

```
ralph:+:103:101:Ralph Cramden:/usr/people/ralph:/bin/csh
```

3. エントリのうちの 1 行（たとえばファイルの最後の行）をコピーし、ファイルの最後のアカウントの後ろに追加します。
4. 最初のフィールド（この例では `ralph`）を新しいアカウントの名前（たとえば `alice`）に変更します。

5. アカウント名の後のコロンと2番目のコロンの間にある文字を削除します。これらの文字を削除すると、アカウントからパスワードが削除されます。パスワードは、管理者または新しいユーザが後で追加できます。
6. 次のフィールド（この例では 103）は新しいアカウントのユーザ ID 番号です。現在システムで使用されている一番大きなユーザ ID 番号よりも1大きい番号に変更します。0 から 100 までの番号はシステム用に予約されているので、ユーザ ID 番号には使用できません。
7. 次のフィールド（この例では 101）は新しいアカウントのグループ ID 番号です。/etc/group ファイルを調べて、新しいユーザ・アカウントに適したグループ ID を選択します。/etc/group ファイルには、システム内のすべてのグループがグループ ID 順に並んでおり、続けてそのグループに現在所属しているユーザのリストがあります。
8. 次のフィールド（この例では *Ralph Cramden*）を新しいユーザ名（この例では *Alice Cramden*）に変更します。このフィールドにはオフィスや電話番号も追加できます。次の例に示すように、ユーザ名の後にカンマを挿入し、オフィスの所在地を入力し、さらにカンマを挿入して、電話番号を入力します。

:Alice Cramden, Brooklyn, (212) 555-1212:

これらのフィールドには任意の情報を記載できます。これらのフィールドは、finger(1) プログラムによって、それぞれユーザ名、オフィス、電話番号と解釈されます。
9. 次のフィールド（この例では /usr/people/ralph）はユーザのホーム・ディレクトリです。新しいユーザ・アカウント名を反映させるためにこのフィールドを変更します。この例では、/usr/people/ralph を /usr/people/alice に変更します。
10. 最後のフィールド（この例では /bin/csh）はユーザのログイン・シェルです。ほとんどのユーザは、C シェル (/bin/csh)、Korn シェル (/bin/sh)、Bourne シェル (/bin/bsh) のいずれかを使用します。特別なシェルを使用しない場合は、このフィールドを変更する必要はありません。特別なシェルについての詳細は、135 ページの「特別なログイン・シェル」を参照してください。シェルを選択すると、/etc/passwd ファイルの編集は終わりです。
11. 変更内容を保存してエディタを終了します。

ユーザ追加のための /etc/group の編集

手順 5-3 では、/etc/group ファイルに新しいユーザを追加します。ただし、この操作は省略することができます。ユーザは、/etc/group ファイルに登録されていなくても、グループのメ

ンバーになることができます。ユーザが所属するグループのリストを保守する場合は、このファイルを編集します。

手順 5-3 /etc/group ファイルの編集

1. 任意のテキスト・エディタで /etc/group ファイルを開きます。次のような行が表示されます。

```
sys::0:root,bin,sys,adm
root::0:root
daemon::1:root,daemon
bin::2:root,bin,daemon
adm::3:root,adm,daemon
mail::4:root
uucp::5:uucp
rje::8:rje,shqer
lp:*:9:
nuucp::10:nuucp
bowling:*:101:ralph
other:*:102:
```

2. 新しいアカウント名（この例では *alice*）を任意のグループの後に追加します。次の例に示すように、アカウント名とほかのアカウント名の間はカンマで区切ります。スペースを挿入しないでください。

```
bowling:*:101:ralph,alice
```

アカウント名を /etc/group ファイルに追加するかどうかは任意ですが、どのユーザがどのグループに属するかを管理しておくことをお勧めします。

また、/etc/group 内のさまざまなグループ名の後にアカウント名を追加することによって、単一のアカウントを複数のグループに割り当てることができます。ユーザは、newgrp コマンドまたは multgrps コマンドを使用して、グループへの所属関係を変更できます。

3. 変更内容を保存してエディタを終了します。

新しいユーザのためのホーム・ディレクトリの設定

新しいユーザのホーム・ディレクトリを作成し、シェル・スタートアップ・ファイルをそのディレクトリにコピーするには手順 5-4 に従ってください。

手順 5-4 ホーム・ディレクトリの設定

1. `mkdir(1)` コマンドを使用して、ユーザのホーム・ディレクトリを作成します。たとえば、ユーザ `alice` のホーム・ディレクトリを作成するには、次のように入力します。

```
mkdir /usr/people/alice
```

ディレクトリの所有者を `alice`、グループを `bowling` にします。

```
chown alice /usr/people/alice
```

```
chgrp bowling /usr/people/alice
```

新しいホーム・ディレクトリに、サイトに対する適切なアクセス権があることを確認します。サイトへのセキュリティを緩やかにするには、次のように入力します。

```
chmod 755 /usr/people/alice
```

詳細については、`chown(1)`、`chgrp(1)` および `chmod(1)` の各マン・ページを参照してください。

2. シェル・スタートアップ・ファイルを新しいユーザのホーム・ディレクトリにコピーします。新しいアカウントで C シェルを使用する場合は、次のように入力します。

```
cp /etc/stdcshrc /usr/people/alice/.cshrc
```

```
cp /etc/stdlogin /usr/people/alice/.login
```

新しいアカウントで Korn シェルまたは Bourne シェルを使用する場合は、次のように入力します。

```
cp /etc/stdprofile /usr/people/alice/.profile
```

3. シェル・スタートアップ・ファイルの所有者は、`root` のままにしておくこともユーザに変更することもできます。どちらの場合もユーザがシステムにログインする方法に影響はありません。ただし、所有者を `root` のままにしておくことにより、ユーザが間違ってファイルを変更し、ログインできなくなる可能性は低くなります。

ユーザがシェル・スタートアップ・ファイルにアクセスできるようにするには、`chmod` コマンドを使用します。次に、C シェルの場合の例を示します。

```
chmod 755 /usr/people/alice/.cshrc /usr/people/alice/.login
```

次に、Korn シェルまたは Bourne シェルの場合の例を示します。

```
chmod 755 /usr/people/alice/.profile
```

ユーザのホーム・ディレクトリ内に *root* が所有するユーザ・ファイルがほかにある場合、それらのファイルも変更してください。

4. 新しいユーザのパスワードを直ちに作成することをお勧めします。次のコマンドを入力します。

```
passwd alice
```

次に、表示されるプロンプトに従ってパスワードを作成します。設定したパスワードをユーザに知らせ、そのパスワードを各自で変更するように伝えます。ユーザのパスワード変更に関する注意については、『IRIX Admin: Backup, Security, and Accounting』を参照してください。

新しいアカウントの確認

`pwck` コマンドを実行して、作業が正しく行われたことを確認します。このコマンドは `/etc/passwd` ファイルについて簡単なチェックを行い、ユーザ ID 番号が重複して使用されていないかどうか、またフィールドに正しく値が入力されているかどうかを確認します。作業が正しく行われると、例 5-1 のようなメッセージが出力されます。

例 5-1 pwck コマンド出力

```
sysadm*:0:0:System V Administration:/usr/admin:/bin/sh
    Login directory not found
auditor::11:0:Audit Activity Owner:/auditor:/bin/sh
    Login directory not found
dbadmin::12:0:Security Database Owner:/dbadmin:/bin/sh
    Login directory not found
tour::995:997:IRIS Space Tour:/usr/people/tour:/bin/csh
    Login directory not found
4Dgifts::999:998:4Dgifts Acct:/usr/people/4Dgifts:/bin/csh
    First char in logname not lower case alpha
    1 Bad character(s) in logname
    Login directory not found
nobody*:-2:-2::/dev/null:/dev/null
    Invalid UID
    Invalid GID
```

上記の内容は、`pwck` によって出力される正常なメッセージです。`pwck` が生成するエラーについての詳細は、`pwck(1M)` マン・ページを参照してください。

シェル・コマンドによるグループの追加

手動でシステムにグループを追加するには手順 5-5 に従ってください。

手順 5-5 システムへのグループの追加

1. `root` でログインします。
2. `/etc/group` ファイルを編集します。このファイルには、システムのグループ・リストが記載されています。各行に 1 つのグループの情報があり、グループ名、パスワード（オプション）、グループ ID 番号、そのグループに所属するユーザ・アカウントの順に並んでいます。

たとえば、グループ ID が 103 である `raccoons` というグループを作成するには、次の行をファイルの終わりに追加します。

```
raccoons:*:103:
```

3. グループに新たに所属するユーザが存在する場合は、そのユーザ名を最後のフィールドに追加します。次の例に示すように、各ユーザ名はカンマで区切ります。

```
raccoons:*:103:ralph,norton
```

4. ファイルを保存して編集を終了します。`/etc/passwd` と `/etc/group` のグループ ID が必ず一致していることを確認してください。

ユーザ・グループについての詳細は、`group(4)` マン・ページを参照してください。

ユーザ・グループの変更

ユーザが所属するグループを変更するには手順 5-6 に従ってください。

手順 5-6 ユーザ・グループの変更

1. `root` でログインします。
2. `/etc/group` ファイルを編集します。ユーザの所属するグループの行にそのユーザのアカウント名を入力します。アカウント名が別のグループに登録されている場合で、両方のグループにこのアカウントを登録しない場合は、どちらかのアカウント名を削除します。
3. 変更内容を保存し、`/etc/group` ファイルの編集を終了します。
4. `/etc/passwd` ファイルを編集します。
5. ファイル内でユーザのエントリを検索します。

6. そのエントリの古いグループ ID を新しいグループ ID に変更します。グループ ID は `/etc/passwd` の 4 番目のフィールド（アカウント名、パスワード、ユーザ ID の後）です。
7. 変更内容を保存し、編集を終了します。

これで、ユーザの所属するグループが変更されました。ユーザに、所有するファイルに対するグループの所有権を変更するよう伝えてください。この変更は、管理者 (*root*) が `find` と `chgrp` を使用して行うこともできます。詳細については、21 ページの「`find` コマンドによるファイルの検索」を参照してください。

システムからのユーザの削除

手順 5-7 で、ユーザのホーム・ディレクトリとその下にあるすべてのファイルやディレクトリを削除する方法を説明します。ユーザのファイルやその他の情報は残し、ユーザ・アカウントだけを閉じたい（または使用不可にしたい）場合は、118 ページの「ユーザ・アカウントのロック」を参照してください。

手順 5-7 ユーザ・アカウントの削除

ユーザ・アカウントを完全に削除するには以下の手順に従ってください。

1. `root` でログインします。
2. 後でユーザ・ファイルのコピーが必要になると考えられる場合は、`tar(1)` または `cpio(1)` のマン・ページを使用し、カートリッジ・テープなどにディレクトリのバックアップ・コピーを作成してください。
3. `/etc/passwd` ファイルを編集し、暗号化されたパスワード（`shadow` パスワードを使用している場合は “+” 記号）を次の文字列に置換えます。

ACCOUNT CLOSED

アスタリスクは、この例に従って使用してください。暗号化されたパスワードのフィールドにアスタリスクを使用すると、そのアカウントにログインできません。アカウントを使用不可にするには、パスワードのフィールドに 12 文字以内の文字を入力する方法もありますが、アスタリスクと明確なロック・メッセージを併用する方をお勧めします。

4. `find(1)` を使用して、ユーザが所有するシステム内のすべてのファイルを検索し、削除するか所有権を変更します。`find` の使用方法についての詳細は、21 ページの「`find` コマンドによるファイルの検索」または `find(1)` マン・ページを参照してください。

システムからのグループの削除

システムからグループを削除するには手順 5-8 に従ってください。

手順 5-8 システムからのグループの削除

1. `/etc/group` ファイルを編集し、該当するグループのエントリを使用されていない新しい名前に変更します。たとえば、`bigproject` グループを `bigproject.closed` に変更します。
2. `/etc/passwd` ファイルを編集し、関連するユーザ・エントリから該当するグループを削除します。
3. `find` を使用して変更前のグループに所属するすべてのファイルやディレクトリを検索します。次に `chgrp` コマンドを使用して、それらのファイルやディレクトリを既存のグループに関連付けます。

ユーザ・アカウントのロック

ユーザ・アカウントへのログインや、`su` コマンドによるユーザ ID 番号への切替えが行われないように、必要に応じて、ユーザのアカウントをクローズできます。手順 5-9 に従ってください。

手順 5-9 ユーザ・アカウントのロック

1. `root` でログインします。
2. `/etc/passwd` ファイルを開き、閉じるユーザ・アカウントのエントリを検索します。
3. 次のように、行の先頭にシャープ記号 (`#`) を入力し、上記アカウントのエントリをコメントにします。

```
# ralph:+:103:101:Ralph Cramden:/usr/people/ralph:/bin/csh
```
4. さらにセキュリティを確保する手段として、次のように暗号化されたパスワード（エントリの 2 番目のフィールド）を有効なパスワードとは解釈されない文字列に置換えます。

```
# ralph:*:103:101:Ralph Cramden:/usr/people/ralph:/bin/csh
```

アスタリスクを入力することにより、意図的にアカウントを閉じたことが後でわかります。
5. 必要に応じて、次のコマンドを使用して、ユーザのホーム・ディレクトリを閉じることもできます。

```
chown root /usr/people/ralph
chgrp bin /usr/people/ralph
chmod 700 /usr/people/ralph
```

上記のコマンドによって、ユーザのアカウントはロックされ、*root* だけがユーザのホーム・ディレクトリにアクセスできる状態になります。将来このユーザがシステムにアクセスする可能性がある場合には、システムから完全に削除する方法ではなく（118 ページの「システムからのグループの削除」を参照）、以下に説明するアカウントを閉じる方法を使用してください。

newgrp および multgrps によるユーザ・グループの変更

通常、ユーザは一度に 1 つのグループにしか所属しませんが、*newgrp* コマンドを使用してグループを一時的に変更できます。グループを変更すると、管理作業を実施する上で便利な場合があります。

特権ユーザは、*/etc/groups* にリストされたグループであればどれにでも所属できます。その他のユーザの場合は、*newgrp* を使用して一時的にグループを変更するためには、変更先のグループのメンバーとして登録されていなければなりません。詳細については、*newgrp(1)* マン・ページを参照してください。

multgrps コマンドを使用して、同時に複数のグループに所属できます。この場合、各ユーザが作成するファイルのグループ ID は、*multgrps* コマンドを実行する前のグループ ID になります。各ユーザは、新たに所属したグループ ID と同じグループ ID を持つファイルに対し、グループのメンバーとしてのアクセス権を取得します。詳細については、*multgrps(1)* マン・ページを参照してください。

ユーザのグループ変更は、*/etc/groups* ファイル内でメンバーとして登録されているグループに対してのみ許可されます。現在所属しているグループを確認する場合は、*id(1M)* コマンドを使用します。

ユーザ情報の変更

この節では、個々のユーザのログイン情報に関連する値の変更方法について説明します。ここで使用するコマンドは、NIS タイプのエントリとしてインストールされたログインに対しては適用できません。ログイン・アカウントが NIS タイプの場合は、ネットワーク・サーバのマスター・

パスワード・ファイルを変更する必要があります。NIS についての詳細は、『NIS Administrator's Guide』を参照してください。

ここでは、以下について説明します。

- 120 ページの「ユーザのログイン名の変更」
- 121 ページの「passwd コマンドによるユーザのパスワードの変更」
- 122 ページの「ユーザのログイン ID 番号の変更」
- 123 ページの「ユーザのデフォルト・グループの変更」
- 123 ページの「ユーザのコメント・フィールドの変更」
- 124 ページの「ユーザのデフォルト・ホーム・ディレクトリの変更」
- 125 ページの「ユーザのデフォルト・シェルの変更」

ユーザのログイン名の変更

ユーザのログイン名を変更するには手順 5-10 に従ってください。

手順 5-10 ユーザのログイン名の変更

1. /etc/passwd ファイルを編集し、ユーザのログインを示すエントリを新しいログイン名に変更します。たとえば、ログイン名を *ralph* から *cramden* に変更するには、まず次の行を検索します。

```
ralph:x:103:101:Ralph Cramden:/usr/people/ralph:/bin/csh
```

ログイン名のフィールドとホーム・ディレクトリを次のように変更します。

```
cramden:x:103:101:Ralph Cramden:/usr/people/cramden:/bin/csh
```

整合性を保持するために、ホーム・ディレクトリは常にユーザ・アカウントと同じ名前にする必要があります。shadow パスワードが使用できるシステムでは、暗号化されたパスワードの代わりに x という文字が表示されます。shadow パスワードについての詳細は、『IRIX Admin: Backup, Security, and Accounting』を参照してください。

変更内容を保存し、編集を終了するとき、ファイルが読み専用または書き込み禁止であるというエラー・メッセージが表示される場合があります。これは、IRIX システムが /etc/passwd ファイルに設定した保護です。この保護をオーバーライドするには、vi エディタで次のコマンドを使用します。

:w!

jot エディタを使用している場合は、ファイルは問題なく保存できます。詳細については、vi(1) マン・ページまたは、使用中のエディタのマン・ページを参照してください。

2. **shadow** パスワードが使用可能なシステムでは、次に `/etc/shadow` ファイルを編集します。次のような変更対象のユーザ名で始まる行を検索します。

```
ralph:XmlGDVKQYet5c:::::::::
```

先頭の名前を **cramden** に変更します。

```
cramden:XmlGDVKQYet5c:::::::::
```

変更内容を保存し、編集を終了します。`/etc/passwd` と同じように、`vi(1)` を使用している場合は、エラー・メッセージが表示されることがあります。

3. ユーザのホーム・ディレクトリの1つ上のディレクトリに移動し、次のコマンドを使用して、ホーム・ディレクトリの名前をユーザの新しいログイン名に変更します。

```
mv ralph cramden
```

4. IRIX システムでは、ログイン名ではなくユーザ ID 番号でユーザが所有するファイルを識別するので、ファイルの所有権を変更する必要はありません。

passwd コマンドによるユーザのパスワードの変更

ユーザはパスワードを忘れてしまうことがあります。この場合、まずユーザに一時的なパスワードを割当て、後でユーザが正式なパスワードに変更します。

一時的なパスワードを割当てするには手順 5-11 に従ってください。

手順 5-11 新しいパスワードの割当て

1. **root** でログインします。
2. **passwd** コマンドを使用し、ユーザ・アカウントのパスワードを変更します。
たとえば、ユーザ *ralph* がパスワードを忘れた場合は、次のように入力します。

```
passwd ralph
```

3. 画面のプロンプトに従います。

```
New password: 2themoon
```

```
Re-enter new password: 2themoon
```

メモ: 上記の例ではわかりやすいようにパスワードが表示されていますが、実際には画面上にパスワードが表示されることはありません。

特権ユーザ (*root*) でログインしたときは、古いパスワードの入力は要求されません。

これで、ユーザのパスワードは `2themoon` に変更されました。ユーザはすぐに一時的なパスワードを別の正式なパスワードに変更してください。

ユーザのログイン ID 番号の変更

ユーザのログイン ID 番号を変更することはお勧めできません。この番号は、ファイルやプロセスの所有者情報とセキュリティを保守する上で非常に重要な役割を果たしているからです。ただし、何らかの理由でユーザのログイン ID 番号を変更する必要がある場合は、手順 5-12 に従ってください。

手順 5-12 ユーザのログイン ID 番号の変更

1. システムにあるユーザのホーム・ディレクトリと作業ディレクトリの完全なバックアップ・テープを作成します。
2. `/etc/passwd` ファイルで行の先頭にシャープ符号 (`#`) を置き、パスワード・フィールドにアスタリスク (`*`) を入力して、ユーザのアカウントをロックします。古いユーザ ID 番号を再利用しないようにするため、このエントリは削除しないでください。編集後のエントリは、次のようになります。

```
# ralph:*:103:101:Ralph Cramden:/usr/people/ralph:/bin/csh
```

3. ユーザが所有するホーム・ディレクトリ、サブディレクトリ、および作業ディレクトリを完全に削除します。
4. システムのルート・ディレクトリから次のコマンドを使ってシステムにユーザが所有するファイルが残っていないかを検索します。検索結果は、ファイル名がコンソールに表示されます。

```
find / -user name -print
```

検索されたすべてのファイルをアーカイブした後で削除します。

5. この章で説明する指示に従って、新しいユーザ・アカウントを作成します。このアカウント名は古いアカウントと同じ名前でもかまいませんが、混乱を避けるため、アカウント名も同時に変更することをお勧めします。このとき、新しいユーザ ID 番号を指定します。
6. バックアップ・テープからホーム・ディレクトリ、作業ディレクトリ、その他のファイルをリストアします。必要に応じて、`chown` および `chgrp` を使用して、新しいユーザ ID を正しく設定します。

ユーザのデフォルト・グループの変更

ユーザはシステム上で複数の異なるグループのメンバーになることができます。ただし、そのうち1つがデフォルトのグループになります。デフォルトのグループとは、ユーザがログインしたときに最初に所属するグループです。

ログイン時のデフォルトのグループを変更する場合は、単に `/etc/passwd` ファイルの適切な行を編集すると変更できます。次に例を示します。

```
cramden:+:103:101:Ralph Cramden:/usr/people/cramden:/bin/csh
```

デフォルトのグループを 101 から 105 に変更するには、上記エントリのフィールドを次のように変更します。

```
cramden:+:103:105:Ralph Cramden:/usr/people/cramden:/bin/csh
```

ただし、このような変更を行う前に、105 が有効なグループとして `/etc/groups` ファイルに登録されているか、またユーザ `cramden` がそのグループのメンバーであることを確認する必要があります。グループの作成についての詳細は、116 ページの「シェル・コマンドによるグループの追加」を参照してください。

ユーザのコメント・フィールドの変更

`/etc/passwd` の 5 番目のエントリには、ユーザ・アカウントに関するコメントを入力するためのフィールドがあります。通常、このフィールドには、ユーザ名や電話番号や席の位置を入力します。

この情報を変更するには、単に `/etc/passwd` ファイルの該当フィールドを変更します。ただし、コメント内でコロンの (:) は使用できないので注意してください。これは、IRIX システムがコ

ロンをコメント・フィールドの終わりとして解釈するためです。次にコメント・フィールドの変更例を示します。

```
cramden:x:103:101:Ralph Crumdin:/usr/people/cramden:/bin/csh
```

この例では、名前の綴りが間違っているため、ユーザ Ralph が間もなく管理者に訂正を依頼することになります。その場合は、その行を次のように訂正します。

```
cramden:x:103:101:Ralph Cramden:/usr/people/cramden:/bin/csh
```

ユーザのデフォルト・ホーム・ディレクトリの変更

`/etc/passwd` の 6 番目のフィールドは、ユーザのホーム・ディレクトリを指定します。ホーム・ディレクトリは、ログイン時のディレクトリであり、シェル変数 `$HOME` として設定されています。ユーザのホーム・ディレクトリを変更する手順は簡単です。手順 5-13 に従ってください。

手順 5-13 ユーザのデフォルト・ホーム・ディレクトリの変更

1. `root` でログインします。
2. `/etc/passwd` ファイルを開き、変更対象のユーザのエントリを検索します。

```
cramden:x:103:101:Ralph Cramden:/usr/people/cramden:/bin/csh
```

上記の例では、ホーム・ディレクトリは `/usr/people/cramden` です。このホーム・ディレクトリを `disk2` という最近追加されたファイルシステムに変更する場合は、上記のエントリを次のように変更します。

```
cramden:x:103:101:Ralph Cramden:/disk2/cramden:/bin/csh
```

変更内容を保存し、編集を終了します。

3. 新しいファイルシステム内にディレクトリを作成し、ユーザのすべてのファイルやサブディレクトリを新しい場所に移動します。この作業が終了した後、これまでのホーム・ディレクトリを削除し、変更作業を終了します。
4. ユーザのホーム・ディレクトリを変更する場合は、その旨をユーザに通知しておく必要があります。ほとんどのユーザには、ホーム・ディレクトリの場所に基づく個人的なエリアやプログラムがあります。ユーザに迷惑をかけることがあるので、システム全体の大規模な変更を行う場合と同じように、ホーム・ディレクトリの変更もむやみに行わないでください。

シェル変数についての詳細は、使用しているシェル (`csh`, `bsh`, `tcsh`, `ksh`) のマン・ページを参照してください。通常、ホーム・ディレクトリは `/usr/people` に置かれますが、別のディ

レクトリを選択しても特に問題はありません。/usr ファイルシステムの領域を節約するため、またはユーザが別のディレクトリ名を好むという理由から、管理者は /usr/people 以外のディレクトリを選択することもできます。

ユーザのデフォルト・シェルの変更

デフォルト・シェルを変更するには手順 5-14 に従ってください。

手順 5-14 ユーザのデフォルト・シェルの変更

1. root でログインします。
2. /etc/passwd ファイルを編集し、ユーザのデフォルト・シェルを示すフィールドを変更します。たとえば、passwd に次のようなエントリがあるとします。

```
ralph:x:103:101:Ralph Cramden:/usr/people/ralph:/bin/csh
```

デフォルト・シェルは /bin/csh です。このユーザのシェルを /bin/ksh に変更する場合、上記の行を次のように編集します。

```
ralph:x:103:101:Ralph Cramden:/usr/bin/ralph:/bin/ksh
```

3. 変更内容を保存し、編集を終了します。新しいデフォルト・シェルは、ユーザが次回以降ログインするときに使用されます。

デフォルト・シェルには、任意の実行可能プログラムを指定できます。IRIX システムでは、ユーザがログインしたとき、指定したプログラムがそのまま実行されます。ただし、コマンド・シェル (csh、sh など) 以外のプログラムを指定すると、ユーザに不都合が生じる場合があるので注意してください。passwd ファイル内でデフォルト・シェルとして識別されるプログラムが終了すると、ユーザはログアウトされます。たとえば、/bin/mail をデフォルト・シェルとして指定した場合、ユーザは mail を終了するとログアウトされるので、ほかの作業を実行できなくなります。

ユーザ環境

ユーザの環境は、特定のシェル・スタートアップ・ファイルによって決定されます。C シェルを使用している場合は、これらのファイルは /etc/cshrc とホーム・ディレクトリ内の .cshrc と .login です。Korn シェルまたは Bourne シェルを使用している場合は、ホーム・ディレクトリ内の /etc/profile と .profile です。

シェル・スタートアップ・ファイルではユーザのログイン環境を設定し、ログイン・セッション中に生成されるサブシェルの特性を制御します。

ログイン・シェル

IRIX システムでは、次のログイン・シェルを利用できます。

`/bin/csh` C シェルには、コマンド履歴機能（入力されたコマンドを記録する機能）、ジョブ制御機能（プロセスを停止してバックグラウンドに置き、再びフォアグラウンドに戻す機能）、ワイルドカード・キャラクタを使用して複数のファイルを指定する機能などがあります。ユーザはCプログラミング言語に似た構文を使用して高度なコマンドやスクリプトを作成できます。C シェルの詳細については、`csh(1)` マン・ページを参照してください。

`/bin/sh`, `/bin/ksh` これまでの `/bin/sh` シェルは、Korn シェルの `/sbin/sh` へのシンボリック・リンクになりました。Korn シェルは Bourne シェルと C シェルの長所を拡張したシェルであり、コマンド行編集、ジョブ制御、シェル・プロンプトからのシェル言語によるプログラミングなどの機能を備えています。このシェルの詳細については、`sh(1)` マン・ページを参照してください。特に、Bourne シェルと Korn シェルとの互換性については、`sh(1)` マン・ページの「Compatibility Issues」を参照してください。Bourne シェルについての詳細は、下記の `/bin/bsh` の説明を参照してください。

`/bin/bsh` Bourne シェルは `csh` よりも単純なシェルです。Bourne シェルには、コマンド履歴機能は組込まれておらず、構文も `csh` とは異なっています。また、Bourne シェルはワイルドカード・キャラクタを使用し、`csh` より比較的小規模で速く起動できます。Bourne シェルについての詳細は、`bsh(1)` マン・ページを参照してください。

`/bin/rsh` このシェルは、ユーザが入力できるコマンドに制限のあるシェルです。`rsh` のコマンド構文は `bsh` と同じですが、次の機能は実行できません。

- ディレクトリの変更
- `ls(1)` コマンドの使用
- シェルの検索パス (`$PATH`) の設定

- / を含むパス名やコマンド名の指定
- > や >> を使ったコマンドのリダイレクト

/bin/rsh に対する制限は、.profile が実行された後に適用されます。

これらのシェルの詳細については、ksh(1)、csh(1) および bsh(1) マン・ページを参照してください。rsh については、ksh(1) マン・ページに記載されています。

メモ: IRIX には、rsh という名前のコマンドが 2 種類あります。/usr/lib/rsh は制限されたシェルであり、もう 1 つの /usr/bsd/rsh はバークレイ版のリモート・シェルです。これら 2 種類のコマンドを混同しないように注意してください。

次に、これらのシェルの設定を行う各種スタートアップ・ファイルについて説明します。

C シェルの設定ファイル

C シェルのユーザがシステムにログインすると、次の順序で 3 種類のスタートアップ・ファイルが読み込まれて実行されます。

1. /etc/cshrc ファイルは、システムを使用するすべてのユーザに適用される環境変数を設定するコマンドやシェル・プロシージャを含む ASCII テキスト・ファイルです。このファイルは、login プロセスによって読み込まれ、実行されます。

次に、/etc/cshrc ファイルの例を示します。

```
# default settings for all users
# loaded <<before>> $HOME/.cshrc
umask 022
if ($?prompt) cat /etc/motd
set mail=$MAIL
if ( { /bin/mail -e } ) then
    echo 'You have mail.'
endif
```

この例では、次のコマンドが実行されます。

- プロンプトが設定されている場合は、その日のメッセージを表示します。
- ユーザのメール・ファイルの場所を設定します。

- メールが到着している場合は、その旨を通知するメッセージを表示します。
2. 個々のユーザの `.cshrc` ファイルは `/etc/cshrc` と似ていますが、ユーザのホーム・ディレクトリに保管されています。`.cshrc` ファイルには、ユーザの環境を詳細にカスタマイズするコマンドや変数があります。たとえば、このファイルを使用すると、ユーザ独自のシェル・プロンプトを設定できます。`.cshrc` ファイルは、ユーザがサブシェルを生成するときに必ず実行されます。次に、`.cshrc` ファイルの例を示します。

```
set prompt = "Get back to work: "
set filec
set history = 20
```

この例では、ユーザ独自のプロンプトを設定し、ファイル名の自動完結機能をオンにし、`history` で表示されるコマンド履歴の数を 20 に指定しています。

3. `.login` ファイルは、ユーザのホーム・ディレクトリにある実行可能コマンド・ファイルです。`.login` ファイルを使用してもユーザの環境をカスタマイズできますが、このファイルはログイン時に一度しか実行されません。したがって、このファイルは、ログイン・セッションごとに一度だけ行う環境変数の設定やシェル・スクリプト・プログラムの実行に使用します。次に、`.login` ファイルの例を示します。

```
eval `tset -s -Q`
umask 022
stty line 1 erase '^H' kill '^U' intr '^C' echoe
setenv DISPLAY wheeler:0
setenv SHELL csh
setenv VISUAL /usr/bin/vi
setenv EDITOR /usr/bin/emacs
setenv ROOT /
set path = (. ~/bin /usr/bsd /bin /usr/bin /usr/sbin \ /usr/bin/X11
/usr/demos /usr/local/bin)
```

上記の例では、`tset` を使用してユーザ端末の特性を詳細に設定した後、ファイル作成マスクを `022` に設定しています。次に、`stty` コマンドを使用して便利なキー・バインドを定義し、ユーザのデフォルト表示を `wheeler` というシステムのコンソール画面に設定しています。また、通常使用するユーティリティやファイルシステムの参照基準点に関するいくつかの重要な環境変数を設定しています。最後に、デフォルトのパスを拡張して、ユーザ独自のバイナリ・ディレクトリとその他のシステム・ディレクトリを追加しています。

上記の例で使用されているシェル・プログラミングのコマンドについての詳細は、`csh(1)` マニュアルページを参照してください。

Bourne シェルと Korn シェルの設定ファイル

Bourne シェルまたは Korn シェルのユーザがシステムにログインすると、次の順序で 2 種類のスタートアップ・ファイルが読み込まれ、実行されます。

1. /etc/profile ファイル

このファイルは、システムのすべてのユーザに適用される環境変数を設定するコマンドやシェル・プロシージャを含む ASCII テキスト・ファイルです。このファイルは、login プロセスによって読み込まれ、実行されます。

次に、/etc/profile ファイルの例を示します。

```
# Ignore keyboard interrupts.
trap "" 2 3
# Set the umask so that newly created files and directories will be
# readable by others, but writable only by the user.
umask 022
case "$0" in
*su )
# Special processing for 'su -' could go here.
;;
-* )
# This is a first time login.
#
# Allow the user to break the Message-Of-The-Day only.
trap "trap '' 2" 2
cat -s /etc/motd
trap "" 2
# Check for mail.
if /bin/mail -e
then
echo "you have mail"
fi
;;
esac
trap 2 3
```

この例では、次のコマンドが実行されます。

- キーボード割込みを有効にします。
- ユーザの *umask* を 022 に設定します。所有者については完全なパーミッションが与えられ、同じグループのメンバーとシステムを使用する一般ユーザについては読み込み権と実行権が与えられます。

- ユーザが最初にログインしたときに、その日のメッセージ (*/etc/motd*) を表示します。メールが到着しているときは、その旨をユーザに通知します。

2. 個々のユーザの .profile

このファイルは */etc/profile* と似ていますが、ユーザのホーム・ディレクトリに保持されています。 *.profile* ファイルには、ユーザの環境を詳細にカスタマイズするコマンドや変数があります。このファイルは、ユーザがサブシェルを生成するときに必ず実行されます。

次に、 *.profile* ファイルの例を示します。

```
# Set the interrupt character to Ctrl+C and do clean backspacing.
stty intr '' echoe
# Set the TERM environment variable
eval `tset -s -Q`
# List files in columns if standard out is a terminal.
ls() { if [ -t ]; then /bin/ls -C $*; else /bin/ls $*; fi }
PATH=/bin:/usr/bin:/usr/sbin:/usr/bsd:$HOME/bin:.
EDITOR=/usr/bin/vi
PS1="IRIX> "
export EDITOR PATH PS1
```

この例では、次のコマンドが実行されます。

- 割込み文字を `<Ctrl>-C` に設定します。
- `tset` を使用して `TERM` 環境変数を設定します。
- `ls` 関数を定義します。この関数は、ユーザがディレクトリ内のファイルをリスト表示するために、端末またはウィンドウから `ls` コマンドを入力すると、`-c` オプション付きで `ls` コマンドを実行します。
- 環境変数の `PATH` と `EDITOR` を設定します。
- ユーザのプロンプト (`PS1`) を `IRIX>` にします。

上記の例で使用されているシェル・プログラミングのコマンドについての詳細は、`ksh(1)` および `bsh(1)` マン・ページを参照してください。

シェル環境変数の設定

すべてのシェルでは変数が使用されます。変数にはシェルに関する情報や、シェルが起動されたログイン・アカウントに関する情報が含まれています。これらの変数は、シェルおよびほかのプロセスに情報を提供します。

シェルの環境は、これらの変数の集合によって設定されています。厳密には変数の設定や操作方法はそれぞれ異なりますが、環境変数および環境の基本概念はどのタイプの IRIX シェルについても共通です。

基本的な環境変数としては、IRIX ファイルシステム内でコマンドを検索する場所を指定する PATH、ユーザ・アカウントのホーム・ディレクトリを示す HOME、現在の作業ディレクトリを示す PWD、ユーザの表示画面や端末との通信に使用する *terminfo* データベースの名前を示す TERM などがあります。

プロセス（シェル）が起動すると、環境に関する情報が exec システム・コールによってプロセスに渡されます。ほかのプロセスはこれらの情報を使用します。たとえば、ユーザ *trixie* の端末が *iris-ansi* (TERM=*iris-ansi*) として定義されています。ユーザがデフォルトのビジュアル・エディタ *vi* を起動すると、*vi* は環境変数 TERM を調べて *iris-ansi* 端末の特性を参照します。

シェル環境の表示

login も 1 つのプロセスなので、環境文字列を使用できます。現在のシェル環境を参照するには、`printenv` コマンドを使用します。例 5-2 に C シェルの一般的な環境を示します。

例 5-2 C シェル環境

```
LOGNAME=trixie
PWD=/usr/people/trixie
HOME=/usr/people/trixie
PATH=./usr/people/trixie/bin:/usr/bsd:/bin:/etc:/usr/sbin:
/usr/bin: /usr/local/bin:
SHELL=/bin/csh
MAIL=/var/mail/trixie
TERM=iris-ansi
PAGER=more
```

```
TZ=EST5EDT
EDITOR=emacs
DISPLAY=myhost:0
VISUAL=vi
```

デフォルト環境変数

C シェル・ユーザの場合、これらの変数はスタートアップ・ファイルの `/etc/cshrc`、`.cshrc` または `.login` で設定します。Korn シェルまたは Bourne シェルを使用している場合は、`/etc/profile` または `.profile` で設定します。

次に、スタートアップ・ファイルでほかの変数が設定されていない場合の C シェル・ユーザに割当てられるデフォルトの環境変数を示します。

- HOME
- PATH
- LOGNAME
- SHELL
- MAIL
- TZ
- USER
- TERM

新しい環境変数の定義

`setenv` コマンド (C シェルのみ) を使用すると、いつでも新しい環境変数を定義したり既存の変数を変更することができます。たとえば、変数 `PAGER` の値を変更する場合は、C シェルで次のように入力します。

```
setenv PAGER pg
```

このコマンドによって、環境変数 `PAGER` の値が `pg` コマンドに設定されます。`PAGER` 変数は `mail` によって使用されます。

Korn シェルまたは Bourne シェルのもとの環境変数を設定する場合は次のようにします。

```
$ PAGER=pg ; export PAGER
```

環境変数は、コマンド行で設定してもシェル・スタートアップ・ファイル (/etc/profile または \$HOME/.profile) で設定してもかまいません。

IRIX のプロンプトの変更

IRIX システムで表示されるプロンプトは変更することができます。たとえば、マシン名や現在の作業ディレクトリの名前などをシェル・プロンプトで表示することができます。これは使用するログイン・シェルに依存します。

C シェルと tcsh で、マシン名と作業ディレクトリの名前をプロンプトで表示する場合には次の一連のコマンドを入力します。

```
set prompt="'hostname -s':'echo $cwd'% "  
alias cd 'cd \!*;set prompt="'hostname -s':'echo $cwd'% "'
```

Korn シェルの場合には、次のコマンドを入力します。

```
PS1='hostname''::${PWD}% '  
export PS1
```

デフォルトのファイル・パーミッション (umask)

ユーザが作成するファイルやディレクトリに対するアクセス権は、umask と呼ばれるシステム・デフォルトによって制御されます。IRIX システムのシステム・デフォルトは 022 で、umask が設定されていません。この値は次のパーミッションを設定しています。

ユーザ 完全なアクセス権（読み込み権、書き込み権、適用可能な場合は実行権）。ディレクトリは実行可能です。つまり、ユーザは自分の所有する任意のディレクトリに移動 (cd) して、そこからファイルをコピーすることができます。

グループ グループのメンバーは、グループ内のメンバーが所有するファイルの読み込みと実行（適用可能な場合）が行えます。ディレクトリに対しても実行可能です。ファイルやディレクトリへの書き込みはできません。

一般ユーザ システムを使用しているほかの一般ユーザはグループと同じアクセス権を持ちます。

022 というシステム・デフォルトの `umask` は、作成したファイルに対して `chmod 644` を実行したり、ディレクトリと実行可能ファイルに対して `chmod 755` を実行するのと同じです。 `umask` を設定しても既存のファイルやディレクトリには影響がありません。 `chmod` と同じように、 `umask` でもファイルのパーミッションの設定に 3 桁の引数を使用します。ただし、 `umask` の引数は `chmod` の引数とは反対の意味を持ちます。つまり、 `umask` の引数はアクセス権の最大値 (ファイルの場合 666、ディレクトリと実行可能ファイルの場合 777) を引下げます。

umask によるデフォルトのファイル・パーミッションの変更

デフォルトのファイル・パーミッションを変更するには、 `umask` シェル・コマンドを使用します。

次のコマンドは、ファイルやディレクトリの作成時に、ユーザ、グループ、一般ユーザに対するパーミッションの変更は行いません。

```
umask 000
```

次のコマンドは、ほかの一般ユーザのアクセス権を 1 だけ減らします (実行権を取消します)。

```
umask 001
```

次のコマンドは、グループのアクセス権を 1 だけ減らし (実行権を取消します)、ほかの一般ユーザのアクセス権を 2 だけ減らします (実行権はありますが書込み権は取消されます)。

```
umask 012
```

次のコマンドは、グループから書込み権と実行権を取消し、ほかの一般ユーザからはすべてのパーミッションを取消します。

```
umask 037
```

詳細については、 `umask(1)` マン・ページを参照してください。

特別なログイン・シェル

次の理由がある場合は、システム・デフォルト以外のログイン・シェルをアカウントに割当てることができる。

- システムに限定した特別のアクセスをするための専用アカウントが必要な場合
- ユーザが特別なシェルを必要とする場合

アカウント用のログイン・シェルには任意のプログラムを指定できます。たとえば、サード・パーティ製のアプリケーションをログイン・シェルとして使用できます。そのようなアプリケーションをログイン・シェルとして使用するユーザは、ログインすると同時にそのアプリケーションの環境下に置かれます。システムとのすべての対話はこのアプリケーションを通じて行われ、アプリケーションを終了すると、ユーザは自動的にログアウトされます。システムへのアクセスを制限するには、管理者が作成したカスタム・シェルを使用する方法もあります。

別の例として、`/usr/lib/uucp/uucico` をログイン・シェルとして使用する `nuucp` アカウントがあります。

ユーザはそれぞれ好みのシェル (`bash` シェルなど) のインストールを管理者に依頼する場合があります。そのような場合は、ほかのソフトウェアをインストールする場合と同じように、インストールするシェルが信頼のおけるソースからのものであることを確認しなければなりません (`bash` シェルはパブリック・ドメイン・ソフトウェアです)。シェルをインストールする前に必ずシステムの完全なバックアップを作成し、しばらくの間そのシェルに問題がないかどうかシステムを監視する必要があります。

セキュリティ上の理由から、コンパイル済みのバイナリ・プログラムを安易に受取り、それをログイン・シェルとしてシステムにインストールしないでください。まず、シェルのソース・コードを調べてセキュリティ上の問題がないことを確認した上で、ソースをコンパイルし、インストールしてください。

特別なログイン・シェルは、ユーザがシステムにログインする前に必ずマウントされるファイルシステムに格納する必要があります。ログイン・シェルを含むファイルシステムがマウントされていないと、そのシェルを使用するユーザは自分のアカウントにログインできません。

メッセージの送信

IRIX システムでは、電子メール、ユーザ通知メッセージ、リモート・ログイン・メッセージ、news、write、wall など、さまざまな手段でユーザと通信できます。

電子メール・プログラム

ユーザは、IRIX システムで提供されている電子メール・プログラムを使用して相互にメッセージを送受信することができます。コマンド行形式や GUI 形式の各種メール・アプリケーションが用意されています。電子メールの設定方法についての詳細は、『IRIX Admin: Networking and Mail』を参照してください。

その日のメッセージ (motd) 機能

すべてのユーザに関連する事項を、`/etc/motd` ファイルを使用して通知することができます。`/etc/motd` の内容は、ログイン処理の一部としてユーザの端末に表示されます。ログイン処理では、次のようなコマンドを含む `/etc/cshrc` (C シェルの場合) が実行されます。

```
cat /etc/motd
```

各ユーザがログインするたびに、`/etc/motd` 内のテキストが表示されます。ユーザがその情報に興味を持てるように、メッセージの内容には常に注意し、期限切れの通知などは削除しなければなりません。

次に、このメッセージの例を示します。

```
5/30: The system will be unavailable from 6-11 pm Thursday, 5/30,  
while we install additional hardware
```

不要になったメッセージは必ず削除してください。

`/etc/motd` ファイルには、その日のメッセージが含まれています。このメッセージは、ユーザが最初にシステムにログインしたとき、**Bourne** シェルの場合は `/etc/profile` によって、**C** シェルの場合は `/etc/cshrc` によってユーザの画面に表示されます。

管理者は、システム運用に関するメッセージを `motd` ファイルに格納しておくことができます。次の例のように、定期的な保守作業、新しいハードウェアやソフトウェアの追加など、ユーザに影響するシステムやサイトの設定に変更があるときユーザに警告します。

ユーザは、ログインするたびにこのメッセージを見るので頻繁に内容を変更してユーザの注意を引くようにしてください。同じメッセージが繰返し表示されると、ユーザは興味をなくし、重要な通知メッセージを見落とす可能性があります。

古くなった通知メッセージは削除してください。特に新しいメッセージがない場合には、`Welcome to the system` など短いメッセージにしておきます。

例 5-3 に典型的な `motd` ファイルの例を示します。

例 5-3 Welcome-to-the-System メッセージ

```
Upcoming Events: -----
```

```
26 November -- The system will be down from 8pm until Midnight for a  
software upgrade. We are installing FareSaver+, Release 3.2.2d.
```

```
Watch this space for further details.
```

```
28 November through 31 November -- We will be operating with a  
minimal staff during the holiday. Please be patient if you need  
computer services. Use the admin beeper (767-3465) if there is a  
serious problem.
```

`motd` ファイルは、ワークステーションよりもサーバで頻繁に使用されますが、ゲスト・アカウントで使用しているネットワーク上のワークステーションに対してメッセージを送る場合には便利な方法です。システムを使用しているすべてのユーザに電子メールを送信することもできますが、この方法はより大きなディスク領域を必要とし、ユーザがメールボックス内のメールを読落とす可能性もあります。

リモート・ログイン・メッセージ

シリアル回線やネットワークを介してシステムにログインするユーザにとって、`/etc/issue` ファイルは `/etc/motd` ファイルに相当します。`/etc/issue` ファイルが存在しない場合、または空である場合、メッセージは表示されません。

ユーザがシリアル回線やネットワークを介してシステムにログインするとログイン・プロンプトが表示される前に、`/etc/issue` メッセージが表示されます。`/etc/issue` がない場合、または空の場合は、メッセージは表示されません。これは、ログイン・プロンプトの前に表示されるという点で、`/etc/motd` とは本質的に異なります。このメッセージは、ユーザにシステム使用に関する規則を通知するときに利用されます。たとえば、ワークステーションやサーバが特定の企業の従業員のために確保されており、侵入者が違法に使用すれば罰せられる、という内容のメッセージなどです。

news コマンドによるメッセージの送信

`/usr/news` ディレクトリと `news` コマンドを使用すると、簡単な電子掲示板機能を設定できます。この `news` 機能によって、管理者はシステムに関連する重要なメッセージを通知できます。このシステムは、公共サービスの Usenet システムとは異なります。システムに関連する記事は、`/usr/news` ディレクトリに格納します。各記事に1つのファイルを作成し、各ファイルにはその内容を示す名前 (`downtime`、`new-network` など) を付けます。各記事は `news` コマンドを使用して表示します。

`news` コマンドは、`/etc/cshrc` ファイルなどのシェル・スタートアップ・ファイルから自動的に起動できます。ログイン時にニュースを読む時間がないユーザは、新しい記事がないかどうかをシェル・スタートアップ・ファイルで確認できます。次に例を示します。

```
news -s
```

`-s` の引数を指定すると、最後にニュースを読んだ後に着信した記事の数が報告されます。

`news` コマンドを使用してニュースを読む場合は、次の方法を選択できます。

すべての記事を読む

最後にニュースを読んだ後に着信したすべての記事を参照する場合は、`news` コマンドに引数を付けずに実行します。

news

記事を選択して読む

記事を選択して読む場合は、記事の名前を引数で指定して `news` コマンドを実行します。

news downtime new-network

記事を拾い読みする

`news` コマンドを実行した後で **<Ctrl>+C** キーまたは **<Break>** キーを押すと、記事の表示を中止して次の記事を表示できます。**<Ctrl>+C** キーまたは **<Break>** キーを2回続けて押すと、プログラムを停止できます。

すべての記事を無視する

忙しくて記事を読む時間がない場合は、それらの記事を後で読むことができます。記事は管理者 (`root`) が削除しないかぎり `/usr/news` に残っているので、ユーザがログインするたびに、ニュース記事のリストが表示されます。

すべての記事を一掃する

すべての記事を一掃し、ニュースを最新の状態にするには2通りの方法があります。

touch .news_time

このコマンドによって、`.news_time` ファイルのアクセス時間と変更時間のフィールドが更新されるので、`news` プログラムはユーザに提示する最新記事がないと判断します。

また、次のコマンドは現在の全記事を表示しますが、画面出力を `/dev/null` に送っているため、画面には何も表示されません。

news > /dev/null

上記のような方法によって、記事を読まなくても更新することができます。

write コマンドによるメッセージの送信

システムを使用しているほかのユーザの画面にメッセージを書込むには、`write` コマンドを使用します。

```
write ralph
```

ユーザ *ralph* の画面に次のようなメッセージが表示されます。

```
root からのメッセージです brooklyn (console) [ (火) 2月 26日 16時47分47秒 ] ...
```

送信側は、*ralph* が応答するのを待つか、メッセージの入力を始めることができます。相手が応答すると、上記と同じメッセージが画面に表示されます。

メッセージを入力して **<Enter>** キーを押すと、メッセージの各行が相手の画面に表示されます。

通常、*write* のセッションでは、各ユーザが交互に順番にメッセージを送受信します。次の例に示すように、各メッセージの最後の行には終了を示す記号を入力します。

```
I noticed that you are using over 50 meg of disk space. Is there anything I can do to help you reduce that?
>
```

最後の不等号の記号は、メッセージを終了し、ユーザ *ralph* の応答を待っていることを示します。相手方も同じ目的でほかの記号を使用する必要があります。

ほかのユーザからの *write* を拒否するには、*mesg* コマンドを使用して端末やウィンドウを書込み禁止にします。

mesg n

n という引数を指定すると端末やウィンドウは書込み禁止となり、*y* を指定すると書込み可能になります。*mesg n* を使用して端末やウィンドウを書込み禁止にしている場合でも、特権ユーザはすべてのユーザに対してメッセージを書込むことができます。

talk は *write* と似たユーティリティであり、一部のユーザは多用しています。詳細については、*talk(1)* マン・ページと *write(1)* マン・ページを参照してください。

wall コマンドによるメッセージの送信

特権ユーザは *wall* コマンドを使用して、システムにログインしているすべてのユーザに対してメッセージを書込むことができます。この機能は、システムの停止をすべてのユーザに通知する場合などに便利です。

wall を使用するには、まず次のように入力します。

wall

メッセージを入力します。入力を完了して **<Ctrl>+D** キーを押すと、メッセージが送信されます。

メッセージを含む `messagefile` というファイルを作成した後に、`wall` を使用してそのメッセージを送信することもできます。

wall < messagefile

`wall` コマンドは、ユーザによる `mesg` の設定には影響されません。つまり、ユーザは `wall` によって画面に表示されるメッセージを抑制できません。複数のウィンドウを持つグラフィックス・ディスプレイでは、すべてのウィンドウにメッセージが表示されます。

ディスクおよびスワップ領域の設定

この章では、管理者が使用できるツールとオプションに関するバックグラウンド情報とヒントについて説明します。ユーザは、ディスク領域をどれだけ使用しているかを意識せずに、ファイルを蓄積することがあります。これは、ワークステーションのユーザにとっては大した問題ではありませんが、複数のユーザがシステム・リソースを共有するサーバにとっては重要な問題です。IRIX システムでは、ディスク使用率を管理するためのユーティリティが多数提供されています。

また管理者は、システムがスワップ領域としてディスク領域を使用できるように割当てする必要があります。スワップ領域とは、オペレーティング・システム・カーネルが一時的なメモリとして使用できるように割当てられたディスク領域のことです。スワップ領域が不足すると、システムで同時に動作するアプリケーションの数とサイズが制限され、システムのパフォーマンス低下の原因となります。

ディスク割当ての設定と保守についての詳細は、『IRIX Admin: Disks and Filesystems』を参照してください。

ディスク使用量に関するコマンド

ここで説明するコマンドを使用すると、システムのディスク使用量の現在の状態を判断できます。

du (ディスク使用量) コマンド

du コマンドは、システムの任意の場所にあるディレクトリやファイルによって使用されているディスク領域を表示します。ディレクトリやファイルのサイズは 512 バイトのブロック単位、または **-k** フラグを指定したときは 1 K バイトのブロック単位で表示されます。詳細については、du (1M) マン・ページを参照してください。

df (フリー・ディスク・ブロック) コマンド

df コマンドは、ワークステーションの各ファイルシステムのディスクの空き領域を表示します。この情報は、すべてのファイルシステム（ローカルなファイルシステムまたは NFS がマウントされたファイルシステム）について報告します。空き領域は 512 バイトのブロック単位、または `-k` フラグを指定したときは 1 K バイトのブロック単位で表示されます。詳細については、df (1) マン・ページを参照してください。

quot (各ユーザのディスク使用量) コマンド

quot コマンドは、指定されたファイルシステムの各ユーザが使用するディスクのキロバイト数を表示します。このコマンドの出力内容を各ユーザにメールで送信し、ディスクの使用率を通知できます。詳細については、quot (1M) マン・ページを参照してください。

diskusg (ディスク・アカウンティング) コマンド

diskusg コマンドは、システム・アカウンティング・パッケージの一部です。quot と同じように、このコマンドはシステムを使用する各ユーザのディスク使用状況を報告します。ただし、通常 diskusg は独立したコマンドとしてではなく、dodisk を使用したシステム・アカウンティング・パッケージの一部として使用されます。diskusg についての詳細は、diskusg (1M) マン・ページを参照してください。

ディスク領域の管理

ここでは、ディスク領域を管理するためのさまざまなコマンドについて説明します。特に、使用可能なディスク領域を最大化できる各種手法について説明します。

ファイルの圧縮とアーカイブ

ディスク使用量を最小限にする方法の 1 つは、ファイルの圧縮やアーカイブをユーザに奨励することです。あまり使用しないが破棄できないようなファイルは、圧縮するのが効果的です。compress ユーティリティを使用すると、ディスク領域を節約できます。ファイルを圧縮すると、システムにファイルを残したまま、そのサイズを 50 % 程度小さくできます。圧縮されたファ

