



OpenWindows ユーザーズガイド (上級編)

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

Part Number 806-2968-10
2000年3月

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

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

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

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

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリコービイマジクス株式会社からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人 日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェイスマスタをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、フォントとして無断複製することは禁止されています。

Sun, Sun Microsystems, docs.sun.com, AnswerBook, AnswerBook2 は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

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

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社で開発されたソフトウェアです。(Copyright OMRON Co., Ltd. 1999 All Rights Reserved.)

「ATOK」は、株式会社ジャストシステムの登録商標です。

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

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

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

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

Unicode は、Unicode, Inc. の商標です。

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

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

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

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われないものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *OpenWindows Advanced User's Guide*

Part No: 806-2902-10

Revision A



目次

- はじめに 13
- 1. SunOS へのログインと OpenWindows の起動 19
 - ログインの方法 19
 - ログインシェル 20
 - ログアウトの方法 21
 - ショートカットキー 21
 - OpenWindows を起動する前に 23
 - OPENWINHOME 環境変数 24
 - 適切な起動ファイルを使う 25
 - OpenWindows ユーザ環境の起動 27
 - OpenWindows の実行 27
 - OpenWindows が実行されない場合 29
 - OpenWindows 環境の終了 29
 - OpenWindows の起動オプション 30
 - ネットワークセキュリティを限定した起動 31
 - 各種モニタを使った起動 31
 - SPARC: 複数のモニタを使った起動 31
- 2. SunOS の基本コマンド 35

コマンドプロンプト	35
コマンドの入力	36
誤った入力の訂正	36
複数のコマンドと長いコマンドの入力	37
前回のコマンドを繰り返す	37
コマンドオプションの指定	39
コマンド出力のリダイレクトとパイプ	39
バックグラウンドでコマンドを実行する	40
OS コマンドによるヘルプ情報の表示	41
man コマンドによるマニュアルページの表示	41
whatis コマンドによる 1 行要約の表示	42
apropos によるキーワードの検索	42
3. ファイルとディレクトリの操作	43
ファイルの概念	43
ファイル操作コマンドの使い方	44
ファイル操作を始める前に	44
テスト用ファイルの作成	45
ファイルの一覧表示 (ls)	45
ファイルのコピー (cp)	45
ファイルの移動とファイル名の変更 (mv)	46
ファイルの削除 (rm)	46
ファイル内容の表示 (more、cat)	47
ファイル形式の表示 (file)	47
ディレクトリとディレクトリ階層	48
ディレクトリ階層	48
作業用ディレクトリの表示 (pwd)	49
ユーザのホームディレクトリ	50
作業用ディレクトリの変更 (cd)	50

	ディレクトリの作成 (mkdir)	52
	相対パス名	52
	ディレクトリの移動とディレクトリ名の変更	52
	ディレクトリのコピー	53
	ディレクトリの削除 (rmdir)	53
	ファイル間の相違点を検出する (diff)	54
	3つのファイルを比較する (diff3)	55
	大きいファイルを bdiff を使って比較する	56
	ファイル名の検索 (find)	56
	ファイルとディレクトリのセキュリティ	58
	アクセス権と状態の表示 (su ls -l)	59
	隠しファイルの表示 (ls -a)	60
	アクセス権の変更 (chmod)	61
	絶対アクセス権の設定	64
4.	ファイルの内容の検索	67
	grep によるパターンの検索	67
	フィルタとして grep を使う	68
	複数ワード文字列の検索	69
	特定の文字列を含まない行の検索	70
	grep の詳細な使い方	70
	メタキャラクターの検索	72
	コマンド行で単一引用符と二重引用符を使う	73
5.	パスワード、プロセス、ディスク記憶装置	75
	パスワードの使い方	75
	パスワードの変更方法	76
	パスワードの有効期限を設定する	77
	プロセスと PID	78
	現在実行されているコマンドの表示 (ps)	78

	プロセスの終了方法 (kill)	79
	ディスク記憶容量の管理	80
	ディスクの使用状況を表示する (df -k)	80
	ディレクトリの使用状況を表示する (du)	80
6.	vi エディタの使い方	81
	vi の起動	82
	ファイルの作成	82
	状態表示行	83
	vi の 2 種類のモード	83
	入力モード	83
	コマンドモード	84
	セッションの終了	85
	ファイルの保存と vi の終了	85
	ファイルの印刷	87
	vi の基本コマンド	87
	ファイル内の移動	87
	テキストの挿入	90
	テキストの変更	91
	変更の取り消し	93
	テキストの削除	94
	テキストのコピーと移動 — yank、delete、put	95
	カウントを使ってコマンドを繰り返す	97
	ex コマンドの使い方	97
	行番号の表示と非表示	98
	行のコピー	98
	行の移動	99
	行の削除	99
	vi による検索と置換	100

文字列の検索	100
高度な検索	101
文字列の置換	102
指定行へジャンプする	103
編集中のファイルに別ファイルを読み込む	103
複数のファイルの編集	104
複数ファイルの編集	105
ファイル間で行をコピーする	105
vi パラメータの設定状況	106
クラッシュからの回復	106
vi の基本コマンドの一覧	106
7. メール	113
mailx の基本	113
mailx の起動	114
自分宛てにメールを送信する	114
自分宛てのメールの読み方	115
mailx の終了	116
メールの読み方	117
メールの削除と削除の取り消し	118
メールの印刷	119
メールの送信	120
送信できないメール	122
送信しないメールの取り消し	122
カーボンコピーとブラインドカーボンコピーの追加	122
メールやファイルのコピーを挿入する	123
メールへ応答する	124
メールの保存と読み取り	125
メールをファイルに保存およびコピーする	125

- メールをフォルダに保存およびコピーする 126
- ファイルやフォルダ内のメールを読み取る 128
- mailx での vi の使い方 129
- メールエイリアス 130
 - .mailrc ファイルでメールエイリアスを設定する 130
 - /etc/aliases でメールエイリアスを設定する 132
- チルドコマンド 135
- ヘルプコマンド: その他の mailx コマンド 137
- 8. プリンタの使い方 139**
 - 印刷要求の実行依頼 139
 - デフォルトプリンタへの印刷要求の実行 140
 - プリンタ名を指定した印刷要求 140
 - 印刷完了通知の要求 141
 - 複数部数の印刷 142
 - lp コマンドのオプション 142
 - プリンタの状態を確認する 143
 - 印刷要求の状態を確認する 143
 - 利用可能なプリンタを確認する 144
 - 状態情報をすべて表示する 144
 - プリンタの状態を表示する 145
 - プリンタの特性を表示する 146
 - lpstat コマンドのオプション 146
 - 印刷要求の取り消し 148
 - ID 番号による印刷要求の取り消し 148
 - プリンタ名による印刷要求の取り消し 148
- 9. ネットワークの使い方 151**
 - ネットワークの一般概念 151
 - リモートログイン (rlogin) 152

	ホームディレクトリなしの rlogin	153
	現在のログイン名以外での rlogin	153
	未登録のマシンに対する rlogin	154
	rlogin セッションの中止	154
	rlogin セッションの中断	155
	ログイン状態の確認 (who am i)	156
	ファイルのリモートコピー (rcp)	156
	リモートマシンからローカルマシンへのコピー	156
	ローカルマシンからリモートマシンへのコピー	157
	コマンドのリモート実行 (rsh)	157
	ユーザ情報の参照 (rusers)	158
10.	動作環境のカスタマイズ	161
	初期設定ファイル	161
	環境変数	162
	ユーザプロファイル	163
	PATH 環境変数の設定	165
	別名 (C シェルのみ)	166
	コマンドプロンプトの変更	166
	その他の便利な変数	168
	デフォルトのファイルアクセス権を設定する	169
	OpenWindows フォントのカスタマイズ	170
	フォントの書体とポイントサイズを指定する	170
	利用可能なフォントを表示する	173
	カラーモニターの調整	174
	モニター調整の概要	175
	表示環境の調整	177
	調整パックの接続	181
	調整ツールの実行	181

エラーメッセージ 188

A. OpenWindows バージョン 3.3 以降への移行 191

SPARC: SunView 環境からの移行 191

.defaults ファイルと .Xdefaults ファイル 191

バージョン 3.3 より前の OpenWindows 環境から移行する 192

OPENWINHOME 環境変数 192

.xinitrc ファイル 193

適切な起動ファイルの使い方 194

ワークスペース・プロパティ 195

ワークスペースメニューのカスタマイズ 196

B. Solaris 2.5 への移行 197

SunOS 4.x からの移行 197

SPARC: SunOS 4.x ユーザに影響のある変更 197

SPARC: SunOS 4.x システム管理者に影響のある変更 198

SPARC: SPARC システム用 SunOS 4.x リリースとの互換性 200

IA: Solaris 2.1 からの移行 201

IA: ユーザに影響のある変更 201

IA: システム管理者に影響のある変更 202

IA: Solaris 2.5 システム管理ツール 203

C. キーボードの設定変更 205

Compose キーの設定解除と設定 205

SPARC: 左きき用のキーの再マッピング 206

SPARC: 再マッピングスクリプトの使い方 206

SPARC: キーボードの再マッピングの取り消し 208

IA: ファンクションキーとコントロールキーの再マッピング 211

IA: 再マッピングスクリプトの使い方 211

IA: キーボードの再マッピングの取り消し 212

D. ネットワークアプリケーションの実行 217

	rlogin を使用してネットワークアプリケーションを実行する	218
	セキュリティに関する詳細	219
	対象読者	219
	アクセス制御機構	219
	認証プロトコル	221
	サーバへのアクセスを制御する	222
	クライアントをリモートで実行する場合とローカルで実行する場合	226
E.	SPARC — DECnet インターネットワーキング (DNI)	229
	DECnet インターネットワーキングの設定	229
	OpenWindows のマシン上でリモートクライアントを表示する	231
	VAX のマシン上でリモートクライアントを表示する	231
F.	システムの管理	235
	Administration Tool の起動	236
	sysadmin グループへのユーザ登録	237
	共通作業に Administration Tool を使用する	239
	ホストの管理	239
	プリンタの管理	239
	シリアルポートの管理	240
G.	PCMCIA カードの使い方	241
	はじめに	241
	サポート要件	242
	参考資料	242
	PCMCIA メモリカードの使い方	242
	ファイルのコピー方法	243
	書き込み保護モード	243
	PCMCIA メモリカードと電源管理システムの保存停止・復元再開機能	244
	tar コマンドによるファイルのコピー	245

- ▼ PCMCIA メモリカードをフォーマットするには 245
- ▼ tar コマンドを使ってファイル名を表示するには 247
 - ファイルのコピー 248
- ボリューム管理の設定によるファイルのコピー 251
- ▼ ラベルなし PCMCIA メモリカードをフォーマットするには 252
- ▼ PCMCIAメモリカードを再フォーマットするには 253
 - ファイルのコピー 255
- ボリューム管理の設定解除によるファイルのコピー 257
- ▼ ボリューム管理の設定を解除するには 257
- ▼ PCMCIA メモリカードをフォーマットするには 257
- ▼ PCMCIA メモリカードをマウントするには 259
 - ファイルのコピー 260
- PCMCIA シリアル/モデムカードの使い方 261
 - PCMCIA シリアル/モデムカードのデバイス名 261
 - PCMCIA シリアル/モデムカードと電源管理システムの保存停止・復元再開機能 261
- 索引 263

はじめに

対象読者

このマニュアルは、Solaris™ システムソフトウェア (バージョン 2.x) のユーザを対象にしています。Solaris は、SunOS™ と OpenWindows™ から構成されています。Solaris 1.x から Solaris 2.x へ移行する場合は、付録 B を参照してください。

前提条件

システムがインストール済みで使用可能な状態になっている必要があります。システムをインストールしていない場合は、作業を進める前にシステムに付属のインストールマニュアルを参照してください。

関連マニュアル

AnswerBook™ を使って、Solaris に関するさまざまなマニュアルを利用できます。これらのマニュアルは、次のようなマニュアルセットで構成されています。

- システム管理者用マニュアルセット

Sun のワークステーションを使った大規模ネットワークなどのさまざまなシステム構成のインストレーションとシステム管理に関する詳細な情報を提供します。

- 開発者用マニュアルセット

ソフトウェア開発者がシステム上のプログラムを作成、デバッグ、維持管理するために必要な情報を提供します。

- リファレンスマニュアルセット

SunOS の全コマンドについて解説します。これらの解説は、マニュアルページと呼ばれるオンラインマニュアルとしてインストールすることもできます。

- 一般ユーザ用マニュアルセット

SunOS システムのさまざまな特徴について解説します。SunOS のコマンドの使い方、OpenWindows の使い方、作業環境のカスタマイズ方法、問題解決の方法、シェルスクリプトの書き方、電子メールの使い方、ネットワークの使い方などが記載されています。

Sun のマニュアルの注文方法

専門書を扱うインターネットの書店 Fatbrain.com から、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、<http://www1.fatbrain.com/documentation/sun> の Sun Documentation Center をご覧ください。

表記上の規則

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

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep `^#define \ XV_VERSION_STRING'

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

コード例は次のように表示されます。

■ C シェルプロンプト

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

■ Bourne シェルおよび Korn シェルのプロンプト

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

■ スーパーユーザーのプロンプト

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

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

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

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

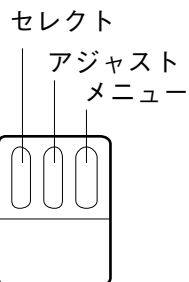
ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

注 - 文字キーは大文字で表記されていますが (上記の Ctrl-D の D キーを参照)、これは、「Shift キーを押して大文字にする」という意味ではありません。特に指定しない限り Shift キーは押さないでください。

注 - Solaris は、SPARC™ と IA マシンの両方をサポートしています。これらはキーボードの配列が異なるため、OpenWindows メニューのコマンドを使用する場合、キーボードによって操作が異なることがあります。

それぞれのプラットフォームで共通にキーを使用できるように、Solaris ではメタキーの組み合わせをサポートしています。コマンド機能と対応するショートカットキーについては、第 1 章を参照してください。

- マウスボタンの表記は、ボタンの位置ではなくその機能で示します。たとえば文章の中では、「左ボタンをクリックする」ではなく、「セレクトボタンをクリックする」と表現します。



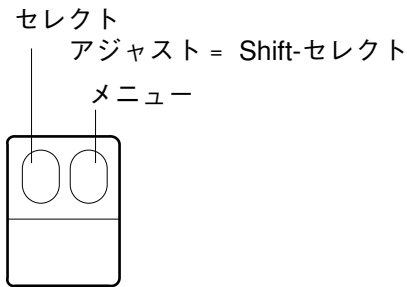


図 P-1 マウスボタンのデフォルトの機能割り当て

3 ボタンマウスの機能は、デフォルトで次のように割り当てられています。

- セレクトボタン=左ボタン
- アジャストボタン=中央のボタン
- メニューボタン=右ボタン

2 ボタンマウスの機能は、デフォルトで次のように割り当てられています。

- セレクトボタン=左ボタン
- メニューボタン=右ボタン

2 ボタンマウスでアジャスト機能を使用する場合は、キーボードの **Shift** キーを押しながらセレクトボタンをクリックしてください。

- 特定の操作については、メタキーと他のキーを組み合わせることによって行うことができます。SPARC キーボードではメタキーは **<>** キーを意味し、IA キーボードではメタキーは **Ctrl** キーと **Alt** キーを同時に押す (**Ctrl-Alt**) ことを意味します。キーボードについての詳細は、第 1 章を参照してください。
- 本文の説明の中でメニュー全体について、またはデフォルトの項目について説明している場合に、メニュー項目中のデフォルトで選択される項目は枠で囲まれています。メニューボタンの上で (メニューを表示しない状態で) セレクトボタンをクリックすると、デフォルトの項目を選択できます。

プラットフォーム固有の表記規則

節または手順のタイトルの前に示される次の記号は、その節または手順が特定のプラットフォームだけに該当することを示します。

SPARC: SPARC プラットフォームだけに該当する節または手順であることを示します。

IA: IA プラットフォームだけに該当する節または手順であることを示します。

プラットフォームに固有な注意事項は次のように示します。

注 **[SPARC]** - この注が SPARC プラットフォームだけに該当することを示します。

注 **[IA]** - この注が IA プラットフォームだけに該当することを示します。

記号の説明項目がプラットフォームに固有な場合は、次のように示します。

- **SPARC:** この記号の説明が SPARC プラットフォームだけに該当することを示します。
- **IA:** この記号の説明が IA プラットフォームだけに該当することを示します。

コーディング例、表、図などがプラットフォームに固有な情報である場合、次に示すようにタイトルの前にプラットフォーム名を示します。

表 1-1 SPARC: SPARC プラットフォームだけに該当する表題

表 1-2 IA: IA プラットフォームだけに該当する表題

一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が、適宜併記されています。
- このマニュアルでは、「IA」という用語は、Intel 32 ビットのプロセッサアーキテクチャを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Celeron、Pentium III、Pentium III Xeon の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。

SunOS へのログインと OpenWindows の起動

OpenWindows ユーザ環境を起動するには、OpenWindows をファイルシステム内のアクセス可能なサーバ上またはハードディスク上にインストールしておく必要があります。インストール済みかどうか不明な場合は、システム管理者に問い合わせるか、ご使用のプラットフォーム用のインストールマニュアルを参照してください。

この章では、システムへのログイン方法、シェルコマンド・インタプリタの使い方、OpenWindows ユーザ環境のマウントおよび起動方法、ウィンドウシステムの終了方法、システムからのログアウト方法について説明します。また、OpenWindows 環境を複数のモニタ用に構成するといった特殊なケースについても説明します。

ログインの方法

一般に作業セッションとは、ユーザがシステムにログインしてからログアウトするまでを指します。SunOS のマルチユーザ環境では、システムを利用するたびに識別名を入力することが必要です。システムがユーザを識別し、またそのユーザを他のユーザと区別するための識別名として、ログイン名 (ユーザ名またはアカウントとも呼ばれる) が使われます。また、パスワードを使うことによって、それを知っているユーザだけがアカウントを利用できるよう制限できます。ログイン名とパスワードを持っていない場合は、システム管理者に問い合わせてアカウントを設定してください。ログイン名とパスワードが設定されれば、ログインできる状態になります。

システムにログインするときは、画面に次のような文字が表示されていなければなりません。

```
login:
```

システム管理者によって割り当てられたログイン名を入力して **Return** キーを押します。たとえば、ログイン名が `spanky` の場合は次のように入力して **Return** キーを押します。

```
login: spanky
```

次に、パスワードを入力するよう要求されます。

```
login: spanky  
Password:
```

プロンプトに対してパスワードを入力して **Return** キーを押します。(アカウントにパスワードが割り当てられていない場合は、パスワードを入力しなくてもログインできます。) パスワードは入力しても画面には表示 (エコー) されません。これは、他のユーザにパスワードを知られないようにするためです。

ログインシェル

2 章以降では、SunOS のコマンドを使います。システムに対してコマンドを入力すると、実際にはシェルと呼ばれるコマンド・インタプリタプログラムに情報を提供することになります。シェルプログラムは提供された情報を読み取り、適切なアクションをシステム内部で実行させます。

SunOS のデフォルトシェルは Bourne シェルですが、オペレーティングシステム内部で C シェルおよび Korn シェルという別の 2 種類のシェルプログラムを利用することもできます。これらのシェルには、それぞれ独自の機能があります。

注 - `man` コマンドで各シェルプログラムを含む SunOS の各コマンドのマニュアルページを表示できます。マニュアルページについての詳細は、第 2 章の 41 ページの「`man` コマンドによるマニュアルページの表示」を参照してください。

初めてシステムにログインした (またはコマンドツールやシェルツールのウィンドウを新たに開く) ときにコマンドプロンプトが表示された場合は、シェルプログラムが自動的に起動されたことを意味します。このシェルは、ログインシェルと呼ばれま

す。システム管理者が別のシェル (C シェルまたは Korn シェル) を設定している場合は、ログインシェルは SunOS のデフォルト (Bourne シェル) にはなりません。

すでに述べたように、各シェルには独自の機能があります。あるシェルで利用できるコマンドや手続きが他のシェルでは利用できない場合もあります。この点を考慮してこのマニュアルでは、SunOS のデフォルト (Bourne シェル) で利用できないコマンドや手続きを例示する場合は、該当する箇所でその旨を明示しています。

ログアウトの方法

作業セッションを完了してオペレーティングシステムを終了する準備ができたなら、次のように入力してログアウトします。

```
$ exit
```

しばらくすると、再びログインプロンプトが表示されます。

```
$ exit  
login:
```

ログインプロンプトが表示された場合は、正常にログアウトできたことを意味し、任意のユーザが続けてログインできる状態になります。

注 - SunOS オペレーティングシステムでは、ワークステーションや端末の電源を切ってもログアウトできるとは限りません。明示的にログアウトしない場合、システムにログインしたままになることがあります。

ショートカットキー

OpenWindows メニューの拡張機能やマウスを必要としない操作については、場合によっては特定のキー操作の組み合わせを使用することによって (キーボードアクセラレータと呼ぶ) これらの操作をスピードアップすることができます。これによりマウスとメニュー操作と同等の操作を行ったり、前もって定義したキーボードのキー操作を行うことができます。

次の表に SPARC と IA ベースのマシン用の一部のコマンド操作と、対応するショートカットキーを示します。

注・メタキーは SPARC キーボード上では <> キーを意味し、IA キーボード上では Ctrl-Alt キーを押すことを意味します。

キーボードアクセラレータを使用するには、最初のキー (メタキー、または Ctrl-Alt キー) を押しながら 2 番目のキーを押します。たとえば、選択したテキストをカットするには、SPARC システムではメタキーを押しながら x キーを押します。IA システムでは Control キーと Alt キーを一緒に押しながら x キーを押します。

表 1-1 キーボードアクセラレータ

操作	ショートカット キー	動作
再実行 (Again)	メタ-a	直前の動作を繰り返す
コピー (Copy)	メタ-c	選択したものをクリップボードにコピーする
カット (Cut)	メタ-x	選択したものを削除し、クリップボードに入れる
検索 (Find)	メタ-f	選択した内容をキャレット以降から検索する
ヘルプ (Help)	Help または F1	ポインタが示す位置のオブジェクトについてヘルプ情報を表示する
新規 (New)	メタ-n	新しいファイルを開く
オープン (Open) (ファイル)	メタ-o	ファイルを開く (たとえば、ファイルマネージャでファイルのアイコンを強調表示してある場合)
オープン (Open) (ウィンドウ)	メタ-w	アイコンを開いてウィンドウにするか、ウィンドウを閉じてアイコンにする
ペースト (Paste)	メタ-v	クリップボードの内容を挿入ポイントにコピーする
印刷 (Print)	メタ-p	ファイルをプリンタに送信する (たとえば、ファイルマネージャでファイルのアイコンを強調表示してある場合)
プロパティ (Props)	メタ-i	ポインタが示す位置のアプリケーションのプロパティウィンドウを表示する

表 1-1 キーボードアクセラレータ 続く

操作	ショートカット キー	動作
再実行 (Redo)	Shift-メタ-p	取り消し操作 (アンドゥ) を元に戻す
保存 (Save)	メタ-s	現在のファイルを保存する
停止 (Stop)	Stop または Esc	現在の操作を停止する
取り消し (Undo)	メタ-u	直前の操作を取り消す

OpenWindows を起動する前に

OpenWindows を起動するにあたって、次の点を注意する必要があります。これらの注意点に該当しない場合は、この章で後述の 27ページの「OpenWindows の実行」に進んでください。

- ネットワーク上で OpenWindows サーバを設定する場合は、ご使用中のプラットフォームのインストールマニュアルを参照してください。
- 現在、OpenWindows バージョン 2 を使っている場合は、付録 A を参照してください。
- 現在、SunView ユーザ環境を使用している場合は、付録 A を参照してください。
- 現在、OpenWindows バージョン 3.3 より前のバージョンを使用している場合は、この章で後述の 24ページの「OPENWINHOME 環境変数」と 25ページの「適切な起動ファイルを使う」を参照してください。

注 [IA] - Solaris をインストールした後にシステム上でマウス、キーボード、またはビデオアダプタカードの型を変更した場合は、devconfig プログラムを実行し、システム構成を更新する必要があります。devconfig(1M) のマニュアルページを参照してください。

OPENWINHOME 環境変数

現在、OpenWindows バージョン 3.3 より前のバージョンを使っている場合は、ユーザが自分で OPENWINHOME 環境変数を設定している場合があります。OpenWindows バージョン 3.3 では、ユーザが自ら手作業や起動ファイルで OPENWINHOME 環境変数を設定することはお薦めしません。

OPENWINHOME 環境変数は、openwin コマンドを実行すると自動的に /usr/openwin に設定されるため、ユーザ側で設定する必要はありません。

ホームディレクトリ内の .profile ファイルや .cshrc ファイルで OPENWINHOME 環境変数を設定している場合は、該当行をコメントに変更するか削除してから、OpenWindows バージョン 3.3 以降を使用してください。

.profile ファイルや .cshrc ファイル内の OPENWINHOME 環境変数に関する行をコメントに変更したり削除するには、次の手順を実行します。

1. vi などのテキストエディタを使って、.profile ファイルまたは .cshrc ファイルを開きます。
2. 下記に示すように変数名の前にポンド記号 (#) を入力してコメントに変更するか、その行全体を削除します。
.profile ファイルを編集するときは次の手順 a を、.cshrc ファイルを編集するときは手順 b を実行します。
 - a. .profile ファイルの場合

```
# OPENWINHOME=/usr/openwin
```

- b. .cshrc ファイルの場合

```
# setenv OPENWINHOME /usr/openwin
```

3. ファイルを保存してエディタを終了します。
4. OPENWINHOME 環境変数の設定を解除 (unset) します (またはログアウトしてログインし直します)。

Bourne シェルまたは Korn シェルを使っているときは次の手順 a を、C シェルを使っているときは手順 b を実行します。

a. **Bourne** シェルまたは **Korn** シェルでは、次のように入力します。

```
$ unset OPENWINHOME
```

b. **C** シェルでは、次のように入力します。

```
example% unsetenv OPENWINHOME
```

環境変数の設定を解除できたら、この章で後述の 27ページの「OpenWindows の実行」の説明に従って、OpenWindows を起動できます。

適切な起動ファイルを使う

現在、OpenWindows バージョン 3.3 より前のバージョンの OpenWindows を使っている場合は、.xinitrc ファイルの内容を確認する必要があります。.xinitrc ファイルはホームディレクトリ内にある OpenWindows の起動ファイルで、ユーザが定義したオプションが入っています。

.xinitrc ファイルが存在するかどうかを確認するには、次のコマンドを入力します。

```
$ cd  
$ ls -a .xinitrc
```

上記のコマンドの出力内容に応じて、次のどれかを実行します。

- .xinitrc ファイルが存在しない(つまり、上記の `ls -a` コマンドで .xinitrc が表示されない)場合は、何もする必要はありません。ホームディレクトリ内に .xinitrc ファイルが存在しない場合は、OpenWindows はシステムデフォルトの起動ファイルを使います。
- .xinitrc ファイルが存在するが(つまり、上記の `ls -a` コマンドで .xinitrc が表示されるが)そのファイルをまったく変更していない、または以前変更した内容を保持しなくてもよい場合は、次節の 26ページの「起動ファイルに関する手続き」の手順 1 を実行します。

- `.xinitrc` ファイルが存在し (つまり、上記の `ls -a` コマンドで `.xinitrc` が表示され)、以前そのファイルに加えた変更内容を保持したい場合は、次節の 26 ページの「起動ファイルに関する手続き」の手順 2 を実行します。

起動ファイルに関する手続き

1. ホームディレクトリから `.xinitrc` ファイルを削除する場合は、次のコマンドを入力します。

```
$ rm .xinitrc
```

2. `.xinitrc` ファイルの変更内容を保持する場合は、次の手順を実行します。

- a. `.xinitrc` を `.xinit.save` にファイル名を変更します。

```
$ mv .xinitrc .xinitrc.save
```

- b. `/usr/openwin/lib/Xinitrc` をホームディレクトリ内に `.xinitrc` というファイル名でコピーします。

```
$ cp /usr/openwin/lib/Xinitrc $HOME/.xinitrc
```

- c. `.xinitrc.save` の中の保持したい行を `.xinitrc` の中に追加します。



注意 - `.xinitrc` ファイルを編集するときに、`olwm` をデュアル・フレームバッファ用などに追加したり、`svenv` を追加したり、`/usr/openwin/lib/openwin-sys` を含む行を削除したりしないよう注意してください。

OpenWindows ユーザ環境の起動

OpenWindows ユーザ環境を起動する一般的な手順は次のとおりです。

1. **NFS** を使って、**OpenWindows** がインストールされたサーバから **OpenWindows** をマウントします。
サーバから OpenWindows へマウントする方法についての詳細は、*OpenWindows* のマニュアルページを参照するか、システム管理者に問い合わせてください。
2. 必要に応じて起動オプションを指定し、コマンド `openwin` を使って **OpenWindows** を起動します。

OpenWindows の実行

OpenWindows をサーバからマウントし、OpenWindows インストールスクリプトを実行すると、OpenWindows を起動できる状態になります。

OpenWindows を起動するには、シェルプロンプトに対して `/usr/openwin/bin/openwin` と入力して Return キーを押します。

```
$ /usr/openwin/bin/openwin
```

このコマンドによって、OpenWindows バージョン 3.3 の画面が表示され、OpenWindows の動作環境が設定されます。

OpenWindows の簡単な実行方法

OpenWindows を正常に起動できるようになれば、OpenWindows のフルパス名を毎回入力しなくてもすむよう短縮名を設定できます。

Bourne シェルまたは Korn シェルを使っている場合は、`.profile` ファイルにシェル関数を挿入します。C シェルを使っている場合は、`.cshrc` ファイルで別名 (エイリアス) を設定します。`.profile` ファイルと `.cshrc` ファイルは両方ともホームディレクトリにあります。

各シェル用の適切なファイル内で短縮名を設定すれば、次のように入力するだけで OpenWindows を起動できるようになります。

```
$ openwin
```

各起動ファイル内で OpenWindows の短縮名を設定する方法については、次の2つの項で説明します

.profile ファイルでの設定

.profile ファイルで OpenWindows の短縮名を設定する手順は次のとおりです。

1. vi などのテキストエディタを使って .profile ファイルを開きます。
2. 次のシェル関数を .profile ファイルに挿入します。

```
openwin () {  
    /usr/openwin/bin/openwin  
}
```

3. ファイルを保存してエディタを終了します。
4. ログアウトし、ログインし直すことによって短縮名が有効になります。または次のように入力します。

```
$ . .profile
```

これ以降は openwin と入力するだけで OpenWindows を起動することができます。

.cshrc ファイルでの設定

.cshrc ファイルで OpenWindows の短縮名を設定する手順は次のとおりです。

1. vi などのテキストエディタを使って .cshrc ファイルを開きます。
2. 次の alias コマンドを .cshrc ファイルに挿入します。

```
alias openwin /usr/openwin/bin/openwin
```

3. ファイルを保存してエディタを終了します。
4. ログアウトし、ログインし直すことによって短縮名が有効になります。または次のように入力します。

```
example% source .cshrc
```

これ以降は `openwin` と入力するだけで `OpenWindows` を起動することができます。

OpenWindows が実行されない場合

`OpenWindows` は、起動時に `/usr/openwin` ディレクトリを通じてアクセスされます。デフォルトでは、このディレクトリに `OpenWindows` がインストールされています。カレンダーマネージャなどのアプリケーションの多くは、`OpenWindows` が `/usr/openwin` にインストールされていないとロードできません。

`/usr/openwin/bin/openwin` と入力しても `OpenWindows` が起動されない場合は、`OpenWindows` がまったくインストールされていないか、`/usr/openwin` 以外のディレクトリにインストールされている可能性があります。詳細については `OpenWindows` のマニュアルページを参照するか、システム管理者に問い合わせてください。

OpenWindows 環境の終了

`OpenWindows` を実行してウィンドウ環境で作業しているときは、`SunOS` の通常のコマンド行セッションと同様の方法ではログアウトできません。まず最初にウィンドウ環境を終了し、その後でログアウトする必要があります。

シェルプロンプトに対して `logout` と入力しても、次のメッセージが表示されます。

```
Not login shell.
```

OpenWindows 環境を終了する場合は、次の手順に従います。

1. 矢印 (ポインタ) が画面のバックグラウンド (ワークスペース) 上にくるようにマウスを移動します。
2. マウスのメニューボタンを押します。
ワークスペースメニューが表示されます。
3. メニュー上でポインタを下向きにドラッグして、最後のメニュー項目の「終了」を強調表示します。
4. マウスボタンを離します。
ポップアップウィンドウが表示され、ウィンドウシステムを終了するかどうか確認を求めてきます。
5. ポインタを「終了」ボタンに移動し、マウスのセレクトボタンをクリックします。
しばらくすると、すべてのウィンドウが消えて画面の左下隅にシステムプロンプトが表示されます。

OpenWindows の起動オプション

大部分のユーザは、この章で後述の 27 ページの「OpenWindows の実行」で説明した基本的な手順に従って OpenWindows を起動できます。必要に応じて、追加のオプションを使って OpenWindows の起動方法を変更することもできます。

この節では、次に示す特殊なケースについて説明します。

- ネットワークセキュリティを限定した OpenWindows の起動
- 各種モニタおよびフレームバッファを使った OpenWindows の起動
- 複数の画面上での OpenWindows の起動

特殊なオプションで OpenWindows を起動する場合には、次の形式の `openwin` コマンドを使います。

```
$ openwin [ options ]
```

上記の *options* は、ウィンドウサーバのデフォルト設定をカスタマイズするためのコマンド行オプションです。次節以降では、比較的よく使われるオプションをいくつか説明します。

ネットワークセキュリティを限定した起動

開放型ネットワーク環境での操作時に、ネットワークのセキュリティを重視する必要がない場合は、自分のシステムから `-noauth` オプションを使って、ネットワーク上の他のユーザがアプリケーションを実行できるようにします。

次のコマンドはデフォルトのセキュリティ機能を無効にし、自分のウィンドウサーバに他のユーザがアクセスできるように指定します。

```
$ openwin -noauth
```

各種モニタを使った起動

グレースケールのモニタ (8 ビット以上のフレームバッファを持つモノクロモニタ) を使う場合は、OpenWindows の起動時に `grayvis` オプションを指定できます。このオプションは、OpenWindows の表示性能をある程度向上させますが必須ではありません。

このオプションを使う場合には、システムプロンプトに対して次のように入力します。

```
$ cd  
$ openwin -dev /dev/fb grayvis
```

SPARC: 複数のモニタを使った起動

注 [SPARC] - この節の説明は SPARC ベースのマシンだけに適用されます。また、これらの手順を実行するには、システム管理の経験が多少必要です。システムを構成した経験がない場合は、システム管理者に問い合わせてください。

複数の画面上で OpenWindows 環境を実行する場合は、実行したい追加のデバイスとディスプレイの型をシステムに通知してください。起動スクリプト `openwin` で提供されるデフォルトのデバイスを使っても、それ以外のデバイスを指定してもかまいません。

2つのモニタ上で OpenWindows を起動する場合は、`openwin` コマンドで2つのオプションを指定します。

```
$ openwin [ [ -dev device ] [ deviceoptions ] ]
```

二重括弧は、[`-dev device`]と [`deviceoptions`]の組み合わせをコマンド行に複数回(デバイスごとに1つ)指定できることを意味します。

[`-dev device`]

コマンド行オプションの `device` は、ウィンドウサーバがディスプレイ (画面) として使うフレームバッファのデバイスを指定します。

コマンド行でこのオプションを指定しない場合、サーバはデフォルトの `/dev/fb` を使います。コマンド行で複数の `-dev` オプションを指定することによって、同一サーバ上で複数のディスプレイを使うことを指示できます。

[`deviceoptions`]

コマンド行オプションの `deviceoptions` は、`-dev` オプションで指定したデバイスの動作を変更するデバイス修飾子を指示します。

デバイスオプションの例

この節では、2つのモニタを左右または上下方向に配置する例を紹介します。

注 - これらの例では、デバイスの指定順序が重要になります。最初に指定するデバイスは、2番目のデバイスの物理的に左または上に配置されるモニタでなければなりません。また、2番目に指定するデバイスは、最初のデバイスの物理的に右または下に配置されるモニタでなければなりません。

[left][right]

次のコマンド行は、2つのディスプレイを起動するようシステムに指示します。左側のディスプレイはデフォルトのフレームバッファで、右側のディスプレイはモノクロです。この設定によって、2つのディスプレイ間でカーソルを左右に移動できます。

```
$ openwin -dev /dev/fb left -dev /dev/fbs/bwtwo0 right
```

次の例は、前記の例と同じ結果になります。デフォルトでは、コマンド行で最初に指定されたデバイスが左側に、2番目に指定されたデバイスが右側に配置されます。

```
$ openwin -dev /dev/fb -dev /dev/fbs/bwtwo0
```

次のコマンド行は、2つのディスプレイを起動するようシステムに指示します。右側のディスプレイはデフォルトのフレームバッファで、左側のディスプレイはモノクロです。この設定によって、2つのディスプレイ間でカーソルを左右に移動できます。

```
$ openwin -dev /dev/fb right -dev /dev/fbs/bwtwo0 left
```

[top][bottom]

次のコマンド行は、2つのディスプレイを起動するようシステムに指示します。上側のディスプレイはCG6(カラー)で、下側のディスプレイはモノクロです。この設定によって、2つのディスプレイ間でカーソルを上下に移動できます。

```
$ openwin -dev /dev/fbs/cgsix0 top -dev /dev/fbs/bwtwo0 bottom
```

次の例は、前記の例と同じ結果にはなりません。デフォルトでは、コマンド行で最初に指定されたデバイスは左側に、2番目に指定されたデバイスが右側に配置されます。

```
$ openwin -dev /dev/fbs/cgsix0 -dev /dev/fbs/bwtwo0
```

次のコマンド行は、2つのディスプレイを起動するようサーバに指示します。下側のディスプレイはCG6(カラー)で、上側のディスプレイはモノクロです。この設定によって、2つのディスプレイ間でカーソルを上下に移動できます。

```
$ openwin -dev /dev/fbs/cgsix0 bottom -dev /dev/fbs/bwtwo0 top
```

その他の注意点

複数のディスプレイを使う場合は、次の点を十分に考慮してください。

- デフォルトでは、すべてのディスプレイが `o1wm` によって管理されます。
- 複数のディスプレイの間でウィンドウを移動することはできません。

SunOS の基本コマンド

この章では、SunOS オペレーティングシステムのユーザコマンドについて概説します。コマンドの入力方法、入力誤りの訂正方法、複数のコマンドや長いコマンドの入力方法、コマンドオプションの使い方、その他 SunOS コマンドに関する有用な情報などについて説明します。

コマンドを入力する場合は、コマンドツールやシェルツールのウィンドウを使います。これらのウィンドウを表示するには、ワークスペースメニューのプログラムサブメニューを選択します。

コマンドプロンプト

ログインした直後は、初期プロンプト (最初にプロンプトが表示されていること) を除いて画面 (ウィンドウ) には何も表示されません。このプロンプトの形状は、使っているシェルの種類やシステム管理者による初期設定によって異なります。SunOS で使われるデフォルトのコマンドプロンプトはドル記号 (\$) なので、このマニュアルで紹介する大部分の例ではこの \$ プロンプトを使います。

コマンドプロンプトを変更する場合は、第 10 章の 166 ページの「コマンドプロンプトの変更」を参照してください。

コマンドの入力

コマンドプロンプトが表示されているときは、システムがコマンドの入力待ち状態であることを示します。次の例に示すように、コマンドプロンプトに対して `date` というコマンドを入力してみてください (`date` と入力して **Return** キーを押します)。

```
$ date
Mon Feb 3 10:12:51 PST 1992
$
```

この例に示すように、`date` コマンドは現在の日付と時間を表示します。次に、同じコマンドの先頭を大文字にして入力してみてください。

```
$ Date
Date: Command not found.
$
```

上記のように、システムは大文字の `D` と小文字の `d` を区別しています。SunOS オペレーティングシステムの大部分のコマンドは小文字です。

誤った入力の訂正

最初に `Date` と入力して **Return** キーを押す前に誤りに気付いたとします。ユーザが入力したテキストは、**Return** キーを押すまではシステムに送信されません。したがって、**Return** キーを押す前であれば、入力の誤りを訂正できます。誤りを訂正するには、次の2つの方法があります。

- **Delete** キーまたは **Backspace** キーを押して誤った箇所まで戻ります。
- **Ctrl-U** (**Ctrl** キーを押したままで **u** キーを押す) を入力して、行全体を消去した上で最初から入力し直します。

これら2つの方法を試して動作を確認してください。(Delete キーと Backspace キーの動作はシステムによって異なる場合があります。Ctrl-U はほとんどのシステムで同様に動作します。)

複数のコマンドと長いコマンドの入力

1 行に複数のコマンドを入力することもできます。次の `date` コマンドと `logname` コマンドの例に示すように、コマンドの間にセミコロン (;) を挿入してください。

```
$ date; logname
Mon Feb 3 10:19:25 PST 1992
spanky
$
```

上記の例に示すように、現在の日付と時間 (`date` コマンドの出力) およびシステムに現在ログインしているユーザのログイン名 (`logname` コマンドの出力) が続けて表示されます。

非常に長いコマンドを入力する場合は、バックスラッシュ文字 (\) を使うことによって次の行に続けて入力できます。次の例を参照してください。

```
$ date; \
logname
Mon Feb 3 10:23:25 PST 1992
hankw
$
```

`date` コマンドと `logname` コマンドは決して長いコマンドではありません。この例では、複数のコマンドを次の行に続けて入力するという概念を、できるだけ簡単に説明するためにこれらのコマンドを使っています。入力するコマンドが画面の幅よりも実際に長いときは、バックスラッシュ文字を使うと非常に便利です。

注 - OpenWindows 環境でシェルツールやコマンドツールのウィンドウを使っている場合は、次の行にコマンドを続けて入力するためにバックスラッシュ文字を使う必要はありません。入力中のコマンドが行の終わりにくると、次の行に自動的に改行され、Return キーを押したときにそれらの全コマンドが実行されます。

前回のコマンドを繰り返す

注 - この節で説明するコマンド繰り返しの機能は、C シェルを使っている場合にだけ利用できます。

最後に入力したコマンドを繰り返すには、!! を入力して Return キーを押す方法が簡単です。システムは、ユーザが入力したコマンドの履歴を保持しているため、以

前入力したコマンドを繰り返し実行できます。たとえば、最後に入力したコマンドが `date` の場合は次のようになります。

```
example% !!
date
Mon Feb 3 10:26:20 PST 1992
example%
```

`!x` と入力すると、以前に入力した任意のコマンドを繰り返し実行できます。`x` は、繰り返すコマンドに対応する履歴リスト上のコマンド番号です。履歴リストを参照するには、`history` と入力して **Return** キーを押します。次の例に示すようなリストが表示されます。

```
example% history
1 pwd
2 clear
3 ls -l
4 cd /usr/home/worker
5 logname
6 date
7 history
```

また、`!` の後に負の番号を入力しても履歴リスト上のコマンドを繰り返すことができます。たとえば、履歴リスト上の最後のコマンドから数えて 3 番目のコマンドを実行するには、次のように入力します。

```
example% !-2
logname
hankw
example%
```

上記の履歴リストを例にとれば、`logname` コマンドが繰り返し実行されます。

`!` の後に以前のコマンド名の先頭から数文字を入力してもコマンドを再実行できます。たとえば、以前に `clear` コマンドを入力して画面をクリアした場合は、`!c1` と入力すれば画面を再度クリアできます。ただし、この方法によるコマンドの繰り返しでは、繰り返したいコマンドを履歴リスト上で一意に識別できる文字数を指定しなければなりません。`!` の後に 1 文字しか指定しなかった場合は、その文字で始まるコマンドのうちで最後に入力したものが繰り返されます。

コマンドオプションの指定

多くのコマンドにはオプションがあり、そのコマンドに関連する特別機能を実行できます。たとえば、`date` コマンドには、ローカル時間ではなくグリニッジ標準時間で日付を表す `-u` というオプションがあります。

```
$ date -u
Mon Feb 3 11:06:51 GMT 1993
$
```

大部分のオプションは、1文字の前にダッシュ (-) を付けたものです。オプションを持たないコマンドもあれば、複数のオプションを持つコマンドもあります。コマンドに対して引数を持たないオプションを複数指定する場合は、それらのオプションを個別に入力しても (`-a -b`) まとめて入力しても (`-ab`) かまいません。

コマンド出力のリダイレクトとパイプ

特に指定しない限り、コマンドの出力結果は画面上に表示されます。特別な記号を使って、コマンドの出力をリダイレクト (出力先を変更) できます。たとえば、コマンド出力を画面に表示せずにファイルに保存できます。次の例は、リダイレクト記号 (>) の使い方を示しています。

```
$ date > sample.file
$
```

この例では、`date` コマンドからの出力が `sample.file` という新規ファイルにリダイレクトされています。次に、`more` コマンドを使って `sample.file` の内容を表示します。

```
$ more sample.file
Mon Feb 3 12:56:26 PST 1993
$
```

この例に示すように、`sample.file` には `date` コマンドからの出力結果が入っています。(`more` コマンドについての詳細は、第3章を参照してください。)

あるコマンドの出力を別のコマンドの入力としてリダイレクトすることもできます。このような方法で複数のコマンドを連結したものをパイプラインと呼びます。パイプラインは、パイプと呼ばれる垂直バー (|) を使って構成します。)

たとえば、コマンドの出力結果をファイルに保存する代わりに、プリントコマンド (lp) の入力としてリダイレクトする場合、パイプ記号 (|) を使えます。date コマンドからの出力をプリンタに直接送信するには、次のように入力します。

```
$ date | lp
$
```

これによって、date コマンドの結果が印刷されます。(lp コマンドを使ったファイルの印刷方法については、第 8 章の 140 ページの「デフォルトプリンタへの印刷要求の実行」を参照してください。

ここで紹介したコマンドのリダイレクト例は非常に簡単なものですが、さらに高度なコマンドを学習するに従って、リダイレクトとパイプのさまざまな用途が理解できます。

バックグラウンドでコマンドを実行する

コマンドプロンプトからコマンドを始動した後で、そのコマンドをバックグラウンドに置くとは便利な場合がよくあります。コマンドをバックグラウンドに置かない場合は、そのコマンドが作業を完了するまでは次のプロンプトは表示されません。しかし、中には完了するまでにかなりの時間がかかるコマンドもあるため、その間に別のコマンドを実行する方が効率的です。

