

UNIX ガイド

第二版

— cc 環境の利用者へ —

京都産業大学
計算機センター事務室

'95.3.23

結 言

UNIX ガイドの初版は 1994 年 3 月である。たちまち、9 月には UNIX ガイド 追補版が発行された。そして、1995 年 4 月には改訂版が出版されることになった。時のたつのも速ければ、コンピュータの新製品が出るのも速い。本学のコンピュータ設備更新も速いし、情報教育の内容もどんどん変わる。当然、ガイドブックの内容も変更せざるを得ない。

それにしても、大学でこのようなガイドブックが発行されることを考えると、UNIX の普及の理由がわかる。それは、メーカーサイドの押し付け販売ではなく、ユーザサイドのボランティアなサポートで飛躍的に UNIX ユーザが増大してきたことが実感できる。

UNIX の出現は大型機全盛の 1970 年代である。ベル研究所でミニコン用に使われていた。1970 年後半には、日本に上陸し、東京大学の石田晴久研究室に導入された。1980 年には東京大学計算機センタでサービスが開始されている。その後の 10 年足らずは、UNIX はコンピュータマニアか、サポータがマニアックに使いながら、普及していった。

本学には 1985 年にスーパーメイト (旧立石電機) が計算機科学研究所に設置された。その UNIX はなんとシステム V バージョン 1.0 である。当然、英文のコンピュータ内のマニュアルか翻訳ホヤホヤの誤植の多い説明書を見ながら使っていた。1987 年には SUN3 や SONY の EWS が設置され、ネットワークを構成し利用者が広がっていった。1989 年の、工学部開設と共に飛躍的に台数がのびる。

この頃には、分散処理指向が強まり、ワークステーションとパソコンをネットワークに接続することが当たり前となった。そして、1994 年に UNIX サーバや UNIX ワークステーション教室の増設となったのである。

コンピュータの利用目的のうち、メール、検索、ワープロ、DTP は不可欠であろう。UNIX にはこれらの機能を最大限にサポートするパブリックドメインのグループウェアがある。このようなソフトを我流で使うのではなく、スマートに使いこなすために、UNIX ガイドは今後も大勢のサポータで版を重ねてゆかなければならない。

工学部 黒住 祥祐

UNIX ガイドの読み方

UNIX ガイドは初心者が京都産業大学の共用 UNIX コンピュータを利用して生活する為の手引書として書かれています。第 1 章「はじめに」では全ての利用者に共通の注意を説明しています。第 2 章から第 5 章までで UNIX の基本的な操作方法から始まり、電子メールやニュースなどが使えるようになるまでを説明しています。ここまでは全ての UNIX 利用者に習得して欲しいと思います。コンピュータの事をほとんど何も知らなくても、何とかここまではたどりつけるように書いたつもりです。頑張ってください。

後の章はそれぞれ以下の事について説明しています。必要に応じて読んで下さい。

第 6 章	UNIX もっともっと	UNIX のより進んだ使い方について
第 7 章	Mule	テキストエディタ Mule について
第 8 章	L ^A T _E X	文書処理システム L ^A T _E X について
第 9 章	AU _C T _E X	L ^A T _E X を Mule と組み合わせてより便利にする AU _C T _E X について
第 10 章	レポートシステム	レポートを電子メールで提出するシステムについて
第 11 章	Mathematica	数式処理システム Mathematica について
第 12 章	GNUPLOT	グラフ作成システム GNUPLOT について
第 13 章	tgif	画像作成システム tgif について
第 14 章	xv	画像処理システム xv について
第 15 章	xpaint	画像作成システム xpaint について
第 16 章	NQS	バッチ処理を実現するシステム NQS について
第 17 章	MODEM から telnet	例えば自宅から UNIX コンピュータを利用する手続きについて
第 18 章	NeXT	NeXT コンピュータの使い方について

また、巻末の付録にはリファレンスとして各種コマンドや記号などの一覧表、情報処理教室を利用する際の注意、著作権法の抜粋などがまとめられています。参考文献の一覧も付けておきました。

目次

第 1 章	はじめに	1
1.1	コンピュータ利用上の注意	1
1.1.1	規則	1
1.1.2	慣習、道徳	1
1.1.3	法律	1
1.2	サポート	2
1.2.1	相談窓口	2
1.2.2	各種ドキュメントはどうやったら得られますか？	2
第 2 章	UNIX はいかが	3
2.1	UNIX 環境	3
2.1.1	計算機センター運用の UNIX 環境	3
2.1.2	ユーザ名とパスワード	4
2.1.3	パスワードの安全対策	4
2.1.4	Sign しよう！	5
2.1.5	ユーザ登録申請	8
2.2	これ以降の読み方	8
2.3	login と logout	10
2.4	キーボード	10
2.5	ウィンドウ環境	11
2.5.1	マウス	11
2.5.2	ウィンドウ環境の画面	12
2.5.3	メニュー	14
2.5.4	ボタン	14
2.5.5	ウィンドウ環境のトラブル傾向と対策	15
2.6	どこから login?	16
2.6.1	パソコンから UNIX を利用する為の予備知識	16
2.7	コンピュータウイルス	18
第 3 章	UNIX への道	19
3.1	これ以降の読み方	19
3.2	DEC-3300 を使う	21
3.2.1	まず部屋へ	21
3.2.2	電源を入れる	21
3.2.3	login する	22
3.2.4	ちょっとメッセージ	23
3.2.5	ターミナルはあるかな？	23

3.2.6	パスワードを変える	25
3.2.7	logout する (セッションを終了する)	26
3.2.8	シャットダウンする	27
3.2.9	電源を切る	27
3.2.10	部屋を出る	28
3.2.11	さて、さて、	28
3.2.12	マニュアルなど	28
3.3	Macintosh の telnet で cc2000 を使う	29
3.3.1	まず教室にはいる	29
3.3.2	電源を入れる	30
3.3.3	うまく電源が入らないとき	31
3.3.4	Macintosh の操作	31
3.3.5	端末ソフトの起動	31
3.3.6	さあ、login しよう	32
3.3.7	ちょっとメッセージ	32
3.3.8	パスワードを変える	33
3.3.9	logout をしよう	33
3.3.10	端末ソフトを終了する	34
3.3.11	電源を切る	34
3.3.12	部屋を出る	34
3.3.13	困った時の対処	35
3.3.14	cc2000 以外のホストコンピュータを使う時の注意	35
3.3.15	マニュアルなど	36
3.4	FMR から telnet で cc2000 を使う	37
3.4.1	まず部屋へ	37
3.4.2	電源を入れる	37
3.4.3	Windows の起動	38
3.4.4	Windows の動かし方	39
3.4.5	Windows に関する基礎知識 (絶対にやってはいけない事)	39
3.4.6	端末ソフトの起動	40
3.4.7	ホストコンピュータへの接続	41
3.4.8	さあ、login しよう	42
3.4.9	ちょっとメッセージ	43
3.4.10	パスワードを変える	43
3.4.11	logout をしよう	44
3.4.12	端末ソフトを終了する	44
3.4.13	Windows を終了する	44
3.4.14	電源を切る	45
3.4.15	部屋を出る	45
3.4.16	さて、さて、	45
3.4.17	困った時の対処	45
3.4.18	マニュアルなど	47
3.5	Contura から telnet で cc2000 を使う	48
3.5.1	まず部屋へ	48

3.5.2	机の引出しからパソコンを取出す	48
3.5.3	電源を入れる	49
3.5.4	Windows の動かし方	49
3.5.5	Windows に関する基礎知識 (絶対にやってはいけない事)	50
3.5.6	端末ソフトの起動	51
3.5.7	ホストコンピュータへの接続	51
3.5.8	さあ、login しよう	52
3.5.9	ちょっとメッセージ	52
3.5.10	パスワードを変える	53
3.5.11	logout をしよう	54
3.5.12	端末ソフトを終了する	54
3.5.13	電源を切る	54
3.5.14	パソコンを机の引出しにしまう	54
3.5.15	部屋を出る	54
3.5.16	さて、さて、	54
3.5.17	困った時の対処	55
3.5.18	マニュアルなど	56
3.6	FMV から X で cc2000 を使う	57
3.6.1	まず部屋へ	57
3.6.2	電源を入れる	57
3.6.3	Windows の動かし方	58
3.6.4	Windows に関する基礎知識 (絶対にやってはいけない事)	58
3.6.5	端末ソフトの起動	59
3.6.6	login する	61
3.6.7	ちょっとメッセージ	61
3.6.8	ターミナルはあるかな?	62
3.6.9	パスワードを変える	63
3.6.10	logout (セッションの終了)	64
3.6.11	電源を切る	64
3.6.12	部屋を出る	64
3.6.13	さて、さて、	64
3.6.14	困った時の対処	65
3.7	FMV から telnet で cc2000 を使う	66
3.7.1	まず部屋へ	66
3.7.2	電源を入れる	67
3.7.3	Windows の動かし方	67
3.7.4	Windows に関する基礎知識 (絶対にやってはいけない事)	68
3.7.5	端末ソフトの起動	69
3.7.6	ホストコンピュータへの接続	69
3.7.7	さあ、login しよう	70
3.7.8	ちょっとメッセージ	71
3.7.9	パスワードを変える	71
3.7.10	logout をしよう	72
3.7.11	端末ソフトを終了する	72

3.7.12	電源を切る	73
3.7.13	部屋を出る	73
3.7.14	さて、さて、	73
3.7.15	困った時の対処	73
3.7.16	マニュアルなど	75
3.8	DECpc から telnet で cc2000 を使う	75
3.8.1	まず部屋へ	75
3.8.2	電源を入れる	76
3.8.3	端末ソフトの起動	77
3.8.4	ホストコンピュータへの接続	78
3.8.5	さあ、login しよう	79
3.8.6	ちょっとメッセージ	79
3.8.7	パスワードを変える	80
3.8.8	logout をしよう	81
3.8.9	端末ソフトを終了する	81
3.8.10	電源を切る	82
3.8.11	部屋を出る	82
3.8.12	さて、さて、	82
3.8.13	困った時の対処	82
3.8.14	マニュアルなど	83
第 4 章	UNIX それから	84
4.1	基礎知識をもう一度	84
4.1.1	login	84
4.1.2	キー表記	84
4.1.3	カーソル	85
4.2	コマンド	86
4.2.1	コマンドって何だ？	86
4.2.2	プロンプト	86
4.2.3	簡単なコマンド	86
4.2.4	引数とオプションのあるコマンド	87
4.2.5	対話的なコマンドとそのサブコマンド	88
4.2.6	コマンドの使い方を調べる	89
4.2.7	UNIX によるコマンドの違い	91
4.2.8	トラブルからの脱出	91
4.3	シェル	93
4.3.1	コマンド入力時の編集	93
4.3.2	ヒストリ	94
4.3.3	イベント	95
4.4	ファイル	96
4.4.1	でもやっぱりファイルって何？	96
4.4.2	ファイルの一覧を見る	96
4.4.3	試しにファイルを作ってみましょう	97
4.4.4	ファイルの内容を見る	97

4.4.5	ファイル名を変える	98
4.4.6	ファイルの複写	99
4.4.7	ファイルの消去	99
4.5	ファイルを編集する	100
4.5.1	Mule での作業の流れ	100
4.5.2	Mule の起動	100
4.5.3	A. の場合：X ウィンドウ環境での Mule の起動とエラー対策	101
4.5.4	B. の場合：非 X ウィンドウ環境での Mule の起動とエラー対策	102
4.5.5	ファイル名の指定	103
4.5.6	編集	103
4.5.7	ファイルへの保存	105
4.5.8	Mule の終了	105
4.5.9	Mule もっともっと	105
4.6	印刷	107
4.6.1	どんなプリンタがあるか	107
4.6.2	ファイルの印刷	107
4.6.3	印刷状況をチェックする	107
4.6.4	印刷の取消し	108
4.6.5	利用上の注意	109
4.7	状況の変化	110
4.8	ファイルの階層構造	111
4.8.1	ディレクトリ	111
4.8.2	ツリー構造におけるファイル名の表記	114
4.8.3	ディレクトリの扱い	115
4.8.4	ディレクトリを意識したコマンドの書き方	116
4.9	EGG：Mule での漢字の入力	120
4.9.1	かな漢字変換	120
4.9.2	Wnn と EGG	120
4.9.3	まとめ	123
4.9.4	ローマ字入力のヒント	123
第 5 章	ネットワークの世界へようこそ	124
5.1	ネットワークサービス紹介	124
5.1.1	電子メールって何？	124
5.1.2	ニュースって何？	124
5.1.3	京都産業大学のネットワーク	125
5.1.4	Internet とは？	125
5.1.5	Internet mail サービスってどんなもの？	126
5.1.6	Internet news サービスってどんなもの？	127
5.1.7	ネットワークでの暮らし方	127
5.2	電子メール準備体操	128
5.2.1	Internet mail アドレスについて	128
5.2.2	計算機センター運用のコンピュータのメールアドレス	128
5.2.3	相手のメールアドレス	130

5.2.4	自分のメールアドレス	130
5.2.5	さあ、本番！	130
5.3	MHE : Mule による電子メールの読み書き	131
5.3.1	はじめに	131
5.3.2	メールを読む	132
5.3.3	メールを書く	135
5.3.4	来たメールの返事を書く	136
5.3.5	メールの整理	138
5.3.6	メールが来ているかどうか確認する	138
5.3.7	メールの実体はどこに？	138
5.3.8	トラブルからの脱出	138
5.3.9	MHE もっともっと	139
5.4	メールを書くときの注意	139
5.5	GNUS : Mule によるニュースの読み書き	141
5.5.1	はじめに	141
5.5.2	GNUS の起動	141
5.5.3	記事を読む	142
5.5.4	GNUS を終了する、再起動する	144
5.5.5	ニュースグループを選ぶ	144
5.5.6	記事を投稿する (けどちょっと待てよ)	145
5.5.7	新しい記事を投稿する	146
5.5.8	記事にフォローする	147
5.5.9	記事のキャンセル	148
5.5.10	メールで返事をする	149
5.5.11	古い記事を読み返す	149
5.5.12	記事の保存	150
5.5.13	ちょっとしたテクニック	151
5.6	記事を書くときの注意	151
第 6 章	UNIX もっともっと	154
6.1	ファイルの整理	154
6.1.1	ファイルの大きさを調べる	154
6.1.2	身に覚えのないファイル	155
6.1.3	ファイルを削除する	157
6.1.4	ディスクの利用量をチェックする	157
6.2	情報検索	158
6.2.1	学内の利用者情報の検索	158
6.2.2	学外の組織などの情報を調べる	159
6.3	ファイルのアクセス権	162
6.3.1	アクセス権	162
6.3.2	UNIX におけるアクセス権	162
6.3.3	アクセス権限を調べる	163
6.3.4	アクセス権限を設定する	164
6.4	シェルよもう一度	166

6.4.1	ファイル名の補完	166
6.4.2	対話的なファイル名の補完	167
6.4.3	シェル変数と環境変数	168
6.4.4	リダイレクション	169
6.4.5	パイプ	171
6.4.6	シェルの鬼へのヒント	172
6.4.7	シェルよ永遠に	174
第 7 章	Mule	175
7.1	なにそれ?	175
7.2	Mule の仕組み	175
7.3	これだけ覚えたら Mule は使える!	176
7.4	もうちょっと覚えよう	177
7.5	日本語入力	179
7.5.1	ためにしに変換	180
7.5.2	特殊な文字列の入力	182
7.5.3	まとめ	183
7.5.4	ローマ字入力のヒント	183
7.6	ちょっと中級者	184
7.6.1	単語登録	184
7.6.2	検索	185
7.6.3	画面分割	185
7.6.4	バッファリスト	186
7.6.5	モードライン	186
7.6.6	Dired モード	187
7.7	上級者に向けて	187
7.8	X 環境での Mule	187
7.9	外国語の入力	188
7.10	とりあえずここまで	189
第 8 章	L^AT_EX	190
8.1	L ^A T _E X(らてふ)って何?	190
8.1.1	L ^A T _E X とは?	190
8.1.2	L ^A T _E X の特徴	190
8.1.3	L ^A T _E X の作業の進めかた	190
8.2	それでは使ってみましょう	191
8.3	L ^A T _E X におけるルール	196
8.3.1	最低限のルール	196
8.3.2	ドキュメントスタイルについて	196
8.3.3	見出しの種類	197
8.4	いろいろなコマンドと環境	200
8.4.1	特殊文字	200
8.4.2	特殊文字でもそのまま出力する	202
8.4.3	文字の空白	202

8.4.4	改行と改ページ	203
8.4.5	水平方向と垂直方向の空白	203
8.4.6	引用	205
8.4.7	箇条書き	206
8.4.8	右寄せ、中央寄せ、左寄せ	209
8.4.9	文字の大きさ	209
8.4.10	書体	210
8.5	表題	211
8.5.1	タイトルの作り方	211
8.5.2	概要の作り方	211
8.6	傍注	212
8.7	脚注	212
8.8	相互参照	212
8.9	箱	214
8.9.1	一行に収まる文字列を囲む	214
8.9.2	複数行にわたる文の箱を作る	216
8.9.3	minipage 環境	217
8.10	表	219
8.11	絵	221
8.12	数式	224
8.12.1	数式の環境	224
8.12.2	添字	226
8.12.3	平方根	226
8.12.4	分数	227
8.12.5	括弧	228
8.13	L ^A T _E X で扱える記号	230
8.13.1	雑記号	230
8.13.2	空白を空ける文字	230
8.13.3	アクセントなど	230
8.13.4	ヨーロッパ系言語特有の記号	231
8.14	数式環境で使える記号	232
8.14.1	雑記号	232
8.14.2	ギリシャ文字	234
8.14.3	関数	235
8.15	数学のテクニック	236
8.15.1	行列を作る。	236
8.15.2	= の位置を揃える方法。	236
8.15.3	数式モードでの書体	236
8.15.4	新しくコマンドを作る	237
8.15.5	コマンドの変更	237
8.16	複雑な数式を少々	238
8.17	さらに例題	239
8.18	エラーの対処	241
8.18.1	エラーの表示	241

8.18.2	ちょっと違うやり方	243
8.18.3	エラーの種類	244
8.19	部分印刷する方法	245
8.20	自分の命令 (マクロ)	247
8.20.1	簡単な命令を作ってみよう	247
8.20.2	引数を持っている命令の作り方	247
8.20.3	マクロの名前の付け方	247
8.21	ファイルを分けて \LaTeX を使う方法	248
8.22	標準以外のスタイルファイル	248
8.22.1	日本語 \LaTeX 定番スタイル集の使い方	248
8.22.2	京都産業大学独自のスタイルファイル	249
8.23	さいごに	251
8.23.1	\LaTeX から \AUCTEX へ	251
第 9 章	\AUCTEX	252
9.1	\AUCTEX で \LaTeX 生活が変わる	252
9.2	\AUCTEX の起動	252
9.3	C-c	253
9.4	C-c C-e	253
9.5	C-c {	253
9.6	C-c C-c	254
9.7	C-c ‘	254
9.8	その他の機能	254
9.8.1	部分的なコンパイル	254
9.8.2	ドキュメントの分割編集	255
9.8.3	アウトラインマイナーモード	255
9.8.4	\LaTeX マクロの入力	256
9.8.5	複数行のコメントの付け外し	256
9.8.6	書体の指定	257
9.8.7	数式モードの支援	257
9.9	最後に	258
第 10 章	レポートシステム	259
10.1	レポートシステムを使う前に	259
10.2	レポートを提出するには	259
10.2.1	そのレポートを初めて提出する場合	260
10.2.2	そのレポートを再び提出する場合	260
10.2.3	提出したレポートの確認	260
10.3	返ってくるメールのメッセージ	260
第 11 章	Mathematica	263
11.1	Mathematica ってなあに?	263
11.1.1	とにかく起動、そしてやってみる!	263
11.1.2	その他の場所の Mathematica	265

11.2	ノートブックとコマンドラインについて	266
11.3	Mathematica の簡単な命令	266
11.4	入出力一般	268
11.5	さらに進みたい人には	271
第 12 章	GNU PLOT	272
12.1	概要	272
12.2	GNU PLOT を初めて使う	273
12.2.1	GNU PLOT の起動	273
12.2.2	とりあえずグラフを!	273
12.2.3	コマンドライン編集	274
12.2.4	サンプル数の変更	274
12.2.5	印刷しましょう	275
12.2.6	オンラインマニュアル	276
12.3	GNU PLOT の終了と再開	276
12.3.1	作業の保存	276
12.3.2	GNU PLOT の再開	277
12.4	数値データからグラフを描く	277
12.4.1	二次元データのプロット	277
12.4.2	エラーバー付きの二次元プロット	278
12.4.3	列の選択など	279
12.4.4	フィッティング	280
12.4.5	自作プログラムで GNU PLOT を利用する	281
12.5	マニュアル類について	284
12.5.1	マニュアルを印刷する	284
12.5.2	オンラインヘルプを日本語にする	284
12.5.3	デモンストレーション・ファイル	285
第 13 章	tgif	287
13.1	tgif のできる事できない事	288
13.2	日本語化版 tgif(tgifj) のオリジナル版 tgif との違い	288
13.3	tgif の起動と Window の名称	289
13.4	Popup menu	290
13.5	tgif 関連のツール	296
13.6	終りに	298
第 14 章	xv	299
14.1	はじめに	299
14.2	まずマニアックな人のために	300
14.3	基本操作	300
14.3.1	xv の起動と終了	300
14.3.2	画像を表示する	301
14.3.3	画像を保存する	302
14.4	応用操作	303

14.4.1	画像データを加工する	303
14.4.2	画面に表示されているウィンドウを取り込む	303
14.4.3	印刷する	304
14.4.4	壁紙を貼る	304
14.4.5	色あいを変える	305
14.4.6	Visual Schnauzer を使う	305
14.5	マニュアルなど	306
14.6	法律に注意	306
第 15 章	xpaint	307
15.1	はじめに	307
15.2	まずマニアックな人のために	308
15.3	基本操作	308
15.3.1	xpaint の起動と終了	308
15.3.2	画像を新規に作成する	309
15.3.3	画像を保存する	310
15.3.4	既存の画像ファイルを読み込む	310
15.4	応用操作	310
15.4.1	道具の設定	311
15.4.2	Region	311
15.4.3	拡大表示	311
15.5	マニュアルなど	312
第 16 章	NQS	313
16.1	今どんなバッチキューがあるか?	313
16.2	バッチジョブの投入	314
16.2.1	ジョブの始まりと終わりのお知らせを貰う	315
16.3	ジョブの標準出力	315
16.3.1	標準出力、エラー出力のファイルを変更したい	315
16.4	ジョブの状態表示	316
16.5	流れているジョブの中身を確認する	317
16.5.1	ジョブの記述を確認する	317
16.5.2	流れているジョブの途中経過を確認する	317
16.6	ジョブの実行を保留する	318
16.7	ジョブの実行を停止、削除する	318
16.8	もっと詳しいキューの情報を調べる	319
16.9	エラーメッセージ	319
16.9.1	あなたの所在地はどこですか?	319
16.9.2	警告: tty にアクセスできないため、このシェルでジョブ制御はできません ...	320
16.10	マニュアルなど	320
第 17 章	MODEM から telnet	321
17.1	計算機センター管理のターミナルサーバー	321
17.2	アクセス (接続) するまえに	321

17.3	アクセスしてみましよう!!	322
17.3.1	Annex ターミナルサーバの場合	323
17.4	さらに詳しい使い方	323
第 18 章 NeXT		324
18.1	初めて NeXTStation を使う人へ	324
18.1.1	まず部屋へ	324
18.1.2	電源を入れる	325
18.1.3	login する	326
18.1.4	ちょっとメッセージ	327
18.1.5	基礎知識	327
18.1.6	パスワードを変える (NeXT に初めて触れるなら…)	327
18.1.7	logout する	328
18.1.8	電源を切る	329
18.1.9	部屋を出る	329
18.1.10	マニュアルなど	329
18.1.11	さて、それから	329
18.2	NeXT ファイル管理術	330
18.2.1	はじめに	330
18.2.2	ファイルビューアの各部名称	330
18.2.3	ファイルとは	331
18.2.4	フォルダで整理	331
18.2.5	コピー	332
18.2.6	移動	332
18.2.7	コピーと移動の違い	332
18.2.8	ファイルをフォルダにドラッグすると…	332
18.2.9	名称変更	333
18.2.10	削除	333
18.2.11	リンク	334
18.2.12	アクセス権	335
18.2.13	アプリケーションドック	335
18.2.14	シェルフ	336
18.2.15	フロッピーディスク	336
18.2.16	光磁気ディスク	338
18.2.17	おわりに	338
18.3	NeXT で使うネットワークサービス	339
18.3.1	最初に	339
18.3.2	文通しよう	339
18.3.3	ニュースしよう	343
18.3.4	最後に	347

A	リファレンス	348
A.1	UNIX コマンド	348
A.1.1	ファイル管理に関するコマンド	349
A.1.2	ファイルに関する雑多なコマンド	350
A.1.3	テキスト処理に関するコマンド	351
A.1.4	プリンタに関するコマンド	352
A.1.5	アクセス権、アクセス制御に関するコマンド	353
A.1.6	マニュアルに関するコマンド	353
A.1.7	雑多なコマンド	354
A.1.8	ファイル圧縮などに関するコマンド	355
A.1.9	プロセスに関するコマンド	356
A.1.10	現在使っているコンピュータに関するコマンド	357
A.1.11	利用者に関するコマンド	357
A.1.12	ネットワークサービスに関するコマンド	358
A.1.13	シェル (tcsh) のサブコマンド	359
A.1.14	索引	361
A.2	UNIX でよく使われる記号など	362
A.2.1	シェル変数の一覧	362
A.2.2	環境変数の一覧	362
A.2.3	リダイレクション記号など	363
A.2.4	ファイル指定のワイルドカードなど	363
A.2.5	コマンド履歴を扱う為の表記法	363
A.2.6	正規表現	364
A.3	Mule コマンド	365
A.3.1	絶対覚えておいた方がいいもの	365
A.3.2	必要に応じて覚えるもの	366
A.4	京都産業大学 FAQ(抄)	373
A.4.1	目次	373
A.4.2	はじめに	374
A.4.3	UNIX 編	375
A.4.4	Mac 編	384
A.4.5	Program 編	384
A.4.6	その他	384
B	情報処理教室利用要項	385
C	著作権法 (抜粋)	387
D	参考文献	392

第1章

はじめに

1.1 コンピュータ利用上の注意

まず初めに、学内のコンピュータを利用するに当たっては、個人が大学の共有資源を利用している、と言う事に留意しておいてください。つまり一つの環境に複数の人間が同居するわけですから、そこには規則、慣習、道徳、法律が存在するということです。この章での説明は、まだコンピュータについて全く予備知識のない人については実像をなさない、抽象的に過ぎる部分を含んでいると思われます。そのような場合は、コンピュータの利用に慣れて行く過程で、幾度か読み返して理解して行くのが良いでしょう。

1.1.1 規則

各設備に関してはそれらの管理者がそれぞれに利用の規則を定めているでしょう。使えるようになったからと言って自由奔放に使っていいと言うものではありません。学内のコンピュータ機器はあなたのものになった訳ではないのですから、それぞれの設備の規則を管理者に確認しながら利用してください。本文中にも様々な規則について言及するかも知れません。注意して読み進んでください。

1.1.2 慣習、道徳

大勢の人達でコンピュータを共用する（交替で使う、と言ってもいいかも知れません）のです。皆が気持ちよく利用できるようにお互いに注意してください。自分が何かをしようと思うとき、それが自分で「良い」と思われる目的の為であったとしても、それに関わる作業のために誰かが困ることがあるのではないか、誰かの迷惑になることはないか、そこに注意してください。だからと言って初心者がやたらと引込み思案になる必要もありません。誰か自分より経験があって、良く知っていそうな人に聞けば良いのです。慣習、道徳とは不文律ですので、ここではこれ以上書きません。後はあなた方自身で修得し、実践してください。

1.1.3 法律

一般社会では著作権法というものが存在します。これはコンピュータを利用する上でも適用されます。例えば著作権法によって保護されているものを勝手にコンピュータ上に持ち込んで利用した場合、罪に問われる可能性があります。新聞記事の無断転載などがこの可能性を持っています。法律違反は「知らなかった」では済まされないことを勿論知っていますね？

1.2 サポート

上記のような注意をあなたが遵守して学内のコンピュータを利用していただくのであれば、以下のサポートを受けることが出来ます。

1.2.1 相談窓口

学内のコンピュータのうち、幾らかは計算機センターが管理しています。これらのコンピュータを利用するに当たって質問、要望、トラブルがあれば計算機センター窓口の担当員に尋ねるのがいいでしょう。学内からの電話であれば内線電話 2578、2575、2576 で連絡が付きます。通常の授業が行なわれている期間の授業時間帯であれば計算機運用補助員が各情報処理教室を巡回していますのでその時につかまえて質問してもいいでしょう。

学内には以下の複数の情報処理教室などに多くのコンピュータが設置してあります。あなたは恐らくこれらの設備を利用する事になるでしょう。各室に設置されているコンピュータの名称と所管部署の一覧を挙げておきます。

建物	教室名	機器		所管
計算機科学研究所棟 2 階	C1 情報処理教室	FMV-466D	32 台	計算機科学研究所
計算機科学研究所棟 3 階	C2 情報処理教室	PanaStation	22 台	理学部
計算機科学研究所棟 3 階	C3 情報処理教室	NeXTstation	15 台	計算機科学研究所
計算機科学研究所棟 4 階	C4 情報処理教室	NeXTstation	6 台	理学部
1 号館 2 階	11 情報処理教室	PowerMacintosh6100	90 台	一般教育研究センター
2 号館 4 階	21 情報処理教室	DEC-3300	40 台	理学部
3 号館 2 階	31 情報処理教室	PowerMacintosh6100	44 台	外国語学部
5 号館 1 階	51 情報処理教室	FMR-280H	35 台	経済学部
5 号館 2 階	52 情報処理教室	Compaq Contura	45 台	経営学部
図書館 1 階	パソコンコーナー	DECpc466MTE	12 台	図書館
図書館 1 階	パソコンコーナー	PowerMacintosh6100	12 台	図書館

慣れている人の為の情報：FMV, FMR, Contura, DECpc は DOS パソコンです。Microsoft Windows が動作します。FMV, Contura, DECpc は IBM PC/AT 互換機です。PowerMacintosh は俗に Macintosh もしくはマックなどと呼ばれているパソコンです。PanaStation, DEC-3300 は UNIX ワークステーションです。X Window が動作します。PanaStation は Sun Workstation 互換機です。NeXTstation は俗に NeXT と呼ばれている UNIX ワークステーションです。一般の UNIX ワークステーションとはかなり操作感覚が違います。

情報処理教室の設備を利用する前には、付録の「情報処理教室利用要項」を一読しておいて下さい。特に C2,C4 情報処理教室の設備は特定の学科にのみ提供されているものですので注意して下さい。机、電灯、空調など部屋の設備についての連絡、相談はその教室を管理する所管の事務室をお願いします。コンピュータの操作、動作不良などについての連絡、相談は計算機センター相談窓口までお願いします。

図書館のパソコンコーナーの机、電灯、空調などの設備やコンピュータについての連絡、相談は全て図書館のカウンターをお願いします。

1.2.2 各種ドキュメントはどうやったら得られますか？

各種のドキュメントの入手については上記の計算機センター相談窓口にお問い合わせください。

第 2 章

UNIX はいかが

UNIX¹は現在、大学や研究機関で最も多く利用されているコンピュータ環境の一つです。産業用にも非常に多く利用されており、今後も多方面で利用されるでしょう。京都産業大学にも UNIX 環境が複数あります。ここではその中でも計算機センターが運用しているマシンで利用出来るサービスについて、その概要と利用方法を説明します。

2.1 UNIX 環境

2.1.1 計算機センター運用の UNIX 環境

計算機センターでも幾つかの UNIX コンピュータを運用しています。計算機センターが運用し、京都産業大学の教員、学生に提供しているコンピュータのうち、UNIX 環境のコンピュータは以下のものです。ホスト名とはネットワーク上の各コンピュータの名前です。以降各コンピュータはホスト名で表現します。

機種名	ホスト名	
SPARCcenter2000	cc2000	計算機センター 1 階に設置の Sun 社製コンピュータ。
DEC-3300	csosf01~40	理学部 4 階 21 情報処理教室に 40 台設置の DEC 社製コンピュータ。
NeXTstation	ccns001~015	計算機科学研究所 3 階 C3 情報処理教室に 15 台設置の NeXT 社製コンピュータ。

上記のコンピュータを利用するためには「ユーザ名」と「パスワード」と呼ばれるものを取得して計算機センターに利用申請をする必要があります。この申請によってあなたは「ユーザ名」でコンピュータに登録され、ようやく各コンピュータが利用可能な状態になります。この登録が行われないと、あなたは UNIX コンピュータを利用する際にコンピュータから問い合わせられる「ユーザ名」と「パスワード」に答えられず、結果としてコンピュータを利用することが出来ません。上記のコンピュータを利用するために必要なユーザ名とパスワードは一つだけです。一つのユーザ名とパスワードで上記の全てのコンピュータを利用できます。

ユーザ名とパスワード以外の要素でも上記 3 種類の UNIX コンピュータたちは密接に連係しています。この環境を「cc 環境」と呼ぶことにします。

¹「ゆにつくす」と読んで下さい。

2.1.2 ユーザ名とパスワード

ユーザ名とは「みんなで使うコンピュータ」をある人が利用する際に、誰がコンピュータを使うかを識別するために、利用者個人ごとに一意に決定されたキーワードです。コンピュータ上での利用者の名前だと認識してもいいでしょう。コンピュータ上で扱い易いように8文字以内の英小文字と数字で構成されています。ユーザ名は次に説明する方法によって利用者が自分で決定します。但し、教員も、学生も、京都産業大学に在籍している間はずっと同じ名前で通さなければなりません。**途中での変更は認めませんので、慎重に名前を考えてください。**ユーザ名は電子メールをUNIX上で利用する際の宛名にもなりますので、あなたを連想し易い名前がいいでしょう。逆に複雑な名前を付けると誰も覚えられなくなります。反社会的な名前を付けたりすると誰からも相手にされなくなったり、後で困るかも知れません。また、自分の希望している名前を既に誰かが使っていたとしたら残念ながら別の名前を考えて貰うこととなります。とにかく他の人と重複しないように一意に決まっていることが重要です。

以下に島田 宏明(しまだ ひろあき)さんを例にとって、これからユーザ名を決める人達の参考の為に典型的なユーザ名のパターンを列挙しておきます。

名字もしくは名前	shimada , simada , hiroaki
その組合せ	shimadah , hshimada , shiroaki , hiroakis , simahiro
縮めて	shima , sima , hiro , aki
ニックネーム	shimasan , simasan , shima3 , simachan , simachon , shimachi
ミドルネーム (?) を入れて	has

この、学内で誰も使っていない名前をユーザ名として獲得する作業は **Sign 登録作業**と呼ばれており、まだ **Sign 登録**が済んでいない利用希望者が自分自身で行います。(Sign 登録は京都産業大学独自のものですので、この用語は一般的ではありません。)

2.1.3 パスワードの安全対策

パスワードとは、本人確認のための秘密のキーワードです。毎回、あなたはコンピュータを利用する最初のときにユーザ名を計算機に通知します。しかし誰かほかの人があなたの名前を「騙る(かたる・他人の名前を悪用する)」かも知れません。これでは困るのでユーザ名と一対一に対応した、コンピュータと利用者本人しか知らない秘密のキーワードを用意します。ユーザ名と、それに対応するパスワードの組み合わせを正しくコンピュータに伝えられた者は、その本人に違い無いと言う訳です。ユーザ名は公開のもですが、パスワードは非公開です。誰にも教えないように、知られないようにして下さい。計算機センターでもパスワードが何であったかを調べることは出来ません。忘れないように注意してください。

ところで最近パスワードを調べて悪意に満ちた事をする人がいます。このような人達のために我々が防御策を立てていかななくてはならないのは大変馬鹿馬鹿しい事ですが、それを怠って悪意に満ちた人の攻撃を受けても困ります。そのようなことがないように、まずパスワードを誰にも推測されにくいものにする事を強く推薦します。

UNIXの世界ではパスワードは8文字以内の英文字と数字で作りますが、例えば english とか alphabet 等の英単語は大変危険です。takako などのローマ字の名字や名前、rakkyo のように国語辞典に載っている単語のローマ字も駄目です。単なる数字の列なども危険です。例えば az802g など、全く誰も考え付かないような文字列が最良ですが、これはなかなか覚えにくいかも知れません。以下に覚え易くて安全な例を挙げます。

takotyū	たこちゅー	どこの国の辞書にもありますまい。基本です。
ikasen	いかせん	これも基本形ですね。
iyabaka	いやんばか	さんま風に、、、
iya3	イヤミー	ちょっと古いネタでしたか、、
ku/mi	くしゃみ	記号も考え次第では覚え易くできます。
bin-	ビンボー	まあヤンボーでもマーボーでも良いのですが。
NikkEi*	日経何でもあり	途中で意味もなく英大文字を入れると強力です。

要はアイデア次第でこういう単語は幾らでも湧いてくるものです。途中で記号、数字、英大文字を含ませたりして判りにくさを補強すればなお良いですね。

パスワードは一度判りにくいものに決めてしまった後でも、半年おきなど定期的に変更することが大切です。もしも忘れてしまった場合は計算機センターの相談窓口に学生証を持って来て頂ければ、新しいパスワードでやり直すことが出来ますから安心して下さい。逆に覚え易くしたい余りに、簡単過ぎるパスワードにしてしまったり、一度決めたきり変更しないで使い続ける方が危険です。注意して下さい。

2.1.4 Sign しよう！

Sign 登録するには幾つかの方法が提供されていますが、ここではそのうちの幾つかを示します。

- 計算機センター相談窓口で担当員に登録して貰う。
但しこの方法だと春先など込み合う時期はかなり時間的に待たされる可能性があります。時間的に空いていそうなときに来てください。
- 2号館の4階21情報処理教室にあるDEC-3300を利用して登録する。
この方法だと窓口が込んでいても待ち時間無しで登録できます。
- 学内のUNIXコンピュータを利用している知人（ないしは指導教員）に手伝って貰う。
この方法だとわざわざ2号館4階まで行かなくても作業が出来ます。cc2000コンピュータにユーザ名 sign でリモートログインすると、DEC-3300を利用した場合の「いよいよ Sign 登録」での作業と同じ事が出来ます。

ここではDEC-3300を利用してSign登録をする手順を紹介します。これだと誰にでも簡単に出来て、計算機センターの手間も省けますので、ぜひこちらにチャレンジしてください。

部屋に入ってDEC-3300の電源を入れる

部屋に入って電源を入れる説明については21ページの3.2.1と、その次の3.2.2に詳しく書いてあります。ちょっと回り道ですが、そこを参照して下さい。

いよいよ Sign 登録

電源をうまく入れることが出来たら以下のような絵が画面中央に表示されているでしょう。

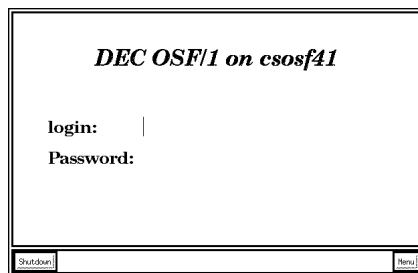


図 2.1 電源投入後に現れる画面

この画面の右下隅に注目して下さい。次のように小さく Menu と書かれた枠が見えるでしょう。



図 2.2 Menu ボタン

マウス（下図参照）の前後左右の動きに連動して画面上を上下左右に動く小さな×印をこの Menu の文字の上に重ねてマウスの左ボタンを一度だけ押してすぐ離します。このマウスのボタンを押してすぐ離す操作の事をクリックと呼んでいます。



図 2.3 マウス

そうすれば画面には新しく枠が表示され、そこには以下のように表示されているでしょう。


Sign 登録システム. version 0.3

- 1 学生
- 2 教員
- 3 院生
- 9 中止

どのような人のための登録か選んで下さい (1-9) ?

これ以降は学生ユーザ安田さんが yasu3 というユーザ名で Sign 登録する例を用いて説明します。例を見ながら、実際にはあなた自身の情報をタイプして行って下さい。教員、大学院生でも操作そのものは殆ど同じです。いずれにしても表示される質問に次々と答えて行くだけです。

コンピュータに対しての返事はキーボードのキーをタイプする事で行ないます。タイプするのはキーボードのキーをその通りに一つずつ押す事です。キーは押しっぱなしにすると繰り返し

て押した事になりますので、押したら指を離して下さいね。もしもキーを打ち間違えた時は delete キー () を押して下さい。一度押すたびに一文字ずつ戻ります。大抵の場合、指定の文字がタイプ出来たらリターンキーを押す必要があります。リターンキーはアルファベットの並びの右端にある、ひときわ大きなキーです。「Return」と書いてあります。

Sign 登録システム. version 0.3

- 1 学生
- 2 教員
- 3 院生
- 9 中止

どのような人のための登録か選んで下さい (1-9) ?

まず、どの種類の人かを尋ねてきています。安田さんは学生ですから、1 をタイプしてリターンキーを押しました。

どのような人のための登録か選んで下さい (1-9) ? 1

何人も続けて登録しますか (y/n) ?

ここでは繰り返して何人も登録するかどうかを尋ねています。今回は安田さん一人分しか登録しませんから n をタイプしてリターンとしました。

何人も続けて登録しますか (y/n) ? n

Sign registration program by Y.Yasuda.