イルの拡張子は **.z** となり、名前は変更できません。圧縮とアーカイブを組合わせて使用するとディスク領域を最大限に節約できます。

ほとんど使用しないファイルについては、テープやフロッピーなどのアーカイブ媒体にコピーすることをお勧めします。後でそれらのファイルが必要になった場合はいつでも復元できますが、その手順は圧縮した場合よりも多少複雑になります。

quotas サブシステムによるディスク領域の管理

quotas サブシステムは、ディスク使用に関する問題の解決を支援します。このサブシステムを使用して、システムの各ユーザが使用できるディスク最大使用量を割当てることができます。このサブシステムについての詳細は、quotas(4) マン・ページを参照してください。

通常、ディスク使用に関する方針を決め、必要に応じてディスク割当ての検討や実施を行うのはサイト管理者です。サイト管理者はディスク使用に関する明確なガイドラインを作成し、たびたび制限を超えるようなユーザに対しては、注意を促す必要があります。また、/tmp などのテンポラリ・ディレクトリの使用、および *guest* や *uucp* のようなゲスト・アカウントの使用に制限を設けることもお勧めします。ルート (/) ファイルシステムの容量を 100% 使い果たするような状況になると、システムが停止し、ユーザに迷惑がかかります。

ディスクの割当てではできるだけ柔軟に行ってください。通常の作業を行っていても一時的に使用割当て制限を超えてしまう場合がよくありますが、常に制限を超えているのではないかぎり特に問題はありません。

いかなる状況でも、適切な警告を通知せずにユーザのファイルを削除しないでください。

システムの各ユーザがホーム・アカウントで使用できるディスク領域を 20 MB (1 ブロック = 512 バイトで約 40,000 ブロック) に制限されているときに、ユーザ *norton* が制限の 2 倍以上のディスクを使用しているとします。この問題を解決するには手順 6-1 に従ってください。

手順 6-1 ディスク割当て実施のガイドライン

1. ユーザに直接事情を聞き、大量のディスク領域が必要となる理由を確認します。ディスク使用に関する方針を再検討し、使用可能なディスク領域を増やす必要がある場合もあります。

ユーザが単に古いファイルを放置している場合は、テープにファイルをバックアップし、システムから削除するように伝えます。たとえば、不要になった電子メール・メッセージをメールボックスに大量に残している場合です。ほとんど使用しないファイルは、テープに保存すると整理がつき、ディスク領域が節約できます。

2. ユーザから直接事情を聞けない場合は、そのユーザに電子メールを送ります。

ディスクの使用状況を `diskusg` によって自動的にチェックし、制限を超えたユーザにメールを送信するようなスクリプトがある場合では、ある期間が経過するとユーザがこれらのメッセージに慣れてしまい、そのメッセージを無視する可能性があるので注意してください。したがって、特定のユーザ宛に、ディスク使用状況について検討する必要があるという個人的なメッセージを送信する方が効果的です。

3. ユーザは、データがシステム内のほかの場所（サイト内のアクセス可能なワークステーション）で使用可能であることを認識していないために、サイトに共通のツールやデータ・ファイルを個人的にコピーしている場合があります。管理者は、ユーザと協力してこのような重複を解消してください。

4. ユーザが所属を異動したことをサイト管理者に通知しない場合があるので、ユーザが実際にシステムを使用しているかどうかを確認する必要があります。

また、特定のワークステーション上のアカウントを使用しなくなったユーザが、ファイルを削除しないで放置している場合があります。このような可能性があるときは、ユーザが最後にシステムにログインした時間を `finger(1)` コマンドで確認します。

finger norton

`finger` コマンドは、ユーザが最後にシステムにログインした日付と時刻やほかの情報を示します。この情報は、`/etc/wtmp` ファイルがある場合は、そこから読取られます。

5. ユーザのファイルを削除する必要がある場合は、テープにそれらのファイルをバックアップしてから削除します。このような処置は十分に注意して行ってください。ユーザのファイルを不用意に削除すると、作業に支障をきたすだけでなくユーザの信頼を失うことになります。ファイルをテープにコピーすることをユーザに十分通知した上で、ファイルを削除してください。

念のため、テープのコピーは2つ作成し、1つはユーザに渡します。コピーしたテープが読取れることを確認してから、ファイルをシステムから削除します。

ネットワーク・ファイル・システム (NFS: Network File System) によるディスク領域の管理

オプションのネットワーク・ファイル・システム (NFS: Network File System) ソフトウェアをシステムにインストールしている場合は、通常使用するディレクトリをエクスポートすることによって、ワークステーション上のディスク領域を節約できます。エクスポートされたディレクトリは、NFS ソフトウェアを実行するすべてのシステムで使用可能になります。たとえば、ネットワーク内に 10 台のワークステーションがあり、各ワークステーションに 5 MB のオンライン・マン・ページとリリース・ノートが格納されている場合、`/usr/man` と `/usr/catman` の各ディレクトリをエクスポートすると、45 MB のディスク領域を節約できます。ほかのすべてのワークステーションは、これらのファイルを削除し、サーバからリモートでマウントできます。NFS のマウントでは、クライアント・ワークステーションのディスク領域をとらないので、そのディスク領域はほかの用途に使用できます。

もう 1 つの方法は、別のシステムから空きディスク領域をリモート・マウントすることです。この方法は、複数のシステム間でディスク使用率が大きく異なる場合に効果的です。たとえば、あるワークステーションでは使用可能なディスク領域の 25 % しか使用していないのに対し、ほかのワークステーションでは 90 % 以上を使用している場合は、ファイルシステムをリモート・マウントすることによって、ディスクの使用率を平均化できます。

このように NFS を活用すると、ネットワーク上で大量のディスク領域を節約できます。ただし、この方法では、再起動や修理を行うためにサーバがオフラインになっている間、そのサーバからマウントされたファイルやプログラムにはアクセスできなくなるので注意してください。

ディスク・パーティションによるディスク領域の管理

ディスク使用量の割当てを実施するための極端な方法は、各ユーザ・アカウントごとにディスク・パーティションとファイルシステムを設定し、そのファイルシステムをユーザのホーム・ディレクトリとして空のディレクトリ上にマウントすることです。この方法では、ユーザがディスク領域を使い果たすと、新しいファイルの作成や既存ファイルの拡張ができません。ただし、`/tmp`、`/var/tmp` の各ディレクトリに余裕がある場合は、これらのディレクトリにファイルを書込むことができます。ユーザが割当ての制限を超えてファイルの作成や書込みを行おうとすると、デバイス上にディスクの空き領域がないというエラー・メッセージが表示されます。

このディスク管理方法は、管理者側に多くの準備と保守作業が必要になり、一度実施すると簡単に変更できないので、お勧めできません。また、ディスクを多くの小さなパーティションに分割

すると使用可能なディスク領域の総容量が減少し、システムにパフォーマンス上のオーバーヘッドがかかります。これは、ディスク・コントローラが、ディスク上で次に使用できる場所ではなく1つの小さなパーティション内にユーザのファイルを書込む必要があるからです。

このようなディスク管理方法は、システムの負荷が非常に高い場合、およびユーザが指示に従わず、ほかの方法では解決できない場合にのみ使用します。たとえば、`/usr` パーティションが長期間 100% 使用され、ディスク領域不足のために毎日のようにシステム・サービスを停止しなければならず、またディスク領域をこれ以上増やせないという場合は、この方法を考慮します。

無駄なディスク領域

システムに無駄なディスク領域が大量にある場合があります。ディスクの空き領域が不足している場合は、大きなファイルがテンポラリ・ディレクトリや管理用ディレクトリの中に蓄積されていないことを確認してください。

`/var/adm` ディレクトリ構造は、大きなログ・ファイル、またはデバッグ用の大きなファイルを蓄積してしまいます。これらのファイルは役に立ちますが、多くのディスク領域を使用します。`/var/adm/crash` ディレクトリは、以前システムに障害が起きたときに生成された IRIX カーネルのイメージ、またはコア・メモリ `vmcore` 全体のコピーをいくつか格納しています。これらのファイルは、メガバイト単位の大容量を占めるので、テープに記録してシステム内から削除します。

システム監査が有効な場合は、非常に大きな監査追跡記録ファイルが作成されるので、定期的（1日に1回など）にテープに記録しなければなりません。

また、使用している `UUCP` や `ftp` のディレクトリをチェックし、`core` ファイルなど大きくて不要なファイルがシステム上にないことを確認する必要があります。

スワップ領域

IRIX オペレーティング・システムは、ディスクの一部をスワップ領域として使用します。スワップ領域は、実行プログラムをすべて読込むには、物理メモリの容量が不十分な場合や、プログラムの一部または全部を一時的に保存するために使用されます。非常に大きなプログラムを多数実

行するときは、スワップ領域も使い果たしてしまう場合もあります。ページングとスワッピング機構についての詳細は、216 ページの「ページングとスワッピング」を参照してください。

IRIX システムでは、一時的にロードされるのは各プログラムの一部のみで、プログラムはシステムの限界を超える領域を専有して実行できます。このため、IRIX システムは将来の用途に基づいてスワップ領域を事前に割り当てる必要がありますが、この予測は外れる可能性があります。スワップ領域が実際に必要になったとき、IRIX システムは割り当てられた実領域ではなく、最も便利な使用可能領域を割り当てます。したがって、物理的な割り当てはアカウントिंग割り当てと区別されません。

システムがすべてのスワップ領域を前もって割り当てている場合、スワップ領域は使用されていないにもかかわらず、使い果たされたかのように表示されることがあります。これは単に、既存のプロセスが将来使用するスワップ領域に対する権限をシステムに前もって割り当てたため、IRIX システムの制限されたスワップ領域のアカウントिंगにより新しいプロセスを領域に割り当てることができなかったのが理由です。

厳密なスワップ領域アカウントINGは常に有効ですが、通常システム・ファイルにより物理的スワップ領域と仮想スワップ領域の両方を追加できるので、管理者はディスク・パーティションの再設定やシステムの再構成、および再起動も行わずに、スワップ領域を追加したり、厳密なスワップ領域アカウントINGを無効にできます。

ページングとスワップ領域の監視

ページングとスワップ領域の使用状況を監視するには、`sar -p`、`sar -w`、`sar -q`、`swap -s`、および `swap -l` を使用します。スワップ領域を使い果たしたときは、2つの解決策があります。メモリを増設するか、スワップ領域を増分します。スワップ領域を増分しても、大きなプログラムのパフォーマンスの改善にはなりません、実行は可能になります。

`sar(1)` コマンドでは、プロセス全体のスワッピングは `-w` フラグで通知されます。パフォーマンスの問題は、ページングに関する入出力が過剰であることや遅いことが原因です。

仮想スワップ領域の追加

十分な物理的領域があると考えられるときに、スワップ領域不足が原因で、システムがプロセスのスタック拡張や新しいプロセスを受付けられない状態の場合には、`/etc/fstab` ファイルに次の行を追加します。

```
/usr/swap swap swap pri=4,vlength=204800 0 0
```

続けて、次のコマンドを実行します。

```
mkfile -v 0b /usr/swap
```

ファイル (`/usr/swap`) はゼロ長になるので、仮想スワップ領域だけが追加され、実際のスワップ領域は追加されません。これで、カーネルが実行できるプロセスは増加します。ただし、システムの限界を超えてアクセスしようとする、IRIX システムは一番大きな実行プログラムをメモリからスワップします。

`swap -l` コマンドによるスワップ領域のリスト

ワークステーションですでに設定済みのスワップ領域の容量を確認するには、`swap (1M)` コマンドを使用します。

```
swap -l
```

プログラムを頻繁にスワップアウトするアプリケーションが実行されているとき、オペレーティング・システムによって使用されるディスク・スワップ領域を微調整する方法もあります。この方法の詳細については、216 ページの「ページングとスワッピング」を参照してください。

`swap -s` コマンドによるスワップの動作のチェック

`swap -s` コマンドは、スワップ領域を追加する必要があるかどうかを確定するための便利なツールです。次に、`swap -s` コマンドの出力内容を示します。

```
total: 0 allocated + 64248 reserved = 64248 blocks used, 17400 blocks available
```

以下のようなフィールドが表示されます。詳細については、`swap (1M)` マン・ページを参照してください。

Allocated	プライベート・ページ（使用中のデータが格納されるページなど）に割当てられている 512 バイト・ブロックの数。
Reserved	現在割当てられているが、プライベート・ページとしてはマークされていない（領域は確保されているが使用されていない） 512 バイト・ブロックの数。
Blocks used	割当てられているか確保されている 512 バイト・ブロックの数（上の 2 つの総数）。
Blocks available	将来、割当てるか確保できる 512 バイト・ブロックの数（ <code>swap -l</code> コマンドで表示されるスワップの総ブロック数から使用済みのブロック数を引いた値）。

次に `swap -s` コマンドの出力例を示します。

```
total: 0 allocated + 34200 reserved = 34200 blocks used, 47448 blocks available
```

上記の出力例は、使用中のスワップは 0 ブロック、34200 ブロックが確保されているが未使用であるために、47448 ブロックが確保可能であることを示しています。したがって、この時点で上記のシステムではスワッピングは行われませんが、このシステムで動作するプログラムによって、実行時のメモリ拡張のために約 17 MB のスワップ領域が要求されたことがわかります。

メモ：10000 ブロックは約 5 メガバイトに相当します。

多くのアプリケーションは、仮想スワップ領域と呼ばれる空間を確保します。つまり、実際に占有するよりも多くのメモリを要求します。アプリケーションの実際のサイズは、アプリケーションが使用する物理的なシステム・リソースの量です。アプリケーションの仮想サイズは、アプリケーションが使用している物理的なシステム・リソースの量に、要求されたが使用されていない追加リソースの量を追加した値になります。上記の出力例でも、確保されたが使用されていない領域が示されています。

負のスワップ領域

`swap -s` の別の出力例を示します。

```
total: 41920 allocated + 58736 reserved = 100656 blocks used, -19400 blocks available
```

使用可能なスワップ領域が負の値のように見えますが、この出力内容は割当てられた、または使用中のページの一部がメイン・メモリ (RAM) 内にあることを意味します。ここでの問題は、`swap -s` の出力ではメイン・メモリが考慮されないことです。負の値で表示されるデータは、実際にはシステム・メモリに格納されたデータです。

割当て済みの領域量 (41920 allocated) からわかるように、約 20 MB の物理スワップ領域が使用中です。したがって、システムの物理スワップ領域は不足していません。物理スワップ領域が不足している場合は、割当て済みのブロック数が `swap -l` コマンドで報告されるブロック数と同じか非常に近くなります。約 30 MB の物理スワップ領域が余分に確保されているので、全部で約 50 MB のスワップ領域が使用できます。したがって、10 MB の超過が生じたように見えます。

負のスワップ領域を解釈するもう 1 つの方法は、負の値はスワップ領域量から使用されるブロック数 (割当て済み数 + 確保済み数) を引いた値と考えることです。したがって、この値の絶対値が物理メモリ容量 (`hinv` コマンドで得られる値) よりも大きい場合以外は、システムのメモリは不足しません。

次の例に、スワップ領域の限界に到達しやすいシステムでの `swap -s` コマンドの出力例を示します。

```
total: 76920 allocated + 23736 reserved = 100656 blocks used, -19400 blocks available
```

合計のブロック数は同じですが、割当てられたブロック数がより多くなっていることに注意してください。この例で、`swap -l` によってシステムの物理スワップ領域が 81000 ブロックと報告された場合は、使用されていない物理ブロックが 4000 ブロックしかないことがすぐにわかります。

仮想スワッピング

`swap -s` によって負の値が報告されるときに、システムが物理的限界に近づいていない場合は、仮想スワップ領域を増やしてください。したがって、実際に必要な量よりも多くの領域を確保するアプリケーションに対し、メモリ領域を割当てることができます。次のコマンドを実行すると、仮想スワッピングを使用可能にします。

```
su
chkconfig vswap on
/etc/init.d/swap start
```

上記のコマンドによって、スワップ領域のうち確保できるが割当てられていない領域が使用可能になります。詳細については、`/etc/init.d/swap` ファイルと `swap(1M)` マン・ページを参照してください。

`chkconfig` によって仮想スワッピングがすでにオンの場合、または割当てられたブロック数が `swap -l` コマンドで報告されるブロック数に近い場合は、物理メモリまたはスワップ領域を追加する方法を使用してください。別のディスク・パーティションまたはスワップ・ファイルを使用してスワップ領域を追加する方法については、`swap(1M)` マン・ページを参照してください。

単一ディスク・システムでのスワップ領域の拡張

マルチディスク・システムがない場合を考えてみます。ここでは、単一ディスクでのスワップ領域のサイズの拡張方法を説明します。この章の始めで述べたように、ディスク・パーティションを再区分することによって使用可能なスワップ領域を拡張するか、ここで説明する `swap` コマンドを使用して領域を追加できます。

`swap` コマンドは、いずれかのディスク領域の一部を追加スワップ領域として指定できます。スワップ領域の追加は常に可能で、新しいスワップ領域が不要になった場合は削除できます。このコマンドで使用可能なオプションがいくつかあります。詳細については、`swap(1M)` マン・ページで説明します。最も便利な使用 방법은通常システム・ファイルを追加スワップ領域として指定する方法です。

ファイルを追加スワップ領域として指定するには、まず `mkfile(1M)` コマンドを使用し、適切なサイズの空ファイルを作成します。たとえば、システムに 10 MB のスワップ領域を追加するとき、その領域を `/usr` ファイルシステムから取得する場合は、次の `mkfile` コマンドを使用します。

```
mkfile -v 10m /var/tmp/moreswap
```

`mkfile` コマンドでは、オプション `-v` を指定すると、ファイルが作成されたことを確認するメッセージが表示されます。

```
/var/tmp/moreswap 10485760 bytes
```

オプション `-v` を指定しないと、`mkfile` はメッセージを表示せずに動作します。`mkfile` コマンドの第 2 フィールドはファイルのサイズです。この場合、`10m` はサイズ 10 MB のファイルを指定します。`b`、`k`、`m` を `size` 引数の後に指定すると、サイズの数値の単位をそれぞれバイト、

キロバイトまたはメガバイトで表すことができます。たとえば、次のコマンドはいずれも 10 MB のファイルを作成します。

```
mkfile -v 10485760b /var/tmp/moreswap
mkfile -v 10240k /var/tmp/moreswap
mkfile -v 10m /var/tmp/moreswap
```

ファイルを作成した後、`swap` コマンドを使用し、そのファイルのスワップ領域としてシステムに追加できます。ファイルを作成する場合は、そのファイルが同じファイルシステム内に常駐するようにします。通常、`/usr` には `root (/)` ファイルシステムよりも使用可能な領域があるので、`/var/tmp` ディレクトリを使用する方法が適切です。ただし、NFS でリモート・マウントされたファイルシステムも使用できます。リモート・マウントされたファイルシステムをスワップ領域に使用する方法についての詳細は、`swap(1M)` マン・ページを参照してください。

新しいファイルのスワップ領域として使用するには、まず次のコマンドを入力します。

```
/sbin/swap -a /var/tmp/moreswap
```

オプション `-a` は指定されたファイルがスワップ領域として即座に追加されることを示します。新しいスワップ領域をチェックするには、次のコマンドを使用します。

```
swap -l
```

このコマンドは、現在のすべてのスワップ領域とその状態をリストします。

新しいスワップ・ファイルに永続性を持たせる（起動時に自動的に追加される）には、`/etc/fstab` ファイルに次の行を追加します。

```
/var/tmp/moreswap    swap    swap    pri=3  0  0
```

`root` ファイルシステムの `/tmp` ディレクトリにスワップ・ファイルを作成すると、そのファイルはシステムの起動時に削除されます。`/var` ファイルシステムの `/var/tmp` ディレクトリは起動時に削除されないため、スワップ・ファイルの位置として選択する場合は、このディレクトリの使用をお勧めします。`root` ファイルシステムにスワップ領域を作成する場合は、まず `/swap` ディレクトリを作成した後、そのディレクトリ内にスワップ・ファイルを作成します。

マルチディスク・システムでのスワップ領域の拡張

マルチディスク・システムにさらにスワップ領域を追加する方法は、単一ディスク・システムに領域を追加する方法と同じです。mkfile コマンドと swap コマンドを使用すると、いつでもシステムにスワップ・ファイルを追加できます。ただし、新しいディスク・パーティションに専用スワップ領域を追加するには手順 6-2 に従ってください。

手順 6-2 専用スワップ領域の追加

1. デフォルトのスワップ領域を 2 倍にするには、別のディスク・ドライブを次のように使用できます。

```
Partition/slice
0          Temporary space (mount as /tmp)
1          Swap space
6          usr2
```

オペレーティング・システムはスワップ領域として使用されるパーティションに書き込み続けるので、そのパーティションにデータが格納されていると、データは完全に破壊されてしまいます。スワップ・パーティションがユーザ・ファイルシステムのどのパーティションともオーバーラップしていないことを確認してください。スワップ・パーティションのサイズはブロック単位で検証します。

2. パーティションを選択した後、このパーティションにスワップ・パーティションとして永続的に追加する `/etc/init.d/addswap` ファイルを作成します。次の行をファイルに追加してください。

```
swap -a /dev/dsk/devicename 0 length
```

引数 `devicename` はスワップ・パーティションの所在位置となるデバイス名 (`ips0d1s1` など) で、`length` はブロック数です。

3. `chmod` コマンドを使用して、ファイルに実行許可を設定します。

```
chmod +x addswap
```

4. その後、次のコマンドを使用し、新しいファイルへのシンボリック・リンクを作成します。

```
ln -s /etc/init.d/addswap /etc/rc2.d/S59addswap
```

`/etc/rc2.d` ディレクトリは、システムがマルチユーザ・モード (実行レベル 2) で起動するときのシステム・アクティビティを制御します。作成したシンボリック・リンク・ファイルの先頭にある `S` は、システムがこの実行レベルを開始すると、ファイル内のコマンドが実

行されることを示しています。文字 **K** で始まるシンボリック・リンク・ファイルは、ファイルに記述されているコマンドが終了されることを示します。リンク・ファイル名の先頭の **S** または **K** に続く番号は、コマンドが実行される順序を示します。

管理者は、`/etc/fstab` ファイルを変更し、選択したパーティションがスワップ・パーティションとして使用されることをコメントの形で記述することもできます。

ユーザ・プロセスの管理

この章では、ユーザ・プロセスを管理するための以下のタスクについて説明します。

- 157 ページの「ユーザ・プロセスの監視」
- 160 ページの「nice によるプロセスの優先順位付け」
- 162 ページの「プロセスの終了」
- 164 ページの「Cpuset の定義と管理」
- 165 ページの「Checkpoint and Restart」、165 ページの「NQE」、165 ページの「Share II」、および 166 ページの「Performance Co-Pilot」

ファイル数が多すぎると使用可能なディスク領域が不足するように、一度に実行させるプロセスの数が多すぎると CPU 時間を使い果たしてしまいます。この状態になるとシステムの応答時間が遅くなり、最終的にはプロセスがまったく実行されなくなります。多くのプロセスを実行できるようにカーネルを調整していない場合、新規プロセスは飽和状態になるかなり前の段階で受付られなくなります。ただし、通常の方法でシステムを使用している場合、システム使用率の変動により、可能な最大数のプロセスを実行する前にシステム・パフォーマンスが不安定になる場合があります。

ユーザ・プロセスの監視

すべてのプロセスが同量のシステム・リソースを必要とするわけではありません。大きなファイルを扱うデータベース・アプリケーションのようなプロセスは、ディスク間での大量の読み書きと大量のディスク領域を必要とするため、ディスク集約的 (disk intensive) な処理になる傾向があります。このようなプロセスは、CPU 時間を大幅に消費し、要求された操作をハードウェアが実行する際にも時間を消費します。プログラムのコンパイルや大量のデータ処理などのプロセスは、多くの CPU 命令を実行する必要があるため CPU 集約的 (CPU intensive) になります。また、大量のデータを読み込んで、それをメモリ内で操作するプロセスはメモリ集約的 (memory intensive) になります。ディスク、CPU、メモリなどのリソースは有限です。そのため、リソー

スを大量に消費する複数のプロセスを同時に実行すると、システム・パフォーマンスが低下する場合があります。

システム管理者は、システム使用率の大まかな傾向を監視して問題が発生した際に対応し、できるかぎり効率的にシステムを運用してください。実行中のプロセスの数が少ないにもかかわらず、システムがオーバーロードであることを示す場合は、すでに大幅に容量を使用しているか、または容量を超過している可能性があります。次にシステム上のプロセスを監視する5つの方法を説明します。

top によるプロセス監視

IRIX システムが提供する `top` と `gr_top` は、システムで CPU 時間を最も多く消費するプロセスを監視するための便利なユーティリティです。これらのユーティリティは、CPU 時間を最も多く消費しているプロセスを動的に表示します。表示されたプロセスが終了すると、次に CPU 時間の消費量が多いプロセスが表示されます。`gr_top` を使用すると `top` と同じ情報がグラフィカルに表示されます。非グラフィック・サーバを使用している場合は、`gr_top` を各自のマシンで実行できませんが、同じネットワーク内でグラフィックス機能を持つ別のシステムの画面に表示できます。`top` と `gr_top` の設定と使用方法についての詳細は、`top(1)` と `gr_top(1)` マン・ページを参照してください。ディスプレイのリセットについては、18 ページの「リモート・ワークステーションでのウィンドウの表示」を参照してください。

Array のプロセス管理

Array Services の製品をご使用の際、`array` でのプロセス管理については、『Getting Started With Array Systems』および `array(1)` マン・ページを参照してください。

osview によるプロセス監視

`osview` と `gr_osview` を使用すると、カーネルの実行統計を動的に表示できます。グラフィックス・ワークステーションでは、`gr_osview(1)` を使用してシステム・メモリと CPU の使用状態をリアルタイムでグラフィカルに表示できます。`osview` を使用すると、同じ情報を ASCII 形式で表示できます。現在のシステムの状態に関するさまざまな情報を表示するよう `gr_osview` を設定することもできます。デフォルトでは、`gr_osview` を実行するとユーザ・プロセスの実行に消費された CPU 時間、システム・オーバーヘッド・タスク、割込み、およびアイドル時間

に関する情報が表示されます。osview と gr_osview の設定と使用方法についての詳細は、osview(1) および gr_osview(1) マン・ページを参照してください。

sar によるプロセス監視

sar (システム・アクティビティ・レポート) は osview と本質的に同じ情報を提供しますが、システムの動的状態ではなくそのスナップショットを表示します。sar は 1 枚のスナップショットを作成するので、それをファイルに保存し、別の時間に収集した同じスナップショットと容易に比較できます。システム管理者は、cron を使用して sar を自動的に使用することにより、システムの一連のスナップショットを何度も撮ることができます。このため、軽い負荷の場合や重い負荷の場合、さまざまな種類の負荷 (CPU 負荷、ネットワーク負荷、ディスク負荷など) の場合のシステム・パフォーマンスの基本線を抑えると、システムのボトルネックの位置を突きとめることができます。sar についての詳細は、『IRIX Admin: Backup, Security, and Accounting』および sar(1) マン・ページを参照してください。sar によるシステム・アクティビティの監視についての詳細は、207 ページの「timex、sar、par」を参照してください。

ps によるプロセスの監視

ps -ef コマンドを使用すると、システムで現在実行されているすべてのプロセスを参照できます。表 7-1 に、ps -ef の出力例を示します。

表 7-1 ps -ef コマンドの出力例

Name	PID	PPID	C	Time	TTY	CPU Time	Process
joe	23328	316	1	May 5	ttyq1	1:01	cs

上記の表では、joe というユーザのプロセスが表示されています。実際のリストには、システムで動作中のプロセスを所有するすべてのユーザが表示されます。リストの各フィールドには、次の有効な情報が含まれています。

Name プロセスを所有するユーザのログイン名。
PID プロセス識別番号。

PPID	表示されたプロセスを起動またはフォークした親プロセスのプロセス識別番号。
C	プロセス実行に関する現在の優先順位。この番号が大きいほど、スケジューリング上の優先順位が低くなります。この番号はプロセスに関する最新のスケジューリングに基づくもので、プロセスの絶対的な優先順位は表していません。
Time	プロセスの実行開始時刻。ps コマンドの実行時にプロセスがすでに 24 時間以上実行されている場合は、プロセスの開始日が表示されます。
TTY	プロセスと関連する TTY (端末またはウィンドウ)。
CPU	該当するプロセスがそれまで消費した CPU 時間の合計。このフィールドは、最大の CPU 時間を使用しているプロセスを確定する上で便利です。短時間に多くの CPU 時間を使用するプロセスは、システムの慢性的な遅延の原因になります。

各プロセスの一般的な優先順位などの詳細情報を調べる場合は、`-l` フラグを指定し `ps` を実行します。ps の出力リストについての詳細は、`ps(1)` マン・ページを参照してください。

プロセスの優先順位

IRIX システムには、CPU 集約的なプロセスを一般的なユーザ・プロセスよりも低い優先順位で実行するための方法が用意されています。`nice(1)` コマンドと `npri(1)` コマンドを使用すると、ユーザは独自にプロセスの優先順位を制御できます。

nice によるプロセスの優先順位付け

`nice` コマンドは、次のように入力します。

```
nice [ -increment ] command
```

`/bin/nice` を使用してコマンド行を作成する場合は、*increment* フィールドに 1 ~ 19 の番号を指定します。番号を指定しない場合は、デフォルト値として 10 が指定されます。*increment* に大きい番号を指定するほど、プロセスの優先順位は低くなります。優先順位が最も低いのは 19 です。19 より大きい番号を指定しても 19 として認識されます。`cs(1)` シェルには、`nice` コマ

ンドとは異なる機能を持つ独自の `nice` 機能が組込まれています。この機能の詳細については、`csch(1)` マン・ページを参照してください。

コマンド行から `nice` コマンドと優先順位を入力し、通常どおりにそのコマンドを入力してください。たとえば、`joe` というユーザが、前述の `ps -ef` で説明したように時間のかかるコンパイル・コマンドを実行すると想定し、その優先順位を最も低くするには、次のコマンドを入力してください。

```
nice -19 cc -o prog prog.c
```

`nice` を使用してプロセスを起動した場合は、そのプロセスの実行に必要な CPU 時間の総計は変更しませんが、プロセスの実行頻度が低くなるので、実行にかかる時間は長くなります。

特権ユーザ (`root`) にかぎり、`nice` にマイナスの値を与えて *increase* プロセスの優先順位を上げることができます。`nice` にマイナスの値を与えるには、次の例に示すように、優先順位の前に2つのマイナス符号を付けます。

```
nice --19 cc -o prog prog.c
```

上記のコマンドは、ユーザ・プロセスが保持できる最高の優先順位をプロセスに与えます。1つのプロセスの優先順位を上げると、ほかのユーザにとって重大なシステムの遅延を招く可能性があるため、特権ユーザはこの機能を頻繁に使用しないでください。`/bin/csh` には、ここで説明したものと少し異なる構文を使用する `nice` プログラムが組込まれています。`csch` についての詳細は、`csch(1)` マン・ページを参照してください。

npri によるプロセスの優先順位付け

`npri` コマンドを使用すると、ユーザが所有するプロセスに対して、絶対的な優先順位を割り当てることができます。通常の操作では、プロセスは実行時間が経過するとともにその優先度を失うため、大きなジョブに割り当てられる単位時間あたりの CPU サイクル (タイム・スライス) は時間の経過とともに低減します。ただし、最低の優先順位は保証されます。このような優先度の低減は、単純なタスクのパフォーマンスを保持するために行われます。`npri` を使用すると、ユーザはプロセスの `nice` 値 (絶対優先順位) を設定し、そのプロセスの許容優先順位を確保できます。また、CPU によってプロセスに割り当てられるデフォルトのタイム・スライスを設定することもできます。さらに、`npri` を使用すると、現在実行中のプロセスの優先順位を変更できます。次の例では、`npri` に適用できるすべてのフラグを使用します。

```
npri -h 10 -n 10 -t 3 cc -o prog prog.c
```

この例では、`-h` フラグによってプロセスの絶対優先順位を設定し、`-n` フラグによって絶対的な `nice` 値を設定しています。また、`-t` フラグは、CPU によってプロセスに割り当てられるタイム・スライスを指定します。IRIX システムのデフォルトのタイム・スライスは 10 ミリ秒に設定されているので、この例ではタイム・スライスを 30 ミリ秒に変更します。`npri` コマンドとそのフラグについての詳細は、`npri(1)` マン・ページを参照してください。

実行中プロセスの優先順位の変更

特権ユーザは、`renice(1M)` コマンドまたは `npri` コマンドを使用して、実行中のプロセスの優先順位を変更できます。これらのコマンドを使用できるのは特権ユーザにかぎられます。次に `renice` の使用例を示します。

```
renice -n increment pid [-u user] [-g pgrp]
```

通常、`renice` は、システム時間を大幅な割合で消費している特定のプロセスの優先順位を下げるために使用しますが、特定のユーザに関連するすべてのプロセスや (`-u` フラグを使用)、特定のプロセス・グループに関連する全プロセス (`-g` フラグを使用) の優先順位を下げることもできます。このコマンドにはほかにも多くのオプションがあります。これらのオプションについての詳細は、`renice(1M)` マン・ページを参照してください。

`npri` コマンドを使用して、実行中のプロセスに関する優先順位などのパラメータを変更することもできます。次に、`npri` を使用して実行中のプロセスのパラメータを変更する方法を示します。

```
npri -h 10 -n 10 -t 3 -p 11962
```

特権ユーザは、`renice` または `npri` を使用してプロセスまたはユーザの優先順位を上げることもできますが、これはシステムのパフォーマンスに大きな影響を与える場合があります。

プロセスの終了

プロセスの実行では、メモリやディスク、CPU 時間を大幅に消費するためシステムがクラッシュする前にプロセスを終了する必要があります。プロセスを強制終了する場合は、そのプロセスを起動したユーザが再度同じプロセスを起動しないようにします。この場合、ユーザに通知してか

らプロセスを終了し、プロセスを途中で終了させたこととその理由を少なくともユーザに通知してください。これで、ユーザは優先順位を低く設定して同じプロセスを再起動したり、システムのジョブ処理機能 (at、batch、cron) を使用して別の時間にプロセスを実行できます。

kill コマンドによるプロセスの終了

プロセスを強制終了するには、kill コマンドを使用します。通常の終了の場合、kill -15 というコマンド形式を使用します。-15 フラグは、オープンされたファイルや記述子をクローズするための猶予時間を与えてプロセスを安全に終了させることを意味します。-9 フラグは、クリーンアップのための猶予時間を与えずにプロセスを即座に終了させます。終了するプロセスに実行中の子プロセスがある場合は、kill -9 コマンドを使用すると子プロセスはプロセス・テーブルに残ります。ただし、この場合入力はできません。wait (1) コマンドを実行してプロセス番号を指定するとプロセス・テーブルから子プロセスを削除できます。kill コマンドの構文と使用方法についての詳細は、kill (1) マン・ページを参照してください。kill コマンドを使用してプロセスを強制終了する場合は、そのプロセスの PID を常に認識する必要があります。

killall コマンドによるプロセスの強制終了

killall コマンドを使用すると、指定したコマンド名を持つプロセスを強制終了できます。たとえば、ユーザが独自に起動した a.out というプログラムを強制終了する場合は、次の構文を使用します。

```
killall a.out
```

このコマンドを使用すると、ps コマンドによるプロセス ID 番号の検索など、時間がかかる作業を行わずにプロセスを終了できます。

メモ：このコマンドはシェルで動作しているプログラムのうち、指定された名前のすべてのプロセスを強制終了します。また、引数なしで起動した場合は、このコマンドを実行したユーザが権限を持つすべてのプロセスを強制終了します。このコマンドが通常のユーザによって実行された場合、対象となるのは該当するユーザが起動したプロセスまたはその子プロセスだけですが、root によって実行された場合は、システムのすべてのプロセスを強制終了します。したがって、このコマンドは十分注意して使用してください。killall についての詳細は、killall (1M) マン・ページを参照してください。

Cpuset の定義と管理

cpuset コマンドは、*cpuset* という CPU セットの定義と管理に使用します。cpuset は、制限付きまたは制限なしで定義できる名前付きの CPU セットです。このコマンドでは、cpuset の作成または破棄したり、既存の cpuset に関する情報を取得したり、1 つのプロセスとそのすべての子プロセスを cpuset に割り振ることができます。また、指定した cpuset に関連付けられているプロセスや CPU の数など、特定の cpuset のプロパティを参照することもできます。

メモ: cpuset コマンドは *miser_cpuset* に取って代わるもので、*miser_cpuset* と互換性もあるため、Miser バッチ処理システムは使用する必要がありません。Miser の用法については、『*IRIX Admin: Resource Administration*』の第4章「Cpuset System」を参照してください。

アクセス制限つき cpuset では、その CPU で実行できるプロセスは cpuset のメンバーのみです。オープンな cpuset では、その CPU ですべてのプロセスを実行できますが、cpuset のメンバーであるプロセスは、その cpuset に属する CPU でしか実行できません。

cpuset は、cpuset 設定ファイルと cpuset 名で定義します。設定ファイルの形式の定義については、cpuset (4) マン・ページを参照してください。cpuset 設定ファイルには、cpuset に属する CPU を記述します。cpuset の定義に必要な引数もこのファイルに記述します。cpuset 名は 3 ~ 8 文字で指定します。2 文字以下の名前は予約されています。

cpuset へのアクセスは、設定ファイルのファイル・パーミッションで定義します。パーミッションを確認する場合は、ファイルの現在のパーミッションを使用します。したがって、特定の cpuset へのアクセスを変更するには、cpuset 自体を破棄して作成し直す必要はなく、そのアクセス・パーミッションだけを変更します。読み込みパーミッションでは、cpuset に関する情報を取得できません。実行パーミッションでは、cpuset にプロセスを付加することができます。

例 7-1 は、3 つの CPU を指定した専用 cpuset の設定ファイルです。

例 7-1 Cpuset 設定ファイル

```
# cpuset configuration file
EXCLUSIVE

CPU 1
CPU 5
```

CPU 10

この仕様では、3つのCPUを含むcpusetが作成され、このcpusetに明示的に割当てられたスレッドだけが各CPUで実行されます。

メモ：Miser で使用中のCPUとcpusetに割当てられたCPU間で競合が発生することがあります。専用に設定されたCPUにはMiserからアクセスできません。

cpusetのコマンド情報および詳細については、cpuset(1)マン・ページを参照してください。

Checkpoint and Restart

CPR (IRIX Checkpoint and Restart) は、実行中のプロセスを保存し、後でチェックポイントが発生した箇所から実行し直すことができる機能です。CPRの使用と管理方法については、『IRIX Checkpoint and Restart Operation Guide』とcpr(1)マン・ページを参照してください

NQE

NQE (Network Queuing Environment) は、バッチ・スケジュール機能と対話式のロード調整機能を提供するワークロード管理環境です。NQEの設定、監視および制御に関する説明は、『NQE Administration』を参照してください。また、次のURLアドレスも参照してください。

<http://www.cray.com/products/software/nqe>

Share II

Share II は、システム・リソースの割当てをダイレクトに管理し制御するための機能を提供する、リソース中心のFair Shareスケジューラです。Share IIの設定と維持方法については、『Share II for IRIX Administrator's Guide』を参照してください。

Performance Co-Pilot

PCP (Performance Co-Pilot) は、高度なパフォーマンス管理アプリケーションのソフトウェア・パッケージです。これは、分配、統合パフォーマンス監視とパフォーマンス管理サービスの実現に役立つシステム・レベルのツール・セットを提供します。PCP の管理方法については、『Performance Co-Pilot User's and Administrator's Guide』を参照してください。

ファイル変更モニタ (fam: File Alteration Monitor) の使用

ファイル変更モニタ (fam: File Alteration Monitor) は、ファイルとディレクトリを監視するデーモンです。アプリケーション開発者は、自分のアプリケーションに特定の関数呼出しを組込み、指定したファイルやディレクトリに対する変更の通知を fam から受取ることができます。たとえば mailbox(1) は新しいメールの到着を知らせるために fam を使用します。fam デーモンは、このデーモンを使用するアプリケーションが実行されているときにのみ動作するので、プログラムがすべて終了すると、このデーモンも終了します。fam の使用方法についての詳細は、fam(1M) マン・ページと fam(3X) マン・ページを参照してください。

構成ファイル

IRIX 6.5.8 リリース以降では、fam の動作を制御するために、コマンド行オプションのほかに構成ファイルを使用できます。このファイルは、デフォルトで /etc/fam.conf に設定されていますが、-c コマンド行オプションを入力して別のファイルを指定することもできます。

例 8-1 に示したように、構成行は「オプション =f1Av の書式を用います。# または ! で始まる行は無視されます。デーモンは、デフォルト・ファイルに示される insecure_compatibility、untrusted_user、local_only、および xtab_verification の各オプションを認識します。これらのオプションについての詳細は、fam(1M) マン・ページを参照してください。

例 8-1 デフォルトの fam.conf ファイル

```
# fam.conf
#
# 以下の構成オプションについての詳細は、
# fam(1M) マン・ページを参照してください。
```

```
#  
# insecure_compatibility は認証を無効にします。これにより、  
# 各クライアント接続によって提示される UID が受け入れられるため、  
# untrusted_user が無視されるようになります。  
#  
# -C コマンド行の引数は、このオプションを無効にします。  
#  
insecure_compatibility = false  
  
#  
# untrusted_user とは、非認証クライアントに対して使われるユーザ  
# のことです。非認証クライアントは、このユーザが示すことができない  
# ファイルを fam を使って監視することはできません。値にはユーザ名  
# または数値 UID を使用できます。  
#  
untrusted_user = nobody  
#  
#  
# local_only により、fam はリモート・クライアントやリモート fam から  
# の要求を無視します。このオプションは、fam が inetd によって開始された  
# 場合は無視されることに注意してください。  
#  
# コマンド行引数 -L は、このオプションを無効にします。  
#  
local_only = false
```

```
#  
# xtab_verification により、fam は書出されたファイルシステムの  
# リストをチェックして、ホストに書出されたファイルシステムに  
# リモート・ホストからの要求が送られたことを確認します。  
#  
xtab_verification = true
```

fam の基礎的なトラブルシューティング

fam を使ったアプリケーションを起動しようとする、次のようなエラー・メッセージが表示される場合があります。

```
Cannot connect with File Alteration Monitor (fam)
```

このメッセージが表示される原因は複数あります。このセクションでは、この問題を解決する一般的な方法について説明します。

以下の2つのメッセージのどちらかが画面に表示される場合は、手順 8-1 に従ってください。

```
Can't connect to fam
```

```
Cannot connect with File Alteration Monitor (fam)
```

手順 8-1 fam のトラブルシューティング

1. /var/adm/SYSLOG ファイル（システム・ログ）を表示して、問題解決に役立つメッセージがあるかどうかを調べます。sysmon ツールの使用方法については、37 ページの「システム・ログを表示する sysmon」を参照してください。
2. 次の ps コマンドを実行して、fam が動作しているかどうかを調べます。

```
ps -ef | grep fam
```

fam が動作している場合は、fam が inetd によって開始されたことを確認します。fam プロセスに対してリストされた親 PID (PPID) は、次の ps コマンドで返される PID と一致する必要があります。

```
ps -ef | grep inetd
```

fam が動作していなければ、必ずしも問題があるとはかぎりません。通常、fam デーモンはアプリケーションが必要とする場合に inetd によって起動し、1 つまたは複数のクライアント・アプリケーションによって使用されている間しか動作しません。

3. fam が /etc/inetd.conf ファイルのリストにあることを確認します。

- a. 次の grep コマンドを実行します。

```
grep fam /etc/inetd.conf
```

- b. 次のようなコマンド出力を探します。

```
sgi_fam/1-2 stream rpc/tcp wait root ?/usr/etc/fam fam
```

fam デーモンは /user/etc/fam です。

- c. 上記の行がない場合はファイルに追加し、次の killall コマンドを実行して、inetd によってファイルを再び読取ります。

```
killall -HUP inetd
```

4. fam デーモンがファイルに含まれ、root による実行可能ファイル(つまり /usr/etc/fam ファイル)であることを確認します。このファイルが見つからない場合は、eoe.sw.base サブシステムをインストールし直す必要があります。

5. fam が /etc/rpc ファイルのリストにあることを確認します。

- a. 次の grep コマンドを実行します。

```
grep fam /etc/rpc
```

- b. 次のようなコマンド出力を探します。

```
sgi_fam          391002
```

6. 次の ps コマンドを実行して、inetd が動作していることを確認します。

```
ps -ef | grep inetd
```

- a. 次のような応答を探します。

```
root    214      1  0   Oct 24 ?          0:01 /usr/etc/inetd
```

- b. 次のメッセージが表示される場合、ネットワークがアクティブでないことを意味します。

```
portmapper failure
```

ネットワーク管理については、『*IRIX Admin: Networking and Mail*』を参照してください。

7. 次の `rpcinfo` コマンドを実行して、`fam` がポートマップに登録されていることを確認します。

```
rpcinfo -p | grep fam
```

`inetd` は、`/etc/inetd.conf` ファイルを読取ると、見つかった RPC サービス (`fam` など) をポートマップに登録します。これにより、クライアント・アプリケーションは、必要とするサービスの指定ポートに接続できるようになります。

- a. 次のようなコマンド出力を探します。

```
391002  1  tcp  1033  sgi_fam
391002  2  tcp  1033  sgi_fam
```

`rpcinfo` コマンドによって出力が生成されない場合、`fam` はポートマップに登録されていません。前の手順を実行した場合 (`killall -HUP inetd` を含む)、ポートマップが動作していない可能性があります。

8. `inetd` または `fam` が登録されたポートで実際にリスンしていることを確認します。

上記の出力例では、ポートは 1033 です。このポートでリスンしているプロセスを調べるには、`fuser` コマンドを実行します。

```
fuser 1033/tcp
```

- a. 次の出力が生成される場合、プロセス 223 とプロセス 627 がポート 1033 を開いています。

```
1033/tcp:      223o      627o
```

これらのプロセスが `inetd` と `fam` であるかどうかを調べるには、`ps` コマンドを実行します。

- b. 一方、`fuser` コマンドによって次のような出力が生成される場合は、ポートマップの情報が混乱しており、おそらくネットワーク・サービスを再起動する必要があります。

```
1033/tcp: fuser: No such file or directory
```

9. `fam` が接続を受信しない原因がほかに考えられない場合は、`/etc/inetd.conf` ファイルの `fam` エントリに `-d` オプションを追加して、`fam` のデバッグを有効にしてみてください。次のように行を編集する必要があります。

```
sgi_fam/1-2  stream  rpc/tcp  wait  root ?/usr/etc/fam  fam
-d
```

- a. inetd.conf ファイルを編集後、killall コマンドを実行して inetd によってファイルを再び読み取ります。

```
killall -HUP inetd
```

- b. fam がまだ動作している場合 (少なくとも 1 つのアプリケーションで fam が使用されている) は、fam を再起動してデバッグ・フラグを有効にします。

```
killall fam
```

デバッグの情報は /var/adm/SYSLOG ファイルに書込まれます。このファイルは、37 ページの「システム・ログを表示する sysmon」で説明される sysmon ツールを使って表示できます。

メモ: アプリケーションで使用中に fam を終了すると、それらのアプリケーションで問題が発生する場合があります。なるべくアプリケーションを先に終了してください。

メモ: 他社製の NIS マスター・システムを使用している場合は、172 ページの「Sun NIS マスターを使用している場合の fam のトラブルシューティング」を参照してください。

メモ: IRIX 6.5.8 リリースでは、クライアントが身元を証明して認証を受けるまで、fam はクライアントを信頼できないものとして扱います。libfam は静的アーカイブ (libfam は、認証を行うバージョンで置換えることができる共有ライブラリではなく、各クライアントにコンパイルされます) のため、古い libfam にリンクされたクライアントは、新しい fam との通信を試みると正常に動作しない場合があります。クライアント・アプリケーションのトラブルシューティングについての詳細は、fam(1M) マン・ページを参照してください。

Sun NIS マスターを使用している場合の fam のトラブルシューティング

サイトにオプションの NIS (YP) ソフトウェアをインストールし、Sun システムを NIS マスターとして使用している場合で、sgi_toolkitbus と sgi_fam に rpc エントリがないときに表示されるエラー・メッセージの対処方法について説明します。

Sun NIS (YP) サーバのオペレーティング・システム (Sun 3.x または Sun 4.0) によって、次のいずれかの解決方法を適用できます。

- Sun 3.x

Sun ワークステーションで Sun OS 3.x が実行されている場合は、Sun NIS サーバの /etc/rpc データベースに次の 2 つのエントリを追加してください。

- sgi_toolkitbus 391001
- sgi_fam 391002

NIS サーバで次のように入力します。

```
cd /usr/etc/yp; make rpc
```

NIS サーバがクライアントに情報を送信するのに 1 時間程度かかる場合があります。

- Sun 4.0

Sun ワークステーションで Sun OS 4.0 以降のリリースが実行されている場合は、Sun NIS (YP) サーバ・マシンの /etc/rpc データベースに次の 2 つのエントリを追加してください。

- sgi_toolkitbus 391001
- sgi_fam 391002

NIS サーバで次のように入力します。

```
cd /var/yp; make rpc
```

NIS サーバがクライアントに情報を送信するのに 1 時間程度かかる場合があります。

メモ：NIS システムが SGI 社あるいは Sun 社のサーバーでない場合でも、同じ rpc エントリを追加する必要があります。ただし、構文が異なる場合もあります。

コマンド (PROM) モニタの使用

この章では、コマンド (PROM) モニタ・プログラムについて説明します。このプログラムは、SGI のワークステーションやサーバの起動環境を制御します。コマンド・モニタを使うと、制御された条件下で CPU を起動したり、操作したり、CPU をコマンド・モニタ・モードで稼働させたり、プログラム（たとえば、オペレーティング・システム・カーネルの `/unix` や、デバッグおよび実行バージョンの特殊なカーネル）をロードできます。

PROM モニタについての詳細は、`prom(1M)` マン・ページを参照してください。

この章では、次について説明します。

- コマンド・モニタを起動する方法。176 ページの「コマンド (PROM) モニタの起動方法」を参照してください。
- コマンド・モニタで使用できるコマンドの種類。179 ページの「コマンド・モニタのコマンド・リスト」を参照してください。
- コマンド・モニタ実行中にヘルプ情報を入手する方法。181 ページの「コマンド・モニタでのヘルプ情報の入手方法」を参照してください。
- コマンド・モニタの便利な使用方法。181 ページの「コマンド・モニタでのコマンドの使用方法」と「コマンド・モニタの実行方法」（185 ページ）を参照してください。
- コマンド・モニタ環境の操作方法。186 ページの「コマンド・モニタの環境変数」を参照してください。
- コマンド・モニタでプログラムを起動する方法。192 ページの「コマンド・モニタでのプログラムの起動方法」を参照してください。

PROM モニタ

PROM とは、プログラム可能読取り専用メモリ (Programmable Read-Only Memory) を意味します。PROM チップには、CPU の起動、システムの管理作業、ソフトウェアのインストールなどを実施するソフトウェアが通常組込まれており、コンピュータに内蔵されて出荷されています。PROM チップはディスクやオペレーティング・システムの一部ではなく、システムで使用できる最も低いレベルのアクセス手段になります。PROM を消去したり、バイパスすることはできません。

通常、PROM チップはシステムの製造後には変更されないため、古いシステムには備わっていない機能もあります。システムによっては、オペレーティング・システムの更新時に、新しいプログラミングに対応する PROM ファームウェアを用意しています。お使いのシステムに必要な PROM については、ハードウェアのユーザ・ガイドを参照してください。

新しいシステムでは、ARCS PROM と呼ばれる PROM が使用されています。ARCS は Advanced RISC Computing Standard (高度 RISC コンピューティング標準) の略です。この PROM はグラフィカル・インタフェースを提供し、マウスによる起動と実行を制御できます。ARCS システムはキーボードの使用もサポートしており、以前からのキー操作も互換性のために残されています。ARCS PROM を使用する SGI システムには、Indigo の一部のモデルと、Indy、Indigo²、Challenge、Onyx、および Crimson システムのほか Origin システムがあります。

コマンド (PROM) モニタの起動方法

コマンド (PROM) モニタを起動するには手順 9-1 に従ってください。

手順 9-1 コマンド・モニタの使用

1. `reboot` コマンドでシステムを再起動します。または、システムの電源がオフになっている場合は、オンにします。

グラフィックス機能がないサーバ・システムでは、次のプロンプトが表示されます。

```
Starting up the system....
```

```
To perform system maintenance instead, press <Esc>
```

グラフィックス機能があるシステムでは、同じようなメッセージが図 9-1 のように画面に表示されます。



図 9-1 ARCS システム起動メッセージ

手順は、グラフィックス・システムの場合もテキスト・システムの場合もほぼ同じですが、グラフィックス・システムの場合は **<Esc>** キーを押す必要はありません。マウス・カーソルで「Stop for Maintenance」ボタンをクリックできます。

- このプロンプトが表示された後、**<Esc>** キーを押すか、適切なボタンをクリックします。次のメニューが表示されます。

```
System Maintenance Menu
1  Start System
2  Install System Software
3  Run Diagnostics
4  Recover System
5  Enter Command Monitor
6  Select Keyboard Layout
```

メモ：メニューの 6 は、Origin システムでは使用できません。

次にメニュー項目とその機能を示します。

Start System デフォルトのオペレーティング・システムを起動します。

Install System Software

inst (1M) のスタンドアロン・バージョンをインストールします。このオプションについての詳細は、『*IRIX Admin: Software Installation and Licensing*』を参照してください。システムに ARCS PROM が用意されている場合は、インストールを実行するために使用するデバイスのタイプを相互に選択し、さらに指定したタイプのデバイスの中から特定のデバイスを選択できます。デバイスのタイプとして、テープ・ドライブ、ネットワーク接続、または CD-ROM ドライブなどがあります。

Run Diagnostics

システム・ハードウェアに関する診断プログラムを実行します。このオプションについての詳細は、システムのユーザ・ガイドを参照してください。

Recover System

以前のシステム・バックアップを読込んで、システム・ディスクを回復します。このオプションについての詳細は、『IRIX Admin: Backup, Security, and Accounting』を参照してください。システムに ARCS PROM が用意されている場合は、回復を実行するために使用するデバイスのタイプを対話的に選択し、さらに指定したタイプのデバイスの中から特定のデバイスを選択できます。デバイスのタイプとして、テープ・ドライブ、ネットワーク接続、または CD-ROM ドライブなどがあります。