コマンドをバックグラウンドで実行する場合は、下記に示すようにコマンド名の後にアンパサンド (&) を入力します。その下に表示されている番号は、プロセス ID です。

```
$ bigjob &
[1] 21414
$
```

この例では、コマンド bigjob はバックグラウンドで実行され、他のコマンドの入力を続行できます。バックグラウンドのジョブが終了すると、次にコマンド (この例では date) を入力したときに下記のようなメッセージが表示されます。

```
$ date
Mon Feb 3 10:23:25 PST 1992
[1] + Done bigjob
$
```


バックグラウンドのジョブが完了する前にログアウトする可能性がある場合は、下記の例に示すように、`nohup` コマンド (`no hangup`) を使ってジョブを起動すると、ログアウトしてもそのジョブを完結させることができます。それ以外の場合は、ログアウトするとバックグラウンドのジョブは終了してしまいます。

```
$ nohup bigjob &  
[1] 21414  
$
```

OS コマンドによるヘルプ情報の表示

この節では、さまざまなオンラインヘルプ機能について説明します。オンラインヘルプ機能を使うことによって、ワークステーションや端末からリファレンス情報を読むことができます。

注 - この節では、OpenWindows のヘルプ機能以外のヘルプ機能について説明しています。

man コマンドによるマニュアルページの表示

コマンドの名前は知っていてもその機能について不明な場合は、`man` コマンドを使うと便利です。`man` コマンドの詳細機能を表示するには、次のように入力します。

```
$ man man
```

このコマンドは、SunOS マニュアルページ `man(1)` の最初の画面をウィンドウの表示領域に出力します。次の画面を表示するには `Space Bar` キーを押します。終了してコマンドプロンプトに戻るには `q` キーを押します。マニュアルページは、利用できる全オプションと適切なコマンド構文を記述しています。マニュアルページによっては、コマンドのさまざまな用途を説明した例もあります。

whatis コマンドによる 1 行要約の表示

コマンド機能の 1 行要約だけを参照する場合は、次に示すような `whatis` コマンドを使います。

```
$ whatis date
date (1)      -display or set the date
$
```

上記の例で、コマンド名の後の括弧内に数字がありますが、この数字はこのコマンドが属するセクションを示します。コマンドは、その機能に応じてさまざまなセクションに分類されています。大部分のユーザコマンドは、セクション 1 にあります。共通の表記法により、セクション番号はコマンド名の後の括弧内に表示されます。紙のマニュアルページでは、同じセクション内の各コマンドはアルファベット順に並んでいます。

apropos によるキーワードの検索

実行したい機能はわかっても、どのコマンドを使えばよいかわからない場合は、`apropos` コマンドが便利です。`apropos` コマンドを使うと、キーワード検索によって適切なコマンドを見つけることができます。`apropos` を実行すると、入力したキーワードを要約行に含むコマンドが、すべて一覧表示されます。入力したキーワードが多数の場所に現れる場合は、表示が非常に長くなる場合があります。

`apropos` の出力例として、次のコマンドのどれかを入力してみてください。

- `apropos who`
- `apropos execute`
- `apropos apropos`

入力するキーワードによって表示が非常に長くなる場合は、**Ctrl-C** を押せば (Ctrl キーを押したままで c キーを押す)、表示が中止されコマンドプロンプトに戻ります。

ファイルとディレクトリの操作

ファイルとディレクトリに関する操作は、SunOS のコマンド行から実行できます。特定の操作を実行するには、SunOS コマンドとファイルやディレクトリの名前を一緒に指定します。この方法は OpenWindows のファイルマネージャを使う場合とは異なります。ファイルマネージャでは、ファイルはマウスのクリックや移動で操作できるアイコンとして表示され、コマンドはメニューから選択できます。

この章では、SunOS のコマンド行からファイルとディレクトリを操作するために必要な手順と概念について説明します。OpenWindows 環境でシェルツールまたはコマンドツールを使う場合やリモート端末からログインする場合でも、SunOS のコマンド行に対してこれらの操作を適用できます。SunOS オペレーティングシステムを活用するためにも、この章で説明する概念を理解しておいてください。

ファイルの概念

ファイルは、SunOS オペレーティングシステムの基本的な単位です。次のものがファイルとして取り扱われます。

- ドキュメント — これには、手紙やレポートのようなテキストファイル、コンピュータのソースコードなど、ユーザが書き込みや保存を行うものすべてが含まれます。
- コマンド — ほとんどのコマンドは実行可能ファイル (特定のプログラムを起動するために実行されるファイル) です。たとえば、第 2 章で説明した `date` コマンドは、現在の日付を出力するプログラムを起動するための実行可能ファイルです。

- デバイス — 端末、プリンタ、ディスクドライブなどは、すべてファイルとして取り扱われます。
- ディレクトリ — ディレクトリは、他のファイルを格納するためのファイルです。

以下の節では、ファイルの作成、一覧表示、コピー、移動、削除に利用できるコマンドを説明します。また、ファイルの内容を表示する方法やファイルの特性を決定する方法についても説明します。

ファイル操作コマンドの使い方

この節で示す各コマンドについては、その使用例が提供されています。本文を読みながら、これらの例を試してください。実際に実行することによって、コマンドとその個々の概念をよく理解でき、簡単に覚えることができます。

ファイル操作を始める前に

ファイル操作を行う前に、ホームディレクトリにいることを確認してください。ホームディレクトリとは、アカウントの作成時にシステム管理者によって各ユーザに割り当てられたディレクトリです。以下の例で示す操作を自分のホームディレクトリで実行していれば、システムのうち他のユーザに影響を及ぼすような領域内でファイルを作成、コピー、移動、(最悪の場合は) 削除する危険性は少なくなります。

`cd` (change directory) コマンドを引数なしで入力することによって、各ユーザは自分のホームディレクトリに移動できます。次に `pwd` (print working directory) コマンドを入力して、ファイルシステム内での現在位置を表示します。このとき表示されるディレクトリが各ユーザのホームディレクトリです。

```
$ cd
$ pwd
/export/home/username
```

上記の例では、ユーザのホームディレクトリは `/export/home/username` です。`username` は、このホームディレクトリを所有しているユーザの名前です。

テスト用ファイルの作成

空のファイルを作成するには、touch コマンドを使います。指定した名前のファイルが存在しない場合は、touch コマンドが空のファイルを作成します。ファイルが存在する場合は、ファイルの最新アクセス時間が更新されます。

```
$ touch tempfile
$
```

ファイルの一覧表示 (ls)

ファイルが作成されたことを確認するために、ls コマンドを使ってファイル名を表示します。

```
$ ls tempfile
tempfile
```

ls コマンドを引数なしで入力すると、カレントディレクトリ内のファイルの一覧が表示されます。特定のファイル名を指定して ls コマンドを入力すると、そのファイルが存在する場合は指定されたファイルだけが表示されます。

ls コマンドについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

ファイルのコピー (cp)

tempfile を copyfile というファイルにコピーするには、cp コマンドを使います。

```
$ cp tempfile copyfile
$
```

ここで両方のファイルの一覧を表示してみましょう。どちらのファイル名も「file」という文字列で終わることに注目してください。任意の1文字または文字列を表すには、ワイルドカード文字のアスタリスク (*) が使えます。したがって、ls *file というコマンドを入力すると、tempfile と copyfile の両方が表示されます。この場合、カレントディレクトリ内に他に file で終わる名前のファイルがあれば、そのファイルも表示されます。

```
$ ls *file
copyfile  tempfile
```

copyfile が最初に表示されることに注意してください。ファイルはアルファベット順に表示されます。ただし、大文字と数字は小文字より先に表示されます。

cp コマンドについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

ファイルの移動とファイル名の変更 (mv)

mv (move) コマンドを使って、ファイルの移動とファイル名の変更が行えます。次の例では、mv コマンドを使ってファイル名を tempfile から emptyfile に変更しています。

```
$ mv tempfile emptyfile
$
```

上記の変更を確認するために、両方のファイルのリストを再表示します。

```
$ ls *file
copyfile  emptyfile
```

このリストから分かるように、tempfile が emptyfile で置き換えられています。

mv コマンドについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

ファイルの削除 (rm)

最後に、rm (remove) コマンドを使って copyfile を削除し、ls コマンドでその結果を確認します。

```
$ rm copyfile
$ ls *file
emptyfile
```



注意 - `rm` コマンドは注意して使ってください。ファイルは一度削除すると永久に消えてしまい、バックアップコピーがない限り復元できません。特にワイルドカード文字 (*) と `rm` を共に使う場合は細心の注意を払ってください。

`rm` コマンドについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

ファイル内容の表示 (`more`、`cat`)

ファイルの内容を表示するには、`more` コマンドを使います。`more` の後に、表示するファイル名を指定します。ファイルの内容は、画面に表示されます。ファイルが 1 画面より長い場合は、次のメッセージが表示されます。

```
--More-- (nn%) [Press space to continue, 'q' to quit.]
```

`nn` はすでに表示されたファイルの割合です。

ファイル内容の表示には `cat` (`concatenate`) コマンドも使えますが、この場合はファイル全体が休止することなく一度に表示されます。`cat` コマンドは、ファイルの表示よりも 2 つ以上のファイルを 1 つの大きいファイルに連結する場合によく使われます。ファイルの連結例を以下に示します。

```
$ cat file1 file2 file3 > bigfile
$ ls *file
bigfile
file1
file2
file3
$
```

`more` コマンドと `cat` コマンドについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

ファイル形式の表示 (`file`)

ファイルによっては、バイナリファイルや実行可能ファイルのように、プリンタに印刷したり画面上に表示したりできないものがあります。ファイル形式が不明な場合は、`file` コマンドを使うと便利です。

file コマンドは、ファイル形式の表示に使用します。

```
$ file copyfile
copyfile:  ascii text
```

ディレクトリとディレクトリ階層

前節までに、ファイルの作成、一覧表示、コピー、削除、ファイル名の変更などの方法を説明しました。しかし、これらのファイルがどこに存在するのかという大きな問題が残っています。この節では、ディレクトリ階層について説明します。以下の解説をよく読んで、各節で示す例を実行してみてください。

ディレクトリ階層

ファイルはディレクトリにまとめられますが、ディレクトリ自体も階層構造になっています。この階層の最上位は「ルート」ディレクトリと呼ばれ、「/」で表されます。

次の図 3-1 の例で示すように、ファイルシステム内の各ディレクトリは、その中に多数のサブディレクトリを持つことができます。表記上は、ディレクトリのレベルを「/」文字で識別します。この点を考慮して、/ (ルート) ディレクトリには、/usr、/bin、/home、/lib というサブディレクトリがあることに注意してください。さらに /home サブディレクトリには、user1、user2、user3 というサブディレクトリがあります。

ディレクトリ (およびその中のファイル) を指定する場合、それが存在するディレクトリの名前も含めます。これをパス名と呼びます。たとえば、図 3-1 の user3 ディレクトリのパス名は、/home/user3 です。

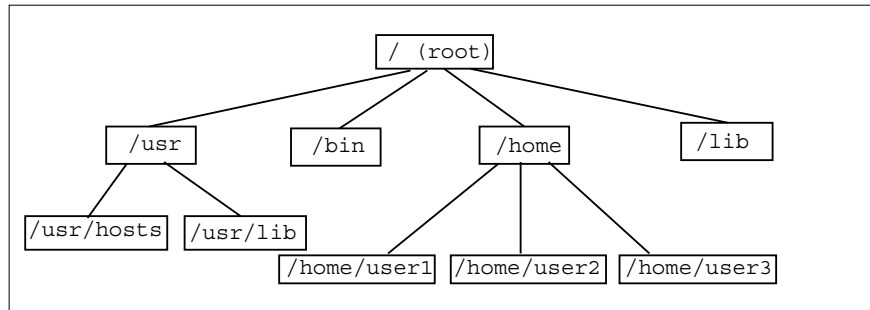


図 3-1 ファイルシステム階層

同一ディレクトリ内のすべてのサブディレクトリとファイルの名前は一意でなければなりません。ディレクトリが異なればこれらの名前が同じであってもかまいません。たとえば、`/usr` ディレクトリには `/usr/lib` というサブディレクトリがあります。`/usr/lib` と `/lib` とではパス名が異なるため、これらの名前は互いに矛盾しません。

ファイルのパス名の機能は、ディレクトリのパス名とまったく同じです。ファイルのパス名は、ファイルシステム階層におけるそのファイルの位置を表します。たとえば、`/home/user2` ディレクトリに `report5` というファイルがあるとすると、そのファイルのパス名は `/home/user2/report5` となります。このパス名は、ファイル `report5` はディレクトリ `user2` の中にあり、`user2` はディレクトリ `home` の中にあり、`home` はルート (`/`) ディレクトリの中にあることを示しています。

ディレクトリの中には、サブディレクトリだけ、ファイルだけ、またはその両方を格納できます。

作業用ディレクトリの表示 (`pwd`)

`pwd` (`print working directory`) コマンドは、ファイルシステム階層内でのユーザの現在位置を表示します。

```

$ pwd
/home/user1
  
```

実際のディレクトリ構造が異なる場合は、`pwd` の出力結果が上記の例とは異なるものになります。作業用ディレクトリがファイルシステム階層内でのユーザの現在位置になることに注意してください。

ユーザのホームディレクトリ

各ユーザはホームディレクトリを持っています。OpenWindows 環境でシェルツールやコマンドツールのウィンドウを開いたときは、最初の位置 (作業用ディレクトリ) がホームディレクトリになります。ホームディレクトリは、アカウントの作成時にシステム管理者によって各ユーザに割り当てられます。

作業用ディレクトリの変更 (cd)

cd (change directory) コマンドを使うと、ファイルシステム階層内を自由に移動できます。

```
$ cd /usr/lib
$ pwd
/usr/lib
```

cd コマンドを引数なしで入力すると、自分のホームディレクトリに戻ります。たとえば、ホームディレクトリが /home/user1 の場合は、次のようになります。

```
$ cd
$ pwd
/home/user1
```

C シェルでは、ホームディレクトリを表す短縮名としてチルド文字 (~) が使われます。たとえば、次のように入力するとホームディレクトリ内の music サブディレクトリに移動できます。

```
example% cd ~/music
```

この短縮名を使って他のユーザのホームディレクトリを指定することもできます。次の例を参照してください。

```
example% cd ~username
```

username は他のユーザのログイン名です。このコマンドによって、そのユーザのホームディレクトリに移動します。

注 - Bourne シェルを使っている場合は、この短縮名 `~` は機能しません。

Bourne シェルを使っているときは、`$home` と入力するとホームディレクトリに移動できるようにシステム管理者がシステムを構成している場合があります。その場合は、次のように入力すると、ホームディレクトリ内の `music` サブディレクトリに移動できます。

```
$ $home/music
```

同様に、次のように入力すると、指定したユーザのホームディレクトリに移動できます。

```
$ $homeusername
```

`username` は他のユーザのログイン名です。

サブディレクトリのすぐ「上」のディレクトリは、親ディレクトリと呼ばれます。前記の例では、`/home` が `/home/user1` の親ディレクトリです。親ディレクトリは、シンボル `..` (ピリオドを2つ) で表します。したがって、`cd ..` というコマンドを入力すると、次の例で示すように、作業用ディレクトリが親ディレクトリに変わります。

```
$ pwd
/home/user1
$ cd ..
$ pwd
/home
```

現在の作業用ディレクトリが `/home/user1` のときに `/home/user2` にあるファイルを操作したい場合は、次のような短縮名を利用すると便利です。

```
$ pwd
/home/user1
$ cd ../user2
$ pwd
/home/user2
```

`../user2` は、`user2` の親ディレクトリを経由するようシステムに指示します。この例に示すように、フルパス名 `/home/user2` を入力するより簡単です。

ディレクトリの作成 (mkdir)

mkdir コマンドの後に新しいディレクトリの名前を指定するだけで、新規ディレクトリを簡単に作成できます。

```
$ mkdir veggies
$ cd veggies
$ mkdir broccoli
$ cd broccoli
$ pwd
/home/user2/veggies/broccoli
```

相対パス名

ファイルやディレクトリのフルパス名はスラッシュ (/) で始まり、そのファイルまたはディレクトリとルートディレクトリとの間のディレクトリ構造全体を表します。ただし、現在の作業用ディレクトリに対するファイルやディレクトリの相対関係を表す省略名 (相対パス名) を使うこともできます。

親ディレクトリにいる場合は、フルパス名ではなくディレクトリ名だけを指定すればサブディレクトリに移動できます。前記の例では、`cd veggies` というコマンドで `veggies` ディレクトリの相対パス名が使われています。現在の作業用ディレクトリが `/home/user2` の場合、このディレクトリのフルパス名は `/home/user2/veggies` となります。

別個のサブディレクトリをいくつか作成し、そのディレクトリ構造の中を移動してみてください。移動の際にはフルパス名と相対パス名の両方を使ってみてください。現在位置は `pwd` コマンドで確認します。

ディレクトリの移動とディレクトリ名の変更

ディレクトリを別の名前に置き換えることによって、ディレクトリ名を変更できます。ディレクトリ名の変更には `mv` コマンドを使います。

```
$ pwd
/home/user2/veggies
$ ls
broccoli
$ mv broccoli carrots
$ ls
carrots
```

mv コマンドを使って、ディレクトリをすでにある別のディレクトリ内に移動することもできます。

```
$ pwd
/home/user2/veggies
$ ls
carrots
$ mv carrots ../veggies2
$ ls ../veggies2
carrots
```

この例では、mv コマンドを使って、carrots ディレクトリを veggies から veggies2 に移動しています。

ディレクトリのコピー

ディレクトリとその下にあるファイルをコピーする場合は、cp -r コマンドを使います。

```
$ cp -r veggies veggies3
$
```

このコマンドは、veggies ディレクトリに含まれるすべてのファイルとサブディレクトリを、新しいディレクトリ veggies3 にコピーします。これは「再帰的コピー」と呼ばれ、-r (recursive) オプションで指定します。このオプションを指定しないでディレクトリをコピーしようとすると、エラーメッセージが出されます。

ディレクトリの削除 (rmdir)

空のディレクトリを削除する場合は、rmdir コマンドを使います。

```
$ rmdir veggies3
$
```

ディレクトリにファイルやサブディレクトリが含まれている場合、rmdir コマンドではディレクトリは削除できません。

ディレクトリとその内容 (すべてのサブディレクトリやその中のファイルも含む) を削除する場合は、次に示すように rm -r (再帰的削除を指示する -r オプションを rm コマンドに付加したもの) を使います。

```
$ rm -r veggies3
$
```



注意 - rmdir コマンドで削除したディレクトリは復元できません。rm -r コマンドで削除したディレクトリとその内容についても同様に復元できません。

ファイル間の相違点を検出する (diff)

各ユーザが別々に同一のファイルをコピーし、そのコピーを編集するという場合があります。その場合 diff コマンドを使うと、バージョンの異なるファイルについて相違点を検出することができます。次のコマンドは、leftfile と rightfile の各行を走査して相違点を表示します。

```
$ diff leftfile rightfile
```

異なる行が見つかり、diff はその相違の原因 (行の追加、削除、変更) と影響を受けた行数を判断し、各ファイル内の個々の相違箇所の行番号と該当テキストを表示します。

相違の原因が行の追加による場合、diff は次の形式で行番号を表示します。

```
l[,l] a r[,r]
```

l は leftfile の行番号、*r* は rightfile の行番号です。

相違の原因が行の削除による場合は、上記の *a* の代わりに *d* が表示され、行の変更による場合は *c* が表示されます。

各ファイル内の該当行は、行番号情報の次の行に表示されます。leftfile 内のテキストの前には左角括弧 (<) が付き、rightfile 内のテキストの前には右角括弧 (>) が付きます。

次の例では、最初に 2 つのファイルの内容を表示し、その後で diff コマンドを使ってそれらの相違点を表示しています。

```
$ cat sched.7.15
Week of 7/15

Day: Time:      Action Item:      Details:
T   10:00      Hardware mtg.      every other week
W   1:30       Software mtg.
T   3:00       Docs. mtg.
F   1:00       Interview
$ cat sched.7.22
Week of 7/22

Day: Time:      Action Item:      Details:
M   8:30       Staff mtg.         all day
T   10:00      Hardware mtg.      every other week
W   1:30       Software mtg.
T   3:00       Docs. mtg.
$ diff sched.7.15 sched.7.22
1c1
< Week of 7/15
---
> Week of 7/22
4a5
> M   8:30      Staff mtg.         all day
8d8
< F   1:00      Interview
```

比較される 2 つのファイルの内容が同じであれば、diff コマンドを実行しても、何も出力されません。

diff コマンドには、この節で説明した以外にも多数のオプションがあります。詳細については、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

3 つのファイルを比較する (diff3)

あるファイルの 3 つのバージョンを同時に比較する場合は、次に示す diff3 コマンドを使います。

```
$ diff3 file1 file2 file3
```

diff3 コマンドはファイルの 3 つのバージョンを比較し、内容が異なる範囲に以下のコードを付加して表示します。

```
====      3 つのファイルがすべて異なる
====1     file1 が他の 2 つと異なる
====2     file2 が他の 2 つと異なる
```

====3 *file3* が他の 2 つと異なる

大きいファイルを `bdiff` を使って比較する

非常に大きいファイルを比較する場合は、`diff` の代わりに `bdiff` コマンドを使います。`bdiff` の使い方と出力内容は、`diff` と同じです。

```
$ bdiff leftfile rightfile
```

3500 行以上のファイルでは、`diff` ではなく `bdiff` を使ってください。

ファイル名の検索 (`find`)

`find` コマンドを使うと、指定した条件に合致するファイル名の検索を、特定のディレクトリから開始できます。たとえば、特定のパターンに一致するファイル名や一定の期間内に変更されたファイル名などを検索できます。

他の大部分のコマンドとは違って、`find` のオプションはいくつかの文字からなり、コマンド行ではオプションより前に、検索を開始するディレクトリ名を指定する必要があります。

```
$ find directory options
```

`directory` は検索を開始するディレクトリ名、`options` は `find` コマンドのオプションです。

各オプションは、ファイルの選択基準を指定します。選択されるファイルは、すべての基準を満たしている必要があります。したがって、指定するオプションの数が多くなるほど、選択されるファイルの範囲は狭くなります。`-print` オプションを指定すると、検索結果が表示されます。

`find` コマンドで `-name filename` オプションを指定すると、`filename` に一致するファイルだけが選択されます。この `filename` は、フルパス名のファイルの一番右側の要素と見なされます。たとえば、`/usr/lib/calendar` の一番右側の要素は `calendar` です。通常、フルパス名中のこの要素は、ファイルのベース名と呼ばれます。

たとえば、カレントディレクトリとそのサブディレクトリ内で `s` で終わるファイルを検索する場合は、次のように入力します。

```
$ find . -name '*s' -print
./programs
./programs/graphics
./programs/graphics/gks
./src/gks
$
```

`find` コマンドでは次のオプションを使えます。

■ `-name filename`

フルパス名の一番右側の要素 (ベース名) が `filename` に一致するファイルを選択します。`filename` にファイル名の置換パターン (正規表現) が含まれる場合は、`filename` を単一引用符 (') で囲みます。

■ `-user userid`

`userid` が所有するファイルを選択します。`userid` はログイン名、ユーザ ID 番号のどちらでもかまいません。

■ `-group group`

`group` に属するファイルを選択します。

■ `-m -time n`

`n` 日以内に更新されたファイルを選択します。

■ `-newer checkfile`

`checkfile` より後に更新されたファイルを選択します。

`\(options\)` のように、エスケープされた括弧でオプションを囲むことによって、検索条件の優先度を指定できます。エスケープされた括弧内では、各オプションの間に `-o` フラグを挿入して、両方のカテゴリではなく、どちらか一方のカテゴリに分類されたファイルを選択するよう `find` に指定できます。

```
$ find . \( -name AAA -o -name BBB \) -print
./AAA
./BBB
```

オプションの前にエスケープされた感嘆符を付けることによって、オプションの意味を逆にできます。この場合、find はオプションで指定された条件に合致しないファイルを選択します。

```
$ find . \!-name BBB -print
./AAA
```

また次のオプションを使うと、find で検索されたファイルに対して特定のコマンドを実行できます。

```
-exec command '{}' \;
```

このオプションは、エスケープされたセミコロン (\;) で終了します。引用符付きの大括弧 ('{}') は、find によって選択されたファイル名で置き換えられます。

たとえば、find コマンドを利用すれば一時的な作業用ファイルを自動的に削除できます。一定の規則に従って一時ファイル名を指定しておくことにより、その存在場所にかかわらず、find を使ってそれらの一時ファイルを検索した上で破棄できます。たとえば、一時ファイル名を junk または dummy と指定した場合は、次のコマンドによってそれらの検索と削除を実行できます。

```
$ find . \( -name junk -o -name dummy \) -exec rm '{}' \;
```

find コマンドについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

ファイルとディレクトリのセキュリティ

注 - この節は注意してよく読んでください。日常のファイル操作では、ファイルのアクセス権を正確に理解しておく必要があります。

ファイルのアクセス権は、ファイルとディレクトリを許可なく読み書きされないように保護します。他のユーザに対してファイルの読み取りは許可しても変更は許可したくない場合があります。また、実行可能ファイル (プログラム) を共有したい場合もあります。このような場合、ファイルのアクセス権を利用して、ファイルに対するアクセスを管理できます。

ファイルとディレクトリに関する基本的なアクセス権の種類は次のとおりです。

- **r** – 読み取り権。ファイルの内容を参照したりコピーしたりするには、そのファイルが読み取り可能でなければなりません。また、ディレクトリの内容を表示するには、そのディレクトリが読み取り可能でなければなりません。
- **w** – 書き込み権。ファイルを変更、削除したり、ファイル名を変更するには、そのファイルが書き込み可能でなければなりません。また、ディレクトリ内のファイルを追加したり削除したりするには、そのディレクトリが書き込み可能でなければなりません。
- **x** – 実行権。実行権付きのファイル (プログラムなど) は実行することができます。ユーザがディレクトリの任意のサブディレクトリにアクセスするには、そのディレクトリが実行可能でなければなりません。

アクセス権の設定対象となるユーザは、次の3つに分類できます。

- 所有者 – ユーザ自身
- グループ – 所有者と同じグループに属するユーザ (例: アカウントを持つすべてのユーザ)。グループの作成と管理は、システム管理者が行います。
- 他のユーザ – 上記以外の全ユーザ

アクセス権と状態の表示 (su ls -l)

ls コマンドを使ってファイルを一覧表示してみます。ls コマンドには多数のオプションがあります。詳細形式の一覧を表示するには、-l オプションを使います。ファイルとディレクトリはアルファベット順に表示されます。このオプションを使って表示したファイル一覧の例を図 3-2 に示します。

```

$ pwd
/home/hostname/user2
$ ls -l
total 8
drwxr-xr-x  2 user2          1024 Feb  9 14:22 directory1
-rw-r--r--  1 user2              0 Feb 10 10:20 emptyfile
-rw-r--r--  1 user2       104357 Feb  5 08:20 large-file
drwxr-xr-x  3 user2          1024 Feb 10 11:13 veggies2

```

アクセス権 リンク 所有者 サイズ 日付 時間 ファイルやディレクトリの名前

図 3-2 アクセス権とステータスの表示

各行の最初の文字は、ファイル形式を示します。ダッシュ (-) は通常ファイル、d はディレクトリ、その他の文字は特殊なファイル形式を表します。

次の9文字は、ファイルやディレクトリのアクセス権を示します。これら9文字は、3文字ずつの3つのグループから構成され、それぞれ所有者、所有者のグループ、その他の全ユーザのアクセス権を表します。たとえば、`emptyfile`のアクセス権は、`rw-r--r--`となっています。このファイルの場合、所有者は読み取りと書き込みが行え、すべてのユーザは読み取りだけは行えますが、どのユーザも実行はできません。また、ディレクトリ `veggies2` のアクセス権は、`rwxr-xr-x` となっています。このディレクトリの場合、どのユーザでも読み取りと実行が行えますが、書き込みが行えるのは所有者に限られます。

ファイルのアクセス権の他に、上記のリストには次の情報が含まれています。

- ファイルやディレクトリに対するリンクの数
- 所有者の名前(この例では `user2`)
- ファイルのバイト数(文字数)
- ファイルやディレクトリの最新更新日付と時間
- ファイルやディレクトリの名前

`cd` コマンドを使って自分のホームディレクトリに移動し、`ls -l` コマンドを入力してみてください。表示される結果は上記の例とは当然異なります。

次に、以下のようなコマンドを入力してみます。

```
$ ls -l dirname
```

dirname は、ユーザのファイルシステム内の実際のディレクトリ名です。ディレクトリ名を指定すると、`ls -l` コマンドは、そのディレクトリ内にあるディレクトリ(ただし、存在する場合だけ)とファイルに関する情報を出力します。

隠しファイルの表示 (`ls -a`)

通常の `ls` コマンドでは表示されないファイルもあります。これらのファイルは、`.cshrc`、`.login`、`.profile` のように `.` (ドット) で始まる名前を持っています。このようなドットファイルを表示するには、`ls -a` コマンドを使います。

```
$ ls -a
.
..
.cshrc
.login
.profile
.emptyfile
```

. で始まるファイルは、他のファイルよりも前に表示されることに注意してください。このリストには、2つの特殊ファイルが表示されています。ファイル . はカレントディレクトリを示し、ファイル .. は親ディレクトリを示します。

一般に、. で始まるファイルはシステムユーティリティによって使われ、通常はユーザが変更するものではありません。ただし、少数ですが例外もあります。

アクセス権の変更 (chmod)

ファイルやディレクトリのアクセス権を変更する場合は、chmod コマンドを使います。アクセス権を変更するには、そのファイルやディレクトリの所有者であるか、root にアクセスできる必要があります。chmod コマンドの一般的な形式は次のとおりです。

```
chmod permissions name
```

permissions は変更用のアクセス権、*name* は影響を受けるファイルやディレクトリの名前を示します。

アクセス権を指定するには、いくつかの方法があります。ここでは最も簡単な方法を紹介します。

1. 次の文字のどれかを使って関連するユーザを指定します。
 - u (所有者)
 - g (グループ)
 - o (他のユーザ)
 - a (上記3つのカテゴリの全ユーザ)
2. アクセス権を付加するか (+) 除去するか (-) を指定します。
3. 次のいずれかの文字を使って関連するアクセス権を指定します。
 - r (読み取り)

- w (書き込み)
- x (実行)

次の例では、carrots ディレクトリに対する書き込み権を、同じグループに属するユーザに付与しています (つまり *permissions* は g+w、*name* は carrots)。

```
$ ls -l carrots
drwxr-xr-x 3 user2      1024 Feb 10 11:15 carrots
$ chmod g+w carrots
$ ls -l carrots
drwxrwxr-x 3 user2      1024 Feb 10 11:15 carrots
$
```

例で示すように、グループのアクセス権のハイフン (-) が chmod コマンドの実行後は w に変更されています。

この同じディレクトリに対する読み取り権と実行権を、グループに属さない他のユーザから除去するには (*permissions* は o-rx)、次のように入力します。

```
$ ls -l carrots
drwxrwxr-x 3 user2      1024 Feb 10 11:15 carrots
$ chmod o-rx carrots
$ ls -l carrots
drwxrwx--- 3 user2      1024 Feb 10 11:15 carrots
$
```

他のユーザのアクセス権の r (読み取り権) と x (実行権) が、両方ともハイフン (-) に変更されたことが分かります。

新しいファイルやディレクトリを作成すると、アクセス権がシステムによって自動的に割り当てられます。

通常、新規ファイルに関するデフォルト設定は次のようになります。

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

また、新規ディレクトリについては次のようになります。

```
drwxr-xr-x
```

したがって、turnip という新規ファイルをその所有者 (user2) が実行できるようにする場合は、次のように入力します。

```
$ ls -l turnip
-rw-r--r-- 3 user2      1024 Feb 10 12:27 turnip
$ chmod u+x turnip
$ ls -l turnip
-rwxr--r-- 3 user2      1024 Feb 10 12:27 turnip
$
```

3つのカテゴリの全ユーザのアクセス権を変更する場合は、`-a` オプションを使います。たとえば、`garlic` という新規ファイルを全ユーザが実行できるようにするには、次のように入力します。

```
$ ls -l garlic
-rw-r--r-- 3 user2      1024 Feb 10 11:31 garlic
$ chmod a+x garlic
$ ls -l garlic
-rwxr-xr-x 3 user2      1024 Feb 10 11:31 garlic
$
```

上記の `chmod` コマンドの結果、3つのカテゴリの全ユーザに `x` (実行権) が与えられます。

ワイルドカード文字の `*` を使って、複数のファイルやディレクトリのアクセス権を変更することもできます。たとえば、カレントディレクトリ `veggies` 内の全ファイルのアクセス権を変更して、所有者だけがファイルの書き込みをできるようにする場合は、次のように入力します。

```
$ pwd
/home/user2/veggies
$ ls -l
-rwxrwxrwx 3 user2      21032 Feb 12 10:31 beats
-rwxrwxrwx 2 user2         68 Feb 10 11:09 corn
-rwxrwxrwx 3 user2     12675 Feb 08 09:31 garlic
-rwxrwxrwx 1 user2      1024 Feb 14 16:38 onions
$ chmod go-w *
$ ls -l
-rwxr-xr-x 3 user2      21032 Feb 12 10:31 beats
-rwxr-xr-x 2 user2         68 Feb 10 11:09 corn
-rwxr-xr-x 3 user2     12675 Feb 08 09:31 garlic
-rwxr-xr-x 1 user2      1024 Feb 14 16:38 onions
$
```

上記の例では、この `chmod` 操作をこれからどのディレクトリで実行するのか確認する必要があることを示すために、`pwd` コマンドを最初に実行しています。

絶対アクセス権の設定

アクセス権に関する前節までの説明では、`chmod` コマンドを使って現在のアクセス権の設定を相対的に変更する方法だけを取り扱ってきました。`chmod` コマンドの別の方法を使うことによって、ファイルやディレクトリの絶対的なアクセス権を設定できます。この方法では、アクセス権を指定する数値コードを使います。

この形式の `chmod` コマンドの構文は次のとおりです。

```
chmod numcode name
```

`numcode` は数値コード、`name` はアクセス権が変更されるファイルやディレクトリの名前です。

数値コードは、3つの数字からなります。1つの数字が3つのユーザカテゴリ(所有者、グループ、他のユーザ)の各々に適用されます。たとえば、次のコマンドは、所有者とグループに対して読み取り権、書き込み権、実行権の絶対アクセス権を設定し、他のユーザに対して実行権だけを設定しています。

```
$ chmod 771 garlic
```

表 3-1 は、`garlic` のアクセス権が 771 というコードによって、どのように表現されるかを示しています。

表 3-1 `garlic` のアクセス権

アクセス権	所有者	グループ	他のユーザ
読み取り権	4	4	0
書き込み権	2	2	0
実行権	1	1	1
合計	7	7	1

表 3-1 の各列は、所有者、グループ、他のユーザというユーザカテゴリにそれぞれ対応しています。読み取り権を設定するには、適切な列に 4 を挿入します。同様に、書き込み権を設定するには 2 を挿入し、実行権を設定するには 1 を挿入します。表の最終行に表示された各列のそれぞれの合計値が最終的な数値コードになります。

この方法によって絶対アクセス権を設定するもう1つの例を次に示します。ここでは `chmod` の結果を確認するために、`ls -l` コマンドを実行しています。

```
$ ls -l onion
-rw-r--r-- 3 user2      1024 Feb 10 11:46 onion
$ chmod 755 onion
$ ls -l onion
-rwxr-xr-x 3 user2      1024 Feb 10 11:48 onion
$
```

ファイル `onion` のアクセス権は、所有者に対しては読み取り、書き込み、実行が可能、グループと他のユーザに対しては読み取りと実行が可能になるよう設定されています。`onion` のアクセス権の設定に使われた数値コードを表 3-2 にまとめます。

表 3-2 `onion` のアクセス権

アクセス権	所有者	グループ	他のユーザ
読み取り権	4	4	4
書き込み権	2	0	0
実行権	1	1	1
合計	7	5	5

また、ファイル `cabbage` に関する読み取り、書き込み、実行の全アクセス権を所有者、グループ、他のユーザに与える場合は、次のように入力できます。

```
$ ls -l cabbage
-rw-r--r-- 3 user2      1024 Feb 10 11:51 cabbage
$ chmod 777 cabbage
$ ls -l cabbage
-rwxrwxrwx 3 user2      1024 Feb 10 11:53 cabbage
$
```

この例の数値コードを表 3-3 にまとめます。

表 3-3 cabbage のアクセス権

アクセス権	所有者	グループ	他のユーザ
読み取り権	4	4	4
書き込み権	2	2	2
実行権	1	1	1
合計	7	7	7

777 という数値コードが、指定できる最大のアクセス権の値です。

相対アクセス権の変更の場合と同様に、ワイルドカード文字の * を使ってカレントディレクトリ内の全ファイルの絶対アクセス権を設定することもできます。たとえば、カレントディレクトリ `veggies` 内の全ファイルに絶対アクセス権を設定することで、所有者は読み取り権と書き込み権と実行権、グループは読み取り権と実行権、他のユーザは実行権だけを持つようにするには、次のように入力します。

```

$ pwd
/home/user2/veggies
$ ls -l
-rwxrwxrwx 3 user2      21032 Feb 12 10:31 beats
-rwxrwxrwx 2 user2         68 Feb 10 11:09 corn
-rwxrwxrwx 3 user2     12675 Feb 08 09:31 garlic
-rwxrwxrwx 1 user2      1024 Feb 14 16:38 onions
$ chmod 751 *
$ ls -l
-rwxr-x--x 3 user2      21032 Feb 12 10:31 beats
-rwxr-x--x 2 user2         68 Feb 10 11:09 corn
-rwxr-x--x 3 user2     12675 Feb 08 09:31 garlic
-rwxr-x--x 1 user2      1024 Feb 14 16:38 onions
$

```

上記の例では、この `chmod` 操作をこれからどのディレクトリで実行するのか確認する必要があることを示すために、`pwd` コマンドを最初に行っています。最初の `ls -l` コマンドは、アクセス権の変更結果を明らかにするために実行しています。相対アクセス権とは違って、絶対アクセス権を設定するときは、アクセス権の現在の設定状態を知る必要はありません。

`chmod` コマンドについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

ファイルの内容の検索

この章では、SunOS の `grep` コマンドを使って、ディレクトリとファイルの内容を検索し、キーワードや文字列を見つける方法について説明します。

grep によるパターンの検索

ファイル内の特定の文字列を検索するには、`grep` コマンドを使います。`grep` コマンドの基本構文は次のとおりです。

```
$ grep string file
```

string は検索するワードや句、*file* は検索されるファイルです。

注 - 文字列とは 1 文字以上の文字の配列です。ワードや文と同様に、文字が 1 つでも文字列とします。文字列には、空白文字、句読文字、制御文字 (表示できない文字) も入ります。

たとえば、Edgar Allan Poe の内線番号を検索する場合は、`grep` の後に名前の一部または全体 (以下の例では `Poe`)、その後に内線番号の情報を含んでいるファイル名 (以下の例では `extensions`) を指定します。

```
$ grep Poe extensions
Edgar Allan Poe    x72836
$
```

ただし、指定されたパターンに一致するのは1行だけとは限らないので、注意してください。

```
$ grep Allan extensions
David Allan        x76438
Edgar Allan Poe    x72836
$ grep Al extensions
Louisa May Alcott  x74236
David Allan        x76438
Edgar Allan Poe    x72836
$
```

grep では大文字と小文字が区別されます。したがって、大文字と小文字の区別に注意してパターンを指定しなければなりません。

```
$ grep allan extensions
$ grep Allan extensions
David Allan        x76438
Edgar Allan Poe    x72836
$
```

上記の最初の例では、小文字の a で始まるエントリがないために検索は失敗に終わっています。

フィルタとして grep を使う

grep は、「フィルタ」として他のコマンドと併用されることがよくあります。grep をフィルタとして使うと、他のコマンドの出力結果から不要な情報を除去できます。grep をフィルタとして使うには、他のコマンドの出力結果を grep を通じてパイプする必要があります。パイプの記号は | です。

次の例では、ファイル名が .ps で終わるファイルのうち、5月 (May) に作成されたファイル名とその詳細情報を表示させます。

```
$ ls -l *.ps | grep May
```

上記のコマンド行の最初の要素は、次の部分です。

```
ls -l *.ps
```

このコマンドだけを実行すると、次のファイルの一覧が出力されます。

```
$ ls -l *.ps
-rw-r--r-- 1 elvis      7228 Apr 22 15:07 change.ps
-rw-r--r-- 1 elvis      2356 May 22 12:56 clock.ps
-rw-r--r-- 1 elvis      1567 Jun 22 12:56 cmdtool.ps
-rw-r--r-- 1 elvis     10198 Jun 22 15:07 command.ps
-rw-r--r-- 1 elvis      5644 May 22 15:07 buttons.ps
$
```

2 番目の要素は、次の部分です。

```
| grep May
```

この部分は、上記のファイルの一覧を `grep` を通じてパイプし、`May` というパターンを検索して表示します。

```
$ ls -l *.ps | grep May
-rw-r--r-- 1 elvis      2356 May 22 12:56 clock.ps
-rw-r--r-- 1 elvis      5644 May 22 15:07 buttons.ps
$
```

複数ワード文字列の検索

複数のワードからなるパターンを `grep` で検索する場合は、単一引用符または二重引用符でそのパターンを囲みます。

```
$ grep ``Louisa May`` extensions
Louisa May Alcott    x74236
$
```

`grep` は複数のファイル内で文字列を検索できます。複数のファイル内でパターンに一致する文字列が見つかった場合は、ファイル名とパターンに一致する行をコロンで区切って出力します。

```
$ grep ar *
actors:Humphrey Bogart
alaska:Alaska is the largest state in the United States.
wilde:book. Books are well written or badly written.
$
```

特定の文字列を含まない行の検索

特定の文字列を含まない行をファイル内で検索するには、`grep` の `-v` オプションを使います。次の例は、あるディレクトリのすべてのファイル内で、文字 `e` を含まない行すべてを検索する方法を示しています。

```
$ ls
actors alaska hinterland tutors wilde
$ grep -v e *
actors:Mon Mar 14 10:00 PST 1936
wilde:That is all.
$
```

grep の詳細な使い方

`grep` コマンドは、正規表現を使ったパターンの検索にも利用できます。正規表現は、`grep` に対して特別な意味を持つ特殊文字に、数字や英字を組み合わせで作成します。メタキャラクターと呼ばれるこれらの特殊文字はシステムに対しても特別な意味を持つため、引用符で囲むかエスケープする必要があります。コマンドプロンプトに対して `grep` の正規表現を使う場合は、必ず引用符で囲むか、メタキャラクター (`& ! . * $? \` など) をバックスラッシュ (`\`) でエスケープします。

- キャレット (^) は行の先頭を表します。したがって、次のコマンドは、ファイル `list` 内の `b` で始まる行を検索します。

```
$ grep '^b' list
```

- ドル記号 (\$) は行の終わりを表します。次のコマンドは、`b` で終わる行をすべて表示します。

```
$ grep 'b$' list
```

また、次のコマンドは、ファイル `list` 内で `b` だけ (`b` が 1 文字だけで、その前後にタブや空白さえ存在しない) からなる行を検索します。

```
$ grep '^b$' list
```

- 正規表現内のドット(.)は任意の1文字を表します。したがって、次のコマンドは、最初の2文字がanの任意の3文字パターンを含む行を検索します。

```
$ grep 'an.' list
```

この3文字パターンには、「any」、「and」、「management」、「plan」(空白やタブも1文字と数えられるため、このplanはnの後に空白があります)などが一致します。

- ある文字のすぐ後にアスタリスク(*)が続いている場合、grepは*を「その直前の文字のゼロ個以上の繰り返し」と解釈します。正規表現の後にアスタリスクが続いている場合、grepは*を「正規表現のパターンに一致する文字のゼロ個以上の繰り返し」と解釈します。

「ゼロ個の繰り返し」という表現のため、その使い方が直観的には理解しにくいかも知れません。たとえば、quという文字列を含むワードをすべて検索する場合は、次のように入力すれば期待通りの結果が得られます。

```
$ grep 'qu*' list
```

しかし、文字nを含むワードをすべて検索する場合は、次のように入力しなければなりません。

```
$ grep 'nn*' list
```

また、nnというパターンを含むワードをすべて検索する場合は、次のように入力しなければなりません。

```
$ grep 'nnn*' list
```

このように指定しなかった場合の結果を確認する場合は、次の例を試してください。

- ファイルlist内で任意の文字のゼロ個以上の繰り返しを検索するには、次のように入力します。

```
$ grep '.*' list
```

メタキャラクタの検索

テキストファイル内でドル記号 (\$) を含む行を検索すると想定します。正規表現内のドル記号の前にバックスラッシュ (\) を付けると、\$ が持つ特殊な意味を無視 (エスケープ) するよう grep に指示できます。その他のメタキャラクタ (& ! . * ? および \ 自身など) についてもこの方法を適用できます。

たとえば、次の正規表現は、ピリオド (.) で始まる行と一致するため、nroff や troff の書式要求 (ピリオドで始まるもの) を検索するときに非常に便利です。

```
$ grep '^\. ' filename
```

次の表 4-1 は、grep で利用できる検索パターン文字のうち、頻繁に使われる文字のリストです。

表 4-1 grep の検索パターン文字

文字	意味
^	テキスト行の先頭
\$	テキスト行の終わり
.	任意の 1 文字
[...]	角括弧内のリストまたは範囲に含まれる任意の 1 文字
[^...]	角括弧内のリストまたは範囲に含まれない任意の 1 文字
*	その直前の文字または正規表現のゼロ個以上の繰り返し
.*	任意の 1 文字のゼロ個以上の繰り返し
\	その後の文字が持つ特殊な意味を無効にする

これらの検索パターン文字は、vi テキストエディタ内での検索にも使えます。

コマンド行で単一引用符と二重引用符を使う

すでに述べたように、`grep` に 1 つの単位として解釈させるテキストは引用符 (") で囲みます。たとえば、`grep` を使って「dang it, boys」という句を含むすべてのファイルを検索するには、次のように入力します。

```
$ grep "dang it, boys" *
```

複数のワードからなる句を 1 つの単位にまとめる場合は、単一引用符 (') も使えます。単一引用符を使うと、`$` などの特殊文字を単なる文字として解釈させることもできます。(history コマンドのメタキャラクタ ! は、バックスラッシュでエスケープしないと、引用符の中にあっても常に特殊文字として解釈されます。) どのような場合でも、`&` `!` `$` `?` `.` `;` `\` などの特殊文字を通常の印字文字として解釈させるときは、それらをバックスラッシュでエスケープすることをお勧めします。