学生番号を入力して下さい :

ここでは登録する人の学生証番号を尋ねています。安田さんの番号は 473088 ですから、473088 とタイプしてリターンとしました。

学生番号を入力して下さい :473088

あなたの名前は... :Yutaka Yasuda

誕生日は何月何日? (mdd) :

学生証番号をタイプするとフルネームを表示して来ますから、間違いないことを確認してください。そして今度は更に確認のために誕生日を聞いてきます。誕生日は月と日をそれぞれ二桁で表します。例えば安田さんの誕生日は 1 月 28 日ですから、タイプする内容は 0128 です。

誕生日は何月何日? (mdd) :0128

希望のユーザ名は何ですか :

誕生日が間違いないければ、いよいよ希望のユーザ名を尋ねてきます。安田さんは素直に名前の通り yasuda が良いなと思って yasuda とタイプしてリターンしました。

希望のユーザ名は何ですか :yasuda

** 'yasuda' は既に他の人に使われています

希望のユーザ名は何ですか :

すると、上記のようなメッセージが表示されて、もう一度ユーザ名の希望を聞いてきました。つまり yasuda は既に他の人が使っているのです。仕方がありませんから安田さんは「やっさん」と呼んでねという積りで yasud3 とタイプしてリターンキーを押しました。

勿論一回で誰も使っていないユーザ名を見つけられる人もいるでしょう。逆に何度やっても自分の希望するユーザ名が全て使われていて、ちょっと考え直したい人もいるかも知れません。そういう場合はユーザ名の代わりに `exit` とタイプしてリターンすれば Sign 登録作業を中断することが出来ます。すると恐らくは「リターンキーを押せば終了します」というメッセージが現れますから、そこでもう一度リターンキーを押せば Sign 登録作業を終了することが出来るのです。

希望のユーザ名は何ですか :yasu3

```
keyword 'yasu3' was registered in resource list.
```

```
**
```

```
** あなたのユーザ名は 'yasu3'
```

```
**      パスワードは 'tdwa'
```

```
** です。覚えて下さい。
```

```
**
```

リターンキーを押せば終了します

yasu3 はどうやら誰も使っていなかったようで、うまく登録できました。ここで表示されているユーザ名（例では yasu3）と初期パスワード（例では tdwa）をどこかに書き留めて記録してください。後で利用申請を計算機センターに出す時と、利用者登録が済んだ後で必要になる情報です。

ひょっとするとここでも何等かの失敗があって、もう一度やり直す必要があるかもしれません。その場合は上記の例とは違って、エラーがあったからやり直してね、という表示が出ますので注意してください。

ユーザ名とパスワードをメモしたら、この状態を終るためにリターンキーを一度押してください。

DEC-3300 をシャットダウンして電源を切り部屋を出る

シャットダウンから電源を切って部屋を出るまでの説明については 27 ページの 3.2.8、3.2.9、3.2.10 に詳しく書いてあります。再びちょっと回り道ですが、そこを参照して下さい。

2.1.5 ユーザ登録申請

Sign 登録が終わったら計算機センターにて自分を Sign 登録したユーザ名で登録手続きをしましょう。そのためには計算機センターが主催する「コンピュータ利用オリエンテーション」²に参加した際に配布された「利用登録申請書」に必要事項を記入し、計算機センター相談窓口の担当員に提出してください。提出の際に再び「騙り（かたり）」を防ぐために必ず本人が、学生証をもって来てください。提出された書類を元に、計算機センターがいつ登録作業を行うかは時期によってまちまちです。急いでいる人はいつ利用可能になるか窓口で確認しておくべきでしょう。

2.2 これ以降の読み方

ユーザ登録が済めばいよいよ UNIX コンピュータを利用することになります。本当はここでいきなり UNIX コンピュータを使わせてあげたいのですが、いろいろと予備知識が必要になると言うのが実情です。つまり UNIX コンピュータを利用するところまでたどり着く前に、あなたのコンピュータに対する理解度、

²まだ参加していない人は次の開催スケジュールを今すぐ掲示板でチェックしましょう。計算機センター相談窓口前にも掲示しています。

環境に応じて事前の学習が必要になると言うことです。これ以降はあなたにとって必要な部分を読み進み、UNIX コンピュータにまでたどり着いてください。

但し計算機科学研究所棟 3 階の C3 情報処理教室に設置してある NeXT Station は、UNIX コンピュータには変わり無いのですが、その使い方はかなり他の UNIX コンピュータと違って独特です。そのためこのガイドではその使い方を一緒に説明することはしません。第 18 章にまとめてありますのでそちらをご覧ください。

これ以降は以下の事項についてそれぞれ説明します。あなたの理解度に応じて必要な所を「つまみ読み」してください。

- login と logout
- キーボード
- ウィンドウ環境
- どこから login?
- コンピュータウイルス

この章の最後には、具体的に UNIX コンピュータを利用する時にどの部屋に行って作業するべきかについて説明します。この所は特に重要です。では、第 3 章まであなたの理解度に応じて読み進みましょう。少々読むのが面倒でもくじけずに、頑張ってください。


2.3 login と logout

UNIX コンピュータを利用する際には、先に述べたユーザ名を使って、誰が UNIX コンピュータをこれから利用するのかをその都度確認する作業があります。これを login と呼びます。また、利用が終了したときに今まで使っていたコンピュータを解放して、他の人が使えるようにするために行う作業を logout と呼びます。login してから logout するまでの間を「セッション」などと呼ぶ場合もあります。login する代わりに「セッションを開始する」、logout する代わりに「セッションを終了する」などと表現するときもあるでしょう。

この login を実際に行なうには、様々な方法があります。後でまとめて説明します。

2.4 キーボード

コンピュータで作業する為には、キーボードによって文字を打ち込まなければならない事が多いと思います。特に UNIX コンピュータは殆どの場合「あれをしろ」とか「これをしろ」と、決められた文法で指示を（呪文のような）文字列としてコンピュータに入力する事によって操作しますから、キーボードは操作には欠かせません。

- キーボードのキーを、入力したい文字の順番に押して行くことを「タイプ」もしくは「打ち込み」などと言います。コンピュータの操作関係の説明で、「～～とタイプする」もしくは「～～と打ち込む」などと書いてあった場合は、その通りキーを押すのだなと解釈してください。
- 単語と単語を分けるために空白を一つ入力したいときはキーボード最下段の一番長いキーを押します。
- 普通にアルファベットのキーをタイプしただけだと一般的には英文字の小文字 (a b c など) が入力されます。英大文字 (A B C など) を入力したい場合はシフトキーを押しながらアルファベットのキーを押すことによって行います。シフトキーはキーボード上に二つあり、左側ならアルファベットの並びの最も左下にある「Z」のすぐ左、右側ならアルファベットの並びの最も右下にある「M」のいくつか右にあります。キーには「shift」などと書いてあるでしょう。キーボードによっては上矢印が書いてあったりします。
- 文字を入力していて次の行に行きたくなくなったり、コマンド（後述）を入力し終って実行させたいときには「リターン」キーを押します。改行キーは右側のシフトキーの真上、もしくは二段上くらいにあります。「Return」「改行」などと書いてあります。
- 打ち間違えた文字の修正には delete キーを利用します。一度 delete キーを押すと、一つ前の文字が消えます。delete キーは大抵改行 (return) キーの真上、もしくは二段上くらいにあります。一般的には「DEL」「delete」などと書いてあります。文字ではなく  や、←などと絵で表現してある場合もあります。
- UNIX で時々「コントロールC」などと言う表現を使うときがあります。表記では Control - C や C-c などと書かれている事が多いようです。この場合はコントロールキーを押しながらアルファベットの C キーを押すことを意味しています。コントロールキーは、大抵アルファベットの A キーの左か、左シフトキーの下かどちらかにあります。キーには大抵「Control」「CTRL」などと書いてあります。

2.5 ウィンドウ環境

最近のコンピュータでは当たり前になってきた、ウィンドウ環境の操作方法について説明しておきます。ウィンドウ環境と曖昧に呼んでいますが、厳密な呼び名はありません。最近のコンピュータはユーザの操作がやりやすい様に、画面に絵柄を表示します。つまりそれぞれの絵柄に意味を持たせ、絵で表示されるものを操作させることでコンピュータにユーザの希望を伝えようと言うわけです。勿論コンピュータがユーザに伝えたい希望（もしくは情報）も絵で表示できるわけで、要はコンピュータとユーザの対話に絵柄を利用しよう、という訳です。このアイディアはアメリカから輸入されたものですから、一般にはグラフィカルユーザインタフェイスなどと英語で呼ばれています。

コンピュータの種類に応じて様々なウィンドウ環境が使われています。UNIX では X Window と呼ばれるシステムをベースにしたウィンドウ環境がほとんどです。パソコンでは Microsoft Windows や Macintosh のいずれかでしょう。

これらのウィンドウ環境でキーボードに代わってコンピュータの操作に活躍するのが次に説明する「マウス」です。

2.5.1 マウス

「マウス」とはキーボードの横に置いてある以下のようなものです。



図 2.4 マウス外観

幾つかのボタンが上の方（奥の方）に付いており、その向こうに細い電線が付いています。ボタンの数はコンピュータの種類によってまちまちです。手前側を頭に見立てると、向こうのおしり側に尻尾が付いている形からネズミのつもりでマウスと呼んでいます。利き手で握って机もしくは専用の台の上を前後左右に滑らせて使います。

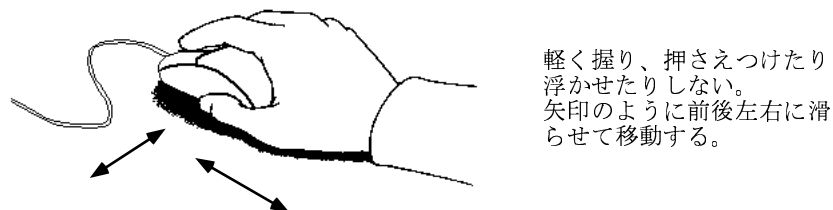




図 2.5 マウスの握り方と移動

初めはちょっと使いにくいと感じるでしょう。これは練習して慣れるしかありません。滑らせて動かすことに注意して下さい。強く握って押さえついたり、前後左右に動かす時に浮かせたりしないように注意して下さい。マウスはキーボードでは指示できない、画面に表示されている「もの」に対して指示を与えるために利用します。画面を良く見ると、マウスの前後左右の動きに連動して画面上を上下左右に動く小さ

な矢印 （もしくは小さな **X** 印）が見つかるでしょう。これを指示したい対象（絵柄）の上まで運んでいって、マウスボタンの操作によって動作をその対象に対して起こさせるのです。

- 小さな矢印  を「指し示すもの」という意味でポインタ、マウスポインタと呼びます。マウスポインタは状況に応じて形が変わります。例えば **X** などになることもありますが、これも同じくマウスポインタと表現します。
- マウスにはボタンが幾つか付いています。それぞれ左ボタン、中ボタン、右ボタンなどと呼んでいます。
- マウスのボタンを押しっぱなしにすることを「プレス」と呼んでいます。
- マウスのボタンを押してすぐ離すことを「クリック」と呼んでいます。
- マウスのボタンを二度続けて短い時間間隔でクリックする（要するに続けて二回クリックすること）を「ダブルクリック」と呼んでいます。一度目のクリックと二度目のクリックの間にマウスが移動しないように注意して下さい。ボタンを強く押さえている人は力が余ってマウスも一緒に押してしまうようです。
- マウスのボタンをプレスして、そのまま移動することを「ドラッグ」と呼んでいます。目標の場所までマウスポインタが移動したら、ボタンを離します。
- 上記のクリック、ドラッグなどの操作の説明で、特にボタンを指定せずに「クリックする」と表現している場合は左ボタンで操作を行なうことを意味しています。ボタンが一つしかないマウスでは当然「その」ボタンです。
- マウスを動かしていて、例えばもっと右端までポインタを移動したいのにマウスを更に右に動かすスペースがなくなってしまう時があるかも知れません。この場合は一旦マウスを持ち上げて少し左にマウスを運び、それからまたマウスを右に滑らせます。
- ポインタを選びたいものの絵柄の上まで持っていって、どのボタンでクリックすると（もしくはダブルクリックすると）どのような反応をするかは状況に応じてまちまちです。一応標準的な規約はあって、このような絵柄のものに対してクリックするとどう反応すると言う事が決まってはいるのですが、そう完全でもありません。これについては慣れて行くしかないでしょう。

2.5.2 ウィンドウ環境の画面

では、DEC-3300 の login 直後の標準的な画面表示を例に、ウィンドウ環境のものの呼び方を説明します。

ウィンドウ

なにはともあれウィンドウ（窓）です。ウィンドウ環境では一般的に以下のような四角い枠の中にさまざまな情報が表示されます。

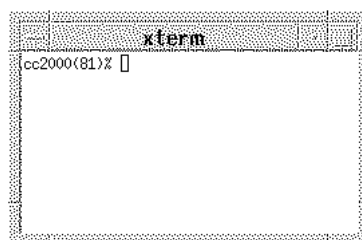


図 2.6 ウィンドウの例

このような枠（窓？）のことを「ウィンドウ」と呼んでいます。ウィンドウはコンピュータの画面の中に一つだけとは限りません。大抵幾つも表示させて使うことになるでしょう。つまり一つの画面の中に複数の小さな画面が幾つも作れる、と言う訳です。

ウィンドウの移動

ウィンドウは画面上のどこにでも配置することが出来ます。ウィンドウの枠の上の部分を見ると、そこにはタイトルが表示されている部分があると思います。



図 2.7 ウィンドウのタイトル部分

このタイトル文字の真上にマウスポインタを移動して、そのままマウスの左ボタンを押してドラッグ（ボタンを押しっぱなしにしたまま移動）します。するとポインタの移動と共にウィンドウ、もしくはウィンドウの外枠が移動するのが判るでしょう。目的の地点までドラッグしたら、マウスのボタンを離します。

ウィンドウの上下関係を変える

ウィンドウを同時に二つ三つ出す程度ならせいぜい重ならない場所に移動するだけでいいのですが、画面の大きさには限りがあって、大抵ウィンドウは重なりあってしまいます。下の図のような状態です。

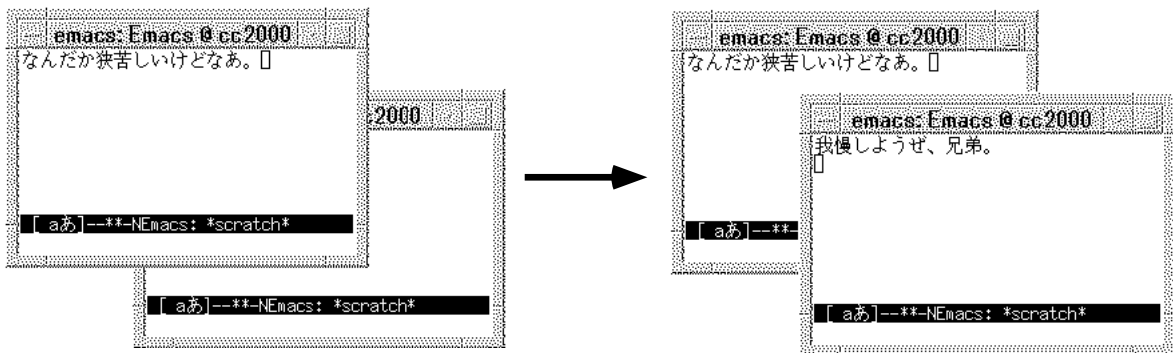


図 2.8 ウィンドウの上下関係を変える

ウィンドウの上下関係を変えたい場合は、自分が上に持てきたいと思うウィンドウのどこかの部分をマウスの左ボタンでクリックします。簡単ですね。この図では初め左上のウィンドウが右下のウィンドウを下敷にしていたのですが、右下のウィンドウの見える部分をクリックすることによってその上下関係が逆転したところを示しています。

ウィンドウを選ぶ

現在のウィンドウ環境ではコンピュータに対する指示は、画面上のたった一つのもの（例えばウィンドウ）にしか同時には行なえません。つまりどのような操作にしても、その指示対象はたった一つのものに向

けられているのです。そのため、常にウィンドウ環境ではたったひとつのウィンドウが指示対象として「選ばれて」います。この選ばれたウィンドウのことを「アクティブウィンドウ」と呼んでいます。「そのウィンドウを指示対象として選ぶ」ということを「そのウィンドウをアクティブにする」などともいいます。

目標のウィンドウをアクティブにするのは簡単で、単にそのウィンドウを一度マウスの左ボタンでクリックするだけです。アクティブなウィンドウは常に画面上ではもっとも上に来ています。先のウィンドウの上下関係を変える例では、アクティブなウィンドウも右側のものに切り替わっているというわけです。

2.5.3 メニュー

ウィンドウ環境で指示を与える方法として、「メニュー」によるものがあります。つまり一覧から自分の望みのものを選ぶというやり方です。勿論一覧の中に自分の望みの指示が含まれていなければいけません。メニューの形はいろいろあって、そのすべてについて網羅的に紹介することは出来ません。ただ、すべてに共通のやり方としては、項目ないしは絵柄、場合によっては画面の背景などをマウスでクリックすればそこから一覧が湧いて出て、その中から自分の望みの指示をマウスでクリックすることによって選択します。ウィンドウシステムによっては項目ないしは絵柄をプレスすればそこから一覧が湧いて出て、その中から自分の望みの指示の項目までドラッグすることによって選択する場合があります。

ここでは例として DEC-3300 に見られるメニューを挙げます。

ルート・メニュー	
Kterm	
Mule	
アクセサリ	▶
手前へ(U)	
奥へ(O)	
次のウィンドウ(N)	
前のウィンドウ(v)	
アイコン整理(P)	
オプション(O)	▶
ヘルプ(H)	▶
再起動(R)...	
セッション終了	

図 2.9 メニューの例：DEC-3300 のルートメニュー

このメニューには 10 以上の項目が用意されています。ここで例えば「セッション終了」の行までマウスポインタを移動させてからマウスボタンを離すと、それはつまり「ルートメニューからセッション終了を選択し、実行する」ということを意味しています。

2.5.4 ボタン

ウィンドウ環境で指示を与える方法として「ボタン」³によるものがあります。つまりあるボタンを押すことによって、そこに割り当てられた動作を指示するという訳です。ボタンの形や配置はいろいろあって、そのすべてについて網羅的に紹介することは出来ません。

ここでは例として DEC-3300 のセッション終了時に見られるボタンを挙げます。

³マウスのボタンと混同しないように書かなくては！

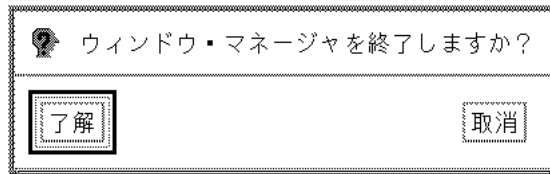


図 2.10 ボタンの例：DEC-3300 のセッション終了時の確認

このウィンドウには「了解」と「取消」の二つのボタンがあります。ここで利用者はそのどちらを実行するかをマウスの左ボタンでクリックすることによってコンピュータに指示するというわけです。

ちなみにこのウィンドウのように、ユーザに強制的に指示を要求する小さなウィンドウのことを「ダイアログ」と呼ぶ場合もあります。

2.5.5 ウィンドウ環境のトラブル傾向と対策

ここでは良く起きるトラブルの傾向とその対策を示します。但しこれですべてのトラブルが解決する訳ではありませんし、本当にコンピュータが故障したのかも知れません。ですから対処は慎重に、ゆっくり確かめながら行なって下さい。

とにかくトラブルなんですけど！

これ以降をじっくり読んで、自分が該当しているトラブルがあるかどうか調べて下さい。どうしても当てはまらない場合はまず良く知っていそうな人に聞くのがいいでしょう。それでも解決しない場合は計算機センターの相談窓口まで連絡してください。気が短い人やパーソナルコンピュータを使い慣れている人の中には突然コンピュータの電源を切ったりする人もいますが、決してそれだけはしないで下さい。特に UNIX コンピュータは非常に壊れやすく、突然電源を切ったりすると次に電源を入れても二度と立ち上がらなくなるが多々あります。

キーボードをタイプしても、目的の場所に文字が入力されない

目的のウィンドウがアクティブになっていないのではないのでしょうか。目的のウィンドウを一度左ボタンでクリックして下さい。

それでも直らない場合は C-q (コントロールキーを押しながら「Q」キーを押す) してみてください。

キーボードをタイプしたらカタカナもしくは変な文字が入力される

キーボード上の「かな」キーを押してしまったのではありませんか？もう一度「かな」キーを押すか「英数」キーを押すなどして解除して下さい。

目的のウィンドウが他のウィンドウの下敷になって見えなくなってしまった

仕方がありません。上にあるウィンドウを移動させて下敷になっているウィンドウが見えるように場所を変えます。ひょっとするとウィンドウの背景(どのウィンドウの上でもないところ)にマウスポインタを持って行って左ボタンをプレスすれば「次のウィンドウ」「前のウィンドウ」などというメニューが現れるかも知れません。それらを使って奥にあるウィンドウを前に出してくれることも出来ます。

2.6 どこから login?

UNIX コンピュータを利用するには以下の二つの形態があります。

- UNIX コンピュータの前に直接座ってそれを利用する。
- UNIX コンピュータを、離れた所にあるパソコンから遠隔操作で利用する。

UNIX コンピュータを遠隔操作で利用するということは、UNIX コンピュータとは離れた所にある別のコンピュータから UNIX コンピュータに login するという事を意味しています。

当然のことですが、UNIX コンピュータを利用するには直接その前に使って利用の方が操作は簡単です。しかし UNIX コンピュータは台数に限りがある上、設置場所が限られていますから、学内のどこに居ても使えると言うわけではありません。それに対してパソコンであれば多くの建物に共同利用のものが置いてあります。様々な事情⁴から UNIX コンピュータの前に直接座るのではなく、これらのパソコンの前に座って作業したい場合があるでしょう。そのような時でも遠隔操作によってパソコン経由で UNIX コンピュータを利用することが出来るのです。

但しこの場合は UNIX コンピュータそのものの使い方に加えて、そのパソコンの使い方もある程度は理解する必要が有りますから少し操作は難しくなるかも知れません。初心者は可能であれば UNIX コンピュータの前に直接座って操作することをお勧めします。

具体的に cc 環境の UNIX コンピュータを利用するには、以下のような方法となるでしょう。自分の事情に合わせてどの部屋から利用するかを決めて下さい。それによって UNIX コンピュータの前に直接座るか、パソコンの前から遠隔操作するかが決まります。

種別	場所	部屋名	設置されているコンピュータ
直接操作	2号館4階	21 情報処理教室	DEC-3300
遠隔操作	1号館2階	11 情報処理教室	PowerMacintosh 6100/66
	3号館2階	31 情報処理教室	PowerMacintosh 6100/60
	5号館1階	51 情報処理教室	FMR-280
	5号館2階	52 情報処理教室	Conpaq Contura
	計算機科学研究所棟2階	C1 情報処理教室	FMV-466D
	図書館1階	パソコンコーナー	DECpc466MTE
	図書館1階	パソコンコーナー	PowerMaintosh 6100/66

もしも直接 UNIX コンピュータを利用しようと考えた場合は、具体的には 21 情報処理教室の DEC-3300 に向かう事になります。これ以降の説明は読み飛ばして、第 3 章から読み続けて下さい。

もしも遠隔操作で UNIX コンピュータを利用しようと考えた場合は、具体的には各情報処理教室のパソコンに向かうこととなります。これ以降に書かれている節を順々に読み進んで下さい。

2.6.1 パソコンから UNIX を利用する為の予備知識

ここではパソコンから UNIX コンピュータを遠隔操作で利用するための最低限の予備知識を説明します。あまり厳密に解釈しなくて結構ですので、イメージだけを頭に描いて読んでください。

遠隔操作

遠隔操作と言いましたが、それほど不便でも面倒でもありません。身近な遠隔操作の例としてビデオデッキを挙げておきましょう。ビデオテープを見る際に、ビデオデッキの再生ボタンを押す場合もあるでしょ

⁴いつも授業で使っていて空気がない、部屋が遠い、階段を上るのが嫌、など

うが、リモコン⁵の再生ボタンを押すときもあるでしょう。ビデオデッキに当たるのが UNIX コンピュータ、リモコンに相当するのがパソコンだと思ってください。つまり UNIX コンピュータに直接向かう代わりにパソコンからリモコン操作で UNIX コンピュータを利用しようと言うことです。

ビデオデッキのリモコンは、デッキの方を向けて「ぴっ」とやってやるだけで良いのですが、コンピュータのリモコンは現在のところ電線をつないでやらないといけません。この役割りを果たす電線がすなわち京都産業大学のコンピュータ・ネットワーク⁶というわけです。ここでリモコンとして紹介するパソコンは全て予めネットワークにつながっていますから、UNIX コンピュータをパソコンから利用する場合にあなたがいちいち電線をつないだりする必要はありません。

用語について

UNIX コンピュータなどの為の遠隔操作機器はビデオデッキなどの場合と違って「リモコン」とは呼ばず、一般的には「端末」ないしは「ターミナル」と呼ばれます。端末を経由して利用しようとするコンピュータそのものはビデオデッキの場合と違って「本体」などとは呼ばず、一般的には「ホストコンピュータ」ないしは短く「ホスト」などと呼ばれます。UNIX コンピュータはほぼ例外なくホストコンピュータとなる為の機能を持っています。UNIX コンピュータ以外のコンピュータでもホストコンピュータとなる為の機能を持っているものは幾らもあります。

端末とホストコンピュータは常にネットワークで接続されていますが、ネットワーク上には複数のホストコンピュータが存在します。一般的にネットワークに接続された端末は、ホストコンピュータをどれにするか選ぶことができます。ちょうどリモコンを「ぴっ」とやる時に、向けた方向によって指示するビデオデッキを選べる⁷ような感じです。つまり端末を使い始める最初の時に、まずどのホストコンピュータを利用するか端末に指示してやるのです。この過程を「(端末をホストコンピュータに) 接続する」と表現しています⁸。但し、実際には cc 環境で遠隔利用する場合のホストコンピュータは cc2000 マシンになると思います。それ以外のマシンをホストに選ぶことはしないで下さい。これ以降の説明でもホストには cc2000 を利用することを前提に行なっています。

例えばあなたが C1 情報処理教室の FMV の前に座って遠隔操作で cc2000 を利用している場合、「FMV を端末に cc2000 をホストコンピュータとして利用している」などというように表現します。

パソコンによる UNIX 端末 : telnet 環境と X 環境

パソコンは UNIX コンピュータとは別の独立したコンピュータですが、UNIX コンピュータを遠隔操作する機能を追加することが出来ます。この「UNIX コンピュータのリモコンとなる為の機能を実現するもの」を端末ソフトないしはターミナル・エミュレータ・ソフトと呼んでいます。

端末ソフトが提供する UNIX 環境は大別して二種類あります。文字情報だけを扱う telnet⁹環境と、X ウィンドウによる文字情報とグラフィクス情報の両方を扱う X¹⁰環境です。様々な要因がある為、telnet 環境と X 環境のどちらがあなたにとって使いやすいかは一概には言えません。いろいろ試してみるのが良いでしょう。

2.6で遠隔操作として紹介した情報処理教室の全てのパソコンに telnet 端末ソフトが用意されています。また、計算機科学研究所 2 階 C1 情報処理教室の FMV-466D には X 端末ソフトが用意されています。

⁵ Remote Controller、リモート・コントローラの略。即ち遠隔操作機器のことです。

⁶ このネットワークには KING(「キング」と読んでください)という名前が付けられています。

⁷ 現実には余程広い部屋の四方の隅にビデオデッキを置かないとうまく行かないでしょうけど。

⁸ これはネットワークなどと言うものが現れる以前には端末にホストを選ばせる事が出来ず、そうしたい場合は端末から出ている電線を実際に別のホストコンピュータにつなぎ変えていた歴史的経緯から来た表現だと思います。

⁹ 「てるねっと」と読んでください。

¹⁰ 単に「えっくす」と読んでください。

2.7 コンピュータウイルス

大変残念なことです。コンピュータを利用する上でコンピュータウイルスが非常に脅威となっています。パーソナルコンピュータを利用する上でもこれを避けて通ることは出来ません。特に計算機センターが管理しているパーソナルコンピュータにはマックと IBM PC 互換機が含まれており、これらは他の国産のパーソナルコンピュータに比べてウイルスの多さなどの理由からウイルス感染の危険性が非常に高いことを意識しておく必要があります。

パーソナルコンピュータに感染したウイルスが引き起こす害悪は様々です。例えば利用者の書類を一瞬で消したり、書類の中の数字をほんの一分だけ加算したりします。それは長時間の努力の積み重ねの仕事を消し去ったり、間違った結果をもたらします。ウイルスに感染することは被害者にとって悲しく腹立たしいことです。しかし、もっと残念なのは、自分が感染していることを知らずに人にもウイルスを移してしまうことです。

情報処理教室のパーソナルコンピュータはいつもウイルスの脅威にさらされています。いつ感染するか分からないし、いつあなたが被害を被るか分からないからです。このような理由から、ウイルスの被害を最小限にするため各情報処理教室への「外部ソフトウェア」の持ち込みは原則として認めておりません。

ここで言う「外部ソフトウェア」とは情報処理教室のパーソナルコンピュータに正式に認められていないソフトウェアを指しています。アプリケーション、システムソフトウェア等がこれに該当します。これらのソフトウェアの持ち込みがウイルス感染の原因となっているからです。

ウイルスからあなたの書類を守るために以下のことに気を配ってください。情報処理教室のパーソナルコンピュータにはウイルス除去用のソフトウェア (scan, Disinfectant 等) が入れてあります。

時々で良いのですが、そのウイルス除去ソフトウェアであなたがこれから使おうとするパーソナルコンピュータと、あなたのフロッピーを調べてください。このソフトウェアの動かし方が分からなければ、誰か知って良そうな人に尋ねるか、計算機センター相談窓口聞いてください。

もし人からフロッピーを預かって仕事をしなければならぬときは、そのフロッピーがコンピュータ・ウイルスに感染していないことを確かめてから仕事をする習慣を付けてください。

もし自分のフロッピーがコンピュータウイルスに感染していることが分かったら、面倒ですが計算機センターの相談窓口ご連絡してください。あなたのフロッピーからウイルスを除去するのは簡単ですが、むしろウイルスの感染経路を特定し、他にも感染している人がいないか調べる必要があるからです。

第 3 章

UNIX への道

ここでは様々な場所の様々なコンピュータから cc 環境の UNIX マシンを利用する方法について、その部屋ごと、そのコンピュータごとに説明します。これまでの説明の部分であなたはもう既にどの部屋のコンピュータから login するか決めているはず¹。

既に説明したように cc 環境の UNIX コンピュータを利用する場合、21 情報処理教室以外の部屋からはパソコンを利用して telnet もしくは X によって遠隔操作で cc2000 マシンを利用する事になります²。

3.1 これ以降の読み方

これ以降は部屋ごと、コンピュータごとに説明を行いません。説明は節を分けて並列になっていますが順に読む必要はなく、あなたが利用しようと思っている部屋の部分についてだけ読めば良いでしょう。

- 21 情報処理教室の DEC-3300 を使う (21ページより)
- 11, 31 情報処理教室及び図書館パソコンコーナーの Macintosh から telnet で cc2000 を使う (29ページより)
- 51 情報処理教室の FMR から telnet で cc2000 を使う (37ページより)
- 52 情報処理教室の Contura から telnet で cc2000 を使う (48ページより)
- C1 情報処理教室の FMV から X で cc2000 を使う (57ページより)
- C1 情報処理教室の FMV から telnet で cc2000 を使う (66ページより)
- 図書館パソコンコーナーの DECpc から telnet で cc2000 を使う (75ページより)

繰り返しますが初心者には UNIX コンピュータを直接操作の方が簡単ですので、最初は 21 情報処理教室の DEC-3300 を利用されることをお勧めします。

C1 情報処理教室の FMV に関しては telnet と X と両方の方法による login が可能ですので、FMV に限ってはその両方とも説明しています。どちらを使えば良いのか判らない人は X 環境を選択して下さい。

¹まだ決まっていなければあなたは重要なところを読み飛ばしています。今すぐ 8ページの 2.2まで戻して下さい。

²telnet, X, 遠隔操作という単語に聞き覚えがなければやはりあなたは重要なところを読み飛ばしています。今すぐ 8ページの 2.2まで戻して下さい。

全ては例示を伴っていますので、この段階でとにかくその部屋に行って実際に作業しながら読み進んでいけばいいでしょう。各節ではその部屋に入ってコンピュータの電源を入れるところから login するまでと、そのコンピュータでの作業が終って logout して電源を切るまでの一連の操作を説明しています。実際に試してみて、一連の操作が出来るようになったら第 4 章に進んで下さい。

3.2 DEC-3300 を使う

DEC-3300 は二号館 4 階の 21 情報処理教室に設置してあります。DEC-3300 それ自身を利用するばかりでなく、計算機センターが用意している SPARCcenter2000 や、DEC-3500 を利用する場合の窓口ともなるでしょう。

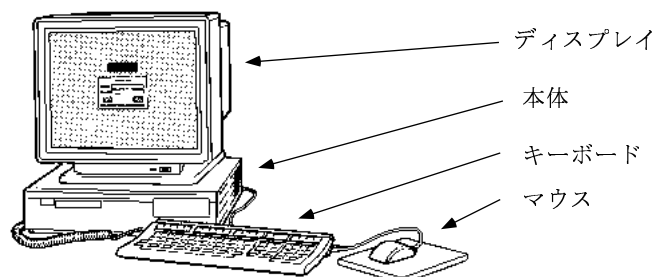


図 3.1 DEC-3300 外観

3.2.1 まず部屋へ

ともあれ、置いてあるところへ行ってみましょう。DEC-3300 は二号館 4 階の 21 情報処理教室に設置してあります。この部屋は常時カードロックシステムが働いており、学生証を入り口のドア横の機械に通すことによって鍵が開きます。入室可能な時間帯、曜日、期間に限りがありますので、付録の「情報処理教室利用要項」を一読してください。もしもカードを通して鍵が開かない場合は、カードの向き、表裏、カードを通過させる速度（遅すぎても速すぎても駄目）を変えてやってください。どうしても駄目な場合はカードの磁気情報がおかしくなっている可能性があります。カードを通す機械にあなたのカードでなぜ鍵が開かないか、その理由が出ていますので、それを控えて計算機センターまで連絡してください。

部屋に入るときは、そこで授業をやっていないことを確認しましょう。もし授業中であれば入室して利用していいかどうか指導教員に確認を取るのが礼儀と言うものです。部屋に誰も居なかった場合は照明、空調機が切れている場合があります。これらのスイッチは部屋に入ったそのすぐ右手の壁に集中して置いてありますので、自分で適当に操作してください。部屋の管理そのものは理学部事務室が行っていますので、何か部屋の状態について質問、要望があれば理学部事務室までお願いします。コンピュータについての質問、要望は計算機センターの相談窓口までお願いします。

3.2.2 電源を入れる

自分が利用するコンピュータを選びます。これが最初の利用であれば、電源の入っていないマシンを選ぶのがいいでしょう。というのは、もし電源が入っていたら、それは誰かが既に使っているものかも知れないからです。もしディスプレイ（テレビの様な画面）の右下のスイッチが「○」の方に押し込まれていたら、まずこれを「|」の方に押し込んでください。電源が入っていないマシンはスイッチの「|」の方が押し込まれているのに、その横の緑色のランプが点灯していないことで識別できます。緑色のランプがどのくらい明るく点灯するか判らないので、点灯しているのかいないのか区別が付かないと言う人は入り口に最も近いマシンのランプを見てください。このマシン (csosf01) は常時電源が入っていますので、比較になるでしょう。

ランプを確認せずに、画面が真っ暗だからと言ってそのマシンの電源が切れているかどうかは判りません。DEC-3300 には、しばらく誰もキーボードを触らなかったら利用途中でも画面を真っ暗にしてしまう機能があるからです。念のためにキーボードの空白キーを一度押して、それでも画面に何の変化もないこ

とを確認しましょう。電源が入っていないマシンが見つかったら、そのマシンの本体（ディスプレイの下敷になっている厚さ 10 センチ位の箱）の、正面に向かって左奥（背面）にある電源スイッチを入れてください。このスイッチは古典的な家庭の壁にある電灯のスイッチに似ていて、シーソーのようにパッチンと倒して入り切りします。

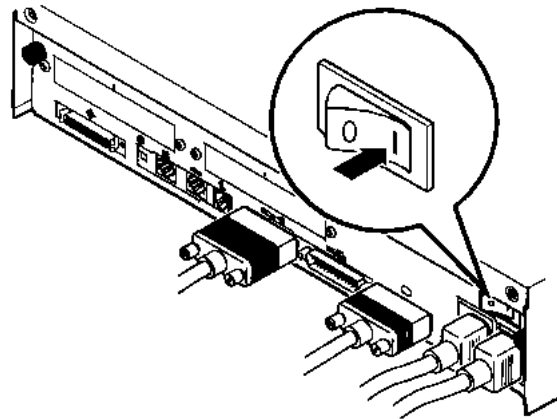


図 3.2 DEC-3300 本体背面：電源スイッチを入れる

電源が入ったらファンが回る音がして、十数秒後に「ピーポーパーポー」と³音がします。それから画面に色々な文字が流れていって、合計 3 分足らずで利用可能な状態になります。

3.2.3 login する

画面中央には以下のウィンドウが表示されていると思います。

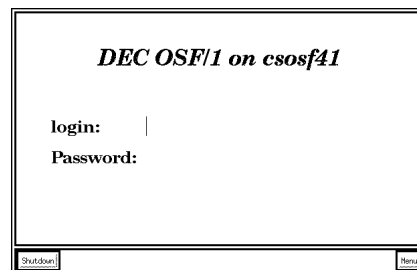


図 3.3 login ウィンドウ

キーボードを使ってまずあなたのユーザ名をタイプします。ユーザ名をタイプし終り、リターンキーを一度押せば今度はパスワードを入力できる状態になります。ここでパスワードを入力するのですが、パスワードは横から見ている他の人にバレないように、タイプしてもその文字が画面に表示されません。キーを打ち間違えないように注意して入力し、もう一度リターンキーを押します。うまくユーザ名とパスワードを入力できれば画面表示が変わります。キーを打ち間違えたりして login に失敗するとウィンドウの下の部分に赤い文字で「Login incorrect」と表示されます。この場合はもう一度ユーザ名のタイプからやり直します。

³ ちょっと前の映画の「未知との遭遇」のテーマみたい？

もしこれがあなたにとって Sign 登録後初めての login であればパスワードは Sign 登録時にあなたがメモしたものの筈です。もし何度試しても login 出来ないようであればユーザ名、パスワードのいずれかが間違っているか、まだあなたのユーザ登録作業が計算機センター側で完了していないのです。計算機センターに自分のユーザ名が既に登録されているかどうか確認してください。もしもユーザ名も正しく、また登録作業も完了しているなら、これはパスワードが違っているとしか考えられません。計算機センター相談窓口で連絡してパスワードを強制変更して貰ってください。

3.2.4 ちょっとメッセージ

login すると、ときには利用者宛の通知、おしらせが以下のように画面に表示されるかも知れません。

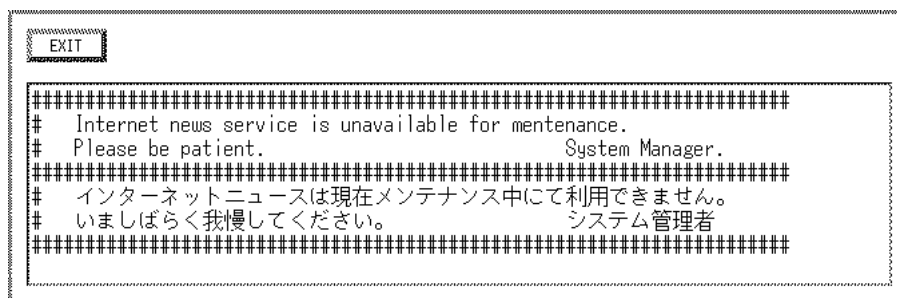


図 3.4 login 直後のメッセージ

よく読んで、自分に関係があるかどうか確認してください。確認が済めば、リターンキーを一度押すだけでこの表示を消すことが出来ます。もしも内容に付いてよく判らないことがあれば計算機センター相談窓口まで連絡するのがいいでしょう。

3.2.5 ターミナルはあるかな？

login に成功すると以下のような画面表示になっていると思います。

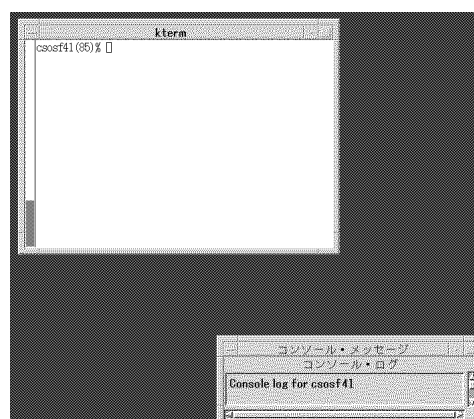


図 3.5 X の初期画面

画面の中に「kterm」というタイトルのついた以下の様なウィンドウがあるかどうかさがしてください。そのようなウィンドウが見つければ、ひとつ飛ばして「ターミナルは大丈夫かな？」まで進んで下さい。もしも見つからなければこのまま次に進んで下さい。

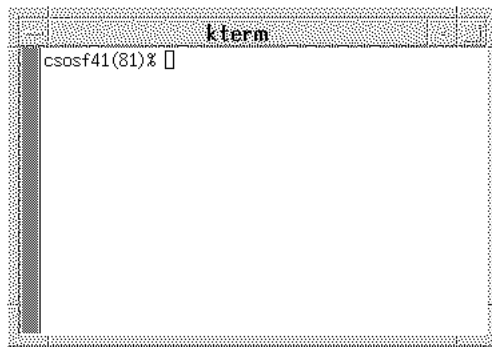


図 3.6 kterm

ターミナルを起動する

画面の背景、特にウィンドウの無い場所にマウスカーソルを持って行って、マウスの左ボタンを押せばなしにします。すると以下のようなメニューが現れるでしょう。この一連の操作を「ルートメニューを表示させる」と呼びます。

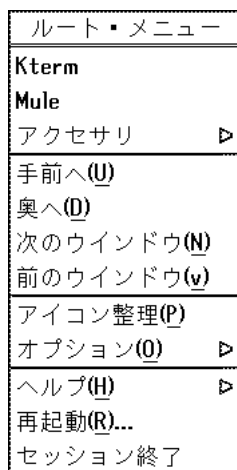


図 3.7 ルートメニュー

マウスのボタンを押したままにして、マウスカーソルを「Kterm」と書いてある行の上に持っていきます。「Kterm」の行の上まで来たらその場所でマウスボタンを離します。これでKtermが起動されるはずです。

この一連の操作を「メニューから「Kterm」を選択する」と呼びます。このメニューでは「Kterm」が選択されると Kterm が起動するようになっています。

ターミナルは大丈夫かな？

ターミナルのウィンドウの左隅には `csosf01(81)%` などと表示されていると思います。(但し数字は違うかもしれません。) ここで試しに `date` とタイプしてリターンキーを押してみてください。文字をタイプしても `csosf01(81)%` に続いて表示されない場合は、ターミナルウィンドウがアクティブになっていません。一度 `csosf01(81)%` 辺りをクリックしてターミナルウィンドウをアクティブウィンドウした後、`date`

とタイプし直してください。以下のように今日の日付と時間が表示されれば大丈夫、ターミナルを使える状態になっています。

```
csosf01(81)% date
1994年02月27日(日)20時05分03秒
csosf01(82)%
```

3.2.6 パスワードを変える

もしもこれがあなたにとって Sign 登録後初めての login ならば、ここでまずパスワードを変更しましょう。Sign 登録時のパスワードは機械的に決められたもので、余り安全とは言えません。必ずパスワードをあなた自身が選んだ単語に変更して下さい。パスワードを決める際には 4 ページの 2.1.3 で既に述べた事に注意してください。

パスワードを変更するには `yppasswd` コマンド⁴を利用します。上の節で説明したターミナルを見ると、

```
csosf01(81)%
```

などとなっていますね。(但し数字は 01 と 81 ではないかもしれません。)ここで `yppasswd` とタイプし、リターンキーを押します。すると以下のような状態に成りますね。(以下はユーザ名 `yasuda` の例。`yasuda` の部分にはあなたのユーザ名が表示されているはずです。)

```
csosf01(81)% yppasswd
Changing NIS password for yasuda
Old NIS password:
```