Enter Command Monitor

この章で説明するコマンド・モニタ・プログラムを起動します。

Select Keyboard Layout

異なる言語用のキーボード・レイアウトの中から、使用する言語のレイアウトを選択できます。

3. ここで数字 5 を入力して、<Enter> キーを押すか、適切なボタンをクリックすると、次のようなコマンド・モニタのプロンプトが表示されます。

>>

これで、コマンド・モニタになりました。

コマンド・モニタのコマンド・リスト

表 9-1 に、コマンド・モニタコマンドとその構文を示します。すべてのコマンドがすべてのシステムで動作するわけではないので注意してください。

表 9-1 コマンド・モニタのコマンド・リスト

コマンド	説明	構文
arcshinv	ハードウェア・リストを表示します。	arcshinv [-v] [-m] [-mvvv] [-g PATH]
auto	デフォルトのオペレーティング・システムを起動します (引数は指定できません)。これは、PROM モニタの初期メニューから「システムの起動 (Start System)」を選択するのと同じことです。	auto
boot	引数で指定されたファイルを起動します。	boot [-f] [-n] <i>pathname</i>
date	日付と時刻を表示 / 設定します (Origin システムでは使用できません)。	date [<i>mmddhhmm</i> [<i>ccyy</i> <i>yy</i>] [<i>ss</i>]]
disable	ハードウェアを使用不可にします。	disable -m MODID [-cpu [<i>abcd</i>]] [-mem [<i>01234567</i>]] [-s SLOTID -pci [<i>01234567</i>]]
eaddr	システムに内蔵されたイーサネット・コントローラのイーサネット・アドレスを表示します (Origin システムでは使用できません)。	eaddr
enable	ハードウェアを使用可能にします。	enable -m MODID [-cpu [<i>abcd</i>]] [-mem [<i>01234567</i>]] [-s SLOTID -pci [<i>01234567</i>]]
enableall	使用不可にされたすべてのコンポーネントを使用可能にします。	enable [-y] [-list]
exit	コマンド・モニタを終了し、PROM メニューに戻ります。	exit
help	コマンド・モニタのコマンド表を表示します。	help [<i>command</i>] ? [<i>command</i>]
hinv	システムで認識されているハードウェアのリストを表示します。オプション・ボードには PROM モニタに認識されないものもあります。	hinv

表 9-1 コマンド・モニタのコマンド・リスト

コマンド	説明	構文
init	変更された環境変数に応じて部分的にコマンド・モニタを再起動します。	init
ls	指定されたデバイス上のファイルを表示します。	ls <i>devicename</i>
modnum	システムの現在のモジュールを表示します。	modnum
off	システムの電源を切ります (ソフトウェアの電源制御機能を持つシステムのサブセットでのみサポートされています)。	off
passwd	PROM のパスワードを設定します。	passwd
pathname	パス名が有効な場合は、そのパス上のプログラムを実行します。	pathname
ping	IPADDR で ping サーバを指定します。	ping [-f] [-i sec] [-c #] IPADDR
printenv	現在の環境変数を表示します。	printenv [<i>env_var_list</i>]
reset	マシンをリセットします。	reset
resetenv	すべての環境変数をデフォルトに設定し直します。	resetenv
resetpw	PROM のパスワードを null に戻します (パスワード不要)。	resetpw
rtc	リアル・タイム・クロックの周波数を取得または設定します。	rtc [KHz]
setenv	環境変数を設定します。-p フラグを付けると、設定が永続的なものとなり、再起動後も有効になります。	setenv [-p] <i>variable value</i>
setpart	パーティションを作成します。	setpart
single	シングル・ユーザ・モードでシステムを起動します。	single
unsetenv	環境変数の設定を解除します。	unsetenv <i>variable</i>

表 9-1 コマンド・モニタのコマンド・リスト

コマンド	説明	構文
update	保存されているハードウェア・リストの情報を更新します。	update
version	コマンド・モニタのバージョンを表示します。	version

コマンド・モニタでのヘルプ情報の入手方法

疑問符 (?) コマンドは、指定したコマンドの簡単な説明を表示します。コマンドを指定しないで ? コマンドを入力すると、コマンド・モニタのコマンド・リストが表示されます。ヘルプ情報を入手するには、**help** または疑問符 (?) を入力します。

help [*command*]

? [*command*]

コマンド・モニタでのコマンドの使用方法

以下では、次の内容について説明します。

- この章で使用するコマンド構文の表記法。
- 表 9-1 に示したコマンドの機能。

コマンド・モニタでのコマンド行エディタの使用方法

表 9-2 のコマンドを使用してコマンド行を編集することができます。

表 9-2 コマンド行エディタのコマンド

コマンド	説明
Ctrl+h, Delete	直前の文字を削除します。
ctrl+u	行全体を削除します。(?) の後に修正を入力します。

表 9-2 コマンド行エディタのコマンド

コマンド	説明
<code>ctrl+c</code>	実行中のコマンドを中止します。
<code>!!</code>	最後のコマンドを繰り返します。

コマンド・モニタでのコマンドの構文

コマンド・モニタでのコマンドの構文は、IRIX オペレーティング・システムの通常のコマンドで使用する構文に似ています。この章では、IRIX の表記法を使用してコマンドを記述する方法を説明します。

- 固定ピッチ・フォント (Fixed-space font) は、コマンド、ファイル、ルーチン、パス名、信号、メッセージ、およびプログラミング言語構造など、綴りどおりの項目を示します。表示されたとおりに入力します。
- 引数を囲む角かっこ ([]) は、その引数がオプションであることを示します。
- 引数を区切るパイプ (|) は、角かっこの中にあるオプションの引数を 1 つだけ指定できることを示します。
- *file* は、ファイル名を指定する必要があることを示します。ファイル名には、182 ページの「コマンド・モニタのファイル名の構文」で説明されているように、デバイスの指定も含まれます。

コマンド・モニタのファイル名の構文

コマンド・モニタ・コマンドのファイル名を指定する場合は、次の構文を使用します。

```
device( [cntrlr, [unit [, partition] ] ] ) file
```

- *device* は、PROM が認識できるデバイス・ドライバ名を指定します。
- *cntrlr* は、複数のコントローラを持つデバイスのコントローラ番号を指定します。
- *unit* は、指定されたコントローラのユニット番号を指定します。
- *partition* は、ユニット内のパーティション番号を指定します。
- *file* は、アクセスするファイルのパス名を指定します。

cntrlr、*unit*、および *partition* を指定しない場合は、デフォルト値の 0 が適用されます。これは、*cntrlr* のみ、*cntrlr* と *unit*、3 つの変数すべて、のいずれかを指定できることを示します。カンマは区切り文字として重要です。たとえば、単一の SCSI ディスク・システムの root パーティション (パーティション 0) は、次のように表します。

```
dksc (0,1,0)
```

- *dksc* は、SCSI ドライバを示します。
- 最初の 0 は、SCSI コントローラ 0 を示します。
- 1 は、SCSI コントローラ 0 のドライブ番号 1 を示します。
- 最後の 0 は、SCSI コントローラ 0 のドライブ 1 のパーティション 0 (root パーティション) を示します。

同じディスクの /usr パーティション (パーティション 3) は次のように表記されます。

```
dksc (0,1,3)
```

コマンド・モニタのデバイス名

表 9-3 に、コマンド・モニタで定義されているデバイスを示します。

表 9-3 コマンド・モニタ・コマンドでのデバイス名

デバイス名	説明
<i>dksc</i>	SCSI ディスク・コントローラ (IRIX システムでは <i>dks</i>)
<i>tpsc</i>	SCSI テープ・コントローラ (IRIX システムでは <i>tps</i>)
<i>tty</i>	CPU ボード <i>duart</i>
<i>tty(0)</i>	ローカル・コンソール
<i>tty(1)</i>	リモート・コンソール
<i>gfx</i>	グラフィックス・コンソール
<i>console</i>	擬似コンソール <i>gfx(0)</i> 、 <i>tty(0)</i> 、 <i>tty(1)</i> のいずれか
<i>bootp</i>	<i>bootp</i> と TFTP プロトコルを使用しているイーサネット・コントローラ
<i>tpqic</i>	1/4 インチ QIC02 テープ・ドライブ

PROM デバイスの表記法は、IRIX デバイスの表記法とは異なります。一部の環境変数 (*root* と *swap* など) は上位レベルのプログラムに渡され、*/dev* デバイス名を表す IRIX 表記法が必要になる場合があります。たとえば、スワップに最も使用される SCSI デバイスのディスク・パーティションは、PROM の表記法では次のとおりです。

```
dksc(0,0,1)
```

IRIX の表記法では同じディスクが次のようになります。

```
dks0d0s1
```

ARCS PROM のパス名の構文

ARCS PROM を使用しているシステム (Indy, Indigo2, Indigo の一部のモデル、Challenge、および Onyx など) では、パス名やディスク・パーティションを指定する場合に多少異なる構文を使用します。

ARCS のパス名には、ハードウェア・リストと同じ構文を使用します。パス名は、ハードウェアのリスト形式に従い、一連のタイプ (ユニット) コンポーネントとして記述されます。

以前のパス名は、自動的に新しいパス名に変換されますが、以前の名前もそのまま使用できます。PROM は、パス名の最初のデバイス名から検索するので、完全なパス名は不要です。表 9-4 に、よく使用されるパス名の例を示します。

表 9-4 ARCS パス名

ARCS のネーミング規約	パス名またはデバイス
scsi(0)disk(1)partition(1)	dksc(0,1,1)
disk(1)part(1)	dksc(0,1,1)
scsi(0)cdrom(5)partition(7)	dksc(0,5,7)
network(0)bootp() <i>host:file</i>	bootp() <i>host:file</i>
serial(0)	最初のシリアル・ポート
keyboard()	グラフィックス・キーボード
video()	グラフィックス・ディスプレイ

コマンド・モニタの実行方法

ここでは、コマンド・モニタの実行に使用するコマンドについて説明します。表 9-1 に、コマンド・モニタにアクセスできるコマンドを示します。

コマンド・モニタによるプロセッサの再初期化

`init` コマンドを実行すると、PROM メモリによってプロセッサが再初期化されます。`init` 実行後はモニタ・プログラムに戻ります。

PROM パスワードの設定

システムには、コマンド・モニタにアクセスしようとするユーザに対してパスワードを要求する機能があります。PROM パスワードを設定するには手順 9-2 に従ってください。

手順 9-2 PROM パスワードの設定

1. まず「System Maintenance」メニューからオプション 5 を選択して、コマンド・モニタを起動します。コマンド・モニタ・プロンプトが表示されます。

```
Command Monitor. Type "exit" or click on "done" to return to the menu.  
>>
```

2. 次のコマンドを入力します。

```
help
```

3. `passwd` コマンドを実行します。

```
passwd
```

次のプロンプトが表示されます。

```
Enter new password:
```

4. システムで使用するパスワードを入力し、**<Enter>** キーを押します。次のプロンプトが表示されます。

```
Confirm new password:
```

もう一度パスワードを入力します。最初と同じパスワードを入力すると、再びコマンド・モニタ・プロンプトが表示されます。入力を間違えると、エラー・メッセージが表示されるので、もう一度はじめてやり直します。エラー・メッセージが表示されない場合は、パスワードの設定は完了です。次回からコマンド・モニタにアクセスする場合、必ずこのパスワードの入力が要求されます。

パスワードの選択と入力には慎重に行ってください。パスワードを誤って入力したり、忘れてしまうと、システムの CPU ボードのジャンパを取外さなければなりません。この手順はシステムの種類によって異なるので、各システムのユーザ・ガイドを参照してください。ただし、一部のシステムでは、*root* でログインし、次のコマンドを実行すると、IRIX オペレーティング・システムから PROM パスワードをリセットできる場合があります。

```
nvram passwd_key ""
```

PROM パスワードを削除するには、引用符内に文字やスペースを挿入しないでください。この操作は、*root* でログインしていなければ実行できません。

コマンド・モニタの `resetpw` コマンドでも、PROM のパスワードをリセットできます。

コマンド・モニタの環境変数

コマンド・モニタは、変数名とそれに対応する値（実際にはテキスト文字列）のリストである環境を保守します。これらの環境変数には、コマンド・モニタが使用する情報や、起動されたプログラムに渡すための情報が含まれています。システムは、頻繁に変更されない重要な環境変数を不揮発性 RAM(nvram) に格納しています。また、マシンの電源を切ったり、リセット・ボタンを押しても、システムはそれらの変数を記憶しています。setenv コマンドを使用し、これらの変数の設定を変更すると、PROM コードは新しい値を自動的に不揮発性 RAM に格納します。

`/etc/nvram` コマンドを使用し、不揮発性 RAM に格納されている変数の値を設定、または表示できます。nvram コマンドの詳細については、nvram(1M) マン・ページを参照してください。

表 9-5 に、システムが不揮発性 RAM に格納する環境変数のリストを示します。

表 9-5 不揮発性 RAM に格納されている環境変数

変数	説明
netaddr	イーサネットを介して起動するためのローカル・ネットワーク・アドレスを指定します。bootp プロトコルを参照してください。
dbgtty	IRIX オペレーティング・システムの対話型デバugga tty を指定します。
root	IRIX システムに渡されるファイルシステム情報を指定します。
dbaud	診断コンソールのボー・レートを指定します。ボー・レートは、75、110、134、150、300、600、1200、2400、4800、9600、19200 のいずれかを設定するか <Break> キーを押して変更できます。IRIS はシステム起動中の診断コンソールのレートとして dbaud を使用します。<Break> キーを押した場合の変更は一時的なもので、「リセット (Reset)」ボタンを押すか init コマンドを実行すると、dbaud または rbaud に設定された値に戻ります。
rbaud	リモート・コンソールのボー・レートを指定します。指定可能な値は上記の dbaud と同じです。
bootfile	自動起動に使用するファイル名を指定します。通常はスタンドアロン・シェル (sash) です。この変数は前述の ARCS PROM に対してのみ有効です。ARCS PROM はこの情報を OSLoader 変数に格納します。
bootmode	<p>前述の ARCS PROM での起動タイプを指定します。ARCS PROM はこの情報を AutoLoad 変数に格納します。各オプションの意味は次のとおりです。</p> <p>c は、自動起動をそのまま実行します。bootfile 変数が示すファイルを使用して sash を起動した後カーネルを起動し、電源投入時の診断を行います。</p> <p>m (デフォルト) は、直接コマンド・モニタに入り、メモリをクリアして電源投入時の診断を行います。</p> <p>d は、直接コマンド・モニタに入りますが、メモリのクリア、電源投入時の診断は行いません。</p>
boottone	起動時の音楽のメロディを選択します。0 では音楽がランダムに選択されます。1 がデフォルトです (この変数は Power Indigo2 と Octane システムでのみサポートされています)。
autopower	電源制御が可能なソフトウェアがあるシステムでは、y にしておくで電源切断時に自動的にリセットされます (この変数は Power Indigo2 と Octane システムでのみサポートされています)。

表 9-5 不揮発性 RAM に格納されている環境変数

変数	説明
console	<p>使用するコンソールの種類を指定します。各オプションの意味は次のとおりです。</p> <p>G は、グラフィックス・コンソールで左上に SGI のロゴを表示します。</p> <p>g (デフォルト) は、グラフィックス・コンソールで SGI のロゴを表示しません。</p> <p>d は、端末を使用不可にします。</p>
keybd	<p>使用するキーボードのタイプを指定します。デフォルトは df です。PROM のバージョンにより、以下のすべて、または一部から選択できます。</p> <p>USA, DEU, FRA, ITA, DNK, ESP, CHE-D, SWE, FIN, GBR, BEL, NOR, PRT, CHE-F</p> <p>または</p> <p>キーボードのレイアウト・セレクトがあるシステムでは以下の設定が可能です。US, DE, FR, IT, DK, ES, deCH, SE, FI, GB, BE, NO, PT, frCH</p> <p>システムによっては JP という設定があり、日本語キーボードを指定できます。</p>
dbgname	デバッグ symmon を取得する場所を指定します。
diskless	<p>ディスクレス・システムの場合に、ネットワーク経由で起動することを指定します。ARCS システムではディスクレス・システムの環境パラメータは次のように設定します。</p> <p>diskless=1</p> <p>SystemPartition=bootp()host:/path</p> <p>OSLoader=kernelname</p>
monitor	認識できないブランドのモニタを使用する場合に、Indy システム上でモニタの解像度を指定します。高解像度を指定する場合は、h または H に設定します。デフォルトは低解像度になります。
nogfxkbd	キーボードを接続する必要がない場合は 1 に設定します。
notape	システムにテープ・ドライブが接続されていないという指定です。システムにテープ・ドライブが接続されていて、ネットワーク上のほかのシステムのテープ・ドライブにアクセスするには、この変数を 1 (真) に設定します。
volume	システム・スピーカの音量を数値で指定します。
pagecolor	テキストポートの背景色を 6 桁の 16 進数 RGB 値で指定します。
ProbeAllScsi	SCSI バス上のすべてのデバイスがディスク・チェックを自動的にを行うことを指定します。
prompoweroff	Indy システムでのみ指定が可能です。y に指定すると、システムを切断するとき電源が切れる前に PROM モニタに戻ります。

表 9-5 不揮発性 RAM に格納されている環境変数

変数	説明
rebound	y に指定すると、カーネルがパニックになった後にシステムが自動的に再起動します。この変数は、AutoLoad 変数および reboot_on_panic カーネル・チューニング・パラメータと相互関係にあります。
RestorePartEnv	パーティション情報が復元されるかどうかを指定します (SGI 3000 Series システムでのみサポートされています)。
sgilogo	y に指定すると、PROM モニタのグラフィック表示画面に SGI のロゴと関連情報を表示します。
diagmode	電源投入時の診断のモードを指定します。v に指定すると、全体を診断します。

表 9-6 に、IRIX オペレーティング・システムの動作に直接影響を及ぼすコマンド・モニタの環境変数を示します。これらの変数は不揮発性 RAM には格納されませんが、PROM と IRIX システムの動作に影響を及ぼします。また、システムの電源を切ると消去されてしまうので注意してください。

表 9-6 IRIX オペレーティング・システムに影響を及ぼす環境変数

変数	説明
showconfig	IRIX 起動時に追加情報を表示します。setenv を使用して設定する場合は、istrue を指定します。
initstate	IRIX システムに渡されて /etc/inittab の initdefault 行を上書きします。可能な値は s および 0 ~ 6 の数値です。init (1M) マン・ページを参照してください。
swap	使用するスワップ・パーティションを IRIX の表記法で指定します。指定がない場合は、オペレーティング・システムで設定されたデフォルトのパーティションになります。通常、それは root 環境変数によって設定されたドライブのパーティション 1 です。
path	特にデバイスが指定されていない場合、コマンド・モニタがファイルを検索するときのデバイスのプリフィックスのリストを指定します。
verbose	システムで詳細なエラー・メッセージを表示します。

コマンド・モニタからプログラムを起動すると、起動されたプログラムには現在の環境変数の設定が渡されます。

ARCS PROM が定義する環境変数には、以前の PROM にはなかったものがいくつかあります。表 9-7 に、それらの変数を示します。

表 9-7 ARCS PROM 環境変数

変数	説明
ConsoleIn/ConsoleOut	システム起動時に自動的に設定されます。
OSLoadPartition	オペレーティング・システムのカーネルが置かれているディスク・パーティション。デフォルトの root パーティションとしても使用され、システム起動時に自動的に設定されます。
OSLoader	オペレーティング・システムをロードするプログラム。デフォルトでは sash (スタンドアロン・シェル) です。システム起動時に自動的に設定されます。
SystemPartition	オペレーティング・システムをロードするプログラムがあるディスク・パーティション。システム起動時に自動的に設定されます。
OSLoadFilename	オペレーティング・システムのカーネルのファイル名。デフォルトでは /unix です。システム起動時に自動的に設定されます。
OSLoadOptions	オペレーティング・システムをロードする boot コマンドのオプションを指定します。boot オプションについての詳細は「コマンド・モニタでのプログラムの起動方法」を参照してください。
AutoLoad	リセットまたは電源切断の後でオペレーティング・システムを自動的に起動するかどうかを指定します。この変数は、bootmode に優先するもので Yes か No で指定します。この変数は、rebound 変数および reboot_on_panic カーネル・チューニング・パラメータと相互関係にあります。

現在のコマンド・モニタ環境変数の表示

printenv コマンドは、コマンド・モニタの現在の環境変数を表示します。

```
printenv [env_var_list]
```

変数を変更 (リセット) する方法については、次を参照してください。

コマンド・モニタ環境変数の変更

`setenv` コマンドは、設定されている環境変数の値の変更、または新しい環境変数の設定を行います。

```
setenv env_var string
```

`env_var` は設定する変数、`string` はその変数に代入する値です。現在の環境変数の設定を確認するには、`printenv` を使用します。

`setenv` を使用して不揮発性 RAM に格納されている環境変数（187 ページの表 9-5 参照）の値を変更すると、システムはその新しい値を自動的に不揮発性 RAM に格納します。このため、システムの電源を切って再び電源を入れる際に、再度入力する必要はありません。

キーボード変数の設定

`keybd` 変数をデフォルトの `df` 以外の値に設定すると、ハード・ディスクのボリューム・ヘッダから適切なキーボード変換テーブルがロードされます。キーボード変換テーブルがない場合、またはロードできない場合は、PROM に格納されているデフォルトの変換テーブルが使用されます。`keybd` 変数は任意値に設定できますが、キーボード変換テーブルはハード・ディスクのボリューム・ヘッダからロードされる必要があります。この変数は、システムにインストールするキーボードの種類を決定する通常システム機構よりも優先的に解釈されます。キーボード診断を行ってから、この変数を変更してください。表 9-8 に、国際キーボードに使用されている `keybd` 変数を示します。

表 9-8 国際キーボード用の `keybd` 変数

変数	説明
<code>BEL</code> または <code>BE</code>	ベルギー語
<code>DNK</code> または <code>DK</code>	デンマーク語
<code>DEU</code> または <code>DE</code>	ドイツ語
<code>DF</code>	デフォルト
<code>FRA</code> または <code>FR</code>	フランス語
<code>FIN</code> または <code>FI</code>	フィンランド語
<code>ITA</code> または <code>IT</code>	イタリア語

表 9-8 国際キーボード用の *keybd* 変数

変数	説明
JP	日本語
NOR または NO	ノルウェー語
PRT または PT	ポルトガル語
CHE-F または freCH	スイス・フランス語
CHE-D または deCH	スイス・ドイツ語
ESP または ES	スペイン語
SE または SWE	スウェーデン語
GB または GBR	イギリス英語
US または USA	アメリカ英語 (すべてのモデルで選択可能)

環境変数の削除

`unsetenv` コマンドは、環境変数の設定を削除します。

```
unsetenv env_var
```

env_var は、設定を削除する環境変数です (191 ページの「コマンド・モニタ環境変数の変更」の `setenv` に関する説明を参照)。ただし、不揮発性 RAM に格納されている変数の設定は削除できないので注意してください。

コマンド・モニタでのプログラムの起動方法

ここでは、コマンド・モニタの各起動コマンドとその使用方法について説明します。再起動するか、「リセット (Reset)」ボタンを押すと、コマンド・モニタが起動します。ただし、通常の状態 (ワークステーションが IRIX オペレーティング・システムを実行している状態) では、「リセット (Reset)」ボタンは押さないでください。

auto コマンドによるデフォルト・ファイルの起動

auto コマンドはオペレーティング・システムを再起動します。このコマンドは、CPU の電源を入れる場合と同じようにデフォルトの起動ファイルを使用します。コマンド・モニタ・プロンプト (>>) に対して次のように入力します。

```
auto
```

PROM の環境変数 *bootfile* は、デフォルトの起動ファイルを指定します。また、環境変数 *root* に、IRIX オペレーティング・システムが *root* ファイルシステムとして使用するディスク・パーティションを設定する必要があります。auto コマンドは、環境変数 *bootfile* で設定されたドライブの *root* で設定されたパーティションに、IRIX システムのデータが常駐しているものとします。

bootfile 名に設定できるのは 14 文字までです。別の起動ファイルを選択する場合は、191 ページの「コマンド・モニタ環境変数の変更」を参照してください。

boot コマンドによる固有プログラムの起動

固有の起動プログラムを使用し、そのプログラムにオプションの引数を指定してシステムを起動する場合は、boot コマンドを使用します。boot コマンドの構文は次のとおりです。

```
boot [-f program] [-n] [args]
```

- **-f** には起動するプログラムを指定します。プログラム名は 19 文字以内でなければなりません。このオプションを指定しない場合は、環境変数 *bootfile* はデフォルトのプログラムを設定します。通常、boot は *sash* をロードします。

プログラムを指定する場合は、デバイス仕様も指定できます。デバイスを指定しない場合は、コマンド・モニタは環境変数 *path* で設定されたデバイス仕様を使用します。コマンド・モニタは、要求されたプログラムが検索されるまで、*path* に設定されているすべてのデバイスを順番に検索します。

- **-n** は停止を意味します。**-n** は指定されたプログラムをロードしますが、そのプログラムに制御を渡しません。コマンド・モニタのコマンド環境に制御を戻します。
- *args* は、起動するプログラムにコマンド・モニタが渡す変数です。ハイフン (-) で始まる *arg* の場合は、その前にハイフンをもう 1 つ追加し、コマンド・モニタがその引数をそれ自体に指定されたものと解釈しないようにします。コマンド・モニタは、起動されたプログラムに

その引数を渡す前に、余分なハイフンを取除きます。詳細については、194 ページの「スタンドアロン・シェル (sash)」を参照してください。

たとえば、カートリッジ・テープ・ドライブからディスク・フォーマッタまたは検査プログラム (fx) を起動するには、次のコマンドを使用します。

```
boot -f SCSI(0)tape(7)partition(0)fx
```

引数の指定がない場合、boot は変数 bootfile に設定されたプログラムをロードします。

スタンドアロン・シェル (sash)

コマンド・モニタは、オペレーティング・システムから独立できるように、最小のサイズで設計されています。このため、コマンド・モニタは、IRIX やそのほかのオペレーティング・システムのファイル・ツリーに常駐しているファイルを直接起動できません。ただし、コマンド・モニタには、ファイルシステムを認識できるように中間プログラムのロードを行う 2 レベルの起動機構が組込まれています。この機構によって、中間プログラムは任意の起動ファイルを検索し、それをロードできます。このプログラムはスタンドアロン・シェルと呼ばれ、sash と表記されます。sash は、コマンド・モニタ・プログラムを再構成した拡張バージョンです。sash にはオペレーティング・システムのファイル構造を取扱うために必要なモジュールが組込まれています。また、デバイスに関する情報も拡張されています。

システム・ソフトウェアをインストールすると、sash のコピーは第 1 ディスクのボリューム・ヘッダに位置付けられます。このヘッダのファイル構造は単純なので、コマンド・モニタもこれを認識できます。また、必要に応じて、テープやネットワーク経由で sash を起動することもできます。

スタンドアロン・シェルの起動

sash をディスクから起動するには、手順 9-3 に従ってください。

手順 9-3 sash を使った起動

1. システムを停止します。
2. 次のようなメッセージが表示されます。

```
Starting up the system...
```

To perform system maintenance instead, press <Esc>

上記のメッセージが表示されたら、<Esc> キーを押します。コマンド・モニタ・パスワードが設定されている場合は、そのパスワードを入力しなければなりません。次のようなメニューが表示されます。

```
System Maintenance Menu
(1) Start System
(2) Install System Software
(3) Run Diagnostics
(4) Recover System
(5) Enter Command Monitor
```

3. 「System Maintenance」メニューからオプション5の「Enter Command Monitor」を選択します。次のメッセージとプロンプトが表示されます。

```
Command Monitor. Type "exit" or click on "done" to return to the
menu.
```

```
>>
```

4. 次のコマンドを入力します。

```
boot -f sash
```

sash は対話型コマンド・モードで動作します。sash プロンプトが表示されます。

```
sash:
```

マルチレベル起動機能を使用する場合は、sash の特定コピーを参照するように、PROM の環境変数 *bootfile* を設定します。通常の設定では、次のように *bootfile* を *dksc(0,0,8)sash* に設定すると、SCSI ディスク・コントローラ 0、ディスク・ユニット 0、パーティション 8（ボリューム・ヘッド）から sash をロードするようにコマンド・モニタに指示します。

```
setenv bootfile "dksc(0)disk(0)partition(8)sash"
```

次に起動コマンドを入力します。この例では、SCSI ドライブになります。

```
boot dksc()unix initstate=s
```

次の動作が実行されます。

- 起動コマンドには、**-f** 引数が指定されていないので、`boot` は `bootfile` の指定どおりに `dksc(0)disk(0)partition(8)sash` をロードします。**-f** 引数が指定された場合、`bootfile` に設定されているデフォルトは無効になります。
- `sash` は 2 つの引数、**dksc()unix** と **initstate=s** を取得します。この結果、システムはシングル・ユーザ・モードに入ります。コマンド・モニタは、引数から先頭のハイフン (-) を取除きます。したがって、先頭にハイフンの付いた引数を子プロセスに渡す場合は、その引数の前に 2 つのハイフンを付けてください。
- `sash` は第 1 引数 (**dksc()unix**) によって指定されたファイルをロードし、次の引数をそのファイルに渡します。

上記のように設定した `bootfile` を使用し、`sash` から `auto` コマンドを実行しないでください。実行した場合は、システムが `sash` を起動しようとするので、エラーが発生して終了します。

`sash` から `auto` コマンドを使用可能にするには、`bootfile` が `dksc()unix` などのカーネルを参照するように設定する必要があります。`auto` コマンドを使用する場合は、**PROM** レベルに戻ることをお勧めします。

bootp プロトコル

ディスクレス・ワークステーションの操作の中核となるのが、`bootp` プロトコルです。`bootp` プロトコルは、SGI 社製のすべてのサーバとワークステーションでサポートされる DARPA 標準プロトコルです。コマンド・モニタが起動用に使用できるデバイスの 1 つに、イーサネット・ネットワークがあります。SGI 社の製品では、ネットワーク上のほかのホストが TCP/IP 起動プロトコルをサポートしている場合、そのホストあるファイルを起動できる TCP/IP 起動プロトコルを提供しています。このネットワーク起動プロトコルが `bootp` プロトコルです。TCP/IP のユーザ・データグラム・プロトコル (UDP: User Datagram Protocol) を使用し、イーサネット・ネットワーク経由でファイルを転送するのは、データグラム・プロトコルです。

bootp によるネットワークを経由した起動

ネットワークを経由して起動する場合は手順 9-4 に従ってください。

手順 9-4 `bootp` を使った起動

1. 起動するマシンのインターネット・アドレスを確定します。

インターネット・アドレスは、システムが接続しているネットワークのネットワーク管理者によって割当てられた番号になります。番号の形式は、ピリオドで区切られた 0 から 255 の範囲の 4 つの 10 進数になります。次の例を参照してください。

```
194.45.54.4
```

2. `setenv` コマンドを使用して、次のように環境変数 `netaddr` をこのアドレスに設定します。

```
setenv netaddr 194.45.54.4
```

環境変数 `netaddr` を設定した後、`bootp` を使用してリモート・ファイルを参照する場合は、次の書式のファイル名を指定します。

```
bootp() [hostname:] path
```

- *hostname* はファイルが常駐しているホストの名前です。指定されたホストは `bootp` サーバ・デーモンである `bootp` を実行していません。*hostname* を省略すると、`bootp` はブロードキャストを実行し、要求を発行しているシステムと同じネットワーク上の任意のホストからファイルを取得します。条件を満たした最初のホストからファイルを取得します。要求に回答できるホストは、`bootp` をサポートするホストだけです。パスが特定のホストに対し一意であることがわかっている場合、または、ファイルのコピーすべてが交換可能であることがわかっている場合のみ、ホスト名を省略できます。

hostname には、起動するマシンとは異なるイーサネット・ネットワーク上のホスト名を指定できます。ただし、ローカル・イーサネット・ネットワーク上のゲートウェイがリモート・ホストへの経路を提供していません。また、このゲートウェイは、クロス・ネットワーク転送を行うために構成されている `bootp` サーバでなければなりません。

ゲートウェイを通じて行う起動についての詳細は、`bootp(1M)` マン・ページを参照してください。また、`/etc/inetd.conf` の設定ファイルについての詳細は、`inetd(1M)` マン・ページを参照してください。

- *path* はリモート・ホスト上のファイルのパス名です。たとえば、次のコマンドは、リモート・ホスト `wheeler` から `/usr/local/boot/unix` ファイルを起動します。

```
boot -f bootp()wheeler:/usr/local/boot/unix
```

次のコマンドは、`bootp` ブロードキャスト要求に応じたネットワーク上の任意のホストから `/usr/alice/help` を起動します。

```
boot -f bootp()/usr/alice/help
```

ネットワークの相互転送を行うゲートウェイを構成するには手順 9-5 に従ってください。

手順 9-5 ゲートウェイの構成

1. **root** でログインするか、`su` コマンドを使用して特権ユーザになります。
2. ゲートウェイ・システムの `/etc/inetd.conf` ファイルを編集します。このファイルは `bootp` サーバを構成します。`bootp` サーバは `inetd(1M)` デーモンによって起動します。
3. `inetd` が **-f** フラグ付きで `bootp` を呼出せるように `bootp` の内容を変更します。ファイル内で次の行を検索します。

```
bootp dgram udp wait root /usr/etc/bootp bootp
```

この行の最後の `bootp` の後に **-f** フラグを追加します。

```
bootp dgram udp wait root /usr/etc/bootp bootp -f
```

4. 次のどちらかの方法で `tftpd` の設定内容を変更します。

`tftpd` の引数リストから **-s** フラグを削除します。

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -s
```

この結果、`tftpd` は読取り可能なディレクトリにアクセスできます。ただし、セキュリティ上の問題がある場合は、`tftpd` がアクセスを必要とするディレクトリを明示的に指定することもできます。この場合は、`/usr/etc` を追加します。

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -s /usr/etc
```

詳細については、`tftpd(1M)` と `tftp(1C)` のマン・ページを参照してください。

5. `inetd` にシグナルを送り、設定ファイルを再び読取らせませす。

```
killall -1 inetd
```

大規模なネットワークを経由した起動

大きなネットワークにアクセスしていて、必要とする起動可能なファイルがネットワーク内でリモートにあるために、`tftp` や `bootp` ではタイムアウトになる場合、または、ネットワーク遅延のために起動が正常に行われない場合は、`bootp` サーバなどの中間ホストを使用できます。

たとえば、ここにリモート・システム `far_host` にあるカーネルを起動する必要がある `local_host` という名前のホストがあるとします。ただしネットワークの負荷が高いため、起動操作が行われる前に、`bootp` と `tftp` とがタイムアウトになってしまいます。3 番目のホストである `near_host`

がオプションの NFS ソフトウェアを備え、`automount` を実行している場合には、`far_host` のファイルへのアクセスが可能になります。この方法で起動するには手順 9-6 に従ってください。

手順 9-6 automount を使った起動

1. 中間のシステムとして動作している `near_host` に `root` としてログインし、`/etc/inetd.conf` ファイルを開きます。このファイルは、`inetd(1M)` デーモンによって起動される `bootp` サーバに影響します。
2. `inetd` が `-f` フラグ付きで `bootp` を呼出せるように `/etc/inetd.conf` ファイルの `bootp` の記述を変更します。

```
bootp dgram udp wait root /usr/etc/bootp bootp
```

この行の最後の `bootp` の後に、`-f` フラグを付けます。

```
bootp dgram udp wait root /usr/etc/bootp bootp -f
```

次の 2 つの方法のどちらかで `/etc/inetd.conf` ファイルの `tftpd` の設定内容を変更します。

`tftpd` 用の引数リストから、`-s /usr/local/boot` という文字列を削除します。次に、この結果を示します。

```
tftpd dgram udp wait guest /usr/etc/tftpd tftpd
```

これで、`tftpd` は誰でも読取りが可能なディレクトリのすべてにアクセスできるようになります。

セキュリティ上の問題がある場合は、`tftpd` がアクセスを必要とするディレクトリを明示的に指定することもできます。この場合は、`/hosts` を追加します。

```
tftpd dgram udp wait guest /usr/etc/tftpd tftpd -s /hosts
```

詳細については、`tftpd(1M)` と `tftpd(1C)` のマン・ページを参照してください。

3. `inetd` にシグナルを送り、設定ファイルを再び読取らせませす。

```
killall -1 inetd
```

4. 離れたサブネットワーク内のシステム `far_host` で、NFS を使用して必要な起動可能カーネルを含むディレクトリをエクスポートします。この例では、`/usr/local/boot/unix` です。ディレクトリのエクスポートについては、`export(1M)` マン・ページを参照してください。
5. 起動するシステム `local_host` で、次のコマンドを入力します。

```
boot -f bootp()near_host:/hosts/far_host/usr/local/boot/unix
```

bootp がタイムアウトになった場合、コマンドを再び実行してください。automount は、リモート・システムからファイルを取得するために若干の時間を必要とする場合があります。

ディスクまたはほかのデバイスからの起動

ディスクや CD-ROM などのさまざまなリソースからスタンドアロンのコマンドをロードするようにコマンド・モニタに指示するには、*path* 環境変数を設定します。191 ページの「コマンド・モニタ環境変数の変更」を参照してください。*path* 変数は、次のように設定します。

```
setenv path "device_name alternate_path"
```

たとえば、次のコマンドを実行します。

```
setenv path "dksc(0)disk(0)partition(8)bootp()/altdir/altbootfile"
```

この結果、コマンド・モニタは `dksc(0)disk(0)partition(8)/altdir/altbootfile` ファイルを起動します。このファイルがない場合は、コマンド・モニタは `bootp()/altdir/altbootfile` ファイルを起動します。このファイルもない場合は、コマンド・モニタはメッセージ `command not found` を表示します。パス名はスペースで区切ります。デバイス仕様がコマンド内または *bootfile* にある場合は、コマンド・モニタは *path* を無視します。PROM が認識するのは、bootp またはボリューム・ヘッダのみです。

システム・パフォーマンスの調整

この章では、システムのパフォーマンスを向上させる手順について説明します。ここでは、特定のニーズに合わせてパフォーマンスを最大化するために、IRIX オペレーティング・システムを調整する基本的な方法について触れます。

システムは、ほとんどすべての状況で最高速度で実行するように構成されています。ただし、特定のパラメータやオペレーティング・システムの値を調整すると、システムの全般的なパフォーマンスを向上させることができたり、ディスク・アクセスなど特定の機能を最適化すると、アプリケーション・ソフトウェアのグラフィックス機能をさらに活用することができます。

この章では、以下の項目について説明します。

- 64 プロセッサまたはそれより上のシステム構成のための、推奨する設定パラメータに関する情報や、システム調整およびカーネル・パラメータに関する一般情報。202 ページの「システム・パフォーマンスの調整」を参照してください。
- オペレーティング・システムを監視し、調整の必要性を決定する方法。206 ページの「オペレーティング・システムの監視」を参照してください。
- オペレーティング・システムの調整と再構成。223 ページの「オペレーティング・システムの調整」を参照してください。
- 再構成の試みに失敗した場合の対処法。228 ページの「起動できないカーネルからの回復」を参照してください。
- 複数のページ・サイズの使用。229 ページの「複数のページ・サイズ」を参照してください。

開発環境でのアプリケーションの調整に関する情報は、付録 C, 「アプリケーションの調整」を参照してください。

システム・パフォーマンスの調整

IRIX の標準的なシステム構成は、幅広い用途のために設計されており、特殊な場合を除き、どんな条件下でも効率的に動作するように自動的に調整されます。オペレーティング・システムは、メモリ内のプログラムの実行や、ディスクとメモリ間でのプログラムの移動を制御します。

手順 10-1 で、システムを調整する基本的な方法を説明します。

手順 10-1 システムの調整手順

1. 各種のユーティリティを使用し、システム・パフォーマンスを監視します。
2. 特定のパラメータ（プロセスの最大数など）を調整します。
3. 必要に応じてシステムを再起動します。
4. 新しいシステム・パフォーマンスをテストし、改善されていることを確認します。

パフォーマンスを調整しても、システムの能力がそのハードウェアの限界以上に向上することはありません。さらにパフォーマンスを向上させるには、ディスクやメモリなどのハードウェアを追加する必要があります。

カーネルの調整に使用するファイル

表 10-1 に、システムの調整と再構成に使用するファイルとディレクトリを示します。

表 10-1 調整に使用するファイルとディレクトリ

ファイル/ディレクトリ	目的
/var/sysgen/system/*	ソフトウェア・モジュールを定義するファイルを含むディレクトリ
/var/sysgen/master.d	カーネル・スイッチおよびパラメータを定義するファイルを含むディレクトリ
/var/sysgen/mtune/*	詳細なチューニング・パラメータを定義するファイルを含むディレクトリ
/var/sysgen/stune	パラメータのデフォルト値を定義するファイル
/var/sysgen/boot/*	オブジェクト・ファイルのディレクトリ
/unix	カーネル・イメージを含むファイル

通常、`systune` コマンドを使用して、`mtune` ディレクトリにあるファイル (`kernel` ファイルなど) 内のパラメータを調整します。詳細については、`systune(1M)` マン・ページを参照してください。

カーネルのチューニング・パラメータの概要

チューニング・パラメータは、プロセス、ファイルおよびシステム・アクティビティの特性を制御します。予想されるシステム負荷を処理できるように、各種テーブルのサイズとシステムのしきい値を設定します。システム構成が大きすぎると、ほかのプロセスのために使用されるはずのメモリ領域を消費してしまい、テーブルの検索に時間を消費してシステムのオーバーヘッドを増大させてしまいます。反対に、システム構造の設定が小さすぎると、特定のパラメータ設定によっては、過剰な入出力や処理の異常終了、さらにシステム・クラッシュまで起こす場合もあります。

ここでは、チューニング・パラメータについて簡単に説明します。各パラメータの説明やデフォルト値、変更する時期に関するヒント、発生し得る問題などについては、付録 A, 「IRIX カーネルのチューニング・パラメータ」を参照してください。

チューニング・パラメータは、ディレクトリ `/var/sysgen/mtune` と `/var/sysgen/master.d` 内の複数のファイルによって指定されています。`mtune` については `mtune(4)` マン・ページを参照してください。`master.d` については、`master(4)` マン・ページを参照してください。

チューニング・パラメータのデフォルト値は、シングル・ユーザ・ワークステーション環境のほとんどの構成で有効です。メモリが大量にある場合、または環境に特別な設定が必要な場合は、そのニーズに合わせてパラメータ値を変更できます。以下に、調整可能なパラメータを示します。

`nproc` システム全体のプロセスの最大数。通常、自動的に設定されます。

`maxup` UID あたりのプロセスの最大数。

`rlimit_core_cur`
コア・ファイルの最大サイズ。

`rlimit_data_cur`
プロセスが使用できるデータ空間の最大量。

`rlimit_fsize_cur`
プロセスが使用できる最大のファイル・サイズ。

<code>rlimit_nofile_cur</code>	プロセスが使用できるファイル記述子の最大数。
<code>rlimit_rss_cur</code>	プロセスが使用できる最大の常駐セット・サイズ。
<code>rlimit_vmem_cur</code>	プロセス用のマップ・メモリの最大量。
<code>sshmseg</code>	プロセスあたりの共有メモリ・セグメントの最大数。

大きなシステム構成のチューニング・パラメータ

表 10-2 に、64 プロセッサ以上のシステム構成で、推奨されるチューニング・パラメータを示します。これらのパラメータについての詳細は、付録 A、「IRIX カーネルのチューニング・パラメータ」を参照してください。

メモ: これらのパラメータはシステム構成に依存します。したがって、リストの値は、推奨される初期値を示しています。特定のシステム構成またはアプリケーション・セットに合わせてこれらのパラメータ値を変更し、変更結果を評価してください。

また、`maxup` パラメータの値は、`nproc` パラメータ値から 20 を引いた値より小さくなります。`sysstune` では `maxup` 値を `nproc` 値以上に設定できますが、システムを再起動するか、カーネルを再設定すると、`nproc` から 20 を引いた値が `maxup` の上限値になります。`sysstune` についての詳細は、`sysstune(1M)` マン・ページを参照してください。

表 10-2 大きなシステム構成のチューニング・パラメータ

パラメータ	推奨する初期値
<code>dump_level</code>	3
<code>maxdmasz</code>	0x2001
<code>rsshogfrac</code>	99
<code>rlimit_stack_max</code>	0x20000000 ll
<code>rlimit_stack_cur</code>	0x20000000 ll

表 10-2 大きなシステム構成のチューニング・パラメータ

パラメータ	推奨する初期値
<code>rlimit_rss_max</code>	<code>0x20000000 ll</code>
<code>rlimit_rss_cur</code>	<code>0 ll</code>
<code>rlimit_data_max</code>	<code>0 ll</code>
<code>rlimit_data_cur</code>	<code>0 ll</code>
<code>rlimit_vmem_max</code>	<code>0 ll</code>
<code>rlimit_vmem_cur</code>	<code>0 ll</code>
<code>nbuf</code>	<code>2000</code>
<code>syssegsz</code>	<code>0xfe800</code>
<code>sshmseg</code>	<code>2000</code>
<code>shmmax</code>	<code>(0x4000000) ll</code>
<code>semmni</code>	<code>2000</code>
<code>semume</code>	<code>80</code>
<code>semopm</code>	<code>80</code>
<code>gpgshi</code>	<code>2000</code>
<code>gpgslo</code>	<code>1000</code>
<code>maxup</code>	<code>7980</code>
<code>nproc</code>	<code>8000</code>
<code>percent_totalmem_1m_pages</code>	<code>0</code>
<code>percent_totalmem_4m_pages</code>	<code>0</code>
<code>percent_totalmem_16m_pages</code>	<code>0</code>
<code>percent_totalmem_64k_pages</code>	<code>0</code>
<code>percent_totalmem_256k_pages</code>	<code>0</code>

オペレーティング・システムの監視

カーネル・パラメータを変更する場合は、まず、どのパラメータをどういう理由で変更する必要があるのかを確認してください。オペレーティング・システムの動作を監視することによって、パラメータの変更がパフォーマンスの改善に役立つのか、または新しいハードウェアが必要なのかを確定できます。

カーネルからのメッセージの受信とテーブル・サイズの調整

システムの要求に見合うだけの十分なサイズがない場合に、テーブルがオーバーフローすることがあります。このような場合は、エラー・メッセージがコンソールに表示され、`/var/adm/SYSLOG` ファイルにエラー・メッセージが書込まれます。コンソール・ウィンドウがクローズまたは保存されている場合は、`SYSLOG` を定期的にチェックしてください。

システム・コールには、システムの状態を示すエラー・メッセージを返すものがあります。パラメータのサイズを増やす必要があることを知らせるエラー・メッセージもその 1 つです。表 10-3 に、エラー・メッセージと調整を必要とするパラメータを示します。これらのパラメータは `/var/sysgen/master.d/kernel` ファイルに格納されています。

表 10-3 システム・コール・エラーと関連するパラメータ

メッセージ	システム・コール	パラメータ
EAGAIN プロセスをこれ以上作成できません	fork(2)	<i>nproc</i> またはスワップ領域を増やしてください。
ELIBMAX システムの上限を超えて共有ライブラリをリンクしようとしています	exec(2)	<i>shlbmax</i> チューニング・パラメータを増やしてください。
E2BIG 引数リストが長過ぎます	shell(1)、 make(1)、 exec(2)	<i>ncargs</i> チューニング・パラメータを増やしてください。

上記のエラーは、別の理由からも発生することがあります。たとえば、仮想メモリの不足で EAGAIN が表示されることもあります。この場合は、スワップ領域を増やす必要があります。こ

これらのメッセージがほかの原因で起こった場合は、ユーザ・ガイドのエラー・メッセージに関する付録を参照してください。

そのほかのシステム・コールでも、呼出しに失敗すると、プロセス間通信 (IPC: **interprocess communication**) 構造に調整が必要であることを示すエラー・メッセージを返します。これらのメッセージと調整を必要とするパラメータについては、付録 A, 「IRIX カーネルのチューニング・パラメータ」を参照してください。

timex, sar, par

システム・パフォーマンスを監視するには、timex, sar, および par の3つのユーティリティを使用します。このユーティリティは、システム内のアクティビティに役立つ情報を提供します。

オペレーティング・システムには、システム内のアクティビティを測定する多くのカウンタがあります。操作が行われるたびに、関連するカウンタは増えます。これらのカウンタの値を参照することによって、システム内のアクティビティを監視できます。

timex と sar の各ユーティリティは、オペレーティング・システムのカウンタの値を監視し、システム・パフォーマンスのサンプリングを行います。どちらのユーティリティもデータ収集コマンドである **sadc** を使用します。このコマンドは、オペレーティング・システムのカウンタからデータを収集し、そのデータをバイナリ形式でファイルに格納します。これらのユーティリティは、timex が一定期間のサンプルを収集するのに対し、sar は指定された間隔でサンプルを収集する点が異なります。また、sar プログラムには、CPU 使用率 (**-u** オプション) やページング (**-p** オプション) などの特定の機能のサンプリングを可能にするオプションもあります。これらのユーティリティはデータを多少異なる形で表示します。

par ユーティリティは、システム・コールとスケジューリング・アクティビティをトレースします。単一のプロセス、関連したプロセスのグループ、およびシステム全体のアクティビティをトレースするためにも使用できます。

これらのユーティリティは、次のように使い分けます。つまり、単一のアプリケーションまたはいくつかのプログラムを実行している場合は、timex を使用します。マルチユーザ / マルチプロセス・システムを使用するか、または、多くのプログラムを実行している場合、あるいはその両方の場合、sar または par を使用します。

システム・パフォーマンスの調整のためにユーティリティを実行する場合、必ずアプリケーションやベンチマークを実行しているときに同時に行ってください。そして調整は、値が一定の期間に渡って許容限度を超えた場合にかぎり行ってください。

timex の使用

timex ユーティリティは、単一のアプリケーションを実行している場合に便利なトラブルシューティング用のツールです。次に例を示します。

```
timex -s application
```

-s オプションは、アプリケーションによるアクティビティだけではなく、アプリケーションの実行時に発生したすべてのシステム・アクティビティを報告します。timex の出力をファイルにリダイレクトするには、次のように入力します。なお、この例では Bourne シェル (sh(1)) を使用しています。

```
timex -s application 2> file
```

同じコマンドを C シェルを使用して入力する場合は、次のように入力します。

```
timex -s application > file
```

sar の使用

sar ユーティリティは、多くのプログラムやプロセスを実行している場合、または、サーバなど、マルチユーザ・システムを使用している場合に便利なトラブルシューティング用のツールです。オペレーティング・システムのカウンタ・サンプルリングを一定の期間 (1 日、数日間または 1 週間) に渡って行えます。

必要に応じて、システム・アクティビティを調べる方法を選択できます。システムの監視は次の時期に実施できます。

- 日常操作の間
- 一定の時間間隔をおき連続的に
- ユーザ制御下のアクティビティの前後
- コマンドの実行中

システム・アクティビティのデータ収集とファイルへの格納を `sar` によって自動的に行われるようにシステムを設定できます。`chkconfig` コマンドを使用すると、`sar` の自動報告機能を利用できます (`sar -A` によるリストが作成されます)。`crontab` エントリは、作業時間帯は 20 分ごとに、また、その日のそれ以外の時間帯は 1 時間ごとに、システム・カウンタを監視するようにシステムに指示します (最近の 7 日間のデータが保存されます)。この機能を使用可能にするには、次のように入力します。

```
/etc/chkconfig sar on
```

収集されたデータは `/var/adm/sa` に `sann` と `sarnn` の形式で格納されます (`nn` は報告日、`sarnn` は ASCII 形式)。システム・アクティビティの結果を出力するには、`sar(1)` コマンドを使用します。

時間間隔を指定した `sar` の連続使用方法

`sar` を使用すると、システムの現在の状態に関する連続的な報告を作成できます。コマンド行に、時間間隔とカウント数を指定します。

```
sar -u 5 8
```

このコマンドによって、CPU 使用に関する情報が 5 秒間隔で 8 回表示されます。

ユーザ制御アクティビティの前後における `sar` の使用方法

アプリケーションの実行の前後 (または、複数のアプリケーションを同時に実行した後) にシステム・アクティビティのカウンタのスナップショットを撮ると有用です。システム・アクティビティのスナップショットを撮るには、`sadc` (データ・コレクタ) に指示して出力をファイルにダンプさせます。次に、標準的なシステム負荷または制限された負荷の下でアプリケーションを実行し、収集を停止する準備が整った後で、再度システム・アクティビティのスナップショットを撮ります。次に、これらの結果を比較し、どのように変化しているかを調べます。

次に、アプリケーションの実行の前後にシステム・カウンタのサンプルを収集するコマンド例を示します。

```
/usr/lib/sa/sadc 1 1 file
```

アプリケーションを実行、または監視する作業を実施した後、次のように入力します。

```
/usr/lib/sa/sadc 1 1 file  
sar -f file
```

ファイルがない場合は、sadc によってファイルが作成されます。ファイルがある場合は、sadc はそれにデータを追加します。

コマンド実行中の sar と timex の使用方法

コマンドを実行している間、システム・アクティビティを調べる場合があります。たとえば、nroff(1) の実行中にすべてのシステム・アクティビティを調べるには、次のように入力します。

```
/usr/lib/sa/sadc 1 1 sa.out
nroff -mm file.mm > file.out
/usr/lib/sa/sadc 1 1 sa.out
sar -A -f sa.out
```

timex を使用すると、1 行のコードで同じ操作が行えます。

```
timex -s nroff -mm file.mm > file.out
```

timex も nroff 要求の実行に費やされた実時間、ユーザ時間およびシステム時間を表示します。

timex と sar には 2 つの小さな違いがあります。sar プログラムは出力を制限することができません (たとえば、-u オプションは CPU アクティビティのみを報告)、timex は常に -A リストを表示します。また、sar はさまざまな形式で動作しますが、timex はコマンドを実行することによってのみ動作します。ただし、このコマンドはシェル・ファイルであってもかまいません。

複数のコマンドを並列に実行しているときにシステム・アクティビティを調べる場合は、コマンドをシェル・ファイルに入れて、そのファイルに対して timex -s を実行します。たとえば、ファイル nroff.sh に次の行が入力されているとします。

```
nroff -mm file1.mm > file1.out &
nroff -mm file2.mm > file2.out &
wait
```