たとえば、次のように入力すると、`list` 内のすべての行が表示されます。

```
$ grep $ list
```

しかし、次のように入力すれば、文字 `$` を含む行だけが表示されます。

```
$ grep '\$' list
```

`grep` コマンドについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

パスワード、プロセス、ディスク記憶装置

SunOS には、コマンド行からシステムに対して実行するコマンドが豊富にあります。この章では、パスワードの設定方法、マシン上で実行されているプロセスの表示方法、不要なプロセスの終了方法、使用中のディスク容量の表示方法について説明します。

パスワードの使い方

システムのセキュリティ保護のため、SunOS ではシステム用のパスワードを設定する必要があります。1年に何度かパスワードを変更するようにすれば、アカウントに簡単にアクセスできるユーザを確実に 1 人だけに制限できます。自分のアカウントを許可なく使ったユーザがいるようであれば、即座にパスワードを変更してください。

パスワードは次の点を考慮して選択します。

- 書き留めなくても覚えられるパスワードを選択する。覚えられないようなパスワードは、簡単に推測できるパスワードよりも不適切です。
- 6 文字以上で、そのうち 1 文字以上が数字からなるパスワードを選択する。
- 自分の名前や頭文字、または配偶者の名前や頭文字は使わない。
- ペットの名前や一般的な趣味に関する名前は使わない。
- 大文字だけからなるパスワードは使わない。
- 複数のアカウントを持っている場合は、各アカウントごとにパスワードを変える。

- パスワードには任意の文字を使えるが、Ctrl-C、Ctrl-Z、Ctrl-U、Ctrl-S、Esc、Tab などの文字、また場合によっては、# や @ などは端末上でシグナルとして解釈されることがある。したがって、これらの文字は使わない。端末によってはこれらの文字が通常のテキスト文字ではなくシグナルとして解釈されるため、パスワードを正しく入力できなくなる場合がある。

パスワードの変更方法

各ユーザのパスワードを変更するには、passwd コマンドを使います。

```
$ passwd
Changing password for hankw on worker
Old password:
New password:
Retype new password:
$
```

1. Old Password: というプロンプトが表示されたら、現在のパスワードを入力します。

(現在、パスワードがアカウントに割り当てられていない場合は、Old Password: プロンプトは表示されません。)パスワードは画面上にエコー (表示) されません。そのため、パスワードを他のユーザに知られなくしてみます。

2. New Password: というプロンプトが表示されたら、新たに設定するパスワードを入力します。

このときも、入力したパスワードは画面上に表示されません。

3. 最後に Retype new password: というプロンプトに対して、新しいパスワードをもう一度入力します。

これは、意図したパスワードを正確に入力できたかどうかを確認するためです。

New Password: に対して入力したのと同じパスワードを入力しないと、パスワードの変更は拒否され、Sorry という応答が返されます。何度もこのような状況になる場合は、システム管理者に問い合わせて新しいパスワードを設定してください。

注 - 6 文字未満のパスワードは使えません。また、新しいパスワードは古いパスワードと比較して 3 文字以上は異ならなければなりません。

パスワードの有効期限を設定する

パスワードの有効期限を設定 (`passwd` コマンドのオプションを使うと設定できる) している場合、パスワードには最長有効期限のみ、または最長有効期限と最短有効期限の両方を持つものがあります。パスワードの有効期限は、システム管理者が設定します。

パスワードの満期日 (最長有効期限) に到達すると、パスワードを変更するようプロンプトが出力されます。このプロンプトはログイン時に出力され、次のようなメッセージが表示されます。

```
Your password has expired. Choose a new one.
```

次に、自動的に `passwd` プログラムが実行され、新しいパスワードを入力するよう要求されます。

また、たとえば、パスワードの最短有効期限が 2 週間と設定されている場合、その期間が経過する前にパスワードを変更しようとする、次のような警告が表示されます。

```
Sorry, less than 2 weeks since the last change.
```

パスワードの有効期限の設定に関する情報を参照する場合は、`passwd` コマンドの `-s` オプションを使います。

```
$ passwd -s  
username ps 2-14-92 1 90 7
```

上記の表示内容は、左から順に、ユーザ名、パスワードの状態 (デフォルトは PS)、現在のパスワードが設定された日付、最短有効期限 (パスワードを変更できるようになるまでの時間)、最長有効期限 (パスワードが無効になるまでの時間)、パスワードの有効期限が経過する前の警告期間となっています。時間はすべて日単位で示しています。

`passwd` と、パスワードの有効期限の設定についての詳細は、『*SunOS リファレンスマニュアル (1): ユーザーコマンド*』を参照してください。

プロセスと PID

各コマンドがシステムによって解釈されると、一意のプロセス識別番号 (PID) を持つ独立したプロセスが生成され、そのコマンドが実行されます。システムは PID を使って各プロセスの現在の状態を追跡します。

現在実行されているコマンドの表示 (ps)

現在実行されているプロセスを調べるには、`ps` コマンドを使います。`ps` コマンドでは、ユーザが所有する各プロセスのプロセス識別番号 (PID の下に表示される) の他に、プロセスが起動された端末 (TTY)、現在までにプロセスが使った CPU 時間 (TIME)、プロセスが実行しているコマンド (COMMAND) などが表示されます。

`ps` コマンドに `-l` オプションを追加すると、現在実行されているプロセスに関するさまざまな詳細情報 (`s` の下に表示される各プロセスの状態など) が表示されます。プロセスの状態を示すコードの意味は次のとおりです。

- `O` - プロセッサ上で実行中
- `S` - スリープ状態。プロセスはイベントが終了するのを待っている。
- `R` - 実行可能状態。プロセスが実行待ち行列内にある。
- `I` - アイドル状態。プロセスは現在作成中。
- `Z` - ゾンビ状態。プロセスはすでに終了し、親プロセスは待機していない。
- `T` - トレース中。親プロセスがそのプロセスをトレース中のため、プロセスはシグナルによって停止されている。
- `X` - `SXBRK` 状態。プロセスは主記憶が空くのを待っている。

`ps` コマンドの実行中でもプロセスの状態は変化します。`ps` で取得できる情報は特定時点の実行状態のスナップショットのため、その出力結果は、`ps` を入力した直後のほんの僅かの時間にしか適用できません。したがって、状態情報が表示された時点で、すでに変わっている可能性もあります。

`ps` コマンドにはこの節で説明している以外に多数のオプションがあります。詳細については、『*SunOS* リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

プロセスの終了方法 (kill)

kill コマンドを使うと、不要になったコマンドプロセスを直接終了できます。このコマンドは、実行に長時間かかるコマンドを誤って入力した場合に特に便利です。

プロセスを終了するには、次の手順を実行します。

1. ps コマンドを入力して終了したいプロセスの **PID** を検索します。
2. **PID** を指定して kill コマンドを入力します。

次の例は、上記の手順を示しています。

```
$ ps
PID  TTY  TIME  COMMAND
1291  co   0:12  -bin/csh (csh)
3250  p0   0:00  ps
1286  p1   0:05  -bin/csh (csh)
3248  p1   0:05  vi commands
$ kill 1291
[1] Terminated  -bin/csh/ (csh)
$
```

ps の出力結果を grep コマンドを通じてパイプすると、適切な PID を簡単に見つけることができます。

```
$ ps | grep commandname
```

commandname は、終了させたいコマンドプロセス名です。

プロセスを強制終了する場合は、次に示すように ps コマンドの `-9` オプションを使えます。

```
$ kill -9 PID#
```

PID# は、強制終了させたいプロセスのプロセス識別番号です。

ディスク記憶容量の管理

ディスク容量は限られた資源であるため、現在使用中のディスク容量を追跡管理することを強くお勧めします。

ディスクの使用状況を表示する (df -k)

df -k を入力すると、システムにマウントされている (直接アクセスできる) 各ディスクの全容量、利用可能な容量、使用中の割合が表示されます。

```
$ df -k
```

ファイルシステムの全容量の 90 % 以上が使用中になった場合は、不要なファイルを削除する必要があります。このためには、不要なファイルを空きの多いディスクやテープに移動したり (cp でコピーしてから rm で削除する)、あるいはそれらのファイルを単に削除したりします。これらの作業は、自分で所有しているファイルについてだけ実行します。

ディレクトリの使用状況を表示する (du)

ディレクトリとそのサブディレクトリの使用状況を 512 バイト単位のブロック数で表示するには、du コマンドを使います。

du コマンドは、各サブディレクトリ内でのディスク使用状況を表示します。ファイルシステム内のサブディレクトリのリストを作成する場合は、cd コマンドでそのファイルシステムに関連するパスに移動し、次のパイプラインを実行します。

```
$ du | sort -r -n
```

sort コマンドの -r (reverse、逆順にソート) オプションと -n (numeric、数字の順にソート) オプションを使った上記のパイプラインによって、各ディレクトリがサイズの大きい順に表示されます。各ディレクトリにあるファイルの大きさ (バイト単位) と更新日付を調べるには、ls -l を実行します。古いファイルや 100 キロバイト以上のテキストファイルは、ディスク容量不足の原因となることがよくあります。

vi エディタの使い方

vi (発音は「ブイアイ」、visual display editor の略) は、SunOS の標準的なテキストエディタです。多目的型エディタである vi はウィンドウベースではないため、各種の端末上でさまざまなファイルの編集に使えます。

vi を使ってテキストの入力と編集ができますが、vi は文書処理プログラムではありません。vi は、一般の文書処理プログラムのような手軽な方法で書式付きテキストを処理できるようには設計されていません。書式付きテキストを出力する場合は、nroff、troff、ditroff などの植字エミュレーションプログラムが必要になります。テキスト中にエミュレータによって解釈されるコードを挿入することで、これらの植字プログラムを使って書式化することができます。

vi には膨大な数のコマンドがあり、そのうち機能の重複するものが相当数あります。初心者なら、最初のうちはその複雑さに圧倒されてしまうかもしれません。この章は vi の最も基本的なコマンドの概要について説明することを目的としています。vi は非常に強力なテキストエディタで、熟練するには多少の時間がかかります。

vi には、view と呼ばれる読み取り専用のバージョンがあります。view を使ってファイルを開くと、vi のコマンドは使えますが、変更内容の書き込みまたは保存はできません。view を使うと、ファイルを参照するときに誤って変更してしまう心配がありません。

vi の起動

次の各節では、vi の起動方法、ファイルへのテキストの入力方法、ファイルの保存(書き込み)方法、vi の終了方法について説明します。また、この章の練習用ファイルも作成します。

ファイルの作成

この例では、vi を起動して paint というファイルを編集します。

```
$ vi paint
```

ファイル paint がすでに存在する場合、vi は既存のファイルを開きます。存在しない場合は、paint ファイルが作成されます。この例では、paint を新規のファイルとします。次のような vi の編集画面が表示されます。

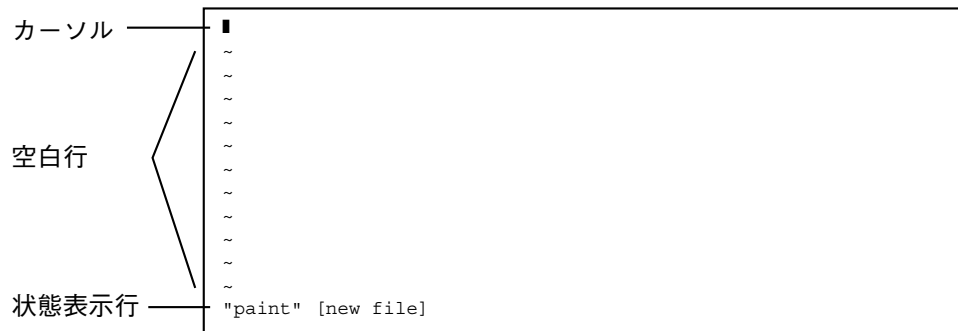


図 6-1 vi の編集画面

カーソルは、画面の左上隅に表示されます。未入力の空白行は、チルド (~) によって示されます。

vi は、ファイル名を指定せず vi と入力するだけでも起動できます。その場合は、vi を終了するときにファイル名を指定できます。

状態表示行

画面の最終行は状態表示行と呼ばれ、ここにはファイル名、ファイルの行数と文字数が表示されます。この例のように新しいファイルを作成した場合は、状態表示行に新規ファイルである旨 (`[new file]`) が表示されます。

vi の 2 種類のモード

vi には、入力モードとコマンドモードという 2 種類の操作モードがあります。入力モードは、ファイルにテキストを入力するために使います。コマンドモードは、vi 特有の機能を実行するコマンドを入力するために使います。vi のデフォルトモードは、コマンドモードです。

vi ではユーザが現在どちらのモードにいるのか明示されないため、vi の初心者にとっては、コマンドモードと入力モードの区別がおそらく最大の混乱要因となります。ただし、早いうちから最低限の基本概念さえ覚えておけば、そのような混乱をしないですみます。

vi を使って最初にファイルを開くと、常にコマンドモードになります。ファイルにテキストを入力するには、vi の入力コマンドを 1 つ入力します。入力コマンドには、現在のカーソル位置にテキストを挿入する `i` (`insert`)、現在のカーソル位置の後にテキストを追加する `a` (`append`) などがあります。(これらの vi 入力コマンドについては、この章の後半で詳細に説明します。)

vi のコマンドモードに戻る場合は、Esc キーを押します。現在どちらのモードにいるのか明らかでない場合は、Esc キーを押してコマンドモードにした上で作業を続けます。vi がすでにコマンドモードのときに Esc キーを押すと、警告音が鳴って画面がフラッシュしますが実害はありません。

入力モード

サンプルファイルの `paint` にテキストを入力するには、vi の「挿入 (`insert`)」コマンドである `i` を入力します。この `i` コマンドにより、vi はコマンドモードから入力モードに切り替わります。(この `i` のように vi のコマンドのほとんどは、表示されません。)

ここで数行の短いテキストを入力し、各行の終わりで `Return` キーを押してみてください。入力した文字はカーソルの左側に表示され、既存の文字は右側に押し出され

ます。入力途中ならば、Return キーを押す前に Back Space キーで後退して再入力することによって誤りを修正できます。入力したテキストの編集については後で説明します。

paint にテキストを入力し終わったら、Esc キーを押してコマンドモードに戻ります。このとき、カーソルは入力した最後の文字に移動します。これで次の vi のコマンドを入力できる状態になります。

vi が予期しない動作をする場合は、「Caps Lock」モードになっていないか確認してください。このモードになっていると、入力した文字がすべて大文字になります。システムによっては、F1 キー (通常は Esc キーの隣にある) が Caps Lock として機能することがあります。そのため、Esc キーの代わりに F1 キーを押してしまうのはよくある誤りです。

注 - 余分なシステムメッセージを削除するためなどで、画面をクリア (再描画) するよう vi に指示しなければならない場合があります。画面をクリアするには、コマンドモードで Ctrl-L を押します。これは、OpenWindows の再表示コマンドと似ています。

コマンドモード

vi を使ってファイルを開くと、コマンドモードになります。このモードでは、広範囲にわたって機能を実行するさまざまなコマンドを入力できます。vi の大部分のコマンドは、1 文字か 2 文字と、任意に指定できる数値からなっています。通常、コマンドには大文字と小文字のバージョンがあり、相互に関連していますが異なる機能を実行します。たとえば、小文字の a を押すとカーソルの右側にテキストを追加できますが、大文字の A を押すとその行の最後にテキストを追加できます。

vi のコマンドの大部分は、Return キーを押さなくても実行されます。ただし、コロン (:) で始まるコマンドは、最後に Return キーを押す必要があります。vi に関する説明では、コロンで始まるコマンドを 3 番目の独立したモード (最終行モード) として取り扱う場合もあります。これは、コマンドモードでコロンを入力すると、コロンとその後に入力した文字が状態表示行に表示されるためです。ただし、この章の説明では、vi の全コマンドはコマンドモードから起動されるものとしします。

コロンで始まるコマンドは、ex のコマンドです。vi と ex は、異なるインタフェースを持つテキスト編集プログラムです。vi が画面指向のインタフェースであるのに対し、ex は行指向のインタフェースです。ex のコマンドは vi 内部から利用できます。コロンを入力した場合は、実際には行指向の ex インタフェースに切

り換えたこととなります。この機能によって、vi を終了しなくても多数のファイル
を操作するコマンドを実行できます。詳細については、この章で後述の 97ページの
「ex コマンドの使い方」を参照してください。

セッションの終了

vi でファイルを編集する場合は、ファイルそのものが直接変更されるのではな
く、vi が一時メモリ内に作成するファイルのコピー (バッファと呼ばれる) が変更
されます。ディスク上にある恒久的なファイルのコピーは、バッファの内容を書き
込んだ (保存した) ときだけ変更されます。

この方法には長所と短所があります。編集セッション中の変更内容をすべて破棄
し、ディスク上のファイルに影響を与えずにファイル編集を中止することができる
反面、システムがクラッシュした場合は、保存していないバッファの内容が失われ
てしまうこととなります。特に、電話回線で接続されたりモート端末上の vi ユー
ザは、予期しない障害にさらされる危険があります。

特に重要な変更を行う場合などは、ファイルを頻繁に保存することをお勧めします。



注意 - 同一ファイルに対して複数の vi セッションを同時に実行することはできま
すが、あまりお勧めできる方法ではありません。ファイルに書き込まれた変更と他の
同時セッションによって上書きされた変更とを見分けることは、非常に困難です。

ファイルの保存と vi の終了

vi には、バッファの内容の保存と vi の終了を制御する類似の意味を持つコマンド
がいくつかあります。これらのコマンドを使うと、「ファイルの保存」、「保存と
終了」、「保存しないで終了」などを選択できます。

保存

バッファの内容を保存する (バッファの内容をディスク上のファイルに書き込む) に
は、次のように入力して **Return** キーを押します。

```
:w
```

保存と終了

ファイルを保存して vi を終了するには、次のように入力して Return キーを押します。

```
:wq
```

上記の代わりに、ZZ を続けて押してもかまいません。

ZZ コマンドについては、前にコロンを入力する必要も、Return キーを押す必要もありません。

保存しないで終了

ファイルを変更していない場合は、次のように入力して Return キーを押せば vi を終了できます。

```
:q
```

ファイルを変更した場合は、:q コマンドでは vi を終了できません。:q と入力すると、「変更後保存していません (:quit! で強制実行します)」というメッセージが表示されます。

変更内容を保存しなくてもよい場合は、次のように入力して Return キーを押します。

```
:q!
```

ファイルの印刷

ファイルを編集して `vi` を終了した後は、次のコマンドを使ってファイル内容を印刷できます。

```
$ lp filename
```

`filename` は、印刷する `vi` ファイル名です。このコマンドによってファイルはデフォルトプリンタに出力されます。ファイルは書式化されず画面上に表示されているとおり各行ごとに印刷されます。プリンタのコマンドについての詳細は、第 8 章を参照してください。

`vi` の基本コマンド

以降の節では、次に示すさまざまな `vi` のコマンドについて説明します。

- ファイル内の移動
- テキストの挿入
- テキストの変更と置換
- テキストの変更の取り消し
- テキストの削除
- テキストのコピーと移動
- コマンドの繰り返し

ファイル内の移動

前節では、ファイルの作成、保存、印刷の方法と `vi` の終了方法を学習しました。ファイルが作成されたので、次にファイル内を移動するために必要な概念を理解しておく必要があります。練習用ファイルを開いて、この節で説明する各コマンドを実際に試してみてください。

カーソルの移動

vi の起動時には、カーソルは vi 画面の左上隅に表示されます。コマンドモードでは、複数のキーボードコマンドを使ってカーソルを移動できます。カーソルの移動には、特定の文字キー、矢印キー、Return キー、Back Space キー (または Delete キー)、Space Bar キーが使えます。

注 - vi の大部分のコマンドでは大文字と小文字が区別されます。そのため、「同じ」キーによるコマンドであっても、大文字と小文字とではまったく異なる結果となる場合があるので注意してください。

矢印キーによる移動

キーボードに矢印キーが付いている場合は、これらを使ってみてください。上下左右の矢印キーを使うと、画面内で自由自在にカーソルを移動できます。カーソルの移動ができるのは、すでに表示されているテキストまたは入力領域内に限られます。

リモート端末から vi を使う場合は、矢印キーが正しく動作しないことがあります。これは端末エミュレータの機能によって決まります。矢印キーが正しく動作しない場合は、次のような方法でカーソルを移動します。

- 左に移動するには、h を押します。
- 右に移動するには、l を押します。
- 下に移動するには、j を押します。
- 上に移動するには、k を押します。

ワード単位の移動

ワード単位で右にカーソルを移動する場合は、w (「word」を表す) を押します。(Solaris 2.5 and ... という文の場合、w を押すごとに S 2 . 5 a の順に移動します。)

ワード単位で左にカーソルを移動するには、b (「back」を表す) を押します。

いちばん近くにある区切り文字を通り越して右または左のワード (スペースで区切られているワード) の最初の文字にカーソルを移動するには、w または B を押します。(Solaris 2.5 and ... という文の場合、w を押すごとに S 2 a の順に移動します。)

現在カーソルがあるワードの最後の文字にカーソルを移動するには、e (「end」を表す) を押します。

行頭または行末への移動

現在カーソルがある行の先頭に、カーソルを移動するには、`^`を押します。

現在カーソルがある行の末尾に、カーソルを移動するには、`$`を押します。

1行下への移動

1行下の、最初の(空白ではない)文字にカーソルを移動するには、`Return`キーを押します。

左への移動

1文字分左にカーソルを移動するには、`Back Space`キーを押します。

右への移動

1文字分右にカーソルを移動するには、`Space Bar`キーを押します。

先頭行への移動

画面の先頭行にカーソルを移動するには、`H`(「high」を表す)を押します。

中央行への移動

画面の中央行にカーソルを移動するには、`M`(「middle」を表す)を押します。

最終行への移動

画面の最終行にカーソルを移動するには、`L`(「low」を表す)を押します。

ページングとスクロール

画面の最終行にカーソルがあるときに下に移動する場合、または画面の先頭行にカーソルがあるときに上に移動する場合は、テキストがスクロールアップまたはスクロールダウンされます。非常に短いファイルの場合はこのテキスト表示方法は効果的ですが、長いファイル内を移動する場合には少々手間がかかります。

ファイル内で 1 画面分または半画面分まとめて上方向や下方向にカーソルをページング、またはスクロールすることができます。(これらのコマンドを `paint` というファイルで試すには、ファイルが十分長くなるようにテキストを追加します。)

スクロールとページングとは基本的に異なります。テキストのスクロールでは、カーソルを一度に行単位で上下に移動しますが、ページングではカーソルを一度に画面単位で上下に移動します。高速のシステム上では、この違いに気付かないこともあります。リモート端末から `vi` を使っている場合やシステムの実行速度が通常より遅い場合は、この違いがきわめて明白になります。

次の 1 画面へのページング

1 画面先 (下方向) にカーソルを移動するには、`Ctrl-F` を押します (Control キーを押しながら `F` キーを押します)。カーソルは、新しい画面の左上隅に移動します。

次の半画面へのスクロール

半画面先 (下方向) にスクロールするには、`Ctrl-D` を押します。

前の 1 画面へのページング

1 画面前 (上方向) にカーソルを移動するには、`Ctrl-B` を押します。

前の半画面へのスクロール

半画面前 (上方向) にスクロールするには、`Ctrl-U` を押します。

テキストの挿入

`vi` にはテキストを挿入するためのコマンドが多数あります。この節では、これらのコマンドのうち最も便利なものを紹介します。これらのコマンドが実行するモードは、入力モードです。これらのコマンドを使う場合、最初はコマンドモードでなければなりません。Esc キーを押してコマンドモードであることを確認してください。

追加

カーソルの右側にテキストを追加するには、**a (append)** コマンドを使います。追加したい位置の左にカーソルを移動して **a** を押し、追加するテキストを続けて入力してみてください。入力が終わったら **Esc** キーを押します。

行の末尾にテキストを追加するには、**A** コマンドを使います。このコマンドの動作を確認するために、追加したい行にカーソルを移動して (その行内ならカーソルはどこにあってもよい)、**A** を押します。カーソルは行の末尾に移動し、その位置からテキストを続けて入力できます。入力が終わったら **Esc** キーを押します。

挿入

カーソルの左側にテキストを挿入するには、コマンドモードで **i (insert)** コマンドを使います。

行の先頭にテキストを挿入するには、**I** を押します。このコマンドを使うと、カーソルの位置がその行の先頭 (空白ではない) 文字に移動し、そこからテキストを入力できます。テキスト入力後にコマンドモードに戻るには、この節で説明する他のコマンドと同様に **Esc** キーを押します。

行の挿入

カーソルがある行の上または下に新規の行を挿入するには、次のコマンドを使います。

カーソルがある行の下に新規の行を挿入するには、**o** コマンドを使います。**o** を入力した後にテキストを入力します。複数行にわたるようなテキストも入力することができます。入力が終わったら **Esc** キーを押します。

カーソルがある行の上に新規の行を挿入するには、**O (o の大文字)** コマンドを使います。

テキストの変更

テキストの変更とは、テキストの一部を別のテキストで置換することです。**vi** には状況に応じてテキストを変更するための方法が複数あります。(テキスト変更のコマンドを入力後、置換される部分を明らかにするために、1文字の場合はその文字が、テキストなどの場合はその最後の文字が、**\$** で表示されます。)

ワードの変更

ワード全体を置換するには、置換するワードの先頭にカーソルを移動し、`cw` と続けて入力してから新しいワードを入力します。入力が終わったら `Esc` キーを押します。

ワードの後半部分だけを変更するには、保存する部分の右側にカーソルを移動し、`cw` と続けて入力してから新しいテキストを入力します。入力が終わったら `Esc` キーを押します。

ワードの前半部分だけを変更するには、保存する部分の先頭にカーソルを移動し、`cb` と続けて入力してから新しいテキストを入力します。入力が終わったら `Esc` キーを押します。

行の変更

行全体を置換するには、変更したい行に移動して (その行内ならカーソルはどこにあってもよい)、`cc` と続けて入力します。すると、その行内の文字はすべて消えます (置換される部分を示す `$` は表示されません)。新しいテキスト (長さは任意) の入力が終わったら `Esc` キーを押します。

行の一部分の変更

行の一部分だけを変更するには、保存する (変更せずにそのままにしておく) 部分の右側にカーソルを移動し、`c` を押してから新しいテキストを入力します。入力が終わったら `Esc` キーを押します。これによって、`c` と押した時のカーソル位置からその行の末尾までの部分が新しいテキストに変更されます。

1 文字以上の置換

カーソルがある文字を、1つ以上の文字からなるテキストに置換するには、`s` を押してから新しいテキストを入力します。コマンドモードに戻るには、`Esc` キーを押します。

1 文字の置換

カーソルがある文字を、別の1文字に置換するには、`r` コマンドを使います。置換する文字の上にカーソルを移動し、`r` を押してから1文字だけ入力します。この置

換が終了すると、vi は自動的にコマンドモードに戻ります (Esc キーを押す必要はありません)。

文字の入れ替え

vi では 2 つのキー操作だけで、前後を逆に入力してしまった文字を修正できます。たとえば the と入力するつもりで誤って teh と入力してしまったと想定します。これを修正するには、入れ替える最初の文字の上 (この場合は e) にカーソルを移動し、xp と続けて入力します。e と h の位置が入れ替わり、vi は自動的にコマンドモードに戻ります (Esc キーを押す必要はありません)。

行の分割または連結

テキストを変更しないで行を分割するには、別の行に分割する空白の位置にカーソルを移動し、r (「replace」を表す) を押した後で Return キーを押します。文字の上カーソルを置いた状態で r を押して Return キーを押すと、その文字が改行文字で置換されてしまうので注意してください。

2 つの行を連結するには、上側の行にカーソルを移動して大文字の J を押します (J を押した後で Esc キーを押す必要はありません)。

変更の取り消し

vi でテキストを編集してファイルを変更するときは、誤った入力を取り消したいと思う場合が必ずあります。vi の u (「undo」を表す) コマンドを使うと、直前の操作を取り消した状態から編集作業を再開できます。

前回のコマンドの取り消し

vi で入力を誤ったり、特定の操作を実行した後で気が変わった場合は、コマンドを実行した直後に u を押すことによってそのコマンドを取り消しできます (u を押した後で Esc キーを押す必要はありません)。もう一度 u を押すと、u コマンド自身が取り消されます。

行に対する変更の取り消し

特定の行に対する変更をすべて取り消すには、`U` コマンドを使います。このコマンドは、その行からカーソルを移動していない場合にだけ有効です (`U` を押した後で `Esc` キーを押す必要はありません)。

テキストの削除

`vi` の削除コマンドを使って、指定した文字、ワード、行の削除ができます。削除後は、`vi` はコマンドモードのままなので、入力モードにするためのコマンドを入力してからテキストを挿入してください。

1 文字の削除

1 文字を削除するには、削除する文字の上にカーソルを移動して `x` を押します。

`x` コマンドは、その文字自身と、その文字が占めていたスペースも削除します。つまり、ワードの真ん中から 1 文字を削除すると、その 1 文字分のスペースも削除されて残りの文字が詰められます。`x` コマンドを使って行内の空白 (つまりその空白自身と、その空白が占めていたスペース) を削除することもできます。

カーソルの前方 (左側) の文字を 1 文字削除する場合は、大文字の `X` を押します。

ワードまたはワードの一部の削除

ワードを 1 つ削除するには、削除するワードの先頭にカーソルを移動して `dw` と続けて入力します。ワードとそれが占めていたスペースが削除されます。

ワードの一部だけを削除するには、保存する (削除せずにそのままにしておく) 部分の右にカーソルを移動して `dw` と続けて入力します。ワードの残りの部分とそれが占めていたスペースが削除されます。

行の削除

行を削除するには、削除したい行に移動して (その行内ならカーソルはどこにあってもよい)、`dd` と続けて押します。その行と、その行が占めていたスペースが削除されます。

行の一部分の削除

行の一部分だけを削除することもできます。

カーソルより右側の全テキストを削除するには、保存する (削除せずにそのままにしておく) 部分の右にカーソルを移動して `D` を押します。

カーソルより左側の全テキストを削除するには、保存する部分の先頭にカーソルを移動して `d0` (`d` のあとにゼロ) と入力します。

ファイルの終わりまでの削除

カーソルがある行からファイルの最終行までの全テキストを削除するには、カーソルがある行で `dG` と続けて入力します。このコマンドによってカーソルがあった行も削除されます。

ファイルの先頭からの削除

ファイルの先頭行から現在カーソルがある行までの全テキストを削除するには、カーソルがある行で `d1G` と続けて入力します。このコマンドによってカーソルがあった行も削除されます。

テキストのコピーと移動 — `yank`、`delete`、`put`

大部分の文書処理プログラムと同様に `vi` エディタでも、テキスト行の「コピー&ペースト」と「カット&ペースト」ができます。`vi` で「コピー&ペースト」を行う場合は、`yank` と `put` コマンドを使います。また、「カット&ペースト」の場合は、`delete` と `put` を使います。

`vi` で比較的短いテキストブロックのコピーや移動を行う場合は、`yank`、`delete`、`put` の各コマンドを組み合わせて使います。

行のコピー

行をコピーするには、「`yank`」を表す `yy` (または `Y`) と、「`put below`」を表す `p` (または「`put above`」を表す `P`) という2つのコマンドが必要です。`Y` と `yy` は同じ働きをします。

1行をバッファに取り込む (`yank`) には、その行の任意の位置にカーソルを移動して `yy` と続けて入力します。次に、取り込んだ行を挿入 (コピー) する位置の上の行

にカーソルを移動して `p` を押します。その取り込んだ行が `p` と押した行の下に表示されます。

取り込んだ行をカーソルがある行の上にコピーするには、`P` を押します。

`yy` コマンドでは行数を指定できます。たとえば、バッファに 11 行を取り込むには、`11yy` と続けて入力します。すると、カーソルがある行から下方向に数えて 11 行目までがバッファに取り込まれ、画面の最終行にその旨を示すメッセージ「11 行をバッファにコピーしました」が表示されます。

先に説明した任意の削除コマンドの直後に `p` または `P` コマンドを使うと、削除したテキストをカーソルがある行の下の行または上の行に挿入できます。



注意 - コピーや移動 (`yank`、`delete`、`put`) の操作中は、カーソルの移動コマンドだけを使ってください。テキストを新しい位置に挿入する前に他のテキストをコピーしたり移動したりすると、先にコピーや移動のためにバッファに取り込んだ内容が失われてしまいます。

行の移動

行の移動についても、`dd` (「delete」を表す) と、`p` (または `P`) という 2 つのコマンドが必要です。

行を移動するには、まずその移動したい行に移動して (その行内ならカーソルはどこにあってもよい)、`dd` と続けて入力します。たとえば、その行とその行から数えて 5 行目までの 5 行分を移動の対象行としたい場合は、その行で `5dd` と続けて入力します。このコマンドによって、移動の対象行が削除されます。

次に、その削除した行を挿入する位置の上の行に、カーソルを移動して `p` を押します。その行がカーソルのあった行の下に挿入されます。

また、その削除した行をカーソルのある行の上に挿入するには、`P` を押します。

名前付きバッファの使い方

別個の行グループを文書内のさまざまな位置に繰り返し挿入 (コピーまたは移動) するために、それらの行を名前付きバッファに取り込むことができます。名前付きバッファを指定するには、コマンドの前に二重引用符 (") とバッファ名 (`a` から `z` までのうちの 1 文字) を付けます。たとえば、`a` という名前付きバッファに 4 行を取り込むには (コピーの場合)、その最初の行で `"a4yy` と続けて入力します。他のバッファを同時にいくつも使えるため、このコピー操作の他に、ある位置から削除した

テキストを他の複数の位置に移動することもできます。たとえば、12 行を削除して *b* という名前付きバッファに取り込むには、“*b12dd*”と続けて入力します。

取り込んだテキストを挿入するには、*p* または *P* コマンドの前に “*n*” を付けます (*n* は名前付きバッファ)。たとえば、*b* というバッファに保存した行を挿入する場合は、“*bp*” または “*bP*” と続けて入力します。

同じ名前付きバッファに新しい行を取り込むと、元のバッファは上書きされます。名前付きバッファは、*vi* を終了するまで保存されたままです。

名前付きバッファを使うと、名前付きバッファに保存したテキストに影響を与えることなく、他のテキストを確実にコピーしたり移動したりできます。ただし、同じ名前付きバッファを意図的に上書きしないことが前提条件です。

カウントを使ってコマンドを繰り返す

ほとんどの *vi* のコマンドの前には、カウントと呼ばれるリピートファクタ (コマンドの前に付ける数値で、その操作を何回繰り返すかを指示する) を指定できます。

これまでの節で説明した大部分のコマンドに、カウントを指定できます。たとえば、*3dd* は行を削除する操作を 3 回繰り返します (したがって、カーソルがあった行も含めて 3 行が削除されます)。*2dw* はカーソルがあった位置のワードも含めて 2 つのワードを削除し、*4x* はカーソルがあった位置も含めて 4 つの文字または空白を削除します。またカウントを移動コマンドの前に付ければ、*3w* や *2Ctrl-F* (2 を押してから、*Ctrl-F* を押す) のように入力してカーソルを移動できます。これらの機能については、*vi* コマンドを習得するにしたがって理解できるようになります。

ピリオド (.) を押すと、前回実行したテキスト編集コマンドが繰り返されます。たとえば、*dd* を使ってある行を削除した場合は、カーソルを別の行に移動してピリオドを押せば、その別の行を削除できます。

ex コマンドの使い方

比較的長いテキストブロックを取り扱う場合は、*yank*、*delete*、*put* よりも *ex* コマンドを利用した方が正確で便利です。*ex* コマンドを使うと、画面上で行を数えたり挿入位置を目で探したりするのではなく、行範囲を指示してから挿入位置の上にくる行番号を指定することで移動またはコピーができます。(削除コマンドの場合、挿入位置の指定は必要ありません。)

行番号の表示と非表示

行番号を表示するには、`:set nu` と入力して **Return** キーを押します。

行番号は左端に表示されます。なお、ファイルを印刷しても、これらの行番号は印刷されません。行番号が見えるのは画面上だけです。

```
1 Oh, when I die, take my saddle from the wall,  
2 Put it on my pony, lead him out of the stall.  
3 Tie my bones to his back, point our faces to the west,  
4 And we'll ride the prairies that we love the best.  
5  
6 Ride around, little doggies,  
7 Ride around real slow.  
8 Firey and Snuffly are rarin' to go.  
~  
~  
~  
~  
~  
~  
~  
:set nu
```

行番号を表示させない場合は、`:set nonu` と入力して **Return** キーを押します。

行のコピー

ex コピーコマンドの基本形式は次のとおりです。

```
:line#,line# co line#
```

コンマで区切られた最初の 2 つの数値は、コピーされる行の範囲を指定します。3 番目の数値は、それを挿入する位置の上の行番号です。

たとえば、1 行目から 5 行目までをコピーしてそのブロックを 12 行目の下に挿入する場合は、次のように入力して **Return** キーを押します。

```
:1,5 co 12
```

行の範囲を指定するときに、次の省略形を使用できます。

- ピリオド(.) は、「現在カーソルがある行」を意味します。
- ドル記号(\$) は、「ファイルの最終行」を意味します。

たとえば、「現在カーソルがある行から 5 行目まで」の範囲をコピーしてそのブロックを 12 行目の下に挿入する場合は、次のように入力します。

```
:.,5 co 12
```

また、「6 行目からファイルの最終行まで」の範囲をコピーしてそのブロックを 2 行目の下に挿入する場合は、次のように入力します。

```
:6,$ co 2
```

行の移動

ex 移動コマンドの基本形式は、上で説明したコピーコマンドの形式と似ています。

```
:line#,line# m line#
```

行範囲と挿入位置 (行) の指定方法はコピーコマンドの場合と同様です。省略形のピリオドとドル記号も使用できます。コピーと移動の機能の違いは、移動の場合はブロックがある位置から削除されて他の位置に挿入されることです。

たとえば、1 行目から 5 行目までを 12 行目の下に移動する場合は、次のように入力して Return キーを押します。

```
:1,5 m 12
```

行の削除

ある範囲の行を削除するには、次の形式のコマンドを使います。

```
:line#,line# d
```

たとえば、1 行目から 5 行目までを削除する場合は、次のように入力します。

```
:1,5 d
```

vi による検索と置換

vi には指定された文字列をファイル内で検索するための方法がいくつかあります。また、広範囲にわたった置換機能も利用できます。

文字列の検索

文字列とは、1つ以上の文字の配列です。文字列には、英字、数字、句読点、特殊文字、空白、タブ、改行を入れることができます。文字列は、文法的に正しいワード、ワードの一部だけのどちらでもかまいません。

文字列を検索するには、/の後に検索する文字列を入力して **Return** キーを押します。カーソルがある位置から検索を始め、その文字列の先頭位置にカーソルが移動します。たとえば `meta` という文字列を検索する場合は、`/meta` と入力して **Return** キーを押します。

それと同じ文字列をさらに検索するには、`n` を押します。カーソル位置よりも前(逆)方向に検索するには、`N` を押します。

ファイル内で逆方向(画面の上の方向)に文字列を検索する場合は、/の代わりに?を使います。その場合、`n` と `N` では / の時と方向が逆になります。

通常の実験では大文字と小文字が区別されます。したがって、`china` の検索では `China` は見つかりません。検索時に大文字と小文字の区別を無視させる場合は、`:set ic` と入力します。デフォルトの実験モードに戻すには、`:set noic` と入力します。

検索する文字列が見つかったら、最初の出現位置にカーソルが移動します。文字列が見つからない場合は、状態表示行に「パターンが見つかりません」というメッセージが表示されます。

特殊文字 (`/ & ! . ^ * $ \ ?`) は検索機能にとって特別な意味を持っているため、検索対象の文字列中で使う場合は「エスケープ」する必要があります。特殊文字をエスケープするには、その文字の前にバックスラッシュ (`\`) を付けます。たとえば `anything?` という文字列を検索する場合は、`/anything\?` と入力して **Return** キーを押します。

これらの特殊文字は、検索機能に対するコマンドとして使えます。したがって、これらの文字を1つ以上含む文字列を検索する場合は、その文字の前にバックスラッシュ

シュを付けてコマンドではないことを示す必要があります。バックスラッシュ自身をエスケープするには、\\ と入力します。

高度な検索

次の特性を表すタグを文字列に付けることによって、さらに高度な検索ができます。

- 行の先頭
- 行の終わり
- ワードの先頭
- ワードの終わり
- ワイルドカード文字

行の先頭を表すには、検索文字列の最初にキャレット (^) を付けます。たとえば、Search で始まる行を検索する場合は、次のように入力します。

```
/^Search
```

行の終わりを表すには、検索文字列の最後にドル記号 (\$) を付けます。たとえば、search. で終わる行を検索する場合は、次のように入力します。(ピリオドをバックスラッシュでエスケープしていることに注意してください。)

```
/search\.$
```

ワードの先頭を表すには文字列の最初に \< を付け、ワードの終わりを表すには文字列の最後に \> を付けます。したがって、文字列ではなくワードを検索する場合は、ワードの先頭を表すタグとワードの終わりを表すタグを検索パターン中で組み合わせて使います。たとえば、search というワード (文字列中にあるものは除く) の出現位置を検索するには、次のように入力します。

```
/\
```

任意の 1 文字を表すには、文字列中のその文字の位置にピリオド (.) を入力します。たとえば、disinformation または misinformation の出現位置を検索するには、次のように入力します。

```
/.isinformation
```

この検索はワードではなく文字列を対象としているため、上記の検索パターンは `misinformationalist` や `disinformationism` という構文とも一致します。

指定した文字のどれかと一致する文字を検索するには、それらの文字を括弧 `[]` で囲みます。たとえば、`/[md]string` という検索パターンは、`m` または `d` で始まる文字列と一致します。また、`/[d-m]string` という検索パターンは、`d` から `m` のうちの 1 文字で始まる文字列と一致します。

文字列中にアスタリスク (`*`) を使うと、`*` の直前の文字のゼロ個以上の繰り返しを表すことができます。括弧 `[]` と `*` を効果的に組み合わせると、検索する文字列を細かく指定できます。たとえば、次の例のように入力すると、文字列 `isinformation` が出現する位置、および `a` から `z` のうちの 1 文字で始まり文字列 `isinformation` で終わる文字列が出現する位置をすべて検索します。

```
/[a-z]*isinformation
```

文字列の置換

以降で使われる文字列を置換する手順は、上記で説明した検索手順に基づいています。検索パターンを表す特殊文字は、検索および置換の手順でもすべて同様に使えます。

置換コマンドの基本形式は次のとおりです。

```
:g/search-string/s//replace-string/g
```

上記のコマンドを入力した後に `Return` キーを押します。

たとえば、`disinformation` という文字列をすべて `newspeak` で置換するには、次のように入力して `Return` キーを押します。

```
:g/disinformation/s//newspeak/g
```

このコマンドで、文字列が見つかるたびに検索を中断して置換を行うかどうかを確認するメッセージを出させることもできます。次のコマンド例では、`gc` (「consult」を表す `c` を `g` に追加したもの) を使って、`disinformation` が見つかるたびに検索を

中断して置換を行うかどうかを確認するメッセージを出すよう設定しています。置換する場合は `y`、置換しない場合は `n` を入力してから Return キーを押します。

```
:g/disinformation/s//newspeak/gc
```

注 - この応答形式の検索と置換を中断するには、Ctrl-C を押します。Ctrl-C の入力以前に `y` と入力されたものに関しては、置換されたままとなります。

指定行へジャンプする

編集ファイルの最終行にジャンプするには、`G` を押します。ファイルの先頭行に戻るには、`1G` と続けて押します。

行番号の数字を押した後で `G` と押せば、指定した行にジャンプできます。

たとえば、あるファイルの 51 行目を編集してから `vi` を終了したとします。その後、51 行目からまた編集したい場合は、`vi` でそのファイルを開いてから `51G` と続けて押してください。その 51 行目にカーソルが移動し、その行から編集を始めることができます。

編集中のファイルに別ファイルを読み込む

`vi` では、編集中のファイルに別のファイルを簡単に「読み込む」(挿入する)ことができます。コマンドの一般的な形式は次のとおりです。

```
:line# r filename
```

行番号 `line#` を指定しない場合 (`:r filename`) は、カーソルがある行の下にファイル `filename` が読み込まれます。

たとえば、`orwell` というファイルをあるファイルの 84 行目の下に挿入したい場合は、次のように入力します。

```
:84 r orwell
```

または、84 行目にカーソルを移動 (84G と続けて入力) してから、次のように入力することもできます。

```
:r orwell
```

複数のファイルの編集

vi では複数のファイルを編集できます。たとえば、paint と orwell というファイルを交互に編集する場合は、次のようにします。

1. 最初に paint の変更内容を保存するために、:w と入力して **Return** キーを押します。
2. orwell を編集するために、:n orwell と入力して **Return** キーを押します。
3. orwell を編集して変更内容を保存します。
4. orwell の編集を完了して変更内容を保存した後は、次の **3** つのどれかを選択できます。
 - vi を終了する - :q と入力して **Return** キーを押します。
 - paint に戻る - :n # と入力して **Return** キーを押します。
 - 2 つのファイルを交互に移動する - :n # を繰り返し実行します。



注意 - :w filename (別の名前のファイルに保存) または :r filename (他のファイルを読み込む) などを編集中に使うと、:n # としても paint や orwell ではなくそのファイル filename になってしまうので注意してください。

複数ファイルの編集

複数のファイルを編集するには、コマンドプロンプトで `vi` を起動するときに、`vi` コマンドの後にファイル名をそれぞれスペースで区切って指定します。

```
$ vi paint orwell
```

各ファイルは、指定された順番で表示されます。上記の例では、`paint` が最初に表示されます。`paint` の編集が終わって `:n` と入力すると、次のファイルの `orwell` の編集に移ります。現在のファイルの変更内容は保存しないで次のファイルの編集に移るには、`:n` の代わりに `:n!` と入力します。

関連したファイル名を持つ複数のファイル (例: `test1`、`test2`、`test3`) を編集する場合は、ワイルドカード文字を使ってそれらのファイルを指定できます。

```
$ vi test*
```