指示通りここでは古い、つまり今先ほどまで使っていたパスワードをタイプし、リターンキーを押します。ここでは login の時と同じくパスワードをタイプしている間、横の人の覗き見によってあなたのパスワードがバレないように、タイプした文字は画面に表示されない事に注意してください。表示されなくともタイプした文字はちゃんと入力されていますから、安心して確実に一文字ずつタイプして最後にリターンキーをタイプして下さい。すると今度は以下のようなプロンプトを表示します。

```
New password:
```

では指示通り新しい、つまり次からこれにしようと言うパスワードをタイプし、リターンキーを押してください。ここでもタイプした文字は表示されません。今度は以下のようなプロンプトを表示します。

```
Retype new password:
```

これはいましがたタイプした新しいパスワードが打ち間違いで無いことを確認するためのものです。もう一度新しいパスワードをタイプしてリターンキーを押してください。今度は以下のような状態になります。

```
NIS passwd changed on ccnic
csosf01(82)%
```

この `NIS passwd changed` が表示されれば成功です。もしもパスワードが短すぎたりタイプミスがあったりしたらなんらかのエラーメッセージが表示されます。途中でおかしくなったと思う場合は `C-c` (つまりコントロールキーを押しながら C キーを押す) で `csosf(82)%` を表示させます。それからもう一度 `yppasswd` コマンドをやり直します。

⁴綴りがちょっと変ですが、間違っではありません。

3.2.7 logout する（セッションを終了する）

他にやりたいことも色々あるでしょうが今回は最初の利用でしょうし、まず一通りの操作をやってみると言う意味で、logout をこの場でやってみることを勧めます。

画面の背景、特にウィンドウの無い場所にマウスカーソルを持って行って、マウスの左ボタンを押せばなしにします。すると以下のようなメニューが現れるでしょう。この一連の操作を「ルートメニューを表示させる」と呼びます。



図 3.8 ルートメニュー

マウスのボタンを押したままにして、マウスカーソルを恐らくは一番下にある「セッション終了」と書いてある行の上に持っていきます。「セッション終了」の行の上まで来たらその場所でマウスボタンを離します。この一連の操作を「メニューから「セッション終了」を選択する」と呼びます。このメニューでは「セッション終了」が選択されると logout（セッションの終了）作業が行なわれるようになっています。

すると念のために以下のような問い合わせをしてくれます。

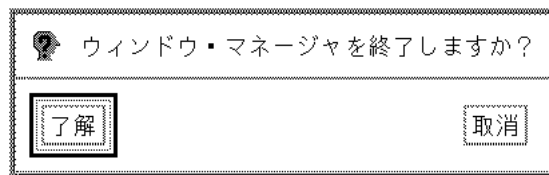


図 3.9 セッション終了の問い合わせ

この問い合わせに対してはマウスのポインタを「了解」ボタンの上に持って行って、マウスの左ボタンでクリックすることで答えます。

これが DEC-3300 での logout 操作です。実際には logout という名前のコマンドも操作も実行しないことに注意してください。DEC-3300 ではつまり logout とはセッションの終了だということです。セッション終了の操作を行なって数十秒すると、最初に電源を入れて login 操作をする直前の状態に戻って図 3.3 が表示されます。もしもその状態にならない、つまり正しくセッションが終了できなければ、計算機センター相談窓口まで連絡下さい。そのまま放置するのはいけません。

3.2.8 シャットダウンする

セッションが終了できれば、最初に電源を入れて login する直前の状態に戻ります。

※ 注意 ※

本体とキーボードの間に「このマシンの本体電源は切らないでください」というようなカードが置いてあった場合は、この節と次の節は読み飛ばして 3.2.10 まで進んでください。恐らくそのマシンは誰かほかの人が夜間利用するつもりなのです。

今度は図 3.3の左下隅のボタンに注目して下さい。

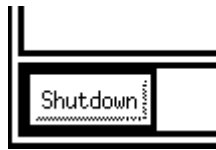


図 3.10 Shutdown ボタン

ここで今度はこのボタンをクリックして下さい。すると数十秒で画面が暗くなって、画面左側に >>> が表示されます。この操作をシャットダウンと呼んでいます。

>>> 表示がどうしてもでない場合は何かトラブルがあったと思われます。決して >>> が表示されていない状態で電源スイッチを切らないで下さい。また、おかしなまま放置するのもいけません。計算機センター相談窓口まで連絡下さい。また、部屋の入り口に最も近い場所にある csosf01 と、csosf01 正面に向かって右隣にある csosf02 の二台については電源は常時入りっぱなしです。このマシンはシャットダウンできませんし、また電源も切らないでください。

3.2.9 電源を切る

>>> 表示が出たら、本体（ディスプレイの下敷になっている厚さ 10 センチ位の箱）の、正面に向かって左奥（背面）にある電源スイッチを切ります。最初に入れたときは逆の方向に倒すわけですね。これで電源が切れます。このときディスプレイ右下のスイッチは操作しないでください。

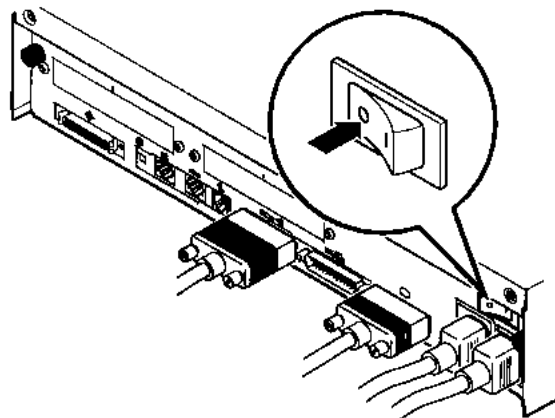


図 3.11 DEC-3300 本体背面：電源スイッチを切る

3.2.10 部屋を出る

マシンの電源が切れたら部屋を出ます。このとき、部屋に誰も残らないようなら空調機、照明を切って行きましょう。

3.2.11 さて、さて、

最初の利用はこれでおしまいです。大変よくできました。あとは精進して UNIX ユーザへの道を歩まれるわけですね。それには UNIX での一般的なコマンド操作などについて説明している、第4章へと進んで下さい。

3.2.12 マニュアルなど

DEC-3300 のマニュアルは部屋の入り口から最も離れた奥のロッカーに数冊置いてあります。但し一部ずつしかないので皆で同時に見ることは出来ません。うまく共有してください。

3.3 Macintosh の telnet で cc2000 を使う

ここでは1号館2階の11情報処理教室と3号館2階の31情報処理教室及び図書館1階のパソコンコーナーのMacintoshからtelnet環境でcc2000をホストコンピュータとして利用する手続きについて説明します。

MacintoshはApple Computer社製のパソコンで、1号館2階の11情報処理教室にPowerMacintosh 6100/66という機種が90台、3号館2階の31情報処理教室にPowerMacintosh 6100/60AVという機種が44台、図書館1階のパソコンコーナーにPowerMacintosh 6100/66という機種が12台、設置してあります。

これらの部屋Macintoshでは端末ソフトとして「ASLTelnet」を利用します。このソフトはMacintoshパソコンでtelnet環境を提供します。

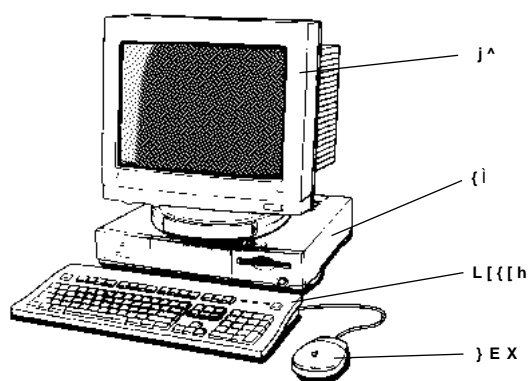


図 3.12 Macintosh(マッキントッシュ)外観

3.3.1 まず教室にはいる

11 情報処理教室にはいる

入室可能な時間帯、曜日、期間に限りがありますので、付録の「情報処理教室利用要項」を一読してください。

部屋に入るときは、そこで授業をやっていないことを確認しましょう。もし授業中であれば入室して利用してよいか指導教員に確認を取るのが礼儀というものです。しかし、授業中で入口に「受講生以外入室禁止」などのカードが掛けてあったら、その時間はあきらめて、入口近くの掲示板で「情報処理教室利用時間割表」を調べ、使えそうなときにもう一度来てください。

この教室では90名の授業が行えますが、受講人数が少ない場合、カーテンで区切り教室の右側だけで授業が行われています。もし廊下から見て教室がカーテンで区切られていれば教室の左半分を自由に使うことができます。3つある扉の一番左側から入って使ってください。

部屋に誰も居なかった場合は照明、空調機が切れている場合があります。照明スイッチは中央のドアから部屋に入ったそのすぐ左手の壁に集中しています。空調機のスイッチもそのあたりに集中していますので、自分で適当に操作してください。

31 情報処理教室にはいる

3号館2階の31情報処理教室は常時カードロックシステムが働いており、学生証を入り口のドア横の機械に通すことによって鍵が開きます。入室可能な時間帯、曜日、期間に限りがありますので、付録の「情

報処理教室利用要項」を一読してください。もしもカードを通して鍵が開かない場合は、カードの向き、表裏、カードを通過させる速度（遅すぎても速すぎても駄目）を変えてやってください。どうしても駄目な場合はカードの磁気情報がおかしくなっている可能性がありますので、計算機センターまで連絡してください。

部屋に入るときは、そこで授業をやっていないことを確認しましょう。もし授業中であれば入室して利用していいかどうか指導教員に確認を取るのが礼儀と言うものです。しかし、授業中で入口に「受講生以外入室禁止」などのカードが掛けてあったら、その時間はあきらめて、入口近くの掲示板で「情報処理教室利用時間割表」を調べ、使えそうなときにもう一度来てください。

部屋に誰も居なかった場合は照明、空調機が切れている場合があります。照明スイッチは部屋に入ったそのすぐ左手の壁に集中しています。空調機のスイッチは正面のホワイトボードの左隅（窓の近く）のロッカーの奥と、教室の廊下側の壁面に2箇所ありますので、自分で適当に操作してください。

部屋の管理そのものは外国語学部事務室が行っていますので、何か部屋の状態について質問、要望があれば外国語学部事務室までお願いします。コンピュータについての質問、要望は計算機センターの相談窓口までお願いします。

図書館パソコンコーナーにはいる

この部屋は中央図書館の1階読書室内の一角に設けてあり、図書館が開館中であれば誰でも利用が可能です。部屋の管理は図書館が行っていますので、何か椅子、机など部屋の状態及びコンピュータについての質問、要望などがあれば図書館カウンタにお願いします。

3.3.2 電源を入れる

11,31 情報処理教室の Macintosh と図書館パソコンコーナーの Macintosh とでは電源の入れ方に違いがあります。以下の記述は情報処理教室の為のものです。パソコンコーナーの Macintosh の電源の入れ方については部屋の掲示などを見て下さい。

今から自分が利用するコンピュータを選びます。もしこれが最初の利用であれば、電源の入っていない Macintosh を選ぶのがいいでしょう。というのは、もし電源が入っていたら、それは誰かが既に使っているかも知れないからです。誰かが使っている Macintosh はディスプレイ（テレビの様なもの）の画面が明るく、画面の右下のランプが点灯しているので識別できます。

まだ電源の入っていない Macintosh が見つければ、その Macintosh のキーボードの上方にある三角マークのあるキー（パワーキー）を押して下さい。

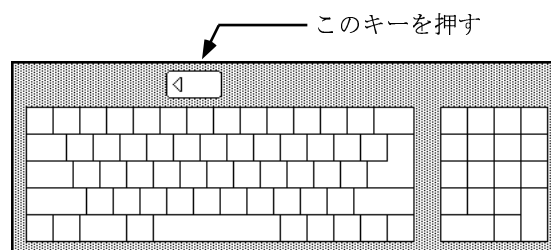


図 3.13 パワーキーを押して電源を入れる

操作するのはこのパワーキーだけです。本体やディスプレイのスイッチを操作する必要はありません、してはいけません。

パワーキーを押せばファンが回る音がして「ポーン」と音がします。それからディスプレイ画面が明るくなり、画面の下に色々なマークが表示されて、2分足らずで利用可能な状態になります。

3.3.3 うまく電源が入らないとき

パワーキーを押して Macintosh の電源を入れるためには、Macintosh 本体のスイッチと、モニタのスイッチが ON の状態になっている必要があります。通常これらのスイッチは常に ON の状態になっている筈ですが、誰かが不用意にこのスイッチを操作してしまい、OFF の状態になっている可能性があります。図 3.13 のパワーキーをいくら押しても電源が入らない場合は、まずこれが原因として考えられます。

Macintosh 本体の左正面に小さな電源ランプがあります。このランプが点灯していると本体の電源が入っています。パワーキーを押してもこのランプが点灯しないときは以下の図のスイッチを押してください。このスイッチはいつも ON の状態にしておいてください。

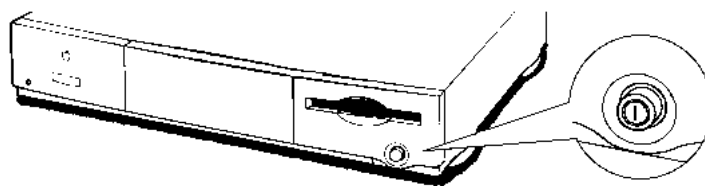


図 3.14 本体のスイッチ

ディスプレイも同じようにパワーキーを押しても電源ランプが点灯していないのであれば、ディスプレイの画面右下にある以下の図のスイッチを押して下さい。このスイッチはいつも ON の状態にしておいてください。

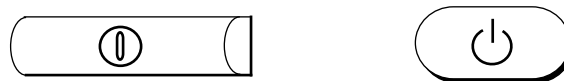


図 3.15 ディスプレイのスイッチ (左 31 情報処理教室 : 右 11 情報処理教室)

3.3.4 Macintosh の操作

せっかく Macintosh の電源が入って使えるようになったのですから、少し Macintosh の操作を学びましょう。マニュアルの置いてある棚から Macintosh User's Guide を持ってきてください。Macintosh のマニュアルは部屋の書架に置いてあります。Macintosh User's Guide の「第 1 章 基本操作の練習」でマウス操作、アイコン、ウインドウ、キーボードの操作を練習してください。

この練習は約 30 分ほどで出来ます。Macintosh の基本的な使い方が全て解るので是非お勧めします。

3.3.5 端末ソフトの起動

まず端末ソフトを起動しましょう。

Macintosh のマウスを使ってアップルメニューから ASLTelnet(cc2000) を選びます。

うまく起動できると、telnet ウインドウが一つ開いて、

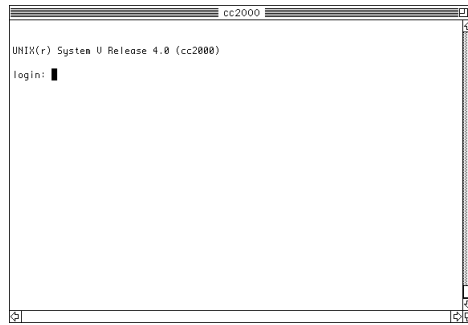


図 3.16 ASLTelnet の login 画面

そこには login: と催促文字が表示されている筈です。

3.3.6 さあ、login しよう

いよいよ login です。ここからはもうあなたのキーボードに対する入力はホストコンピュータに対する遠隔操作となっています。

ユーザ名とパスワードを入力しましょう。

正しくユーザ名をタイプし<Return>キーを押すと、パスワードを聞いて来ます。これも正しくタイプし<Return>キーを押すと login できます。入力するパスワードは横から見ている人にばれないように、タイプしても画面に表示されないの注意して正しく入れてください。

もしパスワードの入力途中で間違ったときは、それまで入力した文字数より少し多めに<Delete>キーを押してもう一度最初からパスワードを正しく入れてみましょう。

ちゃんと login 出来ましたか？

いくらユーザ名とパスワードを入れても login 出来ない方は、ユーザ名とパスワードを間違っているかもしれません。UNIX は大文字と小文字の区別があるので注意してください。

どうしてもうまくいかない場合は 11:31 情報処理教室では計算機センター相談窓口まで、パソコンコーナーでは図書館カウンターまで連絡して下さい。

3.3.7 ちょっとメッセージ

login すると、ときには利用者宛の通知、おしらせが以下のように画面に表示されるかも知れません。

```
Sun Microsystems Inc. SunOS 5.3 Generic September 1993
4月15日深夜、ファイルサーバである ccnfs (cc2000) の
レポートが予定されています。
これにともない cc 環境の全てのマシンの利用が出来なくな
ります。夜間に長時間のジョブを実行する予定のある利用者
はそれまでに実行が終るように注意して下さい。
cc2000(84)% █
```

図 3.17 login 直後のメッセージ

よく読んで、自分に関係があるかどうか確認してください。もしも内容に付いてよく判らないことがあれば計算機センター相談窓口まで連絡するのがいいでしょう。

3.3.8 パスワードを変える

もしもこれがあなたにとって Sign 登録後初めての login ならば、ここでまずパスワードを変更しましょう。Sign 登録時のパスワードは機械的に決められたもので、余り安全とは言えません。必ずパスワードをあなた自身が選んだ単語に変更して下さい。パスワードを決める際には 4 ページの 2.1.3 で既に述べた事に注意してください。

パスワードを変更するには `yppasswd` コマンド⁵を利用します。上の節で説明したターミナルを見ると、
`cc2000(81)%`

などとなっていますね。(但し数字は 81 ではないかもしれません。) ここで `yppasswd` とタイプし、リターンキーを押します。すると以下のような状態に成りますね。(以下はユーザ名 `yasuda` の例。 `yasuda` の部分にはあなたのユーザ名が表示されているはずです。)

```
cc2000(81)% yppasswd
Changing NIS password for yasuda
Old NIS password:
```

指示通りここでは古い、つまり今先ほどまで使っていたパスワードをタイプし、リターンキーを押します。ここでは login の時と同じくパスワードをタイプしている間、横の人の覗き見によってあなたのパスワードがバレないように、タイプした文字は画面に表示されない事に注意してください。表示されなくともタイプした文字はちゃんと入力されていますから、安心して確実に一文字ずつタイプして最後にリターンキーをタイプして下さい。すると今度は以下のようなプロンプトを表示します。

```
New password:
```

では指示通り新しい、つまり次からこれにしようと言うパスワードをタイプし、リターンキーを押してください。ここでもタイプした文字は表示されません。今度は以下のようなプロンプトを表示します。

```
Retype new password:
```

これはいましがたタイプした新しいパスワードが打ち間違いで無いことを確認するためのものです。もう一度新しいパスワードをタイプしてリターンキーを押してください。今度は以下のような状態になります。

```
NIS passwd changed on ccnic
cc2000(82)%
```

この `NIS passwd changed` が表示されれば成功です。もしもパスワードが短すぎたりタイプミスがあったりしたらなんらかのエラーメッセージが表示されます。途中でおかしくなったと思う場合は `C-c` (つまりコントロールキーを押しながら C キーを押す) で `cc2000(82)%` を表示させます。それからもう一度 `yppasswd` コマンドをやり直します。

3.3.9 logout をしよう

次は、logout です。他にやりたいことも色々あるでしょうが今回は最初の利用でしょうし、まず一通りの操作をやってみると言う意味で、logout をこの場でやってみることを勧めます。

画面上に `cc2000(81)%` が表示されていることを確認してください。ただし、括弧の中の数字は気にしないでね!!

ここで、`logout<Return>` と入力するだけで telnet ウィンドウが消え logout されます。

```
cc2000(81)% logout
```

⁵綴りがちょっと変ですが、間違っではありません。

3.3.10 端末ソフトを終了する

logout するといきなり telnet ウィンドウが消えます。少し驚くかも知れませんが、それでいいのです。すでにホストコンピュータとの遠隔操作を終わっているのです。

さて、端末ソフトを終了するにはマウスで「File」メニューの「Quit」を選択します。これで ASLTelnet は終了します。

3.3.11 電源を切る

さて、Macintosh に慣れて使えるようになったのですが、ここで Macintosh の電源を切る操作をしてみましょう。

11,31 情報処理教室の Macintosh と図書館パソコンコーナーの Macintosh とでは電源の切り方に違いがあります。以下の記述は情報処理教室の為のものです。パソコンコーナーの Macintosh の電源の切り方については部屋の掲示などを見て下さい。

普通、パソコンの電源を切るときはいま使っていた機能を終了する必要があります。ワープロならワープロソフトを、通信なら通信ソフトを終了してから行いますが、ここでは今まで使っていた ASLTelnet を既に終了しているので大丈夫です。さあ Macintosh の電源を切ってみましょう。

「特別」メニューより「システム終了」を選びます。選ぶ操作は先ほど Macintosh User's Guide の「第 2 章 デスクトップについて理解する」で練習しましたが、少しおさらいをしながらやってみます。まず、マウスカーソルを「特別」メニューに移動させマウスボタンを押したままにします。そうすると、メニューの内容が下方方向にプルダウン表示されます。マウスボタンを押したままマウスカーソルを「システム終了」までドラッグして「システム終了」が黒くなってからマウスボタンを放します。

この操作で Macintosh の本体からチリチリと音がしてやがて、ディスプレイと Macintosh 本体の電源が切れます。

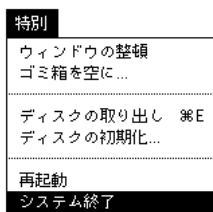


図 3.18 特別メニューからシステム終了を選ぶ

3.3.12 部屋を出る

Macintosh の電源が切れたら部屋を出ます。11,31 情報処理教室では、部屋に誰も残らないようなら空調機、照明を切って行きましょう。

最初の利用はこれでおしまいです。大変よくできました。あとは精進して UNIX ユーザへの道を歩まれるわけですね。それには UNIX での一般的なコマンド操作などについて説明している、第 4 章へと進んで下さい。

また、これからもマニュアルを読みながら立派な「Macintosh 使い」になって、できれば初心者に親切に教えてあげてください。

3.3.13 困った時の対処

文字をタイプしても表示されない

目的の telnet ウィンドウがアクティブになっていないことが考えられます。アクティブになっているかいないかはウィンドウのタイトルバーに横線が現れているかいないかで知ることが出来ます。

入力したいウィンドウの一部をマウスボタンでクリックするとそのウィンドウがアクティブになります。

文字をタイプすると日本語が出てしまう

キー入力が日本語モードになっていることが考えられます。

日本語入力モードをやめて英語入力モードにするには command キーを押しながら Space バー (空白キー) を押します。メニューバーの右から二つ目の小さなアイコンが「米国旗」になれば英語入力モードになっています。

telnet 画面が表示されない

最初にアップルメニューから ASLTelnet(cc2000) を選択した時に telnet 画面が現れないことは、あなたの操作が正しく、かつ cc2000 が稼働しておれば考えられないことです。

このような場合は近くで使っている人が cc2000 をうまく利用できているかどうか調べましょう。と言うのもホストコンピュータである cc2000 そのものが何らかの事情で止まっているとか、ネットワークが不通になっていることが考えられるからです。もし近くの人も cc2000 との接続がうまく行っていないようなら、計算機センター相談窓口に電話をしてください。

日本語が壊れたようになって正しく表示されない

これもアップルメニューから ASLTelnet(cc2000) を選択して使っているときは起こらない現象です。一旦 logout して、アップルメニューから ASLTelnet(cc2000) を選択する操作からやり直して下さい。

3.3.14 cc2000 以外のホスト コンピュータを使う時の注意

ASLTelnet を使って cc2000 以外のホストコンピュータに login する事も出来ます。ASLTelnet アイコンを直接ダブルクリックして ASLTelnet を起動するか、何とかして ASLTelnet が起動されている状態で、File メニューの Open Connection... を選択する事によってホストコンピュータを指定する為のウィンドウが表示されます。ここでホスト名をタイプして<Return>キーを押せば指定したホストコンピュータへの login が可能になるでしょう。

ところでアップルメニューの ASLTelnet(cc2000) を選択して使っている時は計算機センターが用意した各種の設定が効いているので正しく動作していますが、Open Connection... でホストコンピュータを指定した場合はそれらの設定が及んでいませんので幾らかの動作でおかしくなる可能性があります。以下に典型的な症状と原因、対処法を示しておきます。

telnet ウィンドウが表示されない

Open Connection... ウィンドウでタイプしたホストコンピュータの名前を (例えば cc2000 ならば cc200 など) 間違っている可能性があります。File メニューの Open Connection... を選択する操作からやり直して下さい。

ホスト名を間違えていない場合には、目的のコンピュータが何らかの事情で止まっているとか、ネットワークが不通になっている可能性があります。目的のコンピュータの管理者などに確認してみてください。

漢字の部分が正しく表示されない

ASLTelnet がホストコンピュータに送る日本語コードとホストコンピュータから送られてくる漢字コードがうまく合っていないときにこのような状況になります。(この表現の意味が判らない場合は cc2000 以外のホストコンピュータを利用しない方がいいでしょう。)

「Session」メニューに「Transmit Code」と「Receive Code」があります。自分がおもうと思っているホストコンピュータの漢字コードが何かを確認して、これを正しい値(例えば cc2000 ならそれぞれ「EUC」と「EUC & JIS7」)にしてください。

これで日本語が正しく表示されます。

キーボードの最適化

Open Connection... でホスト名を指定して開いた telnet ウィンドウでは C-c や C-s や C-q が入力できません。これでは Mule を使う時に困った事になりますので以下の方法で設定してください。

メニューバーの「Session」から「Setup Keys...」を選択します、すると次のウィンドウが表示されます。

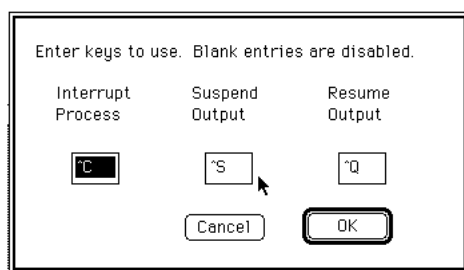


図 3.19 キーのセットアップ

Interrupt process、Suspend Output、Resume Output に指定されている文字を delete キーで全て消します。そして OK を選択します。(単純には<Delete><Tab><Delete><Tab><Delete><Return>とキーをタイプすれば良いでしょう。) これで C-c や C-s や C-q が入力できるようになったはずですが。

3.3.15 マニュアルなど

マニュアルは情報処理教室やパソコンコーナーのロッカーか書架に置いてあります。使っている Macintosh のそばに持ってきて利用して構いませんが、終われば元に戻しておいてください。

3.4 FMR から telnet で cc2000 を使う

FMR-280 は五号館 1 階の 51 情報処理教室に設置してあります。

ここでは 5 号館 1 階の 51 情報処理教室の FMR から telnet 環境で cc2000 をホストコンピュータとして利用する手続きについて説明します。

51 情報処理教室の FMR では端末ソフトとしてアライドテレシス株式会社の「WVTN」を利用します。これは telnet 環境を提供します。

FMR のキーボードでは、一般的にリターンキーと呼ばれるキー⁶には、折れ曲がった矢印記号 (↵) が刻印されています。ここではこのキーを「リターンキー」と表現します。例示に<Return>と書いてある場合は FMR では、この折れ曲がった矢印記号 (↵) キーを押して下さい。

3.4.1 まず部屋へ

FMR-280 は五号館 1 階の 51 情報処理教室に設置してあります。

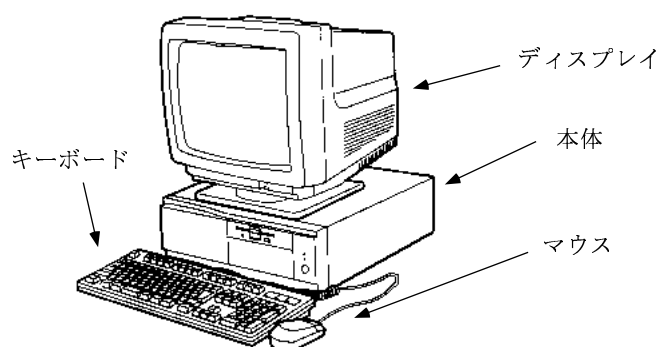


図 3.20 FMR-280 外観

この部屋は常時カードロックシステムが働いており、学生証を入り口のドア横の機械に通すことによって鍵が開きます。入室可能な時間帯、曜日、期間に限りがありますので、付録の「情報処理教室利用要項」を一読してください。もしもカードを通して鍵が開かない場合は、カードの向き、表裏、カードを通過させる速度（遅すぎても速すぎても駄目）を変えてやってください。どうしても駄目な場合はカードの磁気情報がおかしくなっている可能性があります。カードを通す機械にあなたのカードでなぜ鍵が開かないか、その理由が出ていますので、それを控えて計算機センターまで連絡してください。部屋に入るときは、そこで授業をやっていないことを確認しましょう。もし授業中であれば入室して利用していいかどうか指導教員に確認を取るのが礼儀と言うものです。部屋に誰も居なかった場合は照明、空調機が切れている場合があります。照明のスイッチは部屋に入ったすぐ左手の壁、空調機のスイッチは部屋の後部に設置された空調機の前面パネルを開いた所にありますので、自分で適当に操作してください。部屋の管理そのものは経済学部事務室が行っていますので、何か部屋の状態について質問、要望があれば経済学部事務室までお願いします。コンピュータについての質問、要望は計算機センターの相談窓口までお願いします。

3.4.2 電源を入れる

自分が利用するコンピュータを選びます。この場合、電源の入っていないマシンを選ぶのがいいでしょう。というのは、もし電源が入っていたら、それは誰かが既に使っているものかも知れないからです。電源

⁶ キーボード真中やや右の大きなキー

が入っていないマシンは、ディスプレイ（テレビの様な画面）の右下の POWER と書かれた緑色のランプが点灯していないことで識別できます。同様のランプは本体（ディスプレイの下敷きになっている厚さ 15 センチ位の箱）の右下にもあります。ランプを確認せずに、画面が真っ暗だからと言ってそのマシンの電源が切れているかどうかは判りません。FMR-280 には、しばらく誰もキーボードを触らなかつたら利用途中でも画面を真っ暗にってしまう状態もあるからです。電源が入っていないマシンが見つかったら、先ほどのランプのそば（ディスプレイの右下か、本体の右下）にあるスイッチを押してください。このスイッチは押しても形状は変化せず、緑色のランプが点灯することによって電源が入ったことを確認できます。

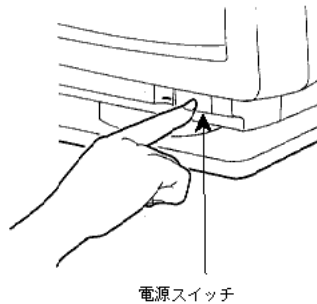


図 3.21 FMR-280 本体前面：電源スイッチを入れる

電源が入ったら画面に色々な文字が流れていって、画面上に「京都産業大学」と書かれたメニュー（初期メニュー）が表示されると利用可能な状態になります。

3.4.3 Windows の起動

まず Windows を起動しましょう。

パソコン本体またはディスプレイ装置のどちらかの電源を入れれば、自動的に図 3.22 のメニューが現れます。キーボード右下にある矢印（←→↑↓）を操作して最上部の「MS-Windows」の色を反転させて下さい。

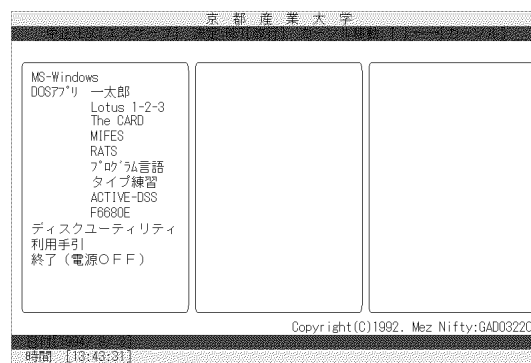


図 3.22 FMR の初期メニュー

ここでリターンキーを押せばマイクロソフト社の Windows が起動します。

3.4.4 Windowsの動かし方

Windowsが立ち上がった状態（画面上に絵が表示された状態）になっています。Windowsは、色々な機能を満載してしながら、基本的な操作はほとんど共通に利用出来ます。焦らずゆっくりとマニュアルを読んで操作すれば大丈夫です。

C1情報処理教室、51情報処理教室には、初心者向けに「Quick ユーザーズガイド」が設置してあります。「Quick ユーザーズガイド」のポイント1からポイント10まで、読まれることをお奨めします。52情報処理教室には、「今日から使えるWindows3.1」と「機能ガイド」があります。「今日から使えるWindows3.1」は初心者向けに書かれてありますので目を通すことをお奨めします。「機能ガイド」はWindowsの詳細な機能が書かれてあります。

マニュアルの設置場所は、「マニュアルなど」の説明に書かれてありますので、そちらをお読みください。

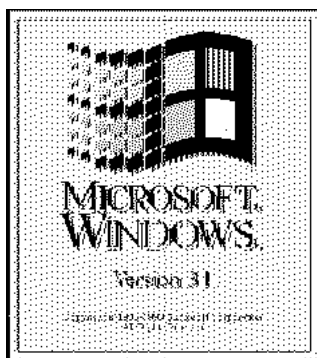


図 3.23 Windows 起動画面

3.4.5 Windowsに関する基礎知識（絶対にやってはいけない事）

いきなり電源オフ

パソコンの操作方法がわからなくなってしまった時、ほとんどの方がとる手段が、電源を切ってしまうことです。この手段は絶対にしてはいけません。急に電源を切ってしまうと、パソコンの中の情報が破壊されて、次に電源をいれても正常に動作しません。あなたがパソコンの操作に迷った時は、取りあえず周りの人に聞くとよいでしょう。周りに人がいないのなら、計算機センターまで隣の部屋のコードレスホンで電話をしてください。それでほとんどが解決するはずです。いきなり電源を切らないでください。

書き込み中のフロッピー取り出し

フロッピーが動作しているかどうか（書き込み中であるかないか）を知るには、本体のやや真ん中右よりのフロッピー差し込み口を見てください。グリーンランプが点灯していれば、まさに書き込み中です。（52情報処理教室のConturaではランプは電源ボタンの上に並ぶランプの内のディスク・ドライブ・ランプです。図3.36を参照して下さい。因みにフロッピー差し込み口は本体右側面にあります。）このランプが点灯しているときに、フロッピーを取り出すと中身が壊れます。あなたが作った物すべてが読めなくなるかもしれません。フロッピーを取り出す時には十分注意しましょう。

むやみに削除を選択すること

Windows は自分が作成した文章、プログラムおよびパソコンを起動させるプログラムを簡単に表示、削除、移動、複写が行えます。あなたが作成した文章やプログラムを操作する分については問題がないのですが、パソコンにとって重要なプログラムを削除すると、特定の機能が使えなくなるか、パソコンが動かなくなります。もしわからなくなった場合は、周りの人に尋ねるか、計算機センター相談窓口まで問い合わせてください。

むやみに設定を変更すること

Windows には、色々な機能を満載していますが、その一つ一つは細々した設定がなされています。パソコンは不特定多数の利用者によって使われるますので、私たち計算機センターは可能な限りいつでも同じ環境を提供したいと思っています。設定が気に入らない場合もあるでしょうが、そのままの状態使ってください。もちろん「このようにした方がもっといいと思いますよ」といった提案はいつでも計算機センター相談窓口までお願いします。

3.4.6 端末ソフトの起動

端末ソフトを起動しましょう。画面には図 3.24 が表示されていますね。

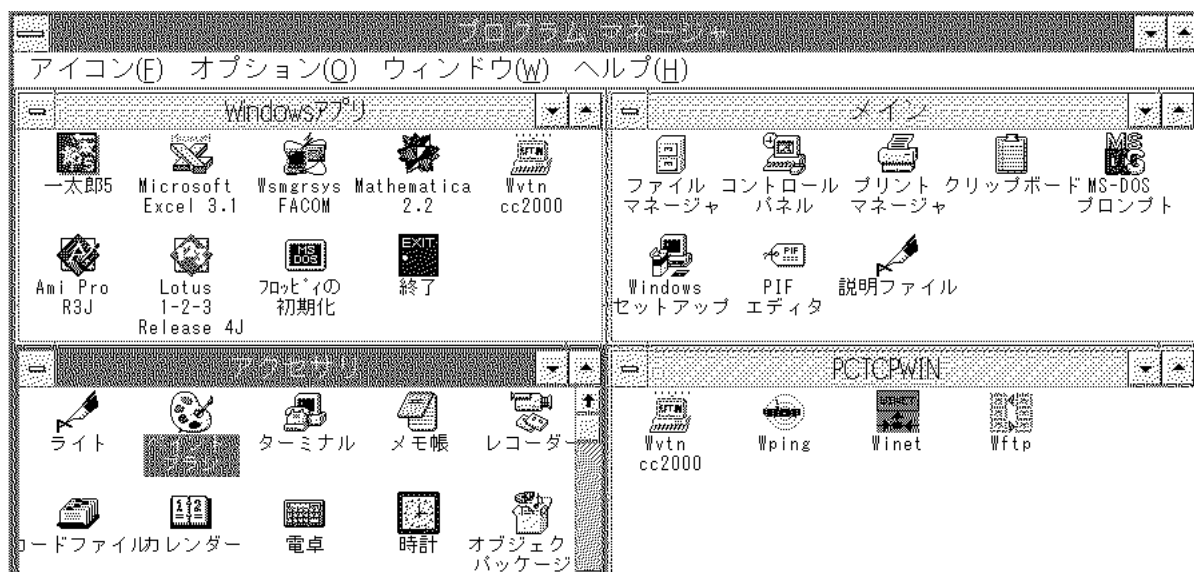


図 3.24 Windows 初期メニュー

図 3.24 の中の一つに「WVTN cc2000」図 3.25 があります。このアイコン（絵）にマウスポインタ（矢印）を重ね合わせて、マウスの左ボタンをダブルクリック（連続 2 回押し）してください。WVTN が起動されます。



図 3.25 WVTN のアイコン

3.4.7 ホストコンピュータへの接続

この端末ソフトは、起動すると自動的に cc2000 に接続されます。

WVTN が起動されると、図 3.26 が画面に表示されます。もし画面に現れなかったら、あなたはソフトウェアの起動に失敗しています。もう一度図 3.25 にマウスポインタ（矢印）を重ね合わせてマウスの左ボタンをダブルクリックしてください。それでも図 3.26 が現れない場合は計算機センター相談窓口まで連絡してください。

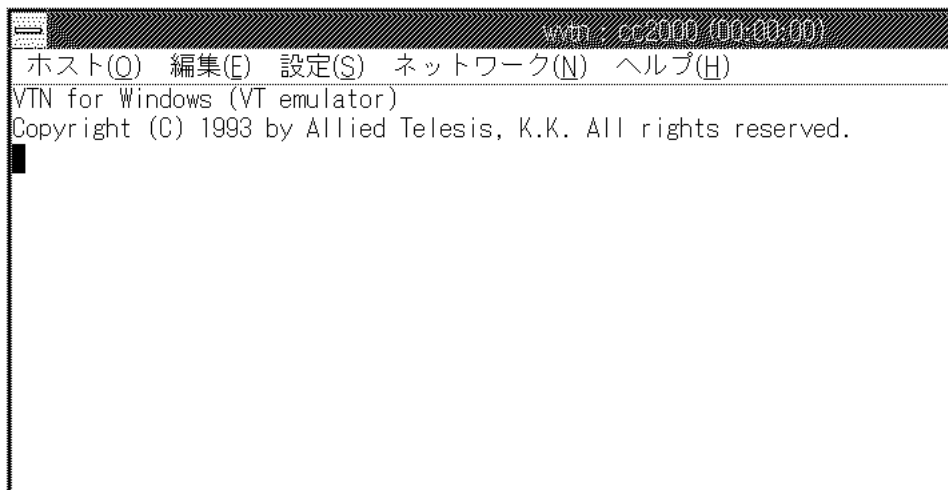


図 3.26 WVTN の初期画面

画面上には、図 3.26 が表示されています。この図の左上に「ホスト (O)」があります。マウスポインタ（矢印）を「ホスト (O)」に重ね合わせて、マウスの左ボタンをクリックするとメニューが表示されます。このメニューの「接続」にマウスポインタを重ね合わせ、マウスの左ボタンをクリックしてください。図 3.27 のようになります。この図のようにならない場合は、WVTN の初期設定が変更されている可能性があります。この後の「困った時の対処」を参照してください。

```

      _____
     |                                     |
     |                               www: cc2000 (00:00:01) |
     |-----|
     | ホスト(O) 編集(E) 設定(S) ネットワーク(N) ヘルプ(H) |
     |-----|
     | VTN for Windows (VT emulator) |
     | Copyright (C) 1993 by Allied Telesis, K.K. All rights reserved. |
     |                                     |
     | UNIX(r) System V Release 4.0 (cc2000) |
     |                                     |
     | login: |
     |_____

```

図 3.27 cc2000 のプロンプト

ここで `login:` などと `login` の為の催促文字が表示されていることを確認して下さい。

3.4.8 さあ、login しよう

いよいよ `login` です。ここからはもうあなたのキーボードに対する入力はホストコンピュータに対する遠隔操作となっているのです。

ユーザ名とパスワードを入力しましょう。`login:`の後にあなたのユーザ名を入力して下さい。ユーザ名には大文字、小文字の区別があるので注意して下さい。ユーザ名の入力が完了したら、一度リターンキーを押して下さい。次にパスワードを入力します。ユーザ名を入力した時とは違い、パスワードは横から見ている他の人にバレないように、タイプしてもその文字が画面に表示されません。キーを打ち間違えないように注意して入力し、もう一度リターンキーを押して下さい。

```

      _____
     |                                     |
     |                               www: cc2000 (00:00:05) |
     |-----|
     | ホスト(O) 編集(E) 設定(S) ネットワーク(N) ヘルプ(H) |
     |-----|
     | VTN for Windows (VT emulator) |
     | Copyright (C) 1993 by Allied Telesis, K.K. All rights reserved. |
     |                                     |
     | UNIX(r) System V Release 4.0 (cc2000) |
     |                                     |
     | login: sheryl |
     | Password: |
     | Last login: Wed Aug  3 16:24:03 from e51nf30.kyoto-su |
     | Sun Microsystems Inc.  SunOS 5.3  Generic September 1993 |
     |                                     |
     | cc2000(81)% |
     |_____

```

図 3.28 cc2000 のログインの SAMPLE

図 3.28 のようになりましたか？もし、ならなかった場合は、この後の「困ったときの対処」を参照してください。

3.4.9 ちょっとメッセージ

login すると、ときには利用者宛の通知、おしらせが以下のように画面に表示されるかも知れません。

```
Sun Microsystems Inc. SunOS 5.3 Generic September 1993
4月15日深夜、ファイルサーバである ccnfs (cc2000) の
レポートが予定されています。
これにともない cc 環境の全てのマシンの利用が出来なくな
ります。夜間に長時間のジョブを実行する予定のある利用者
はそれまでに実行が終るように注意して下さい。
cc2000(84)% □
```