同時に実行されている 2 つの nroff 要求が終了した後すべてのシステム・アクティビティの報告を取得するには、次のように入力して timex を起動します。

```
timex -s nroff.sh
```

par の使用

以下のように、par は sar と同様の使い方ができます。

- 日常業務の作業中
- 一定の時間間隔をおいて連続的に
- ユーザ制御下のアクティビティの前後
- コマンドの実行中

par の具体的な使用方法については、par(1) マン・ページを参照してください。

問題のあるプロセスの詳細を参照する場合は、sar の代わりに par を使用します。timex は、プロセス実行中に、使用されている総時間を報告するだけですが、par は情報を分析し、時間を消費しているプロセスに関して具体的に報告します。特に、次のコマンド・オプションが有用です。

-isSSdu 各システム・コールとその間のタイム・ラグに使用された時間をチェックします。

-rQQ プロセスのスケジューリングについてチェックします。プロセスの実行頻度を上げるか、下げるかを確認できます。

システム・コールをトレースすると、par は主体プロセスが生成したすべてのシステム・コールを一連の引数および戻り値とともに報告します。このモードでは、par は主体プロセスに発信したすべてのシグナルも表示します。スケジュール・トレース・モードの場合、par は監視中にシステムで起こったスケジュール・イベントをすべて報告します。プロセスが実行待ち行列に置かれた時点、実行が開始された時点、および実行が中断された時点では、その原因も含めた情報が報告されます。これらのイベントはタイム・スタンプを含んでいます。par によるシステム・アクティビティ・データの収集およびファイルへの取込みを自動化するようにシステムを設定することが可能です。

par ユーティリティは padc(1) マン・ページの出力結果を処理するものです。これには、次の2通りの方法があります。

- padc を独立して実行し、結果をファイルに保存して、このファイルを別のオペレーションとして par に与えます。
- par が padc を呼出すことで、データの収集と報告をワンステップで行います。

par ユーティリティは、指定したレポート・オプションに基づいて所定の padc データ・セットから、異なるタイプのレポートを作成できます。このため、独立したステップとしてデータ収集を実行することが望ましい場合があります。

sar、par、および timex の要約

これまで、par、sar、および timex を使用する場合とその方法について説明しました。オペレーティング・システムを監視するには、これらのユーティリティのいずれかを使用してください。次に出力結果を調べてパフォーマンス低下の原因を検出します。大きく変動している数値や一定の期間で変更している数値を調べます。一時的に最大値を超えている値に関してあまり心配する必要はありません。

まず、ディスク I/O プロセスをシステムがどのように処理しているかを調べてください。次に、過剰なページングやスワッピングがないかを検査します。最後に、CPU 使用率とメモリ割当てを調べます。

以下の説明では、調整対象のシステムがアクティブ状態である（アプリケーションやベンチマークを実行中）と仮定します。

ディスク I/O のパフォーマンス

システムは、データを格納するためにディスクを使用し、ディスクとメモリの間でデータを転送します。この I/O プロセスはシステム・リソースを大幅に消費するので、入出力を行う場合はオペレーティング・システムの効率をできるだけ高める必要があります。

ディスク I/O のチェック

大規模なアプリケーションを実行する場合や、システム負荷が大きい場合は、ディスク I/O を調整し、システム・パフォーマンスを向上させることができます。sar -A または timex -s を実行し、%busy、%rcache、%wcache、%wio の各フィールドを調べてください。ディスク・サブシステムの調整の必要性を確認するには、表 10-4 の数値を参照して sar -A の出力を検査します。この表では、ディスク使用率 (sar -d) や CPU アクティビティ (sar -u) など、選択した出力のみを表示する sar オプションを右の欄に示します。

表 10-4 に、I/O の負荷が高いシステムを示す `sar` の出力結果を示します。

表 10-4 I/O の負荷が高いシステムを示す出力結果

フィールド	値	sar オプション
%busy (ディスク・ビジー時間率)	>85%	<code>sar -d</code>
%rache (バッファ・キャッシュでの読取り)	low, <85	<code>sar -b</code>
%wcache (バッファ・キャッシュでの書込み)	low, <60%	<code>sar -b</code>
%wio (ディスク I/O 待ちアイドル CPU)	dev. system >30 fileserver >80	<code>sar -u</code>

CPU がディスク I/O 待ちをしていてアイドル状態にある時間の割合を示す %wio の数値については、次の 2 つのシステムでその解釈が異なるので注意してください。

- `make` などが実行される開発用のシステム。この場合、%wio > 30 の場合は、%wio (`sar -u`) の内訳を調べます。%wfs (ファイルシステム待ち) と %wswp (スワップ待ち) を調べると、システムが何を待っているかを正確に把握できます。
- NFS クライアントにサービスを提供し、ファイル・サーバとして動作する NFS システム。このとき、%wio > 80、%wfs > 90 の場合は、システムはディスク I/O の制限を受けます。

I/O パフォーマンスを最大限に向上させるために考慮する要因はほかにも多くあります。次の方法でもパフォーマンスを向上できます。

- 論理ボリュームの使用
- 異なるディスクのパーティションの使用
- ハードウェア (ディスク、コントローラ、メモリ) の追加

論理ボリュームを使用したディスク I/O の改善

論理ボリュームを使用すると、ディスク I/O を改善できます。

- 既存のファイルシステムの内容を損なわずに、既存のファイルシステムのサイズを拡張できます。
- ファイルシステムを複数のディスクにストライプ化できます。別のディスクにストライプ・ボリュームを作成すると、I/O スループットを 50 % まで改善できます。

ストライプ化の効果が最大限に発揮されるのは、ディスクが別のコントローラにある場合です。論理ボリュームを使用すると、最初のファイルシステムを再度作成せずに、より多くの領域を取得できます。ディスクのストライプ化では、より多くの領域とパフォーマンスの向上が得られますが、同時に、ストライプ・データの入ったディスクの 1 つを失うと、ファイルシステムに格納されているすべてのデータを損失する危険性もあります。これは、データがすべてのディスクに分散しているからです。

連続的な論理ボリュームは 1 つのディスクへのデータの格納が終了すると、次のディスクに書き込みを始めます。ストライプ化された論理ボリュームは 2 つのディスクに均等に書き込み、各ファイルをボリューム内のすべてのディスクに展開します。データがストライプ化されている場合は不良ディスクが 1 つでもあるとリストアできませんが、データが連続的な論理ボリュームに格納されているとリストアできます。ストライプ化されたディスク・ボリュームの作成方法の詳細については、『IRIX Admin: Disks and Filesystems』を参照してください。

パーティションと追加ディスクを使用したディスク I/O の改善

システムのスループットを増大させる確実な手段があります。たとえば、ピーク時に実行できるプログラムの数を制限したり、プロセスを非ピーク時に移したり（夜間のバッチ・ジョブの実行など）、プロセスを別のマシンに移したりする方法です。また、パーティションを別々のディスクに設定して、ディスク負荷を再分散したり、ディスクを追加することもできます。

パーティションの説明をする前に、プログラム実行時にディスクがどのように使用されているかを説明します。表 10-5 に、アプリケーションによるディスクへのアクセスの理由を示します。

表 10-5 アプリケーションのディスク・アクセス

アプリケーション	ディスク・アクセス
オブジェクト・コードの実行	テキストとデータ
スワップ領域のデータ、スタックへの使用	/dev/swap
テンポラリ・ファイルの書き込み	/tmp と /var/tmp
データ・ファイルの読み込み / 書き込み	データ・ファイル

上記に示したディスク・アクセスの一部の領域に対し、異なるディスク上に別々のパーティションを設定すると、I/O パフォーマンスを最大化できます。これは、アプリケーションのディスク・アクセス・ルーチンを拡張することに相当し、この結果、入出力の高速化を図ることができます。

デフォルトでは、ディスク・パーティションは次のどちらかに設定されます。

- 2つのパーティション：パーティション0および1
- 1つの大きなパーティション：パーティション7（小さなパーティション2つ分に相当する）

システム・ディスクでは、パーティション0はルート用、パーティション1はスワップ用です。

メモ：旧バージョンのシステムでは、パーティション0、1、および6の3つのパーティションに分かれているディスクもあります。システムディスクでは、パーティション0はルート用、1はスワップ用、6は /usr 用です。大きいパーティションが1つだけの場合は、3つの小さなパーティション分に相当します。

ディスクを1つ追加するたびに、パーティションを1つにするか、いくつかに分けるかを決定し、また各ディスクとパーティションにはどのファイルシステム（またはスワップ）を割当ててかを決定しなければなりません。ファイルシステムをディスク・パーティションに分散させて、異なるディスクに同時にアクセスできるようにすることをお勧めします。

システムの使用方法によって構成が異なります。次に、いくつかの例を示します。

- データ・ファイルからの読取りを頻繁に行うグラフィックス・アプリケーションを実行するシステムを考えます。このアプリケーションは非常に大きいので、ページがスワップ・パーティションに頻繁にスワップアウトされます。
この場合は、スワップ領域とは別のディスクに、アプリケーションのデータ・ファイルを配置すると効果的です。
- このようにシステムを構成した後、スワップ領域が不足している場合は、メモリを増やすか、すべてを別のハード・ディスクにバックアップし、スワップ領域とデータ領域の両方を含むパーティションを作成することを検討します。
- 既存のファイルシステムを格納しているパーティションのサイズを変更すると、そのファイルシステムに格納されているデータにアクセスできなくなることがあります。変更を行う前には必ず現在のフル・バックアップを作成し、また、パーティション情報を記録します。間違ったパーティションを変更しても、そのパーティションで `mkfs` を実行したり上書きしないかぎり、元に戻すことができます。ディスクをカスタマイズした後は、重大なディスク破損が発生しても簡単にリストアできるように、`prtvtoc` コマンド出力を印刷しておくことをお勧めします。

1 つの非常に大きなアプリケーションと 3 つのディスクがある場合は、2 番目のディスクはアプリケーションの実行可能プログラム (/bin および /usr/bin) 用に、3 番目のディスクはデータ・ファイル用に、それぞれディスクにパーティションを設定し、使用します。次に、コンパイル・エンジンとして動作させることが多いシステムについて考えます。

この場合、/tmp ディレクトリは、コンパイルされるソース・コードとは別のディスクに配置する方法をお勧めします。別のディスクにファイルを作成する前に、必ずファイルシステムをチェックしてからマウントしてください。この操作が行えない場合は、別のディスクにあるテンポラリ・ファイル用のディレクトリを使用するように、コンパイラに指示できます。それには、テンポラリ・ファイル用の新しいディレクトリに TMPDIR 環境変数を設定するだけで済みます。次に多くのプログラムを同時に実行し、多くのスワッピングを行うシステムについて説明します。

この場合、スワップ領域を別のディスクの複数のパーティションに分散させることをお勧めします。

ディスク・ハードウェアの追加によるディスク I/O の改善

システムの調整を行っても I/O パフォーマンスが向上しない場合は、ディスク、コントローラ、メモリなどのハードウェアの追加を考慮することも必要です。

システムにハードウェアを追加する場合のディスクまたはコントローラの種類については、ハードウェアに付属のユーザ・ガイドを読直してシステム仕様を検討し、現在サポートされているディスクまたはコントローラのハードウェア仕様を比較してください。この情報に基づいて、特定のニーズに合ったディスクまたはコントローラを選択してください。

コントローラやディスクを使用し、最もアクティブなファイル・システムをバランス良く構成すると、ディスク・アクセスを高速化することができます。

ディスクに対する読取りと書込みの数を減らすもう 1 つの方法は、メモリの追加です。メモリを追加すると、スワッピングとページングを減らすことができます。

ページングとスワッピング

CPU はメイン・メモリ (RAM) にロードされているデータの参照とコードの実行しか行いません。また、CPU は複数のプロセスを実行するので、すべてのプロセス用に十分なメモリがない場合があります。非常に大きなプログラムを動作させる場合は、システム内の物理的なメモリ以上の容

量が必要になります。この場合、プロセスはページ単位でメモリに読込まれます。つまり、十分なメモリが存在しないと、オペレーティング・システムは二次メモリ領域であるスワップ領域にページを一時的に書込んで、メモリを解放します。

IRIX システムは、一度にはメモリに収まりきれないほど多くのプロセスを実行させることができます。各プロセスは、アドレス空間と呼ばれるメモリの仮想セクションを与えられます。仮想セクションには、理論的に、プロセス全体を格納するために十分な大きさがあります。ただし、アドレス空間のうち、現在使用中のページのみが実メモリ上に置かれます。これらのページは、ワーキング・セットと呼ばれます。プロセスの実行のとき、データやコードの新しいページを必要とする場合は、必要とするページがメイン・メモリに読込まれます（ページのフォールト・イン、またはページ・フォールトと呼ばれます）。最近使用されていないページがある場合は、オペレーティング・システムはそのページをメイン・メモリからページをスワップ領域に移し、フォールト・インされた新しいページ用の空き領域を作ります。書出されたページは、後で再びフォールトされることもあります。このプロセスは、ページングと呼ばれます。スワッピングと混同しないよう注意してください。

スワッピングとは、アクティブでないプロセスのすべてのページ領域をメモリから削除し、アクティブなプロセス用に空き領域を作ることです。プロセス全体がディスク上のスワップ領域に書出され、その実行は停止されます。アクティブでないプロセスが再びアクティブになった場合、そのページは実行可能になる前に、ディスクからメモリに回復される必要があります。これはスワップ・インと呼ばれます。パーソナル・ワークステーションでは、スワップ・インは、ディスク・アクティビティにおいてよく知られています。アクティブでないアプリケーションのアイコンをクリックしてから、そのウィンドウが表示されるまでの遅れがスワップ・インの時間です。

過剰なページングとスワッピングのチェック

IRIX システムが多くのプロセスを多重処理しているとき、このスワッピングとページングのアクティビティによって、システムのパフォーマンスが左右されます。sar を使用してこの状況を調べ、ほかのツールでこれを処理することができます。

システムが、ページングやスワッピングによって負荷がかかりすぎているかどうかを判断するには、基準値がどこにあるか認識する必要があります。特定の実行状況での基準値を決定するには、いろいろな状況下でコマンドを使用してみる必要があります。たとえばシステムを起動し、非常に制限された数のプロセスを実行させながら、さまざまなパターンの基準値テストを実行します。さらに、軽い負荷の場合、重いネットワーク・アクティビティの場合、および特に負荷がかかっ

ている場合にパフォーマンスが低下していることを認識できます。システム・ログ・ブックに記録しておくことと基準値を決定する際の手がかりとなります。

表 10-6 に、小さなシステム構成における過剰なスワッピング / ページングの有無を検査するためのフィールドを示します。

表 10-6 過剰なスワッピングとページングの有無を検査するためのフィールド

重要なフィールド	sar オプション
vflt/s - page faults (有効なページがメモリにありません)	sar -p
bswot/s (メモリからディスク・スワップ領域に転送します)	sar -w
bswin/s (メモリに転送します)	sar -w
%swpocc (タイム・スワップ待ち行列が占有されています)	sar -q
rflt/s (ページ参照の障害)	sar -t
freemem (ユーザ・プロセスの平均ページ数)	sar -r

システムのパフォーマンスが低下しているとき、スワップ I/O やほかの要因に関連しているかどうか決定する場合に、次の sar オプションを使用してください。

- u %wswp スワップ I/O による CPU 待ち時間。このオプションは、アクティブなプロセスが、読取りまたは書込みされるページを待ってブロックされている時間の割合を計測します。wio の値も大きくない限り、この値は特に意味がありません。
- p vflt/s プロセスが、メモリにないページにアクセスする頻度。この数値をパフォーマンスのよい場合と悪い場合とで比較します。パフォーマンスの低下が *vflt/s* の急増とともに起こった場合は、*%vswp* が低い値、または 0 であってもスワップ I/O に問題があると考えられます。
- r freemem 未使用のメモリ・ページ。ページング・デーモン (vhand) は、未使用であることを判断するページを回復し、それらをこのプールに戻します。プロセスが新しいページを必要とすると、このプールからページを持ってきます。プールが少ないか空の場合、IRIX システムはほかのプロセスからページを取って、あるプロセス用にページを確保しなければなりません。これは、ページ・フォールトを助長します。

-p pgswp/s 1 秒間にスワップ・ディスクから読取りまたは書込み可能なデータ・ページの数。

-p pgfil/s 1 秒間にディスクから取得できる読取り専用のコード・ページの数。

たとえば、`%vswp` の数が 0 か非常に低い値で、パフォーマンスの悪化に伴って `vflt/s` が増加しない場合、パフォーマンスの問題はスワップ I/O とは関係ありません。

スワップ I/O 問題の対処方法

スワップ I/O に原因があると考えられる場合は、次のような対処方法があります。

- 実メモリを増やします。この方法は特にパーソナル・ワークステーションでは効果的です。相対的に考えると使用できる実メモリを増加するほうが経済的だからです。
- 実行するプロセスの数を減らして、メモリの要求を減らします。システム・ロードの負荷が、対話型ではなく、バッチ・プログラムや長時間に渡って実行されるコマンドからなる場合に有効です。要求度の低いコマンドに対しては、`cron` や `at` を使用して指示を出します。順番に実行した低いスワップ I/O と同時に実行した高いスワップ I/O で、どちらの方が各プログラム・セットの総実行時間が少ないかを調べてください。
- 読取り専用のページのスワップ入力をより効率的にします。たとえば、動的共有オブジェクトのページが、NFS マウント・ドライブから遅いネットワークを介してロードされる場合、動的共有オブジェクトの一部またはすべてをローカルなディスクに移動することで、ページ入力をより速くすることができます。
- 書込み可能なページのスワップ I/O をより効率的にします。たとえば、`swap(1M)` を使用し、スワップ・アクティビティをいくつかのディスクやパーティションに拡大します。ファイルへのスワッピングや、新しいスワップ領域の生成については、148 ページの「スワップ領域」を参照してください。
- プロセスや、`nproc` など、CPU に関するカーネル・パラメータを変更した場合は、それらを前の値に戻してください。
- ページ・フォールトを減らします。ローカル性を考慮し、プログラムを作成します。付録 C、「アプリケーションの調整」を参照してください。
- アプリケーションを構築する場合は、共有ライブラリの使用を検討します。
- `systune` によって、常駐セット・サイズの上限を減らします。適用するパラメータの名称と特性については、“システム・リミット・パラメータ” in 付録 A を参照してください。

ページ・サイズの動的調整については、229 ページの「複数のページ・サイズ」を参照してください。

CPU アクティビティとメモリの割当て

パフォーマンスに問題がある場合は、ディスク I/O とページングについて調べてから、CPU アクティビティとメモリ割当てを確認します。

CPU のチェック

CPU は特定の瞬間には 1 プロセスしか実行できません。CPU の負荷が高くなると、プロセスは実行されずに待機状態になります。CPU の速度を変更することはできませんが、高速の CPU にアップグレードしたり、ハードウェアの許容範囲内でシステムに CPU ボードを追加することは可能です。また、CPU 負荷を監視し、負荷を分散させることも可能です。表 10-7 に、CPU の負荷が高いシステムかどうかを検査するためのフィールドを示します。

表 10-7 CPU の負荷が高いシステムの指標

フィールド	値	sar オプション
%idle (CPU 待機時間率)	<5	sar -u
runq-sz (CPU 待ちのメモリのプロセス)	>2	sar -q
%runocc (占有実行待ち行列および未実行プロセスの割合)	>90	sar -q

top(1) コマンドまたは gr_top(1) コマンドを使用し、最高の CPU 使用量を占めるプロセスを表示できます。これらのコマンド出力では、各プロセスについて、ユーザ、プロセス状態フラグ、プロセス ID とグループ ID、使用された CPU サイクル、現在プロセスを実行しているプロセッサ、プロセス優先順位、プロセス・サイズ (ページ数)、常駐セット・サイズ (ページ数)、プロセスが使用した時間、プロセス名が表示されます。詳細については、top(1) または gr_top(1) マン・ページを参照してください。

CPU パフォーマンスの向上

CPU パフォーマンスを向上させるには、次のように変更します。

- 非ピーク時またはほかのマシンへのジョブのオフロード、効率の良い経路の指定、アプリケーションの調整を実施します。
- ポーリング・ループを削除します。select(2) マン・ページを参照してください。
- slice-size パラメータ（プロセスのタイム・スライスの長さ）の値を増やします。たとえば、slice-size を Hz/30 から Hz/10 に変更します。ただし、対話の応答時間が遅くなることがあるので注意してください。
- より高速の CPU にアップグレードするか、別の CPU を追加します。

使用可能メモリのチェック

216 ページの「ページングとスワッピング」では、プロセスのための十分な物理メモリ（メイン・メモリ）がない場合の対処方法について説明しました。ここでは、物理メモリと論理スワップ領域を含む使用可能なメモリ（仮想メモリ）が不足している場合に起こる問題について説明します。

IRIX の仮想メモリ・サブシステムは、物理メモリを超える大きなプログラムを正常に実行させるための機構です。このサブシステムを使用すると、複数のプログラムの合計所要メモリが物理メモリを上回る場合でも、それらのプログラムを実行させることができます。これは、過剰なデータをスワップ・デバイスに格納して実現されます。

スワップ領域の割当ては、プログラム実行の開始後に行われます。このため、大きな仮想アドレスを持つプログラムは、割当てられた仮想メモリの実際量がマシンのメモリとスワップ・リソースを超えないかぎり実行可能です。

通常、次のようなメッセージがコンソールに出力されるので、メモリをいつ使い果たしたかを明示的に知ることができます。

```
Out of logical swap space...
```

次の場合にこのメッセージが表示されます。

- プロセスが ENOMEM または UMEM を超過した場合。
- 必要なページング不可のデータ構造をカーネルが保持できるだけの十分な物理メモリがない場合。
- 論理スワップ領域が不足した場合。

仮想スワップ領域は任意にシステムに追加できます。スワップ領域の追加方法については、148 ページの「スワップ領域」を参照してください。ただし、次のメッセージが出力される場合は、物理スワップ領域を追加しなければなりません。

```
Process killed due to insufficient memory
```

システム・コール `exec`、`fork`、`brk`、`sbrk` (`malloc` により呼出される)、`mpin` および `plock` は、使用可能なメモリが不足している場合に `EAGAIN` を返します。アプリケーションは戻り値をチェックし、有用なメッセージを出力して安全に終了してください。

実行されているプロセスのサイズ (ページ数) をチェックするには、`ps -el` を実行します (`top` を使用することもできます)。`SZ:RSS` フィールドを調べることで非常に大きなプロセスを見つけることができます。

このフィールドを調べると、プロセスが使用しているメモリの容量を確認できます。非常に大きなプロセスは非ピーク時に実行することをお勧めします。

システム・メモリの容量の確認

メイン・メモリの容量を確認するには、`hinv(1M)` コマンドを使用します。このコマンドを実行すると、システム構成に関するデータが表示されます。

```
Main memory size: 64 Mb
```

メモリの極大化

仮想メモリの容量を増やすには、実メモリまたはスワップ領域の容量を増やします。ページングまたはスワッピングの問題解決は、使用可能なメモリの節約にもつながります。次にこの解決方法を示します。

- プログラムの数を制限します。
- 共有ライブラリを使用します。
- メモリを追加します。
- システム・テーブルのサイズを低減します。

ただし、仮想メモリの容量を増やす最も効果的な方法は、スワップ領域を追加することです。詳細については、148 ページの「スワップ領域」を参照してください。

オペレーティング・システムの調整

オペレーティング・システムの調整は困難ではありませんが、十分注意して実施する必要があります。後で変更を取消す場合に備えて、作業内容を詳しく記録しておいてください。作業の内容を理解した上で実施し、奇跡的な結果は期待しないでください。IRIX システムは、特殊な場合を除き、最適なパフォーマンスを実現できるように設計されています。大量のグラフィックス操作やデータ処理を伴うソフトウェアはシステムにかなりの負荷がかかるので、強靱なシステムにおいても大きな速度の低下をもたらす場合があります。どのような調整を行っても、このような状況を変えることはできません。

オペレーティング・システムの調整手順

システムを調整する場合は、まず、各種のシステム・ユーティリティ（206 ページの「オペレーティング・システムの監視」を参照）を使用し、システム・パフォーマンスを監視します。手順 10-2 で、システムを調整する方法を説明します。

手順 10-2 システムの調整

1. 調整を必要とする一般的な領域（たとえば、ディスクの I/O、CPU）を決定し、`sar` や `osview` などのユーティリティを使用し、システム・パフォーマンスを監視します。監視方法については、206 ページの「オペレーティング・システムの監視」を参照してください。
2. 特定領域に焦点をあて、一定期間に渡りパフォーマンスを監視します。長い期間に渡って大きな変動や変化を示す数値を検索してください。数値が最大値を超えても心配する必要はありません。
3. 調整の効果を確認するために、一度に変更する値や特性は 1 つにしてください（1 つのパラメータを変更する、1 つのコントローラを追加するなど）。どのような変更でもシステム・ノートブックに記録するように習慣付けてください。
4. `systemd` コマンドを使用し、パラメータ値を変更します。変数が、`systemd` を使用して調整できない場合は、`master.d` ディレクトリの構造で変更します。必要に応じて、カーネルを再度構築し、再起動します。
5. パフォーマンスを再測定して、前後の結果を比較します。次に、システム・パフォーマンスの改良結果を評価し、さらに変更が必要であるかどうかを決定します。

調整手順は計算ずくの科学というよりは経験を重ねる手工芸に近いと言えます。つまり、システムを微調整するには、必要に応じて上記の手順を何度も繰返さなければならない場合もあります。また、完ぺきに微調整を行う場合は、監視やテストの範囲を広げる必要があります。

オペレーティング・システムの調整 : パラメータの値の確認

システムを調整する前に、まずチューニング・パラメータの現在値を確認しておく必要があります。カーネル・パラメータの現在値を確認するには、`systune` コマンドを使用します。このコマンドを引数なしで実行すると、システムのすべてのチューニング・パラメータの現在値が表示されます。このコマンドについての詳細は、`systune(1M)` マン・ページを参照してください。

オペレーティング・システムの調整 : パラメータの変更とシステムの再構築

調整するパラメータを変更した後、その変更を有効にするために、システムを再構築する必要があります。場合によっては、`systune` ユーティリティは、パラメータが変更されると、その変更を有効にするためにシステムを再起動する必要があるかどうかを知らせます。手順 10-3 で、システムを再構築する方法を説明します。

手順 10-3 システムの再構築

1. システムをバックアップします。
2. 変更前のカーネルを `unix.save` にコピーします。
3. 変更を行います。
4. 必要に応じてシステムを再起動します。

システムのバックアップ

カーネル・パラメータを変更してシステムを再構築する前に、現在のシステムのフル・バックアップを作成してください。バックアップの手順については、『IRIX Admin: Backup, Security, and Accounting』を参照してください。

注意 : 調整を行う前に、必ずシステム全体のバックアップ (フル・バックアップ) を作成してください。

カーネルのコピー

変更を必要とするパラメータを決定した後（ユーザ数が多い場合は *nproc* の値を増やすなど）、まず次のコマンドを入力し、システムとカーネルをバックアップします。

```
cp /unix /unix.save
```

このコマンドは、現在のカーネルのコピーを作成します。これ以降の例の説明では、これを保存済みの旧カーネルと呼びます。このようなコピーを作成しておくこと、パフォーマンスの調整結果に満足できなかった場合に、元のカーネルに戻すことができます。

パラメータの変更

システムのバックアップが完了すると、*systune* コマンドを実行できます。*systune* への新しい値は、16 進数表記でも 10 進数表記でも表現できます。どちらの値も *systune* によって出力されます。

たとえば、*nproc* の値を増やすために *systune* を実行するには、次のように入力します。

```
systune -i
Updates will be made to running system and /unix.install
systune-> nproc
           nproc = 400 (0x190)
systune-> nproc = 500
           nproc = 400 (0x190)
           Do you really want to change nproc to 500 (0x1f4)? (y/n) y
In order for the change in parameter nproc to become effective
/unix.install must be moved to /unix and the system rebooted
systune-> quit
```

引き続き、システムを再起動します。また、パラメータに対する変更は、必ずシステム・ログ・ブックに記録してください。

注意: `reboot` コマンドを実行すると、システムは現在のカーネル (`/unix`) に作成したばかりのカーネル (`/unix.install`) を上書きします。したがって、再起動する前に、必ず現在のカーネルを安全な場所にコピーしてください。

autoconfig を使用した新しいカーネルの作成と起動

`system` コマンドは新しいカーネルを自動的に作成しますが、`system` を使用せずにパラメータを変更した場合、または新しいシステム・ハードウェア (マルチプロセッサ・システムの CPU ボードなど) を追加した場合は、`autoconfig` を実行して新しいカーネルを生成する必要があります。

`autoconfig` コマンドは、いくつかの環境変数を使用します。環境変数についての詳細は、`autoconfig(1M)` マン・ページを参照してください。以下の変数のいずれかがある場合は、`autoconfig` を実行する前にそれらの設定を解除してください。

- UNIX
- SYSGEN
- BOOTAREA
- SYSTEM
- MASTERD
- STUNEFILE
- MTUNEDIR
- WORKDIR

システムの再構築後に新しいカーネルを作成するには手順 10-4 に従ってください。

手順 10-4 新しいカーネルの作成

1. 次のコマンドを実行し、特権ユーザになります。

```
su
```

2. 次のコマンドを入力し、現在のカーネルのコピーを作成します。

```
cp /unix /unix.save
```

3. 次のコマンドを実行します。

```
/etc/autoconfig -f
```

このコマンドは、新しいカーネルを作成し、/unix.install ファイルに格納します。

4. 次のコマンドを入力し、システムを再起動します。

```
reboot
```

注意：reboot コマンドを実行すると、作成したばかりのカーネル (/unix.install) が現在のカーネル (/unix) に上書きされます。したがって、再起動する前に、必ず現在のカーネルを安全な場所にコピーしてください。

システムの起動時には、/etc/rc2.d/S95autoconfig にある自動構成スクリプトが実行されます。このスクリプトは、次のような状況の場合に新しいカーネルを作成するかどうかを質問します。

- 現在のカーネルに専用のドライバがない新しいボードがインストールされた場合。
- /var/sysgen/mtune のオブジェクト・ファイル、/var/sysgen/master.d のマスター・ファイル、または /var/sysgen/system のシステム・ファイルが変更された場合。ファイルが変更されたかどうかは、これらのファイルとカーネルの変更日付から判別されます。

このどちらかの状況が当てはまる場合、システムの起動時に、オペレーティング・システムの再構築を求めるプロンプトが出力されます。

```
Automatically reconfigure the operating system? y
```

プロンプトに対して **y** と応答すると、スクリプトが lboot を実行し、新しいイメージの /unix.install を生成します。この自動構成スクリプトが実行されないようにするには、/etc/rc2.d/S95autoconfig を /etc/rc2.d/wasS95autoconfig のように **S** 以外の文字で始まる名前に変更します。

起動できないカーネルからの回復

手順 10-5 で、起動できない /unix からの回復方法と、再構築の試行が失敗した後に有効なソフトウェアを実行する方法について説明します。systune ユーティリティを使用する場合は、ここで説明する情報を適用しないでください。systune ではパラメータを無効な値に設定できません。

手順 10-5 起動できないカーネルからの回復

1. システムの再起動に失敗した場合は、さらに何度か再起動を試みます。それでも失敗した場合は、ブート・プロセスを中断し、保存済みの旧カーネル (unix.save) から起動するようにブート PROM に指示します。

2. 「リセット (reset)」ボタンを押します。「System Maintenance」メニューが表示されます。

```
System Maintenance Menu
```

- 1) Start System.
- 2) Install System Software.
- 3) Run Diagnostics.
- 4) Recover System.
- 5) Enter Command Monitor.

3. オプション 5 を選択し、コマンド・モニタを起動します。次のように表示されます。

```
Command Monitor. Type "exit" or click on "done" to return to the menu.  
>>
```

4. >> プロンプトに対して、保存済みの旧カーネルを起動するように PROM に指示します。次のコマンドを実行します。

```
boot unix.save
```

システムは保存済みの旧カーネルを起動します。

5. システムが動作すると、次のコマンドを実行し、保存済みの旧カーネルを /unix と入れ替えます。保存済みの旧カーネルはそのまま unix.save に残ります。

```
cp /unix.save /unix
```

上記の手順によって、システムを通常どおりに起動できるので、新しいカーネルを使用し、問題点を調査します。カーネルを起動できなくなった原因を調査してください。特に、カーネルに対して行った変更内容を再検討してください。

- パラメータ値を大幅に増加または減少していませんか。変更する割合は小さくしてください。
- 複数のパラメータを変更していませんか。パラメータの変更は、1 回につき 1 つにしてください。

複数のページ・サイズ

オペレーティング・システムは複数のページ・サイズをサポートします。このページ・サイズは、以下で説明する方法で調整できます。

推奨ページ・サイズ

サポートされるページ・サイズは、システムのベース・ページ・サイズに基づきます。ベース・ページ・サイズは、`getpagesize()` システム・コールで取得できます。IRIX システムは現在、16K と 4K の 2 種類のベース・ページ・サイズをサポートしています。ベース・ページ・サイズが 16K のシステムの場合、16K、64K、256K、1M、4M、16M というページ・サイズがサポートされます。一方、ベース・ページ・サイズが 4K のシステムの場合、4K、16K、256K、1M、4M、16M というページ・サイズがサポートされます。一般に、大半のアプリケーションでは 4K、16K、64K というページ・サイズで `tlbmiss` オーバーヘッドを十分に抑えることができます。

合体のチューニング・パラメータ

IRIX カーネルは、特定のページ・サイズがシステムの空きメモリ総容量に占めるパーセンテージを一定に保つようになっています。このため、隣接ページのグループを定期的に合体して、大きなページを形成します。以下に示すチューニング・パラメータは、特定のページ・サイズの空きページ数の上限を指定します。大きなページを必要としないシステムは、これらのパラメータを 0 に設定できます。チューニング・パラメータは次のとおりです。

- `percent_totalmem_16k_pages`
- `percent_totalmem_64k_pages`
- `percent_totalmem_256k_pages`
- `percent_totalmem_1m_pages`
- `percent_totalmem_4m_pages`

- `percent_totalmem_16m_pages`

パラメータは、特定のページ・サイズのページ数の上限値として使用される、メモリ総容量のパーセンテージを指定します。たとえば、`percent_totalmem_64k_pages` を 20 に設定すると、合体メカニズムは、64K の空きページ数を、システムのメモリ総容量の 20 % に制限します。上記のパラメータは、実行時に動的に調整できます。ただし、稼働中のシステムで実行時に 1MB 以上のきわめて大きなページを動的に合体することは困難なので、注意してください。このような場合は、システムの起動時にチューニング・パラメータを設定することをお勧めします。上記のチューニング・パラメータを大きな値に設定すると、大量の合体アクティビティが行われます。システムのメモリが少なくなった場合、大きなページは必要に応じて小さなページに分割されます。

大きなページの確保

実行時に 1MB を超えるきわめて大きなページを合体することは、物理メモリの断片化が原因で困難です。このようなページを必要とするアプリケーションは、起動時にチューニング・パラメータを設定することで大きなページを確保できます。パラメータは、ページ数として指定されます。チューニング・パラメータは次のとおりです。

- `nlpages_64k`
- `nlpages_256k`
- `nlpages_1m`
- `nlpages_4m`
- `nlpages_16m`

たとえば、`nlpages_4m` を 4 に設定すると、起動時に 4 つの 4 MB ページが確保されます。システムのメモリが少なくなった場合、確保されたページが小さなページに分割され、ほかのアプリケーションがこれを使用します。`osview` コマンドを使用すると、特定のページ・サイズの空きページ数を表示できます。詳細については、`osview(1)` マン・ページを参照してください。上記のどのパラメータも、デフォルト値は 0 です。詳細については、264 ページの「`nlpages_64k`」を参照してください。

IRIX カーネルのチューニング・パラメータ

付録 A では、カーネルの構造を定義する調整可能なパラメータについて説明します。システムは、プロセス、ファイル、およびシステムのアクティビティを監視します。パラメータ値の多くは、`/var/sysgen/mtune` および `/var/sysgen/master.d` 内のファイルで指定されています。

システムがパラメータの変更に対して正常に応答しない場合は、元の設定に戻すか、または関連するパラメータを変更します。システムが正常に応答しない場合とは、たとえばシステムの速度や能力が指定したとおりに動作しないという程度の状況から、システムの起動や実行ができなくなる状況まで考えられます。後者は、パラメータを大幅に変更した場合に発生します。関連するパラメータを考慮しないで特定のパラメータだけを最大化すると、システムのバランスが損なわれ、操作が不可能になる場合があります。オペレーティング・システムの正しい調整方法については、第 10 章「システム・パフォーマンスの調整」を参照してください。

付録 A では、重要なチューニング・パラメータを機能別に説明します。関連するパラメータは、次のようにまとめて説明します。

- 一般的なパラメータ。233 ページの「一般的なパラメータ」を参照してください。
- システム・リミット・チューニング・パラメータ。242 ページの「システム・リミット・パラメータ」を参照してください。
- リソース・リミット・チューニング・パラメータ。245 ページの「リソース・リミット・パラメータ」を参照してください。
- ページング・チューニング・パラメータ。255 ページの「ページング・パラメータ」を参照してください。
- IPC チューニング・パラメータ (プロセス間通信メッセージやセマフォ、共有メモリを含む)。265 ページの「プロセス間通信 (IPC: interprocess communication) パラメータ」、266 ページの「プロセス間通信 (IPC: interprocess communication) メッセージ・パラメータ」、269 ページの「プロセス間通信 (IPC: interprocess communication) セマフォ・パラメータ」、271

ページの「プロセス間通信 (IPC: interprocess communication) 共有メモリ・パラメータ」を参照してください。

- ストリーム・チューニング・パラメータ。274 ページの「ストリーム・パラメータ」を参照してください。
- シグナル・パラメータ。276 ページの「シグナル・パラメータ」を参照してください。
- ディスパッチ・パラメータ。276 ページの「ディスパッチ・パラメータ」を参照してください。
- EFS (Extent File System) パラメータ。278 ページの「ファイルシステム・パラメータ」を参照してください。
- ロード可能ドライバ・パラメータ。281 ページの「ロード可能ドライバ・パラメータ」を参照してください。
- CPU 動作パラメータ。283 ページの「CPU 動作パラメータ」を参照してください。
- スイッチ・パラメータ。283 ページの「スイッチ・パラメータ」を参照してください。
- タイマ・パラメータ。292 ページの「タイマ・パラメータ」を参照してください。
- NFS (Network File System) パラメータ。294 ページの「ネットワーク・ファイル・システム (NFS: Network File System) パラメータ」を参照してください。
- ソケット・パラメータ。297 ページの「ソケット・パラメータ」を参照してください。
- Indy Video パラメータ。304 ページの「VINO パラメータ」を参照してください。
- ラージ・ページ・パラメータ。305 ページの「ラージ・ページ・パラメータ」を参照してください。
- 拡張アカウンティング・パラメータ。306 ページの「拡張アカウンティング・パラメータ」を参照してください。
- NUMA パラメータ。308 ページの「NUMA パラメータ」を参照してください。
- ページ複製パラメータ。317 ページの「ページ複製パラメータ」を参照してください。
- 移行メモリ・キュー・パラメータ。318 ページの「移行メモリ・キュー・パラメータ」を参照してください。
- XVM パラメータ。319 ページの「XVM パラメータ」を参照してください。
- CXFS パラメータ。320 ページの「CXFS パラメータ」を参照してください。
- コア・ファイル・パラメータ。321 ページの「コア・ファイル・パラメータ」を参照してください。

各節では、まず、パラメータによって制御されるアクティビティについて簡単に説明します。次に、個々のパラメータを以下の項目に分けて説明します。

名前	パラメータ名を指定します。
説明	パラメータを指定するファイルを含むパラメータ、またはパラメータ式について説明します。
値	デフォルト値、必要な場合はその範囲を指定します。通常、各パラメータ値はシングル・ユーザ・グラフィックス・ワークステーション用の値です。値は、8進数あるいは16進数で表します。
変更要件	パラメータを変更するための適切な条件について説明します。
注記	エラーメッセージなど、ほかの関連情報について説明します。

チューニング・パラメータは、システムのリリースに伴って項目が変更する場合があります。

一般的なパラメータ

次のチューニング・パラメータは、さまざまなシステム構造のサイズを指定します。これらのパラメータは、システムを調整する場合に変更します。

- `cachefs_readahead` - ファイルシステム用の先読み（リードアヘッド）ブロックの数を指定します。
- `cachefs_max_threads` - キャッシュ・ファイル・システムのマウントごとの非同期 I/O デーモンの最大数を指定します。
- `nbuf` - ファイルに対するシステム・バッファ・キャッシュ内のバッファ・ヘッダ数を指定します。
- `min_bufmem` - システムがメモリ不足の状態になった場合に、バッファ・キャッシュにキャッシュされたファイルシステム・メタデータによって保持されるメモリの最低量を指定します。
- `callout_himark` - コールアウトの最大数を指定します。
- `ncallout` - コールアウトの初期数を指定します。
- `reserve_ncallout` - 予約コールアウト数を指定します。

- `ncsize` - ネーム・キャッシュ・サイズを指定します。
- `ndquot` - ディスク割当てシステムによって使用されます。
- `nproc` - 指定された時点で使用可能なユーザ・プロセス数を指定します。
- `maxpmem` - 物理的な最大メモリ・アドレスを指定します。
- `syssegsz` - 動的システム・メモリの最大ページ数を指定します。
- `maxdmasz` - 最大 DMA 転送ページ数を指定します。
- `mbmaxpages` - ネットワーク・バッファに使用する単一のページ・クラスタの最大数を指定します。
- `ecc_recover_enable` - マルチビット・エラーに対するシステムの応答を指定します。
- `utrace_bufsize` - 各 CPU に格納されている 48 バイトの `utrace` エントリの数を選択します。
- `dump_level` - ダンプ・レベルを指定します。
- `buf_relse_age` - バッファ・キャッシュ・ページをいつ解放するかを指定します。

`cachefs_readahead`

説明	<code>cachefs_readahead</code> パラメータは、ファイルシステム内で現在読取っているブロックの先読み（リードアヘッド）ブロックの数を指定します。これらのブロックは、非同期に読取られます。
値	デフォルト: 1 (0x1) 範囲: 0 ~ 10

`cachefs_max_threads`

説明	<code>cachefs_max_threads</code> パラメータは、 <code>cachefs</code> のマウントごとに行われる非同期 I/O デーモンの最大数を指定します。
値	デフォルト: 5 (0x5) 範囲: 1 ~ 10

nbuf

説明

nbuf パラメータは、ファイルシステム・バッファ・キャッシュ内のバッファ・ヘッダ数を指定します。各バッファ・ヘッダ用のメモリは必要に応じて動的に割当てられます。メモリのサイズは可変です。書込み用には 1 ~ 128 ブロック (512 ~ 64 KB)、読取り用には 1 ~ 1024 ブロック (512 ~ 512 KB) が割当てられます。

システムは、ファイルシステム・バッファ・キャッシュを使用してファイルシステムの I/O 要求を最適化します。バッファ・メモリはディスクからブロックをキャッシュし、頻繁に使用されるブロックはキャッシュ内に置かれます。これは、過剰なディスクのアクティビティを防ぎます。

バッファはトランザクション・ヘッダとして使用されます。I/O 操作が終了すると、バッファはマップされていたメモリから離れて、バッファ・ヘッダがほかの入出力に使用できるようになります。したがって、多くのシステムでは、バッファ・ヘッダの数が少なくても十分機能します。*nbuf* が 0 に設定されていると、システムは自動的に *nbuf* を平均的なシステム向けの値に設定します。平均的でないシステムに対してこのパラメータを大きくしても、ほとんど負荷はかかりません。

nbuf は /var/sysgen/mtune で定義されています。

値

デフォルト : 0 (自動設定)

式 :

32 ビット・カーネル : $\min(100 + (\text{メモリのキロバイト数}) / 160, 6000)$

64 ビット・カーネル : $\min(100 + (\text{メモリのキロバイト数}) / 160, 600000)$

変更要件

平均的なシステムには自動設定が適しています。sar および osview の出力でキャッシュ・ヒット率の低下が見られる場合は、このパラメータを増やします。また、大量のファイル (1000 以上) のファイルが含まれるディレクトリがある場合も、このパラメータを増やすことができます。

大規模な (64 個以上のプロセッサ) システムに対して推奨する初期値は、2000 です。

min_bufmem

説明	min_bufmem パラメータは、システムがメモリ不足の状態になった場合に、バッファ・キャッシュにキャッシュされるファイルシステム・メタデータによって保持されるメモリの最低量を指定します。値はメモリのページ数で表します。
値	デフォルト: 物理メモリの2パーセント
変更要件	マシンのI/Oの負荷が高いにもかかわらず、ディレクトリ情報にすばやくアクセスする必要がある場合にこの値を変更します。

callout_himark

説明	callout_himark パラメータは、システム全体のコールアウト・テーブル・エントリの最大数を指定します。コールアウト・テーブルはデバイス・ドライバによって使用され、デバイスがコマンドに応答しない場合にシステムが停止しないようにタイムアウトを設定します。 このパラメータは、 <code>/var/sysgen/mtune</code> に次のような式で定義します。 $nproc + 32$ $nproc$ はシステム全体の最大プロセス数です。
値	デフォルト: 0 (自動設定) 式: $nproc + 32$ 範囲: 42 ~ 1100
変更要件	これ以上コールアウトができないことを示すコンソール・エラー・メッセージが表示された場合は、値を大きくします。

ncallout

説明	ncallout パラメータは、起動時のコールアウト・テーブル・エントリ数を指定します。コールアウト・テーブルのエントリ数を使い果たすと、システムは新しいエントリを自動的に割当てます。ただし、エントリの最大数は、 <code>callout_himark</code> パラメータで定義されています。
値	デフォルト: 40

		範囲 : 20 ~ 1000
	変更要件	通常より多くのデバイス・ドライバを使用するとき、このパラメータの値を更新します。システムは、新しいエントリが必要になると自動的に割当て、不要になると解放します。
reserve_ncallout		
	説明	reserve_ncallout パラメータは、コールアウト・テーブルの予約エントリ数を指定します。システムが通常のエントリを使い果たし、追加のエントリも割当てられない場合は、この予約エントリをカーネル割込みルーチン用に使えます。
	値	デフォルト : 5 範囲 : 0 ~ 30
ncsize		
	説明	ncsize パラメータは、ネーム・キャッシュのサイズを指定します。ネーム・キャッシュを使用すると、システムはメイン・バッファ・キャッシュからディレクトリ名を読取る処理を省略できます。このキャッシュ・エントリには、ディレクトリ名、ディレクトリのメモリ内の i ノードやバージョン番号へのポインタ、さらに親ディレクトリのメモリ内の i ノードやバージョン番号への同じようなポインタが含まれています。
	値	デフォルト : 0 (自動設定) 範囲 : 268 ~ 1 M
ndquot		
	説明	ndquot パラメータは、ディスク割当てを制御します。
	値	デフォルト : 0 (自動設定) 範囲 : 268 ~ 6200

nproc

説明	<p>nproc パラメータは、システム・プロセス (proc) テーブルのエントリ数を指定します。各実行プロセスは、メモリ内の proc 構造体を必要とします。nproc は、指定された時点でシステム内に存在できるプロセスの最大数を示します。</p> <p>nproc のデフォルトは、システムのメモリ容量で決定されます。現在の自動設定値を確認するには、systune コマンドを使用します。</p> <p>nproc パラメータは、/var/sysgen/mtune で定義します。</p>
値	<p>デフォルト: 0 (自動設定)</p> <p>範囲: 30 ~ 1 M</p>
変更要件	<p>sar -v 出力の proc -sz ov カラムがオーバフローした場合、またはオペレーティング・システムから次のメッセージを受取った場合は、この値を大きくします。</p> <p>プロセスをこれ以上作成できません</p> <p>このメッセージは、システム内のすべてのプロセス数が現在の設定値に到達したことを示します。プロセスのフォークが行われない (プロセスが生成されない) 場合は、この値を増やします。関連するパラメータに maxup があります。</p> <p>64 プロセッサまたはそれ以上のシステムの推奨する初期値は、8000 です。</p>
注記	<p>プロセスをフォークできない場合は、それがシステム全体の問題なのか、ユーザ ID の問題なのかを確認します。maxup パラメータを参照してください。</p> <p>nproc が小さすぎると、フォークしようとしたプロセスはオペレーティング・システムから次のようなエラー・メッセージを受取ります。</p> <p>EAGAIN: プロセスをこれ以上作成できません。</p> <p>またシェルからも次のようなメッセージを受取ります。</p> <p>fork failed: too many processes</p> <p>sched、vhand、init、または bdflush などのシステム・デーモンがプロセス・テーブル・エントリを割当てることができない場合は、システムは停止し、次のようなメッセージを表示します。</p> <p>No process slots</p>

maxpmem

説明	<p>maxpmem パラメータは、システムが認識できる物理メモリ容量 (ページ数) を指定します。ページ・サイズは、32 ビット対応の IRIX バージョン 5 またはバージョン 6 については 4 KB、64 ビット対応の IRIX バージョン 6 については 16 KB です。ゼロに設定すると、システムはメモリ内の使用可能なページをすべて使用します。ゼロ以外の値を設定すると、システムは指定した物理メモリ・サイズだけを使用します。</p> <p>このパラメータは、<code>/var/sysgen/mtune</code> で定義されています。</p>
値	<p>デフォルト : 0 (自動設定)</p> <p>範囲 : 1024 ページ ~ メモリの総容量</p>
変更要件	<p>ベンチマークで物理メモリ・サイズ未満のメモリ・サイズを特別に指定する場合を除き、このパラメータを変更する必要はありません。このパラメータは、主にカーネルの開発者にとって有効です。次のコマンドを起動コマンド行に追加して起動すると、同じ効果を得ることができます。</p> <p>maxpmem = memory_size</p>

syssegsz

説明	<p>syssegez パラメータは、動的システム・メモリの最大ページ数を指定します。</p>
値	<p>デフォルト : 0 (自動設定)</p> <p>32 ビット範囲 : 0x2000 ~ 0x20000</p> <p>64 ビット範囲 : 0x2000 ~ 0x10000000</p>
変更要件	<p><i>maxdmasz</i> を大きくする場合、または多くの動的メモリを割当てるカーネル・ドライバをインストールする場合は、このパラメータを大きくします。</p> <p>64 プロセッサまたはそれ以上のシステムの推奨する初期値は、0xfe800 です。</p>

maxdmasz

説明	<p>maxdmasz パラメータは、最大 DMA 転送量をメモリ・ページで表したものです。この値は、syssegsz と maxpmem の値未満でなければなりません。</p>
----	---

値	デフォルト: 1025 (32 ビット・カーネル) デフォルト: 257 (64 ビット・カーネル) 32 ビット範囲: 1 ~ <i>syssegsz</i> (最大 0x20000) 64 ビット範囲: 1 ~ <i>syssegsz</i> (最大 0x10000000)
変更要件	大量の読み書きを必要とするシステム・コール、または大規模な DMA 転送を行うシステム・コールを実行する必要がある場合は、このパラメータを変更します。このような状況は、光学スキャナ、フィルム・レコーダ、プリンタなどを使用するとき 발생합니다。 実際の DMA 転送サイズは、このパラメータで設定したページ数より 1 つ少なくなります。そのため、16 MB (1024 ページ) のフル DMA 転送を行うには、このパラメータの値を 1025 に設定します。 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、0x2001 です。

mbmaxpages

説明	<i>mbmaxpages</i> パラメータは、ネットワーク・バッファに割当てることができる単一のページ・クラスタの最大数を指定します。これは、ネットワーク・バッファ (<i>mbufs</i>) が消費できるメモリ総容量の限度を設定します。
値	デフォルト: 物理メモリ容量の 1/4 範囲: デフォルト ~ システム・メモリ総容量 (ページ数)
変更要件	ほとんどのワークステーションや小さなサーバにおいては、デフォルト値で十分です。非常に大きなサーバにおいては、より大きな値が適切です。0 の値を指定すると、カーネルはハードウェアに設定した物理メモリ容量に基づく値 (物理メモリ容量の 1/4) を設定します。 <i>sysstune</i> コマンドで設定できる最小値は 16 ページです。最大値は、32 ビット・システムでは 8192 ページ、64 ビット・システムでは 2097152 ページです。

ecc_recover_enable

説明	ecc_recover_enable パラメータを 0 に指定すると、システムはマルチビット・エラーの回復を実行しません。パラメータに 0 より大きな値を指定した場合は、これを秒数と解釈します。システムはエラー回復を実行しますが、このパラメータで指定した秒数の間、処理できるエラーは最大 32 個です。
値	デフォルト : 60 秒 (0x3c)

utrace_bufsize

説明	utrace は、カーネルのデバッグ情報を収集する軽負荷のトレーシング・メカニズムです。utrace_bufsize パラメータは、各 CPU に格納される 48 バイトの <i>utrace</i> エントリの数を選択します。値を 0 に設定すると、トレースの収集を行いません。サポートされるバッファ・サイズは、0 と 2048 だけです。
値	デフォルト : 0 範囲 : 0 ~ 0x7ffffff

dump_level

説明	dump_level パラメータは、ダンプ・レベルを指定し、システム・クラッシュ時にダンプされるデータを制御します。このパラメータを 0 に設定すると、パニック時に putbuf、errbuf、およびロー・メモリのみをダンプします。1 に設定した場合は、静的、動的、およびバクトレース・カーネル・ページと、該当する場合は pfdats もダンプします。2 に設定した場合には、バッファ・キャッシュ・ページもダンプします。3 に設定した場合には、使用中のまま残っているページもダンプします。4 に設定した場合にはフリー・ページもダンプします。
値	デフォルト : 0 範囲 : 0 ~ 4 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、3 です。

buf_relse_age

説明	<p>buf_relse_age パラメータは、バッファ・キャッシュ・ページをいつ解放するかを指定します。ファイルのデータが少なくとも buf_relse_age 秒使用されていない場合、データは解放されます。</p> <p>このランタイム設定可能な変数は、一部のノードには多くの空きメモリがあり、他のノードではローカル・メモリがすべて消費されるようなシステムで便利です。システムはグローバルにメモリを監視しており、ノードごとのメモリを区別できないため、後者に非常に古いバッファ・キャッシュ・データが大量に含まれる場合、通常のバッファ解放システムではデータが解放されないことがあります。</p>
値	<p>デフォルト: 0 (0 はこの機能を使用しないことを意味します)</p> <p>範囲: 0 ~ 0x7fffffff 秒</p>
変更要件	<p>負荷やニーズはすべてのシステムで異なるので、推奨設定はありません。30 秒未満の設定では処理が多くなりすぎます。また、10 秒未満の設定では大量のスラッシングが発生します。</p> <p>有効な解決方法としては、buf_relse_age を 600 秒 (10 分) に設定し、時間をかけて bufview および osview コマンドの表示を調べ、使用可能なメモリの量を確認してから、適切になるまで buf_relse_age の値を調整します。</p>