編集の対象となるファイルが、ファイル名の数字またはアルファベットの順に表示されます。

ファイル間で行をコピーする

あるファイルから別のファイルに行をコピーするには、次の手順に従います。

1. コピーしたい行を含むファイルを開きます。
2. `yank` コマンドを使って、コピーしたい行を名前付きバッファに保存します。
たとえば、そのファイルの 10 行分をバッファ `a` に保存するには、コピーしたい行のうち先頭の行にカーソルを移動して `"a10Y` と続けて入力します。
3. `vi` を終了せずに、別のファイル (この例では `orwell`) を編集します。

```
:n orwell
```

4. `put` コマンドを使って、手順 2 で保存した行を挿入します。

たとえば、バッファ *a* の内容を現在カーソルがある行の下に挿入するには、そこで "ap と続けて入力します。

vi を終了すると、名前付きバッファの内容はすべて失われることに注意してください。名前付きバッファに関連する操作が完了するまで、終了 (:q) コマンドを使ってはいけません。

vi パラメータの設定状況

vi には、その動作と表示に影響を与えるパラメータが多数あります。vi の実行中に次のコマンドを入力して Return キーを押すことにより、これらのパラメータと現在の設定内容のリストを参照できます。

```
:set all
```

クラッシュからの回復

システムがクラッシュしたときは、バッファの内容が危険にさらされますが、多くの場合は、以下の形式で vi を再起動することによって大部分の編集内容を回復できます。

```
vi -r filename
```

filename は、クラッシュ時に編集していたファイル名です。通常はシステムを再起動したときに、回復可能なファイルが存在する旨のメールが送信されます。

vi の基本コマンドの一覧

次の表は、vi の基本コマンドの一覧です。

表 6-1 vi の基本的なコマンド

コマンド	意味
vi の起動	
vi <i>filename</i> Return キー	ファイルを開くまたは新規作成する
vi Return キー	新規ファイルを開く (ファイル名は後で指定)
vi -r <i>filename</i> Return キー	クラッシュしたファイルを回復する
view <i>filename</i> Return キー	読み取り専用でファイルを開く
カーソル移動コマンド	
h	左に移動
j	下に移動
k	上に移動
l	右に移動
w	ワード単位で右に移動
W	(スペースで区切られた) ワード単位で右に移動
b	ワード単位で左に移動
B	(スペースで区切られた) ワード単位で左に移動
e	現ワードの最後の文字に移動
Return キー	下の行の先頭文字 (空白ではない) に移動
Back Space キー	左に移動
Space Bar キー	右に移動
H	画面の先頭行に移動

表 6-1 vi の基本的なコマンド 続く

コマンド	意味
M	画面の中央行に移動
L	画面の最後行に移動
Ctrl-F	1 画面先のページを表示
Ctrl-D	半画面先にスクロール
Ctrl-B	1 画面前のページを表示
Ctrl-U	半画面前にスクロール
文字と行の挿入	
a <i>string</i> ESC キー	カーソルの右にテキストを追加
A <i>string</i> ESC キー	行の末尾にテキストを追加
i <i>string</i> ESC キー	カーソルの左にテキストを追加
I <i>string</i> ESC キー	行の先頭にテキストを挿入
o <i>string</i> ESC キー	カーソルがある下の行にテキストを挿入
O <i>string</i> ESC キー	カーソルがある上の行にテキストを挿入
テキストの変更	
cw <i>string</i> ESC キー	カーソルのワード (またはワードの右側の部分) を変更
cc <i>string</i> ESC キー	行全体を変更
C <i>string</i> ESC キー	カーソル位置から行の末尾までを変更
s <i>string</i> ESC キー	カーソルの 1 文字をテキストに変換
r <i>character</i>	カーソルの文字を別の 1 文字に置換

表 6-1 vi の基本的なコマンド 続く

コマンド	意味
r Return キー	行を分割 (空白にカーソルがある場合)
J	カーソルがある行とその下の行を連結
xp	カーソルの文字とその右の文字を入れ替える
~	大文字を小文字に、小文字を大文字に変換
u	前回の変更などのコマンドを取り消す
U	カーソルがある行に対する全変更を取り消す
:u Return キー	前回の変更などのコマンドを取り消す
テキストの削除	
x	カーソルの文字を削除
X	カーソルの左の文字を削除
dw	ワード (またはワードの右側の部分) を削除
dd	行を削除
D	ある行のうちカーソルから右側の部分を削除
dG	カーソルがある行からファイルの最終行までを削除
d1G	ファイルの先頭行からカーソルがある行までを削除
:5,10 d Return キー	5 行目から 10 行目までを削除
テキストのコピーと移動	
YY	行をコピー
Y	行をコピー

表 6-1 vi の基本的なコマンド 続く

コマンド	意味
p	コピーまたは移動の対象行をカーソルがある行の下に挿入
P	コピーまたは移動の対象行をカーソルがある行の上に挿入
:1,2 co 3 Return キー	1 行目から 2 行目までを 3 行目の下にコピー
:4,5 m 6 Return キー	4 行目から 5 行目までを 6 行目の下に移動
行番号の設定	
:set nu Return キー	行番号を表示する
:set nonu Return キー	行番号を表示しない
大文字と小文字の区別	
:set ic Return キー	検索時に大文字と小文字の区別をしない
:set noic Return キー	検索時に大文字と小文字を区別する
カーソルのジャンプ	
G	ファイルの最終行にジャンプ
1G	ファイルの先頭行にジャンプ
21G	21 行目にジャンプ
検索と置換	
/string Return キー	文字列を検索
?string Return キー	文字列を逆方向に検索
n	検索方向の前方にある文字列を検索
N	検索方向の後方にある文字列を検索

表 6-1 vi の基本的なコマンド 続く

コマンド	意味
:g/search/s//replace/g Return キー	文字列の検索と置換
画面のクリア	
Ctrl-L	乱れた画面をクリア (再表示)
編集中のファイルに対する ファイルの読み込み	
:r filename Return キー	カーソルがある行の下にファイルを挿入 (読み込み)
:34 r filename Return キー	34 行目の下にファイルを挿入 (読み込み)
保存と終了	
:w Return キー	変更を保存 (バッファをファイルに書き込む)
:w filename Return キー	指定されたファイルにバッファを書き込む
:wq Return キー	変更を保存して vi を終了
ZZ	変更を保存して vi を終了
:q! Return キー	変更を保存しないで vi を終了

メールの使い方

SunOS には、電子メール (email) の送受信に使われる mailx というプログラムがあります。mailx を使って、メッセージの読み取りと書き込み、送信と受信、および保存と削除ができます。mailx プログラムはウィンドウベースではないため、任意の端末上で実行できます。ウィンドウベースのメールのほうが好まれるかもしれませんが、簡単なメッセージを急いで送信する場合は、mailx プログラムのほうが便利です。ユーザ専用のメールエイリアスを設定する場合も、この章を参照してください。

注 - OpenWindows 環境でメールツールのアイコンが画面に表示されている場合は、メールツールを終了してからこの章の例を実行してください。メールツールを終了しないと、2つのメールプロセスが有効になるため、エラーメッセージや警告が出されることがあります。コマンドツールやシェルツールのウィンドウ内では正常にメッセージの送受信ができます。しかし、メールを読み取った後でメッセージを保存したり削除すると、受信箱の「トレイ」が影響を受けるため、メールツールの混乱の原因になります。

mailx の基本

この節では、mailx を使うために最低限必要なことについて説明します。この章の後半では、mailx プログラムを効率よく使うための機能と特徴について説明します。

メールの受信者のログイン名とマシン名は、mailx プログラム用の一意のアドレスとして使われます。メールの受信者が送信者と同じマシン上にいる場合は、ログイン名だけ分かれば十分です。各ユーザは、メールを受信するためのメールボックスを持っています。通常、このメールボックスは /var/mail/username ディレクトリ (username は各ユーザのログイン名) にあります。

メールが受信されると mailx プログラムはその旨をユーザに知らせ、メールボックスにメールを収納します。ユーザがメールを読んだ後、mailx はそれらのメールを自動的に mbox という記録ファイルに格納します。mbox ファイルは、ユーザのホームディレクトリにあります。

mailx の起動

mailx を起動するには、プロンプトに対して次のコマンドを入力し、Return キーを押します。

```
$ mailx
```

メールボックスにメールがない場合は、次のようなメッセージが画面に表示されます。

```
No mail for username
$
```

username は各ユーザのログイン名です。

自分宛てにメールを送信する

mailx の機能を試すには、まず最初に自分自身にメールを送信してみましょう。プロンプトに対して mailx コマンドをもう一度入力しますが、今回は自分のアドレス (ログイン名とマシン名) も指定します。たとえば、ログイン名が rose でマシン名が texas の場合、アドレスは rose@texas となります (@ シンボルは「at」の意味)。ローカルネットワーク上でのログイン名を使うこともできます (疑問点についてはシステム管理者に問い合わせてください)。

```
$ mailx rose@texas
```

mailx プログラムは、Subject: という行を表示します。

```
$ mailx rose@texas
Subject:
```

ここで、メールの内容を説明する表題を入力して、Return キーを押します。次に、メールの本文を入力します。行の長さは短くおさえて、各行の終わりで Return キーを押します (Return キーを押す前ならば、Back Space キーで後退して再入力することによって文を修正できます)。

メールのサンプルを下記に示します (空白行は、Return キーで作成したものです)。

```
$ mailx rose@texas
Subject: to someone who really cares

Dear Rosey,

From the ends of your fingers
To the tip of your nose
You're a cool breeze in August
My sweet Texas Rose.

See you soon,

Rose
```

メールを送信するには、メールの最終行で Return キーを押してから Ctrl-D を押します。メールが送信されると、コマンドプロンプトの状態に戻ります。

自分宛てのメールの読み方

自分宛てのメールを読むには、mailx コマンドをもう一度入力します。画面に以下のようなメッセージが表示されます。

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992 Type ? for help.
`/var/mail/rose``: 2 messages 1 new
U 2 hal@uncertain Fri Feb 14 12:01 14/318 financial status
>N 1 rose@texas Mon Feb 17 08:12 21/453 to someone who
&
```

最初の行には実行中の mail のバージョンが表示され、2 行目には到着するメールが収納されるメールボックス (通常は `/var/mail/username` にある) の場所が表示さ

れます。この例では 3 行目に、自分に送ったメールに関する情報が表示されています。行の先頭の「N」(new を表す)は、これが「新しい」メールであることを示します。「U」(unread を表す)は、新しいメールが到着したのに、前回 mailx プログラムを実行したときには読まれなかったことを意味します。(この画面の内容については、117ページの「メールの読み方」で詳しく説明します。)各メールには、受信時に番号が割り当てられます。

ユーザ rose の自分宛てのメールには、1 というメール番号が付いています。

メールを読むには、アンパサンド (&) で示される mailx プロンプトに対して次のようにメールの番号を入力します。

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Type ? for help.
``/var/mail/rose``: 1 message 1 new
>N 1 rose@texas  Fri Jul 14 12:01 21/453 to someone who
& 1

To: rose@texas
From: rose@texas
Subject: to someone who really cares

Dear Rose,

From the ends of your fingers
To the tip of your nose
You're a cool breeze in August
My sweet Texas Rose.

See you soon,

Rose

&
```

mailx の終了

mailx を使い終えたら、q (quit) または x (exit) コマンドのどちらかを入力してプログラムを終了できます。

今回の例では、mailx プロンプトに対して q と入力して Return キーを押します。

```
& q
```

すると、次のようなメッセージが表示されます。

```
Saved one message in home_directory/mbox.
```

`home_directory` は、ホームディレクトリのパス名です。

`mailx` プロンプトに対して `q` と入力して `mailx` を終了した場合、`mailx` は読み終わったメールをメールボックスから削除してホームディレクトリ内の `mbox` ファイルに保存します。メールを変更したり削除した場合は、その内容も保存されます。

```
& x
```

`mailx` プロンプトに対して `x` と入力して **Return** キーを押した場合、読み終わったメールは `mbox` ファイルに移動されず、変更や削除の内容も保存されません。

メールの読み方

メールが存在する場合、ユーザがログインするたびに `mailx` は次のどちらかのメッセージを表示してその旨を知らせます。

```
You have mail
```

または

```
You have new mail
```

メールを読むには、コマンドプロンプトに対して `mailx` と入力し、**Return** キーを押します。メールがない場合は、次のメッセージが表示されます。

```
No mail for username
```

メールがある場合は、次のようなりストが表示されます。

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992 Type ? for help.
`'/var/mail/rose'`: 4 messages 1 new 2 unread
  1 rose@texas      Fri Feb 14 12:01 21/453 to someone who
  U 2 hank@fretful  Fri Feb 14 18:31 19/353 so lonely I
  U 3 farmer@freeway Sat Feb 15 10:22 24/557 looks like my
  >N 4 hoover@woofer Sun Feb 16 23:59 14/280 big old furry
&
```

最初の行には、`mailx` プログラム自身に関する情報 (バージョン番号と日付) とヘルプの参照方法 (`Type ? for help`) が表示されます。

次の行には、メールボックスの場所、受信されたメールの数とその状態が表示されます。

その次の行には、メールボックス内のメールが番号付きで表示されます。各行のコラムの内容は、左から右に以下のとおりです。

- 状態: メールの状態を新規 (N)、読まれていない (U)、読まれた (記号なし) で示します。行の先頭にある「>」は、最新のメールを示します。削除されたメールは、アスタリスク (*) が付いて表示されます。
- 番号: メールが受信された順序を示します。
- 送信者: メールを送信したユーザ名 (および通常はマシン名も) を示します。
- 時間: メールが送信された日付と時刻を示します。
- サイズ: メールの行数およびバイト数を示します。
- 表題: 送信者が `subject:` 行で入力したメールの表題が表示されます。

メールボックス内のメールの数が多い場合は、このリストにすべてのメールが表示されないこともあります。その場合は、次のように入力します。

- `z` (メールの見出しリストの次の 1 画面を表示)
- `h-` (メールの見出しリストの前の 1 画面を表示)
- `h` (任意の時点でメールの見出しリストを再表示)

メールボックス内の最新のメール (> 付きで表示されたもの) を参照するには、**Return** キーを押します。**Return** キーをもう一度押すと、次のメールが表示されます。リスト内の任意のメールを参照するには、その番号を入力して **Return** キーを押します。

メールの削除と削除の取り消し

メールを読み終えたら、`mbox` ファイルに保存しないで削除することもできます (デフォルトでは、`q` コマンドで `mailx` プログラムを終了すると `mbox` に保存されます)。

読み終わった最後のメールを削除するには、`mailx` プロンプトに対して `d` と入力します。メールボックスから特定のメールを削除するには、次のコマンドを実行します。

`d number`

たとえば、2 番目のメールを削除するには、`mailx` プログラム内から次のコマンドを実行します。

```
& d 2
```

一度に複数のメールを削除することもできます。1 番目と 3 番目のメールを削除するには、次のコマンドを実行します。

```
& d 1 3
```

一定範囲のメールを削除することもできます。たとえば 1 ~ 3 番目のメールを削除するには、次のコマンドを実行します。

```
& d 1-3
```

mailx を終了する前なら、メールボックスから削除したメールを復元 (削除の取り消し) できます。次の形式のコマンドを入力して Return キーを押します。

u number

たとえば、2 番目のメールを復元するには、次のコマンドを実行します。

```
& u 2
```

前回の削除コマンドを取り消すには、削除した直後に mailx プロンプトに対して *u* と入力します。たとえば、前回の削除コマンドが *d 2-5* の場合は、*u* と入力すると 2 ~ 5 番目のメールの削除が取り消されます。

q コマンドで mailx を終了すると、すべての削除操作が確定されます。つまり、削除したメールを復元することはできなくなります。ただし、*x* コマンドで mailx を終了すればメールボックスを以前の状態に保つことができます。前述のように、*x* コマンドで mailx を終了すると、読み終わったメールでも *U* が付けられて表示され、削除したメールも削除されなかったこととなります。

メールの印刷

メールをプリンタコマンドにパイプすることによって、メールを印刷できます。メールの印刷には、mailx プロンプトに対して次の形式のコマンドを実行します。

|*number* lp

| シンボルはパイプと呼ばれます。たとえば、2 番目のメールを印刷するには、次のように入力して **Return** キーを押します。

```
& |2 lp
```

メールの番号が指定されない場合は、現在のメールがプリンタにパイプされます。パイプについての詳細は、第 2 章の 39 ページの「コマンド出力のリダイレクトとパイプ」を参照してください。

メールの送信

mailx プログラムを使ってメールを送信する場合は、メールの受信者のログイン名が必要です。受信者が別のマシン上にいる場合は、そのユーザのマシン名も必要になります。これらの情報を調べるために、`who`、`finger`、`rusers` などのコマンドを使うことができます。

`who` コマンドを入力すると、ファイルサーバに現在ログインしている全ユーザのリストが表示されます。このリストには、ユーザのログイン名、その端末の型、ログインした日付と時間が表示されます。`who` のリストを次に例示します。

```
$ who
elmer    tty15    Feb 20 10:22
susan    tty04    Feb 20 10:37
stormy   tty07    Feb 20 11:49
hankw    tty06    Feb 20 12:02
```

`finger` コマンドを入力すると、`who` コマンドと同様の情報がより詳細に表示されます。システム管理者によるこのコマンドの設定状態によって、表示される情報の内容は異なります。たとえば、次のような情報が表示されます。

```
$ finger
Login    Name      TTY      Idle    When
elmer    Elmer Brown  tty15    43     Thu 10:22
susan    Susan Lake  tty04           Thu 10:37
stormy   Stormy Ball  tty07    12     Thu 11:49
hankw    Hank Wilson  tty06    22     Thu 12:02
```


`rusers` コマンドを入力すると、ローカルネットワークに現在ログインしているユーザに関する情報が表示されます。`rusers` コマンドの使い方についての詳細は、第 9 章を参照してください。

必要なユーザ情報を確認できたら、次の手順に従ってメールを送信します。

1. `mailx` コマンドの後に続けてユーザのアドレスを入力します。

```
$ mailx user@machine
```

`user` は受信者のログイン名、`machine` は受信者のマシン名です。

- `mailx` プログラムをすでに起動している場合は、`mailx` プロンプトに対して `m` の後に受信者のログイン名とマシン名を `@` で区切り入力します。

```
& m user@machine
```

- 同じメールを複数の受信者に送信する場合は、次の例のように各アドレスをスペースかコンマで区切ります。

```
$ mailx hank@fretful sally@dakota tex@twister
```

または

```
$ mailx hank@fretful,sally@dakota,tex@twister
```

2. **Return** キーを押すと、`mailx` プログラムはメールの表題を入力するよう `subject:` と表示してくるので、表題を入力してもう一度 **Return** キーを押します。
3. メール本文を入力します。改行する場合は、**Return** キーを押します。画面上で文が複数行に渡って表示されていても、**Return** キーを押していなければ改行したとは見なされません。

注 - メール各行のテキストの最大長は 256 文字です。この制限を超えると、画面が動かなくなります。その場合は、**Ctrl-C** を 2 回押してメールの入力を中止してください。

4. メールが入力が終わったら、**Return** キーを押して改行します。次に、**Ctrl-D** を押してメールを送信します。

送信できないメール

メールを送信するときに正しくないユーザアドレスを指定すると、システムの応答として次のようなメッセージが表示され、メールはメールボックスに戻されます。

```
user@machine...User unknown
```

次回、`mailx` コマンドを入力したときに、次のように表示されてメールが届かなかった旨が知らされます。

```
N 1 Mailer-Daemon Fri Jan 3 11:13 8/49 Returned mail: User unknown
```

また、メールが届かない場合は、ホームディレクトリ内の `dead.letter` というファイルにそのメールがコピーされます。

送信しないメールの取り消し

送信前ならば `Ctrl-C` を 2 回押すことによって、メールの入力を取り消すことができます。

カーボンコピーとブラインドカーボンコピーの追加

メールを送信する前に、「カーボンコピー」を `To:` で指定した受信者以外に送信するよう指定できます。また、「ブラインドカーボンコピー」も送信できます(メールの全受信者は、カーボンコピーの宛先アドレスを読み取ることができますが、ブラインドカーボンコピーのアドレスは読み取れません)。

自分自身にカーボンコピーやブラインドカーボンコピーを送信すれば、独自の記録として送信したメールのコピーを保管できます。

メールと一緒にカーボンコピーを送信するには、次の 3 つの方法があります。

- あらかじめテキストエディタを使ってホームディレクトリ内の `.mailrc` ファイルを編集し、次の行を挿入します。

```
set askcc
```

mailx プログラムは、表題の入力 (Subject:) プロンプトの下にカーボンコピープロンプト (Cc:) を表示するので、カーボンコピーの宛先となるユーザのアドレスを入力します。宛先が複数の場合は、アドレスをスペースまたはコンマで区切ります。

- メールの本文を入力した後 Ctrl-D を押す前に Return キーを押して改行し、次のコマンドを実行します。

```
~c address(es)
```

この方法によって1人または複数の受信者にカーボンコピーを送信する場合は、次の例のように各受信者のアドレスをスペースで区切ります。

```
~c hank@fretful george@lonesome stormy@snoozer
```

- Cc: プロンプトは、~h コマンドでも表示できます。~h コマンドは、メールのすべての見出し行を表示します。つまり1行ずつ、To:、Subject:、Cc:、Bcc: (ブラインドカーボンコピー) というプロンプトがすべて表示されます。その際に、各空白行は入力することができ、その入力した内容を修正することもできます。他のチルドコマンドと同様に、メールの本文作成中は常に改行してから~h コマンドを実行してください。

注 - ~c、~h などのチルドコマンドについての詳細は、この章で後述の 135ページの「チルドコマンド」を参照してください。

メールやファイルのコピーを挿入する

メールボックス内にある任意のメールのコピーを現在作成中のメールに挿入することができます。同様に、任意のテキストファイルのコピーも挿入できます。

メールの挿入

メールの本文作成中に他のメールを挿入するには、次の形式のコマンドを使います。

```
~m number
```

number は、挿入するメールの番号です。たとえば、メールボックスの 3 番目のメールのコピーを取り込んで新規のメールを作成し、それを他のユーザに送信する場合は、次の手順を実行します。

1. メールの本作文作成中に改行してから `~m 3` というコマンドを入力し、**Return** キーを押します。
2. Interpolating: 3 (continue) というメッセージが mailx によって表示されます。
3. 送信者の画面には **3** 番目のメールの内容は表示されませんが、受信者の画面には表示されます。(continue) の後に続けてメールを入力するか、取り込んだコピーをそのまま送信します。
4. 取り込んだコピーも含めたメール全体を表示するには、`~p` コマンドを入力します。

ファイルの挿入

メールの本作文作成中に任意のテキストファイルのコピーを挿入することもできます。そのコマンドの形式は次のとおりです。

`~r filename`

たとえば、`outline` というファイルを挿入するには、次のように入力します。

```
~r outline
```

メールへ応答する

メールに返事を出すには、mailx プロンプトに対して次のコマンドを実行します。

`r number`

メールの番号を入力しないと、現在のメールへの応答となります。たとえば、2 番目のメールの送信者に応答するには、次のコマンドを実行します。

```
& r 2
```

mailx は自動的に応答先のアドレスを指定し、Re: Subject: という行を表示します。この行には応答するメールの Subject: 行の内容が表示されます。この応答のメールは、通常のメールと同様に送信できます。

R コマンドは r コマンドの変形で、送信者だけでなく、応答するメールの受信者全員に対しても応答を送ります。不必要なメールが作成されるのを防ぐため、このコマンドは本当に必要な場合にだけ使うようにしてください。

注 - 前節で説明したのと同様の方法で、応答にもメールのコピーを挿入できます。応答するメールのコピーを挿入するには、メールの番号を指定しないで ~m コマンドを実行します。

メールの保存と読み取り

メールの送受信の他に、メールを保存しておき、そのメールを後で読むことができます。mailx では、通常のテキストファイルにメールを追加することによってメールを保存できます。また、フォルダと呼ばれる特殊ファイルにメールを追加することもできます。これらの方法について次に説明します。

mailx では、メールの保存とメールのコピーは区別されます。メールを保存すると、そのメールはメールボックスから削除されてファイルやフォルダに追加されます。一方、メールをコピーすると、メールはメールボックス内に残り、そのコピーがファイルやフォルダに追加されます。

メールをファイルに保存およびコピーする

メールをファイルに保存するには、mailx プロンプトに対して次の形式のコマンドを実行します。

s number filename

number は保存するメールの番号、*filename* はメールが保存されるファイル名です。たとえば、3 番目のメールを ~/notes/finance というファイルに保存するには、次のように入力します。

```
& s 3 ~/notes/finance
```

(パス名中の ~ は、各ユーザのホームディレクトリを示します。)

複数のメールを一度に同じファイルに保存することもできます。たとえば、3、5、6、7、8 番目のメールを ~/notes/finance に保存するには、次のように入力します。

```
& s 3 5-8 ~/notes/finance
```

指定したファイルが存在しない場合は、mailx によって作成されます。ファイルが存在する場合は、保存するメールはファイルの終わりに追加されます。

メールをファイルに保存すると、そのメールはメールボックスから削除されます。この場合は、保存されたメールの見出しにアスタリスク (*) が表示されます。

3 番目のメールをメールボックスに残したまま別のファイルに追加 (コピー) する場合は、次のように copy コマンドを入力します。

```
& c 3 ~/notes/finance
```

メールをフォルダに保存およびコピーする

フォルダにメールを保存またはコピーする場合は、ファイルのようなフルパス名を指定する必要はありません。フォルダは、フォルダディレクトリに保管されている特殊ファイルです。

フォルダにメールを保存またはコピーすると、同じディレクトリ内にメールが自動的に保管され、長いパス名を入力しなくてもそのディレクトリに簡単にアクセスできます。

フォルダディレクトリの設定

フォルダを使うには、まず最初にフォルダディレクトリ (フォルダ用のディレクトリ) を設定する必要があります。(フォルダディレクトリは通常、ホームディレクトリ内に設定します。) このため以下の 2 つの手順を実行します。

1. 最初に、mkdir コマンドを使ってディレクトリを作成します。

たとえば、Letters というフォルダディレクトリを設定するには、次のようなディレクトリを作成します。

```
$ mkdir Letters
```

- 次に、フォルダディレクトリのパスを設定するために、テキストエディタを使ってホームディレクトリ内の `.mailrc` ファイルを編集します。`.mailrc` ファイルには、`mailx` で使われるオプションが入っています。

次の例のように、新しく作成したフォルダディレクトリのフルパス名を指定するために、`set folder` 変数を編集します。

```
set folder=/home/austin/rose/Letters
```

C シェルの短縮名である `~` を使ってホームディレクトリを指定することもできます。

```
set folder=~/Letters
```

これでフォルダディレクトリは、フォルダに保存されるメールを受け取るよう設定されます。(`.mailrc` ファイルの変更内容は、次回 `mailx` を起動したときに有効になります。)

フォルダの指定

フォルダにメールを保存またはコピーするときは、ファイルの場合と同じコマンドを使います。ただし、フォルダ名の前にはパス名の代わりにプラス記号 (+) を付けます。+ は、フォルダをフォルダディレクトリ (ここでは `Letters`) に保存するよう `mailx` に指示します。

たとえば、3 番目のメールを `projects` というフォルダに保存するには、次のように入力します。

```
& s 3 +projects
```

mailx は、このコマンドを「3 番目のメールを ~/Letters/projects に保存する」という意味に解釈します。(指定のフォルダが存在しない場合は、mailx によって作成されます。)

メールをフォルダにコピーする場合は、次のように入力します。

```
& c 3 +projects
```

ファイルやフォルダにメールを直接送信する方法

メールのコピーは、任意のファイルやフォルダに直接送信できます。フォルダにコピーを送信する場合は、Cc: プロンプトまたは Bcc: プロンプトに対してフォルダ名(たとえば、projects)を入力するだけです。ファイルにコピーを送信する場合も同様ですが、ファイルのフルパス名を指定する必要があります。

ファイルやフォルダ内のメールを読み取る

ファイル内に保存されたメールを読み取るには、次の形式のコマンドを使います。

```
mailx -f filename
```

たとえば、~/memos/finance ファイル内のメールを読み取るには、次のように入力します。

```
$ mailx -f ~/memos/finance
```

フォルダに保存されたメールを読み取る場合も同じコマンドを使います。その場合は、フルパス名の代わりに + 記号を使います。たとえば、projects フォルダ内のメールを読み取るには、次のように入力します。

```
$ mailx -f +projects
```

このコマンドは、指定されたファイルやフォルダに対して mailx を起動し、それらのファイルやフォルダ内に存在するメールの見出しを表示します。読み取るメールを選択するには、mailx プロンプトに対してメールの番号を入力し、Return キーを押します。

mailx プログラム内からもフォルダを操作できます。フォルダのリストを参照するには、mailx プロンプトに対して次のように入力します。


```
& folders
```

メールボックスからフォルダに切り換えるには、次の形式のコマンドを使います。

```
& folder +foldername
```

メールボックスに戻るには、mailx プロンプトに対して次のように入力します。

```
& %
```

mailx を起動したなかで最後に使用したフォルダに戻るには、次のように入力します。

```
& #
```

mailx での vi の使い方

mailx の実行中に vi テキストエディタを使ってメールを作成できます。これにより、メールを送信する前にテキストの追加や削除、誤りの修正などができるようになります。vi に慣れていない場合は、第 6 章を参照してください。

mailx プログラムでは、vi の標準的なコマンドを使ってテキストの挿入、削除、変更ができます。

vi を使ってメールを作成するには、次の手順に従います。

1. mailx プロンプト (&) またはコマンドプロンプト (\$) に対して、アドレスを指定した mailx コマンドを実行します。
2. Subject: 行に表題を入力して、**Return** キーを押します。
3. 改行してから ~v コマンドを実行して vi を起動します。vi の画面に /tmp ディレクトリ内の空ファイルが表示されます。

4. vi のコマンドを使ってメールの本文を入力し編集します。
5. 編集が完了したら、:wq **Return** キーまたは ZZ コマンドで vi を終了します。
vi を終了すると、mailx によって (continue): というメッセージが表示されます。ここで、vi を使わずにメールに何か書き加えるか、または Ctrl-D を押してメールを送信します。

メールエイリアス

メールエイリアスとは、単一の名前によってまとめられたユーザ名の集合です。

メールを同一のユーザグループに繰り返し送信する場合は、メールエイリアスを使うと便利です。たとえば、hank@fretful、george@lonesome、sally@dakota という 3 ユーザに頻繁にメールを送信する場合は、amigos というメールエイリアスを作成できます。それ以降 amigos 宛てにメールを送信すると、この 3 ユーザがすべてそのメールを受信することになります。

メールエイリアスは、次の 2 箇所で設定できます。

- .mailrc ファイル
- /etc/aliases ファイル

.mailrc ファイルで設定されたメールエイリアスと /etc/aliases ファイルで設定されたメールエイリアスとは、その働きが異なります。それらの相違点は、この節の終わりにある表 7-1 にまとめてあります。

.mailrc ファイルでメールエイリアスを設定する

.mailrc ファイルでメールエイリアスを設定する場合は、次の点に注意してください。

- .mailrc 内のメールエイリアスはユーザ専用です。つまり、そのエイリアスを使えるのは、それを設定したユーザ自身に限られます。たとえば、あるユーザが .mailrc ファイル内で設定した amigos というメールエイリアスに対して他のユーザがメールを送信しようとする、[unknown user] というエラーメッセージが出力されます。

- メールが送信されると、`.mailrc` 内のメールエイリアスはそれが示す全ユーザ名に自動的に展開されます。たとえば、`amigos` にメールを送信すると、そのメールは `amigos` 内の全ユーザ名を受信者として指定した場合とまったく同様に送信されます。受信者側からは、メールの送信にメールエイリアスが使われたかどうかはわかりません。

`.mailrc` ファイルは、各ユーザのホームディレクトリ内にあります。このファイルには、`mailx` とメールツールの動作を制御する多数の設定が入っています。

`.mailrc` にメールエイリアスを追加するには、次のように入力します。

```
$ vi ~/.mailrc
```

注 - `.mailrc` ファイルの編集には任意のテキストエディタを使えます。上記の例は、`vi` エディタを使って編集する方法を示しています。`vi` に慣れていない場合は、第 6 章を参照してください。

各メールエイリアスは、`.mailrc` ファイル内で 1 行ごとに記述されています。つまり、複数の行に渡っていても、途中でキャリッジリターンが入っていることはありません。各メールエイリアスには、以下の情報をスペースで区切って記述しなければなりません。

- 「`alias`」というワード
- メールエイリアスの名前 (1 ワードのみ)
- メールエイリアスに含まれる受信者 (ログイン名とマシン名) で、それぞれスペースで区切る

下記の例は、2 つのメールエイリアスを示しています。最初の `amigos` には 3 ユーザ、2 番目の `softball` には 8 ユーザが含まれています。`softball` の行では、名前が画面上では複数行に渡って表示されていますが、Return キーが入力されていない限り問題はありません。

```
alias amigos hank@fretful george@lonesome sally@dakota
alias softball earl@woofer tex@twister elmer@farmhouse
jane@freeway hank@fretful jj@walker sally@dakota steve@hardway
```

`.mailrc` ファイル内のエイリアスに含まれるユーザにメールを送信する場合は、そのメールエイリアスをアドレスに指定し、マシン名は指定しないようにします。たとえば、次のようにして送信します。

```
$ mail amigos
Subject: Let's eat

Hey Compadres. How about
getting together for lunch on Friday?
Anyone interested?
```

受信者は、次のようなメールを受け取ります。To: の行が展開されていることに注意してください。

```
To: hank@fretful george@lonesome sally@dakota
Subject: Let's eat

Hey Compadres. How about getting together for lunch on Friday?
Anyone interested?
```

/etc/aliases でメールエイリアスを設定する

/etc/aliases ファイルでメールエイリアスを設定する場合は、次の点に注意してください。

- /etc/aliases 内のメールエイリアスは公共用です。つまり、/etc/aliases 内で `softball` というメールエイリアスを設定すれば、全ユーザがこのエイリアスを利用して `softball@your-machinename` 宛てにメールを送信できます。
- メールが送信されても、/etc/aliases 内のメールエイリアスは展開されません。たとえば、`softball@machinename` にメールを送信すると、エイリアスは展開されずそのままの形で受信されます。受信者側には、メールエイリアスの名前は分かりますが、そのエイリアスに含まれる他のユーザ名は分かりません。

/etc/aliases で作成されるメールエイリアスの形式は、.mailrc の形式とは若干異なります。/etc/aliases 内の各エイリアスは、次の形式でなければなりません。

- メールエイリアスの名前とその後のコロン (:)
- コンマで区切った受信者 (ログイン名とマシン名)。メールエイリアスの指定は複数行に渡ってもかまいません。

/etc/aliases ファイルを変更するには、最初に root ユーザになる必要があります。root がパスワードで保護されている場合は、root のパスワードが必要です。

次のコマンドを入力してシステムの root ユーザになります。

```
$ su
Password:
#
```

root ユーザになるとコマンドプロンプトが変わることに注意してください。

次の例は、メールエイリアス `softball@texas` をデフォルトの `/etc/aliases` ファイルに追加する方法を示しています。

```
# vi /etc/aliases
##
#Aliases can have any mix of upper and lower case on the left-
#hand side,
#but the right-hand side should be proper case (usually lower)
#
# >>>>>>>>>>The program ``newaliases`` will need to be run after
# >> NOTE >>this file is updated for any changes to
# >>>>>>>>>>show through to sendmail.
#
#@(#)aliases 1.10 89/01/20 SMI
##
# Following alias is required by the mail protocol, RFC 822
# Set it to the address of a HUMAN who deals with this system's
mail problems.
Postmaster: root

# Alias for mailer daemon; returned messages from our MAILER-
DAEMON
# should be routed to our local Postmaster.
MAILER-DAEMON: postmaster

# Aliases to handle mail to programs or files, eg news or vacation
# decode: ``|/usr/bin/uudecode``
nobody: /dev/null

# Sample aliases:
# Alias for distribution list, members specified here:
#staff:wnj,mosher,sam,ecc,mckusick,sklower,olson,rwh@ernie

# Alias for distribution list, members specified elsewhere:
#keyboards: :include:/usr/jfarrell/keyboards.list

# Alias for a person, so they can receive mail by several names:
#epa:eric

#####
# Local aliases below #
#####
softball@texas: earl@woofer
tex@twister elmer@farmhouse
jane@freeway hank@fretful jj@walker sally@dakota steve@hardway
:wq      (vi を終了して /etc/aliases ファイルを保存する)
# exit   (root ユーザを終了する)
$
```

注 - /etc/aliases ファイルの編集には任意のテキストエディタを使えます。上記の例は、vi エディタを使って編集する方法を示しています。vi に慣れていない場合は、第 6 章を参照してください。

/etc/aliases ファイル内のポンド記号 (#) は、その後のテキストやエイリアスの例が注釈であることを示しています。先頭にポンド記号が付いた行の情報は、実際のエイリアスとしては処理されません。

意図的にエイリアスを使わないようにする場合以外は、このファイルに追加するエイリアスの前にはポンド記号を付けないでください。

/etc/aliases 内のエイリアスに含まれるユーザに対してメールを送信する場合は、そのエイリアスとマシン名をメールのアドレスとして指定します。たとえば、次のようにして送信します。

```
$ mail softball@texas
Subject: Practice Today

Let's meet at the diamond
behind Building 4 after work tonight.
Goodness knows we can use the practice for Saturday's game! Be
there as early as you can.
```

受信者は、次のようなメールを受け取ります。

```
To: softball@texas
Subject: Practice Today

Let's meet at the diamond behind Building 4 after work tonight.
Goodness knows we can use the practice for Saturday's game! Be
there as early as you can.
```

To: の行が展開されていないことに注意してください。

/etc/aliases で設定されたメールエイリアスを使ってメールを送信する場合は、そのエイリアスが設定されているマシン名を必ず指定します。たとえば、マシン freeway 上で riders というメールエイリアスを設定した場合は、riders@freeway と指定してメールを送信しなければなりません。

.mailrc で設定されたメールエイリアスと /etc/aliases で設定されたメールエイリアスとの相違点を表 7-1 にまとめます。

表 7-1 .mailrc と /etc/aliases のメールエイリアス比較

	.mailrc	/etc/aliases
変更するために root になる必要性	なし	あり
メッセージを送信するときのアドレス	<i>alias</i>	<i>alias@machinename</i>
To: 行と Cc: 行で受信者の一覧がつくか	つく	つかない
名前をコンマで区切るか	区切らない	区切る
すべての名前を 1 行内に記述するか	する	しない
他のユーザがメールエイリアスを使えるか	使えない	使える

メールエイリアスについての詳細は、システムプロンプトに対して `man aliases` または `man addresses` と入力してマニュアルページを参照してください。

チルドコマンド

メールの作成中に、チルドコマンドを使ってさまざまな機能を実行できます。通常、チルドコマンドはチルド文字 (~) の後に 1 文字を付けたものです。次の表 7-2 では、便利なチルドコマンドをいくつか説明しています。これらのコマンドの一部はすでにこの章で紹介したものです。

注 - メールにチルド文字そのものを使いたい場合は、チルド文字を 2 つ入力します。これによりチルド文字が 1 つだけ表示されます。

表 7-2 チルドコマンド (mailx)

コマンド	機能
<code>~!command</code>	シェルコマンドを実行する
<code>~.</code>	ファイルの終わりを示すCtrl-Dと同じ機能
<code>~?</code>	チルドコマンドのヘルプメニューを表示する
<code>~b username</code>	ブラインドカーボンコピー (Bcc:) のリストにユーザ名を追加する
<code>~c username</code>	カーボンコピー (Cc:) のリストにユーザ名を追加する
<code>~d</code>	<code>dead.letter</code> ファイルの内容を現在作成中のメールに読み込む
<code>~f number</code>	指定されたメールを送信する。メールの読み取り中にメッセージを送信する場合のみ有効
<code>~h</code>	見出し行のプロンプト (Subject:, To:, Cc:, Bcc:) を表示する
<code>~m number</code>	指定されたメールを現在作成中のメールに挿入する。メールの読み取り中にメッセージを送信する場合にのみ有効
<code>~p</code>	画面に入力されたメッセージを表示する
<code>~q</code>	Ctrl-C を 2 回押すのと同じ機能。作成中のメッセージ本文が空でない場合は、その内容が <code>dead.letter</code> ファイルに保存される
<code>~r filename</code>	指定されたファイルを読み込む
<code>~s string</code>	表題の行 (Subject:) を <code>string</code> に変更する
<code>~t name</code>	指定された名前を To: リストに追加する
<code>~w filename</code>	現在作成中のメールから見出し行を取り除いたものを指定されたファイルに書き込む
<code>~x</code>	<code>mailx</code> を終了する。メッセージが <code>dead.letter</code> ファイルに保存されない点以外は <code>~q</code> と同じ

ヘルプコマンド: その他の mailx コマンド

mailx には、コマンドとその機能のリストを表示するヘルプコマンドが 2 種類あります。コマンドモードで使われるコマンドのリストを参照するには、mailx プロンプト (&) に対して ? を入力します。同様に、入力モード (メールの作成中など) では、同等のコマンドである ~? を入力すれば、チルドコマンド (チルドエスケープとも呼ばれる) のリストを参照できます。

マニュアルページでは、mailx コマンドをより技術的に詳しく説明していません。mailx のマニュアルページを参照するには、次のコマンドを入力します。

```
$ man mailx
```

または、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

プリンタの使い方

LP (ラインプリンタ用サブシステム) プリントサービスは SunOS の一部分です。このサービスは、印刷処理に使えるさまざまなツールを提供します。LP プリントサービスには多種多様の機能がありますが、このマニュアルではその一部分だけを取り扱います。この章では、LP プリントサービスを使って、次に示す基本的な印刷作業を実行するために必要な手順を説明します。

- 印刷要求の実行 (ファイルをプリンタに送信すること)
- プリンタの状態の確認
- 印刷要求の取り消し

LP プリントサービスについての詳細は、『Solaris のシステム管理 (第 2 巻)』を参照してください。

印刷要求の実行依頼

コマンドプロンプトからファイルを印刷するには、そのファイルを印刷するプリンタに対して lp コマンドを使って要求を送信します。要求が送信されると、LP プリントサービスはその要求を印刷待ち行列に登録し、要求 ID 番号を表示した後、シェルプロンプトを再表示します。

デフォルトプリンタへの印刷要求の実行

デフォルトプリンタ用に LP プリントサービスを設定している場合は、次のようにプリンタ名を入力しないで印刷要求を実行します。

```
$ lp filename
```

filename は印刷するファイル名です。

指定されたファイルはデフォルトプリンタの印刷待ち行列に登録され、要求 ID 番号が表示されます。

たとえば、`/etc/passwd` ファイルを印刷する (デフォルトプリンタはこの場合 `pinecone`) には、次のように入力します。

```
$ lp /etc/passwd
request id is pinecone-8 (1 file)
$
```

デフォルトプリンタの指定方法については、『*Solaris* のシステム管理 (第 2 巻)』を参照してください。

プリンタ名を指定した印刷要求

システム上でのデフォルトプリンタの指定の有無にかかわらず、接続されている任意のプリンタに対して印刷要求を実行できます。希望のプリンタに対して印刷要求を実行するには、次のコマンドを入力します。

```
$ lp -d printername filename
```

printername は希望のプリンタ名、*filename* は印刷するファイル名です。

指定されたファイルは宛先プリンタの印刷待ち行列に登録され、要求 ID 番号が表示されます。

たとえば、`/etc/passwd` ファイルを `acorn` というプリンタで印刷するには、次のように入力します。

```
$ lp -d acorn /etc/passwd
request id is acorn-9 (1 file)
$
```

システム上で接続されていないプリンタ (ここでは `thorn`) に対して印刷要求を行うと、次の例に示すようなメッセージが表示されます。

```
$ lp -d thorn /etc/passwd
UX:lp: ERROR: Destination "thorn" is unknown to the
        LP print service.
$
```

プリンタの接続方法については、『*Solaris* のシステム管理 (第 2 巻)』を参照してください。また、システムで利用できるプリンタについては、この章で後述の 143 ページの「プリンタの状態を確認する」を参照してください。

印刷完了通知の要求

大きいファイルの印刷要求を行う場合は、印刷処理が完了したときに LP プリントサービスから完了通知があると便利です。LP プリントサービスに対しては、次の 2 つの形式の完了通知を要求できます。

- 電子メール (email) メッセージの送信
- コンソールウィンドウに対するメッセージの書き込み

電子メールによる通知を要求するには、印刷要求時に `-m` オプションを指定します。

```
$ lp -m filename
```

コンソールウィンドウに対するメッセージの書き込みを要求する場合は、印刷要求時に `-w` オプションを指定します。

```
$ lp -w filename
```

filename は印刷するファイル名です。

複数部数の印刷

1つのファイルを複数部印刷することもできます。複数部数の印刷を要求すると、lp コマンドの `-n` オプションで指定された回数だけファイルが印刷されます。このような印刷要求は1つのプリントジョブと見なされ、ヘッダページは1ページだけ印刷されます。

複数部数の印刷を要求するには、次のコマンドを入力します。

```
$ lp -nnumber filename
```

number は任意の印刷部数、*filename* は印刷するファイル名です。

たとえば `/etc/passwd` ファイルを4部印刷するには、次のように入力します。

```
$ lp -n4 /etc/passwd
request id is pinecone-9 (1 file)
$
```

lp コマンドのオプション

lp コマンドのオプションを使って、用紙、文字セット、フィルタ、表題、バナーページの指定といった印刷要求をカスタマイズできます。lp コマンドで頻繁に使われるオプションを表8-1にまとめてあります。これらのオプションは、コマンド行で1つだけ指定しても任意の順序で組み合わせて指定してもかまいません。複数のオプションを組み合わせる場合は、各オプションの前にハイフン (-) を付け、各オプション間はスペースで区切って指定します。

たとえば、プリンタを指定し、電子メールによる通知を要求し、ファイルの印刷部数を6部と指定する場合は、次のように入力します。

```
$ lp -d printername -m -n6 filename
```

printername は指定するプリンタ名、*filename* は印刷するファイル名です。

表 8-1 頻繁に使われる lp オプション

オプション	説明
-d	印刷宛先。宛先のプリンタ名を指定する
-m	電子メール。ファイルの印刷が正常終了したときに、印刷要求元に電子メールを送信する
-n	部数。印刷の出力部数を指定する
-t	表題。印刷要求の表題 (バナーページ上にだけ印刷される) を指定する
-o nobanner	オプション。個別の印刷要求に関するバナーページの印刷を抑止する
-h	ヘッダ。印刷要求の各ページにヘッダを挿入する
-c	コピー。印刷する前にファイルをコピーする
-w	メッセージ書き込み。ファイルの印刷が正常終了したときに、コンソールウィンドウにメッセージを書き込む。

lp コマンドオプションの詳細なリストについては、『*SumOS* リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

プリンタの状態を確認する

LP プリントサービスの状態を調べるには、lpstat コマンドを使います。lpstat を使って、印刷待ち行列内のジョブ状態の検査、利用可能なプリンタの確認、ジョブを取り消すためのリクエスト ID の確認などを行うことができます。

印刷要求の状態を確認する

スプールされた印刷要求の状態を調べるには、次のコマンドを入力します。