図 3.29 login 直後のメッセージ

よく読んで、自分に関係があるかどうか確認してください。もしも内容に付いてよく判らないことがあれば計算機センター相談窓口まで連絡するのがいいでしょう。

3.4.10 パスワードを変える

もしもこれがあなたに取って Sign 登録後初めての login ならば、ここでまずパスワードを変更しましょう。Sign 登録時のパスワードは機械的に決められたもので、余り安全とは言えません。必ずパスワードをあなた自身が選んだ単語に変更して下さい。パスワードを決める際には 4 ページの 2.1.3 で既に述べた事に注意してください。

パスワードを変更するには `yppasswd` コマンド⁷を利用します。上の節で説明したターミナルを見ると、

```
cc2000(81)%
```

などとなっていますね。(但し数字は 81 ではないかもしれません。) ここで `yppasswd` とタイプし、リターンキーを押します。すると以下のような状態に成りますね。(以下はユーザ名 `yasuda` の例。 `yasuda` の部分にはあなたのユーザ名が表示されているはずです。)

```
cc2000(81)% yppasswd
Changing NIS password for yasuda
Old NIS password:
```

指示通りここでは古い、つまり今先ほどまで使っていたパスワードをタイプし、リターンキーを押します。ここでは login の時と同じくパスワードをタイプしている間、横の人の覗き見によってあなたのパスワードがバレないように、タイプした文字は画面に表示されない事に注意してください。表示されなくともタイプした文字はちゃんと入力されていますから、安心して確実に一文字ずつタイプして最後にリターンキーをタイプして下さい。すると今度は以下のようなプロンプトを表示します。

```
New password:
```

では指示通り新しい、つまり次からこれにしようと言うパスワードをタイプし、リターンキーを押してください。ここでもタイプした文字は表示されません。今度は以下のようなプロンプトを表示します。

```
Retype new password:
```

これはいましがたタイプした新しいパスワードが打ち間違いで無いことを確認するためのものです。もう一度新しいパスワードをタイプしてリターンキーを押してください。今度は以下のような状態になります。

⁷綴りがちょっと変ですが、間違っではありません。

```
NIS passwd changed on ccnic
cc2000(82)%
```

この NIS passwd changed が表示されれば成功です。もしもパスワードが短すぎたりタイプミスがあったりしたらなんらかのエラーメッセージが表示されます。途中でおかしくなったと思う場合は C-c (つまりコントロールキーを押しながらCキーを押す) で cc2000(82)% を表示させます。それからもう一度 yppasswd コマンドをやり直します。

3.4.11 logout をしよう

次は logout です。他にやりたいこともいろいろあるでしょうが、今回は最初の利用でしょうし、まず一通りの操作をやってみると言う意味で logout をこの場でやってみることを勧めます。画面上に cc2000(81)% が表示されていることを確認してください。但し括弧の中の数字は気にしないでね!! 以下のように、logout <Return> と入力してください。

```
cc2000(81)% logout
```

3.4.12 端末ソフトを終了する

logout すると login する前と同じように、再び図 3.26が表示されます。この図の左上に「ホスト(O)」があります。マウスポインタ(矢印)を「ホスト(O)」に重ね合わせて、マウスの左ボタンをクリックするとメニューが表示されます。このメニューの「WVTNの終了」にマウスポインタを重ね合わせ、マウスの左ボタンをクリックしてください。電源を入れた最初の状態(図 3.24)に戻ります。



図 3.30 WVTN の終了

図 3.26 が画面上から消えましたか? 消えない場合は、マウスポインタが「WVTNの終了」の位置に重なっていない事が予想されます。もう一度マウスポインタを「WVTNの終了」の位置に重ね合わせ、マウスの左ボタンをクリックして下さい。

3.4.13 Windows を終了する

Windows を終了させるには、「Exitwin 終了」という絵(アイコン)をマウスポインタで二度クリックします。「Ms-Windows を終了します」という確認メッセージが表示されたら、「はい」をクリックします。Windows の終了ができれば、図 3.22が再び表示されます。

3.4.14 電源を切る

初期メニューから矢印上下(↑↓)キーとリターンキーで「終了(電源OFF)」を選択してください。これで自動的に電源が切れます。故障の原因にもなりますので、本体やディスプレイ右下のスイッチを直接操作して電源を切らないでください。

3.4.15 部屋を出る

マシンの電源が切れたら部屋を出ます。このとき、部屋に誰も残らないようなら空調機、照明を切ってください。また、誰かが電源を切り忘れたパソコンがあるかもしれませんので、もしあれば電源を切っておいてください。

3.4.16 さて、さて、

最初の利用はこれでおしまいです。大変よくできました。あとは精進してUNIX ユーザへの道を歩まれるわけですね。それにはUNIXでの一般的なコマンド操作などについて説明している、第4章へと進んで下さい。

3.4.17 困った時の対処

ホストコンピュータに接続出来ない

- 図 3.26が表示されていない場合。図 3.25をマウスの左ボタンでダブルクリック時に失敗している事が考えられる。再度、図 3.25 をダブルクリックする。
- 図 3.26が表示されないまたは見えない場合。図 3.24 の後ろに隠れてしまっている事が考えられる。「Alt」キー(キーボード左下)を押しながら<ESC>キー(キーボード左上)を押して画面を切替えて探してください。
- 図 3.31が画面に表示され、その後に図 3.26が表示されない場合。パソコンの電源投入時にトラブルが発生したと考えられる。正しい手順通りにWindowsを終了し、図 3.22を表示させてください。再度メニュー上のMS-Windowsを選択しリターンキーを押して下さい。「端末ソフトの起動」から再度、始めてください。それでも、図 3.31が表示されるようなら、計算機センター相談窓口まで連絡して下さい。

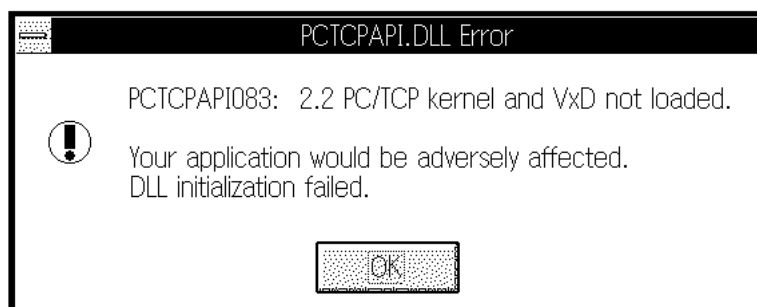


図 3.31 端末ソフトの起動失敗画面

login ができない

- 図 3.32 のように `login incorrect` が表示される。ユーザー名かパスワードが間違っている事が考えられる。大文字、小文字等の入力間違いがないかを確認、再度入力をしてください。

```
wvtn : cc (00:00:38)
-----
ホスト(O) 編集(E) 設定(S) ネットワーク(N) ヘルプ(H)
VTN for Windows (VT emulator)
Copyright (C) 1993 by Allied Telesis, K.K. All rights reserved.

UNIX(r) System V Release 4.0 (cc2000)

login: sheryl
Password:
Login incorrect
login: SHERYL
Password:
Login incorrect
login: █
```

図 3.32 login 失敗の画面

文字が全角になる

- 図 3.33 のように、`sheryl` と入力しているのに「しゅ r y l」と画面上に現れる事がある。これはパソコンの日本語変換機能が起動されている事が考えられる。画面右下に「あ自R漢」が表示されているのであれば、キーボードまん中下の「かな漢字」と刻印されているキーを押して、「あ自R漢」が消えるのを確認してください。これで日本語変換機能が止まります。

```
wvtn : cc (00:00:17)
-----
ホスト(O) 編集(E) 設定(S) ネットワーク(N) ヘルプ(H)
VTN for Windows (VT emulator)
Copyright (C) 1993 by Allied Telesis, K.K. All rights reserve

UNIX(r) System V Release 4.0 (cc2000)

login: しゅ r y l
Password: █
```

図 3.33 文字が全角になる

3.4.18 マニュアルなど

FMR-280 で利用できるソフトウェアのマニュアルは隣の部屋に置いてあります。教室の後ろにある扉を開けて自由に入室して参照してください。但し、教室からは持ち出さないようにしてください。

ソフトウェアの簡単な説明と、起動・終了方法なら FMR-280 を使って見ることができます。初期メニューから「利用手引き」を選択してください。EGDOC というソフトウェアが起動され、以下のような画面が表示されます。

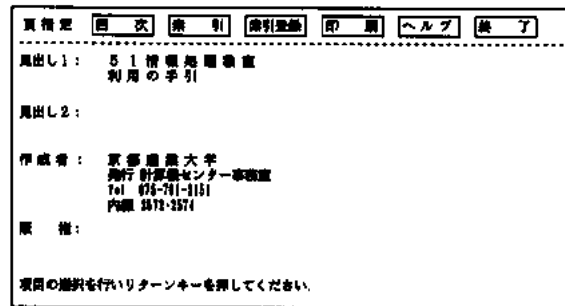


図 3.34 利用手引

あとは初期メニューと同様、カーソルキーとリターンキーを使って表示されている各項目を選択すれば読み進むことが出来ます。

EGDOC そのものの使い方については上の図にある「ヘルプ」を選択することによって見る事が出来ます。

3.5 Contura から telnet で cc2000 を使う

Contura は五号館 2 階の 52 情報処理教室に設置してあります。Contura はコンパック株式会社が発売している IBM 製パーソナルコンピュータ（以下パソコン）の互換機ノート型パソコンです。

ここでは五号館 2 階の 52 情報処理教室の Contura から telnet 環境で cc2000 をホストコンピュータとして利用する手続きについて説明します。

52 情報処理教室の Contura では端末ソフトとして NetManage 社の「Telnet」を使用します。これは telnet 環境を提供します。

Contura のキーボードでは、一般的にリターンキーと呼ばれるキー⁸には「Enter」と刻印されています。ここではこのキーを「リターンキー」と表現します。例示に<Return>と書いてある場合は Contura ではこの「Enter」キーを押して下さい。

3.5.1 まず部屋へ

この部屋にてパソコンを利用するためには、2つの鍵が必要です。部屋のドア用と机の引出し用です。パソコンはそれぞれの机の引出しに入っています。この2つの鍵は経営学部事務室が管理しています。したがって、部屋に入る人は、経営学部事務室で鍵を受取ってください。もし、あなたが学部学生または大学院生であれば、鍵を受取るときに学生証を提示しなければなりません。

部屋に入るときは、そこで授業をやっていないことを確認しましょう。もし授業中であれば入室して利用していいかどうか指導教員に確認を取るのが礼儀と言うものです。部屋に誰も居なかった場合は照明、空調機が切れている場合があります。照明のスイッチは、部屋に入ったそのすぐ左手の壁にありますので、自分で適当に操作してください。空調機のスイッチは、照明スイッチと同じ壁面にあります。部屋の前半部と後半部用にダイヤル式のスイッチが1つずつあります。部屋の温度を考えながら調整してください。部屋の管理そのものは経営学部事務室が行っていますので、何か部屋の状態について質問、要望があれば経営学部事務室までお願いします。コンピュータについての質問、要望は計算機センターの相談窓口までお願いします。

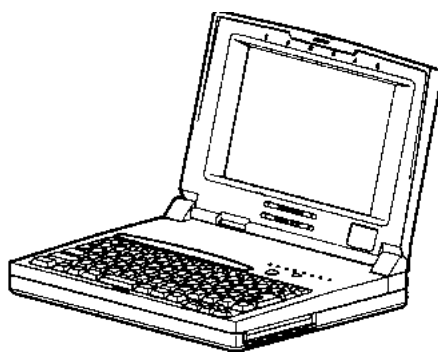


図 3.35 Contura 本体

3.5.2 机の引出しからパソコンを取出す

机の引出し用鍵には、机上番号に対応した番号札が付いています。その番号に対応する机に座ってください。それでは、机の引出しからパソコンとマウスの2つを取出しましょう。机の化粧板の陰になって確認しにくいのですが、パソコンには、赤い腹巻をした小箱状の装置が付いています。この小箱状の装置を

⁸ キーボード真中やや右の大きなキー

机の化粧板にぶつけない様に、引出しを十分開けてから取出してください。パソコン本体には電源や通信用のケーブルを1つに束ねた黒いスパイラルチューブがついています。このスパイラルチューブを引っ張りすぎない様にやさしく取扱ってください。

3.5.3 電源を入れる

パソコンの電源ボタンは液晶画面の下に隠れていますので、先ず液晶画面を開けましょう。この液晶画面を開けるには、手前のボタンを親指で押し下げながら、そのまま指を引っ掛ける様にして上に持ち上げてください。では、本体の電源を入れましょう。電源ボタンはキーボードの右上にある、紫色をした丸い2つのボタンの内の大きい方のボタンです。電源がすでに入っている場合は、電源ボタンの上方のグリーンランプが点灯していますのでグリーンランプが消えているのを確認してから、丸い紫色の電源ボタンを押してください。電源が入ったらモーターの回転音が生じ、画面の左上に数字メータが表示されカウントアップが始まります。それから画面に色々な文字が流れていって、合計1分足らずでマイクロソフト社のWindowsが立ち上がります。画面上に絵が現れたら、これで一段落です。もし、画面が文字の状態のまま全然変化しなかったら、パソコンの後ろに付いている赤い腹巻をした小箱状の装置を上から覗き込んで見てください。この小箱状の装置のランプが1つも点灯していなければ、この小箱状の装置に付属する線類の接触状況を調べてください。おそらくどれかが外れかかっているはずです。外れかかっている線類をしっかり固定してから、電源を再度入れ直してください。電源の入り切りは同じ大きい方の紫色の丸いボタンスイッチで操作できます。

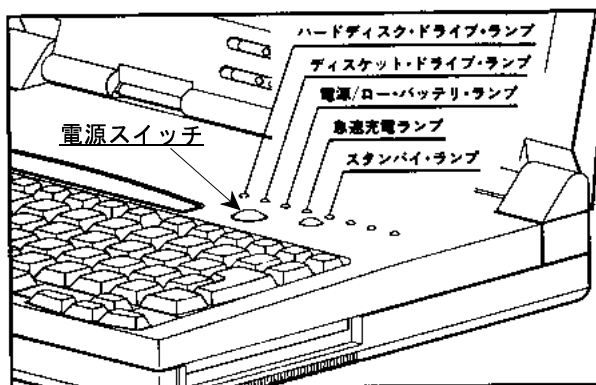


図 3.36 Contura のボタンとランプ

3.5.4 Windows の動かし方

Windows が立ち上がった状態（画面上に絵が表示された状態）になっています。Windows は、色々な機能を満載していながら、基本的な操作はほとんど共通に利用出来ます。焦らずゆっくりとマニュアルを読んで操作すれば大丈夫です。

C1 情報処理教室、51 情報処理教室には、初心者向けに「Quick ユーザーズガイド」が設置してあります。「Quick ユーザーズガイド」のポイント1からポイント10まで、読まれることをお奨めします。52 情報処理教室には、「今日から使える Windows3.1」と「機能ガイド」があります。「今日から使える Windows3.1」は初心者向けに書かれてありますので目を通すことをお奨めします。「機能ガイド」は Windows の詳細な機能が書かれてあります。

マニュアルの設置場所は、「マニュアルなど」の説明に書かれてありますので、そちらをお読みください。

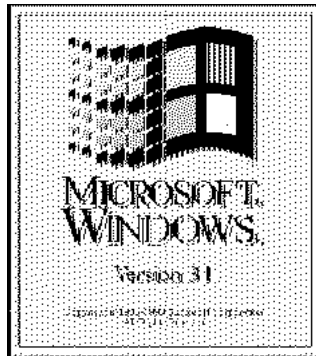


図 3.37 Windows 起動画面

3.5.5 Windows に関する基礎知識（絶対にやってはいけない事）

いきなり電源オフ

パソコンの操作方法がわからなくなってしまった時、ほとんどの方がとる手段が、電源を切ってしまうことです。この手段は絶対にしてはいけません。急に電源を切ってしまうと、パソコンの中の情報が破壊されて、次に電源をいれても正常に動作しません。あなたがパソコンの操作に迷った時は、取りあえず周りの人に聞くとよいでしょう。周りに人がいないのなら、計算機センターまで隣の部屋のコードレスホンで電話をしてください。それでほとんどが解決するはずです。いきなり電源を切らないでください。

書き込み中のフロッピー取り出し

フロッピーが動作しているかどうか（書き込み中であるかないか）を知るには、本体のやや真ん中右よりのフロッピー差し込み口を見てください。緑色のランプが点灯していれば、まさに書き込み中です。（52 情報処理教室の Contura ではランプは電源ボタンの上に並ぶランプの内のディスク・ドライブ・ランプです。図 3.36 を参照して下さい。因みにフロッピー差し込み口は本体右側面にあります。）このランプが点灯しているときに、フロッピーを取り出すと中身が壊れます。あなたが作った物すべてが読めなくなるかもしれません。フロッピーを取り出す時には十分注意しましょう。

むやみに削除を選択すること

Windows は自分が作成した文章、プログラムおよびパソコンを起動させるプログラムを簡単に表示、削除、移動、複写が行えます。あなたが作成した文章やプログラムを操作する分については問題がないのですが、パソコンにとって重要なプログラムを削除すると、特定の機能が使えなくなるか、パソコンが動かなくなります。もしわからなくなった場合は、周りの人に尋ねるか、計算機センター相談窓口まで問い合わせてください。

むやみに設定を変更すること

Windows には、色々な機能を満載していますが、その一つ一つは細々した設定がなされています。パソコンは不特定多数の利用者によって使われるますので、私たち計算機センターは可能な限りいつでも同じ環境を提供したいと思っています。設定が気に入らない場合もあるでしょうが、そのままの状態を使って

ください。もちろん「このようにした方がもっといいと思いますよ」といった提案はいつでも計算機センター相談窓口までお願いします。

3.5.6 端末ソフトの起動

まず端末ソフトを起動しましょう。

パソコン本体に電源が入れば、自動的にマイクロソフト社の Windows が起動し、図 3.38 が現れます。

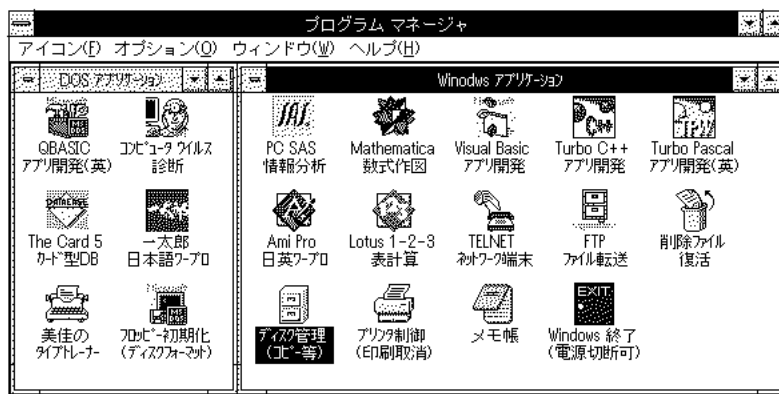


図 3.38 Windows 初期メニュー

図 3.38 の中の一つに「TELNET ネットワーク端末」図 3.39 があります。このアイコン（絵）にマウスポインタ（矢印）を重ね合わせて、マウスの左ボタンをダブルクリック（連続2回押し）してください。Telnet が起動されます。



図 3.39 Telnet のアイコン

3.5.7 ホストコンピュータへの接続

この端末ソフトは、起動すれば自動的に cc2000 に接続されます。

Telnet が起動されると、図 3.40 が画面に表示されます。もし画面に現れなかったら、あなたはソフトウェアの起動に失敗しています。もう一度図 3.39 にマウスポインタ（矢印）を重ね合わせてマウスの左ボタンをダブルクリックしてください。それでも図 3.40 が現れない場合は計算機センター相談窓口まで連絡してください。



図 3.40 cc2000 のプロンプト

ここで login: などと login の為の催促文字が表示されていることを確認して下さい。

3.5.8 さあ、login しよう

いよいよ login です。ここからはもうあなたのキーボードに対する入力は UNIX コンピュータに対する遠隔操作となっているのです。

ユーザ名とパスワードを入力しましょう。login: の後にあなたのユーザ名を入力して下さい。ユーザ名には大文字、小文字の区別があるので注意して下さい。ユーザ名の入力が完了したら、一度リターンキーを押して下さい。次にパスワードを入力します。ユーザー名を入力した時とは違い、パスワードは横から見ている他の人にバレないように、タイプしてもその文字が画面に表示されません。キーを打ち間違えないように注意して入力し、もう一度リターンキーを押して下さい。

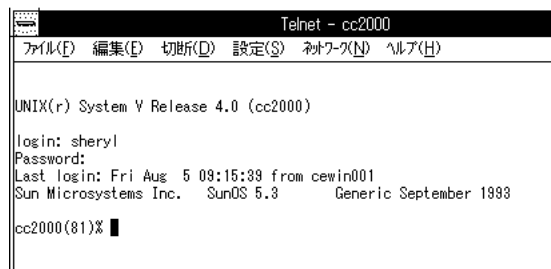


図 3.41 cc2000 のログインの SAMPLE

図 3.41 のようになりましたか？もし、ならなかった場合は、この後の「困ったときの対処」を参照してください。

3.5.9 ちょっとメッセージ

login すると、ときには利用者宛の通知、おしらせが以下のように画面に表示されるかも知れません。

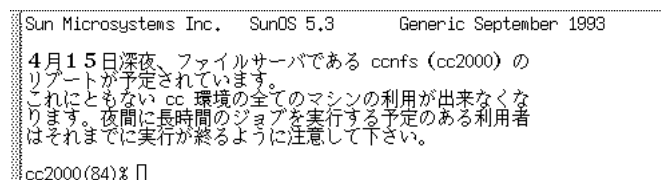


図 3.42 login 直後のメッセージ

よく読んで、自分に関係があるかどうか確認してください。もしも内容に付いてよく判らないことがあれば計算機センター相談窓口まで連絡するのがいいでしょう。

3.5.10 パスワードを変える

もしもこれがあなたに取って Sign 登録後初めての login ならば、ここでまずパスワードを変更しましょう。Sign 登録時のパスワードは機械的に決められたもので、余り安全とは言えません。必ずパスワードをあなた自身が選んだ単語に変更して下さい。パスワードを決める際には 4 ページの 2.1.3 で既に述べた事に注意してください。

パスワードを変更するには `yppasswd` コマンド⁹を利用します。上の節で説明したターミナルを見ると、

```
cc2000(81)%
```

などとなっていますね。(但し数字は 81 ではないかもしれません。) ここで `yppasswd` とタイプし、リターンキーを押します。すると以下のような状態に成りますね。(以下はユーザ名 `yasuda` の例。`yasuda` の部分にはあなたのユーザ名が表示されているはずです。)

```
cc2000(81)% yppasswd
Changing NIS password for yasuda
Old NIS password:
```

指示通りここでは古い、つまり今先ほどまで使っていたパスワードをタイプし、リターンキーを押します。ここでは login の時と同じくパスワードをタイプしている間、横の人の覗き見によってあなたのパスワードがバレないように、タイプした文字は画面に表示されない事に注意してください。表示されなくともタイプした文字はちゃんと入力されていますから、安心して確実に一文字ずつタイプして最後にリターンキーをタイプして下さい。すると今度は以下のようなプロンプトを表示します。

```
New password:
```

では指示通り新しい、つまり次からこれにしようと言うパスワードをタイプし、リターンキーを押してください。ここでもタイプした文字は表示されません。今度は以下のようなプロンプトを表示します。

```
Retype new password:
```

これはいましがたタイプした新しいパスワードが打ち間違いで無いことを確認するためのものです。もう一度新しいパスワードをタイプしてリターンキーを押してください。今度は以下のような状態になります。

```
NIS passwd changed on ccnic
cc2000(82)%
```

この `NIS passwd changed` が表示されれば成功です。もしもパスワードが短すぎたりタイプミスがあったりしたらなんらかのエラーメッセージが表示されます。途中でおかしくなったと思う場合は `C-c` (つまりコントロールキーを押しながら C キーを押す) で `cc2000(82)%` を表示させます。それからもう一度 `yppasswd` コマンドをやり直します。

⁹綴りがちょっと変ですが、間違っていないです。

3.5.11 logout をしよう

次はlogout です。他にやりたいこともいろいろあるでしょうが、今回は最初の利用でしょうし、まず一通りの操作をやってみると言う意味でlogout をこの場でやってみることを勧めます。画面上にcc2000(81)%が表示されていることを確認してください。但し括弧の中の数字は気にしないでね!! 以下のように、logout <Return>と入力してください。

```
cc2000(81)% logout
```

3.5.12 端末ソフトを終了する

logout すると図 3.40 に良く似た画面が表示されていると思います。その図の左上部分に注目してください。「ファイル」という文字があります。それにマウスポインタを重ね合わせ、マウスの左ボタンをクリックしてください。機能の一覧が表示されます。図 3.43 です。その最下行に「終了」があります。マウスポインタを「終了」に重ね合わせてマウスの左ボタンをクリックしてください。電源を入れた最初の状態(図 3.38)に戻ります。

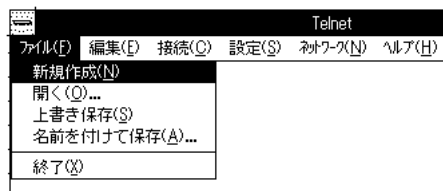


図 3.43 Telnet の終了

3.5.13 電源を切る

Windows を終了させるには、「Windows 終了」という絵(アイコン)をマウスポインタで二度クリックします。「Ms-Windows を終了します」という確認メッセージが表示されたら、「はい」をクリックします。Windows の終了ができたなら、画面に電源を切っても大丈夫ですといった様なメッセージが表示されます。紫色の丸い電源ボタンを押して電源を切ってください。お疲れさまでした。

3.5.14 パソコンを机の引出しにしまう

パソコンとマウスを元通りに机の引出しにしまってから、鍵を閉めてください。引出しがしまりにくいときは、引出しの中の線類をやさしく手前に引いてみてください。無理に引っ張ると断線する恐れがありますので、注意してください。

3.5.15 部屋を出る

部屋に誰も残らないようなら空調機、照明を切り、最後に部屋の鍵を閉めてから経営学部事務室まで鍵を返しに行ってください。

3.5.16 さて、さて、

最初の利用はこれでおしまいです。大変よくできました。あとは精進して UNIX ユーザへの道を歩まれるわけですね。それには UNIX での一般的なコマンド操作などについて説明している、第 4 章へと進んで下さい。

3.5.17 困った時の対処

ホストコンピュータに接続出来ない

- 図 3.40が表示されていない場合。図 3.39をマウスの左ボタンでダブルクリック時に失敗している事が考えられる。再度、図 3.39 をダブルクリックする。
- 図 3.40は表示されているが、図 3.38 の後ろに隠れてしまっている事が考えられる。「Alt」キー（キーボード左下）を押しながら<ESC>キー（キーボード左上）を押して画面を切替えて探してください。
- 図 3.44 が画面に表示されて図 3.40にならない場合がある。接続したいコンピュータが何らかの理由で停止しているかまたは貴方が使用しているパソコンに障害が生じている可能性があります。計算機センターの相談窓口まで連絡してください。



図 3.44 接続失敗の画面

login ができない

- 図 3.45 のように `login incorrect` が表示される。ユーザー名かパスワードが間違っている事が考えられる。大文字、小文字等の入力間違いがないかを確認、再度入力して下さい。

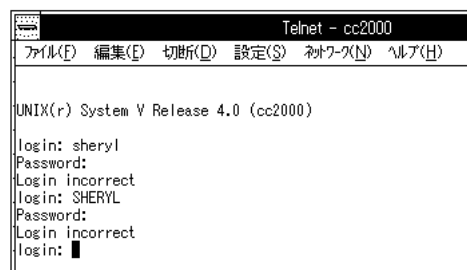


図 3.45 login 失敗の画面

文字が全角になる

- 図 3.46のように、`sheryl` と入力しているのに「しえ r y l」と画面上に現れる事がある。これはパソコンの日本語変換機能が起動されている事が考えられる。画面右下に「あ自R漢」が表示されているのであれば、「Alt」キー（キーボード左下）を押しながら「半角/全角」キー（キーボード左上）を押して、日本語変換機能を止めてください。画面右下の「あ自R漢」が消えます。

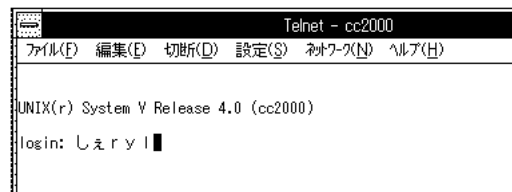


図 3.46 文字が全角になる

3.5.18 マニュアルなど

パソコンのマニュアルは部屋に入った左側に設置してあるロッカーに入っています。どうぞ自由に取り出してみてください。但し、持ち帰りはいけません!!。禁帯出です。

3.6 FMV から X で cc2000 を使う

ここでは計算機科学研究所棟 2 階の C1 情報処理教室の FMV から X 環境でホストコンピュータを利用する手続きについて説明します。利用するホストコンピュータとしては cc2000 を例として示しています。

C1 情報処理教室の FMV では端末ソフトとして Hummingbird Communications 社の「eXceed」を使用します。これは X 環境を提供します¹⁰。

FMV のキーボードでは、一般的にリターンキーと呼ばれるキー¹¹には「Enter」と刻印されています。ここではこのキーを「リターンキー」と表現します。例示に<Return>と書いてある場合は FMV ではこの「Enter」キーを押して下さい。

3.6.1 まず部屋へ

この部屋は常時カードロックシステムが働いており、学生証を入り口のドア横の機械に通すことによって鍵が開きます。入室可能な時間帯、曜日、期間に限りがありますので、付録の「情報処理教室利用要項」を一読してください。もしもカードを通して鍵が開かない場合は、カードの向き、表裏、カードを通過させる速度（遅すぎても速すぎても駄目）を変えてやってください。どうしても駄目な場合はカードの磁気情報がおかしくなっている可能性がありますので、カードを通す機械にあなたのカードでなぜ鍵が開かないか、その理由が出ていますので、それを控えて計算機センターまで連絡してください。

部屋に入るときは、そこで授業をやっていないことを確認しましょう。もし授業中であれば入室して利用していいかどうか指導教員に確認を取るのが礼儀と言うものです。部屋に誰も居なかった場合は照明、空調機、ブレーカー（供給元電源）が切れている場合があります。照明のスイッチは、部屋に入ったそのすぐ左手の壁にありますので、自分で適当に操作してください。空調機は、部屋の奥（窓の下側）に 3 台取り付けてあります。スイッチは各空調機の左側上部の push 式扉の下に隠れています。部屋の温度を考えながら調整してください。ブレーカーのスイッチは、部屋に入って左側柱の奥にある黒いスイッチです。黒いスイッチが上向きになっている状態であれば電源が供給されています。部屋の管理そのものは計算機センター事務室が行っていますので、何か部屋の状態について質問、要望があるばあい、またコンピュータについての質問、要望なども計算機センターの相談窓口までお願いします。

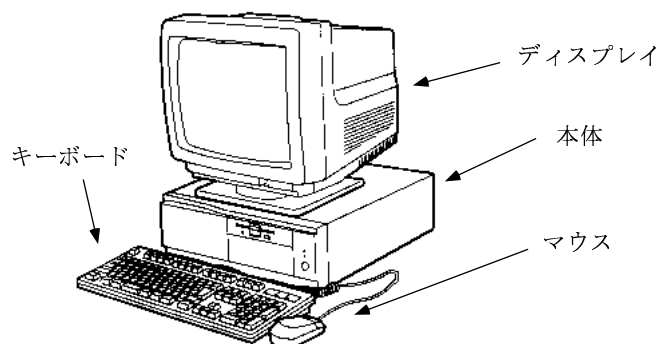


図 3.47 FMV 本体

3.6.2 電源を入れる

自分が利用するコンピュータを選びます。3 2 台ありますので空いている場所を見つけて着席してください。その時、ディスプレイ装置（テレビみたいな物）、本体（ディスプレイ装置の下にある箱）にグリー

¹⁰ このように X 環境を提供する端末ソフトを X 端末ソフトと呼んだりします。

¹¹ キーボード真中やや右の大きなキー

ンのランプが点灯していないか確認してください。もし、グリーンランプが点灯しているのであれば、誰かがすでに利用しているのかもしれない。

ではまず、ディスプレイ装置（テレビみたいな物）の電源を入れましょう。ディスプレイ装置右端下に少し大きめの長四角のボタンがあります。その長四角のボタンの左横に小さい四角窓があり、グリーンランプが点灯していないのを確認してください。点灯していないようなら、長四角のボタンを押してください。グリーンランプが点灯しているのであれば、すでにディスプレイ装置に電源が入っています。次に、本体の電源を入れます。本体はディスプレイ装置の下にある四角い箱です。本体の電源は、右端真ん中にある丸いボタンです。電源がすでに入っている場合は、丸いボタンの上にグリーンランプが点灯していますので、グリーンランプが消えているのを確認してから、丸い電源ボタンを押してください。グリーンランプの点灯がディスプレイ装置、本体と確認できたら、しばらく（30秒ぐらい）待ってください。自動的にプログラムが起動されてマイクロソフト社の windows が立ち上がります。画面上に絵が表れたら、これで一段落です。

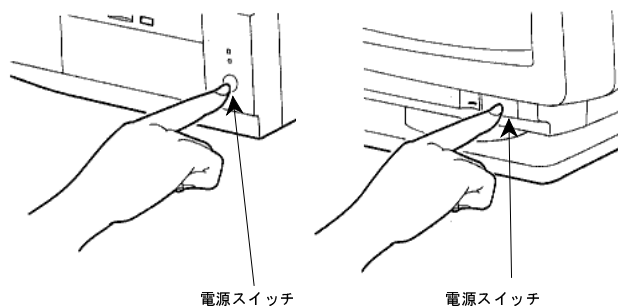


図 3.48 左図：FMV 本体の電源オン、右図：ディスプレイ装置の電源オン

3.6.3 Windows の動かし方

Windows が立ち上がった状態（画面上に絵が表示された状態）になっています。Windows は、色々な機能を満載していながら、基本的な操作はほとんど共通に利用出来ます。焦らずゆっくりとマニュアルを読んで操作すれば大丈夫です。

C1 情報処理教室、51 情報処理教室には、初心者向けに「Quick ユーザーズガイド」が設置してあります。「Quick ユーザーズガイド」のポイント 1 からポイント 10 まで、読まれることをお奨めします。52 情報処理教室には、「今日から使える Windows3.1」と「機能ガイド」があります。「今日から使える Windows3.1」は初心者向けに書かれてありますので目を通すことをお奨めします。「機能ガイド」は Windows の詳細な機能が書かれてあります。

マニュアルの設置場所は、「マニュアルなど」の説明に書かれてありますので、そちらをお読みください。

3.6.4 Windows に関する基礎知識（絶対にやってはいけない事）

いきなり電源オフ

パソコンの操作方法がわからなくなってしまった時、ほとんどの方がとる手段が、電源を切ってしまうことです。この手段は絶対にしてはいけません。急に電源を切ってしまうと、パソコンの中の情報が破壊されて、次に電源をいれても正常に動作しません。あなたがパソコンの操作に迷った時は、取りあえず周

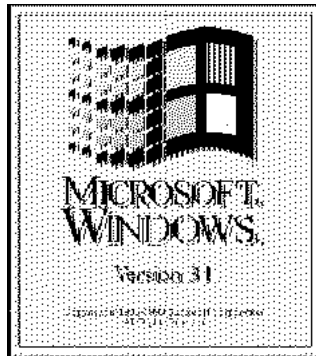


図 3.49 Windows 起動画面

りの人に聞くとよいでしょう。周りに人がいないのなら、計算機センターまで隣の部屋のコードレスホンで電話をしてください。それでほとんどが解決するはずです。いきなり電源を切らないでください。

書き込み中のフロッピー取り出し

フロッピーが動作しているかどうか（書き込み中であるかないか）を知るには、本体のやや真ん中右よりのフロッピー差し込み口を見てください。グリーンランプが点灯していれば、まさに書き込み中です。（52 情報処理教室の Contura ではランプは電源ボタンの上に並ぶランプの内のディスケット・ドライブ・ランプです。図 3.36 を参照して下さい。因みにフロッピー差し込み口は本体右側面にあります。）このランプが点灯しているときに、フロッピーを取り出すと中身が壊れます。あなたが作った物すべてが読めなくなるかもしれません。フロッピーを取り出す時には十分注意しましょう。

むやみに削除を選択すること

Windows は自分が作成した文章、プログラムおよびパソコンを起動させるプログラムを簡単に表示、削除、移動、複写が行えます。あなたが作成した文章やプログラムを操作する分については問題がないのですが、パソコンにとって重要なプログラムを削除すると、特定の機能が使えなくなるか、パソコンが動かなくなります。もしわからなくなった場合は、周りの人に尋ねるか、計算機センター相談窓口まで問い合わせてください。

むやみに設定を変更すること

Windows には、色々な機能を満載していますが、その一つ一つは細々した設定がなされています。パソコンは不特定多数の利用者によって使われるますので、私たち計算機センターは可能な限りいつでも同じ環境を提供したいと思っています。設定が気に入らない場合もあるでしょうが、そのままの状態ですべて使ってください。もちろん「このようにした方がもっといいと思いますよ」といった提案はいつでも計算機センター相談窓口までお願いします。

3.6.5 端末ソフトの起動

まず端末ソフトを起動しましょう。

パソコン本体とディスプレイ装置に電源が入れば、自動的にマイクロソフト社の Windows が起動し、図 3.50 が現れます。

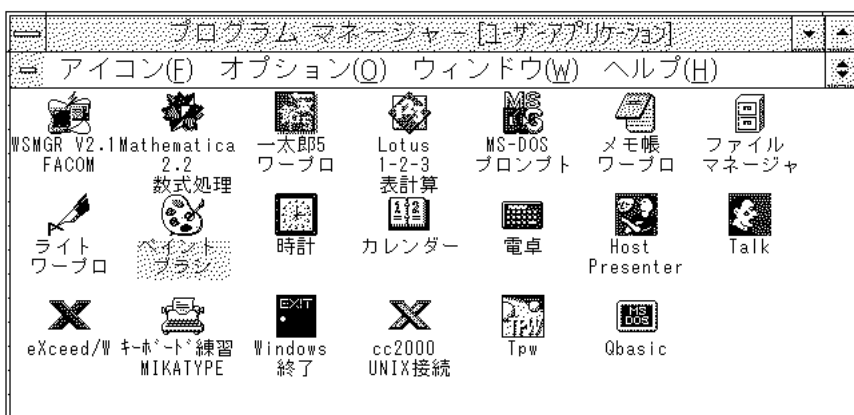


図 3.50 Windows 初期メニュー

図 3.50 の中の一つに「cc2000 UNIX 接続」図 3.51 があります。このアイコン（絵）にマウスポインタ（矢印）を重ね合わせて、マウスの左ボタンをダブルクリック（連続 2 回押し）してください。eXceed が起動されます。すぐ近くに良く似た形の「eXceed/W」というアイコンがあるかも知れませんが、これと間違わないように注意して下さい。



図 3.51 eXceed のアイコン

上記の手続きで eXceed が起動されると、まず画面全体が灰色になります。そして画面中央に図 3.52 が表示されます。もし画面全体が変わらなかつたら、あなたは端末ソフトの起動に失敗しています。もう一度図 3.51 にマウスポインタ（矢印）を重ね合わせてマウスの左ボタンをダブルクリックしてください。それでも図 3.52 が現れない場合は計算機センター相談窓口まで連絡してください。



図 3.52 eXceed の初期画面

3.6.6 login する

いよいよ login です。

ユーザ名とパスワードを入力しましょう。

まずあなたのユーザ名をタイプして下さい。「login:」の右にタイプされた文字が入力されていくのがわかりますね。ユーザ名には大文字、小文字の区別があるので注意して下さい。ユーザ名の入力完了したら、リターンキーを押して下さい。

次にパスワードを入力します。ユーザー名を入力した時とは違い、パスワードは横から見ている他の人にバレないように、タイプした文字は画面に表示されません。キーを打ち間違えないように注意して入力し、ここでリターンキーを押して下さい。



図 3.53 cc2000 の X 端末からのログインの SAMPLE

図 3.53 のようになりましたか？もし、ならなかった場合は、この後の「困ったときの対処」を参照してください。

3.6.7 ちょっとメッセージ

login すると、ときには利用者宛の通知、おしらせが以下のように画面に表示されるかも知れません。

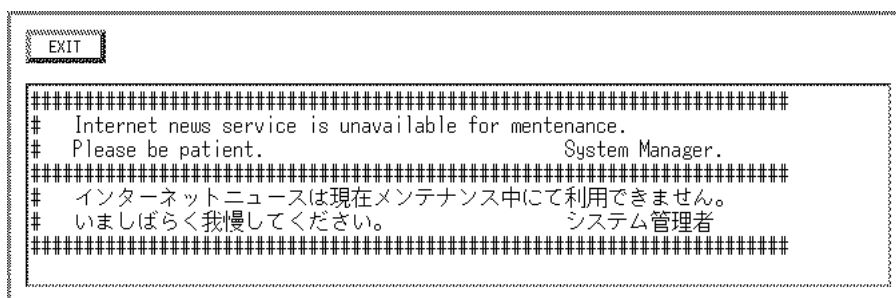


図 3.54 login 直後のメッセージ

よく読んで、自分に関係があるかどうか確認してください。この表示はリターンキーを一度押すことによって消す事が出来ます。もしも内容に付いてよく判らないことがあれば計算機センター相談窓口まで連絡するのがいいでしょう。

3.6.8 ターミナルはあるかな？

画面の中に「kterm」というタイトルのついた以下の様なウィンドウがあるかどうかさがしてください。そのようなウィンドウが見つかれば、ひとつ飛ばして「ターミナルは大丈夫かな？」まで進んで下さい。もしも見つからなければこのまま次に進んで下さい。



図 3.55 kterm

ターミナルを起動する

画面の背景、特にウィンドウの無い場所にマウスカーソルを持って行って、マウスの左ボタンを押せばなしにします。すると以下のようなメニューが現れるでしょう。この一連の操作を「ルートメニューを表示させる」と呼びます。



図 3.56 ルートメニュー

マウスのボタンを押したままにして、マウスカーソルを「Kterm」と書いてある行の上に持っていきます。「Kterm」の行の上まで来たならその場所でマウスボタンを離します。これで Kterm が起動されるはずですが。