システム・リミット・パラメータ

IRIX システムには、システム・リミットを設定するパラメータがあります。たとえば、これらのパラメータを指定し、各プロセスの最大値 (コア・サイズまたはファイル・サイズ) やユーザあたりのグループ数、常駐ページ数などを設定できます。パラメータのタイプは次のとおりです。これらのパラメータはすべて /var/sysgen/mtune/kernel で設定され、定義されます。

- maxup は、ユーザあたりのプロセス数を指定します。
- ngroups_max は、ユーザが所属するグループ数を指定します。
- maxwatchpoints は、プロセスあたりの監視点の最大数を指定します。
- プロファイリングする分離テキスト領域の数を指定します。

- `maxsymlinks` は、パス名で拡張したシンボリック・リンクの最大数を指定します。

maxup

説明	<code>maxup</code> パラメータは、1回のユーザ・ログインで可能なプロセス数を指定します。この値は、必ず <code>nproc</code> より 20 以上小さくしてください。
値	デフォルト : 150 プロセス 範囲 : 15 ~ 30000 (必ず <code>nproc</code> から 20 引いた数以下にする)
変更要件	ユーザあたりのプロセス数を増加するとき、このパラメータ値を増やします。負荷の大きいタイム・シェアリング環境の場合は、この値を小さくし、ユーザあたりのプロセス数を減らすことができます。 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、8000 です。

ngroups_max

説明	<code>ngroups_max</code> パラメータは、各ユーザが同時に所属できるグループの最大数を指定します。 NGROUPS_UMIN \leq <code>ngroups_max</code> \leq NGROUPS_UMAX 定数は、 <code></usr/include/sys/param.h></code> で定義されています。NGROUPS_UMIN は、 <code>lboot</code> 時に選択できるグループ数の最小値です。NGROUPS_UMAX は、 <code>lboot</code> 時に選択できるグループ数の最大値で、コンパイル時に領域が確保されたグループ <code>id</code> スロットの数を示します。NGROUPS は、 <code></usr/include/sys/param.h></code> で定義されているネットワーク・コードと互換性を保持するためのもので、 <code>ngroups_max</code> 以下に設定します。
値	デフォルト : 16 範囲 : 0 ~ 32
変更要件	ほとんどのシステムでは、デフォルト値の使用が適切です。16 を超えるグループに同時にアクセスするシステムの場合は、この値を増やします。

maxwatchpoints

説明	<i>maxwatchpoints</i> パラメータは、プロセスあたりの監視点の最大数を指定します。監視点は <i>proc(4)</i> ファイルシステムで設定し、使用されます。このパラメータは、トレースされるプロセスにある監視対象の仮想アドレス・セグメントの最大数を示します。通常、デバッガによって使用されます。
値	デフォルト: 100 範囲: 1 ~ 1000
変更要件	デバッガが監視点不足となった場合は、このパラメータの値を増やします。

nprofile

説明	<i>nprofile</i> パラメータは、 <i>sprofil(2)</i> システム・コールを使用しているプロファイリング可能な分離テキスト領域の最大数を指定します。共有ライブラリを使用しているプログラムをプロファイリングする場合、またはテキストのセクションごとに異なる実行単位を使用しているアドレス領域をプロファイリングする場合は、このパラメータが有効になります。
値	デフォルト: 100 範囲: 100 ~ 10000
変更要件	現在の設定よりも多くのテキスト領域をプロファイリングする場合は、このパラメータの値を増やします。

maxsymlinks

説明	<i>maxsymlinks</i> パラメータは、ファイル名の検索の際 (<i>open(2)</i> または <i>stat(2)</i> などのシステム・コール実行時)、検索を終了するまでに参照するシンボリック・リンクの最大数を指定します。この制限は、連鎖しているシンボリック・リンクのリンク・ポイントが元のファイル名に戻るようなループを避けるために必要です。
値	デフォルト: 30 幅: 0 ~ 50

変更要件 30 以上のシンボリック・リンクを持つパス名がある場合は、このパラメータの値を増やします。

リソース・リミット・パラメータ

`getrlimit(2)`、`setrlimit(2)`、およびシェル組込みコマンドである `limit` を使用すると、プロセス単位で、使用するリソースの制限値を設定できます。これらの制限値は継承され、元の値は `/var/sysgen/mtune` に設定されています。また、これらの値は、現在のプロセスとそこから派生する子プロセスにだけ適用されるという点で、前のセクションのシステム・リミットの値とは異なります。システム・リミットの値と同じような結果を得るために、**Bourn** シェルや **C** シェル、**Korn** シェル（それぞれ `/bin/sh`、`/bin/csh`、および `/bin/ksh`）内で `limit` コマンドを使用することもできます。リソース・リミットについての詳細は、『*IRIX Admin: Resource Administration*』を参照してください。

制限値には、それぞれデフォルト値と最大値があります。最大値を変更できるのは特権ユーザだけです。各リソースは、チェックをオフにする値を持つことができます。ほとんどのシステムでは、デフォルト値の使用が適切です。

メモ：リソース・リミットの値 (`rlim_t`) は、**n32** および **n64** バイナリの両方で 64 ビットです。したがって、**n32** バイナリで 64 ビットのリミットを設定できます。`rlim_t` は **o32** バイナリでは 32 ビットなので、**o32** バイナリで 64 ビットのリミットを設定することはできません。**IRIX** は、**o32**、**n64**、および **n32** の 3 つのアプリケーション・バイナリ・インタフェース (ABI: Application Binary Interface) をサポートしています。ABI の詳細については、`abi(5)` マン・ページを参照してください。

リソース・リミットに関するパラメータは次のとおりです。

- `ncargs` は、`exec(2)` コール時に引渡される引数のバイト数を指定します。
- `rlimit_core_cur` は、コア・ファイル・サイズの現在の制限値を指定します。
- `rlimit_core_max` は、`rlimit_core_cur` の上限値を指定します。
- `rlimit_cpu_cur` は、プロセスが使用できる CPU 使用時間の現在の制限値を指定します。
- `rlimit_cpu_max` は、`rlimit_cpu_cur` の上限値を指定します。

- `rlimit_data_cur` は、プロセスが使用できるデータ領域の現在の制限値を指定します。
- `rlimit_data_max` は、`rlimit_data_cur` の上限値を指定します。
- `rlimit_fsize_cur` は、プロセスが使用できるファイル・サイズの現在の制限値を指定します。
- `rlimit_fsize_max` は、`rlimit_fsize_cur` の上限値を指定します。
- `rlimit_nofile_cur` は、プロセスが使用できるファイル記述子の現在の制限値を指定します。
- `rlimit_nofile_max` は、`rlimit_nofile_cur` の上限値を指定します。
- `rlimit_pthread_cur` は、プロセスが作成できるスレッド数の現在の制限値を指定します。
- `rlimit_pthread_max` は、`rlimit_pthread_cur` の上限値を指定します。
- `rlimit_rss_cur` は、プロセスが使用できる常駐セット・サイズの現在の制限値を指定します。
- `rlimit_rss_max` は、`rlimit_rss_cur` の上限値を指定します。
- `rlimit_stack_cur` は、プロセスのスタック・サイズの現在の制限値を指定します。
- `rlimit_stack_max` は、`rlimit_stack_cur` の上限値を指定します。
- `rlimit_vmem_cur` は、プロセスの仮想メモリの現在の制限値を指定します。
- `rlimit_vmem_max` は、`rlimit_vmem_cur` の上限値を指定します。
- `rsshogfrac` は、常駐ページに割当てるメモリのパーセントを指定します。
- `rsshogslp` は、プロセスが使用できる常駐セットの最大値以上のページ数を指定します。
- `shlbmax` は、プロセスがリンクできる共有ライブラリの最大数を指定します。
- `cpulimit_gracetime` は、CPU の制限時間を超えるプロセスのグレース・ピリオドを指定します。

`ncargs`

説明 `ncargs` パラメータは、`exec(2)` システム・コール時に引渡すことができる引数の最大サイズをバイトで指定します。

このパラメータは、`/var/sysgen/mtune` で定義されています。

値	デフォルト: 20480 範囲: 5120 ~ 262144
変更要件	ほとんどのシステムでは、デフォルト値の使用が適切です。 <code>exec(2)</code> 、 <code>shell(1)</code> 、または <code>make(1)</code> から次のメッセージが表示された場合は、値を増やします。 E2BIG 引数リストが長すぎます
注記	このパラメータを大きく設定しすぎるとメモリを浪費することになり、メモリのページングは可能でも、プログラムが正常に動作しない場合があります。また、 <code>ncargs</code> より小さい制限値が独自に設定されたシェルがある点に注意してください。

`rlimit_core_cur`

説明	<code>rlimit_core_cur</code> パラメータは、プロセスに対するコア・イメージ・ファイル・サイズの現在の制限値を指定します。
値	デフォルト: <code>0x7fffffffffffff</code> (9223372036854775807 バイト) 範囲: <code>0</code> ~ <code>0x7fffffffffffff</code>
変更要件	コア・イメージ・ファイルのサイズを制限する場合に変更します。

`rlimit_core_max`

説明	<code>rlimit_core_max</code> パラメータは、コア・イメージ・ファイルのサイズの上限值を指定します。
値	デフォルト: <code>0x7fffffffffffff</code> (9223372036854775807 バイト) 範囲: <code>0</code> ~ <code>0x7fffffffffffff</code>
変更要件	コア・イメージ・ファイルのサイズに上限値を設定する場合に変更します。 <code>rlimit_core_cur</code> はこの値より大きくできません。

rlimit_cpu_cur

説明	rlimit_cpu_cur パラメータは、プロセス実行中に使用できる CPU 使用時間（秒）の現在の制限値を指定します。
値	デフォルト : 0x7fffffffffffffff (9223372036854775807 秒) 範囲 : 0 ~ 0x7fffffffffffffff
変更要件	CPU 使用時間を制限する場合に変更します。

rlimit_cpu_max

説明	rlimit_cpu_max パラメータは、プロセス実行中に使用できる CPU 使用時間の上限値を指定します。
値	デフォルト : 0x7fffffffffffffff (9223372036854775807 分) 範囲 : 0 ~ 0x7fffffffffffffff
変更要件	通常の CPU 使用時間に上限値を設定する場合に変更します。

rlimit_data_cur

説明	rlimit_data_cur パラメータは、プロセスのデータ・サイズの現在の制限値を指定します。
値	デフォルト : 0 (<i>rlimit_vmem_cur</i> * NBPP (0x20000000)) に自動設定 範囲 : 0 ~ 0x7fffffffffffffff
変更要件	データ・セグメント・サイズを制限する場合に変更します。 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、0 II です。

rlimit_data_max

説明	rlimit_data_max パラメータは、プロセス実行中に使用できるデータ・サイズの上限値を指定します。
値	デフォルト : 0 (<i>rlimit_vmem_cur</i> * NBPP (0x20000000)) に自動設定

		範囲: 0 ~ 0x7fffffffffffffff
変更要件		プロセスのデータ・セグメント・サイズに上限値を設定する場合に変更します。 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、0 11 です。
<code>rlimit_fsize_cur</code>		
説明		<code>rlimit_fsize_cur</code> パラメータは、プロセスに対するファイル・サイズの現在の制限値を指定します。
値		デフォルト: 0x7fffffffffffffff (9223372036854775807 バイト) 範囲: 0 ~ 0x7fffffffffffffff
変更要件		ファイル・サイズを制限する場合に変更します。
<code>rlimit_fsize_max</code>		
説明		<code>rlimit_fsize_max</code> パラメータは、ファイル・サイズの上限値を指定します。
値		デフォルト: 0x7fffffffffffffff (9223372036854775807 バイト) 範囲: 0 ~ 0x7fffffffffffffff
変更要件		すべてのファイルのサイズに上限値を設定する場合に変更します。
<code>rlimit_nofile_cur</code>		
説明		<code>rlimit_nofile_cur</code> パラメータは、プロセス実行中に使用できるファイル記述子の数の現在の制限値を指定します。
値		デフォルト: 200 範囲: 40 ~ 0x7fffffffffffffff
変更要件		ファイル記述子の数を制限する場合に変更します。

rlimit_nofile_max

説明	rlimit_nofile_max パラメータは、プロセス実行中に使用できるファイル記述子の数の上限値を指定します。
値	デフォルト: 2500 範囲: 0 ~ 0x7fffffffffffffff
変更要件	ファイル記述子の数に上限値を設定する場合に変更します。

rlimit_pthread_cur

説明	rlimit_pthread_cur パラメータは、プロセスが作成できるスレッドの現在の制限値を指定します。
値	デフォルト: 1024 (0x400)

rlimit_pthread_max

説明	rlimit_pthread_max パラメータは、rlimit_pthread_cur の上限値を指定します。
値	デフォルト: 1024 (0x400)

rlimit_rss_cur

説明	rlimit_rss_cur パラメータは、プロセス実行中に使用できる常駐セット・サイズ (指定された時点でのメモリのページ数) の現在の制限値を指定します。この値は、次の 2 つの式のうち大きい方の値になります。 $\text{physical_memory_size} - 4 \text{ MB}$ $\text{physical_memory_size} * 9/10$
値	デフォルト: 0。rlimit_rss_cur が 0 の場合は、次の 2 つの式のうち大きい方の値に自動設定されます。 $\text{physical_memory_size} - 4 \text{ MB}$ $\text{physical_memory_size} * 9/10$ 範囲: 0 ~ 0x7fffffffffffffff

	変更要件	プロセスの常駐セット・サイズを制限する場合に変更します。 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、0 ll です。
<code>rlimit_rss_max</code>	説明	<code>rlimit_rss_max</code> パラメータは、プロセス実行中に使用できる常駐セット・サイズの上限值を指定します。
	値	デフォルト: 0。 <code>rlimit_rss_max</code> が 0 の場合は、次の 2 つの式のうち大きい方の値に自動設定されます。 $\begin{aligned} & \text{physical_memory_size} - 4 \text{ MB} \\ & \text{physical_memory_size} * 9/10 \end{aligned}$ 範囲: 0 ~ 0x7fffffffffffffff
	変更要件	常駐セット・サイズに上限値を設定する場合に変更します。 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、0 ll です。
<code>rlimit_stack_cur</code>	説明	<code>rlimit_stack_cur</code> パラメータは、プロセス実行中に使用できるスタック領域の現在の制限値を指定します。
	値	デフォルト: 64 MB (0x04000000) 範囲: 0 ~ 0x7fffffffffffffff
	変更要件	スタック領域使用を制限する場合に変更します。 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、(0x04000000) ll です。
<code>rlimit_stack_max</code>	説明	<code>rlimit_stack_max</code> パラメータは、プロセス実行中に使用できるスタック領域の上限值を指定します。
	値	デフォルト: <code>rlimit_vmem_cur</code> * NBPP (0x20000000) 範囲: 0 ~ 0x7fffffffffffffff

	変更要件	スタック領域使用に上限値を設定する場合に変更します。 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、(0x20000000) ll です。
rlimit_vmem_cur	説明	rlimit_vmem_cur パラメータは、プロセス実行中に使用できる仮想メモリの現在の制限値を指定します。
	値	デフォルト: 0。rlimit_vmem_cur が 0 の場合は、以下のように自動設定されます。 0x7fffffff (Origin 3000、Origin 2000、Challenge、O200、および Onyx2 システムの場合) 0x80000000 (Indgio 2、Power Indgio 2、Octane、および O2 システムの場合) 0x40000000 (より小規模なシステムの場合) 範囲: 0 ~ 0x7fffffff
	変更要件	仮想メモリ使用のサイズを制限する場合に変更します。 64 プロセッサまたはそれ以上のシステムの推奨する初期値は、0 ll です。
rlimit_vmem_max	説明	rlimit_vmem_max パラメータは、プロセス実行中に使用できる仮想メモリの上限値を指定します。
	値	デフォルト: 0。rlimit_vmem_max が 0 の場合は、以下のように自動設定されます。 0x40000000 (小規模なマシンの場合) 0x80000000 (大規模なマシンの場合) 0x7fffffff (SN アーキテクチャの場合) 範囲: 0 ~ 0x7fffffff

変更要件	<p>仮想メモリ使用のサイズに上限値を設定する場合、あるいは現在のパラメータ値よりもスワップ領域が大きい場合に変更します。</p> <p>64 プロセッサまたはそれ以上のシステムの推奨する初期値は、0 ll です。</p>
rsshogfrac	<p>説明</p> <p>rsshogfrac パラメータは、常駐ページに割当てられたメモリの割合を指定します。</p> <p>指定された時点でプロセスが占有する物理的なメモリ・ページ数を常駐セット・サイズ (RSS: Resident Set Size) と言います。RSS の制限値は、メモリが使用するリソース・リミットによって決定されます。rsshogfrac は、プロセスが RSS の制限値を超えても、常にある程度メモリがあるので、対話型の応答を保証します。消費した総メモリの割合は、100 % から rsshogfrac と最大メモリまたは 100 MB (いずれか値の小さい方) を掛けた値を引いた数になります。</p> <p>プロセスは、次の状態になるまで RSS の制限値を超えることが許可されています。</p> <ul style="list-style-type: none"> • プロセスがデフォルトの制限値を超え (したがって、RSS 専有状態になる)、空きメモリ容量が物理メモリの総容量である rsshogfrac になるまで。 • 空きメモリ容量が gpgshi より少なくなるまで。 <p>どちらの場合も、ページング・デーモンが起動し、RSS の制限値を超えているすべての RSS プロセスからページを削除します。</p> <p>rsshogfrac パラメータは、システムにある物理的なメモリ総容量に対するパーセントで表されます。デフォルト値は 75% です。</p> <p>このパラメータは、/var/sysgen/mtune で定義されています。詳細については、gpgshi、gpgslo、および rsshogslop の各リソース・リミットを参照してください。</p> <p>値</p> <p>デフォルト: メモリの総容量の 75 %</p> <p>範囲: メモリの総容量の 0 ~ 100 %</p> <p>変更要件</p> <p>ほとんどのシステムでは、デフォルト値の使用が適切です。</p> <p>64 プロセッサまたはそれ以上のシステムの推奨する初期値は、99 です。</p>

rsshogslop

説明	<p>rsshogslop パラメータは、プロセスが使用できる常駐セットの最大値を超えたページ数を指定します。スラッシング（コンピュータがその CPU 能力の 100 % をスワッピングとページングに費やしている状態）を避けるために、プロセスは常駐セットの最大値より rsshogslop のページ分多く使用できます。245 ページの「リソース・リミット・パラメータ」を参照してください。</p> <p>このパラメータは、<code>/var/sysgen/mtune</code> で定義されています。詳細については、<code>rsshogfrac</code> を参照してください。</p>
値	デフォルト: 20
変更要件	ほとんどのシステムでは、デフォルト値の使用が適切です。

shlbmax

説明	<p>shlbmax パラメータは、プロセスがリンクできる共有ライブラリの最大数を指定します。</p> <p>このパラメータは、<code>/var/sysgen/mtune</code> で定義されています。</p>
値	デフォルト: 8 範囲: 3 ~ 32
変更要件	ほとんどのシステムでは、デフォルト値の使用が適切です。 <code>exec(2)</code> から次のメッセージが表示された場合は、値を大きくします。 <p>ELIBMAX cannot link</p>

cpulimit_gracetime

説明	<p>cpulimit_gracetime パラメータは、CPU 制限時間を超えるプロセスのグレース・ピリオドを指定します。これは、プロセスが <code>rlimit_cpu_cur</code> パラメータで設定された制限時間を超えて実行できる時間を秒数で示したものです。cpulimit_gracetime が設定されていない場合 (0 の場合)、プロセスまたはジョブの CPU 制限を超えるプロセスには SIGXCPU シグナルが送られ</p>
----	--

ます。カーネルは、このプロセスが実行を継続するかぎり SIGXCPU シグナルを周期的に送ります。プロセスは、SIGXCPU を処理するように登録できるため、CPU 制限を事実上無視することができます。

`cpulimit_gracetime` は、`sysctl(1M)` コマンドによって 0 以外の値に設定されると、その動作が異なります。カーネルは、プロセスが CPU 制限を超えると、SIGXCPU シグナルを一度だけプロセスに送ります。プロセスは、このシグナルを処理するように登録後、必要なクリーンアップ操作やシャットダウン操作を実行できます。プロセスが CPU 制限時間を `cpulimit_gracetime` 秒分超えた後も実行している場合、カーネルは SIGKILL シグナルによってプロセスを終了します。

値

デフォルト: 0

範囲: 0 ~ 86400 (秒)

推奨される初期値は、サイトによって異なります。

ページング・パラメータ

ページング・デーモンである `vhand` は、必要に応じてメモリを解放します。このデーモンは、おおよそのプロセス・ワーキング・セットに対して最も使用されていないものを選択するアルゴリズムを使用し、指定された期間にアクセスされなかったページをディスクに書込みます。ページ・サイズは、32 ビット対応のカーネルでは 4 K、64 ビット対応のカーネルでは 16 K です。

`vhand` は、次の処理を行ってメモリを再利用します。

- 常駐セット・サイズの最大値を超えたプロセスからメモリを取得します。この結果、`bdflush` を使用し、書込み遅延状態にあったデータ・バッファをディスクに書込み、ページを再利用できるようにします。
- 動的にサイズが変更されるデータ構造を小さくするようにシステム・リソース・アロケータに対して要求します。
- 最低優先順位にあるプロセスからページを取得し、そのプロセスで最も使用されていないページから順に取得します。

次のチューニング・パラメータによって、vhand の実行頻度や実行条件が決定されます。これらのパラメータの値は、ほとんどのアプリケーションでデフォルト値を使用できます。

次にパラメータを示します。

- `bdflushr` は、ディスクに一度に書込めるバッファの数を指定します。`bdflush` は、不正なバッファを 1 秒に 1 回フラッシュします。
- `gpgsmsk` は、指定したページをスワップできるかどうかを決定するマスクを指定します。
- `gpgshi` は、空きページがこのパラメータ以上になった場合は、vhand がページの取出しを停止するページ数の境界値を指定します。
- `gpgslo` は、空きページがこのパラメータ以下になった場合は、vhand がページの取出しを始めるページ数の境界値を指定します。
- `maxlkmem` は、非特権ユーザが `mpin(2)` または `plock(2)` によってメモリ内にロックできる物理的なページの最大数を指定します。
- `maxfc` は、一度に空き領域にできる空ページの最大数を指定します。
- `maxsc` は、vhand デーモンが 1 回の操作でスワップできるページの最大数を指定します。
- `maxdc` は、一度にディスクに書込めるページの最大数を指定します。
- `minarmem` は、メモリの最小常駐ページ数を指定します。
- `minasmem` は、メモリのスワップ可能な最小ページ数を指定します。
- `numa_paging_node_freemem_low_threshold` は、vhandfs を開始するかどうかを調べるグローバル・システム・メモリの割当てプロシージャをいつトリガするかを指定します。
- `scache_pool_size` は、ページング・デーモンが使用するために常に予約しておくメモリの容量を指定します。
- `tlbdrop` は、プロセスのワイヤー・エントリをフラッシュするまでのクロック時間を指定します。
- `vfs_syncr` は、`vfs_syncr` が実行されるときに秒数を指定します。
- `maxpglst` は、ページ読出しの待ち行列で保持できる最大ページ数を指定します。

- `zone_accum_limit` は、フリー・メモリのゾーンを調整しグローバル・プールに戻す操作を実行する前に蓄積されるメモリ（NUMA システムのノード上のメモリ）の比率を指定します。

以下で説明するとおり、次のパラメータによってページ・サイズが決定されます。

- `percent_totalmem_16k_pages`
- `percent_totalmem_64k_pages`
- `percent_totalmem_256k_pages`
- `percent_totalmem_1m_pages`
- `percent_totalmem_4m_pages`
- `percent_totalmem_16m_pages`
- `nlpages_64k`
- `nlpages_256k`
- `nlpages_1m`
- `nlpages_4m`
- `nlpages_16m`

`bdflushr`

説明	<p><code>bdflushr</code> パラメータは、<code>bdflush</code> が実行されるたびに調査されるバッファの数を指定します。<code>bdflush</code> は、不正なファイルシステム・バッファを定期的にフラッシュします。これは <code>vfs_syncr</code> と同じ処理を実行します。<code>bdflush</code> デーモンは 1 秒に 1 回実行されます。</p> <p>このパラメータは、<code>/var/sysgen/mtune</code> で定義されています。詳細については、<code>autoup</code> カーネル・パラメータを参照してください。</p>
値	<p>デフォルト : 5</p> <p>範囲 : 1 ~ 31536000</p>
変更要件	<p>ほとんどのシステムでは、デフォルト値の使用が適切です。</p>

gpgsmsk

説明	<p>gpgsmsk パラメータは、ページをスワップできるかどうかを決定するマスクを指定します。ページャ (vhand) は、起動するたびに各アクティブ・ページのソフトウェア参照ビットの値を減らします。その後、プロセスがあるページを参照すると、そのページの参照ビット・カウンタは制限値 (NDREF、<code>/usr/include/sys/immu.h</code> で定義) にリセットされます。ページャが取出せるページを検索するとき (メモリ不足状態) は、参照ビット・カウンタが gpgsmsk 以下になったページだけを検索します。</p> <p>このパラメータは、<code>/var/sysgen/mtune</code> で定義されています。</p> <p>詳細については、<code>/usr/include/sys/immu.h</code> と <code>/usr/include/sys/tuneable.h</code>、およびカーネル・パラメータ gpgshi と gpgslo を参照してください。</p>
値	<p>デフォルト: 2</p> <p>範囲: 0 ~ 7</p>
変更要件	この値は、ほとんどのシステムで適切です。
注記	この値が 4 より大きい場合、ページは gpgsmsk のデフォルト値になる前にスワップ領域に書込まれます。適切なタイミングの前にスワッピング / ページングが実行されるので、システム・リソースが無駄に使用される場合があります。

gpgshi

説明	<p>vhand デーモン (ページ・ハンドラ) がページを取出している場合、空きページ量が gpgshi 以上になると、処理を中止します。</p> <p>つまり、vhand は、システム内の空きページが gpgslo 以下になるとページの取出しを始め、gpgshi 以上になるまで処理を続けます。</p> <p>起動時に gpgslo と gpgshi が 0 である場合は、システムは gpgshi をシステム内の全メモリ・ページ数の 8% とし、gpgslo を gpgshi の半分にします。</p> <p>このパラメータは、<code>/var/sysgen/mtune</code> で定義されています。詳細については、カーネル・パラメータ gpgsmsk と gpgslo を参照してください。</p>
値	デフォルト: 0 (全メモリ・ページの 8% に自動設定)

	範囲 : 30 ページ～全メモリ・ページの半分
変更要件	この値は、ほとんどのシステムで適切です。
	64 プロセッサまたはそれ以上のシステムの推奨する初期値は、2000 です。
注記	このパラメータが小さすぎると、 <i>vhand</i> はシステム全体の要求に対して十分な空きメモリを供給できません。
gpgslo	
説明	<i>vhand</i> デーモン (ページ・ハンドラ) が起動しても、システム内の空きページが <i>gpgslo</i> 未満にならないかぎり、ページの取出しは開始されません。開始すると、空きページが <i>gpgshi</i> になるまで継続します。
	このパラメータは、 <code>/var/sysgen/mtune</code> で定義されています。詳細については、 <i>gpgshi</i> と <i>gpgmsk</i> のカーネル・パラメータを参照してください。
値	デフォルト : 0 (<i>gpgshi</i> の半分に自動設定)
	範囲 : 10 ページ～全メモリの半分
変更要件	この値は、ほとんどのシステムで適切です。
	64 プロセッサまたはそれ以上のシステムの推奨する初期値は、1000 です。
注記	このパラメータが小さすぎると、 <i>vhand</i> はページのスワッピングを開始しないので、プロセス全体をスワップする必要があります。大きすぎると、 <i>vhand</i> は不必要にスワップを行います。
maxlkmem	
説明	<i>maxlkmem</i> パラメータは、非特権ユーザ・プロセスが <code>mpin(2)</code> または <code>plock(2)</code> によってメモリ内にロックできる最大の物理ページ数を指定します。
	このパラメータは、 <code>/var/sysgen/mtune</code> で定義されています。
値	デフォルト : 2000
	範囲 : 0 ページ～物理メモリの 3/4

変更要件	<p>特定のアプリケーションでメモリ上の多くのページをロックする必要がある場合にのみ、値を大きくします。</p> <p>マルチユーザ・サーバの場合、このパラメータと <i>rlimit_vmem_cur</i> の値を小さくすることがあります。</p>
注記	<p>ページがメモリにロックされていると、システムはそのページを再利用できないので、最も効率の高いページング処理が保持できません。</p>

maxfc

説明	<p><i>maxfc</i> パラメータは、<i>vhand</i> デーモンが 1 回の操作で空き領域にできる最大ページ数を指定します。デーモン (<i>vhand</i>) はページング処理を開始すると、ページ・プールに集められたページを収集します。一度に最大 <i>maxfc</i> ページを収集してから、それらを空き領域にします。<i>vhand</i> がページの収集を停止する前に、必要とする空のページ総数を設定する <i>gpgshi</i> パラメータとこのパラメータを混同しないでください。</p> <p>このパラメータは、<i>/var/sysgen/mtune</i> で定義されています。</p>
値	<p>デフォルト: 100</p> <p>範囲: 50 ~ 100</p>
変更要件	<p>この値は、ほとんどのシステムで適切です。</p>

maxsc

説明	<p><i>maxsc</i> パラメータは、<i>vhand</i> デーモンが 1 回の操作でスワップできる最大ページ数を指定します。このデーモンはスワップ処理を開始すると、スワップ領域に書込む必要があるページを収集してから、実際にスワップを行います。その後、それらのページをページ・プールに集めます。一度に最大の <i>maxsc</i> ページを収集してから、スワップアウトします。</p> <p>このパラメータは、<i>/var/sysgen/mtune</i> で定義されています。</p>
値	<p>デフォルト: 100</p> <p>範囲: 8 ~ 100</p>

	変更要件	NFS 上でスワップするシステムでは、値を小さくすることもあります。ディスクレス・システムでパフォーマンスを向上させる場合は常に値を小さくします。
maxdc	説明	<i>maxdc</i> パラメータは、一度にディスクに書き込み保存ができる最大ページ数を指定します。
	値	デフォルト : 100 範囲 : 1 ~ 100
	変更要件	システムのメモリ容量が小さく、NFS によるマウントなど、常にユーザ・メモリをリモート・スワップ領域にページアウトしている場合は、パラメータの値を一度に 10 ページを超えない範囲で小さくします。ただし、このパラメータの設定は通常のシステム・パフォーマンスには影響しません。
minarmem	説明	<i>minarmem</i> パラメータは、デッドロックを避けるために保持する必要がある常駐メモリの最小値を示します。
	値	デフォルト : 0 (自動設定)
	変更要件	どのシステムにおいても、デフォルト値で問題はありません。このパラメータを変更する必要はありません。
minasmem	説明	<i>minasmem</i> パラメータは、デッドロックを避けるために保持する必要があるスワップ可能メモリの最小値を示します。
	値	デフォルト : 0 (自動設定)
	変更要件	どのシステムにおいても、デフォルト値で問題はありません。このパラメータを変更する必要はありません。

numa_pagin_node_freemem_low_threshold

説明	numa_paging_node_freemem_low_threshold	パラメータは、vhandfs を開始するかどうかを調べるグローバル・システム・メモリの割当てプロシージャをいつトリガするかを指定します。
値	デフォルト: 40	

scache_pool_size

説明	scache_pool_size	パラメータは、ページング・デーモンが使用するために常に予約しておくメモリの容量を指定します。値には、予約されたメモリをキロバイト単位で指定します。端数がある場合は、常に次のページ境界に切上げられます。
値	デフォルト: 32	
変更要件		このパラメータの変更には注意が必要です。値を低く設定すると、メモリのデッドロックになる可能性があります。また値を高く設定すると、メモリを浪費することになります。scache ... out of memory というメッセージを表示してシステム・パニックが発生した場合には、このパラメータの値を増やしてください。それ以外は、このパラメータを変更しないでください。

tlbdrop

説明	tlbdrop	パラメータは、プロセスのワイヤ・エントリをフラッシュするまでのクロック時間を指定します。
値	デフォルト: 100	
変更要件		非常に大きなアプリケーションにおいて sar が大きなトランザクション参照バッファ (utlbmiss) 負荷を示した場合は、パラメータの値を大きくする必要があります。通常、アプリケーションが実行中の参照メモリ・フレームを変更することが多くなるほど tlbdrop を大きくしてパフォーマンスを向上させます。各アプリケーションの最適値を見つけるようにしてください。

vfs_syncr

説明	vfs_syncr パラメータは、vfs_syncr プロセスの実行間隔を秒数で指定します。
値	デフォルト : 30 範囲 : 1 ~ 31536000 (1 年)
変更要件	通常的环境下、このパラメータを変更する必要はありません。

maxpglst

説明	maxpglst パラメータは、各ページのページ読取り待ち行列に保持できる最大ページ数を指定します。
値	デフォルト : 0 (自動設定) 範囲 : 50 ~ 1000

zone_accum_limit

説明	zone_accum_limit パラメータは、フリー・メモリのゾーンを調整し、グローバル・プールに戻す操作を実行する前に蓄積されるメモリ (NUMA システムのノード上のメモリ) の比率を指定します。このパラメータを 30 % に設定し、ゾーンに保持されているフリー・メモリの容量が 30 % を超えると調整が開始されます。
値	デフォルト : 30 範囲 : 0 ~ 100

percent_totalmem_64k_pages

説明	percent_totalmem_64k_pages パラメータは、64 KB ページ数の上限値として使用される、メモリ総容量の比率を指定します。
値	デフォルト : 0 範囲 : 0 ~ 100 (%)

変更要件 動的に設定して 64 KB ページを取得します。
64 プロセッサまたはそれ以上のシステムの推奨する初期値は、0 です。

メモ：類似するほかのパラメータについては、229 ページの「複数のページ・サイズ」を参照してください。

nlpages_64k

説明 nlpages_64k パラメータは、起動時に 64 KB ページを確保するために使われます。このパラメータを使用することにより、メモリが少ないシステムで大きなページを確保できるようになります。

値 デフォルト : 0
範囲 : 0 ~ $memory_size / 64 * 1024$ 。 *memory_size* はシステムのメモリの総容量です。

変更要件 本当に必要な場合を除き、このパラメータを変更しないでください。

メモ：類似するほかのパラメータについては、229 ページの「複数のページ・サイズ」を参照してください。

プロセス間通信 (IPC: interprocess communication) パラメータ

IPC パラメータは、プロセス間通信 (IPC: interprocess communication) 構造を設定します。この構造には、IPC メッセージ (`/var/sysgen/mtune/msg` に指定) や IPC セマフォ (`/var/sysgen/mtune/sem` に指定)、IPC 共有メモリ (`/var/sysgen/mtune/kernel` に指定) があります。

IPC 構造が正しく設定されないと、システム・コールが失敗し、エラーが表示される場合があります。

IPC パラメータのサイズを増やす前に、`ipcs(1)` を使用して不要な IPC リソースを削除するかどうかを調べてください。たとえば、アプリケーションがセマフォやメモリ・セグメント、メッセージ待ち行列を削除していない場合、`shmget` は、エラー `ENOSPC` を発行します。

IPC オブジェクトは、すべてのアクティブ参照がなくなっても自動的に解放されない点が IRIX オブジェクトと異なります。特に、オブジェクトを生成したプログラムが残っている場合は解放されません。

表 A-1 に、エラー・メッセージ、エラーを発生するシステム・コール、および調整パラメータを示します。次に、この表に示すパラメータの値を大きくする場合の注意事項について説明します。

表 A-1 システム・コール・エラーと IPC 調整パラメータ

メッセージ	システム・コール	パラメータ
EAGAIN	<code>msgsnd()</code>	下記参照
EINVAL	<code>msgsnd()</code> <code>shmget()</code>	<code>msgmax</code> <code>shmmax</code>
EMFILE	<code>shmat()</code>	<code>sshmseg</code>
ENOSPC	<code>semget()</code> <code>shmget()</code>	<code>msgmni</code> <code>semmni</code> <code>shmmni</code>

EAGAIN IPC_NOWAIT が設定されており、次のような原因がある場合、`msgsnd` は EAGAIN を出力します。

- メッセージ待ち行列の総バイト数が `msgmnb` を超えた場合。

- メッセージが使用した総バイト数が $\text{msgseg} * \text{msgssz}$ を超えた場合。
 - システム全体のメッセージ・ヘッダの総数が msgtql を超えた場合。
- EINVAL** *shmget* (新しい共有メモリ・セグメント識別子を得るシステム・コール) は、指定したサイズが *shmmmin* と *shmmmax* の間にない場合に **EINVAL** を出力します。*shmmmin* は最低値 (1) に、また *shmmmax* は非常に大きい値に設定されているので、この両方を変更する必要はありません。
- EMFILE** *shmat* は、*sshmseg* 以上の共有メモリ・セグメントを持つと、**EMFILE** を出力します。*sshmseg* はプロセスあたりのシステム共有メモリ・セグメントの総数です。
- ENOSPC** *shmget* は、*shmmni* (システム全体の共有メモリ・セグメント数) が小さすぎる場合に **ENOSPC** を出力します。ただし、アプリケーションが共有メモリを生成したまま削除していない可能性もあります。したがって、パラメータの値を変更する前に、*ipcs* (1) を使用し、現在アクティブになっている共有メモリ・セグメントを表示して調べてください。
- semnmi* が小さすぎて、セマフォ識別子の総数が超過している場合、*semget* は **ENOSPC** を返します。
- msgmni* が小さすぎる場合、*msgget* は **ENOSPC** を返します。*ipcs* を使用し、メッセージ・キューが適切に削除されているかどうかを調べてください。

プロセス間通信 (IPC: interprocess communication) メッセージ・パラメータ

システムで IPC メッセージを使用中または使用を予定しているユーザがいない場合は、このモジュールを削除することができます。次に、IPC メッセージに関連するパラメータを示します。*msgctl* (2) マン・ページを参照してください。

- *msgmax* は、メッセージの最大サイズを指定します。
- *msgmnb* は、メッセージ待ち行列の最大長を指定します。
- *msgmni* は、システム全体のメッセージ待ち行列の最大数を指定します。
- *msgseg* は、システム全体のメッセージ・セグメントの最大数を指定します。
- *msgssz* は、メッセージ・セグメントのバイト数を指定します。

- `msgtql` は、システム全体のメッセージ・ヘッダの最大数を指定します。

`msgmax`

説明	<p><code>msgmax</code> パラメータは、メッセージの最大サイズを指定します。</p> <p>このパラメータは、<code>/var/sysgen/mtune/msg</code> で定義されています。</p>
値	<p>デフォルト : $16 * 1024$ (0x4000)</p> <p>範囲 : 512 ~ 0x8000</p>
変更要件	<p>メッセージの最大サイズを大きくする必要がある場合は、このパラメータの値を大きくします。メッセージのサイズを制限する場合は小さくします。</p>

`msgmnb`

説明	<p><code>msgmnb</code> パラメータは、メッセージ待ち行列の最大長を指定します。</p> <p>このパラメータは、<code>/var/sysgen/mtune/msg</code> で定義されています。</p>
値	<p>デフォルト : $32 * 1024$ (0x8000)</p> <p>範囲 : <code>msgmax</code> ~ 物理メモリの半分</p>
変更要件	<p>メッセージ待ち行列での最大バイト数を大きくする場合は、このパラメータの値を大きくします。待ち行列でのバイト数を制限する場合は小さくします。</p>

`msgmni`

説明	<p><code>msgmni</code> パラメータは、システム全体のメッセージ待ち行列の最大数を指定します。</p> <p>このパラメータは、<code>/var/sysgen/mtune/msg</code> で定義されています。</p>
値	<p>デフォルト : 50</p> <p>範囲 : 10 ~ 1000</p>
変更要件	<p>システム内のメッセージ待ち行列の数を増やす場合は、このパラメータの値を大きくします。待ち行列の数を制限する場合は小さくします。</p>

注記 メッセージ待ち行列の数が不足していると、msgget (2) システム・コールは新しいメッセージ待ち行列を作成しようとして、次のエラーを出力します。

ENOSPC: デバイスに空き領域がありません

msgseg

説明 *msgseg* パラメータは、システム全体のメッセージ・セグメントの最大数を指定します。メッセージ待ち行列のメッセージは、メッセージ・セグメントから構成されています。各セグメントのサイズは、*msgssz* パラメータで設定します。

このパラメータは、`/var/sysgen/mtune/msg` で定義されています。

値 デフォルト: 1536

変更要件 メッセージを格納するメモリの適切な容量を予約する場合は、このパラメータを変更します。メッセージ・セグメントに対してメモリを増やす必要がある場合は、パラメータの値を大きくします。メッセージ・セグメントが使用するメモリを制限する場合は小さくします。

注記 このパラメータの値が大きすぎると、メッセージ用のメモリが余り、無駄になる可能性があります。小さすぎると、予約されているメッセージ・バッファに格納できないメッセージもあります。この場合、メッセージ・バッファが使用可能になるまで、msgsnd (2) システム・コールは待ち状態になります。

msgssz

説明 *msgssz* パラメータは、メッセージ・セグメントのバイト数を指定します。メッセージは、連続したメッセージ・セグメントから構成され、メッセージ・テキストを格納する大きさを持ちます。セグメントを使用することで断片化をなくし、メッセージ・バッファの割当てを高速にできます。

このパラメータは、`/var/sysgen/mtune/msg` で定義されています。

値 デフォルト: 8

変更要件 このパラメータは、無駄なメッセージ・バッファを減らすために使用します。ほとんど無駄な領域はないにもかかわらず、多くのメッセージが1つのセグメントに入らない場合にだけ、このパラメータを変更します。

	このパラメータを変更する場合は、 <i>msgseg</i> パラメータも変更する必要があります。
注記	このパラメータが大きすぎると、メッセージ・バッファが断片化されるため無駄になる可能性があります。この結果、メッセージ・バッファの待ち状態の間、メッセージを送信するプロセスがスリープ状態に入ります。
<i>msgtql</i>	
説明	<i>msgtql</i> パラメータは、システム全体のメッセージ・ヘッダの最大数を指定するパラメータですが、結果として未処理の（未読の）メッセージの数を指定します。未処理のメッセージごとに1つのメッセージ・ヘッダが必要です。 このパラメータは、 <code>/var/sysgen/mtune/msg</code> で定義されています。
値	デフォルト: 40 範囲: 10 ~ 4000
変更要件	未処理のメッセージを多く必要とする場合は、この値を大きくします。数を制限する場合は小さくします。
注記	このパラメータが小さすぎると、 <i>msgsnd(2)</i> システム・コールは、 <i>msgtql</i> を制限値以上にするメッセージを送信しようとして、いくつかのメッセージが待ち行列から受取られる（読取られる）まで待ち状態に入ります。

プロセス間通信 (IPC: interprocess communication) セマフォ・パラメータ

システムで IPC セマフォを使用するユーザ、または使用を予定するユーザがいない場合は、このモジュールを削除できます。

次に、IPC に関連するパラメータを示します。*semctl(2)* マン・ページを参照してください。

- *semmni* は、カーネルで認識できるセマフォ識別子の最大数を指定します。
- *semmsl* は、セマフォ識別子あたりの最大セマフォ数を指定します。
- *semopm* は、1回の *semop(2)* システム・コールで実行できる最大セマフォ操作数を指定します。

- `semvmx` は、セマフォの最大値を指定します。
- `semaem` は、最大値に対する終了時での調整値を指定します。

`semnmi`

説明	<p><code>semnmi</code> パラメータは、カーネル内のセマフォ識別子の最大数を指定します。任意の時点でアクティブになるセマフォ・セットの数を示します。セマフォをセットとして生成します。つまり、セットあたり複数のセマフォが生成できます。</p> <p>このパラメータは、<code>/var/sysgen/mtune/sem</code> で定義されています。</p>
値	デフォルト : 300
変更要件	<p>プロセスが多くのセマフォ・セットを必要とする場合は、このパラメータの値を大きくします。大きくすると、セマフォ・セットの状態を保持するメモリも必要になります。このパラメータを変更した場合には、関連するその他のパラメータも変更する必要があります。</p> <p>64 プロセッサまたはそれ以上のシステムの推奨する初期値は、2000 です。</p>

`semmsl`

説明	<p><code>semmsl</code> パラメータは、セマフォ識別子あたりの最大セマフォ数を指定します。</p> <p>このパラメータは、<code>/var/sysgen/mtune/sem</code> で定義されています。</p>
値	デフォルト : 100
変更要件	セマフォ識別子あたりのセマフォ数を増すには、このパラメータの値を大きくします。

`semopm`

説明	<p><code>semopm</code> パラメータは、1 回の <code>semop()</code> システム・コールで実行可能なセマフォ操作の最大数を指定します。このパラメータによって、システムは <code>semop()</code> システム・コールを使用し、セマフォ・セット内の複数のセマフォの値をチェックしたり変更したりできます。</p> <p>このパラメータは、<code>/var/sysgen/mtune/sem</code> で定義されています。</p>
----	--

値	デフォルト : 100
変更要件	1 回の semop() システム・コールで実行可能なセマフォ操作の数を増減する場合は、このパラメータの値を変更します。 <i>semmsl</i> (セマフォのセット数) を増やす場合は、1 回のシステム・コールでセマフォ・セット内のすべてのセマフォをチェックし、変更できるように、このパラメータの値を大きくする必要があります。

semvmx

説明	この <i>semvmx</i> パラメータは、1 つのセマフォの最大値を指定します。 このパラメータは、 <code>/var/sysgen/mtune/sem</code> で定義されています。
値	デフォルト : 32767 (最大値)
変更要件	セマフォの最大値を制限する場合は、このパラメータ値を小さくします。

semaem

説明	<i>semaem</i> パラメータは、最大値に対する終了時での調整値 (別名 <i>semaadj</i>) を指定します。セマフォの値が <i>semop(2)</i> の絶対値以上になった場合は、プログラムがそのセマフォ固有の値を設定していないかぎり、このパラメータが使用されます。このパラメータは、 <code>/var/sysgen/mtune/sem</code> で定義されています。
値	デフォルト : 16384 (最大値)
変更要件	最大値に対する終了時での調整値を小さくする場合は、このパラメータの値を変更します。

プロセス間通信 (IPC: interprocess communication) 共有メモリ・パラメータ

次に、IPC メモリに関連するパラメータを示します。

- *shmmax* は、各共有メモリ・セグメントの最大サイズをバイト単位で指定します。
- *shmmmin* は、共有メモリ・セグメントの最小サイズをバイト単位で指定します。

- `shmmni` は、システム全体の共有メモリ識別子の最大数を指定します。
- `sshmseg` は、追加された共有メモリ・セグメントのプロセスあたりの最大数を指定します。

shmmax

説明	<p><i>shmmax</i> パラメータは、各共有メモリ・セグメントの最大サイズをバイト単位で指定します。</p> <p>このパラメータは、<code>/var/sysgen/mtune/kernel</code> で定義されています。</p>
値	<p>デフォルト: 0</p> <p>32 ビット範囲: 0x1000 ~ 0x7ffffff</p> <p>64 ビット範囲: 0x1000 ~ 0xffffffffffff ll</p>
変更要件	<p><i>shmmax</i> の値が 0 である場合、起動時にシステム・メモリ・サイズの 80% に調整されます。デフォルト値に数値が指定されている場合、カーネルはそれに従います。</p> <p>1 つの共有メモリ・セグメントで大量のメモリが占有されないようにする場合は、このパラメータを小さく設定します。</p> <p>64 プロセッサまたはそれ以上のシステムの推奨する初期値は、(0x4000000) ll です。</p>

shmmni

説明	<p><i>shmmni</i> パラメータは、共有メモリ・セグメントの最小サイズをバイト単位で指定します。</p> <p>このパラメータは、<code>/var/sysgen/mtune/kernel</code> で定義されています。</p>
値	<p>デフォルト: 1 バイト</p> <p>範囲: 1</p>
変更要件	<p>エラー・メッセージを出力したいときに、プロセスが要求する共有メモリの設定が小さすぎる場合は、このパラメータの値を大きくします。</p>

shmmni

説明	<p><i>shmmni</i> パラメータは、システム全体の共有メモリ・セグメントの最大数を指定します。</p> <p>このパラメータは、<code>/var/sysgen/mtune/kernel</code> で定義されています。</p>
値	<p>デフォルト: 0</p> <p>範囲: 5 ~ 100000</p>
変更要件	<p><i>shmmni</i> のデフォルト値が 0 の場合、<code>nproc</code> チューニング変数と同じ値になるように起動時に調整されます。デフォルト値に数値が指定されている場合、カーネルはそれに従います。</p> <p>共有メモリ・セグメントを 1 つ追加する場合は、このパラメータを 1 つ大きくします。また、共有メモリ・セグメントを多く使用するプロセスが <i>shmmni</i> の制限値に到達した場合も大きくします。</p> <p>共有メモリ・セグメントの最大数を減らす場合は、このパラメータを小さくします。また、共有メモリ・セグメント用に取得されるカーネル領域を減らす場合も小さくします。</p>

sshmseg

説明	<p><i>sshmseg</i> パラメータは、プロセスあたりの追加された共有メモリ・セグメントの最大数を指定します。プロセスは、共有メモリ・セグメントが追加されると、データにアクセスできます。</p> <p>このパラメータは、<code>/var/sysgen/mtune/kernel</code> で定義されています。</p>
値	<p>デフォルト: 2000</p> <p>範囲: 1 ~ 10000</p>
変更要件	<p>プロセスが、一度に使用できる共有メモリ・セグメントをデフォルト値より多く追加する場合は、このパラメータの値を大きくします。</p>

ストリーム・パラメータ

次に、ストリーム処理に関連するパラメータを示します。

- `nstrpush` は、ストリームにプッシュできるモジュールの最大数を指定します。
- `nstrintr` は、割込み時間に使用されるストリーム・バッファの数を指定します。
- `strctlsz` は、メッセージの `ctl` 部の最大サイズを指定します。
- `strmsgsz` は、ストリーム・メッセージの最大サイズを指定します。
- `strholdtime` は、ストリームの最大保持時間を指定します。
- `strpmonmax` は、プライベート・ストリームのモニタの最大数を指定します。

`nstrpush`

説明	<code>nstrpush</code> パラメータは、1 個のストリームにプッシュできるストリーム・モジュールの最大数を定義します。
値	デフォルト: 9 (0x9) 範囲: 9 ~ 10
変更要件	追加モジュールが必要な場合は、9 または 10 に変更します。

`nstrintr`

説明	<code>nstrintr</code> パラメータは、割込み時に使用するストリーム・バッファの最大数を定義します。
値	デフォルト: 1024 (0x400) 範囲: 32 ~ 4096

`strctlsz`

説明	<code>strctlsz</code> パラメータは、ストリーム・メッセージの <code>ctl</code> バッファの最大サイズを指定します。ストリーム・メッセージの <code>ctl</code> 部分と <code>data</code> 部分については、 <code>getmsg(2)</code> または <code>putmsg(2)</code> マン・ページを参照してください。
----	---

値	デフォルト : 1024 (0x400)
変更要件	ストリーム・メッセージの <i>ctl</i> 部分に大きなバッファを必要とする場合は、この値を変更します。

strmsgsz

説明	<i>strmsgsz</i> パラメータは、ストリーム・メッセージの最大サイズを指定します。ストリーム・メッセージの <i>ctl</i> 部分と <i>data</i> 部分を合わせた最大サイズを示します。前の <i>strctlsz</i> パラメータと組合わせて使用することにより、 <i>data</i> 部のサイズを設定します。ストリーム・メッセージの <i>ctl</i> 部分と <i>data</i> 部分については、 <i>getmsg</i> (2)、または <i>putmsg</i> (2) マン・ページを参照してください。
値	デフォルト : 0x8000
変更要件	ストリーム・メッセージの全体サイズおよび <i>data</i> 部分のサイズを決定するには、 <i>strctlsz</i> パラメータと関連付けてこのパラメータを変更します。

strholdtime

説明	<i>strholdtime</i> パラメータは、 <code><strsubr.h></code> の <code>STRHOLDTIME</code> マクロを定義します。このマクロは、ストリーム・ドライバのオーバーヘッドを減らします。このオーバーヘッドはシステムの表示されない部分に影響します。
値	デフォルト : 50 (0x32) 範囲 : 0 ~ 1000

strpmonmax

説明	<i>strpmonmax</i> パラメータは、プライベート・ストリームのモニタの最大数を定義します。
値	デフォルト : 4 (0x4) 範囲 : 0 ~ 1024

シグナル・パラメータ

`maxsigq` シグナル・パラメータは、カーネル内のプロセス間シグナル操作を制御し、待ち行列に設定できるシグナルの最大数を指定します。

maxsigq

説明	<code>maxsigq</code> パラメータは、待ち行列に設定できる最大シグナル数です。通常、同じシグナルの複数のインスタンスは 1 つしか送信しません。SA_SIGINFO フラグを使用すると、同じ種類の未処理シグナルはすべて待ち行列に入り、消失することはありません。
値	デフォルト: 64 範囲: 32 ~ 32767
変更要件	プロセスが大量のシグナルを受信すると予想され、64 個の待ち行列領域だけではシグナルが処理前に消失する可能性がある場合は、このパラメータを大きくします。 <code>maxsigq</code> パラメータは必要に応じて適切な値に変更します。