```
$ lpstat
```

印刷要求を実行したファイルのリストが表示されます。

次の例では、fred というユーザがシステム pine 上のプリンタ pinecone で印刷するファイルが 1 つ待ち行列に登録されています。

```
$ lpstat
pinecone-10          fred          1261  Mar 12 17:34 on pine
$
```

lpstat コマンドは、印刷ジョブごとに 1 行ずつ要求 ID、その要求をスプールしたユーザ、バイト単位の出力サイズ、印刷要求を実行したシステムと日付と時刻に関する情報を表示します。

利用可能なプリンタを確認する

システムに接続されているプリンタを確認するには、次のように入力します。

```
$ lpstat -s
```

最初にスケジューラの状態が表示され、その後にデフォルトの宛先プリンタ、利用可能なシステムとプリンタのリストが表示されます。

次の例はスケジューラが実行中であり、デフォルトプリンタは pinecone、ネットワークプリンタは pinecone および acorn という 2 つのプリンタが利用できることを示しています。

```
$ lpstat -s
scheduler is running
system default destination: pinecone
system for pinecone: pine
system for acorn: oak
$
```

状態情報をすべて表示する

lpstat コマンドに `-t` オプションをつけると、LP プリントサービスの全状態を要約したリストが表示できます。

このリストを表示するには、次のように入力します。

```
$ lpstat -t
```

利用可能な状態情報がすべて出力されます。

次の例では、印刷待ち行列内にジョブが存在しません。ファイルが印刷のためにプールされると、それらの印刷要求の状態も表示されます。

```
$ lpstat -t
scheduler is running
system default destination: pinecone
system for acorn: oak
pinecone accepting requests since Wed Jan 2 18:20:10 PST 1991
acorn accepting requests since Mon Mar 4 15:53:47 PST 1991
printer pinecone is idle. enabled since Wed Jan 2 18:20:22 PST
1991. available.

printer acorn is idle. enabled since Mon Mar 4 15:53:44 PST 1991.
available.
$
```

プリンタの状態を表示する

lpstat コマンドに -p オプションをつけると、個々のプリンタに関する状態情報が表示できます。このオプションは、プリンタの活動状況 (アクティブまたはアイドル)、使用可能または不可能になった時間、印刷要求の受付の可否などを表示します。

システム上の全プリンタに関する状態情報を表示するには、次のように入力します。

```
$ lpstat -p
```

次の例は、2つのプリンタがアイドル状態、使用可能、印刷要求受付可能であることを示しています。これらのプリンタの印刷待ち行列にジョブが存在すれば、そのジョブ情報も表示されます。

```
$ lpstat -p
printer pinecone is idle. enabled since Wed Jan 2 18:20:22 PST
1991. available.
printer acorn is idle. enabled since Mon Mar 4 15:53:44 PST 1991.
available.
$
```

個々のプリンタに関する状態情報を表示するには、次のようにプリンタ名を指定します。

```
$ lpstat -p printername
```

printername は指定するプリンタ名です。

プリンタの特性を表示する

接続しているプリンタのすべての特性を参照する場合は、lpstat コマンドに -p オプションと -l (「long」を表す) オプションをつけます。このコマンドは、プリンタの型 (Printer type) とファイルの種類 (Content type) を調べる際に特に便利です。

システム上のすべてのプリンタの特性を表示するには、次のように入力します。

```
$ lpstat -p -l
```

各プリンタについて LP プリントサービスで使われる全構成情報が表示されます。

次の例では、プリンタ `pinecone` に関するフィールドのうち、プリンタの型とファイルの種類以外はすべて空白になっています。

```
$ lpstat -p pinecone -l
printer pinecone is idle. enabled since Wed Jan 2 18:20:22 PST
1991. available.
  Content types: PS
  Printer types: PS
  Description:
  Users allowed:
    (all)
  Forms allowed:
    (none)
  Banner not required
  Character sets:
    (none)
  Default pitch:
  Default page size:
$
```

lpstat コマンドのオプション

lpstat コマンドを使うと、プリンタの状態に関するさまざまな情報を表示できます。lpstat コマンドで頻繁に使われるオプションを表 8-2 にまとめてあります。これらのオプションは、コマンド行で 1 つだけ指定しても、任意の順序で組み合わせ

で指定してもかまいません。複数のオプションを組み合わせる場合は、各オプションの前にハイフン (-) を付け、各オプション間はスペースで区切って指定します。

たとえば、個々のプリンタの状態に関する詳細なリストを表示する場合は、次のように入力します。

```
$ lpstat -p printername -l
```

printername は、状態を調べるプリンタ名です。

表 8-2 頻繁に使われる lpstat オプション

オプション	説明
-a	受付状況。印刷宛先プリンタの要求受付状況を表示する
-c	クラス。プリンタクラスとそのメンバを表示する
-d	宛先。デフォルトの印刷宛先プリンタを表示する
-f	書式。印刷書式を表示する
-o	出力。出力状態を表示する
-p [<i>list</i>] [-D] [-l]	プリンタ指定・説明・詳細リスト。プリンタの状態を表示する
-r	要求。要求スケジューラの状態を表示する
-R	印刷待ち行列内のジョブの位置を表示する
-s	状態情報。要約した状態情報を表示する
-S	文字セット。文字セットを表示する
-u [<i>username</i>]	ユーザ。要求の状態をユーザごとに表示する
-v	プリンタのデバイス名を表示する

lpstat コマンドオプションの詳細なリストについては、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

印刷要求の取り消し

印刷要求が待ち行列内にあるときや印刷実行中に要求を取り消すには、`cancel` コマンドを使います。要求を取り消す場合は、そのリクエスト ID が必要になります。リクエスト ID は必ず、プリンタ名、ハイフン、印刷要求番号から構成されています。印刷要求を実行すると、要求 ID が表示されます。このリクエスト ID を忘れた場合は、`lpstat` コマンドを入力して **Return** キーを押すと確認できます。印刷要求を取り消せるのは、その要求を実行したユーザ、あるいは `root` または `lp` としてログインしたユーザだけです。

ID 番号による印刷要求の取り消し

指定の ID 番号を使って印刷要求を取り消すときには、次のように入力します。

```
$ cancel requestid
```

`requestid` は、指定する印刷要求の ID 番号 (`printername-number` からなる) です。

要求が取り消される旨のメッセージが表示され、印刷待ち行列内の次のジョブの印刷が開始されます。

次の例では、2 つの印刷要求が取り消されています。

```
$ cancel pinecone-3 pinecone4
request 'pinecone-3' cancelled
request 'pinecone-4' cancelled
$
```

プリンタ名による印刷要求の取り消し

リクエスト ID の代わりにプリンタ名を入力しても、現在印刷中のジョブ (自分で実行依頼したものに限る) を取り消すことができます。

```
$ cancel printername
```

`printername` は、ユーザが要求を送信したプリンタ名です。

要求が取り消される旨のメッセージが表示され、印刷待ち行列内の次のジョブの印刷が開始されます。

次の例では、pinecone というプリンタで現在印刷中の印刷要求が取り消されています。

```
$ cancel pinecone
request ``pinecone-3`` cancelled
$
```

システム管理者は root または lp としてログインし、プリンタ名を cancel コマンドの引数に指定することによって、現在印刷中の任意の要求を取り消すことができます。

ネットワークの使い方

ネットワークとは、相互に通信できるように設定されたコンピュータのグループです。使っているマシンがネットワークの一部である場合は、自分のマシン(ローカルマシン)にログインしたままでネットワーク上の他のマシン(リモートマシン)の資源を利用できます。ユーザは、他のマシンに影響を及ぼすリモートコマンドを自分のマシンから実行したり、他のマシンにログインしたりすることができます。

この章では、次の事項について説明します。

- ネットワークの一般概念
- リモートマシンにログインする方法
- リモートマシンからファイルをコピーする方法
- リモートマシン上でコマンドを実行する方法
- リモートマシンに関する状態情報を取得する方法

使っているマシンがネットワークに接続されていない場合は、この章で説明する内容は現在の環境には適用できません。ただし、この章を一通り読めば、ネットワークの利点を大まかに理解することができます。

ネットワークの一般概念

マシン間をネットワークで接続することによって、相互に情報を伝送できます。一般にネットワークは、通常おおよそ 300 メートル以内の狭い地域をカバーするローカルエリアネットワーク (LAN)、数千キロメートル以上の地域をカバーできる

広域ネットワーク (WAN)、それらの中間のキャンパスエリアネットワーク (CAN) に分類できます。

いくつかのネットワークが結びついて構成されているネットワークは、インターネットと呼ばれます。たとえば、ビル内部のローカルなネットワークの一部であるマシンが、同時にそのローカルなネットワークと国中にまたがる同種類のネットワークとを接続するインターネットの一部になることもあります。通常、ネットワークとインターネットとの違いはユーザの目には触れないため、このマニュアルでは「ネットワーク」という用語をネットワークとインターネットの両方の意味で使います。

ネットワークに接続されたマシンは、ネットワークプロトコル (共通のネットワーク言語) を使って通信を行い、適切な場所に情報を伝送します。リレーとも呼ばれるインターネットプロトコルは、ネットワークを相互にリンクするために使われます。

リモートログイン (rlogin)

rlogin コマンドを使って、ネットワーク上の他の UNIX マシンにログインできます。

リモートマシンにログインするには、次のように入力します。

```
$ rlogin machinename
```

machinename はリモートマシン名です。

パスワードの入力を促すプロンプトが表示されたら、リモートマシン上でのパスワードを入力して Return キーを押します。ローカルマシン名がリモートマシン上の /etc/hosts.equiv ファイル内に記述されていれば、リモートマシンはそのローカルマシン名を「信用」していることになり、パスワードの入力を要求しません。


```
$ rlogin lonesome
Password: (パスワードを入力する)
Last login: Mon Jan 6 09:37:55 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
(以下のコマンドは lonesome 上で実行される)
$ pwd
/home/keithp
$ logout
Connection closed.
$
```

ホームディレクトリなしの rlogin

上記の例では、pwd コマンドで示されるように、keithp というユーザがマシン lonesome にログインし、そのマシン上の /home/keithp ディレクトリに移動しています。自分のホームディレクトリが存在しないリモートマシンにログインした場合は、リモートマシン上にホームディレクトリがないというメッセージが rlogin によって表示され、そのマシンのルートディレクトリ (/) に移動します。

```
$ rlogin fretful
Password:
No directory! Logging in with home=/
Last login: Fri Jan 3 10:21:59 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
(以下のコマンドは fretful 上で実行される)
$ pwd
/
$ logout
Connection closed.
$
```

現在のログイン名以外での rlogin

現在のログイン名以外でリモートマシンにログインする場合があります。たとえば、他のユーザのマシン上で (他のユーザのログイン名を使って) 作業をしているときに、自分のマシンに自分のログイン名でログインする場合などです。このような場合は、rlogin の -l オプションを使います。コマンド構文は次のとおりです。

```
rlogin machinename -l username
```

次の例は、マシン blue のユーザ keithp が他のマシン lonesome にユーザ earl としてログインする方法を示しています。

```
$ rlogin lonesome -l earl
Password:
Last login: Wed Jan 8 07:12:25 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
(以下のコマンドは lonesome 上で実行される)
$ pwd
/home/earl
$ logout
Connection closed.
$
```

現在のログイン名以外でリモートマシンにログインすると、指定したユーザのホームディレクトリに移動します。

未登録のマシンに対する rlogin

名前が認識できないリモートマシンにログインしようとする、rlogin はホスト名データベースの検索に失敗し、次のようなメッセージを表示します。

```
$ rlogin stranger
stranger: unknown host
$
```

rlogin セッションの中止

通常は作業セッションの最後に logout と入力して rlogin セッションを終了します。通常の方法でセッションを終了できない場合は、行の先頭にチルド文字とピリオド (~.) を入力することによって rlogin セッションを中止できます。これによりリモートマシンとのログインセッションは中止され、ユーザは自分のマシンに戻ります。

あるリモートマシンからそれ以外のリモートマシンにアクセスすることによって、複数のマシンにログインしている状態で、~. を使ってそれらのマシンのどれかとのセッションを中止した場合も、最初にログインした自分のマシンに戻ります。

```
$ rlogin dakota
Password:
Last login: Fri Jan 10 09:14:43 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
(以下のコマンドは dakota 上で実行される)
$ ~. (~. は画面上に表示されないことがある)
Connection closed.
$
```

1つ前の中間セッションに戻りたい場合は、次のように2つのチルド文字の後にピリオドを付けたコマンド(~~.)を使います。

```
$ rlogin lonesome
Password:
Last login: Tue Jan 7 08:12:49 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
(以下のコマンドは lonesome 上で実行される)
$ rlogin dakota
Password:
Last login: Tue Jan 7 10:17:40 from lonesome
Sun Microsystems, Inc. SunOS 5.1 October 1992
(以下のコマンドは dakota 上で実行される)
$ ~~. (~~. は画面上に表示されないことがある)
Connection closed.
$
```

rlogin セッションの中断

rlogin セッションを一時的に中断して後でまた戻りたい場合は、チルド文字(~)の後に Ctrl-Z を入力します。rlogin セッションは中断されたプロセスになり、ユーザはログイン元のマシンに戻ります。

rlogin セッションを再開するには、fg と入力します。fg の代わりに、パーセント記号(%)の後に中断されたプロセスのプロセス番号を入力することも可能です。(プロセス番号が指定されない場合に % で使われるデフォルトは、最後に中断されたプロセスのプロセス番号です。)

```
$ rlogin lonesome
Password:
Last login: Tue Jan 7 08:12:49 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
(以下のコマンドは lonesome 上で実行される)
~^Z (~^Z は画面上に表示されないことがある)
Stopped
(以下のコマンドは blue 上で実行される)
$ pwd
/home/keithp
$ %
rlogin lonesome

(以下のコマンドは lonesome 上で実行される)
$ logout
Connection closed.
$
```

~~. による rlogin セッションの中止と同様に、2つのチルド文字の後に Ctrl-Z (~~^Z) を入力すると、現在の rlogin セッションが中断されて1つ前の中間セッションに戻ります。

ログイン状態の確認 (who am i)

各種リモートマシンに対して別々のログイン名でログインしている場合、ログイン状態を正確に把握することが必要になる場合があります。その場合、who am i と入力することによって、現在ログインしているマシン名と現在のユーザ名を表示できます。

rlogin についての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

ファイルのリモートコピー (rcp)

rcp コマンドを使って、異なるマシン間でファイルをコピーできます。rcp は、リモートマシン上の /etc/hosts.equiv ファイルと /etc/passwd ファイルを参照して、ユーザがアクセス権を持っているかどうか確認します。rcp の構文は、cp コマンドの構文と似ています。

注 - 異なるマシン間でサブディレクトリとその内容をコピーするには、rcp -r を使います。

リモートマシンからローカルマシンへのコピー

リモートマシンからローカルマシンにファイルをコピーするには、次の構文を使います。

`rcp machinename:source destination`

machinename はリモートマシン名、*source* はコピーするファイル名、*destination* はコピーしたファイルが格納されるローカルマシン上のパス名です。

次の例は、リモートマシン dakota のファイル /home/dakota/doc/letter を、ローカルマシン blue の /tmp ディレクトリにコピーする方法を示しています。

```
$ rcp dakota:/home/dakota/doc/letter /tmp
$
```

rcp では、さまざまな省略名と構文を組み合わせて使うことができます。たとえば、リモートマシン fretful 上のユーザ hank のホームディレクトリにある .doc

で終わる全ファイルを、ローカルマシン blue のカレントディレクトリにコピーするには、次のように入力します。

```
$ rcp fretful:~hank/*.doc .
$
```

ローカルマシンからリモートマシンへのコピー

ローカルマシンからリモートマシンにファイルをコピーするには、前記の構文を逆にした次の構文を使います。

rcp source machinename:destination

source はコピーするファイル名、*machinename* はリモートマシン名、*destination* はコピーしたファイルが格納されるリモートマシン上のパス名です。

次の例は、ローカルマシン上の ~/usa/texas ディレクトリの austin というファイルを、リモートマシン fretful の ~hank/cities ディレクトリにコピーする方法を示しています (~ はローカルマシン上のホームディレクトリ、~hank はユーザ hank のホームディレクトリであることを注意してください)。

```
$ rcp ~/usa/texas/austin fretful:~hank/cities
$
```

rcp コマンドとそのオプションについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

コマンドのリモート実行 (rsh)

rsh コマンド (リモートシェルコマンド) を使うと、形式的にログインしなくてもリモートマシン上で1つのコマンドを実行できます。リモートマシンで1つのコマンドを実行するだけでよい場合は、rsh によって時間を節約できます。

リモートマシン上で1つのコマンドを実行するには、次のように入力します。

rsh machinename command

次の例は、リモートマシン lonesome 上の /home/lonesome/guitar ディレクトリの内容を表示する方法を示しています。

```
$ rsh lonesome ls /home/lonesome/guitar
collings      gibson      santacruz
fender        martin      taylor
$
```

rlogin や rcp と同様に、rsh は、リモートマシンの /etc/hosts.equiv ファイルと /etc/passwd ファイルを参照して、ユーザがアクセス権を持っているかどうか確認します。

rsh コマンドとそのオプションについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

ユーザ情報の参照 (rusers)

rusers コマンド (リモートユーザ) を使って、ネットワーク上の他のマシンにログインしているユーザの情報を表示できます。rusers コマンドを引数なしで入力すると、ネットワーク上の各マシン名とそのマシンにログインしているユーザの情報が表示されます。

```
$ rusers
aspen      susan
blue       keithp
dakota     sally
farmhouse  elmer
freeway    lindab    johnj     karenm
fretful    hank
lonesome   george
twister    tex
$
```

マシン freeway には現在 3 人の異なるユーザがログインしています。

特定のリモートマシンに関する情報を表示するには、次のように rusers コマンドの後にマシン名を指定します。

```
$ rusers freeway
freeway    lindab    johnj     karenm
$
```

rusers コマンドで -l オプションを指定すると、詳細な情報が表示されます。表示される情報は、ユーザ名、マシン名と端末名、各ユーザがログインした時刻、

各ユーザのアイドル時間 (1 分以上の場合のみ)、各ユーザのログイン元のマシン名 (存在する場合) です。

```
$ rusers -l freeway
lindab    freeway:ttyd8    Feb 10 08:12    5:29
johnj     freeway:console  Feb 10 09:16
karenm    freeway:ttyp0    Feb 10 11:56    36
$
```

-l オプションは、マシン名を指定しなくても使えます。

rusers コマンドとそのオプションについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

動作環境のカスタマイズ

SunOS オペレーティングシステムでは、多数の動作環境をユーザが制御したり調整したりできます。システムの初期設定ファイル内の環境変数を変更することによって制御や調整を行います。ユーザがログインすると、初期設定ファイルが読み取られ、その中の環境変数を使ってシステムが構成されます。環境変数を適切に設定することによって、独自の作業をより効率的で簡単に実行できるようにシステムを「カスタマイズ」できます。

この章では、次の事項について説明します。

- 初期設定ファイルを変更し頻繁に使われる環境変数を設定してシステムをカスタマイズする
- SunOS コマンドの別名の設定方法
- システムプロンプトの変更方法
- デフォルトのファイルアクセス権の設定方法
- OpenWindows で使うフォントのカスタマイズ方法
- カラーモニターの調整方法

初期設定ファイル

システムの構成に使われる初期設定ファイルは、システムのインストール時にシステム管理者がデフォルトシェルとして指定したシェルの種類によって異なります。SunOS のデフォルトシェルは Bourne シェルですが、C シェルや Korn シェルも利用できます。これらのシェルには、独自の初期設定ファイル (1 つ以上) があります。

ログインシェルとも呼ばれるデフォルトシェルの種類が不明な場合は、次の手順を実行します。

1. `echo $SHELL` と入力します。

```
$ echo $SHELL
/bin/sh
```

2. 上記のコマンドの出力結果によって次のように判定します。

- `/bin/sh` – Bourne シェル
- `/bin/csh` – C シェル
- `/bin/ksh` – Korn シェル

最初にシステムにログインすると、使っているシェルの種類にかかわらず、通常はシステムプロファイルの `/etc/profile` ファイルが実行されます。このファイルの所有者は通常はシステム管理者であり、その他のユーザは読み取りだけ許可されています (書き込みはできません)。

システムプロファイルの実行が完了すると、次にユーザプロファイルが実行されます。ユーザプロファイルは、各ユーザ独自の動作環境を定義する 1 つ以上の初期設定ファイルです。たとえば、OpenWindows 環境では、シェルツールやコマンドツールのウィンドウを新たに起動するたびに、これらのファイルがチェックされます。

デフォルトシェルとして設定されているシェルの種類によって、ユーザプロファイルは次のどちらかになります。

- `.profile` (Bourne シェルまたは Korn シェルの場合)
- `.login` と `.cshrc` (C シェルの場合)

ユーザプロファイルは各ユーザのホームディレクトリ内にあり、独自の動作環境を構成するために使います。

環境変数

システム環境は、初期設定ファイルで定義された複数の環境変数を使って構成されます。現在の作業環境を一時的に変更する場合は、コマンドプロンプトから直接コマンドを入力します。ただし、動作環境の変更を一時的ではなく、常時有効にし

たい場合は、`.profile`、`.login`、`.cshrc` の各ファイルの中の環境変数の設定を変更する必要があります。

次のように、システムで現在設定されている環境変数を表示できます。

- ◆ `env` コマンドを入力して **Return** キーを押します。

```
$ env
HISTORY=100
HOME=/home/texas/keith
HZ=100
LOGNAME=keith
MAIL=/var/mail/keith
MANSECTS=\1:1m:1c:1f:1s:1b:2:\3:3c:3i:3n:3m:3k:3g:3e:3x11:3xt:3
w:3b:9:4:5:7:8
PATH=/usr/bin
SHELL=/bin/sh
TERM=sun
TZ=EST5EDT
```

注 - `env` コマンドを使って、`SHELL` 環境変数で指示されるログインシェルの確認ができます。上の例では、ログインシェルは `/bin/sh` (Bourne シェルの場合) です。

ユーザプロファイル

この節では、一般的によく使われる環境変数について説明します。これらの環境変数の大部分は、すでにユーザプロファイルに入っています。すでに述べたように、ユーザプロファイルのファイル (Bourne シェルまたは Korn シェルでは `.profile`、C シェルでは `.cshrc`) は、各ユーザのホームディレクトリにあります。

注 - 隠しファイル (ドットファイル) は、`ls -la` と入力すると表示できます。

次のリストは、ユーザプロファイルで使える環境変数の一部です。環境変数を定義する構文は、シェルによって異なります。

- `CDPATH` - フルパス名を指定しないで一意のディレクトリ名を入力したときに検索されるディレクトリを指定します。
- `HISTORY` - `history` コマンドで利用できるコマンドの数を指定します。(hisotry コマンドは、C シェルだけで使用できます。)

- HOME – 各ユーザのホームディレクトリの絶対パス名を指定します。cd コマンドを引数なしで入力したときに移動先となるディレクトリは、この情報に基づいて決められます。
 - LANG – ロケールの言語を指定します。有効な値は、英語 (C)、日本 (ja)、ドイツ (de)、フランス (fr)、スウェーデン (sv)、イタリア (it) などです。
 - LOGNAME – ユーザのログイン名を指定します。LOGNAME 変数のデフォルト値は、passwd データベースで指定されたログイン名となるようログイン処理の過程で自動的に設定されます。passwd データベースについての詳細は、『Solaris のシステム管理 (第 1 巻)』を参照してください。
 - LPDEST – デフォルトプリンタを指定します。
 - MAIL – メールボックスのパス名を指定します。通常、メールボックスは /var/mail/username ディレクトリにあります (username は各ユーザのログイン名)。メールボックスについての詳細は、第 7 章を参照してください。
 - MANSECTS – オンラインのマニュアルページで利用できるセクションを設定します。
 - PATH – コマンドを入力したときに、実行可能プログラムが検索されるディレクトリを順番に指定します。適切なディレクトリが検索パスにない場合は、それを PATH 変数に追加するか、コマンドを入力するときにフルパス名を指定する必要があります。
- PATH 変数のデフォルト値は、.profile ファイル (Bourne シェルまたは Korn シェル) または .cshrc ファイル (C シェル) で指定された値となるようにログイン処理の過程で自動的に設定されます。
- PS1 – コマンドプロンプトを指定します。Bourne シェルと Korn シェルのデフォルトプロンプトは、ドル記号 (\$) です。また、C シェルのデフォルトプロンプトは、パーセント記号 (%) です。root のデフォルトプロンプトは、どのシェルでもポンド記号 (#) です。
 - SHELL – vi や他のツールで使われるシェルを指定します。
 - TERMINFO – terminfo データベースに追加された、デフォルトでない端末のパス名を指定します。terminfo データベース内のデフォルト端末については、この変数を設定する必要はありません。terminfo データベースについての詳細は、『Solaris のシステム管理 (第 2 巻)』を参照してください。
 - TERM – 現在使っている端末を指定します。エディタを実行するときは、TERM 変数で定義された名前と同じ名前のファイルが検索されます。その場合、まず最初に TERMINFO 変数で指定されるパスが検索され (TERMINFO が定義されている場

合)、次にデフォルトディレクトリの `/usr/share/lib/terminfo` が検索されて、端末の特性が決定されます。TERM 変数で定義されたファイルが見つからない場合は、その端末はダム端末と認識されます。

- TZ - システムクロックのタイムゾーンを指定します。

PATH 環境変数の設定

PATH 環境変数は、SunOS ディレクトリ階層内でコマンドを検索するために使われます。PATH 環境変数を設定すると、コマンド名を入力したときに特定のディレクトリが常に検索されるようになります。

たとえば、PATH 変数を設定していない場合にファイルをコピーする場合、`/usr/bin/cp` のように、コマンドのフルパス名を入力しなければなりません。これに対して、`/usr/bin` ディレクトリが PATH 変数に含まれるよう設定しておけば、`cp` と入力するだけでコマンドを実行できます。これは、PATH 変数で指定された各ディレクトリ内で `cp` コマンドが検索され、見つかった時点で実行されるためです。頻繁に使われる SunOS コマンドのディレクトリを PATH 変数に設定しておけば、作業効率を大幅に向上させることができます。

Bourne シェルと Korn シェルについては、次の構文を使ってホームディレクトリの `.profile` ファイル内で PATH 変数を設定します。

```
PATH=./usr/bin:home/bin
```

`home` はホームディレクトリのパス名です。

C シェルについては、次の構文を使ってホームディレクトリの `.cshrc` ファイル内で PATH 変数を設定します。

```
set path=(. /usr/bin home/bin)
```

`home` はホームディレクトリのパス名です。

注 - C シェルでは、短縮名の `~` を使ってホームディレクトリのパス名を表すことができます。

C シェルの環境で PATH 変数を変更した場合は、ログアウトしなくても `source` コマンドを実行すれば、現在のウィンドウ内で PATH の設定を有効にできます。

```
example% source .cshrc
```

Bourne シェルまたは Korn シェルを使っている場合は、ログアウトしなくても次のコマンドを実行すれば、現在のウィンドウ内で PATH の設定を有効にできます。

```
$ . .profile
```

別名 (C シェルのみ)

別名 (エイリアス) は、頻繁に入力するコマンドの便利な短縮名です。たとえば、削除コマンド (rm) のデフォルト設定では、ファイルを削除する前に確認を求められませんが、入力を誤って必要なファイルを削除してしまう危険性があるため、この設定では不都合な場合があります。C シェルでは、別名変数を使ってこの設定を変更できます。別名を使うには、.cshrc ファイルに次のような行を追加します。

```
alias rm 'rm -i'
```

.cshrc ファイルにこの行があれば、rm と入力するだけで rm -i (対話形式の rm コマンド) を入力したのと同じことになります。したがって、ファイルが削除される前に常に確認を求められるようになります。(上記の例で、rm -i の両側の引用符 (') は、rm と -i の間に空白を挿入するために必要です。この引用符がないと、C シェルは空白の後のテキストを正しく解釈できません。)

.cshrc ファイルに対して行なった変更を、ログアウトしなくても現在のウィンドウ内で有効にするには、source コマンドを使います。source コマンドを実行すると、現在の .cshrc ファイルが読み取られてその中のコマンドが実行されます。

```
example% source .cshrc
```

コマンドプロンプトの変更

コマンドプロンプトの変更を使う構文は、Bourne、Korn、C の各シェルによって異なります。

Bourne シェルと Korn シェルの場合

Bourne シェルまたは Korn シェルでは、PS1 コマンドを使ってコマンドプロンプトを変更することができます。次の 3 つの例を参照してください。

```
PS1=": "  
PS1="'hostname': "  
PS1="'hostname'{'id'}": "
```

- 最初の例では、コロン (:) の後に空白が挿入されたプロンプトを設定しています。
- 2 番目の例では、マシン名の後にコロンと空白が挿入されたプロンプトを設定しています。
- 3 番目の例では、マシン名の後に中括弧 {} で囲んだログイン名、コロンと空白が挿入されたプロンプトを設定しています。

上記の例のどれかを入力して現在のコマンドプロンプトを変更してみてください。変更したプロンプトは、再度それを変更するかログアウトするまではそのままになります。

プロンプトを恒久的に変更するには、上記の例の 1 つ (または自分で作成したプロンプト) を `.profile` ファイルに追加します。これにより、ログインしたりシェルを再起動するたびに、指定したプロンプトが表示されるようになります。

C シェルの場合

C シェルでは、`set prompt` コマンドを使ってコマンドプロンプトを変更できます。次の 3 つの例を参照してください。

```
set prompt="% "  
set prompt="'hostname'\!: "  
set prompt="'hostname'{'id'}": "
```

- 最初の例では、パーセント記号 (%) の後に空白が挿入されたプロンプトを設定しています。
- 2 番目の例では、マシン名の後にコマンドの履歴番号が付き、それにコロンと空白が挿入されたプロンプト (「hostname1:」, 「hostname2:」など) を設定しています。
- 3 番目の例では、マシン名の後に中括弧 {} で囲んだログイン名、コロン、空白が挿入されたプロンプトを設定しています。

上記の例のいずれかを入力して現在のコマンドプロンプトを変更してみてください。変更したプロンプトは、再度それを変更するかログアウトするまではそのままになります。

プロンプトを恒久的に変更するには、上記の例の1つ(または自分で作成したプロンプト)を `.cshrc` ファイルに追加します。これにより、ログインしたりシェルを再起動するたびに、指定したプロンプトが表示されるようになります。

その他の便利な変数

`.profile` ファイルや `.cshrc` ファイルでは、上記の他に多くの変数を設定できます。変数についての詳細は、『*SunOS* リファレンスマニュアル (1): ユーザーコマンド』を参照してください。頻繁に使われる変数については、以下で簡単に説明します。

`cp` コマンドを使ってファイルをコピーするときに、誤ってファイルを上書きするのを防ぐには、`set noclobber` を使います。この変数は C シェルだけに適用できます。次の行を `.cshrc` ファイルに入力します。

```
set noclobber
```

履歴リストに保存するコマンドの個数を設定するには、`set history` を使います。`history` コマンドは、以前に入力したコマンドを参照する場合に便利です。(以下の例では 100 個まで参照できます) 履歴リストを使って、以前のコマンドを繰り返すこともできます。この変数は C シェルだけに適用できます。次の行を `.cshrc` ファイルに入力します。

```
set history=100
```

`.profile` ファイルに次の行を挿入すれば、`Bourne` シェルと `Korn` シェルについても同様の設定ができます。(以下の例では 100 個まで参照できます)。

```
HISTORY=100
```


デフォルトのファイルアクセス権を設定する

`umask` コマンドを使って、新規に作成される全ファイルと全ディレクトリに対してデフォルトのファイルアクセス権を設定できます。たとえば、セキュリティを重視するため、グループのメンバとその他の全ユーザに対しては、自分の所有するディレクトリに関する読み取り権や実行権 (`-rwxr-xr-x`) (ファイルの場合は `(-rw-r--r--)`) だけを付与したい場合があります。そのような場合は、新規に作成されるファイルやディレクトリがこのアクセス権で保護されるよう `.cshrc` ファイルまたは `.profile` ファイル内で `umask` を使えます。

`chmod` コマンドと同様に、`umask` では数値コードを使ってファイルの絶対アクセス権を表します。ただし、`umask` で使うコードは、`chmod` のコードとは異なる方法で計算します。

まず、`umask` が `000` に設定されている場合は、新規に作成される全ファイルのアクセス権 (読み取り権と書き込み権があり、実行権なし) は次のとおりです。

```
rw-rw-rw- (モード 666)
```

そして、新規に作成される全ディレクトリのアクセス権 (読み取り権、書き込み権、実行権の全アクセス権がある) は次のとおりです。

```
rwxrwxrwx (モード 777)
```

`umask` で使う値を決定するには、ファイルに割り当てられている現在のデフォルトアクセス権の値から、設定したいアクセス権の値 (`chmod` コマンドで指定する値と同じ) を差し引きます。この差し引き後の値を `umask` コマンドで使います。

たとえば、ファイルのデフォルトモードを `666` (`rw-rw-rw-`) から `644` (`rw-r--r--`) に変更する場合は、`666` から `644` を差し引いた値の `022` を `umask` コマンドで指定します。

```
umask 022
```

`chmod` コマンドの数値コードと同様に、`umask` では次の 3 桁の数字を使います。

- 1 桁目は所有者のアクセス権を制御する。
- 2 桁目の数字はグループのアクセス権を制御する。
- 3 桁目の数字はその他の全ユーザのアクセス権を制御する。

数値コードの各桁で設定されるファイルアクセス権を表 10-1 に示します。

表 10-1 umask のアクセス権

umask コード	アクセス権
0	rwX
1	rw-
2	r-X
3	r--
4	-wX
5	-w-
6	--X
7	--- (アクセス権なし)

umask コマンドについての詳細は、『SunOS リファレンスマニュアル (1): ユーザーコマンド』を参照してください。

OpenWindows フォントのカスタマイズ

OpenWindows アプリケーションで表示されるフォントの書体や大きさは、必要に応じてカスタマイズできます。以下の節では、フォントのカスタマイズ方法について説明します。

フォントの書体とポイントサイズを指定する

ウィンドウ用のデフォルトフォントは、12 ポイント (medium) の Lucida Sans です。また、ウィンドウヘッダ用のデフォルトフォントは、Lucida Sans Bold です。必要に応じて、ウィンドウとウィンドウヘッダのフォントには、別の書体やポイントサイズを指定できます。あるウィンドウのフォントだけを一時的に変更した

り、ワークスペースのプロパティウィンドウを使って、すべてのアプリケーションのフォントを変更し常時有効にすることができます。次項以降ではこれらの各方法について説明します。

固定幅フォントとプロポーショナルフォント

一般的にフォントには、固定幅およびプロポーショナルという 2 つの種類があります。固定幅フォントの各文字は、すべて同じスペースを使います。それに対して、プロポーショナルフォントの各文字は、個々の文字幅に応じてさまざまなスペースが必要です。プロポーショナルフォントの方が見た目はよいですが、アプリケーションの中には固定幅フォントを使ったほうが適切なものもあります (例: コマンドツール、シェルツール、よく使われている端末エミュレータアプリケーションの `xterm` など)。

固定幅フォントとプロポーショナルフォントの選択

コマンドツールとシェルツールで表示されるデフォルトフォントは、プロポーショナルフォントです。このフォントは見た目にはよく感じられますが、端末ウィンドウでプロポーショナルフォントを使うと (スペースとタブを扱うときに) 文字の桁合わせの上で問題が発生することがあります。スペースやタブによる文字の桁合わせに問題がある場合には、これらのウィンドウには固定幅フォントを使うことをお勧めします。以下の例では、端末ウィンドウでは固定幅フォントだけを使い、その他のウィンドウとヘッダではプロポーショナルフォントを使っています。

単一のウィンドウで使うフォントの指定

この節では、フォントの書体とポイントサイズを変更して単一のアプリケーションをオープンする方法について説明します。すでに開いているウィンドウのフォントは変更できないことに注意してください。新しいフォントを表示するには、アプリケーションを再起動する必要があります。アプリケーションを再起動するには、コマンド行にアプリケーション名を入力します。

下記に示す基本的なコマンド形式では、単一のアプリケーション名 *application*、`-fn` (font name、フォント名) オプション、フォントの書体 *fontstyle* と大きさ *pointsize* を指定しています。コマンドの最後にアンバサンド (&) を付けると、コマンドを入力した後でウィンドウにシステムプロンプトが戻されます。これにより、そのアプリケーションを起動したウィンドウを引き続き使用できます。

```
$ application -fn fontstyle-pointsize &
```

次の例は、フォントの書体と大きさを指定して単一のアプリケーションを起動する方法を示しています。

- 下記の例は、プロポーショナルフォントの Lucida Sans Typewriter Bold を使ってコマンドツールを再び起動します。

ポイントサイズが指定されていないため、デフォルトの 12 ポイントが使われます。

```
$ cmdtool -fn lucidasans-typewriter-bold &
```

- 下の例は、Lucida Sans Typewriter Bold を使ってシェルツールを再起動し、フォントの大きさを 12 ポイントから 14 ポイントに拡大しています。

フォントの大きさを変更すると、ウィンドウの大きさも変更されることに注意してください。

```
$ shelltool -fn lucidasans-typewriter-bold-14 &
```

- 下記の例は、16 ポイントの Terminal-Bold フォントで xterm 端末ウィンドウを再起動しています。

```
$ xterm -fn terminal-bold-16 &
```

- 下記の例は、14 ポイントの Helvetica Bold フォントでテキストエディタを再起動しています。

```
$ textedit -fn helvetica-bold-14 &
```

-fn オプションは、任意のアプリケーションでフォントの書体と大きさの指定に利用できます。174ページの「利用可能なフォントのリスト」では、OpenWindows アプリケーションで利用できる全フォントを一覧表示する方法について説明しています。

フォントを変更し常時有効にする方法

カスタマイズしたフォントを継続的に使ってアプリケーションを実行するような場合は、ワークスペースメニューにカスタマイズ内容を追加できます。この設定には、ワークスペースのプロパティウィンドウのプログラムメニューカテゴリを使います。これにより、コマンド行オプションを毎回入力する手間が省けます。たとえば、大きいポイントサイズでテキストエディタを実行する場合は、次のコマンド行をプログラムメニューに追加します。

```
textedit -fn lucidasans-typewriter-14
```

同じアプリケーションを異なる大きさのフォントで実行する場合は、プログラムメニューに複数のコマンド行を追加できます。さまざまなポイントサイズでアプリケーションを実行するときには、この方法が便利です。たとえば、12、14、または 18 ポイントのフォントを使ってテキストエディタを実行できるようにするには、プログラムメニューに次のコマンド行を追加します。

```
textedit -fn lucidasans-typewriter-12  
textedit -fn lucidasans-typewriter-14  
textedit -fn lucidasans-typewriter-18
```

上に示したように、ワークスペースのプロパティウィンドウを使ってプログラムメニューをカスタマイズすれば、プログラムメニューから適切なメニュー項目を選択するだけで、3種類のポイントサイズから必要な大きさを選んでテキストエディタを起動できます。

注・プログラムメニューに追加するコマンド行の後には、アンパサンド (&) を付けてはいけません。

利用可能なフォントを表示する

前の例で紹介したフォントより多くのフォントを参照する場合や、それらのフォントを OpenWindows アプリケーションで使う場合は、まず最初に、利用できるフォントをすべて一覧表示して、その中から任意のフォントを選択します。

利用可能なフォントのリスト

利用可能な全フォントのリストは、端末エミュレータウィンドウ内のプロンプトに対して `xlsfonts` と入力すると参照できます。このリストは 1 画面におさまらないことが多いため、コマンドツールを使って表示するのが最適です。コマンドツールのスクロールバーを利用すれば、リスト全体を簡単に参照できます。

注 - 利用可能なフォントは 400 以上あるため、`xlsfonts` で作成されるリストは非常に長くなります。画面上のリストに表示されるフォントの数が少ないと思われる場合は、システム管理者に問い合わせてください。利用可能な全フォントのうちサブセットだけがインストールされている可能性があります。

各フォントには短縮名の他にフルネームがあります。たとえば、`lucidasans-typewriter` のフルネームは次のようになります。