この一連の操作を「メニューから「Kterm」を選択する」と呼びます。このメニューでは「Kterm」が選択されると Kterm が起動するようになっています。

ターミナルは大丈夫かな？

ターミナルのウィンドウの左隅には cc2000(81)% などと表示されていると思います。(但し数字は違うかもしれません。) ここで試しに date とタイプしてリターンキーを押してみてください。文字をタイプしても cc2000(81)% に続いて表示されない場合は、ターミナルウィンドウがアクティブになっていません。一度 cc2000(81)% 辺りをクリックしてターミナルウィンドウをアクティブウィンドウした後、date とタイプし直してください。以下のように今日の日付と時間が表示されれば大丈夫、ターミナルを使える状態になっています。

```
cc2000(81)% date
```

```
1994年02月27日(日)20時05分03秒
```

```
cc2000(82)%
```

3.6.9 パスワードを変える

もしもこれがあなたにとって Sign 登録後初めての login ならば、ここでまずパスワードを変更しましょう。Sign 登録時のパスワードは機械的に決められたもので、余り安全とは言えません。必ずパスワードをあなた自身が選んだ単語に変更して下さい。パスワードを決める際には 4 ページの 2.1.3 で既に述べた事に注意してください。

パスワードを変更するには `yppasswd` コマンド¹²を利用します。上の節で説明したターミナルを見ると、

```
cc2000(81)%
```

などとなっていますね。(但し数字は 81 ではないかもしれません。)ここで `yppasswd` とタイプし、リターンキーを押します。すると以下のような状態に成りますね。(以下はユーザ名 `yasuda` の例。`yasuda` の部分にはあなたのユーザ名が表示されているはずです。)

```
cc2000(81)% yppasswd
```

```
Changing NIS password for yasuda
```

```
Old NIS password:
```

指示通りここでは古い、つまり今先ほどまで使っていたパスワードをタイプし、リターンキーを押します。ここでは login の時と同じくパスワードをタイプしている間、横の人の覗き見によってあなたのパスワードがバレないように、タイプした文字は画面に表示されない事に注意してください。表示されなくともタイプした文字はちゃんと入力されていますから、安心して確実に一文字ずつタイプして最後にリターンキーをタイプして下さい。すると今度は以下のようなプロンプトを表示します。

```
New password:
```

では指示通り新しい、つまり次からこれにしようと言うパスワードをタイプし、リターンキーを押して下さい。ここでもタイプした文字は表示されません。今度は以下のようなプロンプトを表示します。

```
Retype new password:
```

これはいましがたタイプした新しいパスワードが打ち間違いで無いことを確認するためのものです。もう一度新しいパスワードをタイプしてリターンキーを押して下さい。今度は以下のような状態になります。

```
NIS passwd changed on ccnic
```

```
cc2000(82)%
```

この `NIS passwd changed` が表示されれば成功です。もしもパスワードが短すぎたりタイプミスがあったりしたらなんらかのエラーメッセージが表示されます。途中でおかしくなったと思う場合は C-c (つまりコントロールキーを押しながら C キーを押す) で `cc2000(82)%` を表示させます。それからもう一度 `yppasswd` コマンドをやり直します。

¹²綴りがちょっと変ですが、間違っではありません。

3.6.10 logout (セッションの終了)

他にやりたい事も色々あるでしょうが、今回は最初の利用でしょうし、まず一通りの操作をやってみるという意味で、logout をこの場でやってみる事を勧めます。

画面の背景、特にウィンドウの無い場所にマウスカーソルを持って行って、マウスの左ボタンを押せばなしにします。すると以下のようなメニューが現れるでしょう。この一連の操作を「ルートメニューを表示させる」と呼びます。



図 3.57 cc2000 のルートメニュー

マウスのボタンを押したままにして、マウスカーソルを恐らくは一番下にある「End Session」と書いてある行の上に持っていきます。「End Session」の行の上まで来たらその場所でマウスボタンを離します。この一連の操作を「メニューから「End Session」を選択する」と呼びます。このメニューでは「End Session」が選択されると logout (セッションの終了) 作業が行なわれるようになっています。

これが FMV の X での cc2000 からの logout 操作です。実際には logout という名前もコマンドも操作も実行しない事に注意して下さい。FMV から X で cc2000 を使っている場合ではつまり logout とはセッションの終了だという事です。セッション終了の操作を行なって数十秒すると、端末ソフトも自動的に終了処理がなされ、端末ソフトを起動する前の状態、つまり図 3.50 が表示された状態になります。もしもそうならなければセッション終了操作をやり直しです。

3.6.11 電源を切る

Windows を終了させるには、「Windows 終了」という絵 (アイコン) をマウスポインタで二度クリックします。「Ms-Windows を終了します」という確認メッセージが表示されたら、「はい」をクリックします。Windows の終了ができたなら、本体電源スイッチの上にオレンジランプが点灯していないのを確認してから、丸い電源ボタンを押してください。その次に、ディスプレイ装置の長四角のボタンを押してください。ディスプレイ装置、本体にグリーンランプの点灯がなければ電源が切れています。お疲れさまでした。

3.6.12 部屋を出る

パソコンの電源が切れたら部屋を出ます。このとき、部屋に誰も残らないようなら空調機、照明、ブレーカーを切って行きましょう。ブレーカーを切るときは、すべてのパソコンに電源が入っていないのを確認してからにしてください。

3.6.13 さて、さて、

最初の利用はこれでおしまいです。大変よくできました。あとは精進して UNIX ユーザへの道を歩まれるわけですね。それには UNIX での一般的なコマンド操作などについて説明している、第 4 章へと進んで下さい。

3.6.14 困った時の対処

端末ソフトが起動しない

- 図 3.51をマウスの左ボタンでダブルクリック時に失敗している事が考えられる。再度、図 3.51 をダブルクリックする。
- 表示はされているが、図 3.50 の後ろに隠れてしまっている事が考えられる。「Alt」キー（キーボード左下）を押しながら<ESC> キー（キーボード左上）を押して画面を切替えて探してください。

login ができない

- 赤い文字で「Login incorrect」と表示される。
ユーザー名かパスワードが間違っている事が考えられます。大文字、子文字等の入力間違いがないかを確認して、再度ユーザー名の入力からやり直します。

文字が全角になる

- パソコンの日本語変換が起動されている事が考えられる。キーボードをタイプしても画面上にタイプした文字とは違うひらがなや漢字が表示されたり、画面右下に「あ自R漢」が表示されているのであれば、「Alt」キーを押しながら「半角/全角」キー（キーボード左上）を押して、日本語変換機能を止めて下さい。画面右下の「あ自R漢」が消えます。

3.7 FMV から telnet で cc2000 を使う

FMV は計算機科学研究所棟の C1 情報処理教室に設置してあります。FMV とは (株) 富士通が発売している IBM 製パーソナルコンピュータ (以下パソコン) の互換機です。

ここでは計算機科学研究所棟 2 階の C1 情報処理教室の FMV から telnet 環境で cc2000 をホストコンピュータとして利用する手続きについて説明します。

C1 情報処理教室の FMV では端末ソフトとしてノベル株式会社の「Host Presenter」を利用します。これは telnet 環境を提供します。

FMV のキーボードでは、一般的にリターンキーと呼ばれるキー¹³には「Enter」と刻印されています。ここではこのキーを「リターンキー」と表現します。例示に<Return>と書いてある場合は FMV ではこの「Enter」キーを押して下さい。

3.7.1 まず部屋へ

この部屋は常時カードロックシステムが働いており、学生証を入り口のドア横の機械に通すことによって鍵が開きます。入室可能な時間帯、曜日、期間に限りがありますので、付録の「情報処理教室利用要項」を一読してください。もしもカードを通しても鍵が開かない場合は、カードの向き、表裏、カードを通過させる速度 (遅すぎても速すぎても駄目) を変えてやってください。どうしても駄目な場合はカードの磁気情報がおかしくなっている可能性がありますので、カードを通す機械にあなたのカードでなぜ鍵が開かないか、その理由が出ていますので、それを控えて計算機センターまで連絡してください。

部屋に入るときは、そこで授業をやっていないことを確認しましょう。もし授業中であれば入室して利用していいかどうか指導教員に確認を取るのが礼儀と言うものです。部屋に誰も居なかった場合は照明、空調機、ブレーカー (供給元電源) が切れている場合があります。照明のスイッチは、部屋に入ったそのすぐ左手の壁にありますので、自分で適当に操作してください。空調機は、部屋の奥 (窓の下側) に 3 台取り付けてあります。スイッチは各空調機の左側上部の push 式扉の下に隠れています。部屋の温度を考えながら調整してください。ブレーカーのスイッチは、部屋に入って左側柱の奥にある黒いスイッチです。黒いスイッチが上向きになっている状態であれば電源が供給されています。部屋の管理そのものは計算機センター事務室が行っていますので、何か部屋の状態について質問、要望があるばあい、またコンピュータについての質問、要望なども計算機センターの相談窓口までお願いします。

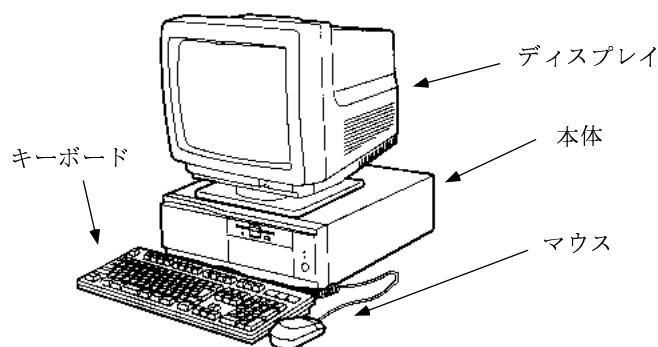


図 3.58 FMV 本体

¹³ キーボード真中やや右の大きなキー

3.7.2 電源を入れる

自分が利用するコンピュータを選びます。3 2台ありますので空いている場所を見つけて着席してください。その時、ディスプレイ装置（テレビみたいな物）、本体（ディスプレイ装置の下にある箱）にグリーンランプが点灯していないか確認してください。もし、グリーンランプが点灯しているのであれば、誰かがすでに利用しているのかもしれませんが。

ではまず、ディスプレイ装置（テレビみたいな物）の電源を入れましょう。ディスプレイ装置右端下に少し大きめの長四角のボタンがあります。その長四角のボタンの左横に小さい四角窓があり、グリーンランプが点灯していないのを確認してください。点灯していないようなら、長四角のボタンを押してください。グリーンランプが点灯しているのであれば、すでにディスプレイ装置に電源が入っています。次に、本体の電源を入れます。本体はディスプレイ装置の下にある四角い箱です。本体の電源は、右端真ん中にある丸いボタンです。電源がすでに入っている場合は、丸いボタンの上にグリーンランプが点灯していますので、グリーンランプが消えているのを確認してから、丸い電源ボタンを押してください。グリーンランプの点灯がディスプレイ装置、本体と確認できたら、しばらく（30秒ぐらい）待ってください。自動的にプログラムが起動されてマイクロソフト社の windows が立ち上がります。画面上に絵が表れたら、これで一段落です。

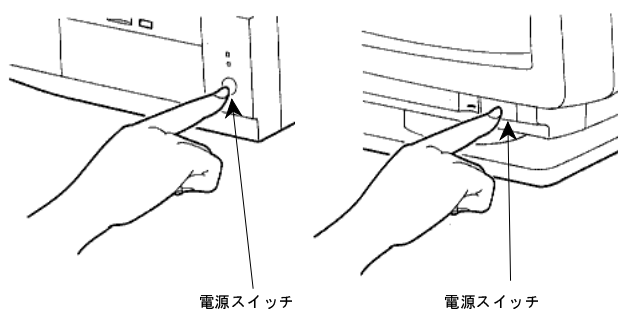


図 3.59 左図：FMV本体の電源オン、右図：ディスプレイ装置の電源オン

3.7.3 Windows の動かし方

Windows が立ち上がった状態（画面上に絵が表示された状態）になっています。Windows は、色々な機能を満載しているながら、基本的な操作はほとんど共通に利用出来ます。焦らずゆっくりとマニュアルを読んで操作すれば大丈夫です。

C 1 情報処理教室、5 1 情報処理教室には、初心者向けに「Quick ユーザーズガイド」が設置してあります。「Quick ユーザーズガイド」のポイント 1 からポイント 10 まで、読まれることをお奨めします。5 2 情報処理教室には、「今日から使える Windows3.1」と「機能ガイド」があります。「今日から使える Windows3.1」は初心者向けに書かれてありますので目を通すことをお奨めします。「機能ガイド」は Windows の詳細な機能が書かれてあります。

マニュアルの設置場所は、「マニュアルなど」の説明に書かれてありますので、そちらをお読みください。

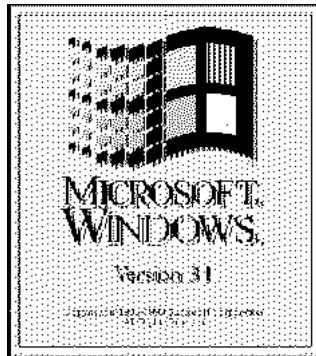


図 3.60 Windows 起動画面

3.7.4 Windows に関する基礎知識（絶対にやってはいけない事）

いきなり電源オフ

パソコンの操作方法がわからなくなってしまった時、ほとんどの方がとる手段が、電源を切ってしまうことです。この手段は絶対にしてはいけません。急に電源を切ってしまうと、パソコンの中の情報が破壊されて、次に電源をいれても正常に動作しません。あなたがパソコンの操作に迷った時は、取りあえず周りの人に聞くとよいでしょう。周りに人がいないのなら、計算機センターまで隣の部屋のコードレスホンで電話をしてください。それでほとんどが解決するはずです。いきなり電源を切らないでください。

書き込み中のフロッピー取り出し

フロッピーが動作しているかどうか（書き込み中であるかないか）を知るには、本体のやや真ん中右よりのフロッピー差し込み口を見てください。緑色のランプが点灯していれば、まさに書き込み中です。（52 情報処理教室の Contura ではランプは電源ボタンの上に並ぶランプの内のディスク・ドライブ・ランプです。図 3.36 を参照して下さい。因みにフロッピー差し込み口は本体右側面にあります。）このランプが点灯しているときに、フロッピーを取り出すと中身が壊れます。あなたが作った物すべてが読めなくなるかもしれません。フロッピーを取り出す時には十分注意しましょう。

むやみに削除を選択すること

Windows は自分が作成した文章、プログラムおよびパソコンを起動させるプログラムを簡単に表示、削除、移動、複写が行えます。あなたが作成した文章やプログラムを操作する分については問題がないのですが、パソコンにとって重要なプログラムを削除すると、特定の機能が使えなくなるか、パソコンが動かなくなります。もしわからなくなった場合は、周りの人に尋ねるか、計算機センター相談窓口まで問い合わせてください。

むやみに設定を変更すること

Windows には、色々な機能を満載していますが、その一つ一つは細々した設定がなされています。パソコンは不特定多数の利用者によって使われるますので、私たち計算機センターは可能な限りいつでも同じ環境を提供したいと思っています。設定が気に入らない場合もあるでしょうが、そのままの状態を使って

ください。もちろん「このようにした方がもっといいと思いますよ」といった提案はいつでも計算機センター相談窓口までお願いします。

3.7.5 端末ソフトの起動

まず端末ソフトを起動しましょう。

パソコン本体とディスプレイ装置に電源が入れば、自動的にマイクロソフト社の Windows が起動し、図 3.61 が現れます。

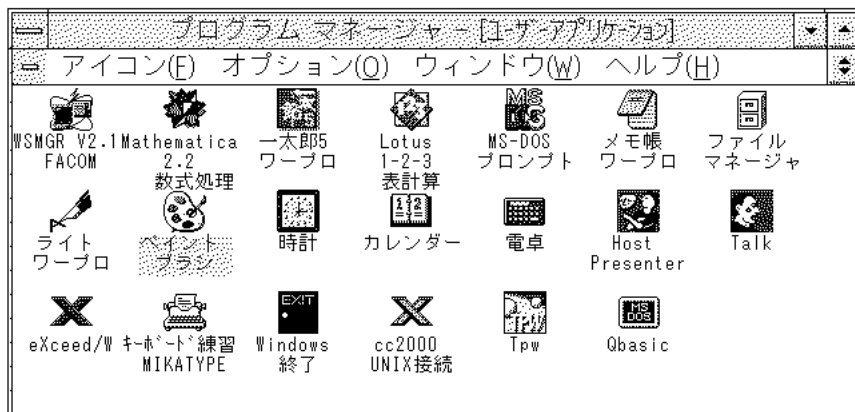


図 3.61 Windows 初期メニュー

図 3.61 の中の一つに「Host Presenter」図 3.62 があります。このアイコン（絵）にマウスポインタ（矢印）を重ね合わせて、マウスの左ボタンをダブルクリック（連続 2 回押し）してください。Host Presenter が起動されます。



図 3.62 Host Presenter のアイコン

3.7.6 ホストコンピュータへの接続

端末ソフトが起動されたら、まずはどのホストコンピュータの端末になるかを指示します。

Host Presenter が起動されると、図 3.63 が画面に表示されます。もし画面に現れなかったら、あなたはソフトウェアの起動に失敗しています。もう一度図 3.62 にマウスポインタ（矢印）を重ね合わせてマウスの左ボタンをダブルクリックしてください。それでも図 3.63 が現れない場合は計算機センター相談窓口まで連絡してください。

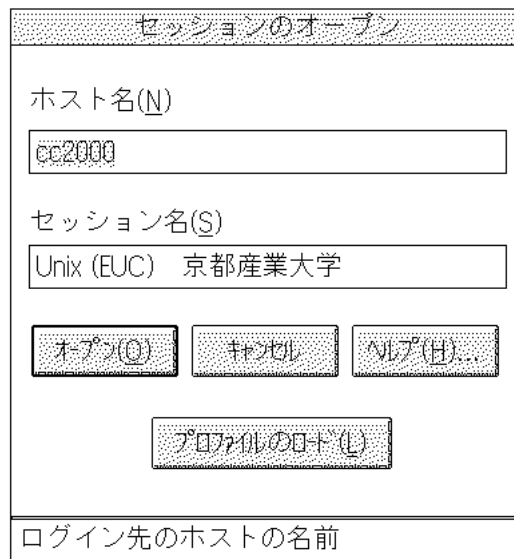


図 3.63 Host Presenter の初期画面

画面上には、図 3.63 が表示されており、ホスト名として「cc2000」が網掛けの文字として表示されています。（もし、表示されていない場合は、英小文字と数字で「cc2000」と入力してください。）そこでリターンキーを押して下さい。画面に図 3.64 のウィンドウが表示されます。

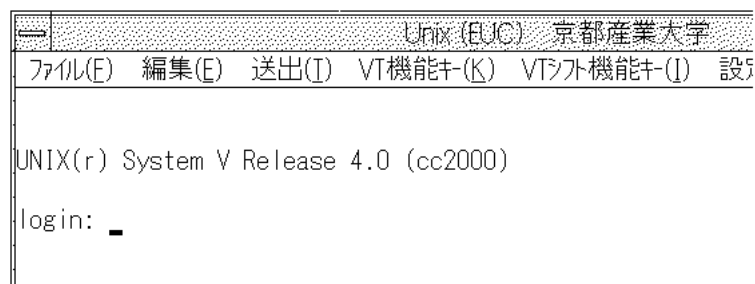


図 3.64 cc2000 のプロンプト

ここで login: などと login の為の催促文字が表示されていることを確認して下さい。

3.7.7 さあ、login しよう

いよいよ login です。ここからはもうあなたのキーボードに対する入力はホストコンピュータに対する遠隔操作となっているのです。

ユーザ名とパスワードを入力しましょう。login: の後にあなたのユーザ名を入力して下さい。ユーザ名には大文字、小文字の区別があるので注意して下さい。ユーザ名の入力が完了したら、一度リターンキーを押して下さい。次にパスワードを入力します。ユーザ名を入力した時とは違い、パスワードは横から見ている他の人にバレないように、タイプしてもその文字が画面に表示されません。キーを打ち間違えないように注意して入力し、もう一度リターンキーを押して下さい。

```
Unix (EUC) 京都産業大学
ファイル(E) 編集(E) 送出(I) VT機能+(K) VTシフト機能+(I) 設定(S)... ^L^P
UNIX(r) System V Release 4.0 (cc2000)
login: sheryl
Password:
Last login: Wed Aug 3 10:11:38 from c1nf01.kyoto-su.
tcsh: using dumb terminal settings.
Sun Microsystems Inc. SunOS 5.3 Generic September 1993
cc2000(81)% _
```

図 3.65 cc2000 のログインの SAMPLE

図 3.65 のようになりましたか？もし、ならなかった場合は、この後の「困ったときの対処」を参照してください。

3.7.8 ちょっとメッセージ

login すると、ときには利用者宛の通知、おしらせが以下のように画面に表示されるかも知れません。

```
Sun Microsystems Inc. SunOS 5.3 Generic September 1993
4月15日深夜、ファイルサーバである cnfns (cc2000) の
レポートが予定されています。
これにともない cc 環境の全てのマシンの利用が出来なくな
ります。夜間に長時間のジョブを実行する予定のある利用者
はそれまでに実行が終るように注意して下さい。
cc2000(84)% □
```

図 3.66 login 直後のメッセージ

よく読んで、自分に関係があるかどうか確認してください。もしも内容に付いてよく判らないことがあれば計算機センター相談窓口まで連絡するのがいいでしょう。

3.7.9 パスワードを変える

もしもこれがあなたにとって Sign 登録後初めての login ならば、ここでまずパスワードを変更しましょう。Sign 登録時のパスワードは機械的に決められたもので、余り安全とは言えません。必ずパスワードをあなた自身が選んだ単語に変更して下さい。パスワードを決める際には 4 ページの 2.1.3 で既に述べた事に注意してください。

パスワードを変更するには `yppasswd` コマンド¹⁴を利用します。上の節で説明したターミナルを見ると、
cc2000(81)%

などとなっていますね。(但し数字は 81 ではないかもしれません。) ここで `yppasswd` とタイプし、リターンキーを押します。すると以下のような状態に成りますね。(以下はユーザ名 `yasuda` の例。`yasuda` の部分にはあなたのユーザ名が表示されているはずです。)

¹⁴綴りがちょっと変ですが、間違っはけません。

```
cc2000(81)% yppasswd
Changing NIS password for yasuda
Old NIS password:
```

指示通りここでは古い、つまり今先ほどまで使っていたパスワードをタイプし、リターンキーを押します。ここでは login の時と同じくパスワードをタイプしている間、横の人の覗き見によってあなたのパスワードがバレないように、タイプした文字は画面に表示されない事に注意してください。表示されなくともタイプした文字はちゃんと入力されていますから、安心して確実に一文字ずつタイプして最後にリターンキーをタイプして下さい。すると今度は以下のようなプロンプトを表示します。

```
New password:
```

では指示通り新しい、つまり次からこれにしようと言うパスワードをタイプし、リターンキーを押してください。ここでもタイプした文字は表示されません。今度は以下のようなプロンプトを表示します。

```
Retype new password:
```

これはいましがたタイプした新しいパスワードが打ち間違いで無いことを確認するためのものです。もう一度新しいパスワードをタイプしてリターンキーを押してください。今度は以下のような状態になります。

```
NIS passwd changed on ccnic
cc2000(82)%
```

この NIS passwd changed が表示されれば成功です。もしもパスワードが短すぎたりタイプミスがあったりしたらなんらかのエラーメッセージが表示されます。途中でおかしくなったと思う場合は C-c (つまりコントロールキーを押しながら C キーを押す) で cc2000(82)% を表示させます。それからもう一度 yppasswd コマンドをやり直します。

3.7.10 logout をしよう

次は logout です。他にやりたいこともいろいろあるでしょうが、今回は最初の利用でしょうし、まず一通りの操作をやってみると言う意味で logout をこの場でやってみることを勧めます。画面上に cc2000(81)% が表示されていることを確認してください。但し括弧の中の数字は気にしないでね!! 以下のように、logout <Return> と入力してください。

```
cc2000(81)% logout
```

3.7.11 端末ソフトを終了する

logout すると login する前と同じように、再び図 3.63 が表示されます。その一部分を拡大したのが、以下の図 3.67 です。この図の「キャンセル」にマウスポインタを重ね合わせ、マウスの左ボタンをクリック (1 回押し) します。



図 3.67 Host Presenter の終了ボタン

図 3.63 が画面上から消えましたか？消えない場合は、マウスポインタが「キャンセル」の位置に重なっていない事が予想されます。もう一度マウスポインタを「キャンセル」の位置に重ね合わせ、マウスの左ボタンをクリックして下さい。

では次に、画面上にはまだ、図 3.64 に良く似た画面が表示されていると思います。その図の左上部分に注目してください。図 3.68 です。「ファイル」という文字にマウスポインタを重ね合わせ、マウスの左ボタンをクリックしてください。機能の一覧が表示されます。その最下行に「終了」があります。マウスポインタを「終了」に重ね合わせてマウスの左ボタンをクリックしてください。電源を入れた最初の状態(図 3.61 に戻ります。



図 3.68 Host Presenter の終了

3.7.12 電源を切る

Windows を終了させるには、「Windows 終了」という絵 (アイコン) をマウスポインタで二度クリックします。「Ms-Windows を終了します」という確認メッセージが表示されたら、「はい」をクリックします。Windows の終了ができれば、本体電源スイッチの上にオレンジランプが点灯していないのを確認してから、丸い電源ボタンを押してください。その次に、ディスプレイ装置の長四角のボタンを押してください。ディスプレイ装置、本体にグリーンランプの点灯がなければ電源が切れています。お疲れさまでした。

3.7.13 部屋を出る

パソコンの電源が切れたら部屋を出ます。このとき、部屋に誰も残らないようなら空調機、照明、ブレーカーを切って行きましょう。ブレーカーを切るときは、すべてのパソコンに電源が入っていないのを確認してからにしてください。

3.7.14 さて、さて、

最初の利用はこれでおしまいです。大変よくできました。あとは精進して UNIX ユーザへの道を歩まれるわけですね。それには UNIX での一般的なコマンド操作などについて説明している、第 4 章へと進んで下さい。

3.7.15 困った時の対処

ホストコンピュータに接続出来ない

- 図 3.63 が表示されていない場合。図 3.62 をマウスの左ボタンでダブルクリック時に失敗している事が考えられる。再度、図 3.62 をダブルクリックする。
- 図 3.63 は表示されているが、図 3.61 の後ろに隠れてしまっている事が考えられる。「Alt」キー (キーボード左下) を押しながら <ESC> キー (キーボード左上) を押して画面を切替えて探してください。

login ができない

- 図 3.69 が画面に表示されて図 3.64にならない場合がある。メモリの空きが少ないことが考えられる、必要以外のソフトウェアを終了し、再度実行する。

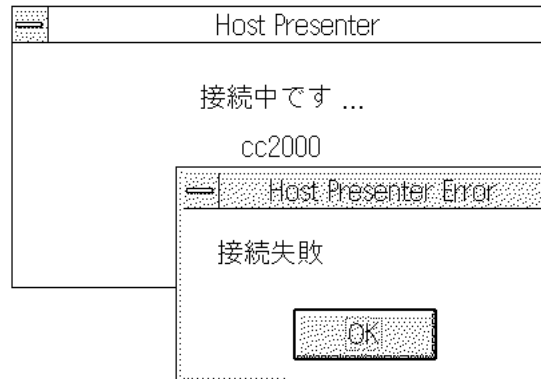


図 3.69 接続失敗の画面

- 図 3.70 のように `login incorrect` が表示される。ユーザー名かパスワードが間違っている事が考えられる。大文字、小文字等の入力間違いがないかを確認、再度入力して下さい。

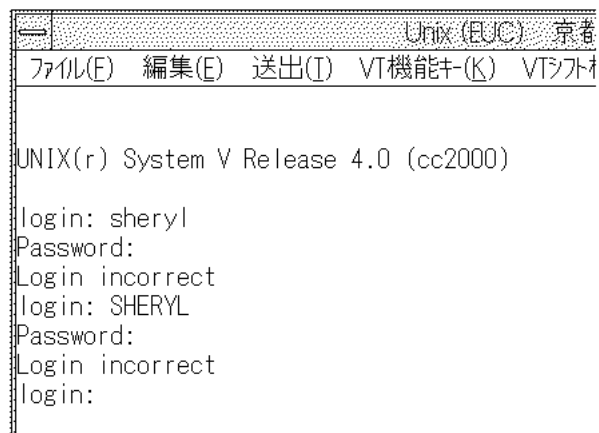


図 3.70 login 失敗の画面

文字が全角になる

- 図 3.71のように、`sheryl` と入力しているのに「しゅ r y l」と画面上に現れる事がある。これはパソコンの日本語変換機能が起動されている事が考えられる。画面右下に「あ自R漢」が表示されているのであれば、「Alt」キー（キーボード左下）を押しながら「半角/全角」キー（キーボード左上）を押して、日本語変換機能を止めてください。画面右下の「あ自R漢」が消えます。

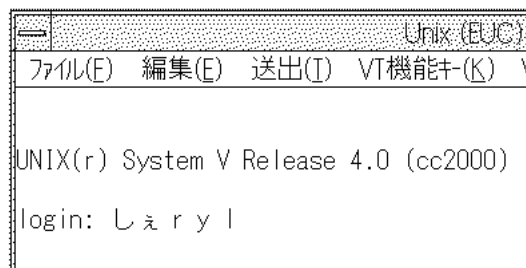


図 3.71 文字が全角になる

3.7.16 マニュアルなど

パソコンのマニュアルは部屋の右側に設置してあるロッカーに入っています。どうぞ自由に取り出してみてください。但し、持ち帰りはいけません!!。禁帯出です。大型汎用機である FACOM のマニュアルはロッカー横の書棚に設置してあります。これも自由にみてください。同じく持ち帰りはいけません。

3.8 DECpc から telnet で cc2000 を使う

DECpc は中央図書館の 1 階読書室内にあるパソコンコーナーに設置してあります。DECpc とは日本 DEC(株) が発売している IBM 製パーソナルコンピュータ (以下パソコン) の互換機です。ここでは中央図書館の 1 階読書室内の DECpc から telnet 環境で cc2000 をホストコンピュータとして利用する手続きについて説明します。中央図書館の 1 階読書室内の DECpc では端末ソフトとしてノベル株式会社の「Host Presenter」を利用します。これは telnet 環境を提供します。DECpc のキーボードでは、一般的にリターンキーと呼ばれるキー¹⁵には「Enter」と刻印されています。ここではこのキーを「リターンキー」と表現します。例示に<Return>と書いてある場合は DECpc ではこの「Enter」キーを押して下さい。

3.8.1 まず部屋へ

この部屋は中央図書館の 1 階読書室内の一角に設けてあり、図書館が開館中であれば誰でも利用が可能です。部屋の管理は図書館が行っていますので、何か椅子、机など部屋の状態及びコンピュータについての質問、要望などがあれば図書館カウンタにお願いします。

¹⁵ キーボード真中やや右の大きなキー

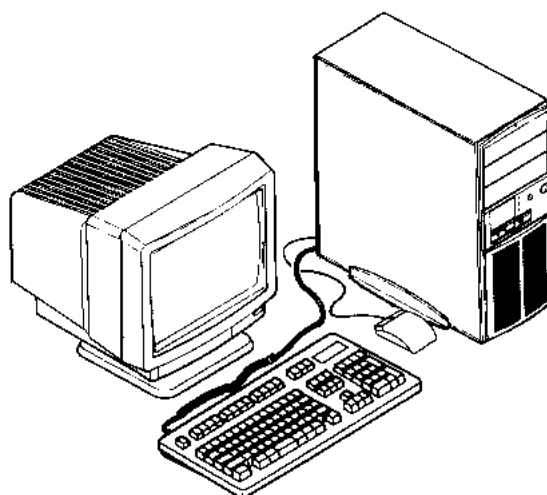


図 3.72 DECpc 本体

3.8.2 電源を入れる

自分が利用するコンピュータを選びます。12 台ありますので空いている場所を見つけて着席してください。その時、ディスプレイ装置 (テレビみたいな物)、本体 (ディスプレイ装置の左横にある箱) にランプが点灯していないか確認してください。もし、ランプが点灯しているのであれば、誰かがすでに利用しているのかもしれないので注意してください。では本体の電源を入れます。本体はディスプレイ装置の左横にある縦長の四角い箱です。本体の電源は、右端真ん中にある丸いボタンです。(図 3.73 参照)

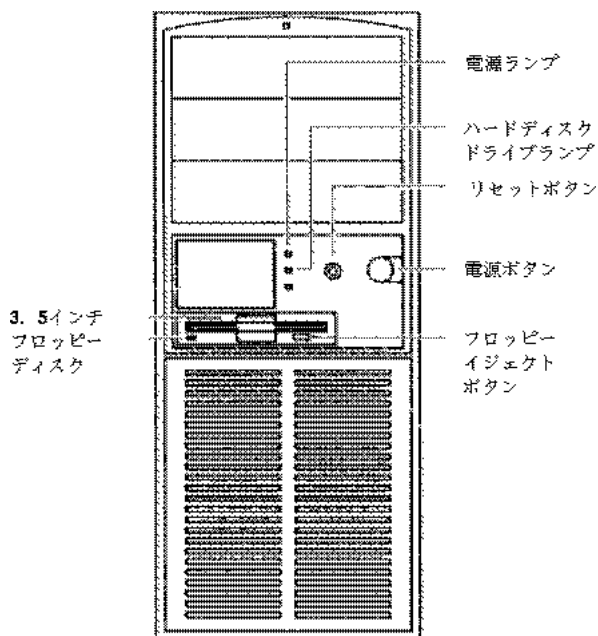


図 3.73 DECpc の電源スイッチ

電源がすでに入っている場合は、丸いボタンの左上にランプが点灯していますので、ランプが消えているのを確認してから、丸い電源ボタンを押してください。ランプの点灯が確認できたら、しばらく (30 秒

ぐらい) 待ってください。ディスプレイ装置に下図のように選択メニューが表示されます。

MS-DOS 6.2 STARTUP MENU

- 1.Windows
 - 2.Windows with DSLINK
 - 3.Windows with Network
- Enter a choice:

この「Enter a choice:」に続いて3をタイプすればマイクロソフト社の Windows が立ち上がります。

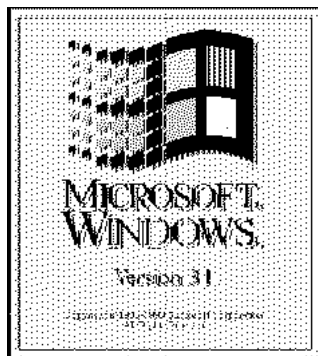


図 3.74 Windows 起動画面

3.8.3 端末ソフトの起動

まず端末ソフトを起動しましょう。画面上に図 3.75が表示されていますね。

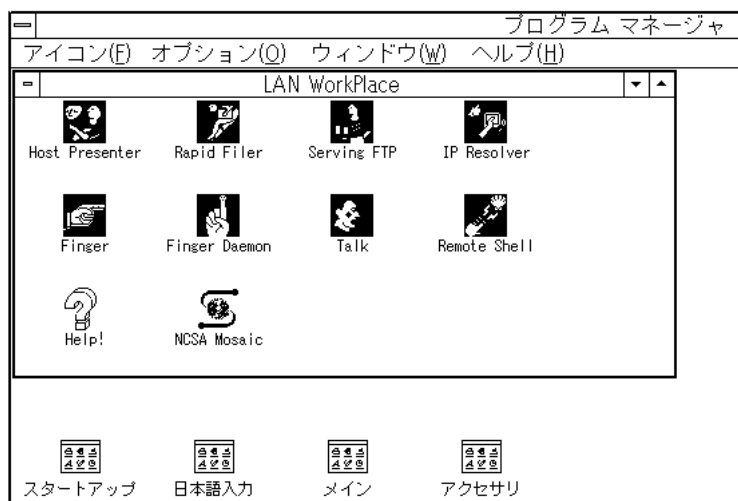


図 3.75 Windows 初期メニュー

図 3.75 の中の一つに「Host Presenter」図 3.76 があります。このアイコン(絵)にマウスポインタ(矢印)を重ね合わせて、マウスの左ボタンをダブルクリック(連続2回押し)してください。Host Presenterが起動されます。



図 3.76 Host Presenter のアイコン

3.8.4 ホストコンピュータへの接続

端末ソフトが起動されたら、まずはどのホストコンピュータの端末になるかを指示します。

Host Presenter が起動されると、図 3.77 が画面に表示されます。もし画面に現れなかったら、あなたはソフトウェアの起動に失敗しています。もう一度図 3.76 にマウスポインタ (矢印) を重ね合わせてマウスの左ボタンをダブルクリックしてください。それでも図 3.77 が現れない場合は図書館のカウンタまで連絡してください。

セッションのオープン

ホスト名(N)
cc2000

セッション名(S)
Unix (EUC) 京都産業大学

オープン(O) キャンセル ヘルプ(H)...