ディスパッチ・パラメータ

カーネルの重要な機能の 1 つに、ディスパッチング処理（優先処理）があります。コマンドが実行され、プロセスが生成されると、カーネルはそのプロセスにいくつかの特性を与えます。たとえば、CPU 時間を取得するための優先順位を与えます。この優先順位は、プロセスを要求したユーザまたは特権ユーザが変更できます。また、CPU 時間の長さ（スライス・サイズ）もディスパッチ・パラメータで調節できます。ディスパッチ・パラメータの一部として、定期デッドライン・スケジューラ (PDS: Periodic Deadline Scheduler) もあります。PDS は、ユーザ・プログラム内から `schedctl(2)` システム・コールによって呼出され、`sys/schedctl.h` ファイルを挿入するように要求します。次に、ディスパッチ・パラメータを示します。

- `memaff_sched` は、スケジューラ・メモリの類似性優先度をオンまたはオフに設定します。
- `slice_size` は、プロセスが取得する CPU 時間を指定します。

memaff_sched

説明	<i>memaff_sched</i> パラメータは、スケジューラ・メモリの類似性優先度をオンまたはオフに設定します。
値	デフォルト : 1 範囲 : 0 ~ 1
変更要件	スケジューラ・メモリの類似性優先度をオンまたはオフにしたい場合に、このパラメータを変更します。

slice_size

説明	<p><i>slice_size</i> パラメータは、デフォルトのプロセス・タイム・スライスで、システム・クロックの時間を示しています。システム・クロックの周波数は、値が 100 の定数 Hz で表します。したがって、<i>slice_size</i> の 1 単位は 10 ミリ秒です。あるプロセスが CPU の制御権を与えられると、カーネルはそのプロセスを <i>slice_size</i> の間実行させます。タイム・スライスが終了するか、プロセスが自発的に CPU の制御権を放棄すると（たとえば、<code>pause(2)</code> を呼ぶか、プロセスを待ち状態にするシステム・コールを呼ぶ）、カーネルは実行待ち行列を調べ、実行する権利がある最高優先順位のプロセスを選択します。</p> <p><i>slice_size</i> パラメータは、次のような形式で <code>/var/sysgen/mtune/disp</code> に定義されています。</p> <pre>#define slice_size Hz / 30 int slice_size = slice_size</pre>
値	<p>デフォルト : シングル CPU システムの場合には 2、マルチプロセッサ・システムの場合には 10。</p> <p>範囲 : 1 ~ 100</p>
変更要件	<p>主に計算中心のジョブ用にシステムを使用し、対話型の応答性が重要でない場合は、<i>slice_size</i> を大きくすることができます。たとえば、10 に設定すると、計算ジョブの効率は大幅に向上します。CPU の制御権を渡されるたびに、100 ミリ秒実行してから制御権を他に渡すからです。</p> <p>計算ジョブと対話型ジョブの両方が起動している環境でこのパラメータの値を大きくすると、対話型の応答時間に影響します。</p>

rtcpus

説明	rtcpus パラメータは、リアルタイム・スケジューラの品質を制御します。このオプションをオンにすると、IRIX オペレーティング・システムで1ミリ秒の応答時間を保証できるようになります。オンにすると、特に大規模なシステムではシステム・パフォーマンスが低下します。このパラメータはプロセッサの数で表し、システム内のプロセッサの数以上の場合にオンになります。
値	デフォルト: 0 (オフ) 範囲: 0 ~ 512
変更要件	ほとんどのマシンでは、デフォルト値が適切です。応答時間保証が必要な場合にオンにする必要があります。

ファイルシステム・パラメータ

IRIX のファイルシステムは、オペレーティング・システムのカーネルと密接に関連して動作します。次のパラメータによって、カーネルとこのファイルシステムのインタフェースが調節されます。

次のパラメータは、efs グループに定義されています。efs_inline は、シンボリック・リンク情報を i ノードに格納するかどうかを指定します。

以下のパラメータは、kernel パラメータ・グループとして設定し、EFS と XFS の両ファイルシステムを調節します。これらのパラメータによって、1回のディスク書き込み操作で何ページのメモリをクラスタ化するかが決定されます。これらのパラメータを調節することによって、ファイルシステムのパフォーマンスを大幅に向上できます。

- cwcluster は、各プッシュでクラスタ化する完遂書き込みページ数を指定します。
- bdflush_interval は、bdflush 機能が更新のあったファイル・データをディスクにプッシュする間隔を指定します。
- dwcluster は、各プッシュでクラスタ化する書込遅延状態のページ数を指定します。
- min_file_pages は、メモリが少なくなった場合にキャッシュに保存する最小のファイル・ページ数を指定します。

- `min_free_pages` は、ファイル・ページが `min_file_page` より上の場合の最小のフリー・ページ数を指定を指定します。
- `autoup` は、`bdflush` デーモンによってディスクに書込まれるまで、バッファが書き込み遅延状態に置かれる秒数を指定します。

efs_inline

説明	<code>efs_inline</code> パラメータを 0 以外の値に設定すると、どのシンボリック・リンク・データも、行の範囲外ではなく、ファイルの i ノード内に格納されます。
値	デフォルト: 0 範囲: 0 ~ 1
変更要件	この値を変更する必要はありません。この値を増やした場合には、シンボリック・リンクの検索速度が多少改善されます。ただし、このように扱われるファイルシステムは、IRIX システムのバージョン 5.3 より前のリリースで動作するシステムに移植できません。

cwcluster

説明	<code>cwcluster</code> パラメータは、各プッシュでクラスタ化する完遂書き込みページの最大数を設定します。このパラメータは NFS I/O のみに適用できます。
値	デフォルト: 64
変更要件	このパラメータを変更する必要はありません。自動設定値を使用できます。

bdflush_interval

説明	<code>bdflush_interval</code> パラメータは、 <code>bdflush</code> 機能が更新のあったファイル・データをディスクにプッシュする間隔を指定します。10 ミリ秒単位で指定します。
値	デフォルト: 100 (1 秒に 1 回)
変更要件	ほとんどのマシンでは、デフォルトで十分です。

dwcluster

説明	dwcluster パラメータは、書き込み遅延状態にあるページを 1 回で最大何ページ書込むかを指定します。この値は maxdmasz 値以下に設定してください。
値	デフォルト: 64
変更要件	このパラメータを変更する必要はありません。自動設定値を使用できます。

min_file_pages

説明	min_file_pages パラメータはメモリが少なくなった場合にキャッシュに保存する最小のファイル・ページ数を設定します。メモリが 0 になると自動的にシステム・メモリの 3% に設定されます。このパラメータを設定する場合には、32 ビット・カーネルでは 4 K のページ・サイズ、64 ビット・カーネルでは 16 K のページ・サイズになる点に注意してください。
値	デフォルト: 0
変更要件	このパラメータを変更する必要はありません。自動設定値を使用できます。

min_free_pages

説明	min_free_pages パラメータはファイル・ページが min_file_page より上の場合の最小のフリー・ページ数を設定します。メモリが 600 M 未満のシステムのデフォルト値は gpgshi*2 です。メモリがこれ以上のシステムのデフォルト値は gpgshi*4 です。
値	デフォルト: 0

autoup

説明	autoup パラメータは、bdflush デーモンによってディスクに書込まれるまで、バッファを書込み遅延状態に置く秒数を指定します。このパラメータは、/var/sysgen/mtune で定義します。詳細については、bdflushr カーネル・パラメータのエントリを参照してください。
----	---

autoup パラメータは、非アクティブとしてマークされているデータまたは *gr_osview rmemc* コマンドの *freec* ラベルで表示されているデータをクリアする時期を判断する場合にも使用します。値は、 $\text{autoup} \times \text{HZ} / 2$ またはデフォルトの $10 \times 100 / 2 = 500$ 秒のアルゴリズムを使用して計算されます。

値	デフォルト: 10 範囲: 1 ~ 30
変更要件	この値は、ほとんどのシステムで適切です。

ロード可能ドライバ・パラメータ

IRIX オペレーティング・システムでは、システムが稼動している間に、デバイス・ドライバをロードし、実行できます。動作中のカーネルを調節し、ロード可能ドライバが要求するリソースを用意する場合があります。次のパラメータを使用して、必要な調節を行うことができます。

- *bdevsw_extra* は、ブロック型デバイス・スイッチのエントリ数を指定します。
- *cdevsw_extra* は、キャラクタ型デバイス・スイッチのエントリ数を指定します。
- *fmodsw_extra* は、ストリーム・モジュール・スイッチのエントリ数を指定します。
- *vfssw_extra* は、仮想ファイルシステム・モジュール・スイッチのエントリ数を指定します。
- *munlddelay* は、ロード可能モジュールを自動的にアンロードするタイムアウト時間の指定を指定します。

bdevsw_extra

説明	<i>bdevsw_extra</i> パラメータは、ブロック型デバイス・スイッチのエントリ数を指定します。ロード可能ドライバ専用のパラメータです。lboot (1M) 時にシステムに 1 個のブロック型デバイスを設定している場合は、エントリを <i>bdevsw</i> に追加する必要はありません。
値	デフォルト: 21 範囲: 1 ~ 254

変更要件 22 個以上のブロック型デバイスを動的にシステムにロードする場合は、このパラメータを変更します。IRIX システムでは、*bdevsw* にデフォルトで 21 個のエントリがあります。

cdevsw_extra

説明 *cdevsw_extra* パラメータは、キャラクタ型デバイス・スイッチのエントリ数を指定します。ロード可能ドライバ専用のパラメータです。lboot (1M) 時にシステムに 1 個のキャラクタ型デバイスを設定している場合は、エントリを *cdevsw* に追加する必要はありません。

値 デフォルト: 23

範囲: 3 ~ 254

変更要件 24 個以上のキャラクタ型デバイスを動的にシステムにロードする場合は、このパラメータを変更します。IRIX システムでは、*cdevsw* にデフォルトで 23 個のエントリがあります。

fmodsw_extra

説明 *fmodsw_extra* パラメータは、ストリーム・モジュール・スイッチのエントリ数を指定します。ロード可能ドライバ専用のパラメータです。lboot (1M) 時にシステムに 1 個のストリーム・モジュールを設定している場合は、エントリを *fmodsw* に追加する必要はありません。

値 デフォルト: 20

範囲: 0 ~ 254

変更要件 21 個以上のストリーム・モジュールを動的にシステムにロードする場合は、このパラメータを変更します。IRIX システムでは、*fmodsw* にデフォルトで 20 個のエントリがあります。

vfssw_extra

説明	vfssw_extra パラメータは、vnode ファイルシステム・モジュール・スイッチのエントリ数を指定します。ロード可能ドライバ専用のパラメータです。lboot 時にシステムに 1 個の vfs モジュールを設定している場合は、エントリを vfssw に追加する必要はありません。
値	デフォルト : 5 範囲 : 0 ~ 254
変更要件	6 個以上の仮想ファイルシステム・モジュールを動的にシステムにロードする場合は、このパラメータを変更します。IRIX システムでは、vfssw にデフォルトで 5 個のエントリがあります。

munlddelay

説明	munlddelay パラメータは、ロード可能モジュールを自動的にアンロードするためのタイムアウト時間を分単位で指定します。
値	デフォルト : 5 (0x5) 分

CPU 動作パラメータ

CPU 動作パラメータは、マルチプロセッサ・システムにおいて、特定のタスクを実行する場合に使用されるプロセッサをユーザが選択できるようにする目的で使用されていました。従来は nactions パラメータで、動作ブロックの数を制御できましたが、現在では、必要に応じてカーネルが naction ブロックを動的に割当てます。nactions パラメータは、互換性確保のため引き続き /var/sysgen/mtune ファイルに残されていますが、実質的な効果はありません。

スイッチ・パラメータ

次に示すパラメータは、特権ユーザのユーザ ID を設定するシェルを許可するかどうかなど、カーネル内の特定の機能に対する許可をオンまたはオフにする単純な切替えを行います。

- panic_on_sbe は、特定のファクトリ・デバッグ・モードを制御します。

- `sbe_log_errors` は、シングル・ビット・エラーのシステム・ログを制御します。
- `sbe_mfr_override` は、シングル・ビット・エラーを無効にするデフォルトのアクションをオーバーライドします。
- `sbe_report_cons` は、シングル・ビット・エラーのコンソールへの通知を制御します。
- `corepluspid` は、コア・ファイル名を制御します。
- `r4k_div_patch` は、`r4kpp` のバイナリのパッチ・コードを制御します。
- `mload_auto_rtsyms` は、カーネル・シンボル・テーブルのロードを制御します。
- `xpg4_sticky_dir` は、特権のあるディレクトリ (sticky directory) のファイル削除を制御します。
- `tty_auto_strhold` は、`tty/pty` における STRHOLD の設定を制御します。
- `reset_limits_on_exec` は、新しい `setuid` プロセスの `rlimit` 値のリセットを制御します。
- `ip26_allow_ucmem` は、POWER Indigo2 システムのキャッシュ・システム・メモリへのアクセスを制御します。
- `restrict_fastprof` は、ユーザがユーザ・レベルの高速なプロファイリング (1 ミリ秒) が行なえるかどうかを制御します。
- `reboot_on_panic` は、カーネル・パニック後のシステムが自動的に再起動するように指定します。
- `svr3pipe` は、SVR3.2 または SVR4 のどちらのパイプを使用するかを制御します。
- `nosuidshells` は、0 に設定すると、アプリケーションから特権ユーザの特権シェルを生成できません。0 以外の値を設定すると、このシェルを生成できません。
- `posix_tty_default` は、デフォルトの SGI 回線制御手順を使用するかどうかを指定します。0 に設定すると、デフォルトの SGI 回線制御手順を使用します。1 に設定すると、POSIX の回線制御手順と設定値を使用します。
- `restricted_chown` は、`chown(2)` システム・コールに関して、BSD UNIX または System V のどちらの形式を使用するかを決定します。

- `use_old_serialnum` は、1 に設定すると、カーネルは `sysinfo -s` で 32 ビット・シリアル番号を計算する場合に、旧方式を採用します。このパラメータは、Onyx と Challenge L または XL のシステムにのみ影響します。
- `subnetsarelocal` は、1 に設定すると、サブネット化されたネットワークで直接接続されたものをローカルとみなすかどうかを指定します。
- `process_core_cpu_logging` は、サービス調整用の特殊なデバッグ・モードを制御します。このモードでは、プロセスのコア・ダンプの原因となる例外が発生したときに CPU がログに書込むことができます。

上記のパラメータはシステム全体の共通の値として使用します。プロセス単位で別々の値を指定できません。

panic_on_sbe

説明	<code>panic_on_sbe</code> パラメータの値を 1 に設定すると、シングル・ビット・エラーと呼ばれる特殊なサービス調整用デバッグ・モードが有効になります。0 に設定すると、この機能は無効になります。
値	デフォルト : 0 (0x0) 範囲 : 0 または 1

sbe_log_errors

説明	<code>sbe_log_errors</code> パラメータの値を 1 に設定すると、システムは SYSLOG に、シングル・ビット・エラーをログとして残します。0 に設定すると、この機能は無効になります。
値	デフォルト : 0 (0x0) 範囲 : 0 または 1
注記	このパラメータは、ECC メモリが搭載されているシステムだけに適用されません。Challenge と Onyx など、R8000 ベースのシステムです。

sbe_mfr_override

説明	sbe_mfr_override パラメータの値を 1 に設定すると、シングル・ビット・エラー・レートが事前に設定した制限値を超えたとき、シングル・ビット・エラーを無効にするデフォルトは無視されます。0 に設定すると、この機能は無効になります。
値	デフォルト : 0 (0x0) 範囲 : 0 または 1
注記	このパラメータは、ECC メモリが搭載されているシステムだけに適用されません。Challenge と Onyx など、R8000 ベースのシステムです。

sbe_report_cons

説明	sbe_report_cons パラメータの値を 1 に設定すると、シングル・ビット・エラーをシステム・コンソールに表示するようにシステムに指示します。0 に設定すると、この機能は無効になります。
値	デフォルト : 0 (0x0) 範囲 : 0 または 1
注記	このパラメータは、ECC メモリが搭載されているシステムだけに適用されません。Challenge と Onyx など、R8000 ベースのシステムです。

corepluspid

説明	corepluspid パラメータの値を 1 に設定すると、コア・ファイル名にプロセス ID 値が含まれます。0 に設定すると、この機能は無効になります。
値	デフォルト : 0 (0x0) 範囲 : 0 または 1

r4k_div_patch

説明	r4k_div_patch パラメータの値を 1 に設定すると、r4kpp 処理されたバイナリの exec パッチ・コードが有効になります。このパッチは R4000 SC rev 2.2 および 3.0 で生じる分岐分割遅延スロット (divide in branch delay slot) 問題に対処するものです。0 に設定すると、この機能は無効になります。
値	デフォルト : 0 (0x0) 範囲 : 0 または 1

mload_auto_rtsyms

説明	mload_auto_rtsyms パラメータの値を 1 に設定すると、カーネルのランタイム・シンボル・テーブルの通常の自動ロードが有効になります。0 に設定されていると、この機能は無効になります。
値	デフォルト : 1 (0x1) 範囲 : 0 または 1

xpg4_sticky_dir

説明	xpg4_sticky_dir パラメータの値を 1 に設定すると、ファイルへの書込みアクセスを所有している、ディレクトリ上に特権が指定されている (sticky bit が指定されている) 場合はファイルを削除できません。0 に設定すると、このようなファイルを削除できます。
値	デフォルト : 1 (0x1) 範囲 : 0 または 1

tty_auto_strhold

説明	tty_auto_strhold パラメータの値を 1 に設定すると、回線規約が標準およびエコー・モードになっているかぎり STRHOLD は自動的に ttys/ptys に設定されます。それ以外のモードの場合、STRHOLD は自動的にクリアされます。0 に設定すると、この機能は任意に使用できます。
値	デフォルト : 0 (0x0)

範囲: 0 または 1

reset_limits_on_exec

説明 reset_limits_on_exec パラメータの値を 1 に設定すると、カーネルは **root** として実行されているプロセスの **rlimit** 値をリセットします。この結果 **root** 以外のプロセスによってリソースが制限されるのを回避します。0 に設定すると、この機能は無効となり、リソースの制限は更新されません。このパラメータを変更すると、セキュリティに影響する場合があります。

値 デフォルト: 1 (0x1)

範囲: 0 または 1

ip26_allow_ucmem

説明 ip26_allow_ucmem パラメータの値を 0 に設定すると、ユーザは IP26 (POWER Indigo2) の非キャッシュ・システム・メモリにアクセスできません。1 に設定すると、その機能が許可されます。

値 デフォルト: 0 (0x0)

範囲: 0 または 1

変更要件 このパラメータの値が 0 に設定された場合、非キャッシュ・メモリへアクセスするとシステム・パニックが生じます。その機能が許可されている場合は、メモリ・パフォーマンスが著しく低下します。

restrict_fastprof

説明 restrict_fastprof パラメータの値を 0 に設定すると、prof によって、ユーザはプログラムの高速プロファイリング (1 ミリ秒) を実施できます。1 に設定すると、その機能は許可されません。

値 デフォルト: 0 (0x0)

範囲: 0 または 1

reboot_on_panic

説明

reboot_on_panic パラメータの値を 1 に設定すると、カーネル・パニック発生時にシステムを自動的に再起動します。無人あるいはリモートで使用しているサーバや、そのほかのシステムには特に便利です。ユーザがシステムを再起動するために出向く必要がありません。

パラメータの値を 0 に設定したときは、システムをコンソールから再起動する必要があります。

IP19、IP20、IP21、IP22、IP25、IP26、IP27、IP28、IP30、IP32、および IP35 搭載のシステム（プロセッサ・タイプは hinv のリストで得られます）は、PROM モニタ環境変数 *rebound* に設定を記憶します。*rebound* 変数を設定し、reboot_on_panic パラメータを -1 に設定すると、システムは PROM 環境をチェックします。デスクトップ・システム（IP20、IP22、IP26、IP28、IP30、IP32）では、*rebound* 変数を *y* に設定した場合、システムは自動的に再起動します。*n* に設定した場合は、手動でシステムをリセットする必要があります。他のプラットフォーム（IP19、IP21、IP25、IP27、IP35）では、*rebound* 変数を *y* に設定した場合（IP19、IP21、IP25）、かつ reboot_on_panic パラメータがデフォルトの -1 の場合、システムは、設定が 1 のときと同様に動作します。*rebound* 変数を設定せず、reboot_on_panic パラメータを -1 に設定すると、システムは指定値を 0 であるとみなします。したがって自動的に再起動されません。*rebound* 変数は必ず reboot_on_panic パラメータの前にチェックされ、このパラメータをオーバーライドできることに注意してください。reboot_on_panic パラメータを 0 に設定し、*rebound* の値を *y* に設定した場合、有効な reboot_on_panic パラメータは 1 になり、システムは自動的に再起動します。

メモ：SGI サーバおよび大規模なグラフィカル・システムのデフォルトは、カーネル・パニック時に再起動する動作です。

値

デフォルト：-1（ハードウェア・プラットフォームの実装に応じて自動的に再起動する）またはプロセッサ・タイプによっては 0。値 1 でもシステムは自動的に再起動されます。

範囲：-1、0、1

変更要件 システム・パニックの後に、自動的に再起動させる場合は、このパラメータを変更します。

svr3pipe

説明 `svr3pipe` パラメータは、デフォルトの SGI 回線制御手順を使用するかどうかを指定します。このパラメータの値を 1 に設定すると、一方向の SVR3.2 形式パイプを指定し、0 に設定すると、双方向の SVR4 形式パイプを指定します。

値 デフォルト: 1 (SVR3.2 形式パイプ)
範囲: 0 または 1

変更要件 SVR4 のスタイル・パイプを利用する場合は、このパラメータを変更します。SVR3 パイプの方がパフォーマンスが高いため、デフォルトになっています。

nosuidshells

説明 プログラムの中には、動作するために特権ユーザの権限を必要とするものがあります。このような動作を実行するために、アプリケーションはユーザが特権ユーザ権限を所有するシェルを作成します。このようなシェルにより多少のシステム・セキュリティ上のリスクが考えられますが、一般的にアプリケーションの開発者は、このようなシェルで実行されるアプリケーションの動作の制限を考慮します。`nosuidshells` が 0 に設定されている場合には、アプリケーションが特権ユーザの権限を持つシェルを作成します。0 以外の値に設定されている場合には、このようなシェルを作成しません。

値 デフォルト: 1 (setuid シェル生成禁止)

変更要件 setuid シェルを許可する場合に変更します。

posix_tty_default

説明 `posix_tty_default` パラメータは、シリアル回線に関する回線制御手順と設定のデフォルトとして IRIX システム値を使用するかどうかを指定します。このデフォルト値は、POSIX が指定するものとは異なります。このスイッチを 0 に設定すると、デフォルトの SGI 回線制御手順を使用し、1 に設定すると、POSIX 回線制御手順とその設定値を使用します。

値	デフォルト : 0 範囲 : 0 または 1
変更要件	POSIX 回線制御手順を使用する場合に変更します。

restricted_chown

説明	<code>restricted_chown</code> パラメータは、 <code>chown(2)</code> システム・コールについて BSD UNIX または System V のどちらの形式を使用するかを決定します。BSD UNIX 形式の場合は、特権ユーザだけがこのシステム・コールを使用してファイルを渡す（ファイルの所有者をほかのユーザに変更する）ことができます。System V 形式の場合は、どのユーザでもファイルやディレクトリを渡すことができます。0 に設定すると System V 形式、1 に設定すると BSD UNIX 形式が選択されます。
値	デフォルト : 0 範囲 : 0 または 1
変更要件	このスイッチは、 <code>chown(2)</code> システム・コールに対して希望する動作を選択する場合に変更します。

use_old_serialnum

説明	1 に設定した場合、 <code>use_old_serialnum</code> パラメータは、 <code>sysinfo -s</code> に対して 32 ビット・シリアル番号を計算するときに強制的に古い方法（IRIX パージョン 5 よりも前）を使用します。この変数は Onyx および Challenge L または XL システムのみに影響します。
値	デフォルト : 0 範囲 : 0 または 1
変更要件	Challenge または Onyx システムで 32 ビット・シリアル番号が必要な古いソフトウェアを使用する必要がある場合、このパラメータを変更します。

subnetsarelocal

説明	subnetsarelocal パラメータを 1 に設定した場合、直接接続されているサブネット化されたネットワークで他のすべてのネットワークがローカルと見なされます。
値	デフォルト: 0 範囲: 0 または 1
変更要件	サブネット化されたシステムが、インターネットのような外部ネットワークに直接接続されていない場合は、このパラメータを変更します。

process_core_cpu_logging

説明	1 に設定した場合、process_core_cpu_logging は、サービス調整用デバッグ・モードを有効にします。このモードでは、プロセスのコア・ダンプの原因となる例外が発生したときに CPU がログに書込むことができます。デフォルトではこの機能はオフ (値 0) です。この機能は、ハードウェアの異常を検出する場合に便利です。
値	デフォルト: 0 (0x0) 範囲: 0 または 1
注記	syslog メッセージは、「Process [<program name>] <pid> core dumped from trap generated on CPU <cpu number>」というフォーマットで生成されます。ここで、<program name> はプロセスの名前、<pid> はプロセスのプロセス ID、<cpu number> は例外がトラップされた物理的な CPU を示す整数です。

タイマ・パラメータ

タイマ・パラメータは、システム・クロックとタイミング機構の機能を制御します。以下のパラメータがあります。

- fasthz は、プロファイリング / 高速 itimer クロック速度を設定します。
- itimer_on_clkcpu は、itimer 要求がクロック・プロセッサの待ち行列に入るか、実行プロセッサの待ち行列に入るかを決定します。

- `timetrim` は、符号付きナノ秒を指定します。システム・クロックは、指定された符号付きナノ秒によって 1 秒ごとに調節されます。

fasthz

説明	<code>fasthz</code> パラメータは、プロファイリング / 高速 <code>itimer</code> クロック速度を設定します。 <code>fasthz</code> は <code>systune</code> コマンドでも変更できます。
値	デフォルト : 1000 範囲 : 500 ~ 2500
変更要件	このパラメータを変更することによって、 <code>gettimeofday</code> 、 <code>getitimer</code> 、および <code>settimer</code> などのシステム・コールのクロックを調節できます。
注記	Indigo、Indigo2、および Indy システムでは、許容値は 500、1000、または 2500 (Hz) です。これ以外の値を指定すると、デフォルト値の 1000 が使用されます。 Challenge システムでは、ハードウェア・タイマーの精度は 21 ナノ秒です。したがって 47 (MHz) 未満の任意の値を指定できます。 Octane システムでは、 <code>fasthz</code> 値を 1000 にする必要があります。 <code>timers(5)</code> マン・ページを参照してください。

itimer_on_clkcpu

説明	<code>itimer_on_clkcpu</code> パラメータの値を 0 または 1 に設定し、 <code>itimer</code> 要求をクロック・プロセッサの待ち行列に入れるか、または実行プロセッサの待ち行列に入れるかを決定します。
値	デフォルト : 0 範囲 : 0 または 1
変更要件	プロセスが <code>gettimeofday</code> を使用し、 <code>itimer</code> の精度を比較する場合は、このパラメータの値を 1 に設定し、クロック・プロセッサを使用します。 <code>itimer</code> 要求がユーザ周波数ベースのスケジューラを実装するためのものである場合は、このパラメータの値を 0 に設定し、その要求を現在実行中のプロセッサの待ち行列に設定します。

timetrim

説明	timetrim パラメータは、符号付きナノ秒を指定します。システム・クロックは、指定された符号付きナノ秒によって 1 秒ごとに調節されます。調節範囲は、3 ミリ秒 (0.3%) です。timed と timeslave は、定期的に /var/adm/SYSLOG に適切な値を設定します。
値	デフォルト: 0 範囲: 0 ~ 1 秒の 0.3% (3 ミリ秒)
変更要件	timed と timeslave が示す値に変更します。処理を行う前に、関連するマン・ページを参照してください。

ネットワーク・ファイル・システム (NFS: Network File System) パラメータ

次のパラメータは、ネットワーク・ファイル・システム (NFS: Network File System) のカーネルの機能を制御します。これらのパラメータの値を小さくすると、システム・パフォーマンスが著しく低下する場合があります。

- portmap_timeout は、ポートマップのクエリ・タイムアウトを設定します。
- sm_timeout は、ステータス・モニタのタイムアウトを設定します。
- GraceWaitTime は、NLM グレース・ピリオドの待ち時間を設定します。
- first_retry は、ポートマップとの最初の接続における再試行回数を設定します。
- normal_retry は、ポートマップとの接続後の再試行回数を設定します。
- lockd_grace_period は、NMI タイムアウトのグレース・ピリオドを設定します。
- lock_share_requests は、シェア要求に対応する IRIX のファイル・ロックを適用します。
- lockd_blocking_thresh は、新しいデーモンを生成する前に、ブロックを可能にするデーモンの数を設定します。
- nfs_portmon を 0 に設定すると、クライアントは任意のポートを使用できます。1 に設定すると、特権ポートしか使用できません。
- svc_maxdupreqs は、キャッシュ NFS 要求数を設定します。

portmap_timeout

説明	portmap_timeout パラメータは、ポートマップ・クエリ・タイムアウトを 1/10 秒刻みで指定します。
値	デフォルト : 5 (0x5) 範囲 : 1 ~ 200 (20 秒)
注記	タイムアウトを小さくすると、システム・パフォーマンスを著しく低下させる場合があります。

sm_timeout

説明	sm_timeout パラメータは、ステータス・モニタの通信タイムアウトを 1/10 秒刻みで指定します。
値	デフォルト : 5 (0x5) 範囲 : 1 ~ 150 (15 秒)
注記	ノーマル・タイムアウトとワーキング・タイムアウトを小さくすると、システム・パフォーマンスを著しく低下させる場合があります。

GraceWaitTime

説明	GraceWaitTime パラメータは、NLM グレース・ピリオドを秒数で指定します。
値	デフォルト : 5 (0x5) 範囲 : 1 ~ 60 (1 分)
注記	ノーマル・タイムアウトとワーキング・タイムアウトを小さくすると、システム・パフォーマンスが著しく低下する点に注意してください。

first_retry

説明	first_retry パラメータは、ポートマップとの最初の接続における再試行回数を指定します。
----	--

値 デフォルト : 1 (0x1)
 範囲 : 0 ~ 10000

normal_retry

説明 normal_retry パラメータは、ポートマップとの接続後の再試行回数を指定します。

値 デフォルト : 1 (0x1)
 範囲 : 0 ~ 10000

lockd_grace_period

説明 lockd_grace_period パラメータは、NLM のグレース・ピリオドを秒数で指定します。

値 デフォルト : 45 (0x2d)
 範囲 : 1 ~ 3600 (1 時間)

lock_share_requests

説明 lock_share_requests パラメータは、システムがシェア要求に対応する IRIX のファイル・ロックを適用するかどうかを指定します。

値 デフォルト : 0 (0x0)
 範囲 : 0 または 1

lockd_blocking_thresh

説明 lockd_blocking_thresh パラメータは、新しいデーモンが実行される前に、ブロックを可能にするデーモンの数を指定します。

値 デフォルト : 0 (0x0)
 範囲 : 0 ~ 1000

nfs_portmon

説明	nfs_portmon パラメータは、クライアントが NFS 要求に対して特権ポートを使用する必要があるかどうかを指定します。特権ユーザの権限を持つプロセスしか特権ポートと結合できません。nfs_portmon パラメータは 2 進数で、0 に設定すると、クライアントは任意のポートを使用できますが、1 に設定すると、特権ポートしか使用できません。
値	デフォルト : 0 範囲 : 0 または 1
変更要件	NFS がマウントされているファイルシステムでルート特権の保持が絶対に必要で、NFS クライアントが特権ポートを要求することが確実な場合にのみ、このパラメータを変更します。非特権ポートを要求するクライアントがいる場合は、このファイルシステムをマウントできません。 nfs_portmon パラメータの値を 1 に変更すると、セキュリティが不正になります。プロセスが特権ポートと結合するにはルート特権が必要ですが、1 台の不正なマシンの存在によって、この特権チェックのセキュリティが損なわれる危険性があります。

svc_maxdupreqs

説明	svc_maxdupreqs パラメータは、キャッシュ NFS 要求を指定します。
値	デフォルト : 409 範囲 : 400 ~ 4096
変更要件	最初の再送要求を受信したときにすぐ応答が行われるように、サービス負荷に合わせてこのパラメータを調節します。

ソケット・パラメータ

ここでは、ソケット・パラメータについて説明します。UDS ソケットと TCP/IP ソケットについて説明します。

UNIX ドメイン・ソケット (UDS: UNIX domain socket) 環境では、各ソケットについて 1 対のバッファがあります。つまり、ソケットの受信側に 1 個、送信側に 1 個あります。このバッファのサイズは、待ち行列に入るデータの最大量を示しています。バッファの動作は、ソケットがストリーム型かデータグラム型かによって異なります。

ストリーム・ソケットでは、データを送信すると、データは受信プロセスの受信バッファに入ります。受信バッファが一杯になると、送信領域バッファに入ります。両方のバッファが一杯になると、ソケットは送信されたデータをブロックします。

データグラム・ソケットでは、受信バッファが一杯になると、それ以降送信されるデータグラムを削除し、エラー EWOULDBLOCK を出力します。このような処理を行うので、デフォルトの受信バッファ・サイズは送信バッファ・サイズの 2 倍です。

多くの接続を持つネットワークのパフォーマンスを改善するために、TCP パケットと UDP パケットの検索にはハッシュ法を使用します。検索コードによって使用されるハッシュ・バケット数は、通常システムの起動時に計算され、その個数はシステム RAM が使用する MB 数に相当する分だけ割当てられます。UDP が使用するバケットはシステム RAM で 1 MB あたり 4 個、予備として 1 個を追加します。TCP は、システム RAM で 1 MB あたり 8 個、予備として 1 個を追加します。

各ハッシュ・バケットは、32 ビットのシステムでは 24 バイトの RAM を、64 ビットのシステムでは 48 バイトの RAM を必要とします。たとえば、128 MB の Challenge XL システムでは、UDP は 513 個のハッシュ・バケットを使用し、TCP は 1,025 個のハッシュ・バケットを使用します。デフォルトの自動設定を使用している場合は、バケットあたり 48 バイトのとき、ハッシュ・テーブルの情報を保持するために、総計で 73,824 バイトのシステム RAM が使用されます。64 MB の Indy では、メモリ・サイズもワード・サイズも小さいので、デフォルト値での総メモリ消費量は、システム RAM で 18,456 バイトになります。

デフォルトのサイズが各自のシステムに最適でない場合は、`udp_hashtablesz` パラメータと `tcp_hashtablesz` パラメータを使用して、テーブル・サイズを変更できます。指定された値に対し、カーネルが制限を課しています。このサイズには、各 TCP ソケットや UDP ソケットが存続している間は領域は動的に割当てられません。これらのパラメータは `inpccb` パラメータの一部で、`sysctl(1M)` によって修正されます。

次のパラメータは、ソケットを制御します。

- `unpst_sendspace` は、ストリーム型送信バッファ・サイズを指定します。

- `unpst_recvspace` は、ストリーム型受信バッファ・サイズを指定します。
- `unpdg_sendspace` は、データグラム型送信バッファ・サイズを指定します。
- `unpdg_recvspace` は、データグラム型受信バッファ・サイズを指定します。
- `udp_hashtablesz` は、ハッシュ・テーブル・サイズを指定します。
- `tcp_sendspace` は、TCP/IP ソケット型送信バッファ・サイズを指定します。
- `tcp_recvspace` は、TCP/IP ソケット型受信バッファ・サイズを指定します。
- `tcp_hashtablesz` は、TCP/IP ハッシュ・テーブル・サイズを指定します。
- `tcp_sack` は、TCP の選択的承認を有効にします。

unpst_sendspace

説明	<code>unpst_sendspace</code> パラメータは、ストリーム・ソケット型送信バッファのデフォルトのサイズを制御します。
値	デフォルト : 0x4000 (16 KB)
変更要件	<p>このパラメータを変更すると、すべてのストリーム・ソケットの送信バッファが変更され、多くのカーネル・メモリを使用することになるため、通常、各ソケットに対して、バッファ・サイズを変更するようにしてください。また、このパラメータの値を増やすと、すべてのソケットで読み込み用のバッファ領域を必要とするため、ソケット応答を待つ時間が長くなります。</p> <p>各ソケットのオプション設定についての詳細は、<code>setsockopt(2)</code> マン・ページを参照してください。</p>

unpst_recvspace

説明	<code>unpst_recvspace</code> パラメータは、ストリーム・ソケット型受信バッファのデフォルト・サイズを制御します。
値	デフォルト : 0x4000 (16 KB)

変更要件	<p>このパラメータを変更すると、すべてのストリーム・ソケットの受信バッファが変更され、多くのカーネル・メモリを使用することになるため、通常各ソケットに対して、バッファ・サイズを変更するようにしてください。また、このパラメータの値を増やすと、すべてのソケットで読み込み用のバッファ領域を必要とするため、ソケット応答を待つ時間が長くなります。</p> <p>各ソケットのオプション設定についての詳細は、<code>setsockopt(2)</code> マン・ページを参照してください。</p>
------	---

unpdg_sendspace

説明	<p><code>unpdg_sendspace</code> パラメータは、ソケットを通じて送信できるデータグラムのサイズを制御します。</p>
値	<p>デフォルト: 0x2000 (8 KB)</p>
変更要件	<p>データグラム・ソケットの動作は、ストリーム・ソケットと多少異なります。ストリーム・ソケットの場合、受信バッファが一杯になると、すべてのデータは送信バッファに送信され、送信バッファが一杯になると、エラー・メッセージが発行されます。データグラム・ソケットの場合は、受信バッファが一杯になると、これ以降のデータグラムは破棄され、エラー <code>EWOULDBLOCK</code> が出力されます。したがって、このパラメータは、受信バッファが受信できる大きさにデータグラムのサイズを制限する場合にだけ変更します。</p> <p>データグラム・ソケットの受信バッファのデフォルトのサイズは送信バッファの2倍になるので、外部からはストリーム・ソケットと同じ動作に見えます。</p> <p>通常、データグラム・ソケットの受信バッファのデフォルトのサイズを変更せずに、このパラメータだけを変更しないようにしてください。受信バッファのサイズ (<code>unpdg_recvspace</code>) を大きくせずにこのパラメータの値を大きくすると、受信バッファの受信量を超えるデータ量を送信することになります。各バッファ・サイズは <code>setsockopt(2)</code> システム・コールで設定してください。各ソケットのオプション設定についての詳細は、<code>setsockopt(2)</code> マン・ページを参照してください。</p>

unpdg_recvspace

説明	unpdg_recvspace パラメータは、データグラム・ソケット型受信バッファのデフォルトのサイズを制御します。
値	デフォルト : 0x4000 (16 KB)
変更要件	<p>このパラメータを変更すると、すべてのデータグラム・ソケットの受信バッファが変更され、多くのカーネル・メモリを使用することになるので、通常、各ソケットに対して、バッファ・サイズを変更するようにしてください。また、このパラメータの値を増やすと、すべてのソケットで読み込み用のバッファ領域を必要とするため、ソケット応答を待つ時間が長くなります。</p> <p>各ソケットのオプション設定についての詳細は、setsockopt (2) マン・ページを参照してください。</p>

udp_hashtablesz

説明	udp_hashtablesz パラメータは、UDP ハッシュ・テーブルのサイズを制御します。
値	デフォルト : 0 (自動設定) 範囲 : 64 ~ 2048
変更要件	<p>通常、このパラメータは変更する必要がありませんが、大規模なネットワーク・アプリケーション処理には使用されないシステム RAM の消費を減少する場合、基本的にネットワーキング・サービスを提供するシステムのハッシュ・バケットの数を増やす場合には、この値を変更する必要があります。値が 0 の場合は、システム RAM に基づいてテーブルの大きさが決定されます。0 以外の値の場合は、UDP に使用されるハッシュ・バケットの数を指定します。</p> <p>システムは、ハッシュ・アルゴリズムの条件を満たすために、この数を自動的に調節します。たとえば、512 を入力すると、513 のバケットが作成されます。</p>
制限	システムに適用するハッシュ・バケットの値は任意ですが、最小値 64、最大値 2,048 の範囲内が適切です。

tcp_sendspace

説明	<p>tcp_sendspace パラメータは、TCP/IP ソケットの送信バッファのデフォルト・サイズを制御します。使用中の各ソケットに対し、1 対のバケットがあります。つまり、ソケットの受信側のバッファと送信側のバッファです。これらのバッファのサイズは、待ち行列に格納するデータの最大量を示します。</p>
値	<p>デフォルト : 60 KB</p> <p>範囲 : 0 ~ 512 KB</p>
変更要件	<p>tcp_sendspace パラメータは、スループット・リンクが遅い場合や、4.2BSD に基づく TCP の実装と互換性をとる必要がある場合に調整します。低速 PPP や SLIP 接続など、遅いスループット・リンクでは、この値とシステムの tcp_recvspace パラメータ値を、最適なパフォーマンス・レベルになるまで下げます。モデムによる PPP や SLIP などの非常に遅いリンクでは、通常、3 KB から 4 KB のバッファ・サイズが最も効率的です。</p> <p>tcp_sendspace パラメータは、スループット・リンクが遅い場合や、4.2BSD に基づく TCP の実装と互換性をとる必要がある場合に調整します。低速 PPP や SLIP コンポーネントを使用した WAN (Wide Area Network) など遅いスループット・リンクでは、この値とイーサネットや FDDI などの高速から PPP または SLIP などの低速に接続方式をとるシステムの tcp_sendspace パラメータを、最適なパフォーマンスになるまで下げます。モデムによる PPP や SLIP など、非常に遅いリンクでは、通常、3 KB から 4 KB のバッファ・サイズが最も効率的です。</p> <p>高速接続では、バッファ・サイズを 60 KB から、最大のパフォーマンスが達成されるまで増やすことにより、その効果を得ることができます。</p> <p>特定のソケットのオプション設定についての詳細は、setsockopt(2) マニュアルページを参照してください。</p>

tcp_recvspace

説明	<p>tcp_recvspace パラメータは、TCP/IP 受信バッファのデフォルトのサイズを制御します。使用中の各ソケットに対して 1 対のバッファがあります。つまり、ソケットの受信側のバッファと送信側のバッファです。これらのバッファのサイズは、待ち行列となるデータの最大量を示します。</p>
----	---

値	デフォルト : 60 KB 範囲 : 0 ~ 512 KB
変更要件	<p><i>tcp_recvspace</i> パラメータは、スループット・リンクが遅い場合や、4.2 BSD に基づく TCP の実装と互換性をとる必要がある場合に調整します。低速 PPP や SLIP コンポーネントを使用した WAN (Wide Area Network) など遅いスループット・リンクでは、この値とイーサネットや FDDI などの高速から PPP または SLIP などの低速に接続方式をとるシステムの <i>tcp_sendspace</i> パラメータを、最適なパフォーマンスになるまで下げます。モデムによる PPP や SLIP など、非常に遅いリンクでは、通常、3 KB から 4 KB のバッファ・サイズが最も効率的です。</p> <p>高速接続では、バッファ・サイズを 60 KB から、最大のパフォーマンスが達成されるまで増やすことにより、その効果を得ることができます。</p> <p>特定のソケットのオプション設定についての詳細は、<code>setsockopt(2)</code> マニュアルページを参照してください。</p>

tcp_hashtablesz

説明	<i>tcp_hashtablesz</i> パラメータは、TCP ハッシュ・テーブルのサイズを制御します。
値	デフォルト : 0 (自動設定) 範囲 : 64 ~ 8192
変更要件	<p>通常、<i>tcp_hashtablesz</i> パラメータは変更する必要がありませんが、大規模なネットワーク・アプリケーション処理には使用されないシステム RAM の消費を減少する場合、基本的にネットワーキング・サービスを提供するシステムのハッシュ・バケットの数を増やす場合には、この値を変更する必要があります。</p> <p>値が 0 の場合は、システム RAM に基づいてテーブルの大きさが決定されます。0 以外の値の場合は、TCP に使用されるハッシュ・バケットの数の指定になります。システムは、ハッシュ・アルゴリズムの条件を満たすためにこの数を自動的に調節します。たとえば、512 を入力すると、1,025 のバケットが作成されます。TIME-WAIT 状態の接続を処理するために、TCP が追加の領域を使用するからです。</p>

制限 システムに適用するハッシュ・バケットの値は任意ですが、最小値 64、最大値 8,192 の範囲内が適切です。

tcp_sack

説明 tcp_sack パラメータは、RFC 2018 により TCP の選択的承認を有効にします。この機能によって、エラーが発生しやすい接続や損失のある接続で TCP のパフォーマンスを大幅に向上させることができます。

値 デフォルト : 1
範囲 : 0、1

変更要件 このパラメータは、不明な TCP オプションを正しく処理できない古いネットワーク・ギアがある場合にのみ無効にします (0 に設定)。

VINO パラメータ

ここでは、Indy Video In No Out (VINO) チューニング・パラメータ vino_mtune_dmrpages について説明します。vino_mtune_dmrpages は、VINO ビデオ・キャプチャによって使用される DMA 記述子のサイズを制限します。

vino_mtune_dmrpages

説明 vino_mtune_dmrpages は、VINO ビデオ・キャプチャによって使用される DMA 記述子テーブルのサイズを制限します。また、デジタル・メディア・リング・バッファ (DMRB) に割当てたページが、ビデオをキャプチャする場合に使用できるページ数を制限します。vino_mtune_dmrpages が示すページ数は、VINO 専用の DMRB サイズが割当てられる場合に制限される値となります。

値 デフォルト : 0 (自動設定)

このパラメータの値を 0 に設定すると、実行時に使用できるメモリの量に基づくデフォルトが使用されます。適切なサイズよりも、小さいサイズに制限する場合は、リング・バッファが使用できる最大のページ数にこの変数を設定します。値が -1 の場合、VINO の DMA 記述子に対して事前に領域を割当ててはいけないことを示します。

注記 DMA 記述子のテーブル・サイズが指定よりも大きなサイズを必要とし、実行時に連続した領域として割当てが行えなかった場合は、ビデオ転送されず、ENOMEM エラーを返します。割当てができない原因としてメモリの断片化があります。エラーを回復するには、より小さな DMRB でキャプチャを試みてください。

ラージ・ページ・パラメータ

ラージ・ページは、事前に割当てを行い予約する必要があります。nlpages_x パラメータで指定された値は、サイズ x のラージ・ページ数を参照します。この値が設定されている場合には、カーネルは適切なサイズで指定した数のページを割当てます。

次に示すパラメータを使用して、ラージ・ページの割当てを行います。

- nlpages_256k は、割当てられる 256 KB ページの数を指定します。
- nlpages_1m は、割当てられる 1 MB ページの数を指定します。
- nlpages_4m は、割当てられる 4 MB ページの数を指定します。
- nlpages_16m は、割当てられる 16 MB ページの数を指定します。

nlpages_256K

説明 nlpages_256k パラメータは、割当てられる 256 KB ページの数を指定します。

値 デフォルト: 0
範囲: 0 ~ 64

nlpages_1m

説明	nlpages_1m パラメータは、割当てられる 1 MB ページの数を指定します。
値	デフォルト: 0 範囲: 0 ~ 64
変更要件	64 プロセッサまたはそれ以上のシステムで推奨する初期値は 0 です。

nlpages_4m

説明	nlpages_4m パラメータは、割当てられる 4 MB ページの数を指定します。
値	デフォルト: 0 範囲: 0 ~ 64

nlpages_16m

説明	nlpages_16m パラメータは、割当てられる 16 MB ページの数を指定します。
値	デフォルト: 0 範囲: 0 ~ 64

拡張アカウントティング・パラメータ

次のパラメータは、システム・アカウントティング機能を制御します。

- do_procacct は、BSD のプロセス・アカウントティングを制御します。
- do_extpacct は、拡張プロセス・アカウントティングを制御します。
- do_sessacct は、Array セッション・アカウントティングを制御します。
- min_local_paggid は、カーネルによって割当てられる最初の Array セッション・ハンドルを制御します。
- max_local_paggid は、min_local_paggid に戻る (ラップ・バックする) までにカーネルによって割当てることのできる最大 Array セッション・ハンドルを制御します。

- `asmachid` は、グローバル ASH 生成用のマシン ID を制御します。
- `dfltprid` は、特殊セッション用のデフォルト・プロジェクト ID を制御します。

do_procacct

説明	<code>do_procacct</code> パラメータの値を 1 に設定すると、システムは BSD のシステム・アカウントティングを実行します。0 に設定すると、システムは、 <code>acct(2)</code> システム・コールを無視し、アカウントティングを実行しません。
値	デフォルト : 1 範囲 : 0 ~ 1

do_extpacct

説明	<code>do_extpacct</code> パラメータの値を 1 に設定すると、システムは拡張プロセス・アカウントティングを実行します。0 に設定すると、システムはアカウントティングを実行しません。
値	デフォルト : 0 範囲 : 0 ~ 1

do_sessacct

説明	<code>do_sessacct</code> パラメータの値を 1 に設定すると、システムは Array セッション・アカウントティングを実行します。0 に設定すると、システムはアカウントティングを実行しません。
値	デフォルト : 0 範囲 : 0 ~ 1

min_local_paggid

説明	<code>min_local_paggid</code> パラメータは、カーネルが割り当てることができる最初の Array セッション・ハンドルを指定します。
----	--

値 デフォルト: 1
 範囲: 1 ~ 0x7fffffff00

max_local_paggid

説明 max_local_paggid パラメータは、min_local_paggid に戻るまでに、カーネルが割当てることができる最大の Array セッション・ハンドルを指定します。

値 デフォルト: 65535
 範囲: 255 ~ 0x7fffffff

asmachid

説明 asmachid パラメータは、グローバルな ASH を生成するためのシステム ID を指定します。Array 内のほかのシステムが、同じシステム ID を持つことはできません。0 に指定すると、カーネルはローカル ASH だけを生成します。

値 デフォルト: 0
 範囲: 0 ~ 0x7fff

dfltprid

説明 dfltprid パラメータは、特殊システム・セッションおよび通常のプロジェクト ID の割当てをスキップするほかのセッションに対し、デフォルトのプロジェクト ID を指定します。

値 デフォルト: 0
 範囲: 0 ~ 0xffffffffffff

NUMA パラメータ

次のパラメータは、Origin2000 システムや SGI 3000 ファミリーのサーバのような NUMA マシンでのみ有効なメモリ移行パラメータです。

- `numa_migr_default_mode` は、デフォルトの移行モードを定義します。
- `numa_migr_default_threshold` は、移行中断要求を生成するのに必要な、ローカル・カウンタと任意のリモート・カウンタの間の最小の差を定義します。
- `numa_migr_threshold_reference` は、メモリ参照カウンタのための固定値を定義します。
- `numa_migr_min_maxradius` は、移行モードがオンになる最小の *maxradius* を指定します。
- `numa_migr_min_distance` は、移行要求を受付けるためにノード距離フィルタ (Node Distance Filter) が必要とする最小距離を指定します。
- `numa_zone_radius_max` は、ゾーン割当てに使用される最大半径の値を指定します。
- `numa_migr_vehicle` は、ページの移行にシステムが使用するデバイスを定義します。
- `numa_refcnt_default_mode` は、デフォルトの拡張参照カウンタ・モードを定義します。
- `numa_refcnt_overflow_threshold` は、ハードウェア拡張参照カウンタがオペレーティング・システムにカウンタ・オーバーフローが発生していることを通知し、そのカウントをソフトウェア拡張参照カウンタに転送するためのカウント数を定義します。
- `numa_migr_memory_low_threshold` は、メモリ・プレッシャ・フィルタ (Memory Pressure Filter) がノードに対する移行要求の拒否を開始するしきい値を定義します。
- `numa_migr_memory_low_enabled` は、メモリ・プレッシャ・フィルタ (Memory Pressure Filter) を有効または無効に設定します。
- `numa_migr_freeze_enabled` は、バウンス・コントロール・フィルタ (Bounce Control Filter) 操作のフリーズを有効または無効に設定します。
- `numa_migr_freeze_threshold` は、ページがフリーズされるしきい値を定義します。
- `numa_migr_melt_enabled` は、バウンス・コントロール・フィルタ (Bounce Control Filter) 操作のフリーズの解凍を有効または無効に設定します。
- `numa_migr_melt_threshold` は、移行カウンタが指定したしきい値を下回ったときにフリーズを解凍します。
- `numa_migr_bounce_control_interval` は、移行カウンタまたは制動カウンタを時効にするループ期間を定義します。
- `numa_migr_dampening_enabled` は、移行制動を有効または無効に設定します。

- `numa_migr_dampening_factor` は、移行が実際に試行される前に、1 ページに必要なとされる移行要求の数を定義します。
- `mem_tick_enabled` は、`clock.c/` から移行コントロール定期操作 (Migration Control Periodic Operation) を実行するループを有効または無効に設定します。
- `mem_tick_base_period` は、1 `mem_tick` あたりの 10 ms system tick の数を定義します。
- `numa_migr_unpegging_control_enabled` は、非固定の定期操作を有効または無効に設定します。
- `numa_migr_unpegging_control_interval` は、ハードウェア・メモリ参照カウンタを非固定にするループ期間を定義します。
- `numa_migr_unpegging_control_threshold` は、システムがカウンタの固定を判断する、ハードウェア・メモリの参照カウンタの値を定義します。
- `numa_migr_traffic_control_enabled` は、トラフィック・コントロール・フィルタ (Traffic Control Filter) を有効または無効に設定します。
- `numa_migr_traffic_control_interval` は、トラフィック・コントロール期間を定義します。
- `numa_migr_traffic_control_threshold` は、待ち行列に入れられた移行要求をバッチ処理で開始する、トラフィック・コントロールのしきい値を定義します。

numa_migr_default_mode

説明	<code>numa_migr_default_mode</code> パラメータはデフォルトの移行モードを定義します。
値	デフォルト: 2 範囲: 0 ~ 4 0: <code>MIGR_DEFMODE_DISABLED</code> 移行を完全に無効に設定します。ユーザは移行モードを使用することができません。 1: <code>MIGR_DEFMODE_ENABLED</code> 移行を常に有効に設定します。ユーザは移行モードを無効にすることはできません。 2: <code>MIGR_DEFMODE_NORMOFF</code> 移行を通常オフに設定します。ユーザはアプリケーションに対する移行モードを有効に設定することができます。