```
-b&h-lucida sans typewriter-medium-r-normal-sans-12-120-72-72-m-0-iso8859-1
```

`xlsfonts` のリスト上のフォントでは、フルネームの後に短縮名が表示されます。この章の説明ではフォントの短縮名を使っています。

フォントを選択できたら、170ページの「フォントの書体とポイントサイズを指定する」の説明に従ってアプリケーションウィンドウのフォントをカスタマイズします。

カラーモニターの調整

Solaris 2.5 に付属の Kodak カラー管理システム (KCMS) を使用することで、画像をスキャンしたり、モニターに表示したり、印刷またはフィルムに記録したりする場合に、正確な色彩で表現できます。

このセクションでは、次の事項について説明します。

- モニター調整の概要
- 表示環境の調整方法
- 調整ツール (調整パック) の接続方法
- 調整ツールの使用方法

注・調整パックは必要不可欠ではありませんが、カラーモニターを調整する場合には使用することをお奨めします。調整パックがなくてもカラーモニターの調整は行うことができます。

モニター調整の概要

スキャナ、モニター、プリンタなどのカラーデバイスは、それぞれカラー再生特性値を持っています。KCMS ソフトウェアは、特定の機種やモデルのカラーデバイスに対応した特性データを使用します。スキャナやモニターの特性データの収集には、専用の計器が必要です。このデータの収集を、キャラクタリゼーションと呼びます。キャラクタリゼーションにより、名目プロファイルと呼ばれるファイルが作成されます。名目プロファイルには、機械が読み取り可能な色再生情報などの詳細情報が含まれます。KCMS は、名目プロファイルを一式備えています。名目は平均的なカラー応答で、個々のデバイスのいくつかのサンプルから得られた規定値です。

注・現在、特性データの収集が可能なのはモニターだけです。スキャナやプリンタでは収集できません。

名目プロファイルは、既知の設定、既知の環境におけるデバイスのカラー再生を表します。名目プロファイルは、ほとんどのワークステーションユーザにとって十分なものとなっています。しかし、カラーデバイスの再生特性値は、年月や媒体、温度などによって変化します。正確にカラーを再生するためには、名目プロファイルを調整して、デバイスの実際の再生特性値を表示環境に反映する必要があります。この名目プロファイルを調整するプロセスを調整と呼びます。表示環境の調整についての詳細は、177ページの「表示環境の調整」を参照してください。調整済みプロファイルについての詳細は、176ページの「調整プロファイルおよびビジュアル」を参照してください。

スキャナとプリンタの調整は簡単ではありませんが、モニターの調整は、一連のプログラム化したテストカラーを表示し、調整パックで出力値を測定して行います。次に、KCMS ライブラリで、モニターの不正確な部分を補正するのに必要な補正係数を計算します。このプロセスがモニター調整です。モニター調整は KCMS 調整ツールを使って行います。調整ツールを使ってモニターを調整する方法の詳細については、181ページの「調整ツールの実行」を参照してください。

モニターのフロントパネルのコントロールつまみ (たとえば輝度、コントラスト、画像または黒レベル) を調整した場合は、モニターのカラー再生を更新するために再調

整する必要があります。カラーを正確に再生したい場合は、これらの設定を調整したつど、または2週間に1回の割合で再調整するようにします。モニターやフレームバッファを交換した場合も、再調整しなければなりません。

アプリケーションプログラムは、KCMS アプリケーションプログラミングインタフェース (API) を介して直接 KCMS ライブラリにアクセスできます。ソフトウェア開発者キット (SDK) を購入されている場合は、KCMS API の詳細は『KCMS *Application Developer's Guide*』を参照してください。

調整プロファイルおよびビジュアル

調整ツールでモニターを調整すると、各フレームバッファのビジュアルごとに調整プロファイルが1つ作成されます。イメージをモニター上に表示するとき、2つの異なるデバイス上でカラーが同じように表示されるかどうかに影響する、カラーの低速シフトと X ビジュアルの使用という2つの条件が存在する場合があります。

再調整により、カラーの低速シフトが修正されます。X ビジュアルは、フレームバッファのハードウェアガンマ検索用テーブル (LUT) により修正されます。ビジュアルとは、表示装置がサポートする表示フォーマットを記述しているデータ構造です。ビジュアルは、ウィンドウ内にある各ピクセルごとの表示特性を記述しています。つまり、ウィンドウのビジュアルは、表示装置のハードウェアガンマ LUT にウィンドウのピクセル値を解釈する方法を指示します。ビジュアルはガンマ LUT を経由して修正されます。

KCMS ソフトウェアを使って修正済み X ビジュアルを調整する場合、その結果得られるカラーは、2つの異なるデバイス上で同じようには表示されません。これはそのビジュアルがガンマ修正を2回受けることになるためです。KCMS ソフトウェアは X ビジュアルがハードウェアガンマ LUT で既に修正済みかどうかを判定して、カラーの一貫性を保証します。X ビジュアルおよびハードウェアガンマ LUT の詳細については、`xgetvisualinfo(3)` および `xsolarisgetvisualgamma(3)` のマニュアルページを参照してください。

モニターを記述している調整済みプロファイルは、`/etc/openwin/devdata/profiles` ディレクトリにコピーされます。読み取り専用の基準プロファイルは、`/usr/openwin/etc/devdata/profiles` にあります。

調整ツールを使って選択したプロファイルのコピーは、(182ページの「モニターを選択するには」を参照してください) ユーザのフレームバッファでサポートされるカラービジュアル形式ごとに作成されます。GrayScale または StaticGray のビジュアルは、カラービジュアルではないため考慮されていません。ユーザのフレームバッ

ファが PseudoColor および TrueColor ビジュアルの両方をサポートする場合、調整ツールは 2 組み以上の測定値を収集します。

表示環境の調整

表示環境を良好にするため、モニターおよび作業環境をさまざまに調整することができます。表示環境が良好であれば、目のストレスも軽減されます。モニターの調整をはじめる前に、良好な表示環境にしてください。調整するのは作業環境またはモニターです。作業環境を調整するときは、モニターの電源をオフに設定します。モニターを調整するときは、電源をオンに設定します。

表示環境の調整についての詳細は、`/usr/openwin/demo/kcms/docs` でオンライン提供される次の説明書を参照してください。

- 『Reducing Eyestrain from Computer Monitors』
- 『Video Monitor Adjustments: “Black Level” and “Picture”』

作業環境の調整

モニターの電源をオフにして、作業環境を次のように調整します。

- 反射光を最小限に抑える
- 室内の明るさを調整する
- 適切な環境の確保
- 画面までの距離を適切に設定する

注 - モニターに専用のスイッチが付いている場合は、コンピュータの電源を切る必要はありません。

反射の抑制

モニター画面にはガラス製のフェースプレートがあり、背後からの光が反射して目に映ります。反射光があると、その反射が起こる場所の表示を認識する目の働きに変化が起こります。モニターのフェースプレートが平らなほど、反射による認識の誤りは減少します。つまり、モニター画面が大きくカーブしていると、背後にある広角度からの反射光が「集まる」こととなります。

画面に反射があるかどうかを判定するには、通常の作業位置に座わり、黒い画面に反射光がないか調べます(反射光は画面のカーブにより歪むことがあります)。強い光源からの光が画面に反射しないように、作業環境を整えてください。装置などの移動ができない場合は光源を移動させるか、カードボードのバッフルプレートなどを使って視界から光源を遮断するようにします。

モニター画面は反射(映り込み)を最小限に抑えるため、反射防止処理が施されていることがあります。このような処理が施されているモニターの場合、電源を切ると画面が黒ずんで見えます。モニターの前面に、外付けで反射防止スクリーンを取り付けることもできますが、その中には光の透過が低く、白色の輝度が使用に耐えないほど低レベルになることがあります。

室内光の調整

背後から来る光だけでなく、自分自身のシルエットなど光以外のオブジェクトも障害になります。光以外のオブジェクトが画面へ反射して見づらくなるのを最小限に抑えるには、室内全体の光のレベル、つまり周囲の照度を抑制します。天井にある蛍光灯は輝度が高いため、この種の反射障害を引き起こす原因となることがよくあります。このような反射が起きているときは、別の光源(照明器具)を使用してください。

適切な環境の確保

画面を見ている時、ディスプレイのもっとも明るい箇所よりさらに明るい光が視野に入ると、視覚ストレスが生じます。カラーサイエンス用語でいう周辺とは、ディスプレイを見ているとき、周辺視覚によって認識される領域をいいます。周辺が明るいと言視覚を妨げるだけでなく、周囲の照度も明るくなります。画面のもっとも明るい白色部分より周囲の方が暗くなるようにしてください。

コンピュータで作業をしているとき、たとえば戸外に面した窓などに目を向けると効果があります。窓があるときはその窓により周辺視覚が乱されないように、窓から十分な距離をおいて座りますが、離れすぎて画面に反射が起きることのないように注意します。

快適な画面距離の確保

画面上でひとつひとつのピクセルが見分けられる場合、座っている位置から画面までの距離が近すぎます。視覚的認識能力、特に読み取り能力は、ドットではなく形状の認識に基づいています。「V」という文字を見たとき、ある角度で交わる2本

の直線として認識するのであり、2つのギザギザのある垂線またはドットの集合体として認識するものではありません。

画面を見るときにストレスを最小限に抑えるには、画面上にあるひとつひとつのピクセルが見えない程度まで、画面から十分な距離をおいて作業をします。十分な距離とは、普通、腕を伸ばしたときの距離をいいます。ワークステーションに座った状態で、腕を伸ばしてください。指先が画面のフェースプレートに触れるようにします。画面からの距離を腕の長さにするので、長時間近くから画面を見つめて起こるストレスを最小限に抑えることができます。

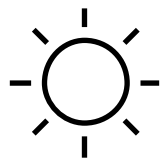
ピクセルが混ざり合い形状として識別できる十分な距離をおいても、文字がうまく判読できない場合は、画面表示に大きなフォントを使うことを検討してください。

モニターの調整

作業環境を適切に設定できたら、モニターを少なくとも1時間ウォームアップして、モニターの黒レベルとピクチャを調整します。



上に示したアイコンはピクチャ (コントラスト) のコントロールつまみを示しています。このつまみで、白の入力信号により再生される輝度を調整します。黒レベルが適切に設定できるとピクチャも快適な表示輝度に設定されます。



上のアイコンは黒レベル (輝度) のコントロールつまみを示します。このコントロールつまみを調整して、真の黒としてモニターに表示されるようにします。このコントロールつまみの調整が不適切な場合、普通、コンピュータモニター、ビデオモニターやテレビ画面などで、ピクチャの再生品質が悪くなるという問題が発生します。

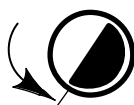
次の条件を満たしていれば、モニターは適切に調整されています。

- 黒の入力信号が真の黒を生成してディスプレイの鮮明度を最大にする
- 白の入力信号が希望する輝度を生成する

モニターを調整するには

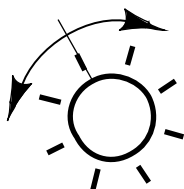
次の手順で、モニターを適切に調整します。

1. モニターのピクチャコントロールつまみを最低位置まで回して、黒のピクチャを表示します。



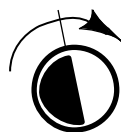
ピクチャコントロールつまみの設定を最低位置にした場合、ピクチャの内容は完全に消滅します。モニターのピクチャを完全に消すことができない場合は、(たとえば、スクリーンセーバーを起動して) 実質的に黒いピクチャを表示するようにしてください。

2. モニターの黒レベルコントロールつまみを回して、黒を適切に調整します。



黒レベルのコントロールつまみをバランスポイント、つまりしきい値まで回します。しきい値は十分に低いため、ピクチャの黒の領域から光は一切出ませんが、コントロールつまみを少しでも上に回すと、その領域がダークグレーに変わります。

3. モニターのピクチャコントロールつまみを回して、輝度レベルを調整します。



黒レベルが適切に設定されたら、ピクチャコントロールつまみを調整して、白の信号が適切な輝度レベルを生成するようにします。このコントロールつまみには、最適な設定というものはありません。設定はユーザの好みに左右されます。モニターの設定は明るすぎないようにしてください。輝度が高すぎると、画面ちらつきが増加し、ピクチャの鮮明度が低下して、CTRの電子ビームの焦点がぼけ、その結果鮮明さが低下します。

注 - 黒レベルのコントロールつまみの調整とピクチャコントロールつまみの調整を数回繰り返し、黒を適切に再生するとともに、白を希望する輝度で生成するよう設定してください。

調整パックの接続

表示環境の調整が済んだら、次に調整デバイス (調整パックと呼びます) をワークステーションに接続します。

注 - 調整パックは必要不可欠ではありませんが、カラーモニターを調整する場合は使用することをお奨めします。調整パックがない場合は、次節の 181 ページの「調整ツールの実行」へ進んでください。

調整パックを接続するには

- ◆ 調整パックをワークステーションのシリアルポート **A(1)** または **B(2)** に接続します。

注 **[IA]** - 調整パックをポート 1 に接続します。ワークステーションが新しい装置 (調整パック) を識別しない場合は、マシンの電源を切りリブートする必要があります。

調整パックは吸引キャップでモニター画面に取り付けます。調整パックの使用時期については、186 ページの手順 4 を参照してください。

調整ツールの実行

表示環境を調整して調整パック (使用している場合) を接続し、モニターのウォームアップを少なくとも 1 時間行なった後、調整ツールはいつでも実行できます。

コマンドツールウィンドウで `kcms_calibrate` を使用して、調整ツールを実行します。`kcms_calibrate` プログラムは、Solaris 2.4 または 2.5 で実行されるため、カラーフレームバッファまたはカラーモニターが必要です。調整ツールは、PseudoColor ビジュアルを調整するのに約 1 分、さらに TrueColor ビジュアルを調

整するのに1分必要です。ユーザのフレームバッファが2種類のビジュアルを両方サポートしている場合は、調整に少なくとも2分必要です。

調整ツールを開始するには

- ◆ `kcms_calibrate` と入力して、調整ツールを実行します。

設定ウィンドウが図 10-1 に示すように表示されます。

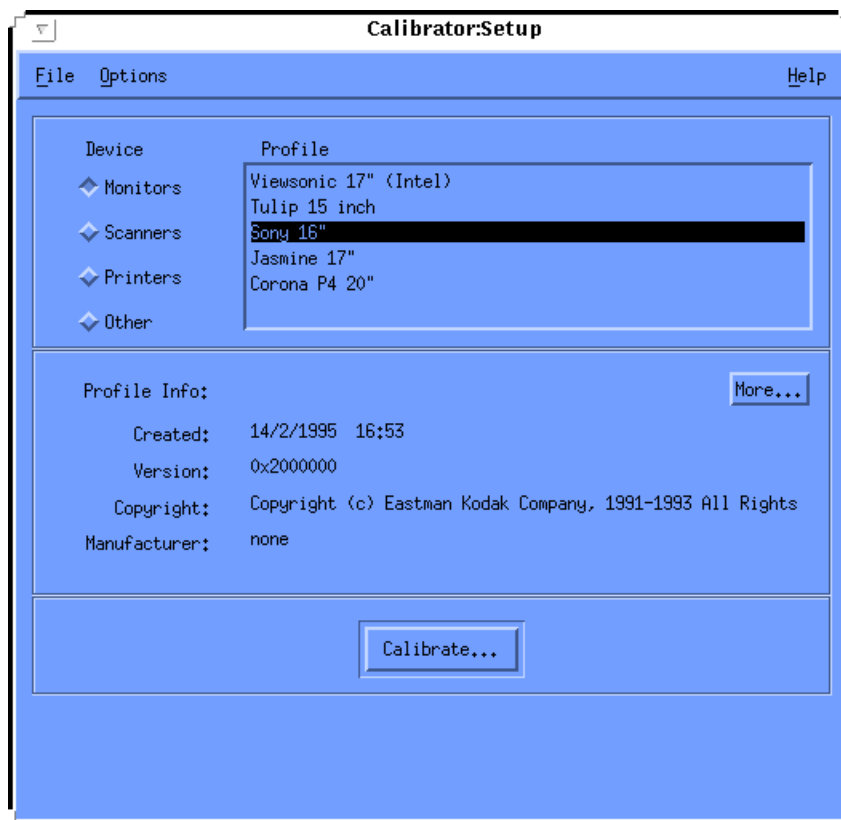


図 10-1 調整ツール設定ウィンドウ

モニターを選択するには

「調整 (Calibrate) ...」ボタンを押す前に、モニターを選択しなければなりません。

1. 「モニター (**M**onitors)」をクリックします。

ユーザ環境で使用できるモニタープロファイルが、図 10-1 に示すように一覧表示されます。

2. モニターのタイプを選択します。

使用しているモニターのタイプがわからない場合は、「詳細 (More) ...」ボタンをクリックすることにより、モニターの一般的情報が得られます。情報は、別のウィンドウに表示されます。次は「Sony 16"」プロファイルを選択して「詳細 (More) ...」ボタンを押した場合に表示される情報の例です。「了解 (OK)」を押してウィンドウを終了します。

- カラー領域 = RGB
- デバイスのメーカー = Sony
- デバイスのモデル = 16"
- ホワイトポイント = 0.964294 1.000000 0.825104

後述の表 10-2 を使ってモニターを選定することもできます。Sun パーツナンバーはモニターのネームプレートに表示されています。使用しているモニターのパーツナンバーを調べ、この表のパーツナンバーと照合してください。該当した行の情報を使ってモニターを選択します。

表 10-2 モニターのプロファイル情報

Sun	パーツナンバー	説明	メーカー	プロファイルの説明
	365-1130-01	P3 16" カラー	Sony	Sony 13/16/19" モニター
	365-1112-01	P3 19" カラー	Sony	Sony 13/16/19" モニター
	365-1159-01	P3 16" カラー	Sony	Sony 13/16/19" モニター
	365-1160-01	P3 19" カラー	Sony	Sony 13/16/19" モニター
	365-1147-01	P3 16" カラー SH (Southern Hemisphere、南半球)	Sony	Sony 13/16/19" モニター
	365-1148-01	P3 19" カラー SH	Sony	Sony 13/16/19" モニター
	365-1288-01	P3 19" カラー ログなし	Sony	Sony 13/16/19" モニター
	365-1289-01	P3 16" カラー ログなし	Sony	Sony 13, 16 および 19" モニター

表 10-2 モニターのプロファイル情報 続く

Sun	パーツナンバー	説明	メーカー	プロファイルの説明
	365-1153-01	Skol 19" P3 MPR2	Sony	なし
	365-1151-02	Rosebud 17" Mid Range (MR) カラー	Sony	なし
	365-1166-02	Rosebud 17" MR カラー ログなし	Sony	なし
	365-1164-02	Rosebud 17" MR SH カラー	Sony	なし
	365-1165-02	Rosebud 17" MPR2 MR	Sony	なし
	365-1068-01	21" カラー	東芝	なし
	365-1286-01	Tulip 15" FS カラー	Nokia	Sony 15" モニター
	365-1167-01	Corona P4 20" カラー	Sony	Sony 20" モニター
	365-1313-01	Corona P4 20" カラー ログなし	Sony	Sony 20" モニター
	365-1317-01	Corona P4 20" カラー SH	Sony	Sony 20" モニター
	365-1316-01	Jasmine 17" N1 カラー	Sony	Sony 17" モニター

モニターを調整するには

1. 「調整 (Calibrate) ...」をクリックします。
 図 10-2 に示すように、別のウィンドウが表示されデバイスの選択を要求してきます。

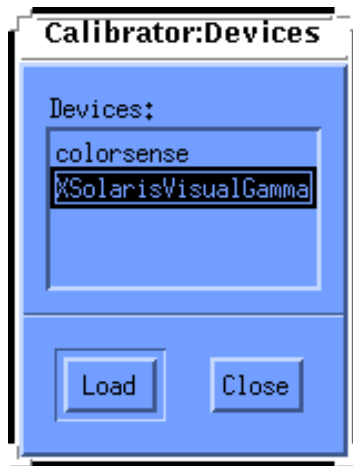


図 10-2 調整デバイスウィンドウ

2. デバイスを選択します。

使用している調整パックがリストに記載されているデバイスと一致する場合は、そのデバイスを選択します。

調整パックを使用しない場合は、「デバイス (Devices) :」から XSolarisVisualGamma を選択します。調整済みのプロファイルは、LUT に入っているユーザのフレームバッファに該当するガンマ値に基づいています。

3. 「ロード (Load)」 をクリックします。

調整パックを使用している場合、図 10-3 に示すように、ウィンドウが別に表示されます。このウィンドウが調整プロファイルウィンドウで、中間色のグレーの円が表示されます。この円は画面の中央に位置します。

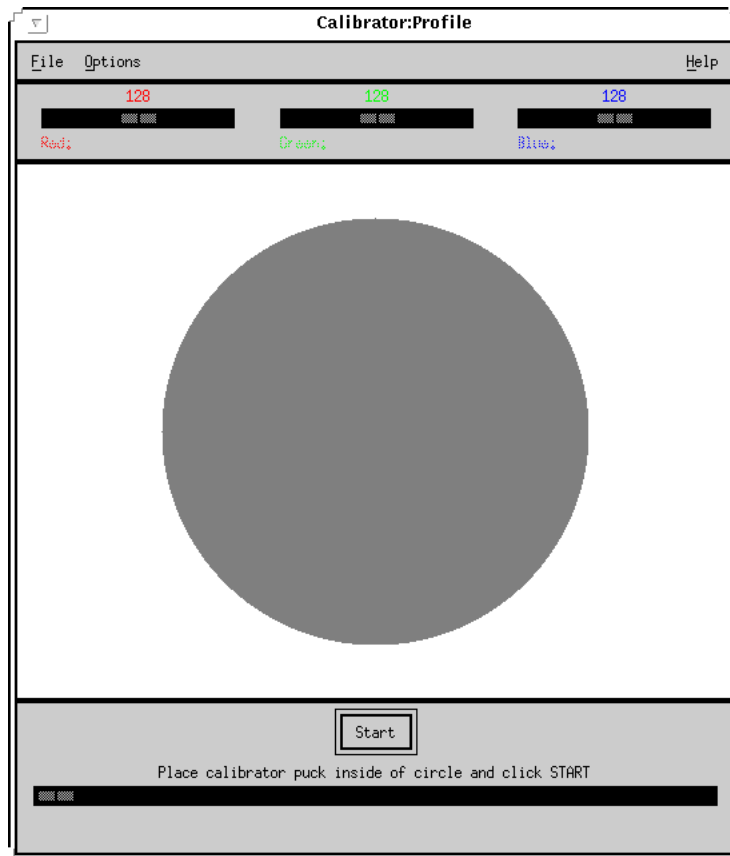


図 10-3 調整プロファイルウィンドウ

画面の中心がもっとも正確な読み取り位置です。調整が完了するまでは、ウィンドウを移動しないでください。また、直接関係のないウィンドウ (たとえば、ポップアップダイアログ) を調整中に表示すると、調整ウィンドウが不鮮明になるので避けてください。

調整パックを使用していない場合は、調整プロファイルウィンドウが表示されません。ウィンドウやポップアップダイアログの移動などを気にする必要はありません。186ページの手順5へ進んでください。

4. 調整パックを使用する場合は、円の中心にしっかりと取り付けます。
5. 「開始 (Start) ...」 ボタンをクリックします。
数秒経つと円は青緑色となり、調整が開始されます。

使用するフレームバッファのタイプに従って、(レッド、グリーン、ブルーの) 測定サイクルが直ちに繰り返されます。ユーザのフレームバッファが PseudoColor と TrueColor の両方のビジュアルをサポートしている場合は、2組の測定が行われます。フレームバッファが、その中の1つしかサポートしていないとき、測定は1組だけとなります。

ポップアップダイアログが色付き円の中心に表示される場合は、調整を再起動する必要があります。この場合、調整データには誤差が生じており正確な測定データは得られません。調整を中止する方法についての詳細は、187ページの「調整を中断するには」を参照してください。

ツールが測定値の読み取りを完了すると、モニターのプロファイルが更新され、メッセージが表示されて、調整が完了したことが分かります。

6. 調整パックを画面から取り外します。

調整が完了したら調整パックを画面から取り外します。取り外した調整パックは、ガラスのフェースプレートにほこりなどが付着しない場所に保管してください。

調整を中断するには

- ◆ 「中止 (Stop)」をクリックします。

ウィンドウが別に表示され、調整を継続するか、終了するか尋ねてきます。調整パックが円を読み取れなかった場合は、調整をいったん終了し、再起動しなければなりません。

調整は、現在のカラー測定がすべて完了するまで中断することはありません。たとえば、レッドの測定インジケータが 24 を示しているとき「中止 (Stop)」をクリックしても、調整は調整ツールがレッドの測定を終了するまで継続します (インジケータが 255 の場合)。

調整ツールを終了するには

- ◆ 「終了 (Quit)」をクリックします。

調整ウィンドウを終了し、調整ツールを終了できます。

エラーメッセージ

調整ツールの実行時に発生するエラーメッセージについて説明します。

スワップ空間

スワップ空間のエラーは、メモリ割り当て時にエラーが発生したことを示しています。実行中のアプリケーションの数が多すぎるか、スワップ空間がさらに必要なことを意味します。次のスワップ空間エラーが表示される場合は、一部のアプリケーションを終了させて、調整ツールを再起動してください。

- スワップメモリ不足のため継続できません。
- ビジュアルデータ構造を初期化できません。
- 調整データ構造を初期化できません。

パッケージのインストール

パッケージのインストールエラーは、KCMS パッケージが適切にインストールされていないことを示しています。次のパッケージインストールのエラーメッセージが表示される場合、KCMS パッケージのインストールをやり直さなければなりません。

- プロファイルをロードできません。
- このデバイス用のプロファイルを更新できません。
- ビジュアルプロファイルを作成できません。
- プロファイルファイルを /etc/openwin に構成できません。

調整パックの接続

調整パックの接続エラーは、調整ツールと調整パックに通信上の障害があることを示しています。調整パックがポートにしっかりと接続されていないか、あるいは正しいポートに接続されていない場合などです。

注 [IA] - 調整パックはポート 1 に接続しなければなりません。

次の調整パック接続エラーが表示される場合、調整がワークステーションのシリアルポート A、またはシリアルポート B のいずれかにしっかりと接続されているか確認してください。

- 調整ツールが正しく機能していません。
- 調整ツールからの応答が得られません。
- 調整パックのバージョン番号を読み取れません。
- 調整パックの感度を獲得できません。
- 調整パックの感度を設定できません。
- 調整パックの積分時間が得られません。
- 調整パックの再表示フィールドの値を獲得できません。
- 調整パックの平均値を設定できません。
- モジュールが輝度データの測定に失敗しました。

OWconfig データ

OWconfig エラーメッセージは、OWconfig データベース内のデータが破損していることを示しています。OWconfig ファイルを調べて、不適切なエントリがないか確認してください。調整ツールを終了して、再起動してください。

デバイスハンドラ

デバイスハンドラのエラーメッセージは、選択したデバイスにデバイスハンドラがないことを示しています。調整ツールは、調整モジュールをロードできません。デバイスハンドラを /usr/openwin/etc/devhandlers ディレクトリにインストールするか、または別のデバイスを選択する必要があります。

モジュールの初期化

モジュールの初期化エラーメッセージは、モジュールが初期化を終了できなかったことを示しています。初期化を終了させるか、または別の共用オブジェクトを使用する必要があります。

モジュール測定の失敗

モジュール測定の失敗というエラーメッセージは、モジュールがモニターの輝度測定に失敗したことを示しています。普通、このような事態は「中止 (Stop)」ボタンを押したときに発生します。調整ツールを終了するかダイアログを閉じて、その後調整を再起動してください。

不正なプロファイル

不正なプロファイルというエラーメッセージは、選択したプロファイルが調整できないことを示しています。有効なプロファイルを選択する必要があります。

プライベートカラーマップエントリの割り当て

プライベートカラーマップエントリの割り当てエラーメッセージは、調整ツールがエントリの割り当てに失敗したことを示しています。調整ツールは独自のカラーを指定するため、使用可能なダイナミックビジュアルを使用して、ウィンドウサーバを実行しているかを確認する必要があります。調整ツールを終了するか、または調整を再起動します。

OpenWindows バージョン 3.3 以降への移行

Solaris ではデフォルトのユーザ環境として OpenWindows が使われています。そのため、現在使っているソフトウェアがこのユーザ環境と適合しない場合は移行が必要です。たとえば、SunView ユーザ環境やバージョン 3.3 より前の OpenWindows を使用している場合は、OpenWindows バージョン 3.3 以降にアップグレードする必要があります。この付録では、OpenWindows バージョン 3.3 以降への移行方法について説明します。

注 - OpenWindows バージョン 3.3 以降の環境では、SunView をサポートしません。したがって、OpenWindows バージョン 3.3 以降に移行すると、SunView アプリケーションは実行できなくなります。

SPARC: SunView 環境からの移行

次の説明を参照すると、SunView 環境から OpenWindows 環境に円滑に移行できます。

`.defaults` ファイルと `.Xdefaults` ファイル

SunView 環境と同様の設定で OpenWindows 環境をカスタマイズする場合は、SunView で使われる `.defaults` ファイルを、OpenWindows で使われる `.Xdefaults` ファイルに変換できます。ホームディレクトリ内に `.defaults`

ファイルがある場合は、次に示すようにホームディレクトリ内で `convert_to_Xdefaults` プログラムを実行してください。

```
$ cd
$ /usr/openwin/bin/convert_to_Xdefaults .defaults
```

これによって、ホームディレクトリ内に `.Xdefaults` ファイルが作成されます。この `.Xdefaults` ファイルは、OpenWindows の起動時にユーザ環境をカスタマイズするために使われます。

バージョン 3.3 より前の OpenWindows 環境から移行する

バージョン 3.3 より前の OpenWindows を従来使用していて、バージョン 3.3 以降の OpenWindows に移行する場合は、この節を注意して読んでください。特にここでの説明は、OpenWindows バージョン 2 を使用していたユーザに当てはまります。

OPENWINHOME 環境変数

OpenWindows バージョン 3.3 より前の OpenWindows を使っている場合は、システムで `OPENWINHOME` 環境変数を使うよう、ユーザの環境を設定している可能性があります。OpenWindows バージョン 3.3 以降では、ユーザが自らコマンドを使ったり、起動ファイルに記述したりして、`OPENWINHOME` 環境変数を設定することはお勧めしません。

`OPENWINHOME` 環境変数は、`openwin` コマンドを実行すると自動的に `/usr/openwin` に設定されるため、ユーザが設定する必要はありません。

ホームディレクトリ内の `.profile` ファイルまたは `.cshrc` ファイルで `OPENWINHOME` 環境変数を設定している場合は、OpenWindows バージョン 3.3 以降を使う前に、該当行をコメントに変更するか削除します。

`.profile` ファイルまたは `.cshrc` ファイル内の `OPENWINHOME` 環境変数に関する行をコメントに変更したり削除したりする場合は、次の手順を実行します。

1. `vi` などのテキストエディタを使って、`.profile` ファイルまたは `.cshrc` ファイルを開きます。

2. 下記に示すように、変数名の前にポンド記号 (#) を挿入してコメントに変更するか、その行全体を削除します。

.profile ファイルを編集するときは次の手順 a を、.cshrc ファイルを編集するときは手順 b を実行します。

- a. .profile ファイルの場合

```
#OPENWINHOME=/usr/openwin
```

- b. .cshrc ファイルの場合

```
#setenv OPENWINHOME /usr/openwin
```

3. ファイルを保存してエディタを終了します。

.xinitrc ファイル

.xinitrc ファイルと /usr/openwin/lib/Xinitrc ファイルに関する重要な注意事項を次に示します。

1. OpenWindows バージョン 2 環境では、openwin スクリプトは自動的に /usr/openwin/lib/Xinitrc をホームディレクトリ内の .xinitrc というファイルにコピーしていました。OpenWindows バージョン 3.3 以降の環境では、このようなコピーは行われません。これは次の理由によります。
 - a. ホームディレクトリ内に .xinitrc ファイルが存在しないと、openwin スクリプトはデフォルトの起動ファイルの /usr/openwin/lib/Xinitrc を使います。Xinitrc は、デフォルトのファイルを上書きします。
 - b. OpenWindows バージョン 3.3 以降に付属の /usr/openwin/lib/Xinitrc ファイルをデフォルトとして使ってください。(ただし、OpenWindows バージョン 2 環境で .xinitrc ファイルに特別な変更を加えた場合は、この節の指示に従ってその内容を保持することもできます。)
2. 複数の画面を使ってウィンドウシステムを実行する場合でも、複数の olwm インスタンスを起動する必要はありません。

適切な起動ファイルの使い方

OpenWindows バージョン 3.3 より前のバージョンの OpenWindows を使っている場合は、`.xinitrc` ファイルの状態を確認する必要があります。`.xinitrc` ファイルはホームディレクトリにある OpenWindows の起動ファイルで、ユーザが定義したオプションが入っています。

`.xinitrc` ファイルを確認するには、次のコマンドを入力します。

```
$ cd
$ ls -a .xinitrc
```

上記のコマンドの出力結果に応じて、次のどれかを実行します。

- `.xinitrc` ファイルが存在しない(つまり、上記の `ls -a` コマンドで `.xinitrc` が表示されない)場合は、何もする必要はありません。ホームディレクトリ内に `.xinitrc` ファイルが存在しない場合は、OpenWindows はシステムデフォルトの起動ファイルを使用します。
- `.xinitrc` ファイルが存在するが(つまり、上記の `ls -a` コマンドで `.xinitrc` が表示されるが)そのファイルを変更していないか、あるいは以前変更した内容を保持しなくてもよい場合は、194ページの「起動ファイルの手順」の194ページの手順1を実行します。
- `.xinitrc` ファイルが存在し(つまり、上記の `ls -a` コマンドで `.xinitrc` が表示され)そのファイルに加えた以前の変更内容を保持したい場合は、194ページの「起動ファイルの手順」の194ページの手順2を実行します。

起動ファイルの手順

1. ホームディレクトリから `.xinitrc` ファイルを削除する場合は、次のコマンドを入力します。

```
$ rm .xinitrc
```

2. `.xinitrc` ファイルの変更内容を保持する場合は、次の手順を実行します。
 - a. `.xinitrc` を `.xinitrc.save` にファイル名を変更します。

```
$ mv .xinitrc .xinitrc.save
```

- b. /usr/openwin/lib/Xinitrc をホームディレクトリ内の .xinitrc にコピーします。

```
$ cp /usr/openwin/lib/Xinitrc $HOME/.xinitrc
```

- c. .xinitrc.save の中の保持したい行を .xinitrc の中に追加します。



注意 - .xinitrc ファイルを編集するときに、olwm をデュアル・フレームバッファ用などに追加したり、svenv を追加したり、/usr/openwin/lib/openwin-sys を含む行を削除したりしないよう注意してください。

ワークスペース・プロパティ

バージョン 3.3 より前の OpenWindows では、ワークスペース・プロパティメニューを使って、変更した内容はホームディレクトリの .Xdefaults ファイルに格納されていました。OpenWindows バージョン 3.3 以降でも、これらの変更内容は同じくホームディレクトリの .OWdefaults ファイルに格納されます。.Xdefaults ファイルも引き続き使えますが、.OWdefaults ファイルで設定したカスタマイズが優先されます。

.Xdefaults ファイルは、ワークスペース・プロパティメニューでは設定できないような補足的なカスタマイズだけに使います。たとえば、vi などのテキストエディタで .Xdefaults ファイルを編集すると、OpenWindows 以外のアプリケーションのカスタマイズや C プリプロセッサマクロを追加することができます。ワークスペース・プロパティメニューを使っても、.Xdefaults の設定内容に影響を及ぼしません。

.Xdefaults ファイルがホームディレクトリに存在し、何の変更も加えない場合は、.OWdefaults ファイルによる設定内容のほうが優先されるため、.Xdefaults ファイルを削除する必要はありません。

ワークスペースメニューのカスタマイズ

OpenWindows バージョン 3.3 以降では、ワークスペース・プロパティメニューを使ってワークスペースメニュー内のプログラムサブメニューをカスタマイズできます。OpenWindows バージョン 3.3 より前のバージョンでは、ホームディレクトリ内の `.openwin-menu` ファイルを編集することによってこのカスタマイズを行なっていました。

注 - ホームディレクトリ内に `.openwin-menu` ファイルがない場合は、次の手順を実行する必要はありません。ワークスペース・プロパティメニューを使って、ワークスペースメニューをカスタマイズできます。

ホームディレクトリ内に `.openwin-menu` ファイルがある場合は、ワークスペース・プロパティメニューを使ってワークスペースメニューをカスタマイズできるように、次の手順を実行する必要があります。

`.openwin-menu` ファイル内で次の行を検索します。

```
``Programs`` MENU /usr/openwin/lib/openwin-menu-programs
```

この行を削除して次の行で置き換えます。

```
``Programs`` INCLUDE openwin-menu-programs
```

`.openwin-menu` ファイル内に削除や置換対象の行が存在しない場合は、上記に示した行を `.openwin-menu` ファイルに追加します。

上記の行を追加したり置き換えたりすると、デフォルトのプログラムサブメニューがワークスペースメニューに追加されます。この変更により、ワークスペース・プロパティメニューを使ってワークスペースメニューをカスタマイズできるようになります。

ワークスペースメニューに不要な項目がある場合は、`.openwin-menu` ファイルを編集して不要な項目に関連する行を削除します。

Solaris 2.5 への移行

この付録では、SPARC システムの SunOS 4.x および IA システムの Solaris 2.1 から Solaris 2.5 へ移行する際、ユーザおよびシステム管理者が直面すると考えられる変更について簡単に説明します。

注 **[SPARC]** - Solaris 2.5 アップグレードオプションは、システムに接続された 1 つ以上のディスクに Solaris 2.1 以降のルートファイルシステムがある場合にだけ使用できます。SunOS 4.1 システムでは、アップグレードオプションは使用できません。

注 **[IA]** - Solaris アップグレードオプションは、Solaris 2.1 から 2.5 への移行では使用できません。

SunOS 4.x から Solaris 2.5 への移行方法、およびこの 2 つのソフトウェア環境の相違点についての詳細は、『Solaris 移行ガイド』および『バイナリ互換性ガイド』を参照してください。

Solaris 2.5 と旧バージョンの相違点については、『Solaris 移行ガイド』を参照してください。

SunOS 4.x からの移行

Solaris 2.5 の環境には、SunOS 5.x システムソフトウェアが含まれます。以降の各セクションではユーザとシステム管理者の方を対象に、SunOS 4.x と SunOS 5.x の相違点について簡単に説明します。

SPARC: SunOS 4.x ユーザに影響のある変更

SunOS 4.x リリースと SunOS 5.x リリースの明確な相違点は、次のとおりです。

- UNIX コマンドの多くが変更されています。詳細については、『Solaris 移行ガイド』を参照してください。
- 印刷サブシステムが変更されています。たとえば、次のとおりです。
 - lpr から lp へ
 - lpq から lpstat へ
 - lprm から cancel へ
 - troff ではプリンタ名が必要
- メールプログラムが変更されています。SunOS 4.x メールボックスとフォルダは、次の3つの Solaris 2.3 メールプログラムと完全な互換性があります。
 - mailtool、DeskSet 環境マルチメディアメールツール
 - mail、コマンドラインユーティリティ
 - mailx、コマンドラインユーティリティ
- システム・ファイルの中には、その位置および名前が変更されているものがあります。たとえば、次のとおりです。
 - /etc/fstab から /etc/vfstab へ
 - /var/spool/mail から /var/mail へ
 - /vmunix から /platform/*/kernel/unix へ

以上についての詳細は、『Solaris 移行ガイド』を参照してください。

SPARC: SunOS 4.x システム管理者に影響のある変更

SunOS 4.x 環境から SunOS 5.x 環境へ移行するときに見られる相違点は、次のとおりです。

- Solaris 2.5 ソフトウェアを SPARC システムへインストールする作業は、ローカルまたはリモートの CD-ROM ドライブ、またはネットワークからのみ可能です。
- Solaris 2.5 ソフトウェアメディアは、パッケージとクラスタで構成されたソフトウェアグループに分散しているため、簡単にインストールできます。Solaris

2.5 パッケージの最新情報については、pkginfoまたは swmtool を参照してください。

- デバイス命名規則が変更されています。たとえばディスクは現在、次のように命名されています。

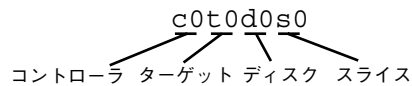


図 B-1 デバイス命名規則

ディスク (xy, xd) の中には、ターゲットコンポーネントのないものがあるので注意してください。ターゲットのないディスク名は、c0d0s0 となります。

- ファイルシステムのディレクトリ構造が変更されています。たとえば、
 - unix と呼ばれるカーネル、およびその関連モジュールは、/kernel ディレクトリに格納されています。
 - /opt ディレクトリは新しいディレクトリです。オプションのソフトウェアアプリケーションを格納またはマウントするために、インストール時に作成されます。
 - /devices ディレクトリはデバイスファイルの階層です。/dev ディレクトリへシンボリックリンクしているので、SunOS 4.x デバイス命名規則と互換性があります。
 - /usr ディレクトリが再構成されています。
 - /etc/termcap から /terminfo データベースへ変わっています
 - /etc/fstab から /etc/vfstab ディレクトリへ変わっています
 - /var/spool/mail から /var/mail ディレクトリへ変わっています
 - /etc/printcap から /etc/lp ディレクトリへ変わっています
 - リモートファイルシェアリングサービス (RFS) のオプションのソフトウェア・パッケージは、Solaris 2.2 以降サポートされていません。
- SunOS 5.x カーネルは動的であるため、ユーザがカーネル構成ファイルの再構築や編集を行うことはできません。
- システムに追加された新しいデバイスに必要なドライバは、カーネルが自動的にロードします。
- Solaris 2.5 の起動とシャットダウンの手順が変更されています。
 - init コマンドは各実行レベルで異なるスクリプトを使用します。

- shutdown コマンドは、SunOS 4.1 システムで使用できるすべてのオプションをサポートしていません。
- fastboot および fasthalt コマンドは、SunOS/BSD ソース互換パッケージの使用時だけ使用できます。
- コマンドの出力だけでなく、多くのコマンドが変更されています。したがって、スクリプトを書き換える必要がある場合もあります。詳細については、『Solaris 移行ガイド』を参照してください。
- NIS (ネットワーク情報サービスのネームサービス) は NIS+ に変わりました。しかし、NIS+ では NIS データを使用するので、NIS+ への移行は簡単です。
- ユーザによるローカルシステム管理を可能とする Motif アプリケーションである、システム管理ツール (Administration Tool) は、システムデータベース情報、ホスト情報、プリンタ、シリアルポート、ユーザアカウント、ソフトウェアなどを管理します。
- 印刷管理コマンドが変更されています。詳細なリストについては、『Solaris 移行ガイド』を参照してください。
- 端末装置およびモデムは、サービスアクセス機能 (SAF) およびサービスアクセス制御 (SAC) によって制御されます。したがって、/etc/ttytab ファイルはなくなりました。

SPARC: SPARC システム用 SunOS 4.x リリースとの互換性

SPARC システムの SunOS 5.x システム・ソフトウェアへの移行を容易にするため、Solaris 2.5 環境は次の 2 つのパッケージを提供します。

- SunOS/BSD ソース互換パッケージ
- SunOS バイナリ互換パッケージ

このパッケージを使用する場合は、このパッケージが次の方だけを対象とした、移行補助のための一時的な手段であることに注意してください。

- SunOS 4.x アプリケーションソースコードをコンパイルして、Solaris 2.5 環境で実行する開発者
- Solaris 2.5 環境などの SVID 準拠の環境で実行するように、SunOS 4.x ベースのアプリケーション・バイナリをまだ移植していないユーザ

SPARC: SunOS/BSD ソース互換パッケージ

このソフトウェア・パッケージがインストールされている場合、SunOS 5.x システム・ソフトウェアに含まれていない SunOS/BSD コマンドの一部、または同ソフトウェアで変更済みの SunOS/BSD コマンドの一部がサポートされます。SunOS 4.x 環境に適合しているアプリケーションソースコードは、SunOS/BSD ソース互換パッケージを使って、Solaris 2.5 環境でコンパイルされ実行されます。

詳細については、『バイナリ互換性ガイド』および『*Source Installation and Media Preparation Guide*』を参照してください。

SPARC: SunOS バイナリ互換パッケージ

OpenWindows や SunOS 4.1.x¹ の環境で作成し、動的または静的にリンクした実行可能アプリケーションは、SunOS バイナリ互換パッケージおよび OpenWindows バイナリ互換パッケージにより Solaris 2.5 環境でサポートされます。

このパッケージを使用するためには、SunOS 4.x で記述されたアプリケーションは、次の処理を実行できません。

- カーネルに直接トラップする
- システムファイルに直接書き込む
- /dev/kmem または libkvm ルーチンを使用する
- 未発行の SunOS インタフェースを使用する
- SunOS 以外のドライバを使用する

SunOS バイナリ互換パッケージを使用するには、SunOS/BSD ソース互換パッケージをインストールしなければなりません。

IA: Solaris 2.1 からの移行

Solaris 2.1 リリースから Solaris 2.5 システム環境へ移行するユーザおよび管理者は、いくつかの相違点が存在します。たとえば、インストールの手順やディスクのフォーマット方法に変更があり、印刷コマンドも異なります。

1. 「SunOS 4.1.x」リリース/環境には、SunOS 4.1 リリースとこれ以降のすべての 4.1 リリース、たとえば、4.1.1、4.1.1 Rev B、4.1.2、4.1.3 および 4.1.3c などが含まれます。

IA: ユーザに影響のある変更

IA キーボードのキーボード・マッピングが変更されています。キーマップの完全な表については、付録 C を参照してください。

IA: システム管理者に影響のある変更

ネットワークされた IA システムでは、Solaris 2.5 ソフトウェアのインストールに、いくつかの相違点があります。

- `kdmconfig` は `sysidconfig` によって呼び出され、インストール時に `devconfig` を置き換えます。
- `bootparams` データベースサーバの照会時に、Ethernet とホストマップが適切に設定されていれば、インストールは支障なく実行されます。矛盾する事項がある場合、またはスタンドアロンシステムの場合、システムに取り付けられているキーボード、マウスおよびディスプレイの定義が要求する画面が表示されます。`kdmconfig(1M)` を参照してください。

IA システムでは、SCSI ディスクと IDE ディスクのフォーマットに違いがあります。

- SCSI ディスクのフォーマット、ラベル付け、分析、修理などを行うための、`format(1M)` ユーティリティが使用できます。このユーティリティは、IA リリースの Solaris 2.1 で使用可能な `addbadsec(1M)`、`diskscan(1M)`、`fdisk(1M)`、および `fmthard(1M)` に含まれています。IDE ディスクをフォーマットするには、DOS フォーマット・ユーティリティを使用します。ただし、IA システムで IDE ディスクのラベル付け、分析、修理などをするには、Solaris `format(1M)` ユーティリティを使用します。

IA システムでは、パラレル PostScript™ プリンタのサポートに変更があります。

- `lp` ドライバが修正されたため、IA システムでのパラレル PostScript プリンタの設定は、現在、シリアル・プリンタの設定とまったく同じです。唯一の相違点は、ポートのデバイス名を指定する必要があることです。
- 次のファイルは、Solaris 2.5 ではサポートされていません。

- `/etc/lp/fd/ppostio.fd`
- `/etc/lp/fd/ppostior.fd`
- `/usr/share/lib/terminfo/P/PPS`
- `/usr/share/lib/terminfo/P/PPSR`

- IA システムの lp ドライバは、現在、postio の追加 ioctl's を利用していません。

IA: Solaris 2.5 システム管理ツール

Solaris 2.5 リリースでのシステム管理上の利点は、Solaris システムソフトウェアと ONC ネットワーキングプロトコルおよび管理アプリケーションを改良した結果です。中間リリースを使用していない IA ユーザのために、これらの改良点について、以降の各節で説明します。

IA: Solaris 2.3 リリース以降

次の管理機能が Solaris 2.3 リリース以降、改良されています。

- キーボード、マウス、ディスプレイの構成が、kdmconfig により更新されます。
- Bootparams によりキーボード、ディスプレイ、マウスが指定されています。
- ブートシーケンスは IA マシンでは異なります。

IA: Solaris 2.1 リリース以降

次の管理機能が Solaris 2.1 リリース以降、改良されています。

- 端末およびモデム設定を簡略化するため、システム管理ツールが改良されています。
- 自動セキュリティ拡張ツール (ASET) により、システム管理者は簡単にシステムのセキュリティの向上を図ることができます。
- ダイナミックカーネルとロード可能モジュールにより、次の機能が簡略化されています。
 - 異なるシステム用カーネルの管理
 - 動作中のシステムへの新規デバイスの追加
 - カーネルを再構築しないデバイスドライバの追加
- NIS+ ネームサービスによりサブドメインが作成され、ネットワークのリソースに対してセキュリティが割り当てられます。

キーボードの設定変更

この付録では、キーボードの再マッピングについて説明します。キーボードの特殊キーのマッピングを変更する方法を取り上げます。また、キーボード上の Compose キーの設定解除と設定の方法についても説明します。

マウスボタンの再マッピングについて詳しくは (たとえば、左ききの人がマウスを操作しやすいようにするなど)、『OpenWindows ユーザーズガイド』を参照してください。

Compose キーの設定解除と設定

注 [IA] - Compose キーは、IA システムでは Ctrl-Shift-F1 キーに割り当てられています。

Compose キーを使用しない場合は、そのキー操作を無効にすることができます。まず Multi_key のキーコードを探します。

```
$ xmodmap -pk | grep Multi_key
```

次のような行が表示されます。

```
nn 0xff20 (Multi_key)
```

ここで重要な情報は、行の先頭の *nn* で示される 2 桁のキーコード番号です。このキーコード番号を使って、`.xinitrc` ファイル内に次の行を挿入します。

```
xmodmap -e 'keycode nn = NoSymbol'
```

Compose キーを再設定するには、`.xinitrc` ファイル内の上記の行をコメントに変更した後で OpenWindows を再起動します。

SPARC: 左きき用のキーの再マッピング

この節で示すキーの再マッピングスクリプト (Type-4 と Type-5 のキーボード用) を使って、キーボードの左右パネル上の大部分の特殊キー (つまり、キーボードの左側と右側にあるキーパッド) を再マッピングします。

注 **[SPARC]** - この節の以降の説明はすべて、SPARC ベースのマシンだけに適用されるので注意してください。

SPARC: 再マッピングスクリプトの使い方

再マッピングスクリプトを作成し使用する手順は次のとおりです。

1. 任意のテキストエディタを使って、`lefty.data` というスクリプトファイルを作成します。

このファイルは、どのディレクトリにあってもかまいません。ただし 206 ページの手順 4 は、このファイルを作成したのと同じディレクトリで実行しなければなりません。

2. 後述の表 C-1 に示すスクリプトを入力します。
先頭に感嘆符 (!) が付いた行はコメント行で、実際の操作は実行しません。
3. ファイルを保存してエディタを終了します。
4. プロンプトに対して、次のコマンドを入力します。

```
$ xmodmap lefty.data
```

上記のコマンドは、スクリプトファイル `lefty.data` と同じディレクトリ内で入力してください。

5. ワークスペース内でマウスボタンをクリックして、スクリプトの内容を有効にします。

上の手順を実行すると、左ききの人向けにマッピングされたキーボードを使えるようになります。

手順 1 で説明したように、次のスクリプトを `lefty.data` ファイルに入力します。

表 C-1 lefty.data スクリプト

```

!
! lefty.data
!
! Data for xmodmap to set up the left and right function keys
! for left-handed use on Sun Type-4 keyboard. To use this data,
! type the following where <filename> is the name of the file
! (i.e., lefty.data).
!
! xmodmap <filename>
!
! The comments below correspond to the keycode assignments
! following immediately thereafter.
!