プロフィールのリスト(L)

ログイン先のホストの名前

図 3.77 Host Presenter の初期画面

画面上には、図 3.77 が表示されており、ホスト名として「cc2000」が網掛けの文字として表示されています。(もし、表示されていない場合は、英小文字と数字で「cc2000」と入力してください。) そこでリターンキーを押して下さい。画面に図 3.78 のウィンドウが表示されます。

```
Unix (EUC) 京都産業大学
ファイル(E) 編集(E) 送出(I) VT機能+(K) VTシフト機能+(I) 設定(S)... ^L^P
UNIX(r) System V Release 4.0 (cc2000)
login: _
```

図 3.78 cc2000 のプロンプト

ここで `login:` などと `login` の為の催促文字が表示されていることを確認して下さい。

3.8.5 さあ、login しよう

いよいよ `login` です。ここからはもうあなたのキーボードに対する入力はホストコンピュータに対する遠隔操作となっているのです。

ユーザ名とパスワードを入力しましょう。`login:`の後にあなたのユーザ名を入力して下さい。ユーザ名には大文字、小文字の区別があるので注意して下さい。ユーザ名の入力が完了したら、一度リターンキーを押して下さい。次にパスワードを入力します。ユーザー名を入力した時とは違い、パスワードは横から見ている他の人にバレないように、タイプしてもその文字が画面に表示されません。キーを打ち間違えないように注意して入力し、もう一度リターンキーを押して下さい。

```
Unix (EUC) 京都産業大学
ファイル(E) 編集(E) 送出(I) VT機能+(K) VTシフト機能+(I) 設定(S)... ^L^P
UNIX(r) System V Release 4.0 (cc2000)
login: sheryl
Password:
Last login: Wed Aug 3 10:11:38 from c1nf01.kyoto-su.
tcsh: using dumb terminal settings.
Sun Microsystems Inc. SunOS 5.3 Generic September 1993
cc2000(81)% _
```

図 3.79 cc2000 のログインの SAMPLE

図 3.79 のようになりましたか?もし、ならなかった場合は、この後の「困ったときの対処」を参照してください。

3.8.6 ちょっとメッセージ

`login` すると、ときには利用者宛の通知、おしらせが以下のように画面に表示されるかも知れません。

```

Sun Microsystems Inc. SunOS 5.3 Generic September 1993
4月15日深夜、ファイルサーバである ccnfs (cc2000) の
レポートが予定されています。
これにともない cc 環境の全てのマシンの利用が出来なくな
ります。夜間に長時間のジョブを実行する予定のある利用者
はそれまでに実行が終るように注意して下さい。
cc2000(84)% □

```

図 3.80 login 直後のメッセージ

よく読んで、自分に関係があるかどうか確認してください。もしも内容に付いてよく判らないことがあれば計算機センター相談窓口まで連絡するのがいいでしょう。

3.8.7 パスワードを変える

もしもこれがあなたにとって Sign 登録後初めての login ならば、ここでまずパスワードを変更しましょう。Sign 登録時のパスワードは機械的に決められたもので、余り安全とは言えません。必ずパスワードをあなた自身が選んだ単語に変更して下さい。パスワードを決める際には 4 ページの 2.1.3 で既に述べた事に注意してください。

パスワードを変更するには `yppasswd` コマンド¹⁶を利用します。上の節で説明したターミナルを見ると、

```
cc2000(81)%
```

などとなっていますね。(但し数字は 81 ではないかもしれません。) ここで `yppasswd` とタイプし、リターンキーを押します。すると以下のような状態に成りますね。(以下はユーザ名 `yasuda` の例。 `yasuda` の部分にはあなたのユーザ名が表示されているはずです。)

```

cc2000(81)% yppasswd
Changing NIS password for yasuda
Old NIS password:

```

指示通りここでは古い、つまり今先ほどまで使っていたパスワードをタイプし、リターンキーを押します。ここでは login の時と同じくパスワードをタイプしている間、横の人の覗き見によってあなたのパスワードがバレないように、タイプした文字は画面に表示されない事に注意してください。表示されなくてもタイプした文字はちゃんと入力されていますから、安心して確実に一文字づつタイプして最後にリターンキーをタイプして下さい。すると今度は以下のようなプロンプトを表示します。

```
New password:
```

では指示通り新しい、つまり次からこれにしようと言うパスワードをタイプし、リターンキーを押してください。ここでもタイプした文字は表示されません。今度は以下のようなプロンプトを表示します。

```
Retype new password:
```

これはいましがたタイプした新しいパスワードが打ち間違いで無いことを確認するためのものです。もう一度新しいパスワードをタイプしてリターンキーを押してください。今度は以下のような状態になります。

```

NIS passwd changed on ccnic
cc2000(82)%

```

¹⁶ 綴りがちょっと変ですが、間違っってはいません。

この NIS passwd changed が表示されれば成功です。もしもパスワードが短すぎたりタイプミスがあったりしたらなんらかのエラーメッセージが表示されます。途中でおかしくなったと思う場合は C-c (つまりコントロールキーを押しながらCキーを押す) で cc2000(82)% を表示させます。それからもう一度 yppasswd コマンドをやり直します。

3.8.8 logout をしよう

次は logout です。他にやりたいこともいろいろあるでしょうが、今回は最初の利用でしょうし、まず一通りの操作をやってみると言う意味で logout をこの場でやってみることを勧めます。画面上に cc2000(81)% が表示されていることを確認してください。但し括弧の中の数字は気にしないでね!! 以下のように、logout <Return>と入力してください。

```
cc2000(81)% logout
```

3.8.9 端末ソフトを終了する

logout すると login する前と同じように、再び図 3.77 が表示されます。その一部分を拡大したのが、以下の図 3.81 です。この図の「キャンセル」にマウスポインタを重ね合わせ、マウスの左ボタンをクリック (1 回押し) します。

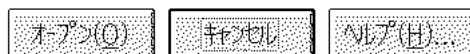


図 3.81 Host Presenter の終了ボタン

図 3.77 が画面上から消えましたか?消えない場合は、マウスポインタが「キャンセル」の位置に重なっていない事が予想されます。もう一度マウスポインタを「キャンセル」の位置に重ね合わせ、マウスの左ボタンをクリックして下さい。

では次に、画面上にはまだ、図 3.78 に良く似た画面が表示されていると思います。その図の左上部分に注目してください。図 3.82 です。「ファイル」という文字にマウスポインタを重ね合わせ、マウスの左ボタンをクリックしてください。機能の一覧が表示されます。その最下行に「終了」があります。マウスポインタを「終了」に重ね合わせてマウスの左ボタンをクリックしてください。電源を入れた最初の状態 (図 3.75 に戻ります)。



図 3.82 Host Presenter の終了

DECpc から telnet 端末環境で cc2000 を利用する場合に必要な手続きの説明は以上です。もちろんこの後パソコンの利用を終了する場合は、正しく決められた手続きで Windows を終了し電源を切って下さい。間違ってもこの段階でいきなり電源スイッチを切ったりしてはいけません。

3.8.10 電源を切る

Windows を終了させるには、まず図 3.75の状態であることを確認してください。画面左上の「アイコン(F)」にマウスポインタを合わせ一度クリックすると、機能一覧メニューが表示されます。機能一覧メニューの最下行に「WINDOWS の終了(x)」があるので、マウスポインタを合わせマウスの左ボタンをクリックしてしてください。「Windows を終了します」という確認メッセージが表示されたら、マウスポインタを「OK」に合わせて左ボタンをクリックします。Windows の終了ができれば、画面上に E:¥ が表示されます。この状態になった時にしばらく(約 10 秒程度)時間をおいて本体電源ボタンを押してください。ディスプレイ装置、本体にランプの点灯がなければ電源が切れています。お疲れさまでした。

3.8.11 部屋を出る

パソコンの電源が切れたら部屋を出ます。

3.8.12 さて、さて、

最初の利用はこれでおしまいです。大変よくできました。あとは精進して UNIX ユーザへの道を歩まれるわけですね。それには UNIX での一般的なコマンド操作などについて説明している、第 4 章へと進んで下さい。

3.8.13 困った時の対処

ホストコンピュータに接続出来ない

- 図 3.77が表示されていない場合。図 3.76をマウスの左ボタンでダブルクリック時に失敗している事が考えられる。再度、図 3.76 をダブルクリックする。
- 図 3.77は表示されているが、図 3.75 の後ろに隠れてしまっている事が考えられる。「Alt」キー(キーボード左下)を押しながら<ESC>キー(キーボード左上)を押して画面を切替えて探してください。

login ができない

- 図 3.83 が画面に表示されて図 3.78にならない場合がある。メモリの空きが少ないことが考えられる、必要以外のソフトウェアを終了し、再度実行する。

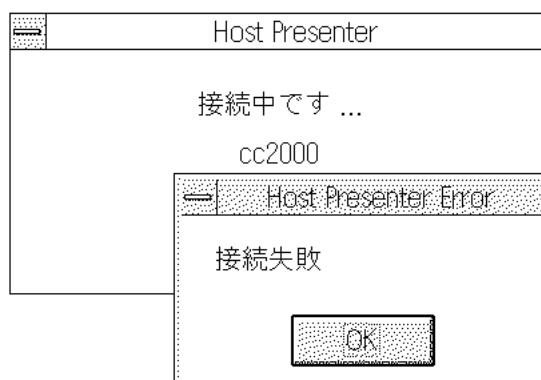


図 3.83 接続失敗の画面

- 図 3.84 のように `login incorrect` が表示される。ユーザー名かパスワードが間違っている事が考えられる。大文字、小文字等の入力間違いがないかを確認、再度入力して下さい。

```

Unix (EUC) 京巻
ファイル(F) 編集(E) 送出(I) VT機能+(K) VTシフト

UNIX(r) System V Release 4.0 (cc2000)

login: sheryl
Password:
Login incorrect
login: SHERYL
Password:
Login incorrect
login:

```

図 3.84 login 失敗の画面

文字が全角になる

- 図 3.85 のように、`sheryl` と入力しているのに「しえ r y l」と画面上に現れる事がある。これはパソコンの日本語変換機能が起動されている事が考えられる。画面右下に「あ自R漢」が表示されているのであれば、「Alt」キー（キーボード左下）を押しながら「半角/全角」キー（キーボード左上）を押して、日本語変換機能を止めてください。画面右下の「あ自R漢」が消えます。

```

Unix (EUC)
ファイル(F) 編集(E) 送出(I) VT機能+(K) \

UNIX(r) System V Release 4.0 (cc2000)

login: しえ r y l

```

図 3.85 文字が全角になる

3.8.14 マニュアルなど

パソコンのマニュアルは部屋内の本棚に入っています。どうぞ自由に取り出してみてください。但し、持ち帰りはいけません!!。禁帯出です。

第 4 章

UNIX それから

ここでは計算機センターが管理している UNIX 環境を例に取りながら、比較的一般的な UNIX 環境の利用方法を説明します。ここでの説明は網羅的なものではなく、部分を取り上げて曖昧に説明しています。これは本文の読者のコンピュータそのものについての知識のハードルを高くしすぎないためで済ませるためです。読者が UNIX、つまりコンピュータの利用に慣れて行くにしたがって自分でマニュアル、書籍、ネットワークなどから情報を常に取り込んで理解することが大切です。

4.1 基礎知識をもう一度

4.1.1 login

UNIX コンピュータを利用する為には、初めに誰がこの UNIX コンピュータをこれから利用するのかを、その都度確認する login と呼ばれる作業をしなければなりません。本文では既に login が済んでいるものとして説明を続けます。また、ターミナルが起動され、コマンドを入力できる状態になっていることを前提にしています。機種ごとの login の方法、ターミナルの起動の方法については第 2 章の「UNIX はいかが？」をご覧ください。

ところで login するのは良いのですが、ときどき logout しない（もしくはセッションを終了しない）人がいます。logout しないと他の人がそのコンピュータを使えないばかりか、逆に他人に悪用されてしまいますから、しばらくしてまた戻ってくる場合でも席を外す際は logout するべきです。

4.1.2 キー表記

キーボードから様々な文字列を打ち込むことを「タイピング」と呼んでいます。「abc」と打ち込むことを「abc とタイプする」「abc と入力する」などと表現することもあります。UNIX では通常のアルファベットや数字以外に、様々なキーがあります。例えばアルファベットの A と書かれたキーを押せば、小文字の「a」が入力されるでしょう。もしも大文字の「A」をタイプしたければ「シフトキー」を押しながら A キーを押すこととなります。これらのことは既に第 2 章の 2.4 で説明しました。ですからこれ以降は「aBc」とタイプする、と書けば「アルファベットの A キーを押して、シフトキーを押しながら B キーを押して、C キーを押す」というように解釈してください。同様に、以下のような記述で様々なタイピングを表現します。

<Return>	リターンキーを意味します。
<Delete>	削除キーを意味します。一般的には Delete キーに当たります。
<Space>	空白キーを意味します。
<Tab>	タブキーを意味します。一般的にはアルファベットの Q の左にあります。
<ESC>	エスケープキーを意味します。一般的には数字の 1 キーの左にあります。
<ESC> X	エスケープキーを押してから X キーを押すことを意味します。
<Control> X	コントロールキーを押しながら X キーを押すことを意味します。
C-X	コントロールキーを押しながら X キーを押すことを意味します。
M-X	メタキー（一般的には Escape キーに同じ）を押してから X キーを押すことを意味します。

4.1.3 カーソル

タイピングしている最中に、次にキーを押したらどこにその文字が入力されて表示されるかを判り易くするために表示する目印のことを「カーソル (cursor)」と呼んでいます。形は状況に応じて様々で、丁度一文字分の大きさの■（黒い長方形）だったり、|（縦棒）だったりします。他の文字と見分けが付き易いように点滅している場合もあります。

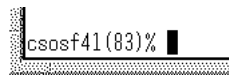


図 4.1 カーソルの例（黒い長方形）

4.2 コマンド

4.2.1 コマンドって何だ？

コンピュータと言うのは機能の集合体だと考える事が出来ます。これからあなたが利用しようと思っ
ているコンピュータの中には星の数ほどの（いやそんなに無いな、山ほどの）機能があり、その中から利用
者は自分の希望の機能を選んで実行させるのです。山ほどの機能にはそれぞれ固有の名前が付けられてお
り、利用者は機能その名前を選んで言うわけです。逆に言うと機能の名前を知らなければどんなに便利
な機能がコンピュータの中にあつたとしても利用できませんし、名前を間違えて指定すれば、望みの機能
とは違う機能が働き出します。この「機能にそれぞれ付けられている固有の名前」をコマンドと呼んでい
ます。山ほどあるコマンドを利用者が覚え易いように、その機能を連想し易い名前がコマンド名として付
けられています。

あるコマンドを入力するとは、そのコマンドに対応する機能を実行するように指示するという事です。

4.2.2 プロンプト

コンピュータの利用と言うのは実際にはこのコマンド実行の繰返しだと言えます。コンピュータが「コ
マンドをどうぞ」と言うメッセージを表示し、利用者がコマンドを入力する。コンピュータはコマンドの
結果をメッセージとして表示し、「次のコマンドをどうぞ」と、またメッセージを表示する。そして利用者
が再びコマンドを入力する、という具合です。見方を変えればコンピュータと利用者はメッセージとコマ
ンドで「対話」しているようにも見えます。このコンピュータが利用者との対話のために用いている「次の
コマンドは何ですか？」という催促メッセージのことを一般に「プロンプト (prompt)」と呼んでいます。

cc 環境での一般的なプロンプトは `csosf01(81)%` などと表示されます。DEC-3300 で login した直後
に見える DECterm ウィンドウの左隅に見つけることが出来るでしょう。

4.2.3 簡単なコマンド

date コマンド

ここで一つ非常に簡単なコマンドを実行してみましょう。date コマンドです。プロンプトに続いて `date`
<Return>とタイプしてください。以下のように時間が表示されるでしょう。つまり `date` は日付と時間を
表示するという機能のコマンドです。

以下に示す例は実際にあなたが試してみたときとは結果が違うでしょう。この例を試したときと、今と
では状況が違っているからです。これ以降に挙げる例も、全く同じ結果が表示されるとは限らないことに
注意してください。

```
csosf01(81)% date
1994年02月27日(日)20時05分03秒
csosf01(82)%
```

もしも `date` とタイプしたのにその通りに入力できない様な場合は 第2章の 2.5.5 をチェックして下さい。

もしも以下の様に `Command not found` などというメッセージが表示されてしまった場合はコマンドの
打ち間違いです。もう一度、今度は正確にコマンドをタイプして下さい。

```
csosf01(81)% data
data: Command not found
csosf01(82)%
```

この正しくない操作を利用者が実行しようとしたためにコンピュータの方から「それは駄目だったよ」という意味のメッセージが返ってくる時があります。このようなメッセージを一般に「エラーメッセージ」と呼んでいます。

hostname コマンド

`hostname` コマンドで自分が使っているコンピュータのホスト名を得られます。

```
csosf01(82)% hostname
csosf01
csosf01(83)%
```

上記の例から分かるように、`cc` 環境ではホスト名はプロンプトに含まれていつでも表示されています。

4.2.4 引数とオプションのあるコマンド

finger コマンド

`finger` コマンドで、利用者の情報を得ることが出来ます。`finger` に続いて調べたいキーワードをタイプして<Return>です。ここでは `yasuda` さんについての情報を表示させてみましょう。

```
csosf01(41)% finger yasuda
Login name: yasuda                In real life: Yutaka Yasuda
Directory: /NF/home/syokuin0/yasuda  Shell: /bin/csh
On since Mar  6 21:04:04
    on ttyp2 from ds10.kyoto-su.ac.jp
No Plan.

Login name: hayato                In real life: YASUDA Hayato
Directory: /NF/home/g930/hayato      Shell: /bin/csh
Never logged in.
No Plan.
csosf01(42)%
```

例から判るように二件の情報が表示されました。一つはユーザ名 `yasuda` さんのもので、もう一つはユーザ名 `hayato` さんのものです。`hayato` さんはユーザ名ではなくフルネームの部分に指定したキーワードがマッチしたから表示されたのですね。

`finger` ではコマンドに続いてキーワードを指定しましたね。この様にコマンドの後ろにそのコマンドが実行するために必要な情報を付ける場合があります。このコマンドの後ろに付けるものを「引数 (ひきすう)」と呼んでいます。パラメータなどと呼ぶ場合もあります。引数は常にコマンドの後ろに書きます。逆に言うとコマンドが必ず一番前に来ます。

また、コマンドの処理内容を少し変える指示を与えることが出来る場合もあります。例えば `finger` コマンドの表示は少し長いので、これを短くするために `-s` という指示をコマンドの後ろ、キーワードの前に与えます。

```
csosf01(43)% finger -s yasuda
```

```

Login      Name           TTY Idle   When           Office
yasuda     Yutaka Yasuda   p2         Sun 22:29
hayato     YASUDA Hayato   < . . . . >
csosf01(44)%

```

しかしやはりユーザ名 yasuda さんの情報だけが見たいと思っています。finger コマンドにはキーワードのマッチングをユーザ名だけで行う指示を与えることも出来ます。先ほどの -s という指示に今度は -m という指示を加えます。

```

csosf01(44)% finger -s -m yasuda
Login      Name           TTY Idle   When           Office
yasuda     Yutaka Yasuda   p2         Sun 22:29
csosf01(45)%

```

なるほどユーザ名 hayato さんの情報が表示されなくなりましたね。

この -s や -m などのように、コマンドの処理内容を少し変化させるような指示を「オプション」もしくは「コマンドオプション」と呼んでいます。オプションは常にコマンドの後ろに書きます。逆に言うとコマンドが必ず一番前に来て、それからオプション、その後その他の引数という並びになります。オプションは大抵 - 記号（引き算記号、ハイフン）に導かれています。また、オプションが複数並ぶ場合については -s -m と並べて書くことも出来ますし、縮めて -sm と書くことも出来る場合があります。つまり上記の例だと finger -sm yasuda と書いても同じ意味です。

4.2.5 対話的なコマンドとそのサブコマンド

bc コマンド

bc コマンドで、簡単な四則演算電卓を利用することが出来ます。bc <Return>でカーソルが画面の左端で止まります。そこで四則演算の式を一行入力して<Return>すれば、すぐ次の行に結果が表示されます。プロンプトは特に表示されませんが、そこでまた式を入力すればまたその結果が表示されますから計算は何度でも繰り返して行えます。

```

csosf01(65)% bc
1 + 2 * ( 3 + 4 )
15

```

この状態では、コンピュータは四則演算の式を計算する bc コマンドの最中で、それ以外の利用者の指示は受け付けられません。式以外、例えば date のような通常のコマンドをこの状態でタイプしても以下のようにエラーメッセージを表示するだけです。

```

date
syntax error on line 1 (null)

```

計算を止めてコマンドプロンプトの状態に戻りたければ quit <Return>です。

```

quit
csosf01(66)%

```

bc などのように、コマンドの中には実行すると更にユーザからの指示を待つような（つまり「対話的な」）動きをするものもあります。bc における quit などのようにコマンドの処理中にユーザがそのコマンドに対して更に細かい指示を与えるコマンドの事を「サブコマンド」と呼んだりします。

4.2.6 コマンドの使い方を調べる

さて、コマンドと一口に言っても、どうやらその機能と名前を覚えるだけでは済みそうにないことが判りますね。コマンドをどれか一つ取ってみても、それぞれ独自のオプション、引数、もしくはサブコマンドなどがあり、どのコマンドにどんなオプション等があったかを覚えないと実際困りそうです。でもそんなものいつでも全部覚えておけるほどみんな暇ではありません。その為にコマンドの正確な機能、使い方を教えてくれるコマンドがあります。man¹コマンドです。例えば先ほど取り上げた `finger` コマンドのオプションなどを忘れてしまった場合にはすかさず `man finger <Return>` です。

```
csosf01(79)% man finger
finger(1)                                     finger(1)
NAME
  finger, f - Displays user information
SYNOPSIS
  finger [-bfhilmqpsw] [user ...]
```

The `finger` command displays information about the users in the `passwd` file.

というわけです。このマニュアル表示は大抵一画面では収まり切りませんから、一ページごとに一旦停止します。<Space>で一ページ分送ります。<Return>で一行分送ります。この表示が送られて行く様子を「スクロール」と呼んでいます。全てを表示し切ったら `man` コマンドは終了します。一旦停止している状態で、`q` キーを押すと最後まで表示せずにその状態で `man` コマンドの表示を終了させることが出来ます。`h` キーか、`?` キーのいずれかを押せば、どのようなキーを押せば一旦停止後にどのような操作が効くかを表示してくれます²。

man によるマニュアルの書式

`man` コマンドが表示する内容を良く見てみると、幾つかの項目に分けて説明してあるのが判るでしょう。`finger` の場合だと **NAME**, **SYNOPSIS**, **DESCRIPTION**, **FLAGS**, **EXAMPLES**, **FILES**, **RELATED INFORMATION** という具合です。上記の項目分けは、全ての UNIX において全く同じではなく、幾らか違う部分もありますが大抵は共通の形式で書かれています。以下にその項目の意味を書いておきます。

- **NAME**, 名前

コマンドの名前とその概要。

- **SYNOPSIS**, 形式

コマンドの形式。コマンドに適用できるオプション、引数などを列挙します。

ここでの表記にはルールがあって、例えば `finger` の例を以下に挙げると、

```
finger [-bfhilmqpsw] [user ...]
```

この [] に囲まれた部分は「なくてもいいよ」つまり省略可能だという事を意味しています。特に上記の例のように [] に多くのオプション文字がくくられていた場合は、その中のどれでもピックアップして同時に与えていいよ、という意味です。また、... は、その直前のものを繰り返して書いてもいいよという意味です。つまり `finger` ではキーワードを複数書けるのですね。コマンドにオプションや引数をつけてタイプする場合、ここに表示された順番に注意して下さい。

¹ 「マニュアル (manual)」 のつもりで `man` です。

² つまりこれらは「表示一旦停止機能」のサブコマンドと言うわけです。

- **FLAG, OPTION, フラグ, オプション**

それぞれのオプションの働きについて詳しく書いてあります。あるオプションとは相反する指示だから、これとこのオプションは同時に指定してはいけないよ、などということも書いてあります³。

- **DESCRIPTION, 機能説明**

コマンドの詳細説明。コマンドの機能が詳しく書かれています。ここにオプションの説明を含めている UNIX もあります。

- **FILES, ファイル**

コマンドに関するファイル⁴の名前が列挙されます。

- **SEE ALSO, RELATED INFORMATION, 関連項目**

コマンドに関連する項目。深い関係のあるコマンドなどが列挙されます。この項目は再び `man` コマンドで参照できますから、このコマンドのマニュアルだけを読んで良く判らない場合はここを追い掛けて行くのがお勧めです。

- **BUGS, バグ**

コマンド使用上の制限事項。コマンドの動きがどうもおかしいという時は注意してみましょう。

コマンド名を調べる

コマンドの名前が判らないのだけれど、このような機能を持ったコマンドを探したい、と言う時にも `man` コマンドは有効です。キーワードでコマンドを検索するオプションとして `-k` オプションがあります。`man -k password` などとすれば `password` に関係のあるコマンドなどの一覧が表示されます。結構沢山出ますが一行で一つのコマンドを紹介してくれています。一番目の項目がコマンド名、その次の括弧に囲まれた数字がマニュアルの分類番号で、残りがコメントです。

```
csosf01(86)% man -k password
conflict (8)          - search for alias/password conflicts
lock (1)             - Requests and verifies a user password
passwd (4)           - Password files
passwd, chfn, chsh (1) - Changes password file information
popwrd (8)           - Sets password for a POP subscriber
printpw (8)          - Outputs the contents of the password database
pwck, grpck (8)      - Checks the password and group files for inconsistencies
yppasswd (1)         - change password in Network Information Service (NIS)
yppasswdd, rpc.yppasswdd (8) - server daemon for modifying the Network Information Service (NIS) password file
csosf01(87)%
```

このような感じですね。上記の括弧に囲まれたマニュアルの分類番号の一覧を以下に示しておきます。この分類のことをセクションと呼んでいます。

³ こういうのを排他的なオプションなどと表現することもあります。

⁴ 後述。4.4 参照

セクション番号	分類
1	ユーザコマンド（一般利用者の為のコマンド）
2	システムコール（プログラム言語から利用します）
3	関数（プログラム言語から利用します）
4,5	各種ファイルフォーマット
6	ゲームとデモ
8	保守用コマンド（システム管理者が利用します）

ところでこのセクション番号の割当てですが、UNIX によってちょっと違いがあります。セクション 1, 2, 3 位まではどの機種でも同じなのですが、4, 5 あたりについては上記の表は余り当てにならないことに注意して下さい。

さて、例に挙げた password キーワードでのマニュアル検索ではセクション 1 とセクション 4 と両方に passwd という項目がありましたね。ここで `man passwd` とすると常に前の方だけ、つまりセクション 1 の方についてだけが表示されます。このような状況でセクション 4 の passwd について知りたい場合は、`man` コマンドでセクション番号を明示してやります。ここでちょっと気にしなければいけないのはセクション番号の指定の仕方が UNIX によってまちまちだと言うことです。以下に cc 環境の代表的な機種である DEC-3300 (OSF/1) の場合と SPARCcenter2000(Solaris2) の場合をそれぞれ示します。

機種	OS 名称	コマンド記述
DEC-3300	OSF/1	<code>man 4 passwd</code>
SPARCcenter2000	Solaris2	<code>man -s 4 passwd</code>

4.2.7 UNIX によるコマンドの違い

UNIX は一種類ではなく各メーカーからたくさんの種類の UNIX コンピュータが出荷されています。UNIX はそれぞれのメーカーで独自に改造され、コマンドも少しづつ動きが違います。違う UNIX を採用している場合には、コマンド名は同じでもオプションが違うと言う事は良くあることです。先述の `man` コマンドもその例です。

cc 環境は DEC-3300(OSF/1) , SPARCcenter2000(Solaris2) , NeXTStation(NEXTSTEP) の 3 種類の UNIX マシンの混成です。このドキュメントは一般的な UNIX なら大抵あてはまるようにして書いていますが、今後例示されるコマンドのうちいくつかはその通りでは動作しない可能性があります。おかしいな、と思ったらすぐ `man` コマンドで確認する習慣を身につけましょう。

4.2.8 トラブルからの脱出

UNIX を操作していて、どうにもおかしな状態になってしまって困ることがあります。もう一度始めからやり直したいんだけど、今どういう状況なのか良く判らないなあどうしようどうしよう、、

とにかくコマンドを中断して最初からやり直したい

C-c を試してください。何度か C-c するとうまくプロンプトに戻る場合があります。

でもキー入力が全然受け付けられていないようなんですけど

C-q を試してください。C-s でキー入力をロックしてしまう場合が時々あります。C-s の解除が C-q です。C-q の後でならキー入力が効くのでは無いでしょうか。

正確な表現をすると C-s でロックされているのはキー入力では無く、画面表示です。つまりキー入力は受け付けられているのだけれども、その結果の表示がロックされているので、あたかもキー入力効いていないかのように見えるのです。注意して見れば C-q の後に今までタイプしていた分と、その結果表示が一気に画面に表示されるはずですが。

それでも C-c が効いてないようなんです

ええい、仕方がありません。C-z を試してください。それで `Suspended` などと表示されてシェルのプロンプトが表示されたらしめたものです。その状態ですぐさま `kill %%` とやってください。そのあとは普通にコマンドが打ち込める状態になっているのでは無いでしょうか。

画面表示がどうにもおかしくなっているんです

Mule 利用中であれば C-l を試してください。<Control>と英字の L です。それ以外のコマンドなどで画面が乱れている場合はとりあえず C-c などしてシェルのプロンプトまで戻り、そこで `tset` コマンドを試してください。

それで戻らなければ `login` し直すのが早いですね。

でもやっぱりどうにもならないんです

仕方がありません。計算機センター相談窓口まで連絡下さい。

4.3 シェル

今までコマンドの例を挙げてきました。その時、コンピュータそれ自体が利用者からのコマンドを受け付けて逐一実行してくれているように書いてきましたが、じつはそれは正確な表現ではありません。いまままで「`csosf01(81)%`」などのプロンプトを表示して、利用者からのコマンドを受け付けてくれていたのは「シェル」と呼ばれるプログラムだったのです。

コンピュータはいろいろなプログラムを実行できます。逆に言うと、コンピュータが実行できるのはプログラムだけです。利用者のキーボード入力からコマンドを受け付けてそれに対応するプログラム⁵を実行するための仕掛けがシェルなのです。勿論シェルもコンピュータの中ではプログラムで実現されていますから、シェルプログラムなどとも呼ばれます。とにかく利用者がキーボードからのコマンドでコンピュータを操作するときに、利用者とコンピュータを仲介してくれるプログラムなのだと考えてください。

参考

シェルは一種類ではなく、何種類もあります。一般的に UNIX でシェルと言えは `csh`⁶ がポピュラーなのですが、`cc` 環境では `tcsh`⁷ と呼ばれるシェルを標準的に採用しています。`tcsh` は `csh` の拡張版のようなもので、基本的な動作は `csh` と同じだと考えてください。世間一般で売られている `csh` のための書籍に書いてある事項は殆どそのまま `tcsh` にも適用できますので、安心して `csh` の参考書を使って貰って結構です。

4.3.1 コマンド入力時の編集

シェルには便利な機能があって、コマンドの入力時に利用者のタイピングを手助けしてくれたりします。今までタイピングで間違えたときには、`<Delete>` で一文字ずつ戻って間違えたところから打ち直してくれと書いてきました。確かにこれが確実な方法なのですが、しかし例えば `fonger -sm yasuda` とタイプし終わったところで「あっ、`fonger` が `fonger` になっている！」と気が付いた場合、最初の `f` 以外全て打ち直しになって悲しい思いをすることになります⁸。

しかし安心してください。ここでカーソルキーを紹介しましょう。カーソルキーとはリターンキーの少し右辺りに配置してある矢印キーのことです。カーソルについては既に 4.1.3 で説明しましたが、このカーソルが左矢印 (`←`) を一つ押すことによって一つ左に移動します。

上記の悲しい `fonger` の例だと、左矢印キーを十数回押して `n` までたどり着き、そこで `<Delete>` を押して `o` を消し、それから `i` を押します。これでめでたく `fonger -sm yasuda` が完成しましたね。ここで元気良く `<Return>` とすれば完成したコマンドで実行が行なわれます。

カーソルは右にも動きます。例えば今まさに `fonger` に直したのに、今回は `yasuda` さんではなく `yasuo` さんを検索する筈だった事に気が付いたとします。今度は右矢印キー (`→`) を押して `yasuda` の最後の `a` までたどり着き、そこで `<Delete><Delete>` で `da` を削除します。後は `o` をタイプして `<Return>` です。

カーソルを左右に動かしたりする編集キーには他にも幾つかあります。以下に一覧を載せておきます。

⁵つまりそれがコマンドの本体なのですね。

⁶「しーしえる」と読んでください。

⁷「ていしーしえる」と読んでください。

⁸特にこのような打ち間違いをするのは初心者の方が多く、タイピングに慣れていない初心者にとっては非常に悲しいものです。

キー	アクション
左矢印 (←)	一文字分左へ
C-b	左矢印に同じ
右矢印 (→)	一文字分右へ
C-f	右矢印に同じ
C-a	コマンド行先頭 (左端) へ
C-e	コマンド行末尾 (右端) へ
<Delete>	カーソル位置の直前の一文字を消去
C-k	カーソル位置から末尾までを消去
C-u	コマンド行全てを消去

これでもう悲しい思いをせずに済みますね。

4.3.2 ヒストリ

コマンドを何度もタイプしていると、非常に良く似たコマンドを何度も繰り返したり場合によっては何度も同じコマンドを繰り返している事に気が付くでしょう。幾つか前に打ち込んだあのコマンドをもう一度!と思うこともあるでしょう。もっと悲しい場面としては非常に苦労してタイプした長いコマンドが、実はタイプミスを含んでいてもう一度全部タイプし直さなくてはならない、と言う場合です。

しかし安心してください。そんな時に便利なのがヒストリ (history) 機能です。一つ前のコマンドを呼び戻すにはカーソルキーの上矢印 (↑) キーを一度押します。そこで現れた一つ前のコマンドも、左矢印キーや右矢印キーでカーソルを移動させながら編集し直して実行することが出来ます。二つ前のコマンドを呼び戻すには上矢印キーをもう一度 (つまり二度) 押します。上矢印を押しすぎて、目標のコマンドより戻りすぎた場合は下矢印を押します。

キー	アクション
上矢印 (↑)	一つ前のコマンド
C-p	上矢印に同じ
下矢印 (↓)	一つ次のコマンド
C-f	下矢印に同じ

ここで `history` コマンドを試してください。一体どれだけのコマンドを覚えているかが判ると思います。上下の矢印キーによって、このヒストリの中を上下することが出来ると言うわけです。

```
csosf01(86)% history
  7 21:20 goto label
  8 21:20 w | grep tubo
..... (中略)
 82 21:48 finger yasuda
 83 21:48 date
 84 21:48 finger -m yasuda
 85 21:48 finger -sm yasuda
 86 21:48 history
csosf01(87)%
```

これでまた一つ悲しい思いをせずに済むようになりましたね。

4.3.3 イベント

また、`history` のリストを見ると、左に番号があるのが判ります。この番号はプロンプトの括弧の中に出てくる数字に一致しています。この番号で、何番目をもう一度実行せよ、という指示も出来ます。例えば 84 番目のコマンドを（もしくは 84 番目のイベントを）もう一度実行したいと言うときは `!84` です。

```
csosf01(88)% !84
finger -m yasuda
Login name: yasuda                In real life: Yutaka Yasuda
Directory: /NF/home/syokuin0/yasuda  Shell: /usr/local/bin/tcsh
On since Mar  8 21:47:43 on pts/2 from ds10.kyoto-su.ac.jp
No unread mail
No Plan.
csosf01(89)%
```

84 番目のコマンド `finger -m yasuda` が念のために `!84` のすぐ次の行に表示されているのが判りますね。

また番号ではなく、一番最近に実行した `○×△` で始まるコマンドをもう一度実行するという指示も可能です。上記の例の状態、`!d` とすると 83 番目の `date` にマッチして、それが実行されます。

```
csosf01(89)% !d
date
1994年02月28日(月)19時08分25秒
csosf01(90)%
```

83 番目のコマンド `date` が念のために `!d` のすぐ次の行に表示されているのが判りますね。コマンド名のマッチングの為に `!` に続けて指定する文字は何文字でも構いません。書いた文字数の分だけでマッチングし、一番最近のコマンドから逆にさかのぼって一番最初にヒットした（適合した）コマンドが実行されます。もしもさかのぼってヒットするコマンドがなかった場合は「そんなイベントはない」という意味のエラーメッセージが表示されます。実際「イベント」という、この機能にまつわる名前を見るのはこのとき位です。

これで随分幸せになりましたね。（^_^）

4.4 ファイル

ファイルとは何かと言うことを本質的に理解できるようになるにはコンピュータの構造を知るところから始めなくてはなりません。これはファイルがコンピュータが発展する過程での歴史的経緯から発生し、現在に至っているからです⁹。

ここではファイルとは何か、と言う事については余り言及せず、その仕掛けを利用するとこんな事が出来る、というところを説明します。

4.4.1 でもやっぱりファイルって何？

気になる人の為に少し中途半端なものになるのを覚悟で「ファイル」とは何か説明しましょう。気にならない人は読み飛ばして下さい。

身のまわりの電化製品を見渡すと、ディスクやテープなどの「記録媒体」などと呼ばれるものを容易に見つけることが出来ると思います。例えば音楽を聞くのに使っているCD（コンパクトディスク）や、ビデオテープなどです。前者はキラキラ光る面に溝を刻んで、後者は茶色の鉄粉が塗ってあるテープに磁石で印を付けながら、とにかく「なにか」を記録していきます。UNIX コンピュータにもこれと同じように「なにか」を記録できる「ディスク」が幾つも付いています。きっとあなたも今までにワープロ用のフロッピーや、音楽用の MiniDisk など、コンピュータ用のディスクの類似品を目にしたことがあると思います。このディスクの中にいろんなものを記録していくわけですが、コンピュータのディスクはCDみたいに交換できない上に、UNIX コンピュータは大勢の人が使っているので、整理して「もの」を配置しないとどこに何を記録したのか判らなくなってしまいます¹⁰。CDだってアルバム一枚に幾つもの曲を入れていませぬ。そうしないと5曲目だけ取り出すなんていう事が出来なくて不便です。

コンピュータのディスクの中にもものを記録する場合、記録はそれぞれ他の記録と混同しないように分割して残され、他の記録と区別するためにそれぞれ名前が付けられています。このそれぞれ名前を付けられたひとかたまりの情報の集まりを「ファイル」と呼んでいます。つまりコンピュータのディスクの中にはそれぞれ名前を付けられたファイルが幾つも記録されているということです。

4.4.2 ファイルの一覧を見る

UNIXに限らず現在市場に出ている殆どのコンピュータはファイルという形で情報（データなど）を格納しています。丁度様々な記録を紙に書いて、ファイル（file、書類差し）に入れて保存するようなものです。あなたの机に文書整理用のファイルがいくつも置いてあるように¹¹、コンピュータの中のあなたの記録場所にも幾つもファイルを置くことが出来ます。幾つも作ることが出来るので、それぞれを区別するために名前が付けられています。「ファイル名」として表現します。

ファイル名には実際には殆ど全ての文字が利用できます。しかし様々な理由から、ファイル名として利用する文字としては、アルファベット大文字、小文字、数字、_（アンダースコア）、-（ハイフン）、.（ピリオド）、,（コンマ）、:（コロンの）、;（セミコロン）、#（シャープ）、~（オーバライン）程度にすべきです。これら以外の記号文字、漢字などはファイル名としては利用しない方が無難です。また、ファイル名の長さは255文字が最大です。

ls¹²コマンドで、今あるファイルの名前の一覧をチェックすることが出来ます。

```
csosf01(82)% ls
```

⁹など書いていますが、実はこの文章を書いている人が良く理解していないから説明できないのです。済みません。

¹⁰長らく使っている留守録用のビデオテープのように

¹¹ない人も、まああると思ってください。

¹²字が判りにくいかも知れませんが、英小文字のLとSです。listを縮めたつもりなのです


```
Apps          Library      Mail          jsykojin.dic
csosf01(83)%
```

つまり「Apps」「Library」「Mail」「jsykojin.dic」という名前の4つのファイル¹³があるということです。

4.4.3 試しにファイルを作ってみましょう

ファイルの一つの利用方法として、あなたが実行したコマンドの記録を取る方法を紹介します¹⁴。script ファイル名 <Return>で、ファイル名のファイルにコマンドの実行結果が記録できます。ここでは例として時刻、実行しているマシンのUNIXの種類、ある月のカレンダーを表示させるコマンドを実行した記録を取ります。ファイル名として、とりあえず test を指定してみます。script コマンドの終了は exit です。つまり今回のサンプルでのタイピングは、

```
script test <Return> date <Return> uname <Return> cal 7 1999 <Return> exit <Return>
```

となります。

```
csosf01(83)% script test
Script started, file is test
csosf01(81)% date
1994年03月08日(火)23時22分51秒
csosf01(82)% uname
OSF1
csosf01(83)% cal 7 1999
      7月1999
日 月 火 水 木 金 土
          1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
csosf01(84)% exit
csosf01(85)% Script done, file is test
```

test という名前のファイルが出来た事を ls コマンドで確認しましょう。

```
csosf01(84)% ls
Apps          Library      Mail          jsykojin.dic  test
csosf01(84)%
```

4.4.4 ファイルの内容を見る

cat ファイル名 <Return>で、出来たファイルの内容を表示する事が出来ます。

```
csosf01(85)% cat test
Script started on Tue Mar 08 23:22:47 1994
csosf01(81)% date
```

¹³ cc 環境では作った覚えがなくても、上記の4つのファイル程度は既にホームディレクトリ(後述)に作られています。

¹⁴ 学生のレポート提出などに有効でしょう。

```
1994年03月08日(火)23時22分51秒
```

```
csosf01(82)% uname
```

```
OSF1
```

```
csosf01(83)% cal 7 1999
```

```
7月 1999
```

```
日 月 火 水 木 金 土
```

```
1 2 3
```

```
4 5 6 7 8 9 10
```

```
11 12 13 14 15 16 17
```

```
18 19 20 21 22 23 24
```

```
25 26 27 28 29 30 31
```

```
csosf01(84)% exit
```

```
csosf01(86)%
```

```
script done on Tue Mar 08 23:23:19 1994
```

このように情報をファイルにする事によって、いつでも取り出せる形で保存する事が出来るのです。一般的には、ファイルは明らかに消すと言う操作をしない限り消えることはありません。logout して、再び login しても、コンピュータの電源を切っても残っています。

ファイルの内容が長い場合

cat コマンドでファイルの内容を表示させた場合、その内容が長いと一画面に入り切らずにどんどんスクロールして行ってしまいます。そのような場合には more コマンドで一画面ごとに表示を一旦停止させる事が出来ます。more **ファイル名** <Return>です。

一旦停止している時に、以下のキーで様々なアクションを指示出来ます。

キー	アクション
<Space>	一画面分スクロール
<Return>	一行分スクロール
f	一画面分スキップして一画面分スクロール
b	二画面分戻って一画面分スクロール
h	どのキーを押せばどんなアクションをするかを教えてくれる

4.4.5 ファイル名を変える

ファイル名は変更することが出来ます。UNIX においてファイル名を変更するという作業は、実はファイルを別のところへ移動すると言うことを意味します。つまり前のファイルは消えてなくなって、新しい名前で生まれ変わるという事です。コマンドは mv¹⁵です。引数が最低二つ必要で、最初の引数が元のファイル名、最後の引数が移動先のファイルです。例えば以下の例ではとりあえず test と付けたファイル名を log という名前に変えています。これはつまり test というファイルを log というファイルへ移動する、という事です。

```
csosf01(86)% mv test log
```

test という名前のファイルがなくなって log という名前のファイルが出来た事を ls コマンドで確認しましょう。

¹⁵move を縮めたつもりなのです

```
csosf01(86)% ls
Apps          Library      Mail          jsykojin.dic  log
csosf01(87)%
```

4.4.6 ファイルの複写

ファイルはその内容をそっくりそのままに複写することが出来ます。コマンドは `cp`¹⁶です。引数が最低二つ必要で、最初の引数が元のファイル名、最後の引数が複写先のファイルです。例えば以下の例では `log` ファイルを `log2` という名前で新しく作り、内容はそっくり `log` ファイルから引き写しています。先の `mv` との違いは、元のファイルがそのまま残る (`cp`) か、それとも消える (`mv`) かです。

```
csosf01(87)% cp log log2
csosf01(88)%
```

`log2` という名前のファイルが増えた事を `ls` コマンドで確認しましょう。

```
csosf01(88)% ls
Apps          Mail          log
Library      jsykojin.dic log2
csosf01(89)%
```

4.4.7 ファイルの消去

ファイルは消去することが出来ます。逆に消去しない限りいつまでもそこに残っています。ファイルの置き場所は容量的には限りがあり、しかも複数の人で同じ置き場所を使っている場合が多いので、不要なファイルは削除するように心がけましょう。コマンドは `rm`¹⁷です。引数としてファイル名を与えます。例えば以下の例では `log2` ファイルを削除しています。

```
csosf01(89)% rm log2
csosf01(90)%
```

`log2` という名前のファイルが消えた事を `ls` コマンドで確認しましょう。

```
csosf01(88)% ls
Apps          Library      Mail          jsykojin.dic  log
csosf01(89)%
```

¹⁶ `copy` を縮めたつもりなのです

¹⁷ `remove` を縮めたつもりなのです

4.5 ファイルを編集する

今まで扱ってきたファイルは、先ほど `script` コマンドで作成したものです。ファイルは、このようにして作成するばかりでなく、利用者が自分の好きなようにその内容を変更したり、情報を追加したり出来ます。例えば先の `script` コマンドで作成したファイルも、レポートのつもりであれば自分の学部、学科、氏名を先頭に入れたいものです。

そこでファイルを自由に編集する方法を紹介します。`mule` です。ここでは Mule¹⁸を使ってファイルを編集する作業を簡単に説明します。Mule は全く多機能なソフトウェアで非常に多くの機能があります¹⁹が、ここではそれらのほとんどを説明しません。

4.5.1 Mule での作業の流れ

初めに Mule でファイルを編集するときの作業の流れを示しておきます。

- Mule の起動
- ファイル名の指定
- 編集
- ファイルの保存
- Mule の終了

Mule ではコントロールキーを押しながらの作業が多くなります。念のため今一度説明しておきます。

コントロールキーを押しながら `x` キーを押すことをここでは `C-x` と表記します。

エスケープキーを押してから `x` キーを押すことをここでは `M-x` と表記します。

4.5.2 Mule の起動

Mule を利用する環境には二通りあります。

- A. X ウィンドウ環境が目のある場合。計算機センターが管理しているコンピュータ環境における具体例としては DEC-3300 の前に座って利用している場合です。
- B. X ウィンドウ環境が目前にない場合。計算機センターが管理しているコンピュータ環境における具体例としては、パソコンが置いてある情報処理教室から SPARCcenter2000 を利用している場合などです。

いずれの場合も Mule でファイルを編集する場合のコマンド名は、`mule` ですが、それぞれの場合に応じてすこし作法が変わります。以下に Mule を起動するところを別々に説明します。注意深く読んでください。図 4.2 に、最初にうまく Mule が起動できた場合の画面表示を示しておきます。

¹⁸ 「みゅーる」と読んでください。

¹⁹ それを全て紹介した本の厚さは軽く 2 センチありますね。

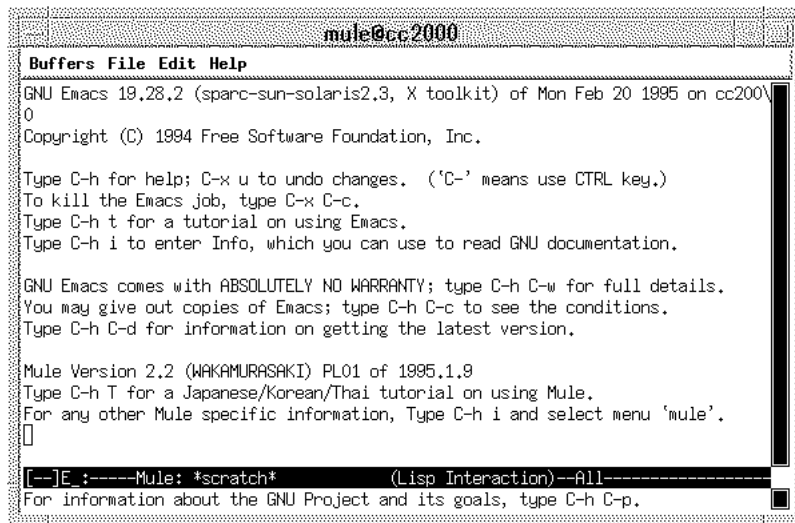


図 4.2 Mule の起動直後

Mule は起動すると常に最初の時点で画面内にメッセージを書いています。いつでも同じメッセージですので、特に気にしないで下さい。

Mule の画面を良く見ると、下から二行目に黒く色が反転した行がありますね。Mule は、この行を境にして、画面をそれより上の数十行の部分とそれより下の一行の部分に分けて使います。上から順番に以下のように呼ばれています。この呼び名は時々説明に現れますので、覚えてください。

- ウィンドウ 編集するファイルの内容が表示される。(X ウィンドウなどのウィンドウと区別するために Mule のウィンドウ、テキストウィンドウなどと表現することもあります。)
- モードライン 編集しているファイルなどについての情報が表示される。常に白黒反転して表示される。
- エコーライン Mule からのメッセージなどが表示される

4.5.3 A. の場合：X ウィンドウ環境での Mule の起動とエラー対策

`mule &` <Return> で Mule を起動します。

```
csosf01(85)% mule &
[1] 9503
csosf01(86)%
```

新たなウィンドウが一つ開いて、図 4.2 のような画面表示になると思います。もしもそうならない場合は、まず <Return> を一度押して、エラーメッセージがあるかどうかをチェックして下さい。エラーメッセージを見て以下のいずれの症状が発生しているかをよくチェックして、その対処を施し、もう一度 `mule &` コマンドを試みて下さい。

エラーメッセージ：Please set the environment variable TERM; see tset(1).