3: *MIGR_DEFMODE_NORMON* 移行を通常オンに設定します。ユーザはアプリケーションに対する移行モードを無効に設定することができます。

4: *MIGR_DEFMODE_LIMITED* 次に説明する *numa_migr_min_maxradius* より少ない値で、最大の NUMA link 距離を持つマシン構成に対して、移行モードを通常オフに設定します。それ以外の場合には、通常オンに設定します。ユーザはこのモードをオーバーライドすることができます。

numa_migr_default_threshold

説明 *numa_migr_default_threshold* パラメータは、移行中断要求を生成するのに必要な、ローカル・カウンタと任意のリモート・カウンタの間の最小の差を定義します。

```
if ( (remote_counter - local_counter) >= ((numa_migr_threshold_reference_value / 100) * numa_migr_default_threshold)) { send_migration_request_intr(); }
```

値 デフォルト : 50

範囲 : 0 ~ 100

numa_migr_threshold_reference

説明 *numa_migr_threshold_reference* パラメータは、メモリ参照カウンタのための固定値を定義します。これはマシン構成に依存します。Origin2000 システムの場合には、次の値を使用します。

値 デフォルト : 0

範囲 : 0 ~ 1

0: *MIGR_THRESHREF_STANDARD* = しきい値参照は 2048 (11 ビット・カウンタ) です。この値は、STANDARD DIMMS を持つシステムで使用可能な最大しきい値になります。

1: *MIGR_THRESHREF_PREMIUM* = しきい値参照は 524288 (19 ビット・カウンタ) です。この値は、*all* premium SIMMS を持つシステムで使用可能な最大しきい値になります。

numa_migr_min_maxradius

説明 `numa_migr_min_maxradius` パラメータは、移行が開始される最小の *maxradius* を指定します。このモードは、`numa_migr_default_mode` がモード 4 (`MIGR_DEFMODE_LIMITED`) に設定されている場合に使用されます。このモードの場合、`numa_migr_min_maxradius` より少ない値で、最大の NUMA link 距離を持つマシン構成では通常移行モードはオフになります。これ以外の場合には、通常オフに設定されます。

値 デフォルト: 2

numa_migr_min_distance

説明 `numa_migr_min_distance` パラメータは、移行要求を受付けるためにノード距離フィルタ (Node Distance Filter) が必要とする最小距離を指定します。

値 デフォルト: 1

numa_zone_radius_max

説明 `numa_zone_radius_max` パラメータは、ゾーン割当てに使用される最大半径の値を指定します。

値 デフォルト: 2

numa_migr_vehicle

説明 `numa_migr_vehicle` パラメータは、ページの移行にシステムが使用するデバイスを選択します。値が 0 のときには、ブロック転送エンジン (BTE: Block Transfer Engine) を選択し、値が 1 のときには、プロセッサを選択します。BTE が選択されているときに、システムがオプションの `poison` ビットを装備している場合には、システムは自動的に Lazy TLB シュートダウン・アルゴリズムを使用します。

<code>MIGR_VEHICLE_IS_BTE</code>	0
<code>MIGR_VEHICLE_IS_TLBSYNC</code>	1

migr_vehicle の値は最初は BTE になります。システムに BTE がない場合には、NUMA 初期化コードがこれを TLBsync に変更します。さらに、プロセッサのリビジョンが BTE 使用に適さない場合にも、TLBsync に変更されます。

値 デフォルト : 1

numa_refcnt_default_mode

説明 numa_refcnt_default_mode パラメータは、デフォルトの拡張参照カウンタ・モードを定義します。次の値を使用します。

値 デフォルト : 0

範囲 : 0 ~ 3

0: *REFCNT_DEFMODE_DISABLED* 拡張参照カウンタを無効に設定します。ユーザは拡張参照カウンタにアクセスすることができません (*refcnt(5)* マニュアルを参照)。

1: *REFCNT_DEFMODE_ENABLED* 拡張参照カウンタを常に有効に設定します。ユーザはこれを無効にすることはできません。

2: *REFCNT_DEFMODE_NORMOFF* 拡張参照カウンタを通常無効に設定します。ユーザはアプリケーションに対してカウンタを有効または無効に設定することができます。

3: *REFCNT_DEFMODE_NORMON* 拡張参照カウンタを通常有効に設定します。ユーザはアプリケーションに対してカウンタを有効または無効に設定することができます。

numa_refcnt_overflow_threshold

説明 numa_refcnt_overflow_threshold パラメータは、ハードウェア拡張参照カウンタがオペレーティング・システムにカウンタ・オーバーフローが発生していることを通知し、そのカウントをソフトウェア拡張参照カウンタに転送するためのカウント数を定義します。これは、*numa_migr_threshold_reference* で定義される参照しきい値の割合で表します。

値 デフォルト : 50

numa_migr_memory_low_threshold

説明	numa_migr_memory_low_threshold パラメータは、メモリ・プレッシャ・フィルタ (Memory Pressure Filter) がノードに対する移行要求の拒否を開始するしきい値を定義します。このしきい値は、ノード上の物理メモリの総容量の割合で表します。
値	デフォルト: 20

numa_migr_memory_low_enabled

説明	numa_migr_memory_low_enabled パラメータは、メモリ・プレッシャ・フィルタ (Memory Pressure Filter) を有効または無効に設定します。
値	デフォルト: 1

numa_migr_freeze_enabled

説明	numa_migr_freeze_enabled パラメータは、バウンス・コントロール・フィルタ (Bounce Control Filter) 操作のフリーズを有効または無効に設定します。
値	デフォルト: 0

numa_migr_freeze_threshold

説明	numa_migr_freeze_threshold パラメータは、ページがフリーズされるしきい値を定義します。移行カウンタでサポートされる最大カウンタの割合で表します (Origin2000 システムの場合に 7%)。
値	デフォルト: 20

numa_migr_melt_enabled

説明	numa_migr_melt_enabled パラメータは、バウンス・コントロール・フィルタ (Bounce Control Filter) の解凍操作を有効または無効に設定します。
値	デフォルト: 0

numa_migr_melt_threshold

説明	numa_migr_melt_threshold パラメータは、移行カウンタが指定したしきい値を下回ったときにフリーズを解凍します。このパラメータは、移行カウンタでサポートされる最大カウンタの割合で表します (Origin2000 システムの場合は 7%)。
値	デフォルト : 90

numa_migr_bounce_control_interval

説明	numa_migr_bounce_control_interval パラメータは、移行カウンタまたは制動カウンタを時効にするループ期間を定義します。これは、 <i>mem_ticks</i> の数を使用して指定します。 <i>mem_ticks</i> の単位は、以下に説明する <i>mem_tick_base_period</i> で定義します。0 に設定した場合には、システムは <i>mem_tick</i> に対して 4 ページを処理します。この場合、実際の期間はノード上に存在する物理メモリの容量に依存します。
値	デフォルト : 0

numa_migr_dampening_enabled

説明	numa_migr_dampening_enabled パラメータは、移行制動を有効または無効に設定します。
値	デフォルト : 0

numa_migr_dampening_factor

説明	numa_migr_dampening_factor パラメータは、移行が実際に試行される前に、1 ページに必要なとされる移行要求の数を定義します。移行要求カウンタでサポートされる最大カウンタの割合で表します (Origin2000 の場合には、3%)。
値	デフォルト : 90

mem_tick_enabled

説明	mem_tick_enabled パラメータは、 <i>clock.c/</i> から移行コントロール定期操作 (Migration Control Periodic Operation) を実行するループを有効または無効に設定します。
値	デフォルト:1

mem_tick_base_period

説明	mem_tick_base_period パラメータは、1 <i>mem_tick</i> あたりの 10 ms system tick の数を定義します。
値	デフォルト:1

numa_migr_unpegging_control_enabled

説明	numa_migr_unpegging_control_enabled パラメータは、非固定の定期操作を有効または無効に設定します。
値	デフォルト:1

numa_migr_unpegging_control_interval

説明	numa_migr_unpegging_control_interval パラメータは、ハードウェア・メモリ参照カウンタを非固定にするループ期間を定義します。これは、 <i>mem_tick</i> の数を使用して表します。 <i>mem_tick</i> の単位は、 <i>mem_tick_base_period</i> で定義します。0 に設定した場合には、システムは <i>mem_tick</i> に対して 8 ページを処理します。この場合、実際の期間はノード上に存在する物理メモリの容量に依存します。
値	デフォルト:0

numa_migr_unpegging_control_threshold

説明	numa_migr_unpegging_control_threshold パラメータは、システムがカウンタの固定を判断する、ハードウェア・メモリの参照カウンタの値を定義します。numa_migr_threshold_reference で定義される最大カウンタの割合で表します。
値	デフォルト: 90

numa_migr_traffic_control_enabled

説明	numa_migr_traffic_control_enabled パラメータは、トラフィック・コントロール・フィルタ (Traffic Control Filter) を有効または無効に設定します。これはテスト用のモジュールであるため、常に無効に設定します。
値	デフォルト: 0

numa_migr_traffic_control_interval

説明	numa_migr_traffic_control_interval パラメータは、トラフィック・コントロール期間を定義します。これは実験モジュールです。
値	デフォルト: 400

numa_migr_traffic_control_threshold

説明	numa_migr_traffic_control_threshold パラメータは、待ち行列にある移行要求をバッチ処理で開始するためのトラフィック・コントロールのしきい値を定義します。これは試用モジュールです。
値	デフォルト: 80

ページ複製パラメータ

ページ複製パラメータは numa_kernel_replication_ratio で、メモリ・ノード数とカーネル・テキスト・イメージ数の比率を指定します。

numa_kernel_replication_ratio

説明	numa_kernel_replication_ratio パラメータは、メモリ・ノード数に対するカーネル・テキスト・イメージ数の比率を指定します。マスター・ノードには必ずコピーが存在し、ほかのノードには n ノードおきにコピーが作成されます。 n 値として 0 を指定すると、複製されません。比率値 n を指定すると、 n ノードごとにコピーが作成されます。デフォルト値 1 は、カーネル・テキストのコピーがすべてのノードに作成されます。
説明	numa_repl_traffic_highmark_percentage パラメータは、複製トラフィックが使用可能なフル・ユーティライゼーションの割合を指定します。
値	デフォルト: 1

移行メモリ・キュー・パラメータ

次のパラメータは、移行メモリ・キューを制御します。

- numa_migr_coaldmigr_mech は、メモリを合体させて移行するための移行実行モードを定義します。
- numa_migr_user_migr_mech は、ユーザ要求の移行のための移行実行モードを定義します。
- numa_migr_auto_migr_mech は、メモリ参照のカウンタ・トリガ移行のための移行実行モードを定義します。

numa_migr_coaldmigr_mech

説明	numa_migr_auto_migr_mech パラメータは、メモリを合体させて移行するための移行実行モードを定義します。0 は即実行を示し、1 は遅延実行を示します。現在有効なモードは、即実行の 0 のみです。
値	デフォルト: 0 範囲: 0 ~ 1

numa_migr_user_migr_mech

説明	numa_repl_mem_lowmark パラメータは、ユーザが要求した移行を実行するための移行実行モードを定義します。0 は即実行を示し、1 は遅延実行を示します。現在有効なモードは、即実行の 0 のみです。
値	デフォルト: 0 範囲: 0 ~ 1

numa_migr_auto_migr_mech

説明	numa_migr_auto_migr_mech パラメータは、メモリ参照のカウンタ・トリガ移行のための移行実行モードを定義します。0 は即実行を示し、1 は遅延実行を示します。現在有効なモードは、即実行の 0 のみです。
値	デフォルト: 0 範囲: 0 ~ 1

XVM パラメータ

XVM ボリューム・マネージャ・チューニング・パラメータは、`var/sysgen/mtune/xvm` ファイルにあります。このファイルには、XVM 論理ボリュームのミラー回復リソースを制御するパラメータが含まれています。ミラー回復時のシステム・パフォーマンスが遅いと思われる場合は、これらのリソースを再構成する必要があります。XVM ミラー回復リソース・パラメータは動変数であるため、システムを再起動しなくても `systune(1M)` コマンドを使って変更できます。

以下の XVM ミラー回復リソース・パラメータを使用できます。

- `xvm_max_revive_threads` は、一度に回復するプロセスの数を制御します。
- `xvm_max_revive_rsc` は、回復プロセスで使われる並列 I/O プロセスの数を制御します。

一般的なガイドラインとして、以下の操作を行ってください。

- 通常の I/O プロセスに対するパフォーマンスの影響が許容範囲である場合、回復を最適化するには、スレッドとリソースを増やします。

- XLV と XVM のボリューム・マネージャが同一システムを共有している場合は、スレッドを減らします。
- 特定のファイルシステムに対する影響を少なくするには、リソースを減らします。

xvm_max_revive_threads

説明	xvm_max_revive_threads パラメータは、一度に回復するプロセスの数を制御します。
値	デフォルト: 4 範囲: 1 ~ 12
変更要件	XLV と XVM のミラー論理ボリュームが同一システムで使われている場合は、スレッドの数を減らします。これにより、XVM ボリューム・マネージャが XLV ボリューム・マネージャから過剰なリソースを取得するのを防ぐことができます。 並列実行される回復の数を増やすには、スレッドを増やします。

xvm_max_revive_rsc

説明	xvm_max_revive_rsc パラメータは、回復プロセスで使われる並列 I/O プロセスの数を制御します。
値	デフォルト: 5 範囲: 1 ~ 10
変更要件	並列 I/O プロセスの数を減らすと、開いているファイルシステムとの干渉は少なくなりますが、データの回復にかかる合計時間は長くなります。

CXFS パラメータ

CXFS では、以下のパラメータを利用できます。

- cms_reset_timeout
clconfd がリセット要求を確認するまで待機する秒数を指定します。

- `cms_reset_timeout_action`

`clconfd` がリセット要求を確認しない場合 (`cms_reset_timeout` によって指定) に実行する動作を指定します。

CXFS システム調整可能パラメータの詳細については、『CXFS Software Installation and Administration Guide』の第 6 章「管理とメンテナンス」を参照してください。

コア・ファイル・パラメータ

coremask

説明	<code>coremask</code> パラメータは <code>/var/sysgen/mtune/kernel</code> ファイルで設定し、コア・ファイルのパーミッションの種類を決定します。
値	デフォルト: 0 範囲: 最小 0 ~ 最大 0177
変更要件	現在のバージョンでは、プロセスがコア・ファイルを作成する場合、プロセスの所有者の <code>umask</code> を使用して、コア・ファイルのパーミッションを決定します。このため、機密情報が含まれるコア・ファイルが、どのユーザでも読取り可能なパーミッションで作成される可能性があります。この <code>sysune</code> 変数を使用して、システム管理者はコア・ファイルへのアクセスを制限できます。デフォルトは 0 で、現在の動作です。ただし、 <code>coremask</code> 変数を他の値に設定すると、 <code>umask</code> としてその数値が使用され、プロセスの所有者の <code>umask</code> は無視されます。たとえば、 <code>coremask</code> を 0177 に設定し、プロセスの所有者の <code>umask</code> が 022 の場合、コア・ファイルに設定されるパーミッションは 644 ではなく 600 になります。この変数は、入力した数字が 8 進数であると想定します。

システム・エラー・メッセージを使用したシステム構成に関するトラブルシューティング

付録 B では、IRIX オペレーティング・システムから受取るエラー・メッセージについて説明します。また、エラー・メッセージに関するシステム関数について説明する適切な参照資料を紹介しています。

システム・エラーが起こるといくつかのエラー・メッセージが現れ、実際にどんな問題が起こったかを認識できます。メッセージに関する各節を読んで、どのメッセージが最も重大な問題を示しているか、また、それがシステムのほかの部分に影響を与えていることを示すメッセージなのかを適確に判断してください。

このマニュアルおよび IRIX 管理マニュアルに記載されていないエラー・メッセージについては、`/usr/include/sys/errno.h` ファイル、`intro(2)` マン・ページ、およびシステムのユーザ・ガイドを参照してください。

一部のエラー・メッセージは、システムの設定変更によってカスタマイズされます。この付録で説明するメッセージでは例示できない固有の情報については、省略符号 (...) が付きます。このマニュアルに記載されたメッセージと完全に一致していないメッセージについては、類似のものであるという注釈が付きます。

ディスク領域に関するメッセージ

以下のメッセージは、標準的なディスク・オペレーションに関するものです。使用可能なディスク領域の低減や不足を示すメッセージなどがあります。

```
unix: <disk id>: Process ... ran out of disk space
```

```
unix: <disk id>: Process ... ran out of contiguous space
```

ディスクが一杯になると、新しくファイルを作成したり既存のファイルに情報を書き加えることができません。システム・ディスクが一杯になると、システムがコマンドに対して応答しなくなる場合があります。

メモ: ディスクに領域を確保せずに、システムを停止したり、再起動しないでください。ディスク領域に空きがないと、システムは正しく再起動できない場合があります。

以下のいずれかの方法でディスク領域を確保してください。

1. 「ツールチェスト (Toolchest)」の「デスクトップ (Desktop)」から「ゴミ箱を空にする (Empty Dumpster)」を選択し、ゴミ箱を空にしてください。
2. 古いファイル、または大きなファイルやディレクトリを削除するか、保存します。
 - 古いファイルや大きなファイルを検索するには、ラウンチ・アイコンをダブルクリックし、「ファイルの検索 (Search)」ツールを起動し、オンライン・ヘルプを参照してください。
 - `core` というファイル名を検索してください。このファイルは、問題を起こしたアプリケーションによって作成されることが多いので、サイズが非常に大きい場合があります。
 - デスクトップからファイルを消去した後、ゴミ箱を空にします。
 - ファイルを保存する (バックアップ・テープにコピーする) には、「バックアップ / リストア・マネージャ (Backup and Restore Manager)」ツールを使用します。ラウンチ・アイコンをダブルクリックし、このツールを起動します。
3. システム・ディスクがほぼ一杯の場合は、次のファイルやディレクトリをチェックしてください。
 - `/var/tmp` および `/tmp` は、頻繁に一杯になる共有ディレクトリです。不要なファイルやディレクトリを発見した場合は削除してください。
 - `/var/adm/SYSLOG` ファイルが非常に大きく、200 KB を超える場合は、数行を残してすべて消去します。ファイル全体を消去しないでください。
 - システムが重大な故障 (クラッシュ) を起こした場合、システムは情報を `vmcore.number` および `unix.number` の2つの `/var/adm/crash` ファイルに記録します。この名前が付いたファイルを発見した場合は、テープにバックアップを取り、

日本シリコングラフィックス株式会社のサポート部門までお問い合わせください。その後、システムから削除することができます。

- デスクトップからファイルを削除した後、再度ゴミ箱を空にします。
 - `mbox` は、すべてのホーム・ディレクトリにあります。このファイルが大きい場合は、重要なメッセージ以外は削除するようにファイルの所有者に依頼します。
4. オプションのソフトウェアやアプリケーション・ソフトウェアを削除します。
- 「ソフトウェア・マネージャ (Software Manager)」を起動するには、「ツールチェスト (Toolchest)」の「システム (System)」から「ソフトウェア・マネージャ (Software Manager)」を選択します。

ディスク使用率が 90 % になったときに通知されるなど、ディスク領域の変化を知りたい場合には、特権ユーザとして手順 B-1 に従ってください。

手順 B-1 ディスク使用率の通知

1. ラUNCH・アイコンをダブルクリックして、「ディスク・マネージャ (Disk Manager)」を起動します。
2. 変更したいしきい値を含むディスクの絵のボタンをクリックします。システム・ディスクは 0、1 とラベル付けされます。
3. 「警告 (Notify when)」フィールドに書かれた数字を選択して強調表示にします。90 と入力し、「OK」をクリックします。

143 ページの「ディスク使用量に関するコマンド」も参照してください。

一般的なシステム・メッセージ

以下のメッセージは、注意する必要がある一般的なシステム構成の問題を示します。

ファイルのパーミッションの問題

次のメッセージが表示されることがあります。

```
unix: WARNING: inode 65535: illegal mode 177777
```

このメッセージは、プログラムがアクセス権のないファイルにアクセスしようとしたことを示しています。

find コマンドを使用して、ファイル名、パーミッション、およびそのファイルがあるディレクトリ・パスを識別します。ファイルの位置を認識するには、このコマンドに **-inum** オプションと *i* ノード番号を指定してください。ファイルの位置がわかった場合は、ファイルの所有者あるいは特権ユーザ (**root**) がファイルのパーミッションを変更します。

IP (ネットワーク) アドレスの問題

次の問題は、ネットワーク・システムの基本構成、または、ネットワークを構成しているハードウェアに関連しています。

デフォルトのインターネット・アドレス

次のメッセージが表示されることがあります。

```
unix: network: WARNING IRIS'S Internet address is the default
```

このシステムには IP アドレスは設定されていません。IP アドレスを設定するには、79 ページの「ネットワーク・アドレスの設定」を参照してください。

IP アドレスの重複

次のメッセージが表示されることがあります。

```
unix: arp: host with ethernet address ... is using my IP address ...
```

```
unix: arp: host with ethernet address ... is still using my IP address ...
```

システムの IP アドレスが、ほかのシステムのもと同じです。ネットワークが正しい場所に情報を送信するには、ネットワーク上の各システムは、一意の IP アドレスでなければなりません。アドレスの競合は、新しいシステムがネットワークに接続された場合に、システム管理者がすでに使用されている IP アドレスを割当てたときなどに起きます。

システムが正しい IP アドレスを使用していることを確認してから、もう一方のシステムの所有者に連絡し、いずれかのシステムのアドレスを変更します。

イーサネット・ケーブルの問題

次のメッセージが表示されることがあります。

```
unix: ec0: no carrier: check Ethernet cable
```

このメッセージは、イーサネット・ケーブルに緩みが生じていること、または、どこかで切断していることを示しています。ネットワーク上のほかのシステムが停止され、電源が切られた場合に表示されることもあります。

```
unix: ec0: late xmit collision
```

このメッセージは、ネットワーク上のほかのシステムがイーサネットの標準トランスミッション・タイミングを越えるトランスミッションを送信していることを、イーサネット・インタフェースが検知したことを示しています。

頻繁に起こる問題として、ネットワークの仕様の範囲を超えた設定による原因が挙げられます。

- 長すぎる AUI トランシーバ・ケーブル。AUI ケーブルの長さは必ず 50 m 以下にします。
- 新規に、または最近追加したネットワークの総長を超えているネットワーク・セグメント。
- トランシーバの故障。
- イーサネット・コントローラの故障。

これ以外のケースでも問題が起こる場合は、ネットワーク管理者または日本シリコングラフィックス株式会社のサポート部門までお問い合わせください。

root ファイルシステムが見つからない場合

起動時に root ファイルシステムが見つからない場合は、PROM に間違った ROOT 変数が設定されていないかどうかを確認します。ROOT 変数が間違っている場合は、モニタ・プロンプトに対して `unsetenv root` と入力し、再起動します。

ログインおよび su の問題

次のメッセージは、情報を示しています。これらのメッセージは、ほかのユーザがシステムにログインしようとした場合、または、ほかのアカウントを使用しようとした場合に、正しく行われたかあるいは失敗したかを示します。

ログイン・メッセージ

次のメッセージが表示されることがあります。

```
login: failed: <user> as <user>
```

このメッセージは、正しいパスワードを入力しなかったか、ログイン名の綴りに誤りがあるなどが原因で、login に失敗したことを示します。/var/adm/SYSLOG ファイルには、ローカル・システムからログインしようとしたホスト名、またはローカル・システムにログインしようとしたアカウント（ユーザ名）が記録されます。

su メッセージ

次のメッセージが表示されることがあります。

```
su: failed: ttyq# changing from <user> to root
```

su メッセージは、情報を示します。このメッセージは、ユーザ・アカウントを変更しようとした場合に表示されます。通常、ユーザが **root**（特権ユーザ）にアカウントを変更しようとした場合に表示されます。

このメッセージは、正しいパスワードを入力しなかったか、ログイン名の綴りに誤りがあるなどが原因で、su に失敗したことを示します。/var/adm/SYSLOG ファイルには、ホスト名、tty ポート番号 (ttyq#)、su コマンドを実行しようとしたユーザの名前、およびユーザが使用したアカウントが記録されます。

ネットワークの起動の問題

このメッセージは、bootp プログラムが同じシステム内のリモート端末から呼出されたことを示します。

```
bootp: reply boot file /dev/null
```

通常、bootp コマンド後に指定するファイル名には、リモート・システムを起動できるコードが含まれます。この起動ファイルは、ディスクレスのシステムの再起動、インストール・プログラム (sa) の起動、sash でのシステムの起動、または、X 端末の起動などに使用します。bootp プログラムは、システム間およびリモートのネットワーク・デバイス間で通信し、ネットワークを越えて起動ファイルを簡単に読取るための通信プログラムです。

オペレーティング・システム再構築の問題

次のメッセージが表示されることがあります。

```
lboot: Automatically reconfiguring the operating system
```

このメッセージは、オペレーティング・システムに関するファイルに変更があったこと、またはシステムを最後に再起動した後にシステムのハードウェアに変更があったことを示しています。システムは、変更を取込むために自動的に新しいカーネルを構築します。システムが一度再起動されると、その変更が有効になります。

カーネルを変更したり、カーネルのチューニング・パラメータの変更や追加を必要とする追加ソフトウェアをインストールするなど、オペレーティング・システム・ファイルが変更されるとこのメッセージが表示されます。このメッセージが、システムの起動時に毎回表示される場合は、オペレーティング・システム・ファイルの日付をチェックしてください。ファイルの日付が、未来の日付に設定されている可能性があります。

電力障害の検出

次のメッセージが表示されることがあります。

```
unix: WARNING: Power failure detected
```

このメッセージは、交流入力電圧が許容レベルより低くなったことを、システムが検出したことを示します。情報の内容は、/var/adm/SYSLOG に記録されます。

これは、情報を表示するメッセージですが、すべての交流のアウトレット (コンセント) の接続や、ディスク・ドライブの故障や基板のオーバーヒートがないかなど、システムのコンポーネントをチェックすることをお勧めします。Challenge システムや Onyx システムでは、システ

ム・コントローラはシステムを安全に停止しようとします。停止する場合は、すべてのプロセスが停止され、ディスクが同期します。

予備電力供給ユニット障害の検出

次のメッセージが表示されることがあります。

```
MAINT-NEEDED: module xx MSC: Power Supply Redundancy is Lost
```

このメッセージは、システムが、モジュール xx で予備電力供給ユニットの障害を検出したことを示します。ユニットは、操作中またはインストール時に停止することがあります。システム・コントローラ割込みハンドラで障害を監視し、この情報メッセージを /var/adm/SYSLOG に送信します。

障害の検出された電力供給ユニットは修理または交換しなければなりません。サポート部門に連絡してください。

予備電力供給ユニットが修復された場合、次の情報メッセージが /var/adm/SYSLOG に送信されます。

```
module xx MSC: Power Supply Redundancy is Restored
```

SCSI コントローラのリセット

次のようなメッセージが表示されることがあります。

```
unix: wd93 SCSI Bus=0 ID=7 LUN=0: SCSI cmd=0x12 timeout after 10 sec.  
Resetting SCSI  
unix: Integral SCSI bus ... reset
```

このメッセージは、SCSI コントローラがデバイス (ID 番号がある位置) に対する問い合わせをしても、応答がないことを示しています。

このメッセージは、システムの SCSI デバイスへアクセスした場合に問題が起こったことを示しています。次に、このメッセージが表示される原因を示します。

- アクセスされた SCSI デバイスが、コントローラが問い合わせたタイプをサポートしていません。

- SCSI デバイスまたは SCSI コントローラに物理的な問題があります。

このメッセージが引続き表示される場合は、`/var/adm/SYSLOG` ファイルを調べ、このような問題を回避したり特定できる手がかりとなるメッセージがあるかどうかを調べます。または、日本シリコングラフィックス株式会社のサポート部門までお問い合わせください。

syslogd デーモンの問題

次のメッセージが表示されることがあります。

```
syslogd: restart
```

syslogd メッセージは、syslogd デーモンの開始と停止を示します。このメッセージは、システムが停止、または再起動された場合に、`/var/adm/SYSLOG` に書込まれます。

システム・クロックと日付の問題

システム・クロックと日付の問題を通知するメッセージがあります。

システムの電源切断のメッセージ

次のメッセージが表示されることがあります。

```
unix: WARNING: clock gained ... days
```

このメッセージは、システムが何日間か物理的に電源が切断されていたことを示します。この日数は、`/var/adm/SYSLOG` にあるメッセージからわかります。

この問題を修正するには、システムの日付と時間とをリセットする必要があります。システム時間の設定についての詳細は、`date(1)` マン・ページと、83 ページの「システムの日付と時刻の変更」を参照してください。

時刻と日付のメッセージ

次のメッセージが表示されることがあります。

```
unix: WARNING: CHECK AND RESET THE DATE!
```

このメッセージは、時間や日付に関するメッセージに続いて表示されます。このメッセージの中には単に情報を表示しているだけのものもありますが、システムの日付、時間、またはハードウェアに問題があったことを示すものもあります。これ以外の時計、日付、または時間に関連する問題については、ログ・ファイルの `/var/adm/SYSLOG` を調べてください。日付、時間、または時計に関するメッセージが見つからない場合は、`chkconfig` の `verbose` オプションをオンに設定してください。

日付と時間の設定についての詳細は、`date(1)` マン・ページと 83 ページの「システムの日付と時刻の変更」を参照してください。`chkconfig` オプションについては、`chkconfig(1M)` マン・ページを参照してください。

タイム・サーバ・デーモンのメッセージ

```
timed: slave to gate-host-name
```

タイム・サーバ・デーモン (`timed`) に関する情報は、`/var/adm/SYSLOG` にあります。これに関してユーザは特に何もする必要はありません。タイム・サーバ・デーモンについての詳細は、`timed(1M)` マン・ページを参照してください。

システムの再起動の情報

次のメッセージが表示されることがあります。

```
INFO: The system is shutting down.
```

```
INFO: Please wait.
```

```
syslogd: going down on signal 15
```

```
syslogd: restart
```

```
unix: [IRIX Release ...
```

```
unix: Copyright 1987-2000 Silicon Graphics, Inc
```

```
unix: All Rights Reserved.
```

システム起動時に記録されるログ・メッセージには、システムが使用しているオペレーティング・システム環境に関する情報が含まれています。起動メッセージには、システムにロードされた IRIX オペレーティング・システムのバージョンと、Silicon Graphics, Inc. の著作権の情報が記録され

ます。オペレーティング・システムのバージョン情報は、システムで発生した問題をプロバイダに報告する際に有効です。

システム停止時に記録されたログ・メッセージも /var/adm/SYSLOG に送信されます。これは、システムにログインしているすべてのユーザとシステム・コンソールに送信されるメッセージで、特に何もする必要はありません。

保持または無視されたトラップ

次のメッセージが表示されることがあります。

```
unix: WARNING: Process ... generated trap, but has signal 11 held or ignored
```

```
unix: Process ... generated trap, but has signal 11 held or ignored
```

このメッセージは、プロセスが無限ループに入ったため、続いて表示されるはずのシグナルやトラップのメッセージが保持されたか、無視されたことを示します。

このメッセージは、通常、一時的にリソースがなくなった場合やプログラムのバグによって起こされます。リソースが問題である場合は、再度このメッセージが表示されることはなく、プログラムを再実行できます。プログラムを実行したときに再度メッセージが表示された場合は、プログラムにバグがあります。

メモリとスワップのメッセージ

以下のメッセージは、システム・メモリとスワップ領域の問題、およびシステムによるリソースの管理方法を示します。

growreg の不足しているメモリ

次のメッセージが表示されることがあります。

```
unix: growreg - temporarily insufficient memory to expand region
```

このメッセージは、システム上でプログラムが実行時に使用できるメモリが、実メモリ、仮想メモリともに不足していることを示しています。新しいプロセスやプログラムの起動に使用できるメモリが不足しています。

このメッセージが引続き表示される場合は、336 ページの「スワッピングとページングのメッセージ」を参照し、問題を解決してください。

ページ解放のパニック

次のメッセージが表示されることがあります。

```
PANIC: panic pagefree: page use 0
```

メモリのページ・アドレスの値が妥当な範囲にない場合、または、すでに解放されたメモリのページを解放しようとしている場合、システムはこれらを認識し、このメッセージを表示します。

このパニック・メッセージが表示されると、システムは即座に停止します。システムは、停止するときにカーネル・スタックの内容とメモリ使用情報をクラッシュ・ファイルに書込もうとします。通常、ページ解放のパニックは、メモリの物理的問題、またはディスクの不良によって起こされます。再度このメッセージが表示された場合は、日本シリコングラフィックス株式会社のサポート部門までお問い合わせください。

物理メモリの問題

次のようなメッセージが表示されることがあります。

```
unix: CPU Error/Addr 0x301 <RD PAR>: 0xbd59838
```

システムのランダム・アクセス・メモリは、いくつかのモジュール・バンクと呼ばれるものから構成されています。それぞれのバンクには、1 個の SIMM があります。これらの SIMM が正しく接続されていない場合や不良な場合に問題が発生することがあります。この問題を解決すると、システムやアプリケーションが正しく動作します。

手順 B-2 に従ってください。

手順 B-2 SIMM のチェックリスト

1. システムを停止します。
2. システムのユーザ・ガイドを参照してください。
 - 正しく接続されていない SIMM を識別し、接続し直す方法を説明しています。SIMM が正しく接続されている場合は、おそらく交換が必要です。日本シリコングラフィックス株式会社のサポート部門までお問い合わせください。
 - システムのユーザ・ガイドに SIMM についての情報が説明されていない場合は、日本シリコングラフィックス株式会社のサポート部門までお問い合わせください。

回復可能なメモリ・エラー

ハードウェアがメモリ・パリティ・エラーを検出し、パリティ状況から回復できた場合にこのメッセージが表示されます。メッセージが表示される頻度が増えていないかぎり、特に何もする必要はありません。なお、16 進数を控えておいてください。これは、SIMM のメモリ位置を示します。

```
unix: Recoverable parity error at or near physical address 0x9562f68
<0x308>, Data: 0x8fbf001c/0x87b00014
```

このメッセージは、メモリ容量が割当てられたプログラムをシステムが読取ろうとして、エラーが返された場合に表示されます。通常返されるエラーは、次のようなメモリ・パリティのエラーです。

```
unix: Recoverable memory parity error detected by CPU at 0x9cc4960
<0x304> code:30
```

CPU がメモリ・パリティ・エラーを検出し、それを /var/adm/SYSLOG に報告した場合に、このメッセージが表示されます。このメッセージが表示される頻度が増えていないかぎり、特に何もする必要はありません。なお、16 進数を控えておいてください。これは、SIMM のメモリ位置を示します。

```
unix: Recoverable memory parity error corrected by CPU at 0x9cc4960
<0x304> code:30
```

CPU がメモリ・パリティ・エラーを検出し、パリティ状況から回復できた場合に、このメッセージが表示されます。このメッセージが表示される頻度が増えていないかぎり、特に何もする必要

はありません。なお、16 進数を控えておいてください。これは、SIMM のメモリ位置を示しています。

savecore I/O エラー

次のメッセージが表示されることがあります。

```
savecore: read: I/O Error
```

システム・クラッシュの後でシステムが再起動されたとき、savecore プログラムが `/dev/swap` を読取れなかった場合に、このメッセージが表示されます。この読取り動作は、クラッシュ時のメモリのコア・イメージを保存するために行われます。

savecore プログラムは、システムが特権ユーザの権限を持って再起動してから実行されます。savecore が、メモリのコア・イメージの `/dev/swap` を読取れなかった場合は、ディスクのスワップ・パーティションに問題がある可能性があります。このメッセージの後に、ディスク・エラーのメッセージが続く場合があります。この問題が引続き発生する場合は、日本シリコングラフィックス株式会社のサポート部門までお問い合わせください。

スワッピングとページングのメッセージ

スワッピングとページングは、複数のプロセスやシステム・アクティビティに、メモリ・リソースを提供するためのオペレーティング・システムによって制限されるメモリ・リソースの管理方法です。通常、オペレーティング・システムは、動作しているプログラムのうち、現在使用されている、または最近使用されていた部分にだけ、実際の RAM メモリを確保します。プログラムの新しいセクションが必要になると、それらはメモリにページング（あるいはフォルト）されます。古いセクションは必要ではないため、ページアウトされます。

スワッピングは、ページングに似ています。ページングでは、各メモリ・ページがスワップアウトされますが、スワッピングではプロセス全体がスワップアウトされるという点が異なります。システムは、ハードディスクのある部分をスワッピング用に保持しています。この領域が一杯になると、さらにプログラムをスワップアウトすることができなくなり、これ以上プログラムを生成できません。

次のメッセージは、スワッピングまたはページングの問題を示しています。

```
Swap out failed on logical swap
```

オペレーティング・システムは、ディスクのスワップ領域にプログラムを書込むことができません。プロセスがまだメモリ上にあるので、特に何もする必要はありません。148 ページの「スワップ領域」を参照してください。

```
Paging Daemon (vhand) not running - NFS server down?
```

vhand が実行されていないことをシステムが認識しました。NFS サーバへの入出力の転送終了を *vhand* が待っている可能性があります (特に、NFS ファイルシステムが手動マウントの場合)。必要に応じてシステムが vhand を再起動するので、対処の必要はありません。

```
bad page in process (vfault)
```

メモリにフォルトされているページは、認識できるタイプではありません。認識できるタイプとは、デマンド・フィル、デマンド・ゼロ、イン・ファイル、または、オン・スワップです。システムを再起動してください。エラーが続く場合は、アプリケーションとディスクをチェックしてください。

```
unix: WARNING: Process ... killed due to bad page read
```

メモリにフォルトされているページを読むことができず、プロセスが異常終了しました。プログラムまたはシステムを再起動してください。エラーが続く場合は、アプリケーションとディスクをチェックしてください。

```
unix: Process ... killed due to insufficient memory/swap
```

システムが、アプリケーションを動作させるために、物理メモリ (RAM) とディスクの一部 (スワップ領域) を組合わせて使用しています。現在、システムに十分なメモリとスワップ領域がありません。プログラムを停止し、メモリを解放する必要があります。

```
unix: ... out of logical swap space during fork - see swap(1M)
```

現在、システムに十分なメモリとスワップ領域がありません。新しいプロセスを開始できませんでした。

スワップ領域やメモリが頻繁に不足する場合は、手順 B-3 に従ってください。

手順 B-3 メモリやスワップ領域が不足している場合のチェックリスト

1. 使用していないアプリケーションがある場合は、終了させてください。また、すべてのデスクをチェックしてください。
2. 日本シリコングラフィックス株式会社のサポート部門に問い合わせ、メモリを追加します。
3. 仮想スワップ領域をオンにします。swap (1M) マン・ページと 148 ページの「スワップ領域」を参照してください。

管理者は、*root* でログインし、次のコマンドを入力してください。

chkconfig

chkconfig のリストに、vswap off という 1 行がある場合は、次のコマンドを入力します。

```
chkconfig vswap on
```

```
/etc/init.d/swap start
```

vswap がすでにオンの場合は、次の手順に進んでください。

4. システムが追加スワップ・システムとして使用できるファイルを作成します。このファイルのサイズの分だけ、使用できるディスク領域が減ることに注意してください。10 MB のスワップ・ファイルを作成すると、10 MB 分のディスク領域が使用できなくなります。

10 MB のスワップ・ファイルを作成するには、管理者は *root* でログインし、次のコマンドを入力します。

```
mkdir -p /var/swap
```

```
/usr/sbin/mkfile 10m /var/swap/swap1
```

```
/sbin/swap -a /var/swap/swap1
```

新しいスワップ・ファイルを継続して使用する（起動時に自動的に追加される）には、*/etc/fstab* に次の行を追加します。

```
/var/swap/swap1 swap swap pri=3
```

詳細については、swap (1M) マン・ページ、または 148 ページの「スワップ領域」を参照してください。

5. ディスクのパーティションを再分割すると、スワップ領域を永続的に使用することができません。この実行方法の詳細については、『IRIX Admin: Disks and Filesystems』を参照してください。

その他のメモリ・メッセージ

以下のようなメッセージが表示されることがあります。

```
unix: Memory Parity Error in SIMM S9 (Bank 0, Byte 2)
```

CPU は、表示された SIMM にメモリ・パリティ・エラーを検出しました。パリティ・エラーは、それぞれのメモリ・ビットの一部またはすべてが誤って読み書きされたことを示します。SIMM に関する情報を記録した後、システムを再起動します。SIMM が同じエラーを繰り返す場合は、334 ページの「物理メモリの問題」で説明するように SIMM をチェックします。

```
unix: Process...sent SIGBUS due to Memory Error in SIMM S9
```

SIMM に関する情報を記録した後、システムを再起動します。同じ SIMM がエラーを繰り返す場合は、334 ページの「物理メモリの問題」で説明している SIMM をチェックしてください。

```
Ran out of action blocks
```

CPU 間の割込みのために、マルチプロセッサ・カーネルが使用しているリソースがなくなりました。このメッセージが頻繁に表示される場合は、`sysstune(1M)` コマンドを使用し、`nactions` というパラメータ値を増やします。実行するには、223 ページの「オペレーティング・システムの調整」を参照してください。

```
mfree map overflow
```

メッセージ・キューなど、断片化によって、リソースに損失が生じているものがあります。これに関しては、特に何もする必要はありません。

システム・パニック・メッセージ

次のメッセージが表示された場合は、システムを再起動することにより問題を解決できます。このメッセージが頻繁に表示されないかぎり、特に何もする必要はありません。ここに記載していない別の PANIC メッセージもあります。これらについては、すべて PANIC メッセージの指示に従ってください。

```
bumpcnt - region count list overflow
```

オペレーティング・システムでは解決できない問題が発生しました。システムを再起動します。

```
PANIC: kernel fault
```

このメッセージは、カーネルが解決できない問題に遭遇し、停止したことを示しています。このメッセージが、システム・メッセージ・ログにある場合は、システムを再起動する必要があります。再度このメッセージが表示された場合は、参考のためにシステム・ログ・ブックにメッセージを正確に記述してください。

ソフトウェアやハードウェアが、それ以上動作できない状態のことを、システム・パニックと言います。システムが再び故障した場合や、非常に多くのエラー・メッセージを受取った場合は、日本シリコングラフィックス株式会社のサポート部門までお問い合わせください。このメッセージは、サポート部門には役立つ情報なので保存しておいてください。

- /var/adm/crash ディレクトリにファイルがある場合は、それらをテープにバックアップします。ラウンチ・アイコンをダブルクリックし、「バックアップ / リストア・マネージャ (Backup and Restore Manager)」のツールを起動します。
- ファイルをバックアップした後は、それを削除することができます。
- 「システム・ログ・ビューア (System Log Viewer)」をチェックし、表示されたすべてのメッセージを保存します。ラウンチ・アイコンをダブルクリックし、「システム・ログ・ビューア (System Log Viewer)」を起動します。
- カーネルのチューニング・パラメータを最近変更した可能性があります。この場合、それらを以前の値か、デフォルト値か、自動設定の値に設定します。223 ページの「オペレーティング・システムの調整」を参照してください。

アプリケーションの調整

システム・リソース制限を厳守してアプリケーションを調整することにより、多くの場合システム・パフォーマンスを改善できます。システム・パフォーマンスの低下が気になる場合は、最初にアプリケーション・ソフトウェアを調べ、オペレーティング・システムが効率的に使用されているか確認する必要があります。ユーザが独自に作成したアプリケーションを実行する場合は、パフォーマンスを改善するための手順があります。また、市販のアプリケーションがパフォーマンス低下の原因になっている場合は、問題点を認識し、その情報に基づいてシステムの調整を行ったり新規ハードウェア導入を決めたり、アプリケーションの使用法や使用のタイミングを調整します。この節では、アプリケーションの検査方法と調整方法について説明します。アプリケーションの調整に関する詳しい説明は、『MIPSpro Compiling and Performance Tuning Guide』を参照してください。

timex によるアプリケーション・パフォーマンスの確認

アプリケーションの実行が遅いなど、システムの速度が遅い場合は、まずアプリケーションを確認します。設計に不備のあるアプリケーションは、調整してもパフォーマンスは向上しません。逆に効率良く設計されたアプリケーションは、コード・サイズの縮小と実行時間の短縮ができます。

timex ユーティリティは、問題の原因を把握するために使用する便利なユーティリティです。timex は特定のアプリケーションによる CPU 処理時間を報告します。形式は次のとおりです。

```
timex -s program
```

このコマンドを実行すると、*program* の *real* (実経過時間)、*user* (プロセスがそのコードの実行に要した時間) および *sys* (カーネルのシステム・コールの処理に要した時間) が表示されます。

```
timex -s ps -el
```

このコマンドは、*ps -el* コマンドを実行した後、そのプログラムの所要時間を次のように表示します。

```
real 0.95
user 0.08
sys 0.41
```

アプリケーション調整

アプリケーションが `user` 領域または `sys` 領域で大幅な時間を消費する理由は多くあります。アプリケーションを調整するために、ここでは過度のシステム・コールとコードの配置の不備について説明します。

通常、パフォーマンスの調整ができるのはユーザが独自に開発したアプリケーションのみです。市販のアプリケーションのパフォーマンスは調整できません。市販のアプリケーションのパフォーマンスについては、販売元に問い合わせてください。

User 時間短縮のためのガイドライン

アプリケーションの時間が、主に `user` 領域で使用されている場合に、まず行う必要がある調整は、`pixie` と `prof` コマンドを使用して、`user` 時間を短縮することです。これらのコマンドについては、各マン・ページを参照してください。膨大な `user` 時間を短縮するには、プログラムが次の条件を満たすようにします。

- システム・コールを必要な回数だけ行うこと。`timex -s` を使用し、プログラムが行う 1 秒あたりのシステム・コールの回数を確認します。**scall/s** をできるかぎり最小値にしてください。システム・コールは、`read` や `exec` の、マン・ページの第 2 節で説明されています
- システム・コールの `read` と `write` に最低でも 4 K バイトのバッファを使用すること。または、ユーザ・データをバッファに入れる標準入出力ライブラリ・ルーチンの `fread` と `fwrite` を使用します。
- 可能な場合、レコード・ロッキングではなく共有メモリを使用すること。レコード・ロッキングではファイルを読み書きするたびにレコードのロックが検査されます。パフォーマンスを改善するには、共有メモリとセマフォを使用し、共有データへのアクセスを制御します。`shmop(2)`、`semop(2)` および `usinit(3P)` のマン・ページを参照してください。

- 効率の良い検索パス (\$PATH 変数) を定義すること。最も頻繁に使用されるディレクトリ・パスを先に指定し、必要なエントリだけを指定すると、あまり使用されないディレクトリは毎回検索されません。
- ポーリング・ループを削除すること。select(2) マン・ページを参照してください。
- 動作待ちを削除すること。sginap(2) を使用します。
- システム・エラーをなくすこと。システム・エラー・ログの /var/adm/SYSLOG を調べ、プログラムが生成したエラーを検査し、問題点を解決します。

過剰ページングの減少のためのガイドライン

timex を再度実行します。アプリケーションが大幅な user 時間または sys 時間を占めている場合は、テキストとデータの配置の不備による過剰ページングの可能性がります。コードが局在化しているアプリケーションは、プログラム・ループとサブルーチンを使用し、局所的なテキスト領域で命令を実行します。このような場合は、プログラムが次の条件を満たすようにすることで、膨大な user/sys 時間の短縮を試みます。

- サブルーチンをグループ化すること。ロードされたプログラム内で頻繁に使用されるサブルーチンとあまり使用されないルーチンが混在している場合は、プログラムはルーチンが使用頻度の順にロードされた場合よりも多くのシステム・メモリ・リソースを必要とします。これは、あまり使用されないルーチンがページの一部としてメモリに格納されるからです。
- 物理メモリ内に収まるワーキング・セットを使用すること。これで、システムによって実行される必要があるページングとスワッピングの量を最小限にできます。
- Fortran から C への移植を正しく行うこと。Fortran の配列の構造は C の配列の構造と異なります。Fortran の配列は列単位ですが、C の配列は行単位です。プログラムを正しく移植しないと、アプリケーションのデータ配置に問題が生じます。

プログラムの調整を行った後、再度 timex を実行します。それでも sys 時間が長い場合は、オペレーティング・システムを調整し、この時間を短縮できます。

I/O スループット向上のためのガイドライン

アプリケーションの I/O スループットを向上させる方法がいくつかあります。シングル・ユーザ・ワークステーションを使用している場合は、次の点について確認してください。

- アプリケーションが複数のドライブを使用して I/O 帯域幅を上げること (適用可能な場合)。アプリケーションが複数のファイルに対して同時に入出力を行う場合は、ファイルを異なるファイルシステム内に設定します。さらに、可能であれば異なるデバイス上に、または異なるコントローラ上に存在するように設定します。
- アプリケーションが断片化されていないファイルを取得すること。ファイルシステムに一度に書込まれるファイルは 1 つだけになるように、アプリケーションを設定します。ファイルシステムに書込まなければならないファイルが複数ある場合にそれらを 1 つずつ書込むか同時に書込むかを選択できる場合は、ファイルを 1 つずつ書込む方が領域の割当ての効率が良くなります。この結果、I/O スループットが向上します。
- マルチユーザのサーバを使用している場合、ほかのアプリケーションのシステムへのアクセス制御は困難です。I/O サイズは大きく設定し、16 K バイト以上にしてください。また、ユーザごとに別のファイルシステムを設定することもできます。timex からの sys 時間の出力が多い場合は、オペレーティング・システムを監視し、この時間が多い理由を確認する必要があります。

アプリケーションの検証と再配置

多くのアプリケーションには、何度も繰り返し実行されるルーチンがあります。こうしたルーチンをソース・コード内で変更し、プログラムのパフォーマンスを最適化できます。次に、プログラムの調整に役立つツールについて説明します。

prof を使用したプログラム動作の分析

プロファイリングを行うと、実行中のプログラム動作を監視し、そのプログラムの各ルーチンに費やされている時間を確認できます。プロファイリングには 2 つのタイプがあります。

- プログラム・カウンタ (PC) のサンプリング
- 基本ブロックのカウント

PC のサンプリングとは、プログラムに頻繁に割り込み、プログラム・カウンタの値を統計的に記録する方法です。これに対し、基本ブロック・カウントは、pixie ユーティリティを使用して、各基本ブロック (分岐命令を含まない命令のシーケンス) の先頭に各ブロックが実行された回数をカウントするコードを挿入することによって、プログラム・モジュールを変更して行われます。

どちらのタイプのプロファイリングも効果的ですが、基本ブロック・カウントが決定的であるのに対して、PC サンプリングは統計的である点が異なります。PC のサンプリングを行うには、プログラムを **-p** オプションでコンパイルします。この結果、プログラムが実行されると、PC のサンプリング情報を待つ出力ファイルが作成されます。サンプリングの情報は、prof(1) ユーティリティを使用して分析できます。prof と pixie は出荷時の IRIX の基本構成にはありませんが、IRIS Development Option の提供ソフトウェアには含まれています。

手順 C-1 で、基本ブロックのカウント方法を説明します。

手順 C-1 基本ブロックのカウント

1. プログラムをコンパイルします。
2. このプログラムに対して pixie を実行し、カウントを実行するための特殊命令を含む新しいバイナリ・ファイルを生成します。

結果作成されたプログラムを実行すると、出力ファイルが生成されます。このファイルを prof で使用することにより、各基本ブロックが消費するサイクル数に関するレポートを生成することができます。

3. prof の出力を使用して、プログラムの動作を分析し、プログラムの時間を消費するアルゴリズムを最適化します。

プロファイリング機構の詳細については、cc(1)、f77(1)、pixie(1) および prof(1) の各マニフェストページを参照してください。

pixie を使用したプログラムの再配置

ユーザ・プログラム・テキストは、一度に 1 ページ（現在は 4 K バイト）ずつ要求に基づいてロードされます。したがって、現在はメモリになく、ユーザのアドレス空間にマップされている命令を参照すると、命令を含むページがメモリに読取られ、その後、ユーザのアドレス領域にマップされます。ロードされたプログラム内に頻繁に使用されるサブルーチンとあまり使用されないサブルーチンが混在している場合は、プログラムはルーチンが使用頻度の順にロードされた場合よりも多くのシステム・メモリ・リソースを必要とします。これは、あまり使用されないルーチンが、別のルーチンに属する命令ページの一部としてメモリに格納されるからです。

pixie、prof および cc などのツールは、プログラムの実行履歴を分析し、記録された実行履歴に従って、ルーチンの使用頻度が高い順にロードされるようにプログラムを並べ替えるための

機能を持っています。これらのツールを使用し、キャッシュ・ヒット率の最大化や (`sar -b` を実行してチェック)、ページングを最小化 (`sar -p` を実行してチェック) することによって、プログラムの実行時間を大幅に短縮できます。`fetch` という名前のプログラムを再配置するには手順 C-2 に従ってください。

手順 C-2 プログラムの再配置

1. `pixie` コマンドを実行し、プロファイリング・コードを `fetch` に追加します。

pixie fetch

このコマンドによって出力ファイル `fetch.pixie` と、基本ブロックのアドレス含むファイル `fetch.Addrs` が作成されます。

2. `fetch.pixie` (前の手順で作成された出力ファイル) を標準セットまたはデータ・セット上で実行します。この結果、基本ブロックのカウントを含む `fetch.Counts` という名前のファイルが作成されます。
3. コンパイラによってローダに渡されるフィードバック・ファイルを作成するには、`prof` を実行します。

prof -pixie -feedback fbfile fetch fetch.Addrs fetch.Counts

この結果、`fbfile` という名前のフィードバック・ファイルが作成されます。

4. 元のフラグとオプションに次の 2 つオプションを追加し、プログラムをコンパイルします。

-feedback fbfile

詳細については、`prof(1)` および `pixie(1)` マン・ページを参照してください。

市販のアプリケーションの調整

通常、市販のアプリケーションのパフォーマンスは大幅には調整できません。市販のアプリケーションがシステム・パフォーマンス低下の原因であることがわかった場合は、次の対策を取ることができます。

- アプリケーション内の別の領域のシステム負荷を軽減して速度の向上を図ります。たとえば、システムの負荷レベルが許すかぎりファイルやプログラムのバッチ処理を行うと、パフォーマンスは向上します。詳細については、25 ページの「`at`、`batch`、`cron` コマンドによるタスクのスケジューリング」を参照してください。