! swap L2 (Again) with R1 (Pause)
! swap L3 (Props) with R6 (KP_Multiply)
! swap L4 (Undo) with R4 (KP_Equal)
! swap L5 (Front) with R9 (KP_9)
! swap L6 (Copy) with R7 (KP_7)
! swap L7 (Open) with R12 (KP_6)
! swap L8 (Paste) with R10 (KP_Left)
! swap L9 (Find) with R15 (KP_3)
! swap L10 (Cut) with R13 (KP_1)
!
! chng R3 (Break) to L1 (Stop)
! chng R2 (Print) to R10 (Left)
! chng R5 (KP_Divide) to R12 (Right)
!
! chng Linefeed to Control-R
!
keycode 10 = R1 R1 Pause
keycode 28 = L2 L2 SunAgain
keycode 32 = R6 R6 KP_Multiply
keycode 54 = L3 L3 SunProps
keycode 33 = R4 R4 KP_Equal
keycode 52 = L4 L4 SunUndo
keycode 56 = R9 R9 KP_9 Prior
keycode 77 = L5 L5 SunFront
keycode 58 = R7 R7 KP_7 Home

keycode 75 = L6 L6 SunCopy
keycode 79 = Right R12 KP_6
keycode 100 = L7 L7 SunOpen
keycode 80 = Left R10 KP_4
keycode 98 = L8 L8 SunPaste
keycode 102 = R15 R15 KP_3 Next
keycode 121 = L9 L9 SunFind
keycode 104 = R13 R13 KP_1 End
keycode 119 = L10 L10 SunCut
keycode 30 = L1 L1 SunStop
keycode 29 = Left R10 KP_4
keycode 53 = Right R12 KP_6
keycode 118 = Control_R
add control = Control_R

```


SPARC: キーボードの再マッピングの取り消し

キーボードのキーの設定を元に戻すには、次の2つの方法があります。1つは、OpenWindows を一度終了して再起動します。もう1つは、キー設定を元に戻すための別のスクリプトを作成して、必要に応じてそのスクリプトを実行します。2番目の方法は、キー設定を定期的に再設定する場合に望ましい方法です。

再設定用のスクリプトを作成するには、次の手順を実行します。

1. 任意のテキストエディタを使って、`nolefty.data` というスクリプトファイルを作成します。

このファイルは、`lefty.data` スクリプトと同じディレクトリに作成してください。

2. 後述の表 C-2 に示すスクリプトを入力します。
先頭に感嘆符 (!) が付いた行はコメント行で、実際の操作は実行しません。
3. ファイルを保存してエディタを終了します。
4. コマンドプロンプトに対して、次のコマンドを入力します。

```
$ xmodmap nolefty.data
```

`nolefty.data` ファイルを有効にするには、上記のコマンドを `nolefty.data` が存在するディレクトリで実行してください。

表 C-2 nolefty.data スクリプト

```
!  
! nolefty.data  
!  
! Data for xmodmap to reset the left and right function keys  
! after being set for left-handed use on the Sun type-4 keyboard.  
! To use this data, type the following where <filename> is the name  
! of the file.  
!  
! xmodmap <filename>  
!  
! Reassign standard values to left function keys.  
!  
keycode 10 = L2 L2 SunAgain  
keycode 32 = L3 L3 SunProps  
keycode 33 = L4 L4 SunUndo  
keycode 56 = L5 L5 SunFront  
keycode 58 = L6 L6 SunCopy  
keycode 79 = L7 L7 SunOpen  
keycode 80 = L8 L8 SunPaste  
keycode 102 = L9 L9 SunFind  
keycode 104 = L10 L10 SunCut  
!  
! Reassign standard values to right function keys.  
!  
keycode 28 = R1 R1 Pause  
keycode 29 = R2 R2 Print  
keycode 30 = R3 R3 Scroll_Lock Break  
keycode 52 = R4 R4 KP_Equal  
keycode 53 = R5 R5 KP_Divide  
keycode 54 = R6 R6 KP_Multiply  
keycode 75 = R7 R7 KP_7 Home  
keycode 77 = R9 R9 KP_9 Prior  
keycode 98 = Left R10 KP_4  
keycode 100 = Right R12 KP_6  
keycode 119 = R13 R13 KP_1 End  
keycode 121 = R15 R15 KP_3 Next  
!  
! Reassign the Linefeed key as such and remove from control map.  
!  
remove control = Control_R  
keycode 118 = Linefeed
```

IA: ファンクションキーとコントロールキーの再マッピング

IA マシンのファンクションキーを再マッピングして、SPARC キーボード上にある Help、Cut、Copy、Paste、Undo、および Front などの各キーとして機能させることができます。また右側のコントロールキーを再マッピングして、メタキーにすることもできます。

注 [IA] - この節の以降の説明は、IA ベースのマシンだけに適用されるので注意してください。これらのキーを再マッピングした場合、キーボードの再マッピングを取り消してからでないと `kdmconfig` を使って設定情報やビデオ情報を変更することはできません。

IA: 再マッピングスクリプトの使い方

再マッピングスクリプトを作成し使用する手順は次のとおりです。

1. 任意のテキストエディタを使って、ユーザのホームディレクトリ上に `fkeys` というスクリプトファイルを作成します。
2. 後述の表 C-3 に示すスクリプトを入力します。
3. ファイルを保存して、エディタを終了します。
4. プロンプトに対して、次のコマンドを入力します。

```
$ xmodmap fkeys
```

上記のコマンドは、スクリプトファイル `fkeys` と同じディレクトリ内で入力してください。

5. ワークスペース内でマウスボタンをクリックして、スクリプトの内容を有効にします。

上の手順を実行すると、ファンクションキーを Help キー、Cut キー、Copy キー、Paste キー、Undo キー、および Front キーとして使用することができます。

手順 1 で説明したように、次のスクリプトを `fkeys` ファイルに入力します。

表 C-3 fkeys スクリプト

```
!  
keySYM F2 = L10  
keySYM F3 = L6  
keySYM F4 = L8  
keySYM F5 = L9  
  
keySYM F8 = L4  
keySYM F9 = L5  
  
remove control = Control_R  
keycode 0x47 = Meta_R  
add mod1 = Meta_R
```

IA: キーボードの再マッピングの取り消し

キーボードを元の設定に戻すには、次の 2 つの方法があります。1 つは OpenWindows を一度終了して、再起動します。もう 1 つはキー設定を元に戻すための別のスクリプトを作成して、必要に応じてそのスクリプトを実行します。2 番目の方法は、望ましい方法です。

再設定用のスクリプトを作成するには、次の手順を実行します。

1. 任意のテキストエディタを使って、`normal` というスクリプトファイルを作成します。
このファイルは、`fkeys` スクリプトファイルと同じディレクトリに作成します。
2. 後述の 213 ページの「`normal` スクリプト」に示すスクリプトを入力します。
3. ファイルを保存してエディタを終了します。
4. プロンプトに対して、次のコマンドを入力します。

```
$ xmodmap normal
```

上記のコマンドは、スクリプトファイル `normal` と同じディレクトリ内で入力してください。

手順1で説明したように、次のスクリプトを normal ファイルに入力します。

normal スクリプト

```
keycode 8 = grave asciitilde
keycode 9 = 1 exclam
keycode 10 = 2 at
keycode 11 = 3 numbersign
keycode 12 = 4 dollar
keycode 13 = 5 percent
keycode 14 = 6 asciicircum
keycode 15 = 7 ampersand
keycode 16 = 8 asterisk
keycode 17 = 9 parenleft
keycode 18 = 0 parenright

keycode 19 = minus underscore
keycode 20 = equal plus
keycode 21 =
keycode 22 = BackSpace
keycode 23 = Tab
keycode 24 = Q
keycode 25 = W
keycode 26 = E
keycode 27 = R
keycode 28 = T
keycode 29 = Y
keycode 30 = U
keycode 31 = I
keycode 32 = O
keycode 33 = P
keycode 34 = bracketleft braceleft
keycode 35 = bracketright braceright
keycode 36 = backslash bar brokenbar
keycode 37 = Caps_Lock

keycode 38 = A
keycode 39 = S
keycode 40 = D
keycode 41 = F
keycode 42 = G
keycode 43 = H
keycode 44 = J
keycode 45 = K
keycode 46 = L
keycode 47 = semicolon colon
keycode 48 = apostrophe quotedbl
keycode 49 =
keycode 50 = Return
keycode 51 = Shift_L
keycode 52 =
keycode 53 = Z
keycode 54 = X
```

(続く)

続き

```
keycode 55 = C
keycode 56 = V
keycode 57 = B
keycode 58 = N
keycode 59 = M
keycode 60 = comma less
keycode 61 = period greater
keycode 62 = slash question
keycode 63 =
keycode 64 = Shift_R
keycode 65 = Control_L
keycode 66 =
keycode 67 = Alt_L
keycode 68 = space
keycode 69 = Alt_R
keycode 70 =
keycode 71 = Control_R
keycode 72 =
keycode 73 =
keycode 74 =
keycode 75 =
keycode 76 =
keycode 77 =
keycode 78 =
keycode 79 =
keycode 80 =
keycode 81 =
keycode 82 = Insert
keycode 83 = Delete
keycode 84 =
keycode 85 =
keycode 86 = Left
keycode 87 = Home
keycode 88 = End
keycode 89 =
keycode 90 = Up
keycode 91 = Down
keycode 92 = Prior
keycode 93 = Next
keycode 94 =
keycode 95 =
keycode 96 = Right
keycode 97 = Num_Lock
keycode 98 = Home KP_7 KP_7
keycode 99 = Left KP_4 KP_4
keycode 100 = End KP_1 KP_1
keycode 101 =
keycode 102 = KP_Divide
keycode 103 = Up KP_8 KP_8
keycode 104 = KP_5 KP_5 KP_5
keycode 105 = Down KP_2 KP_2
keycode 106 = KP_Insert KP_0 KP_0
keycode 107 = KP_Multiply
```

(続く)

続き

```
keycode 108 = Prior KP_9 KP_9
keycode 109 = Right KP_6 KP_6
keycode 110 = Next KP_3 KP_3
keycode 111 = Delete KP_Decimal KP_Decimal
keycode 112 = KP_Subtract
keycode 113 = KP_Add
keycode 114 =
keycode 115 = KP_Enter
keycode 116 =
keycode 117 = Escape
keycode 118 =
keycode 119 = F1
keycode 120 = F2
keycode 121 = F3
keycode 122 = F4
keycode 123 = F5
keycode 124 = F6
keycode 125 = F7
keycode 126 = F8
keycode 127 = F9
keycode 128 = F10
keycode 129 = SunF36
keycode 130 = SunF37
keycode 131 = Print SunSys_Req
keycode 132 = Scroll_Lock
keycode 133 = Pause Break
keycode 134 =
keycode 135 = Multi_key
keycode 136 = Mode_switch
```


ネットワークアプリケーションの実行

この付録では、ネットワーク上の他のマシンにあるアプリケーションを実行できる、OpenWindows 環境の高度な機能について説明します。

注・ユーザの大部分はこの付録を参照する必要はありません。ネットワークアプリケーションの可能性を研究してみたい場合は、システム管理者に問い合わせるネットワーク上で実行できる適当なアプリケーションを紹介してもらってください。

通常の OpenWindows 環境では、画面上のすべてのアプリケーション (メールツールやカレンダーマネージャなど) は、ローカルマシン上で実行されるプログラムです。ただし、使用しているワークステーションがネットワークの一部であれば、リモートマシン上でアプリケーションを実行してその結果をローカルマシンの画面に表示できます。この方法でアプリケーションを実行すれば、ローカルマシン上の資源を節約できる上、ネットワーク上のすべてのアプリケーションにアクセスできます。

この付録では、リモートマシン上でアプリケーションを実行して、その結果をローカルマシンの画面に表示する最も簡単な方法を紹介します。使用しているマシンの環境はそれぞれ異なるため、次の説明は柔軟に解釈する必要があります。219 ページの「セキュリティに関する詳細」では、ネットワークアプリケーションを実行する際の複雑さについて解説しています。

次の手順に従ってリモートマシン上のアプリケーションを実行するには、次の要件を満たしている必要があります。

- リモートマシンに対するアクセス権限を持っている
- NFS を使用してホームディレクトリをリモートマシン上にマウントできる

- アプリケーションと適切なライブラリがリモートマシン (ホスト) 上にインストールされている

これらの要件を満たしているかどうか不明な場合は、システム管理者にお問い合わせください。

rlogin を使用してネットワークアプリケーションを実行する

リモートマシン上でネットワークアプリケーションを実行するには、環境変数の設定が適切でなくてはなりません。

- リモートマシン上のシェルの HOME 環境変数を、ローカルユーザのホームディレクトリに設定します。
- リモートマシン上のシェルの DISPLAY 環境変数を、ローカルマシンの画面に設定します。
- OpenWindows ライブラリが標準の /usr/lib または /usr/local という共有ライブラリのディレクトリにインストールされていない場合は、LD_LIBRARY_PATH 環境変数を適切なディレクトリ (/usr/openwin/lib) に設定します。

下記は、rlogin を使用してリモートマシン上でコマンドツールを実行する例です。この例では、ホームディレクトリはリモートマシンの /home/mydirectory にマウントされ、OpenWindows はリモートマシンの /usr/openwin にあります。mydirectory および mymachine という変数は、実際の設定に応じて適当に変更してください。また、cmdtool の部分は任意のアプリケーション名で置き換えることができます。

```
$ rlogin remotemachine
.
.
(以下のコマンドはリモートマシン上で実行される)
.
.
$ HOME=/home/mydirectory
$ DISPLAY=mymachine:0
$ LD_LIBRARY_PATH=/usr/openwin/lib
$ /usr/openwin/bin/cmdtool &
```

最終行を入力すると、コマンドツールのウィンドウがローカルマシンの画面に表示されます。このコマンドツールは、画面上の他のアプリケーションとまったく同様に操作できますが、実際にはリモートマシン上で実行されています。

このような方法でコマンドツールを実行する利点は特にありませんが(コマンドツールはローカルマシンですでに利用可能であり、コンピュータ資源もそれ程使用しません)、この例は、リモートマシン上にあるアプリケーションの利用方法を示しています。

セキュリティに関する詳細

この節では、ネットワーク上でアプリケーションを実行するときに必要と思われるネットワークセキュリティの基本事項について説明します。具体的には次のような事項です。

- ユーザベースおよびホストベースのアクセス制御機構
- 認証プロトコルの MIT-MAGIC-COOKIE-1 と SUN-DES-1
- サーバに対するアクセス制御を変更する方法とその時期
- アプリケーションをリモートで実行 (別のユーザとしてローカルで実行) する方法

対象読者

次のどれかの構成でアプリケーションを実行する場合以外は、OpenWindows バージョン 3.3 以降のデフォルトのセキュリティ構成を変更する必要はありません。

- X11R4 または OpenWindows バージョン 2 より前のバージョンの xlib や libcps を使ってリンクされたアプリケーションを実行する場合
- OpenWindows バージョン 2 ライブラリと静的にリンクされたアプリケーションを実行するときに、SUN-DES-1 認証プロトコルを使用する場合
- リモートサーバ上でアプリケーションを実行する場合

アクセス制御機構

アクセス制御機構は、どのクライアントまたはアプリケーションが X11 サーバにアクセスできるかを制御します。アクセスが許可されたクライアントだけがサーバに

接続できます。アクセスが許可されていないその他のクライアントはすべて次のエラーメッセージを表示して終了します。

```
Xlib: connection to hostname refused by server
Xlib: Client is not authorized to connect to server
```

接続の結果は、次のようにサーバコンソールに表示されます。

```
AUDIT: <月日 時刻 年>: X: client 6 rejected from IP 129.144.152.193 port 3485
Auth name: MIT-MAGIC-COOKIE-1
```

アクセス制御機構には、ユーザベースとホストベースの 2 種類があります。(つまり、前者は特定のユーザのアカウントにアクセスを許可し、後者は特定のホスト、すなわちマシンにアクセスを許可します。) `openwin` とともに `-noauth` オプションを指定しないと、ユーザベースとホストベースのアクセス制御機構が両方ともアクティブ状態となります。詳細は、この付録で後述の 222 ページの「サーバへのアクセスを制御する」を参照してください。

ユーザベース

ユーザベースすなわち認証ベースのアクセス制御機構では、あるユーザに対し、任意のホストマシン上の特定のユーザに対するアクセスが明示的に許可されます。ユーザのクライアントは、サーバに認証データを渡します。そのデータがサーバの認証データと一致すれば、そのホストのすべてのユーザはサーバのアクセスを許可されます。

ホストベース

ホストベースのアクセス制御機構には、汎用性があります。この機構では、特定のホストへのアクセスを許可し、そのホスト上のすべてのユーザがサーバに接続できるようにします。これは、ゆるやかなアクセス制御方式です。ホストにサーバに対するアクセス権が与えられていれば、そのホスト上のすべてのユーザがサーバに接続できます。

Solaris 環境では、ホストベースの制御機構は、下位互換性を目的として使用されます。X11R4 または OpenWindows バージョン 2 より前のバージョンの `xlib` や `libcps` を使ってリンクされたアプリケーションでは、新しいユーザベースのアクセス制御機構に対応できません。これらのアプリケーションがサーバに接続でき

るようにするためには、ホストベース機構に切り換えるか、あるいは新しいバージョンの `xlib` と `libcps` にリンクし直さなければなりません。

注 - 旧バージョンの `xlib` や `libcps` とリンクしたクライアントは、できれば新バージョンのライブラリを使ってリンクし直してください。これにより、新しいユーザベースのアクセス制御機構の下でサーバに接続できるようになります。

認証プロトコル

OpenWindows は、MIT-MAGIC-COOKIE-1 と SUN-DES-1 の 2 種類の認証プロトコルをサポートしています。これらの 2 種類のプロトコルでは、使用される認証データの形式は異なりますが、アクセス制御機構は似ています。サーバに実装できるのは 1 種類のプロトコルに限られます。OpenWindows のデフォルトは、ユーザベースのアクセス制御機構を使用した MIT-MAGIC-COOKIE-1 プロトコルです。

MIT-MAGIC-COOKIE-1

MIT-MAGIC-COOKIE-1 認証プロトコルは、マサチューセッツ工科大学で開発されました。サーバの起動時に、サーバとシステムを起動したユーザについてマジッククッキー (*magic cookie*) が作成されます。接続を要求するたびに、ユーザのクライアントは、接続パケットの一部としてマジッククッキーをサーバに送信します。このマジッククッキーは、サーバのマジッククッキーと比較されます。マジッククッキーが一致すれば接続は許可され、一致しなければ拒否されます。

SUN-DES-1

米国 Sun Microsystems, Inc. が開発した SUN-DES-1 認証プロトコルは、機密保護 RPC (遠隔手続き呼び出し) に基づいており、DES (データ暗号化ソフトウェア) を必要とします。認証情報としては、マシンに依存しないユーザの `netname` (ネットワーク名) が使われます。この認証情報は暗号化され、接続パケットの一部としてサーバに送信されます。サーバは暗号化された認証情報を復号化して、`netname` が登録されているものであれば接続を許可します。

SUN-DES-1 プロトコルは、MIT-MAGIC-COOKIE-1 よりも高水準のセキュリティを提供します。あるユーザが他のユーザの `netname` を使ってそのユーザのサーバにアクセスする方法はありませんが、マジッククッキーを使うと可能な場合もあります。

このプロトコルは、OpenWindows バージョン 3 以降の環境に入っているライブラリでしか使用できません。OpenWindows バージョン 3 より前の環境の静的ライブラリ、特に Xlib で構築されたアプリケーションは、この認証プロトコルを使用できません。

226ページの「SUN-DES-1 を使用する場合のアクセス許可」では、他のユーザの `netname` を自分のサーバのアクセスリストに追加することによって、自分のサーバに対する他のユーザのアクセスを許可する方法について説明しています。

デフォルトの認証プロトコルの変更

デフォルトの認証プロトコル MIT-MAGIC-COOKIE-1 は、システムでサポートされるもう 1 つの認証プロトコル SUN-DES-1 に変更できます。また、ユーザベースのアクセス制御機構をまったく使わないようにすることもできます。デフォルト設定の変更は、`openwin` コマンドのオプションを指定することによって行います。たとえば、デフォルトを MIT-MAGIC-COOKIE-1 から SUN-DES-1 に変更するには、次のオプションを指定して OpenWindows を起動します。

```
example% openwin -auth sun-des
```

ユーザベースのアクセス制御機構を使わないで OpenWindows を実行する場合は、`-noauth` コマンド行オプションを指定します。

```
example% openwin -noauth
```



注意 - `-noauth` オプションを指定した場合、セキュリティの水準は低下します。この設定は、OpenWindows をホストベースのアクセス制御機構だけで実行した場合と同じです (サーバはユーザベースのアクセス制御機構を無効にします)。あるマシンでアプリケーションを実行できるユーザであれば、そのマシンのサーバに対するアクセスを許可されることになります。

サーバへのアクセスを制御する

`openwin` を `-noauth` オプションとともに使用していない場合は (222ページの「デフォルトの認証プロトコルの変更」を参照してください)、ユーザベースとホストベースのアクセス制御機構が両方ともアクティブ状態になります。サーバは最初

にユーザベースの制御機構をチェックした後、ホストベースの制御機構をチェックします。デフォルトのセキュリティ構成では、MIT-MAGIC-COOKIE-1 がユーザベースの制御機構として使われ、空のアクセスリストがホストベースの制御機構として使われます。ホストベースのアクセスリストは空であるため、実質的にはユーザベースの制御機構だけが有効になります。-noauth オプションを指定すると、サーバはユーザベースのアクセス制御機構を無効にし、ホストベースのアクセスリストにローカルホスト名を追加してリストを初期化します。

xhost と xauth の 2 つのプログラムのどちらかを使って、サーバのアクセス制御機構を変更できます。詳細は、マニュアルページを参照してください。これらのプログラムは、認証プロトコルによって作成される 2 つのバイナリファイルにアクセスします。これらのファイルには、各セッション固有の認証データが入っています。一方のファイルは、サーバから内部的に使用するためのもので、もう一方のファイルは下記のファイル名でユーザの \$HOME ディレクトリにあります。

.Xauthority (クライアント認証ファイル)

サーバ上にあるホストベースのアクセスリストを変更するには、xhost プログラムを使用します。これにより、アクセスリストに対するホストの追加と削除ができます。デフォルトの構成 (空のホストベースのアクセスリスト) で OpenWindows を起動した場合は、xhost を使ってアクセスリストにマシン名を追加すると、セキュリティレベルを低下させます。つまりサーバは、デフォルトの認証プロトコルで指定されるユーザの他に、リストに追加されたホストに対してもアクセスを許可します。ホストベースのアクセス制御機構のセキュリティレベルが低いと見なされる理由については、220ページの「ホストベース」を参照してください。

xauth プログラムは、.Xauthority クライアントのファイル内の認証プロトコルデータにアクセスします。アクセスを操作する側のユーザは自分の .Xauthority ファイルからデータを抽出して、他のユーザがそのデータを自分の .Xauthority ファイルにマージできるようにします。これにより、操作側ユーザのサーバまたはそのユーザが接続しているサーバに対して他のユーザがアクセスできるようになります。

xhost と xauth の使用例は、225ページの「MIT-MAGIC-COOKIE-1 を使用する場合のアクセス許可」を参照してください。

クライアント認証ファイル

クライアント用の許可ファイル `.Xauthority` には、次の形式のエントリがあります。

```
connection-protocol      auth-protocol      auth-data
```

デフォルト設定では、`.Xauthority` ファイルに `auth-protocol` として MIT-MAGIC-COOKIE-1 が格納され、ローカル表示に関するエントリが `connection-protocol` および `auth-data` としてのみ格納されます。たとえば、ホスト `anyhost` 上の `.Xauthority` ファイルに次のエントリが格納されている場合が考えられます。

```
anyhost:0      MIT-MAGIC-COOKIE-1  82744f2c4850b03fce7ae47176e75
localhost:0    MIT-MAGIC-COOKIE-1  82744f2c4850b03fce7ae47176e75
anyhost/unix:0 MIT-MAGIC-COOKIE-1  82744f2c4850b03fce7ae47176e75
```

クライアントの起動時に、`connection-protocol` に対応するエントリが `.Xauthority` から読み取られ、`auth-protocol` および `auth-data` が接続パケットの一部としてサーバに送られます。デフォルトの構成では、`xhost` によって、空のホストベースのアクセスリストが戻され、認証が実行可能な状態であることを示します。

認証プロトコルをデフォルトの MIT-MAGIC-COOKIE-1 から SUN-DES-1 に変更した場合、`.Xauthority` ファイルのエントリには、`auth-protocol` として SUN-DES-1 が格納され、`auth-data` としてユーザのネットワーク名 (`netname`) が格納されます。`netname` の形式は次のとおりです。

```
unix.userid@NISdomainname
```

たとえば、ホスト `anyhost` 上で `.Xauthority` ファイルに次のエントリが格納されている場合が考えられます。この `unix.15339@EBB.Eng.Sun.COM` は、マシンに依存しないユーザの `netname` です。

```
anyhost:0      SUN-DES-1           'unix.15339@EBB.Eng.Sun.COM'
localhost:0    SUN-DES-1           'unix.15339@EBB.Eng.Sun.COM'
anyhost/unix:0 SUN-DES-1           'unix.15339@EBB.Eng.Sun.COM'
```

注 - 自分のネットワーク名、またはマシンに依存しないネットワーク名が不明な場合は、システム管理者に問い合わせてください。

MIT-MAGIC-COOKIE-1 を使用する場合のアクセス許可

MIT-MAGIC-COOKIE-1 認証プロトコルを使用している場合は、次の手順により自分のサーバに他のユーザがアクセスできるようにすることができます。

1. サーバを実行しているマシン上で `xauth` を実行し、`hostname:0` に対応するエントリを抽出して、ファイルに入れます。

例として、`hostname` が `anyhost`、ファイルが `xauth.info` の場合を示します。

```
myhost% /usr/openwin/bin/xauth nextract - anyhost:0 > $HOME/xauth.info
```

2. アクセスを要求しているユーザに、手順 1 で作成したエントリが入っているファイルを送ります。(mailtool、rcp、またはその他のファイル転送プログラムを使用します。)

注 - 認証情報が入っているファイルの転送には、`rcp` よりもメールを使用するほうがより安全です。`rcp` を使用する場合は、他のユーザが簡単にアクセスできるディレクトリにそのファイルを入れないようにしてください。

3. 他のユーザは、エントリを自分の `.Xauthority` ファイルにマージしなければなりません。

例として、`userhost` が `xauth.info` を自分の `.Xauthority` ファイルにマージする場合を示します。

```
userhost% /usr/openwin/bin/xauth nmerge - < xauth.info
```

注 - `auth-data` は特定のセクションに対応します。したがって、サーバが再起動されるまでの間、有効です。

SUN-DES-1 を使用する場合はアクセス許可

SUN-DES-1 認証プロトコルを使用している場合は、次の手順により自分のサーバに他のユーザがアクセスできるようにすることができます。

1. サーバを実行しているマシン上で `xhost` を実行しサーバに新規ユーザを通知します。

例として、新規ユーザ `somebody` に `myhost` 上での実行を許可する場合は示します。

```
myhost% xhost + somebody@
```

2. 新規ユーザは、`xauth` を使用して自分の `.xauthority` ファイルにエントリを追加します。

例として、新規ユーザのマシンに依存しない `netname` が、`unix.15339@EBB.Eng.Sun.COM` の場合を示します。下記のコマンドは、途中で Return キーなどを入れずに 1 行に入力しなければならないことに注意してください。パイプ記号の後に、空白と続いてコマンドの残りの部分を入力します。

```
userhost% echo 'add myhost:0 SUN-DES-1 `unix.15339@EBB.Eng.Sun.COM`' |  
$OPENWINHOME/bin/xauth
```

クライアントをリモートで実行する場合とローカルで実行する場合

X クライアントは、環境変数 `DISPLAY` の値を使用して接続するサーバ名を取り出します。

クライアントをリモートで実行するか、または他のユーザとしてローカルで実行するには、次の手順を実行します。

1. サーバを実行しているマシン上で、他のユーザのアクセスを許可します。

使用する認証プロトコルに応じて、225ページの「MIT-MAGIC-COOKIE-1 を使用する場合はアクセス許可」または 226ページの「SUN-DES-1 を使用する場合はアクセス許可」で説明した手順を実行します。

2. 環境変数 `DISPLAY` の値として、サーバを実行しているホスト名に設定します。
例として、ホスト名が `remotehost` の場合を示します。

```
myhost% setenv DISPLAY remotehost:0
```

3. クライアントプログラムを実行します。

```
myhost% client_program&
```

クライアントは、リモートマシン `remotehost` 上に表示されます。

SPARC — DECnet インターネットワーキング (DNI)

この付録では、NSP DECnet トランスポートプロトコルを使って、OpenWindows 環境と DECwindows™ 環境をインターネットワークで接続する方法について説明します。

注 **[SPARC]** - この章全体の説明は、SPARC ベースのマシンにだけ適用されることに注意してください。DECnet インターネットワーキングは、8.x DNI でだけ利用できます。

DNI には次の 2 通りの使い方があります。

- VMS オペレーティングシステムで稼働する VAX システム上で X11 クライアントを実行し、OpenWindows マシン上で X11 クライアントのウィンドウを表示する場合
- OpenWindows マシン上で X11 クライアントを実行し、VAX システム上で X11 クライアントのウィンドウを表示する場合

この付録ではまず、2 つの方法について DNI ソフトウェアの設定方法を説明した後、各方法について詳しく説明します。

DECnet インターネットワーキングの設定

DECnet インターネットワーキングを設定するには、次の手順を実行します。

1. **DNI** 経由の接続を指定します。

OpenWindows サーバとクライアントのライブラリでは、DNI トラnsポートライブラリ libdni の中で動的にロードできるバージョンを使用します。

OpenWindows サーバとクライアントのライブラリが libdni をロードできるようにするには、libdni.so がインストールされたディレクトリを指すように、環境変数 DNI_X_ENABLE を設定してください。

次の例では、pkgadd を使って DNI をデフォルトの位置にインストールしていません。

```
$ DNI_X_ENABLE=/opt/SUNWconn/dni/lib
```

2. **OpenWindows** サーバを起動します。

デフォルトでは、OpenWindows サーバは「MIT-MAGIC-COOKIE」セキュリティ機構がサポートします。このセキュリティ機構は、ホストベースではなくユーザベースです (つまり、サーバに接続できるマシンを指定するのではなく、サーバに接続できるユーザを指定します)。デフォルトモードでは、xhost コマンドは空のアクセスリストを戻し、セキュリティ機構だけが使用可能である旨を通知します。openwin コマンドを `-noauth` オプションとともに指定することによって、デフォルトのセキュリティ機構を無効にして、従来のバージョンの OpenWindows サーバのセキュリティ機構に戻すこともできます。

```
$ openwin -noauth
```

3. **X11** クライアントを **OpenWindows** サーバに接続する許可を、**DEC VAX** システムに与えるため、**OpenWindows** を実行しているマシンの所有者に対して、**xhost** コマンドを実行するよう依頼します。

DNI を使って X11 クライアントを OpenWindows サーバに接続するには、DECnet ノードのアドレスを DECnet ノード名に割り当てる必要があります。そのために、NCP データベースを作成して初期化します。この操作は DEC VAX システムに対しても実行する必要があります。

```
$ xhost decvax::
```

2 つのコロンは DECNet トランスポートを表します。

OpenWindows のマシン上でリモートクライアントを表示する

SunLink DNI の `dnilogin` コマンドを使って VAX システムにログインすることによって、VMS から X11 クライアントを実行できます。ログイン後まず最初に、ローカルマシンがリモートマシンの X11 サーバになるように、ローカルマシン (VAX) 上の環境変数 `DISPLAY` を設定します。次に、X11 クライアントを実行するためにクライアント名 (この例では `x11_client`) を入力します。VMS オペレーティングシステムの使い方について詳細は、『VMS DECwindows User's Guide, Running Applications Across the Network』を参照してください。

リモートクライアントの実行例

```
$ dnilogin decvax
.
.
.
$ define DECW$DISPLAY OW_machine::0
$ spawn/nowait run x11_client
```

VAX のマシン上でリモートクライアントを表示する

リモートの VAX システムを指すように環境変数 `DISPLAY` を設定すると、OpenWindows マシン上で X11 クライアントを実行して、DECwindows サーバ上に表示できます。

X11 クライアントをこの方法で実行するためには、DECwindows サーバ上で OpenWindows フォントのコンパイルとインストールを実行する必要があります。これらのフォントは、X11R5 リリースまたは OpenWindows に付属のオプションフォントパッケージで提供されています。DECwindows サーバに適切なフォントをインストールするには、次の手順を実行します。

1. **OpenWindows** のオプションフォントソース (または **X11R5** リリースのフォントソース) を **OpenWindows** マシン上にインストールします。

2. フォントのインストールについて詳細は、『**Solaris X Window System** 開発ガイド』を参照してください。

3. **VAX** システム上のディレクトリにフォントソースをコピーします。

```
$ cd /usr/openwin/share/src/fonts/misc
$ dnicp *.bdf `decvax::[vaxdir]`
```

4. **VAX** システム上でカーソル用のフォントをコンパイルします

コンパイルの結果、`olcursor.decw$font;1`、`olglyph10.decw$font;1` などのファイルが作成されます。

```
$ font olcursor.bdf
$ font olglyph10.bdf
...
```

5. **VAX** システム上の `sysfont` ディレクトリにフォントをコピーします。

```
$ set def sys$sysroot:[sysfont.decw.user_cursor16]
$ copy [vaxdir]olcursor.decw$font;1 *
```

注 - `sysfont` ディレクトリにフォントをコピーするには、DEC VAX システムに「system」としてログインする必要があります。

6. `/usr/openwin/share/src/fonts/75dpi` と `/usr/openwin/share/src/fonts/100dpi` にある残りのカーソル用フォントおよび **Lucida** フォントについても、上記の手順 2~5 を実行します。

注 - **Lucida** フォントは、`sys$sysroot:[sysfont.decw.user_75dpi]` および `sys$sysroot:[sysfont.decw.user_100dpi]` にインストールしてください。

OpenWindows DeskSet ツールを実行するために最低限必要なフォントセットを次に示します。デフォルトフォントを使ってアプリケーションを実行する場合

は、これらのフォントをインストールするだけで十分です。ただし、必要に応じてこれ以外のフォントをインストールすることもできます。

- olcursor.bdf
- olglyph10.bdf
- olglyph12.bdf
- olglyph14.bdf
- olglyph19.bdf
- luBS08.bdf
- luBS10.bdf
- luBS12.bdf
- luBS14.bdf
- luRS08.bdf
- luRS10.bdf
- luFS12.bdf
- lutBs12.bdf
- lutRS10.bdf
- lutRS12.bdf

7. **DECwindows** サーバを再起動します。
8. フォントがインストールされたことを確認するために、**DECwindows** サーバで利用できるフォントのリストを表示します。

```
§ DISPLAY=decvax:0
§ xlsfonts | grep Sun   (カーソル用フォント名)
§ xlsfonts | grep Lucida
```

9. **DECwindows** セッションマネージャの **Security** メニューを使って、**OpenWindows** のノードに **DECwindows** サーバ上で画面を表示する許可が与えられていることを確認します。
10. **OpenWindows DeskSet** ツールなどの **X11** アプリケーションを実行します。

```
$ DISPLAY=decvax::0
$ mailtool
```

注 - DNI トランスポートライブラリ `libdni` が存在するディレクトリを指すように、環境変数 `DNI_X_ENABLE` を設定しておく必要があります。詳細は、229ページの「DECnet インターネットワーキングの設定」の手順 1 を参照してください。

次に示すようなエラーメッセージが出力される場合は、指摘されたフォントを DECwindows サーバにインストールしてからアプリケーションを実行してください。

```
'XView warning: Cannot load font  '-b&h-lucida-medium-r-*-*-*80-*-*-*-*-*-*-*-*
*' (Font package)'
```

このエラーメッセージは、`luRS10.bdf` フォントをインストールする必要があることを示しています。

フォントについての詳細は、『*Solaris X Window System 開発ガイド*』を参照してください。

システムの管理

Solaris 2.5 にはシステム管理ツール (Administration Tool) の新しいバージョンが入っています。この Administration Tool はいくつかの管理作業を実行するためのグラフィカルユーザインターフェースで、次の機能を備えています。

- ユーザアカウントの管理—Administration Tool を使って、ユーザアカウントを追加、削除、修正できます。システムの `/etc/passwd` ファイルを適切に変更します。
- グループの管理—Administration Tool を使って、グループを追加、削除、修正できます。システムの `/etc/group` ファイルを適切に変更します。
- ホストの管理—Administration Tool を使って、ホストを追加、削除、修正できます。システムの `/etc/inet/hosts` ファイルを適切に変更します。
- プリンタの管理—Administration Tool を使って、プリンタへのアクセスを追加、削除、またはシステムのプリンタ設定を変更できます。システムの `/etc/lp` ディレクトリを適切に変更します。
- シリアルポートサービスの管理—Administration Tool を使って、シリアルポートサービスを機能設定および設定解除できます。共通の端末やモデムを構成するためのテンプレートが用意してあるので、シリアル・ポートに接続したモデムや端末を使用するのに必要なソフトウェアサービスを設定できます。
- ソフトウェアの管理—Administration Tool を使って、ソフトウェアを追加、削除できます。プロダクト用 CD またはハードディスクから、インストール済みの実行中のシステムにソフトウェアを追加します。また、インストール済みの実行中のシステムからソフトウェアを削除します。

注 - Administration Tool はローカルシステム、つまり Administration Tool を実行しているシステム上でファイルを変更します。NIS または NIS+ などのグローバルネットワークデータベースを変更または更新しません。

Administration Tool の起動

ここでは Administration Tool の起動についての基本事項と、Administration Tool を使ってホスト、プリンタ、シリアルポートなどを管理するときに役立つ情報について説明します。

Administration Tool を使ってまず実行しなくてはならないのは、UNIX `sysadmin` グループのメンバー登録です (`sysadmin` グループにグループ ID 番号として 14 が指定されているため、グループ 14 とも呼ばれます)。`sysadmin` グループのメンバーとして登録されると、スーパーユーザでログインするのではなく通常のユーザアカウントでログインし、Administration Tool を使ってローカルシステムの管理作業を行うことができます。

ユーザ自身を `sysadmin` グループのメンバーとして登録するには、237ページの「`sysadmin` グループへのユーザ登録」の手順に従ってください。

次の手順により、Administration Tool ソフトウェアを起動します。

1. システムにログインします。

2. `root` になります。

UNIX `sysadmin` グループ (GID 14) のメンバーでない限り、Administration Tool を使用するには `root` にならなければなりません。`root` とは、システムファイルを変更できる特別な許可を持つシステムユーザのことです。

次のように、`su` コマンドを使って `root` になります。

```
$ su
Password: (root のパスワードを入力)

#
```

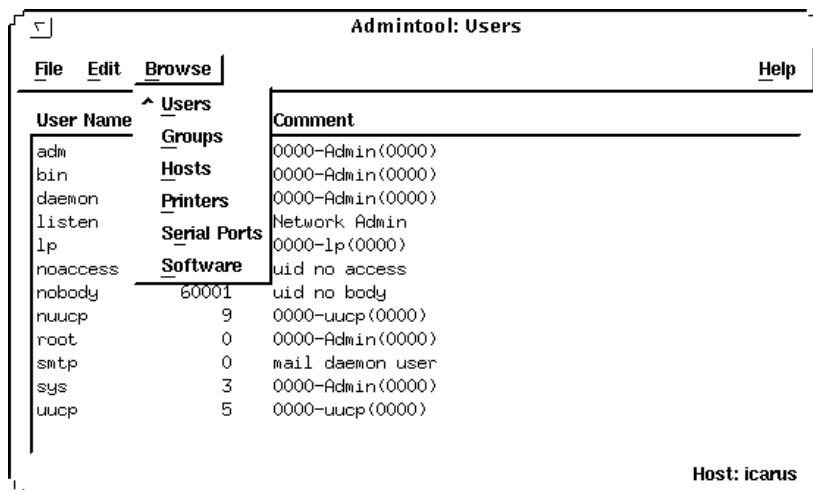
ルートアカウントにパスワードが設定されている場合は、そのパスワードを知っておく必要があります。ルートアカウントにパスワードが設定されていない場合

は、パスワードのプロンプトで Return キーだけを押します。パスワードを知らない場合は、システム管理者に問い合わせてください。

3. **Administration Tool** を起動します。

```
# Admintool &
```

4. **sysadmin** グループのメンバーになっていない場合は、237ページの「**sysadmin** グループへのユーザ登録」を参照してください。
5. 「**ブラウズ (Browse)**」メニューオプションを使って、**Administration Tool** で実行したい作業の種類を選択します。次に示すように、「**ユーザ (Users)**」、「**グループ (Groups)**」、「**ホスト (Hosts)**」、「**プリンタ (Printers)**」、「**シリアルポート (Serial Ports)**」、「**ソフトウェア (Software)**」を選択できます。

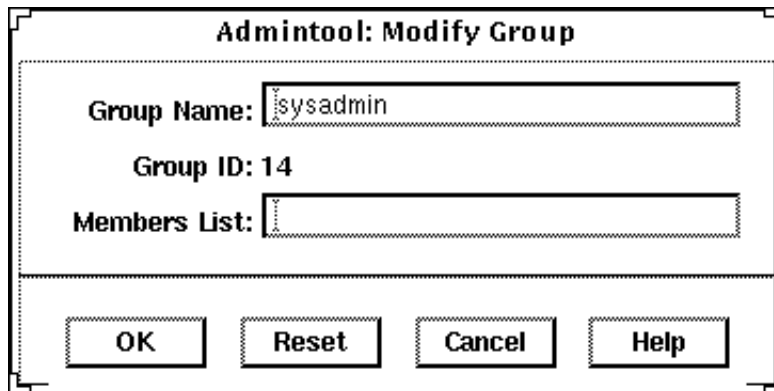


sysadmin グループへのユーザ登録

次の手順により、**sysadmin** グループにユーザ登録します。

1. システムにログインして **root** になります。

2. **Administration Tool** を起動します。
3. 「ブラウズ (**Browse**)」メニューから「ブラウズ (**Groups**)」を選択します。
4. 「**admintool: グループ (Groups)**」ウィンドウのグループ名の中から「**sysadmin**」をクリックします。
5. 「編集 (**Edit**)」メニューから「変更 (**Modify**) ...」を選択します。
6. ユーザアカウント名を「メンバーリスト (**Members List**)」に登録します。



注 - メンバーリストに名前を入力するときは、たとえば cathy,brett,joan のように、名前と名前の間は空白を空けずカンマで区切ります。

7. 「了解 (**OK**)」をクリックします。
これでユーザは sysadmin グループに追加されます。sysadmin グループのメンバーとして登録すると、スーパーユーザとしてログインしなくても Administration Tool を起動してシステムを管理できます。

共通作業に Administration Tool を使用する

Administration Tool を使ってほとんどのローカルシステム管理作業を行えます。Administration Tool にはオンラインヘルプシステムが含まれており、通常作業を実施するときに起こる基本的な問題を解決する手がかりを与えてくれます。しかし一部の作業には、特別な要件があるものや、周辺装置の設定が必要となるものもあります。この節では、特別な知識や設定が必要となる作業について、その概要を説明します。

ホストの管理

Administration Tool を使って、アクセスしたいリモートシステムを定義できます。Administration Tool を使ってホストを追加することにより、そのホスト名を使って他のシステムに遠隔でログインできます (/etc/hosts ファイルにエントリがない場合は、ホストの IP アドレスを使って他のシステムに遠隔ログインする必要があります)。



注意 - Administration Tool を使ってユーザのローカルシステムにホストを追加し、ユーザサイトで NIS または NIS+ などのネットワーク名サービスを使用している場合は、Administration Tool で追加したホストが稼動しても、希望した結果が得られないことがあります。これは、ネットワーク名サービスにある情報がローカルの /etc/hosts ファイル (このファイルで Administration Tool は情報を更新します) にある情報より優先して扱われるためです。Administration Tool による情報取得の動作を、ネットワーク名サービスデータベース情報の取得より優先させたい場合は、システム管理者に問い合わせてください。

プリンタの管理

Administration Tool を使うことにより、システムに接続されたプリンタ、またはユーザサイトのネットワークで使用可能なプリンタにアクセスできます。プリンタをシステムに接続し、Administration Tool を使用する前または後に電源をオンにすることにより、そのプリンタにアクセスできます。プリンタをシステムに接続するのに必要な手順は、通常、次のとおりです。

- プリンタとシステムを物理的に接続します。

- 必要なスイッチを設定し、必要に応じてプリンタのボーレート、ポートなどを設定します。(スイッチの設定およびケーブル設置の要件について詳細は、プリンタに付属のマニュアルおよびシステムのハードウェアインストールマニュアルを参照してください)。プリンタケーブルは通常、シリアルポートに接続しますが、使用するプリンタの仕様によってはパラレルポートに接続するものもあります。
- プリンタを電源コンセントに接続します。
- システムにログインして root になります。
- Administration Tool を起動して「ブラウズ (Browse)」メニューから「プリンタ (Printers)」を選択し、プリンタを使用するのに必要なローカルシステムファイルを更新します。

プリンタのインストールと管理について詳細は、『Solaris のシステム管理 (第 2 巻)』を参照してください。

シリアルポートの管理

モデムは電話回線を経由してシステム間で情報を送受信できる装置です。モデムは通常、シリアルポートに接続します。したがって、モデムを使用するには、Administration Tool を使ってシステムのシリアルポートを設定する必要があります。モデムをシステムに接続し、Administration Tool を使用する前または後に電源をオンにするとモデムにアクセスできます。モデムをシステムに接続する手順は、正常、次のとおりです。

- モデムとシステムを物理的に接続、またはインストールします(モデムカードの場合)。
- 必要なスイッチを設定し、必要に応じてモデムのボーレート、ポートなどを設定します(これらの作業の実行に必要な情報については、モデムに付属のマニュアルおよびシステムのインストールマニュアルを参照してください)。
- 必要に応じて、モデムまたはそのアダプタを電源コンセントに接続します。
- システムにログインします。
- Administration Tool を起動して「ブラウズ (Browse)」メニューから「シリアルポート (Serial Ports)」を選択し、モデムを使用するのに必要なローカルシステムファイルを更新します。

モデムのインストールおよび設定について詳細は、『Solaris のシステム管理 (第 2 巻)』を参照してください。

PCMCIA カードの使い方

はじめに

パーソナルコンピュータメモリカード・インターナショナルアソシエーション (PCMCIA) カードは、頑丈なクレジットカードサイズのユーザがインストールできるデバイスです。

PCMCIA メモリカードはフロッピーディスクと同じように使用できますが、PCMCIA メモリカードの方がより大量のデータを格納できます。

PCMCIA シリアルモデムカードを使うと、RS-232 インタフェースまたはデータ/FAX モデムの機能を SPARCstation™ に簡単に追加できます。

PCMCIA カードは多くのベンダーから入手できます。SunService または PCMCIA カードベンダーに問い合わせて、取り扱っているカードが SPARCstation と互換性があるか確認してください。

この付録は、次の事項について説明しています。

- 242ページの「PCMCIA メモリカードの使い方」
- 245ページの「tar コマンドによるファイルのコピー」
- 251ページの「ボリューム管理の設定によるファイルのコピー」
- 257ページの「ボリューム管理の設定解除によるファイルのコピー」
- 261ページの「PCMCIA シリアル/モデムカードの使い方」

サポート要件

PCMCIA メモリカードおよび PCMCIA シリアル/モデムカードは、PCMCIA リリース 2.1 以降のバージョンに準拠しているカードを、次のプラットフォームでサポートしています。

- SPARCstation Voyager™ システム
- PCMCIA インタフェース/SBus カードを持つすべての SPARCstation

この Solaris リリースでは、SRAM (不揮発性スタティックランダムアクセスメモリ)、DRAM (ダイナミックランダムアクセスメモリ)、MROM (メモリ読み取り専用メモリ)、PCMCIA メモリカードをサポートしています。

注 - このリリースでは、PCMCIA FLASH、EEPROM および OTP (One-Time Programmable) PROM カードはサポートしていません。

特にこのソフトウェアリリースでは、次の機能を備えている PCMCIA メモリカードをサポートしています。

- カード情報構造 (CIS)
- 512 キロバイトから 64 メガバイトまでの記憶密度
- MS-DOS ファイルシステムおよび UNIX ファイルシステム

参考資料

システムの Solaris ソフトウェアに付属のハードウェアプラットフォームに関するマニュアルなどを参照して、使用したい PCMCIA デバイスに特別なインストール手順がないか確認してください。

PCMCIA メモリカードの使い方

PCMCIA メモリカードの一般的な使用方法、たとえば、書き込み保護モードを設定することでカードデータを誤って削除しないようにする方法などについて説明します。書き込みモードの設定方法については、243ページの「書き込み保護モードを設定するには」を参照してください。

ファイルのコピー方法

後述の3つの節では、ファイルのフォーマット方法、PCMCIA メモリカードからハードディスクへのファイルのコピー方法、ハードディスクから PCMCIA メモリカードへのファイルのコピー方法について説明します。次の3つの方法があります。

- tar、cpio、または dump/restore コマンド

tar コマンドを使ってファイルをコピーする場合は、245ページの「tar コマンドによるファイルのコピー」へ進んでください。(cpio または dump/restore コマンドの使用方法について詳細は、マニュアルページを参照してください)

- ボリューム管理の設定

ボリューム管理を設定してファイルをコピーする場合は、251ページの「ボリューム管理の設定によるファイルのコピー」へ進んでください。

- ボリューム管理の設定解除

ボリューム管理を設定解除してファイルをコピーする場合は、257ページの「ボリューム管理の設定解除によるファイルのコピー」へ進んでください。

書き込み保護モード

書き込み保護モードを設定することにより、PCMCIA メモリカードのデータを誤って削除しないようにすることができます。いったん PCMCIA メモリカードに書き込み保護モードを設定すると、書き込み保護モードの設定を解除するまでデータのコピーはできません。

書き込み保護モードを設定するには

- ◆ ドライバなどの先端が細くとがった工具を使って、図 G-1 に示すように、書き込み保護スイッチを **PCMCIA** メモリカードのエッジ方向へスライドさせ、書き込み保護モードを設定します。

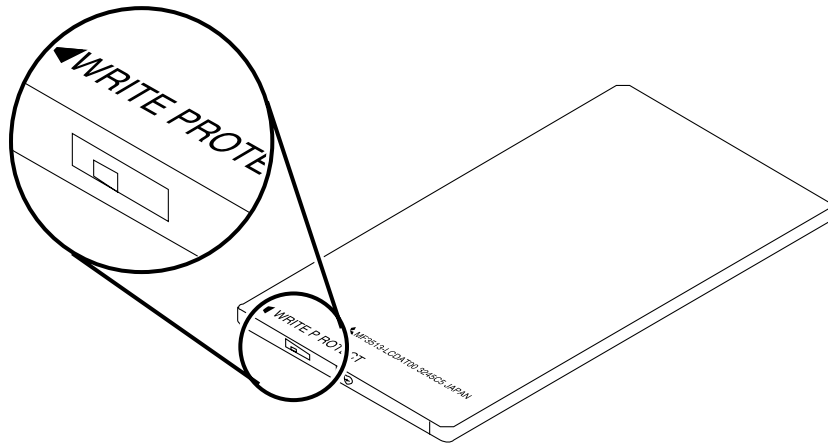


図 G-1 書き込み保護モードの設定

書き込み保護モードを設定解除するには

- ◆ ドライバなど先端が細くとがった工具を使って、書き込み保護スイッチを **PCMCIA** メモリカードのエッジから反対方向へスライドさせ、書き込み保護モードの設定を解除します。

PCMCIA メモリカードと電源管理システムの保存停止・復元再開機能

PCMCIA カードと電源管理システムソフトウェアをインストールしたシステムについて説明します。電源管理システムの詳細については、『電源管理システムユーザーマニュアル』を参照してください。



注意 - システムの中断時または再起動時には、PCMCIA カードを挿入したり取り出したりしないでください。PCMCIA カードは再起動動作後に識別されなくなります。

PCMCIA メモリカードがボリューム管理で管理されていて、システムが中断している場合、PCMCIA メモリカードは自動的に取り出されません。システムが再起動しても、PCMCIA メモリカードは自動的に装着されません。PCMCIA メモリカードを再装着したい場合は、手動で装着してください。

システムの中断時にボリューム管理が設定解除されていて、PCMCIA カードを手動で装着した場合、PCMCIA メモリカードは自動的に取り出されます。したがって、

システムが再起動しても、PCMCIA メモリカードは自動的に装着されません。
PCMCIA メモリカードを装着するときは、手作業で取り付けなければなりません。

tar コマンドによるファイルのコピー

ファイルフォーマット方法、および PCMCIA メモリカードからハードディスクへまたはハードディスクから PCMCIA メモリカードへファイルをコピーする方法について最初に説明します。

注 - cpio または dump/restore コマンドの使用方法については、マニュアルページを参照してください。

ここでは、次の作業について説明します。

- PCMCIA メモリカードのフォーマット
- ファイル名の表示
- ファイルのコピー

▼ PCMCIA メモリカードをフォーマットするには

PCMCIA メモリカードを使用する前にフォーマットする必要があります。
fdformat ユーティリティを使用すると、フロッピーディスクと PCMCIA メモリカードの両方をフォーマットできます。



注意 - フォーマットすると、PCMCIA メモリカードに既に入っていたデータはすべて削除されます。

- ◆ **PCMCIA** メモリカードをフォーマットするには、次のコマンドを入力します。