このエラーメッセージが表示された場合は以下のコマンドを実行し、再度 `mule &` <Return> です²⁰。

²⁰ UNIX に慣れた人はおやつと思われるかも知れませんが、一般的に X 環境下で DISPLAY 変数が設定されていれば TERM 変数の問題でトラブルは出ないのです。故にまず DISPLAY 変数が設定されていないことを疑うべきです。

```
csosf01(86)% setenv DISPLAY unix:0.0
```

エラーメッセージ: Xlib: connection to "unix:0.0" refused by server 他

```
Xlib: connection to "unix:0.0" refused by server
```

```
Xlib: Client is not authorized to connect to Server
```

```
X server not responding. Check your DISPLAY environment variable.
```

上記のようなエラーメッセージが出た場合はちょっと問題です。本当にあなたが利用している環境は X ウィンドウ環境ですか？あなたが該当するのは B. のケースではないですか？一度確認してください。B. のケースであれば B. のケースでのこのエラーの欄を参照してください。

もし本当に X ウィンドウ環境でこのようなことが起きるのなら、以下のコマンドを実行し、再度 `mule & <Return>` です。「`'`」バッククォートに気を付けてください。

```
csosf01(86)% setenv DISPLAY 'hostname':0.0
```

エラーメッセージ: X server not responding. Check your DISPLAY environment variable.

上記のようなエラーメッセージが出た場合はちょっと問題です。本当にあなたが利用している環境は X ウィンドウ環境ですか？あなたが該当するのは B. のケースではないですか？一度確認してください。B. のケースであれば B. のケースでのこのエラーの欄を参照してください。

4.5.4 B. の場合: 非 X ウィンドウ環境での Mule の起動とエラー対策

`mule <Return>` で、Mule を起動します。

```
csosf01(85)% mule
```

これで図 4.2 のような画面表示になると思います。もしもこうならない場合は、以下のいずれの症状が発生しているかをよくチェックして、その対処を施し、もう一度 `mule` コマンドを試みてください。

エラーメッセージ: Please set the environment variable TERM; see tset(1).

このエラーメッセージが表示された場合は以下のコマンドを実行し、再度 `mule <Return>` です。

```
csosf01(86)% set term=vt100
```

エラーメッセージ: Xlib: connection to "unix:0.0" refused by server 他

```
Xlib: connection to "unix:0.0" refused by server
```

```
Xlib: Client is not authorized to connect to Server
```

```
X server not responding. Check your DISPLAY environment variable.
```

上記のエラーメッセージが表示された場合は以下のコマンドを実行し、再度 `mule <Return>` です。上記の `"unix:0.0"` の部分が多少違っていても対処は同じです。

```
csosf01(86)% unsetenv DISPLAY
```

エラーメッセージ: **X server not responding. Check your DISPLAY environment variable.**

上記のエラーメッセージが表示された場合は以下のコマンドを実行し、再度 **mule <Return>** です。上記の "**unix:0.0**" の部分が多少違っていても対処は同じです。

```
csosf01(86)% unsetenv DISPLAY
```

4.5.5 ファイル名の指定

Mule が無事に起動できたら、まずは編集するファイルの名前を指定しましょう。指定した名前のファイルが存在しない場合、Mule はその名前で新たにファイルを作成します。指定したファイルが既存のものならば、Mule はそのファイルを読み込みます。

起動した直後の状態では、モードラインの左の部分に ***scratch*** と表示されているでしょう。ちょっと覚えておいてくださいね。

ファイルを指定するためには **C-x C-f** です。まずはじめに **C-x** を押してください²¹。数秒待つとエコーラインに **C-x-** と現れるでしょう²²。ここで更に追い打ちを掛けるように **C-f** です。するとエコーラインには以下のように表示されますね。

```
Find File: ~/
```

この状態で編集したいファイル名をタイプし、そして **<Return>** です。今回は先ほど作成した **log** ファイルを編集する事にして、ファイル名のところに **log** とタイプしましょう。**<Return>** で **log** ファイルの内容が Mule のテキストウィンドウ部分に読み込まれましたね。

さて、先ほどちょっと覚えておいてと言っていたモードラインの左の部分に **log** と表示されているのが判りますか? つまりモードラインのこの位置は、今現在編集しているファイルの名前を常に表示しているのです。最初に ***scratch*** だったのはまだ何のファイルも編集していないよと言うことを示していたのです。

4.5.6 編集

いよいよファイルの編集です。テキストウィンドウには目標のファイルが読み込まれていますね。この状態で、カーソルは恐らくテキストウィンドウの左上端だと思われます。この状態でタイピングすれば、カーソルのある位置にタイプした文字が入力されます。タイプした文字を消したい場合は **<Delete>** です。

カーソルの移動

カーソルは以下の方法でテキストウィンドウの任意の位置に移動することが出来ます。移動した先でタイプすればその文字がカーソルのある位置に入力されます。

²¹ **C-x** の後で **<Return>** などしてはいけません。

²² 実はこのエコーラインの表示を待つ必要はありません。慣れたら待たずに次のキーを押して下さい。

左矢印 (←),C-b	カーソルを左に一文字移動
右矢印 (→),C-f	カーソルを右に一文字移動
上矢印 (↑),C-p	カーソルを上に一文字移動
下矢印 (↓),C-n	カーソルを下に一文字移動
C-a	カーソルを行の先頭に移動
C-e	カーソルを行の末尾に移動
M-b	カーソルを左の単語に移動
M-f	カーソルを右の単語に移動
M-a	カーソルを文の先頭に移動
M-e	カーソルを文の末尾に移動

画面の移動

カーソルをどんどん下に動かして行けばいつかテキストウィンドウの下端に到達します。そこで更にカーソルを下に移動させれば画面が一ページ弱スクロールします。カーソルキー以外にも画面を移動させる方法としては、以下のキーがあります。

C-v	画面を下に（つまり次の画面に）移動
M-v	画面を上（つまり前の画面に）移動
M-<	文頭（つまりファイルの先頭に）移動
M->	文末（つまりファイルの最後に）移動
C-l	（<Control>と英字の L です。）カーソルのある行を画面中央に持ってくるように画面を移動

行の移動

Mule でしばらく編集していると、ある行の前後関係を入れ換えたい、もしくはある行を別の行を数行以上またいだ別の位置に移動したいと思うことがあるでしょう。このような場合は以下のステップを追うことで行の移動として実現できます。

1. 移動したい行を C-k で削除する。
2. 移動先にカーソルを移動させる。
3. C-y で 1. により削除した行を複製する。

つまり C-k を一度押すことによってカーソルの位置より右の一行分を削除することが出来ます。C-k を続けて何度か押すことによって更に下の行もまとめて削除することが出来ます。注意しなければならないのは、この削除の最中にカーソルを移動したり、何かほかの操作をしてはならないということです。あくまで連続した C-k の繰返しとして実行する必要があります。

移動する行の削除が済んだら、今度は移動先の位置にカーソルを移動させます。その後で C-y とすると、先ほど削除した行がその場所に割り込むように複製されます。

このようにしておこなう行の移動は、一行でも複数行でも構いません。C-y は何度でも行えますから、行の複製としても応用できます。C-k をやったすぐ後でカーソルを全く動かさずに C-y を実行すれば削除する前の状況に戻りますから、それからカーソルを移動してまた C-y をすれば、行を違う場所に複製することにも使えます。様々な応用が利く方法ですのでぜひ覚えてください。何度か失敗するかも知れませんが、機会を見てじっくり練習するのがお勧めです。

4.5.7 ファイルへの保存

カーソルを自由に動かして自分の思うようにファイルを編集したら、ここで保存をしましょう。ファイルは Mule に読み込まれて Mule の中で編集されているだけで、元のファイルは全く編集されずにそのまま残っています。Mule の中の編集された結果を元のファイルに書き込むことによって、編集の結果が反映されると言う仕掛けです。この種の操作を一般的に「保存」「セーブ (save)」と呼んでいます。

ファイルへの保存は `C-x C-s` です²³。保存がうまく行けばエコーラインに

```
Wrote /NF/home/syokuin0/yasuda/log
```

などと表示されます。

4.5.8 Mule の終了

さて、保存も済んだら Mule を終了したいところですが、再び 4.5.2 の場合分けに戻ります。

A. の場合だと、Mule は別の X のウィンドウとして起動されていますから、もともと Mule を起動したシェルが動いているウィンドウは別に残っており、そこでまた別のコマンドを実行できますから、特に Mule を終了する必要はないでしょう。シェルのウィンドウをクリックしてそちらをアクティブにするだけで良いですね。

B. の場合だと、Mule を終らないとシェルのプロンプトが現れず²⁴、次のコマンドが実行できませんから Mule を終らなければなりません。

いずれにしても Mule を終了するには `C-x C-c` です。もしもまだ保存していないファイルを編集集中に Mule を終了しようとした場合は、以下のようなメッセージがエコーラインに現れます。

```
Save file /NF/home/syokuin0/yasuda/log? (y ,n, !, ., q, C-r or C-h)
```

編集中のファイルを保存して Mule を終了する場合は `y` をタイプします。それでシェルのプロンプトが現れるでしょう。

保存しない場合は `n` をタイプします。この場合、Mule は念のためもう一度以下のような確認の問い合わせをします。

```
Modified buffers exist; exit anyway? (yes or no)
```

今度は `yes<Return>` とタイプします。これで編集集中の内容はファイルに保存されず、シェルのプロンプトが現れるでしょう。

4.5.9 Mule もっともっと

ここまでで非常に簡単に Mule の使い方を紹介してきました。でもこれだけの機能で日常的にファイルを編集するのはやはり不便があると思います。

実際、Mule にはもっともっとさまざまな機能があります。ここではさらに深く Mule の使い方を知りたい人のためにいくつかの方法を紹介しましょう。

その前にちょっとしたテクニック

Mule は非常に多くの `<Control>` キーや `<ESC>` キーを利用した機能があります。これらの機能を実行しようとして、キー操作を間違えてしまったり、操作の途中でおかしくなった場合、下手をするとどんどんと深い失敗の谷に落ちて行くときがあります²⁵。そういう事にならないためには、やはり「あぶ

²³ くだいですが `<Return>` キーなど押さずにコントロールキーを押しながら `x s` と押します。

²⁴ 実はそんな事はなくて Mule の中からシェルを呼び出すことも出来るのですが、ここでは説明しません。

²⁵ これが実は結構怖いんです。

ない」と思ったときにはすぐに作業を中断するのが得策です。Mule では作業の中断は一般的に C-g です。何が置けてもこれさえ覚えておけばもうそれ以上失敗の傷を広げることはありません。覚えておくといいでしょう。

もっと詳しいドキュメント

第7章「Mule」に、より詳しい Mule の使い方についての説明がありますのでそちらを参照して下さい。また、付録の参考文献にも幾らか挙げておきますので、そちらも参照して下さい。

漢字をタイプしたい

今まではアルファベットをタイプする方法しか説明しませんでしたから、普通の人が日常的な言葉をファイルに書き込むような事は出来ませんね。でも安心して下さい。Mule を利用してかな漢字変換もできます。4.9 に、Mule 上でのかな漢字変換機能についての説明がありますのでそちらを参照して下さい。

チュートリアル

Mule には自己学習の為の機能がついています。図 4.2 に出ている表示を良く読めば判るように、Mule を起動した後、C-h T²⁶ とすれば Language: と表示されます。そこで Japanese とタイプして<Return>して下さい。ここで先頭の J が英大文字である事に注意！シフトキーを押しながら J キーを押します。後はシフトキーを離して `apanese` です。以下のような画面表示になると思います。



図 4.3 Mule 入門（部分）

この入門編を読みながらそこに書かれている通りに操作をしていけば、Mule の殆どの機能について知ることが出来ます。（もしも入門編が英語で表示されたら先ほどの Japanese の綴りを間違えたのでしょう。一旦終わってからやり直すのがお勧めです。）この入門編を終るには（勿論終わり方も入門編に書いてありますが）4.5.8 を読んで下さい。入門編を始めた時のモードラインを見れば判るのですが、TUTORIAL.jp という名前のファイルを編集している事になっています。その為、終了作業をすると以下のように保存するかどうかを問い合わせる場合があります。

```
Save file /NF/home/syokuin0/yasuda/TUTORIAL.jp? (y or n)
```

ここで 4.5.8 の記述にしたがって保存しておくこのファイルが残ってしまいます。特に必要でない限りはここでは保存しないのがいいでしょう。チュートリアルの中に保存したとしても、もう要らないと思ったらこまめに `rm TUTORIAL.jp <Return>` で消去するのがおすすめです。

²⁶T は大文字の T です。つまりシフトキーを押しながらアルファベットの T キーを押します。

4.6 印刷

UNIX 環境では様々なものをプリンタから印刷する事が出来ます。cc 環境でも、幾つかプリンタを用意しており、これは cc 環境を利用出来るコンピュータからならどこからでも誰でも利用出来るものです。

4.6.1 どんなプリンタがあるか

cc 環境から利用出来るプリンタは以下の通りです。プリンタは複数あるので、それぞれ名前が付けてあります。

プリンタ名	設置場所
ccpr01	計算機科学研究所 2 階ミニコン室 (白くて小さな方)
ccpr02	計算機科学研究所 2 階ミニコン室 (茶色の大きな方)
cspr01	2 号館 4 階 21 情報処理教室
clpr01	3 号館 2 階 31 情報処理教室
cepr01	5 号館 1 階 51 情報処理教室
c1kpr01	第一研究室棟 2 階共同利用室
c9pr01	9 号館

プリンタを利用する最初のときには、まずそのプリンタのある場所まで行って、実際のプリンタに出力されるのか確認しておく事をお勧めします。上の表は恒久的なものではなく、様々な要因でプリンタは配置換えをしたり名前を変更したりされる可能性があります。この種の状況の変化については 4.7 に説明している `ccinfo` コマンドを利用して知ることが出来ます。後で参照して下さい。

プリンタの名前はそれぞれのプリンタ自体に貼り付けてありますから、まずプリンタのところに行って、プリンタの名前を確認しておくことがトラブルを起こさないためには重要です。

4.6.2 ファイルの印刷

`lpr -P プリンタ名 ファイル名 <Return>` とすれば指定のプリンタに指定のファイルの内容を印刷する事が出来ます。

```
csosf01(81)% lpr -Pcspr01 log
```

-P オプションに続くプリンタ名の指定は、上記の例のようにくっつけて書いてください。

例えば `-P cspr01` などとしてはいけません。第二の引数となるファイル名とプリンタ名の間には一つ以上の空白を開けてください。

注意

上記の方法でファイルを印刷する場合、どんなファイルでも印刷出来るとは限りません。`cat` コマンドなどで内容が確認出来るような文字ばかりのファイルに限ります。それ以外の、`cat` したら画面に変な文字がいっぱい表示されるようなファイルは印刷しないでください。プリンタが止まってしまったり無駄に数百ページ印刷されたりします。

4.6.3 印刷状況をチェックする

`lpq -P プリンタ名 <Return>` で指定のプリンタの現在の状況の確認が出来ます。プリンタの状態を表すメッセージが何行か帰ってきます。

プリンタがすいている状態

以下はプリンタに何も出力待ちのものが無い場合の例です。

```
csosf01(82)% lpq -Pcspr01
csosf01: Tue Mar  8 13:48:02 1994:
no entries
csosf01(83)%
```

上記のようではなく、単に `no entries` メッセージだけが返ってくる場合もあります。

プリンタがなにかを印刷している状態

以下のようなメッセージが帰ってきたら、それはプリンタが何か印刷している最中か、もしくは印刷のための準備中だと言う事です。

```
csosf01(88)% lpq -Pcspr01
csosf01: Fri Mar 11 15:29:59 1994:
cspr01 is ready and printing
Rank  Owner      Job  Files          Total Size
active tanaka    43  sample.ps     152 bytes
1st   yasuda     46  test.text      8 bytes
csosf01(89)%
```

上記の例では tanaka さんの sample.ps というファイルが現在印刷中 (active) で、その次 (1st) の yasuda さんの test.text というファイルが印刷待ちだという事です。

ユーザ名の右に出ている 43 や 46 の数字は印刷要求それぞれに割り当てられた番号で、ジョブ番号と呼ばれています。

4.6.4 印刷の取消し

一旦 `lpr` コマンドでプリンタに流し込んだ出力要求を取り消して、印刷しないようにできます。 `lprm -P プリンタ名 ジョブ番号 <Return>` です。一つ上の例の、yasuda さんは慌て者で、間違っ別ファイルの印刷するようにコマンドを実行してしまいました²⁷。この yasuda さんの印刷要求に付けられたジョブ番号は 46 です。これを取り消す例を示します。

(自分以外の印刷要求は決して取り消すことは出来ません。)

```
csosf01(99)% lprm -Pcspr01 46
benkei.kyoto-su.ac.jp: dfA046csosf01 dequeued
benkei.kyoto-su.ac.jp: cfA046csosf01.kyoto-su.ac.jp dequeued
csosf01(100)%
```

上記のメッセージは例で、実際これとはかなり違ったメッセージが表示されるかも知れません。重要なのは `dequeued` で、このメッセージが表示されればまず間違いなく印刷要求は取り消されています。念のためもういちど `lpq -Pcspr01 <Return>` などして印刷要求が消えていることを確認するのがいいでしょう。

²⁷ こんな人が紙を無駄遣いするんですね。

4.6.5 利用上の注意

ちょっとひとこと

最近プリンタの紙の無駄遣いや古紙の散乱が目立ちます。プリンタに印刷しようと思ったら、紙は使い切っているわ、周りは古紙だらけで汚いわ、では困ります。印刷するなど言うつもりはありませんが、慣れるに従ってお互い無駄な印刷はしなくて済ませる様にし、また積極的に散乱しているプリント結果の整理整頓をお願いします。

プリンタのトラブル

プリンタを使っていると、時々紙詰まり（ジャムと呼ばれる）や印刷が薄いなどのトラブルが発生します。この種のトラブルが発生したら計算機センターまで連絡下さい。また、計算機センターが配備している計算機運用補助員と呼ばれる学生が各情報処理教室を回っていますから、彼らに頼むのも良いでしょう。紙やトナー（インクのようなものです）の補給も彼らが行います。

大量の印刷をする場合

大量の印刷をする場合は他の利用者の迷惑にならないように、印刷要求の少ないときに行ってください。また、紙の補給を行わなければならない可能性のあるくらいページ数の多い印刷をしている場合は極力プリンタの見える位置で作業してください。

我々が利用しているプリンタは低速です。殆どのプリンタは毎分6ページ以下の印刷しか出来ません。これはつまり60ページのマニュアルを印刷するには10分以上掛かるという事です。他の人が何か印刷しようとしても、あなたのマニュアルを印刷し終わるまで10分も待たなくてはならないかも知れません。この点に注意して利用者みんなでうまくプリンタを共用しましょう。

ところで大量の印刷については、多少部屋が遠くても計算機科学研究所2階ミニコン室に設置してあるccpr02を利用するのがお勧めです。このプリンタは最大毎分20ページの印刷が可能ですので、通常の1/3以下の時間で印刷が終了します。

4.7 状況の変化

cc 環境は全く固定的な環境では無く、常に変化しています。例えば先に挙げたプリンタの配置や名前は、新しいプリンタの導入や利用者の要求に応じて変化して行きます。このドキュメントは印刷物ですから、印刷した時点で固定されてしまいます。最近に起こった変化は吸収していない場合もあるでしょう。

それを解決するために cc 環境では `ccinfo` というコマンドを用意しています。これは cc 環境で配布されているドキュメントや様々な情報の最新のもを利用者が簡単に取り出せる事を目標に設定されました。

`ccinfo<Return>` で起動できます。以下のようなガイドメニューが表示されるでしょう。この `ccinfo` コマンドを起動した直後のメニューをトップメニューと呼んでいます。例えばプリンタの情報は「1. cc 環境の設備について」以下にあります。

```
csosf01(125)% ccinfo
```

```
-----  
ここでは一般的な情報の検索が可能です。  
まずは分野を選んでください。
```

- 1 cc 環境の設備について
- 2 各種ドキュメント
- 3 ニュースグループについて
- 4 ネットワーク関係の情報
- 5 各種連絡先
- 6 そのほかの話題

番号を入力してください（0 で終了します）：

`ccinfo` コマンドは基本的にはメニューに表示される項目について、その番号をタイプすることによって選択する様になっています。番号として 0（ゼロ）を入力すると一つ前のメニューに戻ります。トップメニューで 0 を入力すると `ccinfo` コマンドそのものを終了します。

`ccinfo` コマンドのメニュー内容は常に更新され、その機能も常に変化していくでしょう。しかしどのように変化したとしても、基本的にはメニューの番号を選ぶか、問い合わせに対して y または n など返事をすることによって操作が進行するように作られています。

様々な情報が掲載されていますので、一度覗いてみることをお勧めします。

4.8 ファイルの階層構造

これまでで一般的なファイルの操作について説明してきました。ところで UNIX ではファイルは階層化されています。ここでは階層化されたファイルの概念と扱い方を説明します。

4.8.1 ディレクトリ

コンピュータをしばらく使っていると結構ファイルが増えて来て、そのうちどのファイルが何のためのものだったのか判らなくなるものです。ls コマンドで一覧を見たら一画面では収まり切らなくなったりして大変な状況の人も出てくるでしょう。そうならないために例えばファイルの名前を長くして、その名前を見ればファイルの内容の想像が付くようにするなど利用者は色々な対策を取ったりします。しかしそういう手法で全てが解決するわけでもありませんね。

第2章の4.4で、UNIXが扱うファイルはあなたの机の上に並ぶファイルのようなものだと言いました。もしもあなたの机の上にファイルが非常にたくさん並びはじめ、ファイルの背表紙にちょっと長めの名前を書いておくくらいでは目的のファイルがどこにあるのかすぐに探せなくなってしまったとしましょう。あなたはどうしますか？

多くの人は「整理が必要だ」と感じるでしょう。ファイルを分類し、分類ごとに大きな区分を作ってファイルを束にして置いておけば良いと言うわけです。分類とは物事を階層化して整理するということです。まず大分類があって、それから中分類、更に必要なら小分類、と言う感じですね。以下に日常的な机の上の分類の例を示します。四角で分類、丸でファイルを表しています。

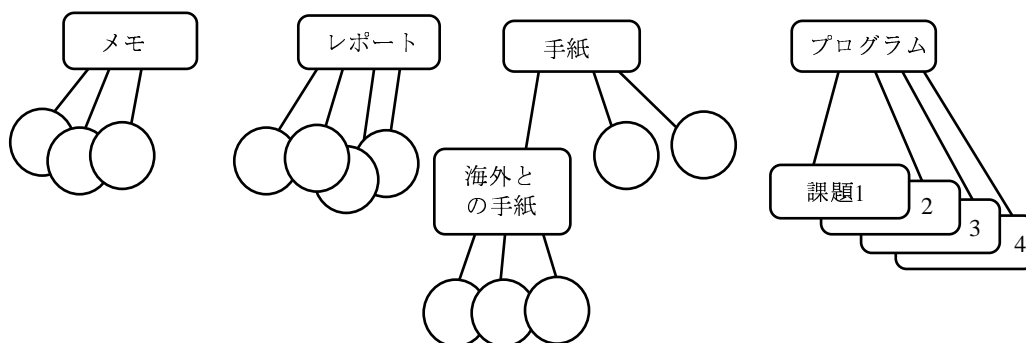


図 4.4 階層化された分類の例 (1)

UNIX でもファイルを階層化して整理することが可能です。以下に先の例に合うような階層化を行ったファイルの配置の例を示します。

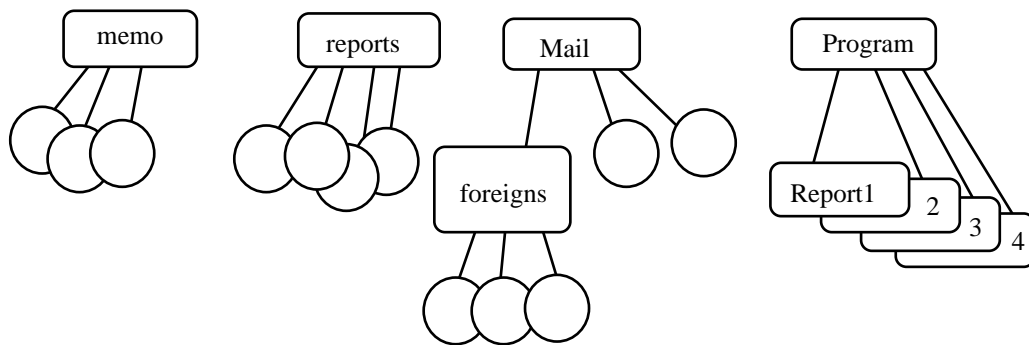


図 4.5 階層化された分類の例 (2)

しかし実は UNIX コンピュータは yasuda さん一人のものではなくて、数多くの人が同時に使っているということを忘れてはいけません。コンピュータにとっては、「yasuda のファイル」という分類がまず最初に既に存在しているのです。

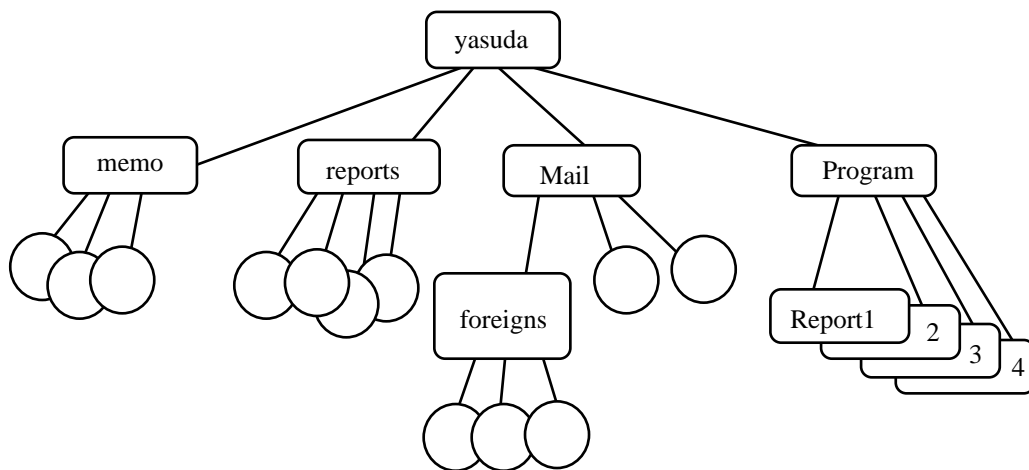


図 4.6 階層化された分類の例 (3)

そして「yasuda のファイル」は以下のようにもっと大きな分類の下に配置されているのです。

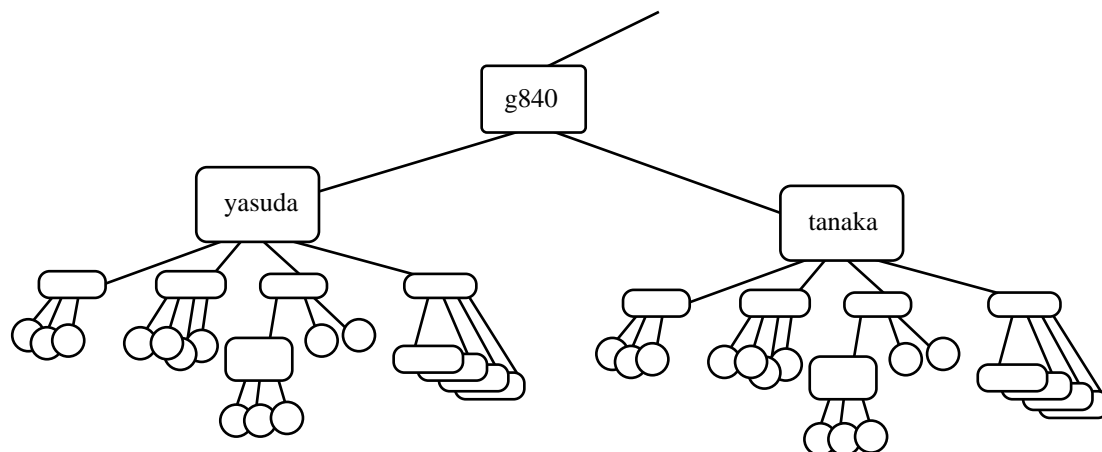


図 4.7 階層化された分類の例 (4)

上の例での「yasuda のファイル」より一つ大きい分類の名前は「g840」です。これは「ある年度に入学してきた学生のファイル」という意味の分類です。これよりもう少し上の存在として「全ての利用者のファイル」というものが有り、更にたどって行くと最終的に「全てのファイル」という最大の分類に到達します。

先ほどからの例では四角で表してきた、ファイルの分類なるものの存在を UNIX では「ディレクトリ²⁸」と呼んでいます。

UNIX では全てのファイルはたどって行けば「全てのファイル」を意味する唯一で最大の分類に到達します。この分類の根本（もしくは頂上）なるものの存在を「ルートディレクトリ²⁹」もしくは「トップディレクトリ」などと呼んでいます。

「yasuda のファイル」は実は yasuda という名前が付いたディレクトリなのです。この、利用者ごとに割り当てられた利用者自身のためのディレクトリを「ホームディレクトリ」と呼んでいます。ホームディレクトリは常に利用者のユーザ名が付けられて、利用者登録の際にシステム管理者³⁰の手によって作成されます。利用者は自分のホームディレクトリ以下に自分のファイルもしくはディレクトリを作り、保存することが出来ます。

いくつかのディレクトリを挙げましたが、いずれにしてもディレクトリには違い有りません。

UNIX のディレクトリとファイルの構造を見ると、それが木の根の構造のように見えると思います。また上下をひっくり返せばルートディレクトリを根にしてディレクトリの分類過程が枝のように、そしてファイルが葉のように見えるかも知れません。このようなイメージで表せる構造のことをコンピュータの世界では「木構造」「ツリー (tree) 構造」と呼んでいます。ディレクトリの木構造をディレクトリ・ツリーなどと呼んだりします。

ところで UNIX ではディレクトリはファイルの一種として扱われます。（扱われる、のです。普通のファイルとディレクトリはその意味に違いがあります。）そのため「ディレクトリファイル」などと表現されるときもあります。ディレクトリもファイルと同じ様に名前を付けてそれを他のものと区別して扱います。名前のルール（名前に利用できる文字、文字数の制限など）は普通のファイルと同じです。

²⁸ directory : 住所録? 何故この呼び名を採用したのか、私は知りません。(筆者)

²⁹ root directory : 根、ですね。

³⁰ cc 環境におけるシステム管理者は計算機センターです

4.8.2 ツリー構造におけるファイル名の表記

ファイルにはファイル名が有ることは説明しました。しかし UNIX のディレクトリ・ツリーの中で名前によってそのファイルを指定するにはツリー構造を含めて表現できる方法が必要ですね。

絶対パスによるファイル名の表記

例えばユーザ名 yasuda さんのホームディレクトリの名前は yasuda です。このディレクトリファイルを、UNIX コンピュータ全体のツリー構造の中では「/NF/home/g840/yasuda」などと表現します³¹。これが正にディレクトリ・ツリーの中で絶対的な位置と名前を表す表記法です。

先頭の「/」はルートディレクトリを表しています。それ以降の「/」はディレクトリ構造の区切り、つまり図 4.7 でのディレクトリとディレクトリもしくはファイルを結ぶ「線」に相当します。「/」には含まれた名前は全て途中に存在するディレクトリの名前であり、最後の名前はディレクトリ、もしくはファイルの名前です。

こうして表現することによってディレクトリ・ツリーの中で、名前によって完全にファイルが指示できるようになります。これを「絶対パスによるファイル名の表記」などと呼んでいます。絶対パスによる表記の場合、その表記はルートディレクトリを起点にして、たどって行くディレクトリの道のり (path : パス) を表現していると看做せます。

相対パスによるファイル名の表記

でも絶対パスによってしかファイルの名前が表現できないとしたらこれは非常に不便なことです。例えば yasuda さんが自分のホームディレクトリの直下にある log というファイルを old-log という名前に変えたいと思ったときに、こんな風にコマンドを書かなくてはいけません。

```
csosf01(82)% mv /NF/home/g840/yasuda/log /NF/home/g840/yasuda/old-log
```

いやこれはたまりません。これでは単に長い名前をファイルに付けているようなもので、何のためにディレクトリと言う概念を導入したのか判りません。

そこでファイルを表現するのに、ディレクトリ・ツリーの前半部分をタイプしなくても済むように覚えておいて、ツリーの残りの部分だけ表現すればいいような表記方法があります。そのために UNIX のシェルは、常にディレクトリ・ツリーのどれか一つのディレクトリに注目しています。そのディレクトリまでのツリーの記述は省略可能となるわけです。例えば先の例の mv において、今注目しているディレクトリがホームディレクトリだったとすると、以下のようにコマンドを短く書くことが出来ます。

```
csosf01(82)% mv log old-log
```

この「今注目しているディレクトリ」もしくは「今省略可能であるディレクトリ」を「カレントディレクトリ³²」もしくは「ワーキングディレクトリ」と呼んでいます。カレントディレクトリはコマンドによって変更することが出来ます。(後述)

pwd コマンドでカレントディレクトリを確認する事が出来ます。

```
csosf01(81)% pwd
/NF/home/g840/yasuda
csosf01(82)%
```

³¹ あなたのホームディレクトリはあなたのユーザ名が使われているはずですが。実際のあなたのホームディレクトリの名前が知りたければ login 直後に pwd コマンドで確認できます。

³² current directory : 現在のディレクトリ

先の例の、省略された `mv` コマンドのファイルに関する表記では、この `/NF/home/g840/yasuda` が省略されていた³³ と言うわけです。

ところで、`login` 直後のカレントディレクトリは常にホームディレクトリです。つまり今まで 第2章の 4.4 などで試してきたファイルは、この省略された表記法によって表現された、あなたのホームディレクトリ直下にあったファイルだったのですね。`ls` コマンドなどでその一覧が表示されていたのも、あなたのホームディレクトリ以下の内容だったというわけです。

こうして表現することによってディレクトリ・ツリーの中で、簡単な表記によってファイルが指示できるようになります。これを「相対パスによるファイル名の表記」などと呼んでいます。相対パスによる表記の場合、その表記はカレントディレクトリを起点にして、たどって行くディレクトリの道のり (`path`: パス) を表現していると看做せます。

パスによる表記でもう少し

言い遅れましたが「絶対的な表記」と「相対的な表記」の区別は、その表記の先頭が「/」であるか否かで判断されます。

また、パス中には以下の記号が利用できます。

記号	意味
.	カレントディレクトリを意味します 例えば <code>./sample</code> と書けば、カレントディレクトリにある <code>sample</code> というファイルを意味します。つまり単に <code>sample</code> と書いたのと同義です。(しかしそう書いたのでは駄目な場合もあるのです。)
..	一つ上のディレクトリを意味します。 例えば <code>../sample</code> と書けば、カレントディレクトリの一つ上のディレクトリにある <code>sample</code> というファイルを意味します。 例えば <code>../../sample</code> と書けば、カレントディレクトリの二つ上のディレクトリにある <code>sample</code> というファイルを意味します。

また、シェルからコマンドの引数としてファイルを記述するとき、パスの先頭であれば以下のような書き方も出来ます。(但し `sh` では駄目です。 `tcsh` か `csh` で有効です。)

記号	意味
~	自分のホームディレクトリを意味します
~username	ユーザ名 <code>username</code> のホームディレクトリを意味します

4.8.3 ディレクトリの扱い

ディレクトリの作成

ディレクトリを作成するには `mkdir`³⁴ コマンドを利用します。書式は以下の通りです。

`mkdir` ディレクトリ名...

ディレクトリの消去

ディレクトリを消去するには `rmdir`³⁵ コマンドを利用します。書式は以下の通りです。

³³ 厳密には `pwd` の結果の最後にもう一つ `/` を付けないといけませんね。

³⁴ `make directory` の略なのです。

³⁵ `remove directory` の略なのです。

rmkdir ディレクトリ名...

ディレクトリの消去は、そのディレクトリより下にディレクトリまたはファイルが含まれていては出来ません。消去したいディレクトリ以下のファイルまたはディレクトリを `rm` または `rmkdir` コマンドで予め消しておいてください。

カレントディレクトリの表示

カレントディレクトリを表示するには `pwd`³⁶ コマンドを利用します。書式は以下の通りです。

```
pwd
```

カレントディレクトリの変更（移動）

カレントディレクトリを変更するには `cd`³⁷ コマンドを利用します。書式は以下の通りです。

```
cd [ディレクトリ名]
```

`login` した直後はカレントディレクトリは常にホームディレクトリです。それから `cd` コマンドでどこかのディレクトリに移動しても、単に `cd<Return>`（つまり引数であるディレクトリ名を省略）とすると、常にホームディレクトリに移動します。

4.8.4 ディレクトリを意識したコマンドの書き方

今まで紹介してきたファイルを扱うコマンドは、そのほとんどがディレクトリに対しても適用できます。これ以降に以下の図の状況を例に取って説明します。

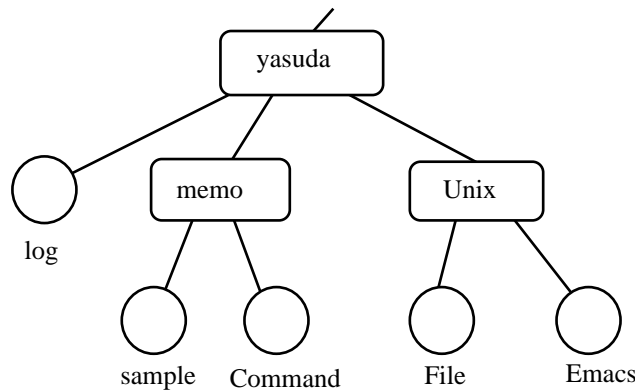


図 4.8 ディレクトリの例

まずは練習材料を作る

まず初めに練習のために図 4.8 の状況を作るのがおすすめです。以下にその手順を示します。今のカレントディレクトリはホームディレクトリではないかも知れませんが念のために最初に `cd<Return>` してホームディレクトリに移動しておきましょう。

³⁶ `print working directory` の略なのです。

³⁷ `change directory` の略なのです。

```

csosf01(84)% cd
csosf01(84)% mkdir memo Unix
csosf01(84)% ls
Apps          Mail          jsykojin.dic  memo
Library       Unix          log
csosf01(84)%

```

ファイルのコピー

cp コマンドには以下の3通りの書き方があります。

1. cp [-i] file1 file2
2. cp [-i] file... dir
3. cp -r dir1 dir2

-i オプション³⁸を与えると、コピーする際に同名のファイルが既に存在し、コピーすることによって書きされて元の内容がなくなってしまうような場合に実行してよいかどうか問い合わせを行う。これに y と答えると実行し、それ以外の入力であれば実行しない。

1. の書き方では、file1 は file2 に単にコピーされます。
2. の書き方では、(もし複数書けば複数の) file は、dir のすぐ下に元のファイル名でコピーされます。
3. の書き方では、dir1 以下のファイルを全て含めてディレクトリごと dir2 のすぐ下に元のファイル名、ディレクトリ名でコピーされます。dir2 が存在しないとそれを作成して、dir1 以下のファイルを全て dir2 のすぐ下に元のファイル名、ディレクトリ名でコピーされます。

例えば log ファイルを memo ディレクトリ以下に同じく log という名前でコピーする場合、以下のようなさまざまな表現が出来ます。

カレントディレクトリ	書き方	コマンド記述
yasuda	1.	cp log memo/log
yasuda	2.	cp log memo
memo	1.	cp ../log log
memo	1.	cp ../log ./log
memo	2.	cp ../log .