- アプリケーションに、より多くの CPU 時間を割当てするには、nice、renice、npri、および runon の各ユーティリティを使用し、ほかのプロセスの優先順位を変更してください。160 ページの「nice によるプロセスの優先順位付け」、および 162 ページの「実行中プロセスの優先順位の変更」を参照してください。
- ディスクのストライプ化やスワップ領域の増設などを行い、オペレーティング・システムの I/O の効率を最大化すると、システム・パフォーマンスの改善に関する一般的な問題を解決できます。詳細については、『IRIX Admin: Disks and Filesystems』を参照してください。
- メモリの増設、ディスクの追加、高速 CPU への切替えなどを行います。
- 同じ機能を持つ別のアプリケーション（ただし、システムに対して負荷がより少ない）を検索します。この方法は、ほかの方法に失敗したときに使ってください。

IRIX ディレクトリとファイル

この章では、システム管理者が使用するディレクトリとファイルについて簡単に説明します。システム・ファイル形式の詳細については、IRIX セクション 4 マン・ページを参照してください。

IRIX `root` ディレクトリ

以下に、`root (/)` ファイルシステムにある主なディレクトリを示します。

<code>/</code>	ハードウェア固有のファイルとシステムの起動に必要なファイルがあります。
<code>bin</code>	共有の実行可能なコマンド（一部は <code>root</code> 専用）があります。
<code>debug</code>	<code>/proc</code> にリンクしています。
<code>dev</code>	システム上のすべてのデバイスを定義する特殊ファイルがあります。
<code>etc</code>	管理プログラムと管理テーブルがあります。
<code>lib</code>	共有ライブラリがあります。
<code>lost+found</code>	<code>fsck(1M)</code> で使用され、切断されたファイルとディレクトリを保存します。
<code>proc</code>	<code>dbx(1)</code> などのデバッガによって使用されるプロセスを実行するためのインタフェースを提供します。
<code>tmp</code>	テンポラリ・ファイルを格納するために使用されます。
<code>usr</code>	<code>/usr</code> ファイルシステムのマウント、またはシステム間で共通のファイルを格納するために使用されます。これらのファイルは書き込み禁止です。
<code>var</code>	各システムに固有のファイルを格納するために使用されます。通常、 <code>/var</code> の各ファイルは <code>/usr</code> に対するシンボリック・リンクが作成されています。

重要な IRIX システム・ディレクトリ

次に、システム管理において重要なディレクトリを示します。

`/etc/init.d` 上位または下位の実行レベルにシステムを移行させるときに使用するシェル・スクリプトを含むディレクトリ。これらのシェル・ファイルは、`/etc/rcn.d` ディレクトリ (n は適切な実行レベルに置換える) にある `S (start)` または `K (kill)` で始まるファイルにリンクされます。

`/etc/config` 起動時および実行時の設定情報を含むディレクトリ。

`/etc/rc0.d` システムを実行レベル 0 に移行させるために `/etc/rc0` によって実行されるファイルを含むディレクトリ。このディレクトリにあるファイルは、`/etc/init.d` ディレクトリにあるファイルからリンクされ、`K` または `S` で始まります。`K` は削除されるプロセスを示し、`S` は実行レベル 0 になったときに起動されるプロセスを示します。

`/etc/rc2.d` システムを実行レベル 2 に移行させるために `/etc/rc2` によって実行されるファイルを含むディレクトリ。このディレクトリにあるファイルは、`/etc/init.d` ディレクトリにあるファイルからリンクされ、`K` または `S` で始まります。`K` は削除されるプロセスを示し、`S` は実行レベル 2 になったときに起動されるプロセスを示します。

`/etc/rc3.d` システムを実行レベル 3 に移行させるために `/etc/rc3` によって実行されるファイルを含むディレクトリ。このディレクトリにあるファイルは `/etc/init.d` ディレクトリにあるファイルからリンクされ、`K` または `S` で始まります。`K` は停止されるプロセスを示し、`S` は実行レベル 3 になったときに起動されるプロセスを示します。

`/var/adm/acct`
アカウントング・サブシステムによって収集される情報を含むディレクトリ。

`/var/adm/crash`
システムのクラッシュ・ダンプが格納されるディレクトリ。クラッシュの原因分析の後、日本シリコングラフィックス株式会社のサポート部門から特に指示がないかぎり、これらのダンプは削除できます。詳細については、`savecare (1M)` マン・ページを参照してください。

<code>/var/adm/sa</code>	<code>sar(1)</code> によって収集される情報を含むディレクトリ。 <code>%wio</code> 、 <code>%idle</code> 、 <code>wioq-sz</code> 、および <code>%wioocc</code> のフィールドは IRIX 6.5.13 リリースで変更されているので注意してください。詳細については、 <code>sar(1)</code> マン・ページ、特に <code>-U</code> および <code>-q</code> オプションを参照してください。
<code>/usr/people</code>	システムまたはネットワークのユーザのホーム・ディレクトリを含むディレクトリ。このディレクトリは <code>/var/people</code> に対するリンク、またはまったく別のファイルシステムのマウント・ポイントにすることができます。
<code>/usr/share</code>	すべてのシステムに共通のファイルを含むディレクトリ。
<code>/var/spool</code>	スプール・ディレクトリを含むディレクトリ。このディレクトリにあるディレクトリには、送信されるメールやプリント要求などのデータが保持されます。
<code>/var/spool/cron/crontabs</code>	<code>adm</code> 、 <code>root</code> 、 <code>sys</code> の各ログイン ID、および <code>cron.allow</code> にリストされている一般ユーザのための <code>crontab</code> ファイルを含むディレクトリ。
<code>/var/sysgen/master.d</code>	ハードウェア・デバイスの構成、ソフトウェア・サービスとユーティリティ、およびエリアスを定義するファイルを含むディレクトリ。
<code>/var/sysgen/stune</code>	すべてのカーネルのチューニング・パラメータのデフォルト設定を定義するファイルを含むディレクトリ。
<code>/var/sysgen/mtune</code>	すべてのカーネルのチューニング・パラメータの現在の設定を定義するファイルを含むディレクトリ。

重要な IRIX システム・ファイル

次に、システム管理において重要なファイルを示します。

`/etc/cshrc` `/bin/csh` のユーザの標準（デフォルト）環境を定義するファイル。

<code>/etc/exports</code>	オプションの NFS ソフトウェアがインストールされている場合、起動時に NFS クライアントに対してエクスポートされる NFS ファイルシステムのリストを含むファイル。
<code>/etc/fstab</code>	マウントされるファイルシステムを指定するためのファイル。
<code>/etc/gettydefs</code>	<code>getty</code> が回線の速度と端末の設定に使用する情報を含むファイル。
<code>/etc/group</code>	システムを使用する各グループを示すファイル。
<code>/etc/hosts</code>	ネットワーク上の既知のホストに関する情報を含むファイル。
<code>/etc/hosts.equiv</code>	特権ユーザ以外の一般ユーザに <code>rlogin</code> と <code>rsh</code> の実行を許可するホストのリストを含むファイル。
<code>/etc/inittab</code>	初期化の各段階で <code>init</code> によって生成または終了するプロセスを定義する命令を含むファイル。
<code>/etc/issue</code>	ネットワークやシリアル回線を通じてシステムにログインするユーザに対して表示されるメッセージを含むファイル。
<code>/etc/lvtab</code>	ワークステーションで使用される論理ボリュームに関する情報を含むファイル。このファイルは、論理ボリューム・ユーティリティによって読取られます。
<code>/etc/motd</code>	今日のメッセージ (<code>message-of-the-day</code>) を含むファイル。
<code>/etc/passwd</code>	システムを使用する各ユーザを識別するファイル。
<code>/etc/profile</code>	<code>/bin/sh</code> のユーザの標準 (デフォルト) 環境を定義するファイル。
<code>/etc/rc0</code>	システムが実行レベル 0 に移行するときに <code>/etc/rc0.d</code> ディレクトリにあるシェル・スクリプトを実行するスクリプトを含むファイル。
<code>/etc/rc2</code>	システムが実行レベル 2 に移行するときに <code>/etc/rc2.d</code> ディレクトリと <code>/etc/rc.d</code> ディレクトリにあるシェル・スクリプトを実行するスクリプトを含むファイル。

`/etc/shutdown`
システムのバックアップやスケジュール済みのダウンタイムに備えて、システムを安全に停止するシェル・スクリプトを含むファイル。

`/etc/sys_id` システム名を含むファイル。

`/etc/ttytype`
端末ポートにログインする可能性がある端末の種類をポート順に並べたリストを含むファイル。

`/etc/TIMEZONE`
デフォルトの時間帯を示すシェル変数 `TZ` を設定するためのファイル。

`/etc/utmp` システムの現在の実行状態に関する情報を含むファイル。

`/etc/wtmp` システムのログイン履歴を含むファイル。

`/etc/xwtmp` システムの拡張ログイン履歴を含むファイル。

`/var/adm/sulog`
`su` コマンドの使用履歴を含むファイル。このファイルはサイズが大きくなりすぎないように定期的にチェックし、アーカイブを作成する必要があります。

`/var/adm/SYSLOG`
システムとデーモンのエラー・メッセージを含むファイル。

`/var/yp/ypdomain`
ワークステーションで NIS を使用している場合のドメイン・ネームを含むファイル。

`/var/cron/log cron`
によって実行されたすべての動作の履歴を含むファイル。このファイルはサイズが大きくなりすぎないように定期的にチェックし、必要に応じて圧縮する必要があります。

`/usr/lib/cron/cron.allow`
`crontab(1)` の使用を許可されたユーザのリストを含むファイル。このファイルと `cron.deny` をシステム上に同時に置くことはできません。

`/usr/lib/cron/cron.deny`

`crontab(1)` へのアクセスを禁止されたユーザのリストを含むファイル。このファイルは、`/usr/lib/cron/cron.allow` がない場合にチェックされません。

IRIX デバイス特殊ファイル

ここでは、最も重要なデバイス・ファイルとディレクトリを一覧にします。これらは、`/dev` ディレクトリ構造にあります。

<code>dsk/</code>	ディスク用のブロック・デバイス・ファイルのあるディレクトリ。ディスク・パーティションのデバイス名については、 <code>ips(7)</code> 、 <code>dks(7M)</code> 、および <code>xy1(7)</code> の各マン・ページを参照してください。
<code>rdsk/</code>	ディスク用のロー・デバイス・ファイル (キャラクタ・デバイス・ファイル) のあるディレクトリ。ディスク・パーティションのデバイス名については、 <code>ips(7)</code> 、 <code>dks(7M)</code> 、および <code>xy1(7)</code> の各マン・ページを参照してください。
<code>root</code>	一般的な <code>root</code> パーティション (ブロック・デバイス)。
<code>rroot</code>	一般的な <code>root</code> パーティション (ロー・デバイス)。
<code>usr</code>	一般的な <code>usr</code> パーティション (ブロック・デバイス)。
<code>rusr</code>	一般的な <code>usr</code> パーティション (ロー・デバイス)。
<code>swap</code>	一般的な <code>swap</code> パーティション (ブロック・デバイス)。
<code>rswap</code>	一般的な <code>swap</code> パーティション (ロー・デバイス)。
<code>vh</code>	一般的な <code>root</code> ボリューム・ヘッダ (ブロック・デバイス)。
<code>rvh</code>	一般的な <code>root</code> ボリューム・ヘッダ (ロー・デバイス)。
<code>mt/</code>	テープ用のブロック・デバイス・ファイルのあるディレクトリ。ISI の 1/4 インチのテープ・ドライブのデバイス名については、 <code>ts(7)</code> マン・ページを参照してください。SCSI の 1/4 インチのテープ・ドライブのデバイス名については、 <code>tps(7M)</code> マン・ページを参照してください。Xylogics の 1/2 インチのテープ・ドライブ名については、 <code>xmt(7)</code> マン・ページを参照してください。

rmt/	テープ用のロー・デバイス・ファイルがあるディレクトリ。ISI の 1/4 インチのテープ・ドライブのデバイス名については、ts(7) マン・ページを参照してください。SCSI の 1/4 インチのテープ・ドライブのデバイス名については、tps(7M) マン・ページを参照してください。Xylogics の 1/2 インチのテープ・ドライブ名については、xmt(7) マン・ページを参照してください。
tape	一般的なテープ・デバイス。バイト・データは、IRIS シリーズ 2000 および 3000 との下位互換性のためにスワップされます。mtio(7) マン・ページを参照してください。
nrtape	巻戻しのない、一般的なテープ・デバイス。バイト・データは、IRIS シリーズ 2000 および 3000 との下位互換性のためにスワップされます。mtio(7) マン・ページを参照してください。
tapens	一般的なテープ・デバイス。バイト・データはスワップされません。mtio(7) マン・ページを参照してください。
nrtapens	巻戻しのない、一般的なテープ・デバイス。バイト・データはスワップされません。mtio(7) マン・ページを参照してください。
mem	メモリ。mem(7) マン・ページを参照してください。
mmem	マッピング可能なメモリ。mmem(7) マン・ページを参照してください。
kmem	カーネル・メモリ。kmem(7) マン・ページを参照してください。
null	ヌル・デバイス (入力とした場合は長さは 0 となり、出力とした場合はデータはなくなります)。null(7) マン・ページを参照してください。
SA/	システム管理ツールによって使用されるブロック・デバイス。sysadm(1M) および sa(7) マン・ページを参照してください。
rSA/	システム管理ツールによって使用されるロー・デバイス。sysadm(1M) および sa(7) の各マン・ページを参照してください。
audio	オーディオ・インタフェース。
dn_ll	4DDN 論理リンクを作成するために使用されるファイル。dn_ll(7) マン・ページを参照してください。
dn_netman	4DDN ネットワーク管理ソフトウェアによって使用されるファイル。dn_netman(7) マン・ページを参照してください。

cent	Centronics のカラー・グラフィックス・プリンタ・デバイス。
tek	Tektronics のカラー・グラフィックス・プリンタ・デバイス。
vers	Versatec のカラー・グラフィックス・プリンタ・デバイス。
vp0	<i>vers</i> へのハード・リンク。
gpib*	GPIB (IEEE-488) デバイス。gpib(7) マン・ページを参照してください。
gse	Spectragraphics の同軸デバイス。gse(7) マン・ページを参照してください。
plp	パラレル・ラインのプリンタ・インタフェース。plp(7) マン・ページを参照してください。
prf	オペレーティング・システム・プロファイラによって使用されるファイル。prf(7) マン・ページを参照してください。
t3270	IBM 3270 Cluster Controller 用のロー・デバイス・ファイル。t3270(7) マン・ページを参照してください。
hl/	IRIS GTX シリーズのマシン・ハードウェア・スピンロック・ドライバによって使用されるファイルを含むディレクトリ。usnewlock(3P) マン・ページを参照してください。
log	システム・ロギング・デーモンによって読取られる名前付きパイプ。syslogd(1M) マン・ページを参照してください。
ptc	クローン化が可能な疑似 tty コントローラ。clone(7)、および ptc(7) の各マン・ページを参照してください。
grconc	グラフィックス・コンソール用のマスター疑似ターミナル・タイプ。pty(7M) マン・ページを参照してください。
grcons	グラフィックス・コンソール用のスレーブ疑似ターミナル・タイプ。pty(7M) マン・ページを参照してください。
gm	IRIS GT モデルおよび GTX モデルのマシンのグラフィックス・マネージャ (GM) のための論理コンソール・デバイス。GM ボードの 68020 で動作するソフトウェアからのメッセージはこのデバイスに出力されます。
grin/	独立した論理グラフィックス入力デバイスを含むディレクトリ。
console	システム・コンソール・デバイス。

<code>syscon</code>	<code>/dev/console</code> へのハード・リンク。
<code>systty</code>	<code>/dev/console</code> へのハード・リンク。
<code>queue</code>	グラフィックス・キュー・デバイス。グラフィックス・プログラムは、グラフィックス・キューに入力があるとき、そのデバイスの “select” を呼出すことで認識します。このデバイスに対して読み書きは行えません。
<code>dials</code>	ダイヤルおよびボタン・ボックスに接続されたシリアル・ポート用のデバイス。
<code>keybd</code>	キーボードに接続されたシリアル・ポート用のデバイス。
<code>mouse</code>	マウスに接続されたシリアル・ポート用のデバイス。
<code>tablet</code>	デジタイザ・タブレットに接続されたシリアル・ポート用のデバイス。
<code>ttyd[1-12]</code>	シリアル・ポートの 1 から 12 まで。
<code>ttyf[1-12]</code>	ハードウェアのフロー制御を認識するデバイスのためのシリアル・ポートの 1 から 12 まで。
<code>ttyM[1-12]</code>	モデム用のシリアル・ポートの 1 から 12 まで。
<code>ttyq*</code>	疑似 <code>tty</code> デバイス。pty(7M) マン・ページを参照してください。
<code>zero</code>	ゼロ・デバイス (ゼロを無限に含むファイルで、読取り時には常にゼロを返す)。 zero(7) マン・ページを参照してください。

ASCII 変換テーブル

ASCII キャラクタ・セットは、文字と 8 ビットの値の 1 対 1 のマッピングを定義します。以下のテーブルは、ASCII キャラクタを対応する 8 進数、16 進数、および 10 進数に変換するための対応表です。これらの表は、`ascii(5)` マン・ページでも利用できます。

表 D-1 ASCII から 8 進数へのマッピング

000 nul	001 soh	002 stx	003 etx	004 eot	005 enq	006 ack	007 bel
010 bs	011 ht	012 nl	013 vt	014 np	015 cr	016 so	017 si
020 dle	021 dc1	022 dc2	023 dc3	024 dc4	025 nak	026 syn	027 etb
030 can	031 em	032 sub	033 esc	034 fs	035 gs	036 rs	037 us
040 sp	041 !	042 "	043 #	044 \$	045 %	046 &	047 '
050 (051)	052 *	053 +	054 ,	055 -	056 .	057 /
060 0	061 1	062 2	063 3	064 4	065 5	066 6	067 7
070 8	071 9	072 :	073 ;	074 <	075 =	076 >	077 ?
100 @	101 A	102 B	103 C	104 D	105 E	106 F	107 G
110 H	111 I	112 J	113 K	114 L	115 M	116 N	117 O
120 P	121 Q	122 R	123 S	124 T	125 U	126 V	127 W
130 X	131 Y	132 Z	133 [134 \	135]	136 ^	137 _
140 `	141 a	142 b	143 c	144 d	145 e	146 f	147 g
150 h	151 i	152 j	153 k	154 l	155 m	156 n	157 o
160 p	161 q	162 r	163 s	164 t	165 u	166 v	167 w
170 x	171 y	172 z	173 {	174	175 }	176 ~	177 del

表 D-2 ASCII から 16 進数へのマッピング

00 nul	01 soh	02 stx	03 etx	04 eot	05 enq	06 ack	07 bel
08 bs	09 ht	0a nl	0b vt	0c np	0d cr	0e so	0f si
10 dle	11 dc1	12 dc2	13 dc3	14 dc4	15 nak	16 syn	17 etb
18 can	19 em	1a sub	1b esc	1c fs	1d gs	1e rs	1f us
20 sp	21 !	22 "	23 #	24 \$	25 %	26 &	27 '
28 (29)	2a *	2b +	2c ,	2d -	2e .	2f /
30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7
38 8	39 9	3a :	3b ;	3c <	3d =	3e >	3f ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G
48 H	49 I	4a J	4b K	4c L	4d M	4e N	4f O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W
58 X	59 Y	5a Z	5b [5c \	5d]	5e ^	5f _
60 `	61 a	62 b	63 c	64 d	65 e	66 f	67 g
68 h	69 i	6a j	6b k	6c l	6d m	6e n	6f o
70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w
78 x	79 y	7a z	7b {	7c	7d }	7e ~	7f del

表 D-3 ASCII から 10 進数へのマッピング

0 nul	1 soh	2 stx	3 etx	4 eot	5 enq	6 ack	7 bel
8 bs	9 ht	10 nl	11 vt	12 np	13 cr	14 so	15 si
16 dle	17 dc1	18 dc2	19 dc3	20 dc4	21 nak	22 syn	23 etb
24 can	25 em	26 sub	27 esc	28 fs	29 gs	30 rs	31 us
32 sp	33 !	34 "	35 #	36 \$	37 %	38 &	39 '
40 (41)	42 *	43 +	44 ,	45 -	46 .	47 /
48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7
56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G
72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [92 \	93]	94 ^	95 _
96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g
104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {	124	125 }	126 ~	127 del

EPSF (Encapsulated PostScript File) バージョン 3.0 と PostScript ファイル形式

システムを保守しているとき、さまざまなバージョンの PostScript 形式のファイルを取得することがあります。ここでは、EPSF (Encapsulated PostScript File) バージョン 3.0 の形式と PostScript ファイル形式の主な違いについて説明します。

- EPSF は、単一のページを記述するために使用されます。PostScript 形式は、1 ページまたは複数ページを記述するために使用されます。
- EPSF は、次の 2 つのドキュメント構造化規則 (DSC: document structuring conventions) のヘッダ・コメントを必要とします。

```
%!PS-Adobe-3.0 EPSF-3.0 %%BoundingBox: llx lly urx ury
```

PS 3.0 ファイルが DSC に準拠する場合は、次のコメントで始まる必要があります。

```
%!PS-Adobe-3.0
```

PS ファイルが DSC に準拠しない場合は、DSC コメント文を含む必要はありません。

- `copypage`、`erasepage`、または `exitserver` など、一部の PostScript 言語オペレータは、EPS ファイル内では使用できません。

`nulldevice`、`setscreen`、または `undefinefont` など、ほかの PostScript 言語オペレータが EPS ファイル内で使用されている場合は、ある種の制約があります。

すべての PostScript オペレータは、PS ファイル内で使用できます。

- EPS ファイルは PS ファイル内に挿入できます。ただし、EPS ファイル内に PS ファイルは挿入できません (PS ファイルの挿入が有効な EPS ファイルの作成条件に違反している場合)。
- EPS ファイルには、そのファイルの出力イメージ用にデバイス固有のスクリーン・プレビューを含む場合があります。通常、PS ファイルに EPS ファイルが含まれている場合にかぎり、スクリーン・プレビューがあります。

- EPS ファイルのファイル名の拡張子として推奨されているのは、**.EPS** または **.eps** です。PS ファイルのファイル名の拡張子として推奨されているのは、**.PS** または **.ps** です。

文書の記述の残りの部分に影響を及ぼさず、単一のページ、または図表のようにページの一部の PostScript の記述を取込むために EPSF 形式は設計されました。したがって、EPS コードは、カプセル化されている必要があります。EPS コードを囲む PS コードには干渉できません。また、それらのコードに依存することもできません。

通常、EPSF 形式は Illustrator など、イラストや図の作成のために設計されたプログラムからの出力、または Adobe System の FrameMaker など、DTP のプログラムへの入力に使用されます。ほとんどの DTP のプログラムは、EPS コードを含む PostScript 形式の文書データを生成します。

これらの形式の詳細については、『PostScript Language Reference Manual』(Addison-Wesley Publishing Company, Inc., Reading, MA, 1990) の第 2 版を参照してください。Adobe Systems PostScript ファイル・サーバから EPS 形式および PS 形式に関する情報を入手できます。UNIX のプロンプトに対して次のコマンドを入力し、そのファイル・サーバからの指示に従ってください。

```
mail ps-file-server@adobe.com
```

```
Subject: help
```

```
<Ctrl>+D キーを押します。
```

上記のファイル・サーバに、次に示すメッセージを送ると、PS 形式にある EPSF 形式の記述を入手できます。

```
send Documents EPSF2.0.ps
```

参考文献

Internet Request For Comment は、次のアドレスのインターネット・ネットワーク情報センター (INTERNIC) から入手できます。

Network Solutions
Attn: InterNIC Registration Services
505 Huntmar Park Drive
Herndon, VA 22070
電話番号 : 1-800-444-4345 または 1-703-742-4777

Bach, M., *The Design of the UNIX Operating System* (Englewood Cliffs, NJ: Prentice Hall, 1986).

Braden, R., "Requirements for Internet Hosts." Internet Request For Comment 1112 (1989).

Costales, B., *sendmail*. (Sebastopol, CA: O'Reilly & Associates, Inc., 1993).

Deering, S., "Host Extensions for IP Multicasting." Internet Request For Comment 1112 (1989).

Everhart, C., Mamakos, L., Ullmann, R., Mockapetris, P. "New DNS RR Definitions." Internet Request For Comment 1183 (1990).

Fiedler, D., Hunter, B., *UNIX System V Release 4 Administration* (Carmel, IN: Hayden Books, 1991).

Frisch, A., *Essential System Administration*. (Sebastopol, CA: O'Reilly & Associates, Inc., 1991).

Gilly, D., *UNIX in a Nutshell*. (Sebastopol, CA: O'Reilly & Associates, Inc., 1992).

Hunt, C., *TCP/IP Network Administration*. (Sebastopol, CA: O'Reilly & Associates, Inc., 1992).

- Leffler, S., *The Design and Implementation of the 4.3 BSD UNIX Operating System*. (Menlo Park, CA: Addison Wesley, 1989).
- Lottor, M., "Domain Administrator's Guide." Internet Request For Comment 1033 (1987).
- Lottor, M., "TCP Port Service Multiplexer (TCPMUX)." Internet Request For Comment 1078 (1988).
- Mockapetris, P., "DNS Encoding of Network Names and Other Types." Internet Request For Comment 1101 (1989).
- Mockapetris, P., "Domain Names — Concept and Facilities." Internet Request For Comment 1034 (1987).
- Mockapetris, P., "Domain Names — Implementation and Specification." Internet Request For Comment 1035 (1987).
- Mogul, J., Postel, J., "Internet Standard Subnetting Procedure." Internet Request for Comment 950 (1985).
- Nemeth, E., Snyder, G., Sebass, S., *UNIX System Administration Handbook* (Englewood Cliffs, NJ: Prentice Hall, 1989). lpr プリント・システムに関する有用な情報が記載されています。
- Partridge, C., "Mail Routing and The Domain System." Internet Request For Comment 974 (1986).
- Stahl, M., "Domain Administrator's Guide." Internet Request For Comment 1032 (1987).
- Thomas, R., *UNIX System Administration Guide for System V*. (Englewood Cliffs, NJ: Prentice Hall, 1989).

索引

? コマンド 181

A

ACL 89

archshinv コマンド 179

ARCS PROM システム 176

Array 158

Array Services 158

array コマンド 158

ASCII 文字セット 358

asmachid パラメータ 307、308

at コマンド 25

autoconfig コマンド 226

AutoLoad 変数 190

automount コマンド 199、71

autopower 変数 186

autoup パラメータ 257、280

auto コマンド 179、193

availmon システム 38

B

batch コマンド 25

bdevsw_extra パラメータ 281

bdflush_interval パラメータ 279

bdflushr パラメータ 256、257

/bin/csh シェル 126

/bin/rsh シェル 126

/bin/sh シェル 126

bootfile 変数 186

bootmode 変数 186

bootp コマンド 196、329

boottune 変数 186

boot コマンド 179、193、195、197、199

Bourne シェル

スタートアップ・ファイル 129

説明 17、126

brk システム・コール 222

bsh コマンド 126

BTE 91

buf_relse_age パラメータ 234、242

C

cachefs_max_threads パラメータ 233、234

cachefs_readahead パラメータ 233、234

callout_himark パラメータ 233、236

cdevsw_extra パラメータ 281、282

cd コマンド 24

- Checkpoint and Restart 165
chgrp コマンド 25, 117
chkconfig コマンド 70, 209, 332, 338
chmod コマンド 3, 23, 26, 85, 155
chown コマンド 24
chown システム・コール 284, 291
chproj コマンド 110
colorview コマンド 20
ConsoleIn 変数 190
ConsoleOut 変数 190
ConsolePath 変数 103
console 変数 186
coremask 321
corepluspid パラメータ 284, 286
cpio コマンド 24, 117
CPR 165
cpr コマンド 165
cpulimit_gracetime パラメータ 246, 254
cpuset 164
CPU 集約的処理 157, 160
CPU 制限時間 254
CPU 動作パラメータ 283
CPU の負荷が高いシステム 220
cp コマンド 226
crontab コマンド 25
cron コマンド 25, 51
.cshrc ファイル 88, 125
cs コマンド 20, 161
cwcluster パラメータ 279
C シェル
 ショートカット 14
 スタートアップ・ファイル 127
 説明 126
C ブリック 101, 102
- D**
- DAC (Discretionary Access Control) 88
DARPA 標準プロトコル 196
date コマンド 179, 332
dbaud 変数 186
dbgname 変数 186
dbgtty 変数 186
dfaultprid パラメータ 307, 308
df コマンド 144
diagmode 変数 186
disable コマンド 179
diskless 変数 186
diskusg コマンド 144
DISPLAY 環境変数 75
DMA 転送 239
do_extpacct パラメータ 306, 307
do_procacct パラメータ 306, 307
do_sessacct パラメータ 306, 307
dump_level パラメータ 204, 234, 241
du コマンド 143
dwcluster パラメータ 280
- E**
- eaddr* コマンド 179
EAGAIN エラー・メッセージ 265

ecc_recover_enable パラメータ 234、241
 EINVAL エラー・メッセージ 266
 emacs による編集 17
 EMFILE エラー・メッセージ 266
 enableall コマンド 179
 enable コマンド 179
 ENOMEM エラー・メッセージ 305
 ENOSPC エラー・メッセージ 266
 /etc/cshrc ファイル 125
 /etc/fstab ファイル 6、150、156
 /etc/group ファイル 6、108、113
 /etc/init.d/swap ファイル 153
 /etc/inittab ファイル 52
 /etc/issue ファイル 138
 /etc/motd ファイル 4、136
 /etc/nologin ファイル 27
 /etc/passwd ファイル 3、6、107、111
 /etc/profile ファイル 125
 /etc/sys_id ファイル 78
 /etc/wall (ブロードキャスト・メッセージ) 5
 EWOULDBLOCK エラー・メッセージ 300
 exec システム・コール 222、247、342
 exit コマンド 179

F

fam(1M) マン・ページ 167
 fam(3X) マン・ページ 167
 fasthz パラメータ 292、293
 find コマンド 21、117、326
 finger コマンド 146

first_retry パラメータ 294、295
 fmodsw_extra パラメータ 281、282
 fork システム・コール 222
 fread ルーチン 342
 fru コマンド 35
 fsck コマンド 48
 fwrite ルーチン 342

G

gated コマンド 71
 gConsoleIn 変数 186
 gConsoleOut 変数 186
 getitimer システム・コール 293
 getrlimit システム・コール 245
 gettimeofday システム・コール 293
 gettimeofday パラメータ 293
 getty コマンド 56
 gfxinfo コマンド 69
 gpgshi パラメータ 204、256、258、260
 gpgslo パラメータ 204、256、258、259
 gpgsmsk パラメータ 256、258、259
 gr_osview コマンド 158、281
 gr_top コマンド 158、220
 GraceWaitTime パラメータ 294、295
 grpck コマンド 3

H

help コマンド 179、181
 hinv コマンド 61、179、222

hwgraph マン・ページ 62

I

iconlogin (*pandora*) 72

icrash コマンド 33

id コマンド 119

if_cl デバイス 92

ifconfig コマンド 92

if/etc/issue ファイル 138

initstate 変数 189

init コマンド 50、179、185、186

inpcb パラメータ・グループ 298

insecure_compatibility オプション 167

intro マン・ページ 323

I/O スループット 343

I/O デーモン 234

I/O のパフォーマンス 212

I/O の負荷が高いシステム 213

ip26_allow_ucmem パラメータ 284、288

IPC パラメータ 265

IP アドレス 326

IP ネットワーク・インタフェース 67

IRIX 167

IRIX シェル 11

IRIX ディレクトリ 349

IRIX ファイル 349

itimer_on_clkcpcu パラメータ 292、293

I ブリック 102

K

keybd 変数 186、191

killall コマンド 163、198、199

kill コマンド 163

Korn シェル 17

L

lboot コマンド 281、282

limit コマンド 245

local_only オプション 167

lock_share_requests パラメータ 294、296

lockd_blocking_thresh パラメータ 294、296

lockd_grace_period パラメータ 294、296

login コマンド 27

.login ファイル 88、125

lpadmin コマンド 70

lpstat コマンド 70

lp コマンド 56

ls コマンド 25、85、179

M

MAKEDEV コマンド 67

make コマンド 247

man コマンド xliv、31

master マン・ページ 203

max_local_paggid パラメータ 306、308

maxdc パラメータ 256、261

maxdmasz パラメータ 204、234、239、280

maxfc パラメータ 256、260

maxlkmem パラメータ 256、259
 maxpglst パラメータ 256、263
 maxpmem パラメータ 234、239
 maxsc パラメータ 256、260
 maxsymlinks パラメータ 243、244
 maxup パラメータ 203、204、242
 maxwatchpoints パラメータ 242、244
 mbmaxpages パラメータ 234、240
 mem_tick_base_period パラメータ 310、316
 mem_tick_enabled パラメータ 310、316
mesg コマンド 140
 min_bufmem パラメータ 233、236
 min_file_pages パラメータ 280
 min_free_pages パラメータ 280
 min_local_paggid パラメータ 306、307
 minarmem パラメータ 256、261
 minasmem パラメータ 256、261
Miser の使用法 164
mkdir コマンド 24
mkfile コマンド 153
mkpart コマンド 93
 mload_auto_rtsyms パラメータ 284、287
 monitor 変数 186
mpadmin コマンド 76、78
mpin システム・コール 222、259
msgget システム・コール 266
mtune マン・ページ 203
multgrps コマンド 119
multi コマンド 57
 munlddelay パラメータ 281、283
mv コマンド 26

N

nactions パラメータ 283
named コマンド 72
 nbuf パラメータ 204、233、235
 ncallout パラメータ 233、236
 ncargs パラメータ 245、246
 ncsizе パラメータ 234、237
 ndquot パラメータ 234、237
 netaddr 変数 103、186
Network Queuing Environment 165
newgrp コマンド 119
newproj コマンド 109
news コマンド 138
NFS 261
 nfs_portmon パラメータ 294、297
NFS パラメータ 294
 ngroups_max パラメータ 242、243
nice コマンド 160
 nlpages_16m パラメータ 230、257、305、306
 nlpages_1m パラメータ 230、257、305、306
 nlpages_256K パラメータ 305
 nlpages_256k パラメータ 230、257
 nlpages_4m パラメータ 230、257、305、306
 nlpages_64k パラメータ 230、257
 nogfxkbd 変数 186
 normal_retry パラメータ 294、296
 nosuidshells パラメータ 284、290
 notape 変数 186
npri コマンド 162
 nproc パラメータ 203、204、225、234、238

- nprofile* パラメータ 242、244
- NQE 165
- nroff* コマンド 31、210
- numa_kernel_replication_ratio* パラメータ 317、318
- numa_migr_auto_migr_mech* パラメータ 318、319
- numa_migr_bounce_control_interval* パラメータ 309、315
- numa_migr_coaldmigr_mech* パラメータ 318
- numa_migr_dampening_enabled* パラメータ 309、315
- numa_migr_dampening_factor* パラメータ 310、315
- numa_migr_default_mode* パラメータ 309、310
- numa_migr_default_threshold* パラメータ 309、311
- numa_migr_freeze_enabled* パラメータ 309、314
- numa_migr_freeze_threshold* パラメータ 309、314
- numa_migr_melt_enabled* パラメータ 309、314
- numa_migr_melt_threshold* パラメータ 309、315
- numa_migr_memory_low_enabled* パラメータ 309、314
- numa_migr_memory_low_threshold* パラメータ 309、314
- numa_migr_min_distance* パラメータ 309、312
- numa_migr_min_maxradius* パラメータ 309、312
- numa_migr_threshold_reference* パラメータ 309、311
- numa_migr_traffic_control_enabled* パラメータ 310、317
- numa_migr_traffic_control_interval* パラメータ 310、317
- numa_migr_traffic_control_threshold* パラメータ 310、317
- numa_migr_unpegging_control_enabled* パラメータ 310、316
- numa_migr_unpegging_control_interval* パラメータ 310、316
- numa_migr_unpegging_control_threshold* パラメータ 310、317
- numa_migr_user_migr_mech* パラメータ 318
- numa_migr_vehicle* パラメータ 309、312
- numa_paging_node_freemem_low_threshold* パラメータ 256、262
- numa_refcnt_default_mode* パラメータ 309
- numa_refcnt_overflow_threshold* パラメータ 309、313
- numa_repl_mem_lowmark* パラメータ 319
- numa_repl_traffic_highmark_percentage* パラメータ 318
- numa_zone_radius_max* パラメータ 309、312
- NUMALink 89、92
- NUMA システム 263
- NUMA パラメータ 308
- O**
- off* コマンド 179
- passwd* コマンド 179
- open システム・コール 244
- OSLoader 変数 190
- OSLoadFilename 変数 190
- OSLoadOptions 変数 190
- OSLoadPartition 変数 103、190
- osview* コマンド 158、223

P

- pagecolor 変数 186
- pandora コマンド 72
- panic_on_sbe パラメータ 283、285
- partmgr コマンド 96
- par コマンド 207
- pathname コマンド 179
- path 変数 189
- PCP 166
- percent 229
- percent_totalmem_16k_pages パラメータ 229、257
- percent_totalmem_16m_pages パラメータ 204、230、257
- percent_totalmem_1m_pages パラメータ 204、229、257
- percent_totalmem_256k_pages パラメータ 204、229、257
- percent_totalmem_4m_pages パラメータ 204、229、257
- percent_totalmem_64k_pages パラメータ 204、257、263
- Performance Co-Pilot 166
- ping コマンド 179
- pixie コマンド 342
- plock システム・コール 222、259
- portmap_timeout パラメータ 294、295
- posix_tty_default パラメータ 284、290
- POSIX 回線制御手順 290
- PostScript 形式 361
- printenv コマンド 179、190
- ProbeAllScsi 変数 186
- process_core_cpu_logging パラメータ 285、292
- .profile ファイル 88、125
- prof コマンド 342、344
- prompoweroff 変数 186
- prom コマンド 175
- PROM パスワード 185
- PROM モニタ 49、176
- pset コマンド 76
- ps コマンド 159、222、341
- pwck コマンド 3、27、111、115

Q

- quotas サブシステム 145
- quot コマンド 144

R

- r4k_div_patch パラメータ 284、287
- rbaud 変数 186
- read システム・コール 342
- reboot 67
- reboot_on_panic パラメータ 284、289
- reboot コマンド 57、176、226
- rebound 変数 186
- renice コマンド 162
- reserve_ncallout パラメータ 233、237
- reset_limits_on_exec パラメータ 284、288
- resetenv コマンド 179
- resetpw コマンド 179
- RestorePastEnv 変数 186

- restrict_fastprof* パラメータ 284, 288
restricted_chown パラメータ 284, 291
rlimit_core_cur パラメータ 203, 245, 247
rlimit_core_max パラメータ 245, 247
rlimit_cpu_cur パラメータ 245, 248, 254
rlimit_cpu_max パラメータ 245, 248
rlimit_data_cur パラメータ 203, 204, 246, 248
rlimit_data_max パラメータ 204, 246, 248
rlimit_fsize_cur パラメータ 203, 246, 249
rlimit_fsize_max パラメータ 246, 249
rlimit_nofile_cur パラメータ 204, 246, 249
rlimit_nofile_max パラメータ 246, 250
rlimit_pthreads_cur パラメータ 250
rlimit_pthreads_max パラメータ 250
rlimit_rss_cur パラメータ 204, 246, 250
rlimit_rss_max パラメータ 204, 246, 251
rlimit_stack_cur パラメータ 204, 246
rlimit_stack_max パラメータ 204, 246, 251
rlimit_vmem_cur パラメータ 204, 246, 252, 260
rlimit_vmem_max パラメータ 204, 246, 252
rlogin コマンド 19
rm コマンド 25
root ディレクトリ 2, 349
root パーティション 67
root ファイルシステム 327
root 変数 103, 186
routed コマンド 72
rpcinfo コマンド 171
rsh コマンド 19
rsshogfrac パラメータ 204, 246, 253
rsshogslp パラメータ 246, 254
rtcpus パラメータ 278
rtnetd コマンド 72
rwhod コマンド 72
- ## S
- sar* コマンド 56, 72, 149, 159, 207, 223
sash 194
savecore コマンド 33, 336
sbe_log_errors パラメータ 284, 285
sbe_mfr_override パラメータ 284, 286
sbe_report_cons パラメータ 284, 286
sbrk システム・コール 222
scache_pool_size パラメータ 256, 262
SCSI コントローラ 330
sed コマンド 21, 24
select システム・コール 221, 343
semnmi パラメータ 266
semnms パラメータ 204
semopm パラメータ 204
semume パラメータ 204
setenv コマンド 179, 191, 195, 197
setpart コマンド 93
setrlimit システム・コール 245
setsockopt システム・コール 299, 303
settimer システム・コール 293
SGI 3000 Series マシン 89
sgilog 変数 186
Share II スケジューラ 165
shell コマンド 247
shlbmax パラメータ 246, 254

shmat システム・コール 266
shmget システム・コール 266
shmmax パラメータ 204、266
shmmin パラメータ 266
shmop システム・コール 342
showconfig 変数 189
shutdown コマンド 50
sh コマンド 208
SIGKILL シグナル 255
SIGXCPU シグナル 254
single コマンド 57、179
sm_timeout パラメータ 294、295
sprofil システム・コール 244
sshmseg パラメータ 204、266
stat システム・コール 244
subnetsarelocal パラメータ 285、292
Sun NIS マスター 172
su コマンド 118、226、328
svc_maxdupreqs パラメータ 294、297
svr3pipe パラメータ 284、290
swap コマンド 149、150、338
swap 変数 189
sysadm
 chgloginid 121
 chgpasswd 121
 chgshell 121
 電源の切断 48
syslogd メッセージ 331
syslog コマンド 37
sysmon コマンド 37
syssegsz パラメータ 204、234、239

SystemPartition 変数 103、190
sysctl コマンド 203、255、298
sysctl システム・コール 319

T

tar コマンド 117
tcp_hashtablesz パラメータ 298、299、303
tcp_recvspace パラメータ 299、302
tcp_sack パラメータ 299、304
tcp_sendspace パラメータ 299、302
TCP/IP 起動プロトコル 196
TCP/IP ソケット 297
tcsh シェル 17
telinit コマンド 56
telnet コマンド 19
timed コマンド 72、83、332
timers マン・ページ 293
timetrim パラメータ 293、294
timex コマンド 207、341
tlbdrop パラメータ 256、262
top コマンド 158、220
tty_auto_strhold パラメータ 284、287
TZ 環境変数 80

U

udp_hashtablesz パラメータ 298、299、301
UDS ソケット 297
umask コマンド 133
uname コマンド 70

/unix.install ディレクトリ 226
unpdg_recvspace パラメータ 299、301
unpdg_sendspace パラメータ 299、300
unpst_recvspace パラメータ 299
unpst_sendspace パラメータ 298、299
unsetenv コマンド 179、192
untrusted_user オプション 167
use_old_serialnum パラメータ 285、291
usinit ルーチン 342
/usr/include/sys/immu.h ファイル 258
/usr/news ディレクトリ 138
utrace_bufsize パラメータ 234、241

V

/var/adm/crash ディレクトリ 143、148
/var/adm/SYSLOG ファイル 48、206、294、324、328、332
/var/sysgen/master.d/kernel ファイル 206
/var/sysgen/master.d ディレクトリ 203、227、231
/var/sysgen/mtune ディレクトリ 203、227、231
/var/sysgen/system ディレクトリ 227
verbose 変数 189
versions コマンド 68
version コマンド 179
vfs_syncr パラメータ 256、257、263
vfssw_extra パラメータ 281、283
vhand プログラム 255
Video In No Out (VINO) パラメータ 304
vino_mtune_dmrpages パラメータ 304

vi コマンド 18
volume 変数 186

W

wall コマンド 140
whodo コマンド 4、5
write コマンド 139
write システム・コール 342
w コマンド 5

X

xfontsel コマンド 21
xhost コマンド 76
xpc ドライバ 92
xpg4_sticky_dir パラメータ 284、287
xtab_verification オプション 167
xvm_max_revive_rsc パラメータ 319、320
xvm_max_revive_threads パラメータ 319、320
XVM パラメータ 319
xwsh コマンド 19

Z

zone_accum_limit パラメータ 257、263

あ

アーカイブ 144
アカウント管理 107

アカウントのパスワード 2
アクセス権 84、133
 デフォルト 3
 ファイル 84
アクセス・コントロール・リスト 89
新しいユーザ・アカウント 111
新しいユーザ・グループ 116
アップグレード
 ソフトウェア 4
 ハードウェア 4
アプリケーション、調整 201
アプリケーションの調整 201、341

い

イーサネット
 起動 196
 ゲートウェイ 197
 ケーブル 327
移行メモリ・キュー・パラメータ 318
一覧
 ハードウェア 61
一般的なパラメータ 233
色、選択 20
インターネット・アドレス 326

う

ウィンドウ
 カスタム・シェル 19
 リモート・ワークステーション 18

え

エディタ 120
エラー・メッセージ 206、265、323

お

大きなページのパラメータ 229
オプション、システム・ソフトウェア 73
オペレーティング・システム
 再構築の問題 329
 調整 201、223
 バージョン 70
オペレーティング・システムの調整 231

か

カーネル・チューニング 231
カーネル・チューニング・パラメータ 201、231
カーネルのコピー 225
カスタマイズ・ソフトウェア 73
カスタム・シェル・ウィンドウ 19
仮想スワップ領域 150、152
仮想メモリ 221
合体 229
環境変数
 確認 131
 コマンド変数 186
 設定 132
環境変数の設定 132
管理システムの概要 9
管理パーティション 77

き

キーボード変数 191

起動

bootp 197

fx 194

ゲートウェイ 197

コマンド 193

コマンド・モニタ 192

固有プログラム 193

ディスクまたはほかのデバイス 200

デフォルトのファイル 193

ネットワークを介した～ 196

起動環境 175

起動時に記録されるメッセージ 332

起動できないカーネル 228

機能のリスト 70

行エディタ 181

く

グラフィックス・ハードウェア 69

グループ

メンバーの確認 119

グループ・アカウント

ID 番号 108

一時的なグループ 119

削除 118

ユーザ・グループの変更 119

クロックの問題 331

け

ゲートウェイ 197

こ

コア・ファイルのパーミッション 321

コア・ファイルのパーミッションの設定 321

構成ファイル 61、167

構文 182

PROM モニタ 182

国際キーボード 191

コマンド

エリアシング 15

構文 182

ツール 9

モニタ 179

コマンド (PROM) モニタ・プログラム 175

コマンド・モニタ

環境変数 186

キーボード変数 191

起動 176、192

コマンド行エディタ 182

コマンド・リスト 179

再初期化 185

デバイス名 184

ファイルの構文 182

変数の変更 191

コマンド履歴 16

コメント・フィールド 125

コンソール診断 48

さ

- 再帰コマンド 24
- 再構築 224
- サイトの方針
 - アカウント 2
 - ディスクの使用 143
 - パスワード 2
 - プライバシー 3
 - ユーザ ID 番号 2
 - ルート・アカウントへのアクセス 2
- 作業の実行からの CPU の除外 78
- サポートされている構成 101
- 参考文献 363

し

- シェル
 - コマンド 111
 - スクリプト 14
 - 変数 131、132
 - ユーザのデフォルト 125
- シェル・ウィンドウ
 - 色 20
- シェル・ウィンドウのフォントの選択 20
- シェル・スタートアップ・ファイル 125
- 時間帯 80
- 時刻と日付 83
- システム
 - エラー・メッセージ 323
 - 再構築 224
 - 状態 48
 - 調整 201、231
 - ディレクトリ 350

- 名前 78
- パーティション 89
- システム監視ツール 32
- システム・クロック 331
- システム構成
 - アクセス・コントロール・リスト 89
 - オプション 70
 - 確認 70
 - システム・ディスプレイ 75
 - システム名 78
 - ソフトウェア・オプション 73
 - ディレクトリのアクセス権 84
 - デフォルト 75
 - 日付と時刻 83
 - プロセッサ割当て 76
 - 変更 73
- システム・コール 206
- システム・コールのエラー 265
- システムの起動 47、332
- システムの構成
 - 確認 61
- システムのシャットダウン
 - 通知 2
- システムの調整 231
- システムの停止 48
 - `/etc/inittab` 59
 - マルチユーザ・モード 48
- システムのバックアップ 224
- システム・パスワード 185
- システム・ファイル 350
- システム・マネージャ 9
- システム・リミット・パラメータ 242
- システム・ログ

ハードコピー 6
ビューア 37
実行レベル 50
使用率モニタ 38
ショートカット
C シェル 14
Korn シェル 17
ジョブ制御 15、18
シングルユーザ・モード 58
シンボリック・リンク 244

す

スイッチ・パラメータ 283
スケーラビリティ 90
スケジュール 25
スタンドアロン・シェル 194
ストライプ化 214
スワップ領域 144、148、216、333、336

せ

正規表現 12
制限のあるシェル 126
専用スワップ領域 155

そ

ソケット・パラメータ 297
その日のメッセージ 136
ソフトウェア・アップグレード 4
ソフトウェア構成オプション 73

ソフトウェアのカスタマイズ 73
ソフトウェア目録 68

た

タスクの自動化 25

ち

チューニング・パラメータ 231
調整 201

つ

追加
グループ 116
通知、トラブル 7
ツール 9

て

ディスク集約的処理 157
ディスクの使用
サイトの方針 143
ストライプ化 214
入出力 212
パーティション 67、147、214
フォーマット 194
領域に関するメッセージ 323
ディスクの使用状況を表示するコマンド 143
ディレクトリ
IRIX リスト 349
root 349

アクセス権 84
 パーミッション 86
 ファイル・パーミッション 86
 デモン 70
 テキストの操作 29
 デジタル・メディア・リング・バッファ 304
 デバイス特殊ファイル 354
 デバイス・ファイル 354
 デバイス名 184
 デフォルト・グループ 123
 デフォルト・シェル 125
 変更 125
 デフォルトのパーミッション 133
 デフォルト・プリンタ 80
 デフォルト・ホーム・ディレクトリ 124
 電子メール 136
 電力障害 329

と

特別なログイン・シェル 135
 特権ユーザ・グループ 119
 トラブルシューティング
 famの問題 169
 システム・エラー・メッセージ 323
 ディスクの使用 148
 トラブルの通知 7

に

入力のリダイレクト 15

ね

ネットワーク・アドレス 326
 ネットワークの起動 196、328
 ネットワーク・ファイル・システム 147、294

は

は xliv
 バークレイ版のリモート・シェル 127
 パーティション
 ガイドライン 102
 使用 89
 設定 93
 通信 92
 パーティション間の通信 91
 パーティションの分割
 PROM からのパーティションの分割 95
 欠点 91
 パーティション・サイズ 90
 利点 91
 ハードウェア・アップグレード 4
 ハードウェア・グラフ 67
 ハードウェアの一覧 61
 ハードウェアのメモリ・バリア 91
 ハードウェア目録 61
 ハード・ディスク 148
 パーミッション
 umask コマンド 88
 ファイル 86
 変更 87
 パスワード
 PROM 185

- アカウント 2
 - システム 185
 - 変更 121
 - 忘れてしまった場合 121
 - パスワード・ファイル 2、111
 - レイアウト 111
 - バックアップ 224
 - バッファ・キャッシュ 235
 - パーティション
 - 構成ツール 96
 - パーティション構成ツール
 - 使用上の必要条件 96
 - パニック・メッセージ 334
 - パフォーマンスの調整 201、231
 - パラメータ
 - 大きなシステムの調整 204
 - カーネル 201、231
- ひ
- 表記上の決まり xlvii
- ふ
- ファイル
 - IRIX リスト 349
 - カーネル調整 201
 - 操作 21
 - ファイルの圧縮 144
 - ファイルのパーミッション 325
 - ファイル変更モニタ 167
 - ファイル名の構文 182
 - ファイル名の補完機能 14
- フォントの選択 20
 - 不揮発性 RAM 186
 - 複数のページ・サイズ 229
 - 不正行為 5
 - 物理的なメモリ・ページ数 253
 - プリンタ 70、80
 - ブロードキャスト・メッセージ (/etc/wall) 5
 - プロジェクト ID 番号
 - chproj 109
 - newproj 109
 - プロセス
 - cpuset 164
 - 監視 157
 - 終了 162
 - 優先順位 160
 - プロセス間通信 (IPC) 構造 265
 - プロセスの絶対的な優先順位 161
 - プロセッサ割当て 76
 - ブロック転送エンジン 91
- へ
- ページ・サイズ 229
 - ページのフォールト 217
 - ページング 149、216、336
 - ページング・パラメータ 255
 - ヘルプの入手 181
 - 変更
 - デフォルト・シェル 125
 - パスワード 121
 - 変数
 - 解除 192

環境 191
キーボード 191
起動ファイル 193
削除 192
パス 193

ほ

ホーム・ディレクトリ 124
ホスト名 78

ま

マウスのショートカット 28
マルチディスク・システム 155
マルチプロセッサ・システム 76
マルチユーザ環境 107
マン・ページ xliv、31

め

メール 136
メタキャラクタ 12
メタルータ 102
メモリ集約的処理 157
メモリのメッセージ 333

も

目録
ソフトウェア 68
ハードウェア 61

問題の抑制 90

ゆ

ユーザ ID 番号 2、107
/etc/passwd 107
ユーザ・アカウント
確認 115
グループ ID 番号 108
コメント・フィールド 123
削除 117
情報の変更 120
デフォルト・グループ 123
デフォルト・シェル 125
パスワード 121
パスワードの変更 121
ホーム・ディレクトリ 124
ユーザ ID 番号 107
ユーザ環境 125
ログイン ID 番号 122
ログイン・シェル 126
ログイン名 120
ロックされたアカウント 118
ユーザ環境
説明 125
特別なシェル 135
ログイン・シェル 126
ユーザ環境変数 131
ユーザ通知メッセージ (/etc/motd) 136
ユーザ通知メッセージ・ファイル 4
ユーザ・トラブル・ログ 6
ユーザの通信 136、138、139
ユーザ・プライバシー 2
ユーザ・プロセス 157

ユーザへの通知 2
優先順位 160、162

ら

ラージ・ページ・パラメータ 305

り

「リセット (Reset)」ボタン 192
リソース・リミット・パラメータ 245
リモート・ログイン・メッセージ 138
リモート・ワークステーション 18

る

リモート・ログイン・メッセージ 138

ろ

ロード可能ドライバ・パラメータ 281
ロードの調整 165
ログイン
ID 番号 122
アイコン 72
禁止 27
問題 328
ログインシエル 126
論理スワップ領域 164
論理ボリューム 213、319