```
% fdformat option device_name
```

注 - format ユーティリティは、PCMCIA メモリカードでは使用できません。
fdformat ユーティリティだけ使用できます。

表 G-1 に `fdformat` ユーティリティで使用できるオプションを示します。

表 G-1 `fdformat` ユーティリティオプション

オプション	説明
<code>-U</code>	PCMCIA メモリカードを取り外す
<code>-d</code>	MS-DOS ファイルシステムをインストールする (デフォルトは UNIX ファイルシステム)
<code>-f</code>	フォーマット開始前に確認メッセージを表示しない
<code>-q</code>	印刷状態のメッセージを表示しない
<code>-x</code>	Solaris ラベルまたは MS-DOS ファイルシステムをインストールする。PCMCIA メモリカードはフォーマットしない。
<code>-b label</code>	PCMCIA メモリカードに対して UNIX または MS-DOS ラベルを指定する
<code>-t dos</code>	MS-DOS ファイルシステムをインストールする (デフォルトは UNIX ファイルシステム)
<code>-B filename</code>	特殊なブートローダをインストールする

注 - `fdformat` ユーティリティには、PCMCIA メモリカードに NEC-DOS ファイルシステムをインストールするためのオプションはありません。

PCMCIA メモリカードをフォーマットする場合、PCMCIA メモリカードにデバイス名を指定しなければなりません。この指定がないと、`fdformat` ユーティリティはデフォルトのデバイスとして、フロッピーディスクドライブを自動的に指定します。

PCMCIA メモリカードのデバイス名の指定形式は、次のいずれかを使用します。

```
/dev/rdisk/cntndnsn
```

```
/dev/dsk/cntndnsn.
```

たとえば、デバイス名 `/dev/dsk/c1t6d0s2` は、論理ソケットコントローラ番号が 1、技術番号が 6、スライス番号が 2 の PCMCIA SRAM メモリカードを示しています。

表 G-2 に fdformat ユーティリティで使用可能なデバイス名オプションを示します。

表 G-2 fdformat ユーティリティのデバイス名オプション

デバイス名オプション	説明
<i>n</i>	10 進数を表す
<i>cn</i>	<i>n</i> はコントローラ番号を表す
<i>tn</i>	<i>n</i> は技術領域番号を表す 0x1 ROM、0x2 OTPROM、0x3 EPROM、 0x4 EEPROM、0x5 FLASH、0x6 SRAM、0x7 DRAM
<i>dn</i>	<i>n</i> は技術領域のタイプを表す
<i>sn</i>	<i>n</i> はスライスを表す (このリリースでは、PCMCIA メモリカードの 1 つのパーティションだけサポートしています。したがって、デバイス名のパーティション番号 <i>sn</i> は s2 でなければなりません)

▼ tar コマンドを使ってファイル名を表示するには
PCMCIA メモリカードに格納されているファイル名を表示したい場合があります。

◆ 次のコマンドを入力します。

```
% tar tvf device_name
```

表 G-3 ファイル名を表示する tar コマンドオプション

オプション	説明
t	PCMCIA メモリカードに格納されているファイルを一覧表示または画面表示する
v	冗長リストを指定する
f	入力デバイス名を指定する

ファイルのコピー

PCMCIA メモリカードからハードディスクへ、またはハードディスクから PCMCIA メモリカードへ、ファイルまたはディレクトリをコピーするには、PCMCIA メモリカードがフォーマット済みでなければなりません (245ページの「PCMCIA メモリカードをフォーマットするには」を参照してください)。また、ハードディスクの宛先ディレクトリの書き込み権を得ていなければなりません。

次の手順でファイルまたはディレクトリをコピーすると、そのファイルまたはディレクトリの所有者、アクセス権、グループ、変更時刻が保存されます。

注 - tar コマンドを使ってファイルを PCMCIA メモリカードにコピーするときは、tar コマンドを使って、PCMCIA メモリカードからファイルを抽出またはコピーしなければなりません。

ハードディスクから PCMCIA メモリカードへファイルをコピーするには

1. 次のコマンドを入力します。

```
% cd directory
```

上記の *directory* は、コピーしたいファイルが存在するディレクトリ名です。
たとえば、ファイル /home/samples/design をコピーするには、次のコマンドを入力します。


```
% cd /home/samples
```

2. 次のコマンドを入力します。

```
% tar cvf device_name filename
```

表 G-4 ファイルを PCMCIA メモリカードにコピーする tar コマンドのオプション

オプション	説明
c	バックアップ用アーカイブを作成する
v	冗長リストを表示する
f	入力デバイス名を指定する

たとえば、ハードディスクにあるファイル `./design` を、デバイス名 `/dev/rdisk/c1t6d0s2` の PCMCIA メモリカードにコピーするには、次のコマンドを入力します。

```
% tar cvf /dev/rdisk/c1t6d0s2 ./design
```

PCMCIA メモリカードからハードディスクへファイルをコピーするには

1. 次のコマンドを入力します。

```
% cd directory
```

上記の `directory` は、コピーしたいファイルが存在するディレクトリ名です。この場合、ディレクトリは PCMCIA メモリカードにあります。

たとえば、ファイル `/home/samples/design` をコピーするには、次のコマンドを入力します。

```
% cd /home/samples
```

2. 次のコマンドを入力します。

```
% tar xvpf device_name filename
```

表 G-5 ファイルを PCMCIA メモリカードからコピーする tar コマンドオプション

オプション	説明
x	バックアップアーカイブを抽出する
v	冗長リストを表示する
f	入力デバイス名を指定する
p	ファイルのオリジナルモードを保持する



注意 - 現在コピー中のものと同じ名前のファイルまたはディレクトリが、作業ディレクトリに既に存在する場合は、そのファイルまたはディレクトリは自動的に上書きされます。

たとえば、アドレスが `/dev/rdisk/c1t6d0s2` の PCMCIA メモリカードから、ハードディスクへファイル `./design` をコピーするには、次のコマンドを入力します。

```
% tar xvpf /dev/rdisk/c1t6d0s2 ./design
```

ボリューム管理の設定によるファイルのコピー

PCMCIA メモリカードのフォーマット方法、およびメモリカードとハードディスク間のファイルのコピー方法について、2つ目の方法を説明します。ここでは、ボリューム管理がシステムで設定されている場合の作業方法について説明します。

注 - PCMCIA メモリカードは、ボリューム管理により自動的に管理されます。したがって、システムから PCMCIA メモリカードへファイルをコピーするために、スーパーユーザになる必要はありません。



注意 - PCMCIA メモリカードをソケットから取り外し、そのカードを別のソケットに挿入する場合は、少なくとも 2～3 秒間待ってください。待たないと、ボリューム管理が PCMCIA メモリカードを正しく装着できない場合があります。ボリューム管理が PCMCIA メモリカードを正しく装着できない場合は、必要に応じて `vold` デーモンを再起動するか、またはシステムをリブートします。また、PCMCIA メモリカードをソケットに挿入した後、すぐに取り外す場合、ボリューム管理はそのソケットを識別することができません。この問題の場合も同様に、必要に応じて `vold` デーモンを再起動するか、またはシステムをリブートして対処します。

次の各事項を十分に理解してください。

- ボリューム管理でサポートされる PCMCIA メモリカードは 1 つだけです。
- `volcheck(1)` コマンドは、PCMCIA メモリカードをサポートします。
- PCMCIA ハードウェアは手動機構を使用します。`eject(1)` コマンドを使用した場合、PCMCIA メモリカードを手動で取り出せるので、ボリューム管理はファイルシステムのマウントを解除することができます。
- `filemgr(1)` は、PCMCIA メモリカードが挿入されている場合、ファイルマネージャのポップアップメニューを表示しません。ただしディレクトリ名を表示して `/pcmem/pcmemn` ディレクトリ (n は PCMCIA ソケット番号) でファイルを操作できます。

この節では、次の作業について説明します。

- ラベルなし PCMCIA メモリカードのフォーマット
- PCMCIA メモリカードの再フォーマット
- PCMCIA メモリカードの取り付け

- ファイルのコピー
- PCMCIA メモリカードの取り出し

▼ ラベルなし PCMCIA メモリカードをフォーマットするには

- PCMCIA メモリカードが既にフォーマットされている場合は、先に進んでください。
- フォーマット済みの PCMCIA メモリカードを再フォーマットする場合は、253 ページの「PCMCIA メモリカードを再フォーマットするには」へ進んでください。

注 - `format` ユーティリティは、PCMCIA メモリカードでは使用できません。`fdformat` ユーティリティだけ使用できます。

1. **PCMCIA** メモリカードを **PCMCIA** ソケットに挿入します。
2. コマンドを入力して、希望するファイルシステム (**UNIX** または **MS-DOS**) でメモリカードをフォーマットします。
UNIX ファイルシステムを使ってメモリカードをフォーマットするには、次のコマンドを入力します。

```
% fdformat vol_alias_device_name  
% newfs /vol/dev/aliases/vol_alias_device_name
```

たとえば、UNIX ファイルシステムを使って、PCMCIA ソケット番号 0 で PCMCIA メモリカードをフォーマットするには、次のコマンドを入力します。

```
% fdformat pcmem0  
% newfs /vol/dev/aliases/pcmem0
```

MS-DOS ファイルシステムを使ってメモリカードをフォーマットするには、次のいずれかのコマンドを入力します。

```
% fdformat -t dos vol_alias_device_name
```

または

```
% fdformat -d vol_alias_device_name
```

たとえば、MS-DOS ファイルシステムを使って、PCMCIA ソケット番号 0 で PCMCIA メモリカードをフォーマットするには、次のいずれかのコマンドを入力します。

```
% fdformat -d pcmem0
```

または

```
% fdformat -t dos pcmem0
```

3. **PCMCIA** メモリカードをいったん取り出してから、挿入します。

この手順は、ボリューム管理にマウントディレクトリ /pcmem/pcmem0 を再マウントさせるために必要です。

また、PCMCIA メモリカードを取り出さずに、次の手順を実行することもできます。次のコマンドを入力します。

```
% eject pcmem0  
% volcheck
```

ボリューム管理は、PCMCIA メモリカードを自動的に再マウントします。

▼ PCMCIAメモリカードを再フォーマットするには

PCMCIA メモリカードが既にフォーマットされている場合、先に進んでください。

1. **PCMCIA** メモリカードを **PCMCIA** ソケットに挿入します。
2. コマンドを入力して、希望するファイルシステム (**UNIX** または **MS-DOS**) でメモリカードを再フォーマットします。

UNIX ファイルシステムを使ってメモリカードを再フォーマットするには、次のコマンドを入力します。

```
% fdformat -U vol_alias_device_name
% newfs /vol/dev/aliases/vol_alias_device_name
```

たとえば、UNIX ファイルシステムを使って PCMCIA ソケット番号 1 で PCMCIA メモリカードをフォーマットするには、次のコマンドを入力します。

```
% fdformat pcmem1
% newfs /vol/dev/aliases/pcmem1
```

また、MS-DOS ファイルシステムを使ってメモリカードを再フォーマットするには、次のいずれかのコマンドを入力します。

```
% fdformat -U -t dos vol_alias_device_name
```

または

```
% fdformat -Ud vol_alias_device_name
```

たとえば、MS-DOS ファイルシステムを使って PCMCIA ソケット番号 1 で PCMCIA メモリカードを再フォーマットするには、次のいずれかのコマンドを入力します。

```
% fdformat -d pcmem1
```

または

```
% fdformat -t dos pcmem1
```

3. PCMCIA メモリカードをいったん取り出してから、挿入します。

この手順は、ボリューム管理にマウントディレクトリ /pcmем/pcmем0 を再マウントさせるために必要です。

また、PCMCIA メモリカードを取り出さずに、次の手順を実行することもできます。次のコマンドを入力します。

```
% eject pcmem0
% volcheck
```

ボリューム管理は、PCMCIA メモリカードを自動的に再マウントします。
次の表 G-6 は、fdformat ユーティリティオプションの一覧を示します。

表 G-6 fdformat ユーティリティオプション

オプション	説明
-U	PCMCIA メモリカードを取り出す
-d	MS-DOS ファイルシステムをインストールする (デフォルトは UNIX)
-t dos	MS-DOS ファイルシステムをインストールする (デフォルトは UNIX)

ファイルのコピー

cp、rm、diff、ls などのコマンドを使って、PCMCIA メモリカードのファイルのディレクトリ名をコピー、削除、比較、一覧表示できます。

ハードディスクから **PCMCIA** メモリカードへファイルをコピーするには

- ◆ 次のコマンドを入力します。

```
example% cp filename /pcmem/pcm0/filename
```

PCMCIA メモリカードからハードディスクへファイルをコピーするには

- ◆ 次のコマンドを入力します。

```
example% cp /pcmem/pcmem0/filename /pathname/filename
```

eject (1) コマンドを使って **PCMCIA** メモリカードを取り出すには

ファイルシステムがマウントされている状態で PCMCIA メモリカードを取り出す場合は、eject (1) コマンドを使用しなければなりません。



警告 - マウント中の PCMCIA メモリカードを取り出すと、システムはパニック状態になります。

1. 次のいずれかのコマンドを入力します。

```
% eject vol_alias_device_name
```

または、

```
% eject pcmem0
```

削除可能なメディアマネージャのポップアップメニューが表示されます。

2. **OK** ボタンをクリックします。
3. **PCMCIA** メモリカードを取り出します。

注 - 引き続き PCMCIA メモリカードを使用する場合は、PCMCIA ソケットにそのままマウントしておきます。volcheck(1) コマンドを使って、ボリュームマネージャが PCMCIA メモリカードを再マウントできるようにします。このコマンドを実行するには、volcheck と入力します。

ボリューム管理の設定解除によるファイルのコピー

最後に、PCMCIA メモリカードのフォーマット、メモリカードとハードディスク間でファイルをコピーする3つ目の方法について説明します。ここでは、ボリューム管理が設定解除されているときの実行方法について説明します。

この節では、次の作業について説明します。

- ボリューム管理の設定解除
- PCMCIA メモリカードのフォーマット
- PCMCIA メモリカードのマウント
- ファイルのコピー
- ボリューム管理の設定

▼ ボリューム管理の設定を解除するには

1. ワークスペースメニューから「プログラム (**Programs**)」⇒「コマンドツール (**Command Tools**)」を選択します。
2. 次のコマンドを入力してスーパーユーザになります。

```
example% su
Password: root_password
example#
```

3. 次のコマンドを入力して、ボリューム管理の設定を解除します。

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

▼ PCMCIA メモリカードをフォーマットするには

PCMCIA メモリカードが既にフォーマットされている場合は、先に進んでください。

1. **PCMCIA** メモリカードを **PCMCIA** ソケットに挿入します。
2. コマンドを入力して、希望するファイルシステム (**UNIX** または **MS-DOS**) でメモリカードをフォーマットします。

注 - `format` ユーティリティは PCMCIA メモリカードには使用できません。`fdformat` ユーティリティだけ使用できます。

UNIX ファイルシステムでメモリカードをフォーマットするには、次のコマンドを入力します。

```
% fdformat device_name
% newfs device_name
```

たとえば、指定のデバイス名が `/dev/dsk/c1t6d0s2` のデバイスドライブで PCMCIA メモリカードをフォーマットするには、次のコマンドを入力します。

```
% fdformat /dev/dsk/c1t6d0s2
% newfs /dev/dsk/c1t6d0s2
```

MS-DOS ファイルシステムでメモリカードをフォーマットするには、次のいずれかのコマンドを入力します。

```
% fdformat -d device_name
```

または

```
% fdformat -t dos device_name
```

たとえば、指定のデバイス名が `/dev/dsk/c1t6d0s2` のディスクドライブで PCMCIA メモリカードをフォーマットするには、次のいずれかのコマンドを入力します。

```
% fdformat -d /dev/dsk/c1t6d0s2
```

または

```
% fdformat -t dos /dev/dsk/c1t6d0s2
```



警告 - マウント中に PCMCIA メモリカードを取り出すと、システムはパニック状態になります。

▼ PCMCIA メモリカードをマウントするには

- ◆ 次のコマンドを入力して、**UNIX** ファイルシステムで **PCMCIA** メモリカードをマウントします。

```
# mount mount_directory device_name
```

/mnt ディレクトリがない場合は、次のコマンドを入力します。

```
example# mkdir /mnt
```

たとえば、指定のアドレスが c1t6d0s2 のディスクドライブの /mnt ディレクトリに UNIX ファイルシステムをマウントするには、次のコマンドを入力します。

```
example# mount /dev/dsk/c1t6d0s2 /mnt
```

- ◆ 次のコマンドを入力して、**MS-DOS** ファイルシステムで **PCMCIA** メモリカードをマウントします。

```
# mount -F pcfs mount_directory device_name
```

/pcfs ディレクトリがない場合は、次のコマンドを入力します。

```
example# mkdir /pcfs
```

たとえば、指定のアドレスが `c1t6d0s2` のディスクドライブの `/pcfs` ディレクトリに MS-DOS ファイルシステムをマウントするには、次のコマンドを入力します。

```
example# mount -F pcfs /dev/dsk/c1t6d0s2 /pcfs
```

ファイルのコピー

`cp`、`rm`、`diff`、`ls` などのコマンドを使って、PCMCIA メモリカードのファイルのディレクトリ名をコピー、削除、比較、一覧表示できます。

ハードディスクから **PCMCIA** メモリカードへファイルをコピーするには

- ◆ 次のコマンドを入力します。

```
example# cp filename /pcfs/filename
```

PCMCIA メモリカードからハードディスクへコピーするには

- ◆ 次のコマンドを入力します。

```
example# cp /pcfs/filename /pathname/filename
```

マウント中に PCMCIA メモリカードを誤って取り出した場合は、マウントディレクトリもマウント解除してください。

- ◆ メモリカードを **PCMCIA** ソケットに挿入し、次のコマンドを入力してマウントディレクトリのマウントを解除します。

```
# umount mount_directory
```

たとえば、

```
example# umount /pcfs
```

ボリューム管理を設定するには

- ◆ 次のコマンドを入力して、ボリューム管理を設定します。

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

PCMCIA シリアル/モデムカードの使い方

デバイスの操作方法についての詳細は、PCMCIA モデムカードまたは PCMCIA シリアルインタフェースカードに付属の製品マニュアルを参照してください。

PCMCIA シリアル/モデムカードのデバイス名

PCMCIA シリアルデバイスのデバイス名は、`/dev/term` および `/dev/cua` ディレクトリ内に作成されます。

デバイス名は `pcN` です。

上記の `N` は PCMCIA ソケット番号です。

8つのエントリは `/etc/remote` ディレクトリに作成され、PCMCIA の最初の 8つのソケットに対応しています。

PCMCIA シリアル/モデムカードと電源管理システムの保存停止・復元再開機能

この節では、PCMCIA カードと電源管理システムソフトウェアをインストールしたシステムの追加情報を提供します。電源管理システムについて詳細は、『電源管理システムユーザーマニュアル』を参照してください。



注意 - システムの中断時または再起動時には、PCMCIA カードを挿入したり取り出さないでください。PCMCIA カードが、再起動操作後に識別されなくなります。

システムの中断時にアプリケーションが PCMCIA シリアル/モデムカードにアクセスしている場合、ハングアップ状態となりアプリケーションが終了することがあります。

たとえば、tip コマンドを使って PCMCIA シリアル/モデムカードにアクセスし、システムを中断しようとする場合、tip コマンドはシステムが再起動すると自動的に終了します。UUCP や PPP などのその他のアプリケーションは、PCMCIA シリアル/モデムカードに自動的にアクセスしようとします。

索引

A

Administration Tool 240, 235
 概要 235
 グループの管理 235
 使用方法 236, 237
 シリアルポート 240
 シリアルポートの管理 235
 ソフトウェアの管理 235
 プリンタ 239
 プリンタの管理 235
 ホスト 239
 ホストの管理 235
 ユーザアカウントの管理 235
apropos コマンド 42
-auth オプション 223

B

bdiff コマンド 56
Bootparams コマンド 203
Bourne シェル
 初期設定ファイル 161

C

cancel コマンド 198
cat コマンド 47
CDPATH 変数 163
cd コマンド 50, 51
chmod コマンド 61, 64
client_program& 227
cpio コマンド 243
CPU 時間 78

cp コマンド 45, 53
.cshrc ファイル
 OPENWINHOME 環境変数 192
 OPENWINHOME 環境変数 193
C シェル
 history コマンド 37
 初期設定ファイル 161, 162
 別名 (エイリアス) 166

D

date コマンド 36
DECnet インターネットワーキング
 (DNI) 229, 234
 OpenWindows のマシン上でリモートク
 ライアントを表示する 231
 VAX のマシン上でリモートクライアント
 を表示する 231, 234
 概要 229
 設定 229
.defaults ファイル 192
.defaults ファイル (SunView) を変換する
 (OpenWindows) 191, 192
/dev/cua ディレクトリ 261
/dev/term ディレクトリ 261
deviceoptions オプション 32
/devices ディレクトリ 199
-dev オプション 32, 33
df コマンド 80
diff3 コマンド 55
diff コマンド 54
ditroff プログラム 82

dump/restore コマンド 243
du コマンド 80

E

eject コマンド 256
email 113
env コマンド 163
/etc/fstab ファイル 198, 199
/etc/group ファイル 235
/etc/hosts ファイル 239
/etc/inet/hosts ファイル 235
/etc/lp ディレクトリ 199
/etc/passwd ファイル 235
/etc/printcap ディレクトリ 199
/etc/remote ディレクトリ 261
/etc/termcap ディレクトリ 199
/etc/ttytab ファイル 200
/etc/vfstab ディレクトリ 198
/etc/vfstab ディレクトリ 199
ex コマンド 85, 97, 99

F

fastboot コマンド 200
fasthalt コマンド 200
FAX モデム 241
fdformat ユーティリティ 246
file コマンド 47
finger コマンド 120
fkeys ファイル 211, 212

G

grep コマンド
構文 67, 68
正規表現 70
フィルタとして 68
メタキヤラクタ 71, 72
文字列 69

H

history コマンド 37
指定 163
HISTORY 変数 163
HOME 変数 164
hosts ファイル 235

I

IA
Solaris 2.5 への移行 201, 203
IA システム
アップグレードオプション 197
ID 番号による要求 148
init コマンド 200

K

kdmconfig プログラム 203
/kernel ディレクトリ 199
kill コマンド 79

L

LANG 変数 164
lefty.data ファイル 207, 209
.login ファイル 163
logname コマンド 37
LOGNAME 変数 164
LPDEST 変数 164
lprm コマンド 198
lpr コマンド 198
lpstat コマンド 143, 147
構文 147
lp コマンド 139, 198
オプション 8, 142
LP プリントサービス 141
ls コマンド 45
隠しファイルの表示 60
詳細形式のリスト 59, 60

M

mailbox
パス名の指定 164
mailtool プログラム 198
mailx プログラム 114, 198
終了 116
MAIL 変数 164
MANSECTS 変数 164
man コマンド 20, 41
man ページ 41
MIT-MAGIC-COOKIE 221
mkdir コマンド 52
メール 126

more コマンド 47
mv コマンド 46

N

-noauth オプション 31, 220, 222
nohup コマンド 41
nolefty.data ファイル 210
nroff プログラム 82

O

olwm 193, 195
.openwin-menu ファイル 196
OpenWindows
 MIT-MAGIC-COOKIE 221
 SunView 環境からの移行 191, 192
 キーボード設定の変更 205, 215
 起動ファイル 194, 195
 初期設定ファイル 163
 バージョン 3.3 より前の OpenWindows
 環境から移行する 192, 196
 メール 113
OPENWINHOME 環境変数 192, 193
openwin コマンド
 -noauth オプション 31
/opt ディレクトリ 199
OWconfig データエラーメッセージ 189
.OWdefaults ファイル 195

P

passwd コマンド 76
PATH 変数 164, 166
PCMCIA SRAM メモリカード 246
PCMCIA カード
 使い方 241
 ファイルのコピー 243
PCMCIA シリアル/モデムカード 261
PCMCIA メモリカード 241
 からハードディスクへファイルをコ
 ピー 255
PID 78, 79
/platform/*/kernel/unix ファイル 198
.profile ファイル
 OPENWINHOME 環境変数 192, 193
PS1 コマンド 167
PS1 変数 164

ps コマンド 78
put コマンド 95
pwd コマンド 44, 49

R

rcp コマンド 156, 157
rlogin コマンド 152, 155
 中止 154
 中断 155
rmdir コマンド 53
rm コマンド 46, 54
root ユーザ 132
RS-232 インタフェース 241
rsh コマンド 157, 158
rusers コマンド 121, 158

S

SHELL 変数 164
Solaris 2.5 への移行 197, 203
sort コマンド 80
SPARC マシン
 OpenWindows 環境と DECwindows 環境
 のインターネットワーキン
 グ 229, 234
 SunOS 5.x 環境への移行 198
 SunView 環境からの移行 191, 192
 アップグレードオプション 197
 キーボード設定の変更 206, 209, 210
SunLink 231
SunOS/BSD ソース互換パッケージ 200
SunOS 4.x
 からの移行 197, 201
 からの変更点 198, 200
SunOS 5.x
 システムソフトウェア 198
 システムファイルの位置 198
SunOS バイナリ互換パッケージ 200
SunView 環境からの移行 192, 193
SunView との適合性 191
 .Xdefaults ファイル 192
 convert_to_xdefaults プログラム 192
swmtool 199

T

tar コマンド 245, 250, 255

PCMCIA メモリカードをフォーマットする 245
/terminfo データベース 199
TERMINFO 変数 164
TERM 変数 165
touch コマンド 45
troff プログラム 82, 198
TZ 変数 165

U

umask コマンド 169, 170
/usr/openwin/etc/devhandlers ディレクトリ 189
/usr/openwin/lib/Xinitrc ファイル 193
/usr/openwin ディレクトリ 193
/usr ディレクトリ 199

V

/var/mail ディレクトリ 199
/var/mail ファイル 198
/var/spool/mail ディレクトリ 199
/var/spool/mail ファイル 198

VAX 上での

OpenWindows 環境と DECwindows 環境
のインターネットワーキング
229, 234

X11 クライアントアプリケーションの実
行 229

vi

1 文字以上の置換 92
1 文字の削除 94
1 文字の置換 93
append コマンド 91
ex コマンド 85, 97, 99
ex のコピーコマンド 98
put コマンド 95
yank コマンド 95
カーソル 82, 87
カウントの使い方 97
行挿入コマンド 91
行の移動 96, 99
行の削除 94, 99
行の分割 93
行の変更 92
行の連結 93
行番号 98

クラッシュからの回復 106
検索時における大文字、小文字の区
別 100

検索と置換 100
コマンドの繰り返し 97
コマンドモード 84
指定行へのジャンプ 103
状態表示行 83
スクローリング 89
操作 87, 90
挿入コマンド 91
定義 81
テキストのコピー 95
テキストの削除 94, 95
テキストの挿入 90, 91
テキストの変更 91, 93
名前付きバッファ 97
入力モード 83, 84
パラメータの設定 106
ファイルの印刷 87
ファイルの作成 82
複数ファイルの操作 104, 106
ページング 89
別ファイルの読み込み 103
変更の取り消し 93
メールの使い方 129
文字の入れ替え 93
ワードの変更 92

view コマンド 82
vi エディタ 81
vi ファイル 87
/vmunix ファイル 198
volcheck コマンド 256

W

whatis コマンド 41
who コマンド 120

X

X11 サーバ
OpenWindows 環境と DECwindows 環境
のインターネットワーキング
229, 234
.Xauthority ファイル 223, 226
xauth プログラム 225

.Xdefaults ファイル 192
SunView アプリケーション 191
変換 192
.xinitrc ファイル 193
起動ファイルの手順 194, 195
バージョン 3.3 より前の環境から移行する 196
バージョン 3.3 より前の環境から移行する 192
xmodmap コマンド 205, 213
X ビジュアル 176

Y

yank コマンド 95

あ

アカウント 19
アクセス権
種類 58
絶対アクセス権の設定 64, 66
設定対象の分類 59
変更 61, 63
ワイルドカード文字*の 63
アクセス制御機構 220, 221
アップグレードオプション
SPARC 197
IA 197
アプリケーション
DECnet インターネットワーキング 229
リモートマシン上で 218

い

移行
Solaris 2.5 への 197, 203
SunOS 4.x からの 197, 201
SunView 環境からの移行 191, 192
バージョン 3.3 より前の OpenWindows 環境から移行する 192, 196
一般的な作業セッション 19
移動
ex コマンドによる vi テキスト 99
vi テキスト 99
ディレクトリ 52
ファイル 46

印刷 139
完了通知の要求 141
状態の確認 143, 147
デフォルトプリンタ 140
特定のプリンタ 140
複数部数 142
要求の取り消し 148, 149
利用可能なプリンタの確認 144
印刷サブシステムの変更 198
インストール、Solaris 2.5 198
インターネットワーキング、OpenWindows 環境と DECwindows 環境 229, 234
DECnet インターネットワーキングの設定 229
OpenWindows のマシン上でリモートクライアントを表示する 231
VAX のマシン上でリモートクライアントを表示する 231, 234
概要 229
インターネットワーク 152
引用符 73

え

エコー 20, 76
エラーメッセージ
このデバイス用のプロファイルを更新できません 188
スワップメモリ不足のため継続できません 188
調整ツールが正しく機能していません 189
調整ツールからの応答が得られません 189
調整データ構造を初期化できません 188
調整パックの感度を獲得できません 189
調整パックの感度を設定できません 189
調整パックの再表示フィールドの値を獲得できません 189
調整パックの積分時間が得られません 189
調整パックのバージョン番号を読み取れません 189
調整パックの平均値を設定できません 189

ビジュアルデータ構造を初期化できません 188
ビジュアルプロファイルを作成できません 188
プロファイルファイルを /etc/openwin に構成できません 188
プロファイルをロードできません 188
モジュールが輝度データの測定に失敗しました 189

お

大文字、小文字の区別

vi エディタ 36

コマンド 36

オプション 39

親ディレクトリ 51

オンラインヘルプ機能 41

か

カーネルによりロードされたドライバ 199

カーボンコピー 122

書き込み権 59

書き込み保護モード 243

隠しファイル

.xinitrc 193

各種の終了方法 85

画面

複数の画面を使ってウィンドウシステム
を実行する 193

カラー

正確さを保つ 175

モニター上でのシフト 176

カラーモニター 177

オンライン参考資料 177

カラーのシフト 176

環境変数

DISPLAY 218, 226

LD_LIBRARY_PATH 218

OPENWINHOME 192, 193

ネットワークアプリケーション 218

き

キーボードの再マッピングの取り消し

(SPARC) 209

キーボードの設定変更 205, 215

Compose キーの設定と設定解除 205, 206

コントロールキーの再マッピング

(IA) 211, 215

再マッピングの取り消し 209, 212, 215

左きき用キーの再マッピング

(SPARC) 206, 210

ファンクションキーの再マッピング

(IA) 211, 215

キーボードマッピングの再設定 211, 215
起動

OPENWINHOME 環境変数 192, 193

SunView との適合性 191

.xinitrc ファイルの注意事項 193, 194

手順 199

複数の画面 34

起動オプション 30

キャラクターゼーション 175

キャンパスエリアネットワーク (CAN) 152

行挿入コマンド 91

く

クライアント認証ファイル 223, 225

け

検索

find コマンド 56, 58

grep コマンド 67, 68

こ

広域ネットワーク 152

コピー

vi テキスト 97

vi ファイル間での行 105

ディレクトリ 53

ファイル 45

フォルダに対するメール 126

リモートマシンから 156

リモートマシンへ 157

コマンド

vi コマンドの繰り返し 97

大文字、小文字の区別 36

オプション 39

キーワード検索 42

繰り返し 37, 38

構文 41

- 出力のパイプ 39
- 出力のリダイレクト 39, 40
- 前回の繰り返し 37
- チルドコマンド 135
- 長いコマンドの入力 37
- バックグラウンドでの実行 40
- 複数のコマンドの入力 37
- リファレンス情報の取得 41
- リモートマシン上での実行 157, 158
- 履歴 37, 38
- コマンドオプション 39
- コマンドツール
 - リモートマシン 218, 219
 - リモートマシン上で 218
- コマンドプロンプト 20, 33, 35
- コマンドモード 83, 84

さ

- サービスアクセス制御 (SAC) 200
- 削除
 - vi テキスト 94
 - .xinitrc ファイル 193
 - グループ 235
 - シリアルポートサービス 240, 235
 - ソフトウェア 235
 - ディレクトリ 53
 - ファイル 46
 - プリンタ 239, 235
 - ホスト 235
 - メール 118, 119
 - ユーザアカウント 235

し

- システムからのログアウト 21
- システム管理者 19
- システム管理ツール 235
- システムクロック 165
- システムファイルの位置、SunOS 5.x での 198
- システムプロファイル 162
- システムへのログイン 20, 35
 - OpenWindows 30
 - 現在のログイン名以外のユーザ名による 153
 - リモートマシンに 152, 155
- 実行可能ファイル 43

- 実行権 59
- 自動セキュリティ拡張ツール (ASET) 203
- 状態表示行 83
- 初期設定ファイル 161
 - 環境変数 163, 166
- シリアルポートサービスの管理 240, 235
- シンボル .. 51

せ

- 正規表現 70
- セキュリティ
 - MIT-MAGIC-COOKIE-1 221
 - noauth オプション 220
 - SUN-DES-1 認証プロトコル 221
 - .Xauthority ファイル 223, 226
 - xauth プログラム 223
 - アクセス制御機構 220
 - クライアントをリモートで実行する場合とローカルで実行する場合 226
 - サーバへのアクセスを制御する 223, 226
 - 注意 222
 - 認証プロトコル 221, 227
 - 利用方法 219
- 絶対アクセス権 64, 66
- 設定解除
 - シリアルポートサービス 240, 235
 - ボリューム管理 257

そ

- 相対パス名 52
- 挿入コマンド 91

た

- 端末装置 200

ち

- 中断、調整 187
- 調整 177
 - 概要 175, 177
 - 中断 187
 - モニター調整の概要 175, 177
- 調整ツール 176
 - エラーメッセージ 188

の開始 182
の実行 181
の終了 187
割り当てエラー 190
調整バック 174, 175
エラー 188
接続 181
チルドエスケープ 137
チルドコマンド 135

つ

追加

- グループ 235
- シリアルポートサービス 240, 235
- ソフトウェア 235
- プリンタ 235
- ホスト 235
- ユーザアカウント 235

て

ディスクの使用状況 80

ディレクトリ

- dead.letter 122
- アクセス権の変更 61, 63
- 移動 52
- 階層 48
- 現在の位置の表示 49
- コピー 53
- 削除 53
- 作成 52
- 使用状況を調べる 80
- セキュリティ 58, 66
- 定義 43, 44
- 名前の変更 52
- パス名 48
- フォルダディレクトリの設定 126
- 変更 50, 51
- ホーム 50
- ルート 48

テキストのコメント化 134

テキストファイル 43

デバイス 199

デバイスハンドラのエラーメッセージ 189

デバイス命名規則 199

デフォルト

- コマンドプロンプト 35

シェル 161
ディレクトリ 44
プリンタ 140, 164
電子メール 113

と

ドットファイル 60
.xinitrc ファイル 193
ドライバ、カーネルによりロードされた 199

な

名前付きバッファ 97
名前の変更

- ディレクトリ 52
- ファイル 46

に

入力モード 83
認証プロトコル 221, 222

- MIT-MAGIC-COOKIE-1 221
- SUN-DES-1 認証プロトコル 221

ね

ネットワーク

- セキュリティの基本事項 223, 226
- 定義 151
- プロトコル 152

ネットワークアプリケーション

- DECnet 229

は

バージョン 3.3 より前の OpenWindows 195

パイプ

- grep コマンドを介しての ps 出力 79
- grep コマンドを介しての出力 68
- lp コマンドへのメール 119
- sort コマンドを介しての du 出力 80
- コマンド出力 39

パイプ記号 40

パス名 48

- 相対 52

パスワード 19

- 選択 75

変更 76
有効期限の設定 77
バックグラウンドでのコマンドの実行 40
パニック状態 256
パラメータ
vi の起動 106
設定状況 106

ひ

ピクセルごとの表示特性 176
ピクチャ(コントラスト)つまみ 179
ビジュアル 176, 177
左きき用キーの再マッピング (SPARC) 206,
209
左きき用キーボード 206
表示
ディスクの使用状況 80
ディレクトリの使用状況 80
ファイルサーバ上のユーザ 120
ファイルのアクセス権 60
ファイルの内容 47
プリンタの状態 144, 146
メールボックスのリスト 117
リモートマシン上のディレクトリの内容 158
リモートマシンのログイン状態 156
リモートユーザ 158, 159
表示環境の調整 177, 181
表示環境の変更 177, 181

ふ

ファイル
.cshrc 162, 170
diff コマンド 54, 55
/etc/aliases 132
/etc/hosts.equiv 156, 158
/etc/passwd 156, 158, 161
.login 162
.mailrc 131
.profile 162, 170
アクセス権 60
アクセス権の変更 61, 63
印刷 140
隠しファイルの表示 60
検索 56, 58
コピー 45

作成 45
初期設定 161, 162
セキュリティ 58, 66
定義 43, 44
内容の表示 47
ファイル形式を調べる 47
メールの挿入 123
メールの保存 125
ファイルシステムの変更 199
フォルダ 126, 129
フォント
DECwindows サーバにインストールする 231, 234
デフォルトの変更 171
ネットワーク対応アプリケーション 233
リスト 174
複数の画面 31
複数のモニタを使用した OpenWindows の起動 31
不正なプロファイルエラーメッセージ 190
ブラインドコピー 122
プログラムメニュー (ワークスペースプロパティ)
カスタマイズ 196
プロセス
実行 78
終了 79
プロセス識別番号 79

へ

別名(エイリアス) 166
変更
.xinitrc ファイル 193
サーバへのアクセスを制御する 223, 227
シリアルポートサービス 240, 235
認証プロトコル 222
プリンタ設定 239, 235
ユーザアカウント 235

ほ

ホームディレクトリ 44, 50
rlogin コマンド 153
指定 164
短縮名(チルド文字) 50
定義 164

ホスト管理 239, 235
ボリューム管理の設定解除 257
 PCMCIA メモリカードをフォーマットする 257
ポンド記号
 コメントに変更 192

ま

マニュアルページ 20, 41, 164

め

名目プロファイル
 概要 175
 の調整 175
メール 120
 copy コマンド 126
 mbox ファイル 114
 vi の使い方 129
 印刷 119
 エイリアス 130, 132, 135
 応答 124
 カーボンコピー 122
 起動 114
 削除 118, 119
 終了 116
 送信 120 - 122
 送信しないメールの取り消し 122
 送信できないメール 122
 挿入 123
 チルドコマンド 135
 バージョンを調べる 116
 ファイルの挿入 124
 フォルダ 126, 129
 ブラインドコピー 122
 ヘルプコマンド 137
 保存 125, 128
 メールボックス 114
 読む 115, 117, 118
 リストを表示 117
メールエイリアス 130
メールフォルダ 126
メールボックス 114
メタキャラクタ 72

も

モジュール測定の失敗エラーメッセージ 189
モジュールの初期化エラーメッセージ 189
文字列 100
モデムの管理 200, 240, 235
モニター
 画面距離の確保 178
 カラーの調整 174, 190
 輝度が高すぎる 181
 左右に配置する 32
 室内の明るさの調整 178
 上下に配置する 32
 調整 179
 調整ツール用に選択 182, 183
 反射の抑制 177, 178
 反射防止処理 178
 プロファイル情報 183, 184

ゆ

ユーザアカウントの管理 235
ユーザアドレスの決定 121
ユーザのネットワーク名 (netname) 224
ユーザプロファイル 162
ユーザ名 19

よ

読み取り権 59

ら

ラインプリンタ用サブシステム 139

り

リモートマシン上でアプリケーションを実行 218
リモートログイン 152
リレー 152

る

ルートディレクトリ 48

れ

連結 47

ろ

ローカルエリアネットワーク 152

ログインシェル 20, 21

ログイン名 19

指定 164

ロケールの言語 164

わ

ワイルドカード文字 45