例えば sample ファイルを Unix ディレクトリ以下に同じく sample という名前でコピーする場合、以下のようなさまざまな表現が出来ます。

³⁸inquiry 問い合わせ、の積りでしょうか

カレントディレクトリ	書き方	コマンド記述
yasuda	1.	<code>cp memo/sample Unix/log</code>
yasuda	2.	<code>cp memo/sample Unix</code>
memo	1.	<code>cp sample ../Unix/sample</code>
memo	2.	<code>cp sample ../Unix</code>

例えば memo ディレクトリをツリーごと Unix ディレクトリ以下にコピーする場合、以下のようになります。(今度は書き方は 3. しかありません。)

カレントディレクトリ	コマンド記述
yasuda	<code>cp -r memo Unix</code>
memo	<code>cp -r ../memo ../Unix</code> (どういうわけか <code>cp -r . ../Unix</code> は駄目なのです)
Unix	<code>cp -r ../memo .</code>

この結果、ディレクトリ・ツリーは以下ようになります。

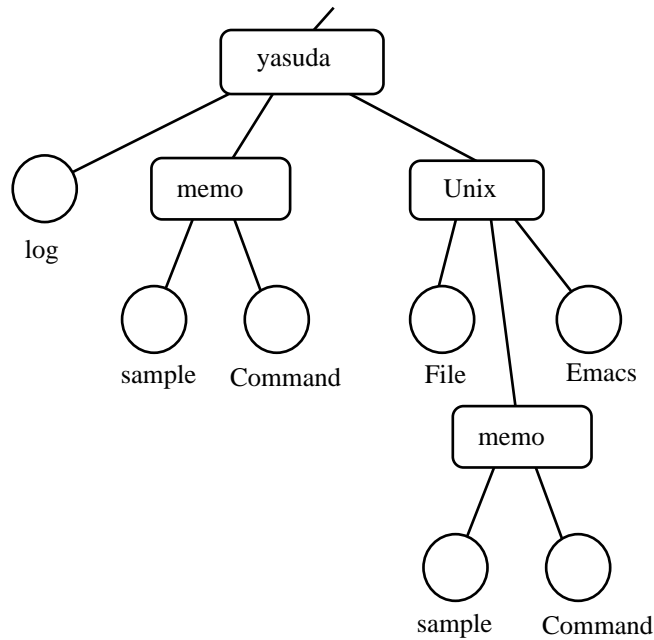


図 4.9 コピー後のディレクトリ・ツリー

ファイルの移動

mv コマンドには以下の 3 通りの書き方があります。

1. `mv [-if] file1 file2`
2. `mv [-if] file... dir`
4. `mv dir1 dir2`

`-i` オプション³⁹を与えると、移動する際に同名のファイルが既に存在し、移動することによって上書きされて元の内容がなくなってしまうような場合に実行してよいかどうか問い合わせを行う。これに `y` と答えると実行し、それ以外の入力であれば実行しない。

`-f` オプション⁴⁰を与えると上書きされて元の内容がなくなってしまうような場合でも問い合わせせず実行する。

1. の書き方では、`file1` は `file2` に単に移動されます。

2. の書き方では、(もし複数書けば複数の) `file` は、`dir` のすぐ下に元のファイル名で移動されます。

3. の書き方では、`dir1` 以下のファイルを全て含めてディレクトリごと `dir2` のすぐ下に元のファイル名、ディレクトリ名で移動されます。`dir2` が存在しないとそれを作成して、`dir1` 以下のファイルを全て `dir2` のすぐ下に元のファイル名、ディレクトリ名で移動されます。

`mv` コマンドはすぐ判るように `cp` コマンドとほとんど同じ記述方法が出来ます。働きもほとんど同じで、単に元のファイルが残る (`cp`) か残らない (`mv`) かだけです。`cp` コマンドと同じですので、ここでは例を挙げません。

最後の 3. の書き方で `memo` ディレクトリを Unix ディレクトリにディレクトリ・ツリーごと移動した場合は、図 4.9 に挙げた例の左側の元の `memo` ディレクトリと、それ以下の `sample` と `Comand` ファイルがなくなった状態になります。

ファイルの名前を変える

ファイルの名前の変更は `mv` コマンドで行ないます。つまり `sample` というファイルの名前を `otamesi` に変えるという作業は `sample` ファイルを `otamesi` に移動するという事と考えるのです。具体的には単純に `mv sample otamesi<Return>` です。

ファイルの消去

`rm` コマンドは以下の書き方をします。

```
rm [-ifr] file...
```

`-i` オプション⁴¹を与えると、消去する際に場合に実行してよいかどうか問い合わせを行う。これに `y` と答えると実行し、それ以外の入力であれば実行しない。

`-f` オプション⁴²を与えると問い合わせせず実行する。

`-r` オプション⁴³を与えて `file` の部分がディレクトリだった場合は `file` 以下のファイルを全て含めてディレクトリ・ツリーごと消去する。

特に `rm` コマンドに `-r` を指定してディレクトリ・ツリーごと消去するという例は非常に有用です。なにしろディレクトリの消去である `rmdir` コマンドは、そのディレクトリ以下の内容が空になっていなければいけないのです。そのような場合は `rm -r` で一発消去ができます。

³⁹inquiry 問い合わせ、の積りでしょうか

⁴⁰force 強制、の積りでしょうか

⁴¹inquiry 問い合わせ、の積りでしょうか

⁴²force 強制、の積りでしょうか

⁴³recursive 回帰的、の積りでしょうか

4.9 EGG : Mule での漢字の入力

いままでタイピングと言えばアルファベットのタイピングについてのみ説明してきました。しかしやはりひらがなや漢字をファイルの中にかきたいものです。ここでは Mule を用いたひらがなや漢字などのタイピングの方法について説明します。

4.9.1 かな漢字変換

アルファベットの場合タイピングは簡単です。つまりキーボード上のキーに書いてある文字が、そのキーを押すことによって入力されるのです。ひらがなのタイピングについては何とかこの方法で済ませられるかもしれませんが、漢字についてはそういうわけには行きません。漢字は数万字（日常的に使う漢字だけでも数千字）あって、とても数千のキーを並べるわけには行かない⁴⁴からです。

そこで「かな漢字変換」による漢字の入力の登場です。つまりまず「かな」を目標の漢字の読みとして入力することによって、それを漢字に変換しようと言うアイデアです。最近市販されているワープロは殆ど全てこの方式を採用しています。UNIX コンピュータでもこの方法で漢字をタイプします。しかも「かな」の入力はローマ字からの変換です。つまり「かな」すらキーボードには載っていない（もしくは載っていても使わない）ので、まずアルファベットのキーを利用してローマ字で「かな」を目的の「漢字」の読みとしてタイプし、それを変換するのです。

4.9.2 Wnn と EGG

Wnn⁴⁵ は 京都大学数理解析研究所、オムロン株式会社、株式会社アステックの 3 者によって開発されたかな漢字変換システムです。cc 環境ではこの Wnn を標準的な漢字変換システムとして採用しています。

EGG⁴⁶ は電子技術総合研究所の戸村哲氏が中心となって開発した、Mule 上で Wnn を利用するためのシステムです。cc 環境ではこの EGG を Mule 上での標準的なかな漢字変換システムとして採用しています。

EGG のモード切り替え

さて、Mule 上で漢字をタイプするためには、まず EGG をローマ字かなモードにします。Mule が起動されている状態で、C-\ です。（キーボードによっては C-\ の代わりに、C-¥ かもしれません。）モードラインの左端に注目してください。以下のようにするのが判ると思います。

```
[あ]E_:---*-Mule: *scratch*          (Lisp Interaction)--All-----
```

この状態から元に戻るためには、再び C-\ です。モードラインが元に戻りましたね。C-\ でモードラインの左端がくるくる変わるのを確認してください。

```
[--]E_:---*-Mule: *scratch*          (Lisp Interaction)--All-----
```

モードラインの左端が [--] となっている状態を EGG の「透過モード」と呼んでいます。（透過モードと言うのは、つまり今までどうりアルファベットのタイピングがそのまま行える状態です。） [a あ] となっている状態を「ローマ字かなモード」と呼んでいます。ローマ字かなモードにしておくと、タイプしたアルファベットはまず EGG に受けとられ、そこでローマ字として解釈されてひらがなが表示されます。それから漢字変換に関するキーを操作して目的の漢字かなまじり文へと変換するのです。

⁴⁴ 昔の漢字タイプライタは正にその通り数千のキーを並べていましたけどね。コンピュータではその方法は採用していません。

⁴⁵ 「うんぬ」と読みます。名前の由来は「私の名前は中野です」の略から来ているそうです。

⁴⁶ 「えっぐ」と読みます。名前の由来は「たくさんまたせてごめんなさい」の略「たまご」から来ているそうです。

ためしに変換

ローマ字かなモードにして、アルファベットで「wata sinonamae hanakanodesu」とタイプしてください。打ち込んだアルファベットが縦棒にはさまれながら次々とひらがなに変換されて行くのが判るでしょう。

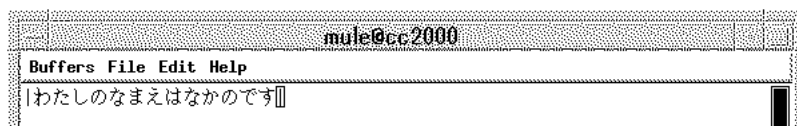


図 4.10 EGG のフェンスモード

この縦棒を EGG ではフェンスと呼び、変換途上のかな文字がフェンスにはさまれている状態をフェンスモードと呼んでいます。

フェンスの中の文字を編集するには普通の Mule の編集のときと同じくカーソルキー (←や→) と <Delete> キーです。カーソルキーが効かない場合は C-b でカーソル左、C-f でカーソル右と同じ動きをします。

ローマ字が全てかなに変換されたら <Space> キーを押すことによって今度はかなを漢字に変換する作業が始まります。

注意

あなたがそのコンピュータで初めてかな漢字変換を行うときは、以下のような問い合わせが行われます。



図 4.11 ユーザ辞書作成の問い合わせ

これらの質問には全て y と答えてください⁴⁷。

再び注意

あなたがその Mule で初めてかな漢字変換を行うときは、エコーラインに「ホスト local の WNN を起動しました」などというメッセージが表示されて、かな漢字変換が開始されるのに若干時間が掛かるかも知れません。

かな漢字変換が始まると、モードラインの左端が以下ようになります。

```
[漢]E :--**Mule: *scratch* (Lisp Interaction)--All-----
```

フェンスの中は以下のようになっているでしょう。このモードラインの左端が [漢字] となっている状態を「漢字変換モード」と呼んでいます。

⁴⁷ 結構何回も y と答えなければなりません。今試してみたら合計 12 回も必要でした！



図 4.12 EGG の漢字変換モード

ここで<Return>とすれば現在表示されている漢字かなまじり文の候補「私の 名前は 中野です」が採用されて、Mule にタイプされます。フェンスがなくなって、漢字が Mule の普通の操作で編集できる状態になりましたね。

この作業を「確定」と呼んでいます。これでまず一つ漢字の文章がタイプされました。一安心です。

再変換

一度変換するだけで自分の目的の漢字に変換してくれれば良いのですが、先ほどのようにうまく行くことばかりではありません。そこで、EGG が最初に挙げてくれた漢字の候補を再変換しなければなりません。一度でうまく行かない例として、今度は「tonarinokyakuhayokukakikuukyakuda」とタイプして<Space>で変換してください。恐らく以下のようなになるでしょう。

|戸なりの 客は 良くか 聞く 右-客だ|

惜しいですね。ここでは最終的に「隣の客は良く柿食う客だ」に変換し直す例をあげます。

かな漢字変換では変換は文節単位に行なわれます。はじめはローマ字でタイプされた一連のひらがなの列から EGG が自分で文節の区切りを見つけて変換します。上記の例ではその結果「戸なりの」「客は」「良くか」「聞く」「右」「客だ」と文節の区切りを付けました。EGG の漢字変換モードでは文節の区切りを一つの空白で表します。「戸なりの」と「客は」などの間にそれぞれ一つの空白があるのが判りますね。

次の候補を上げる

今、カーソルは「戸なりの」の上にあると思います。そこで<Space>を押すことによって次の候補が表示されます。この例では一度<Space>を押すと「隣の」が現れると思います。現れなかったら、現れるまで<Space>を押して下さい。あまり急いで<Space>を押すすぎると肝心の「隣の」が現れても勢いでもう一度<Space>を押してしまう時もあるでしょう。そういう時は C-p で一つ前の候補を再表示する事が出来ます。

文節の区切りを変える

|隣の 客は 良くか 聞く 右-客だ|

こうなりましたね。次は「良くか」の文節に移りましょう。フェンスの中の文節を移動するには普通の Mule の編集のときと同じくカーソルキー（←や→）です。カーソルキーが効かない場合は C-b でカーソル左、C-f でカーソル右と同じ動きをします。→もしくは C-f で「良くか」の上にカーソルがやってきました

この文節の区切りを「良くか」から「良く」に訂正しなくてはなりません。文節の区切りを縮めるには C-i です。一度の C-i で「良く」に直りましたね。

|隣の 客は 良く 化 聞く 右-客だ|

次に→もしくは C-f で「化」の文節に移りましょう。今度はこの文節を伸ばしてやる必要がありますね。文節の区切りを伸ばすには C-o です。一度 C-o すれば「書き」になりますね。これは望みの漢字ではない

ので先ほどやった<Space>で「柿」になるまで次の候補を上げて下さい。先にも書きましたが、変換が行き過ぎてしまったら C-p で戻れます。

目的の漢字になれば<Return>で確定、です。

4.9.3 まとめ

以下にキー操作のまとめを示しておきます。

ローマ字かなモードでのキー操作

<Delete>	タイプミスの修正、一文字削除
←または C-b	カーソル一文字左
→または C-f	カーソル一文字右
<Space>	かな漢字変換の開始

漢字変換モードでのキー操作

←または C-b	カーソル一文節左
→または C-f	カーソル一文節右
C-o	文節を伸ばす
C-i	文節を縮める
<Space>	次候補の表示
C-p	前候補の表示
M-s	候補の一覧をエコーラインに表示する。数字による選択かカーソルキー（もしくは C-b ,C-f ,C-p ,C-n ）による選択を行い<Return>で確定。 候補一覧表時から抜けて元に戻るには C-g 。

4.9.4 ローマ字入力のヒント

記号を入力したいのです

大抵の記号に関しては「○」は「まる」などのように、その記号の読みが登録されていますから、それで変換出来ます。

「たんい」（もしくは「んお？」など）のように「ん」の次に母音が来るのです

「ん」を単独でタイプするために N (大文字) と n' が用意されていますので、それを利用してください。「taNi」もしくは「tan'i」とタイプすれば「たんい」とローマ字変換されます。

「あっ」（もしくは「んあ？」など）のような小さい「っ」「あ」を入力したいのです

小さい「あいうえお」などは、x を前に付けてタイプします。「xa」とタイプすれば「あ」とローマ字変換されます。「xtu」で「っ」、「xyu」で「っ」です。

第5章

ネットワークの世界へようこそ

京都産業大学内では多くのコンピュータが導入され、教員や学生に提供されています。これらを使って行く上で、利用者同士の情報交換は非常に有用です。今や殆ど全てのコンピュータは電線で結ばれ、ネットワークと呼ばれる構造によってコンピュータ同士の間で通信が可能です。つまりコンピュータの利用者の間でお互いのメッセージを交換することが出来るのです。

典型的なメッセージ交換の形として「電子メール」「ニュース」などの方法が存在します。これらの機能を総称してネットワークサービスなどと呼んだりします。ここでは、それぞれのネットワークサービスがどのようなものか説明します。

5.1 ネットワークサービス紹介

ここでは「電子メール」と「ニュース」サービスがどのようなものかを説明します。すでに知識と経験のある人は5.3まで読み飛ばして下さい。但し5.2.2については読んでおく方が良いかも知れません。

5.1.1 電子メールって何？

電子メールは通常の郵便システムと似た機能を持ちます。大きな相違は紙や物を運ぶのではなく、文字をデータ化して相手に届けるという点です。つまり通常の葉書や封書による郵便は、実際には紙としての葉書や物としての封書を届けるのが第一の目的では無く、そこに書かれた内容、つまり文字を相手に届けることが本質です。しかし実際には紙や物を届けることによって実現しているというのが郵政省の郵便システムと言えます。

それに対して電子メールでは、一般的に文書（整形された文字の集合）の内容をコンピュータに入力する事によってデータ化し、またコンピュータの機能を使って相手に送り付けます。受け取った側でも、データ化された文書を、受け取った側に読めるようなかたちでコンピュータによって表示します。そうして送り手が書いた内容が受け手に伝わる、と言う仕掛けです。

仕掛けはともあれ、一般の郵便も電子メールも誰かが書いた文章をある特定の相手に届けると言う本質は変わりませんね。

後の5.1.5でもっと具体的な利点などを紹介します。

5.1.2 ニュースって何？

電子メールが個人と個人間のメッセージ交換であったのに対して、ニュースは個人が大勢に対してメッセージをやり取りできるシステムです。

ニュースではあたかも掲示板に個人がメッセージを書くようなもので、大勢の人が（もしも興味があれば）そのメッセージを読み、そして自分の意見を再びそれぞれ大勢の人に見えるように書き込みます。こうすることによって、広く一般に向けて公開の議論が可能になると言うわけです。

一般の掲示板は本当の木の板にチョークか何かで書くのですが、ネットワーク上のニュースシステムでは、コンピュータを介する事によってメッセージをデータ化して行います。木の板の文字かデータかはともあれ、個人が書いたメッセージを多くの人が覗いて公開の返事を書くという本質は変わりませんね。後の 5.1.6 でもっと具体的な利点などを紹介します。

5.1.3 京都産業大学のネットワーク

京都産業大学のコンピュータは、その多くが学内のネットワークに接続されています。計算機センターが管理しているコンピュータの多くも学内ネットワークに接続され、また上記のネットワークサービスが受けられます。

京都産業大学のネットワークは全世界を覆うネットワークに接続されています。この巨大なネットワークのことを Internet と呼んでいます¹。京都産業大学のコンピュータの多くは、つまり Internet につながれているということです。そして京都産業大学で行われているネットワークサービスの多くはこの Internet で行われているものと同じものです。それ故に京都産業大学の電子メールサービス、ニュースサービスは、世界中の電子メールサービス、ニュースサービスと通じ合えると言う訳です。

今やあなたも京都産業大学内のコンピュータを介して世界中の人と電子メールを交換することが出来ます。また、あなたは学内のコンピュータを介して世界中の人とニュースによって意見を交換することが出来ます。

5.1.4 Internet とは？

Internet は世界中のネットワークを相互接続したもので、学術研究のための実験ネットワークです。Internet には主体となる運営組織は具体的には存在しません。また中心となるコンピュータもどこにも存在しません。世界中にあるネットワークを接続して、お互いの好意で協調して運用しています。商用のいわゆるパソコンネットとはこれらの点で異なります。このネットワークの特徴として、主に以下のような特徴が挙げられます。

- 各組織（ドメインとも言います）の組織内ネットワーク同士を結合することにより、メールやニュース等のサービスを提供しています。サービスについては次節でもう少し説明します。
- 各 Internet 参加組織間（および組織内）のリンクの管理の多くは各組織のボランティアによって行なわれています。（全てでは無い）
- 各組織のネットワーク（特にメールサービス）の管理者をポストマスターといいます。
- 国内での Internet の利用は非営利目的な利用に限られます。営利目的に用いることは一切禁止されています。たとえば、Internet での商売、商品の宣伝、リクルート活動など。ただし海外では一概にそうとは言えません。それはその国の法律が決定するでしょう。
- 国内では Internet に参加している組織は、大学などの教育機関、企業、国立の研究機関などで、'95年 3 月現在約 1700 組織です。全世界だと、数万組織は接続しているのでしょうか？

¹なんだか話が SF チックになってきましたね。

これ以上の情報を得るためには、コンピュータ系の雑誌や Internet news の記事等から自分で勉強するように努力してみてください。図書館にも置いてある各種のコンピュータ雑誌も推薦出来ます。Internet news については 5.1.6 で説明します。

5.1.5 Internet mail サービスってどんなもの？

Internet で行われている電子メールサービスを利用すると、世界中に散らばる数万組織の人達に対して手紙を出すことが出来ます。しかも大学などの研究機関が主にネットワークに参加していますから、我々大学関係者などが連絡をとりたい相手もこの電子メールサービスを利用している可能性は結構高いのです。

この電子メールサービスは郵政省が行なっているメールサービス（郵便）に比べると以下の点で優れています。

- 手紙が速く相手の手元に到着する
たとえ相手がアメリカでもオーストラリアでもイギリスでも、大抵の場合（控え目に見積っても）数分間以内に相手の手元に手紙が届きます。実際アメリカなどへは郵便では 10 日くらいかかったりします。
- 基本的に 24 時間営業、年中無休である
夜にアメリカ宛に出したメールは恐らく向こう時間の朝に到着するでしょう。日曜日は配達してくれなかったり、年賀状シーズンに停滞したりすることはありません。
- 郵便ポストまで歩いて行かなくて済む
京都産業大学のネットワークに接続できる端末がありさえすれば、そこから全ての操作が出来ます。自宅から作業をする事も出来ます。
- 紙がたまらないで済む
頻繁に手紙のやり取りをしていると、結構大きさのまちまちな保存しにくい紙がたまるものです。しかし電子メールでは手紙はコンピュータの中に残り場所を取りません。そして、消さない限りいつでも取り出して読み返せます。更に、計算機の中に入っている限りキーワードで検索可能です。紙をめくって斜め読みする必要はありません。

逆に、以下の点では劣っています。

- 書留郵便がない
メール配送は先に述べた通りまだボランティアベースで行なわれている部分があります。出したメールを必ず（何があっても）相手に届けるようなサービスは行なわれていません。
- 自分がネットワークに加入していないといけない
しかもコンピュータを使わなければなりません。しかし最近ではコンピュータもずいぶん使いやすくなりました。
- 相手がネットワークに加入していないといけない
しかもコンピュータを使ってくれなければなりません。しかし最近では結構大学関係者はこういうものに参加しています。

電子メールは世の中の殆どの便利なものと同じ様に、決して万能ではありません。しかし、ある局面では他の何ものにも代えられない位役に立つことがあるのです。

5.1.6 Internet news サービスってどんなもの？

Internet news は最近普及しつつある NIFTY-serve など、つまり商用のいわゆるパソコンネットの電子掲示板 (BBS) システムに似ています。つまりたくさんの人が掲示板に自分の言いたいことを書き込むと言うものです。この掲示板はまた多くの人に見られていますから、読んでいる人が書き込まれたことに反論したり、意見を添えたりします。

具体的には「私は今度行われる教育改革には反対だ。」「いや、私はこの点で賛成する。」「私も賛成だ。」と言った議論や、「先日若狭に釣りに行ってきました。今は鯛が好調です。」「私も行ってきました。少し沖合いに出るとハマチが来ます。餌は生き餌がいいようです。」「私も行きたいのですが、舟を紹介してください。」と言った趣味の話などが行われています。

また、何かについて困っているときに質問を書き込めば、誰か親切な人が解決法を教えてくれるかも知れません。そんな都合の良い話は無いって？いいえ、これは冗談ではありません。ネットワークの向こうには何千人もの非常に親切でお節介な人が暮らしているのです！²

要は多くの人が見ているところに書き込むわけですから、勢い情報交換の場となるのです。しかもこれは世界中の人が参加しています。つまり我々は海外の情報も居ながらにして読むことが出来るのです³。非常にたくさんの人が参加して、いつも多くの書き込みがありますから興味のある話題ごとにグループ化されています。

また、京都産業大学の内部向けに幾つかのニュースグループが用意されています。「sandai.」が頭に付くニュースグループです。外部の情報には興味がない人も（せめてここだけでも）是非覗かれることをお勧めします。有用な情報の広報などはここで良く行われます。

Internet news は生まれが USENET と呼ばれる大学間で始まった実験ネットワークです。その歴史的な経緯でニュースシステム上で用いられる用語が少し一般のパソコンネットの電子掲示板システムとは違っています。

共通の興味によってくられる話題は決められた場所に書き込むことになっていますが、これをパソコンネットでは SIG もしくはボード、フォーラムなどと呼んでいます。Internet news ではこれをニュースグループと呼んでいます。

ニュースシステムにユーザが書き込んだメッセージを Internet news では記事 (アーティクル) と呼ぶ場合があります。

記事を書き込むことを Internet news では投稿 (ポスト) と呼んでいます。

5.1.7 ネットワークでの暮らし方

ネットワークサービスと共にコンピュータを利用するのは非常に快適なものです。ですが、その環境で快適に暮らすにはある程度ルールを心得ておくことが重要です。ネットワークサービスは機械によって提供されていますが、相手をしているのは機械ではなく、人間であることをとたく忘れがちです。

第1章の「はじめに」などでも述べているように、京都産業大学のコンピュータ環境には一般社会と同じ様に規則、慣習、道徳があり、そして法律も適用されます。京都産業大学のネットワークは Internet の一部でもあります。京都産業大学のコンピュータ利用環境よりもっと大きな利用環境である Internet にも勿論様々な規則、慣習があります。お互いに協調して暮らして行けるように心に留めておきましょう。

²嘘だと思うならコンピュータ関係のニュースグループを少し覗いてみてください。

³もちろん海外のニュースは英語が殆どです。でも日本国内のニュースは日本語 (漢字) で流れていますから、英語を読みたくない人も御安心下さい。「fj.」が頭に付くニュースグループが漢字のものです。

5.2 電子メール準備体操

5.2.1 Internet mail アドレスについて

Internet でサービスされているメールサービス⁴でも一般的な郵便システムと同じ様に、宛先の住所を明記する必要があります。Internet は世界中でサービスされているので、世界で一意に決まる⁵住所がメールサービスを受ける人それぞれに必要です。すなわちそれが住所であり、Internet mail サービスがアメリカ生まれであることからアドレスと呼びます。メールサービスのためのアドレスですから、メールアドレスと言うとその意味合いがもつとはっきりするでしょう。

現在のところ、メールサービスがコンピュータを介して行われているものであるため、メールアドレスはそれぞれのコンピュータの利用者に割り当てられます。逆に言えばメールサービスを利用するためには、メールサービスが利用可能なコンピュータの利用者とならなくてはなりません。

京都産業大学の幾つかのコンピュータは、Internet mail サービスが利用可能です。それらのコンピュータの利用者は Internet mail サービスを利用することが出来るでしょう。つまり Internet mail サービスを受けている世界中に数百万台（もつとかな？）存在するコンピュータのユーザそれぞれとメールを交換することが出来るという事になります。

京都産業大学のメールサービスが利用出来るマシンにおけるメールアドレスは一般的には以下のような書式となっています。

`foo@bar.kyoto-su.ac.jp`

アドレスの表記そのものは西歐式になっており、右側に大きな区分、左側に小さな区分が書かれています。ピリオド (.) で区切られており、右側からそれぞれの区分の意味を以下に示します。

<code>jp</code>	Internet mail アドレス最大の区分で、国を示しています。
<code>ac</code>	<code>jp</code> 以下に存在する中区分であり、教学関係であることを示す。他に <code>co</code> が一般企業、 <code>or</code> がその他の組織などとして定義されています。
<code>kyoto-su</code>	京都産業大学を示しています。
<code>bar</code>	ここには何が来るか一概には言えません。京都産業大学内で、既に決められている宛先に配送されることを示しています。大学のネットワーク管理者が決定した区分のようなものだと考えてください。
<code>@</code>	これより左はユーザ名であることを示しています。
<code>foo</code>	<code>foo</code> (仮称) というユーザ名宛てに配送されることを示しています。

`kyoto-su.ac.jp` は、京都産業大学のネットワークを示すアドレスであり、世界中を覆う Internet で一意な名前です。ネットワーク環境の中で、このように階層付けされて一意に確保されているような名前をドメインなどと呼んだりします。`kyoto-su.ac.jp` より左の `foo@bar` は京都産業大学内で一意に保たれるように管理されています。あなたが電子メールを利用するときはあなたが利用者として登録されているコンピュータの管理者に、自分のメールアドレスがどのようなものかを確認することが重要です。

5.2.2 計算機センター運用のコンピュータのメールアドレス

ここでは計算機センターが運用しているコンピュータのメールアドレスに限定して説明します。ここで説明するルールは普遍的なものでは無いので、他のマシンのユーザのメールアドレスを類推する役には立

⁴ これ以降単にメールと言ったら電子メールを差します

⁵ 一意に決まる、とは「間違いなくたった一つのものに特定出来る」と言う意味です。

たないでしょう。他のマシンのメールアドレスについては、それらのマシンの管理者に問い合わせる必要がある事に注意してください。

計算機センターが運用し、京都産業大学の教員、学生に提供しているコンピュータのうち、メールサービスが利用可能なのは以下のマシンです。ホスト名とはネットワーク上の各コンピュータの名前です。以降各マシンはホスト名で表現します。

機種名	ホスト名	
SPARCcenter2000	cc2000	計算機センター 1 階に設置の Sun 社製コンピュータ。
DEC-3300	csosf01~40	2 号館 4 階 21 情報処理教室に 40 台設置の DEC 社製コンピュータ。
NeXTstation	ccns001~015	計算機科学研究所 3 階 C3 情報処理教室に 15 台設置の NeXT 社製コンピュータ。
DEC-3500	ksuvx1	2 号館 1 階に設置の DEC 社製コンピュータ。

以上のマシンでメールを利用する場合、メールアドレスはそれぞれ以下の通りとなります。

ホスト名	メールアドレス
cc2000 および csosf01~40	username@cc.kyoto-su.ac.jp
ccns001~ccns015	username@ccnext.kyoto-su.ac.jp
ksuvx1	username@ksuvx1.kyoto-su.ac.jp

上記のうち@より左の **username** には各ユーザのユーザ名を書きます。例えば神山太郎さんは taro というユーザ名で cc 環境にユーザ登録されているとすると、cc2000 や csosf シリーズのマシンでメールを扱うためのメールアドレスは

`taro@cc.kyoto-su.ac.jp`

となります。また、太郎さんは ccns シリーズも使っています。ユーザ登録情報については cc2000, csosf シリーズと ccns シリーズのマシン群は全て共通なのですが、メールアドレスについては ccns シリーズだけ別個になります。太郎さんが ccns シリーズのマシンでメールを扱うためのメールアドレスは

`taro@ccnext.kyoto-su.ac.jp`

です。

ところで太郎さんは上記のように二種類のメールアドレスを持っています。複数のマシンにユーザ登録しているから複数のメールアドレスを持つようになるわけで、これは自然なことです。例えば太郎さんが二箇所に部屋を借りているような状態だと想像すれば良いでしょう。しかし太郎さんにメールを送る場合には、上記のうちのどれに送るのが妥当かは送る前に太郎さんに電話なり手紙なりで確認する必要があります。つまり太郎さんは二箇所に部屋を借りていて、二つのメールボックスを持っているが、そのどちらを毎日チェックしているかは誰にも想像出来ないからです。これについては次の「相手のメールアドレス」にもっと詳しく書きます。

ところでよくよく考えてみると太郎さんは cc2000, csosf01~40 マシン群及び ccns001~ccns015 マシン群と合計すると数十台に及ぶ非常に多くのマシンに登録されていることとなりますね。しかしそんなにたくさんメールボックスがあっては面倒なので、cc2000 と csosf01~40 マシン群はメールボックスを共有しており、`username@cc.kyoto-su.ac.jp` 宛てに送られたメールは、この共用のメールボックスに配送されます。故に cc2000 及び csosf01~40 のいずれのコンピュータを用いてメールボックスをチェックしても、配

送られてきたメールを読むことが出来ます。どのコンピュータからメールボックスをチェックしたとしても、結果的には共通の、たった一つのメールボックスを見ている事になるからです。

また、ccns001～ccns015マシン群もメールボックスを共有しており、`username@ccnext.kyoto-su.ac.jp`宛てに送られたメールは、この共用メールボックスに配送されます。つまり計算機科学研究所3階に並んでいる15台のNeXTのどれに座っても`username@ccnext.kyoto-su.ac.jp`宛てに送られてきたメールを見ることが出来ます。

共用メールボックスをcc2000, csosf01～40マシン群と、ccns001～ccns015マシン群とに分けている理由はccnsマシン群が扱うNeXTメール⁶をcc2000, csosfマシン群が扱えないからです。

5.2.3 相手のメールアドレス

いざメールを誰かに送ろうとした場合、相手のメールアドレスが必要になるでしょう。相手のアドレスを調べるには、その相手に聞くしか方法がありません。例えば神山太郎さんが上記の計算機センター管理のコンピュータ、cc2000やcsosf01～40にhanakoでユーザ登録されている神山花子さんにメールを出そうとした場合、`hanako@cc.kyoto-su.ac.jp`宛てにいきなり出すのは好ましくないと言えます。コンピュータシステムは正直にcc2000及びcsosf01～40コンピュータの為のhanakoユーザの共用メールボックスに太郎さんのメールを配送するでしょうが、花子さんが本当にcc2000などのマシンでメールボックスをチェックしてくれるとは限らないからです。ひょっとしたら花子さんは電子メールなど全く使っていないかもしれません。ひょっとしたら、花子さんは他のコンピュータにも何らかの名前で登録されておりメールはもっぱらそこで利用しているかもしれませんね。

では太郎さんが花子さんにメールを送る場合はどうすれば良いのでしょうか？結論は「君に今後メールを送りたいのだけれど、いったいどこに送れば良いの？」と最初の一回目に（勿論メール以外の方法で）聞くことです。馬鹿馬鹿しいようですがこれは非常に重要なことです。

5.2.4 自分のメールアドレス

自分のメールアドレスが何であるかは自分がユーザ登録されているコンピュータの管理者に確認するのがいいでしょう。但し計算機センターが管理しているコンピュータについてのメールアドレスに関しては先に述べた通りです。

ある人にメールを送って欲しいと思った場合は、どこ宛てに送って欲しいか、すなわち自分が日常的にチェックしているメールアドレスを相手にはっきり通知することが大切です。これは先に書いた相手のメールアドレスをはっきり聞く、という事の裏返しです。同じく非常に重要なことです。

5.2.5 さあ、本番！

準備体操はこのくらいにしておきましょう。実際にメールを読んだり書いたりするのは現在ではコンピュータを操作すると言う事にほかなりません。即ち普遍的に「こうすればメールを読み、書く事が出来る」という方法はなく、そのコンピュータ独自の操作方法を修得する事になります。

次に計算機センターが運用しているcc環境のUNIXコンピュータを利用してメールを読み書きする方法について説明します。

⁶NeXTメールとは絵や音や様々なデータを簡単な操作でメールに含ませて送ることが出来るシステムで、現在この機能はNeXTコンピュータでしか利用出来ません。cc環境ではccns001～ccns015までのマシンでNeXTメールが扱えるという事です。この使い方などについてはNeXTのマニュアルを参照してください。

⁷当然関西弁でも可

5.3 MHE : Mule による電子メールの読み書き

MHE⁸はMuleを利用して電子メールを読み書きする機能を提供します。Muleと共に働きますから、Muleの操作方法についてある程度理解していることを前提に説明します。

5.3.1 はじめに

ここでは以下の流れに従ってメールを扱う方法を説明します。

- メールを読む
- メールを書く
- 来たメールの返事を書く
- メールの整理

この流れの通り、まずメールを読む方法を説明したいところなのですが、ちょっと問題があります。つまり恐らくあなたはまだ誰からもメールを送ってもらっていないので、メールを読む練習をするにも、読むべきメールが届いていないだろうと言うことです。

そこでまず練習のために、自分自身にめがけて実験メールを送るコマンドを紹介します。`mailself` コマンドです。

```
csosf01(81)% mailself
csosf01(81)%
```

`mailself` コマンドは実行に数秒以上掛かります。うまく実行出来た場合は、上記のように何もメッセージを表示せずに終了し、プロンプトが返ってくるでしょう。これで一通、新しいメールがあなた宛に届いているはずですが、次の節からは、そのメールを読むことで練習して行きましょう。

MHEの全ての操作はMule上で行います。さあ、`mule` コマンドでMuleを起動して下さい。

⁸名前の由来は「Emacs front end to the MH mail system」から来ています。`mh`についてはここでは説明しません。`man mh`でマニュアルが用意されていますが、特に理解する必要はありません。Emacs（「いーまっくす」と読みます）とはMuleの古いバージョンです。

5.3.2 メールを読む

メールを読むためには、Mule が起動されている状態で `M-x mh-rmail <Return>` とします。



図 5.1 mh-rmail の起動

新規にメールが届いている場合はこれで以下のような画面表示となるでしょう。

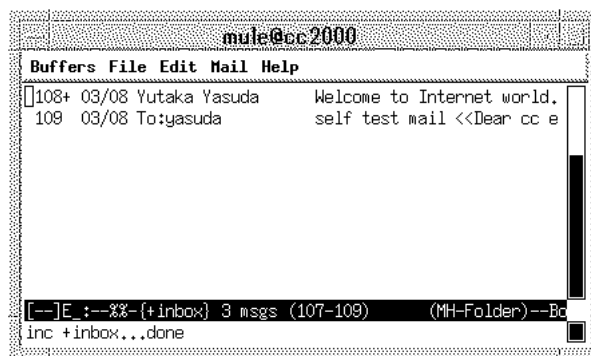


図 5.2 メールの一覧表示

到着したメールが一覧として一通一行の形で表示されているでしょう。カーソルは一覧表示の左側にあるはずですが、ここでカーソルを上下させて、自分が読みたいメールの行の左にカーソルを移動させます。カーソルの上下は Mule 上でのファイルの編集の際のカーソルの上下と同じです。つまり上（一つ前の行）に移動したいときは上矢印（↑）もしくは `C-p` キー、下に移動したいときは下矢印（↓）もしくは `C-n` キーです。 `M-<` や `M->` で一番先頭や末尾の行への移動が出来ます。

自分が読みたいメールの行の左にカーソルを移動させて「. (ピリオド)」を押せば、そのメールの内容が Mule のウィンドウを二分割して下半分に表示されると思います。

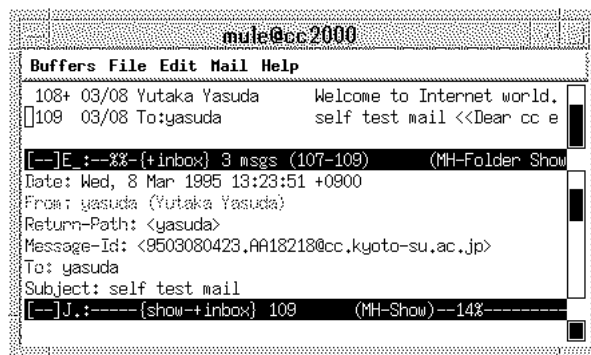


図 5.3 メールの内容の表示

一つのメールの行が長すぎて Mule のウィンドウにおさまらないうときは <Space>キーで一画面分スクロールさせる事が出来ます。スクロールして過ぎてしまった部分の内容を巻き戻してみたい場合は <Delete>キーです。

用語説明

まず一覧に現れる情報について少し説明しておきます。以下に一覧表示の例を挙げます。

```
19+ 03/17 Yutaka Yasuda      Welcome to Internet world. <<Dear friends,
```

左から順番に項目別に説明して行きます。

19	順番に振られる番号です。番号が新しければ新しいほど昔の、古いメールです。
+	現在このメールに注目しているよという印です。
03/17	日付です。
Yutaka Yasuda	メールの送り元、つまり誰から来たかを示しています。 mailself コマンドで送り付けたテストメールであれば恐らくここは To: に続いてあなたのユーザ名が来るでしょう。
Welcome to Internet world.	メールの表題です。
<<	ここから内容の一部だよという印です。
Dear friends,	恐らくまだ続きがあるはずですが、これがそのメールの本文の先頭部分です。表題だけでメールの内容の想像が付かないときなどは重宝します。

上記の例での Welcome to Internet world.、つまり表題はメールの世界では Subject と呼ばれ、本文とは少し区別して扱われます。つまりメールは主に「宛名」「送り元」「Subject」「本文」の4つの部分からなっているというわけです。

メールの一覧表示とはつまりこの主たる4つの部分のうち、宛名を除いた三つを並べているという事です。この一覧が並んでいる状態を、フォルダモードと呼んでいます。

さて、今度はメールの内容を読んでいる時に表示される部分の初めの方に、必ず付いてくる To: などに導かれた数行に注目してください。これらはメールのシステムが付けたもので、そのメールの補助的な情

報が記録されています。一般の郵便で言うと表書きや消印の情報に相当します。この部分をメールのヘッダと呼んでいます。以下にメールヘッダの例を挙げます。

Date: Mon, 21 Mar 1994 12:53:31 +0900
From: Tanaka Hideki <tanaka>
Return-Path: <tanaka>
Message-Id: <9501130246.AA07556@cc.kyoto-su.ac.jp>
To: yasuda
Subject: Re: NOMIKAI again
Cc: tanaka, ryo, omatsu

以下に順番にそれぞれの行ごとに説明して行きます。

Date:	このメールが書かれた日付です。
From:	誰から送られてきたかを示しています。ここでは tanaka さんです。Tanaka Hideki は tanaka というユーザ名の人のフルネームを示しています。
Return-Path:	返事を書いた場合の宛先になりますが、気にしないで下さい。
Message-Id:	これまた気にしないで下さい。
To:	誰宛に送られた手紙かを示しています。ここでは yasuda さん宛ですね。
Subject:	表題です。これはこのメールを書いた人が付けたものです。既に説明しましたね。
Cc:	Carbon Copy を意味しています。Carbon Copy とは複製を意味していて、このメールの複製を誰宛に同時に送ったかを示しています。ここでは tanaka, ryo, omatsu の 3 人に送っています。

次のメールを読む

図 5.3 のようにメールの内容が表示されている状態で、(もしあったとして) 次 (もしくは一つ前) のメールを読むには幾つかやり方があります。

- 図 5.3 の状態のまま、カーソルを一つ下 (もしくは一つ上) に移動して、そこでもう一度「. (ピリオド)」で内容を読む。
- 図 5.3 の状態のまま、**n** キーを押して次のメール (もしくは **p** キーを押して一つ前) の内容をすぐに表示させる。

MHE を終る、再起動する

メールを読み終って、普通の Mule の操作に戻りたいと思ったときは **q** キーを押します。これで MHE を起動する前、つまり図 5.1 の状態に戻ります。Mule を終りたい場合はいつも通りに **C-x C-c** です。再びメールを読みたいと思った時は単にもう一度 (必要ならば) Mule を起動し、**M-x mh-rmail<Return>** とするだけです。

以前読んでしまったメールを再び読み返す

MHE を再起動してまたメールを読みたいと思っても、**mh-rmail** はまだ一度も読んでいないメールの一覧を表示しようとするから、前回メールを読んでから新しいメールが来ていなければ「読んでいないメールは一つも無いよ」と言ってメールを一行も表示しない時があります。あるいは以前に貰っているメールはもっとたくさんある筈なのに僅かしか表示しない時もあります。

そこで以前に読んでしまったメールをもう一度読み返したい場合は `mh-rmail` を実行してから `M-r` とします。以前に読んでしまったメールをさかのぼって表示してくれます。とりあえず最近の数通が一覧表示されます。それ以前のメールが見たい場合は `C-p` もしくは上矢印 (↑) キーなどでカーソルを上へ移動させてさかのぼってあげればよいのです。状況によっては `C-p` しても何の変化もない場合があります。その場合でも実は画面のさらに上に古いメールが持ってこられている場合がありますから、確認のためとにかく `C-p` してみると良いでしょう。

5.3.3 メールを書く

さて、メールを読むことが出来るようになったら今度はメールを書いて送ってみましょう。一番良いのは誰か知人に相手になって貰うことですが、それが出来ないようならまず自分自身宛に送って、その結果をチェックするのがよいでしょう。

新たにメールを書いて発信するには Mule が起動されている状態で、`M-x mh-smail` とするか、`mh-rmail` を実行してメールの一覧が表示されている図 5.2 の状態で `s` もしくは `m` とします。

するとまずエコーラインに `To:` と表示され、宛先を問うて来ます。ここでメールを送る相手のメールアドレス (5.2.3 で説明したことに注意して下さい) をタイプします。<Return>すると今度は `Cc:` と表示され、Carbon Copy が必要かどうかを聞いてきます。Carbon Copy とは複写のことで、そのメールの複写を控えとして送ることを差します。もしもあなたがそのメールの Carbon Copy をどこかに (例えば自分自身にでも) 送りたいのであれば `Cc:` の問いかけに対してメールアドレスをタイプすることで答えます。アドレスは幾つでも並べて書けますので、複数の相手に Carbon Copy を送ることも可能です。並べて書くときはアドレスとアドレスの間に「, (カンマ)」で区切りを入れてやらなければならないことに注意してください。もしも Carbon Copy が必要無ければ単に <Return> してください。

今度は `Subject:` と表題を聞いてきますので、何かわかりやすい表題をタイプしてください。Subject には漢字やかなは使わず、アルファベットと数字、記号程度で表現してください⁹。

以上の `To:`, `Cc:`, `Subject:` をタイプし終ると画面表示が変わり、手紙の内容を書くウィンドウが用意されます。

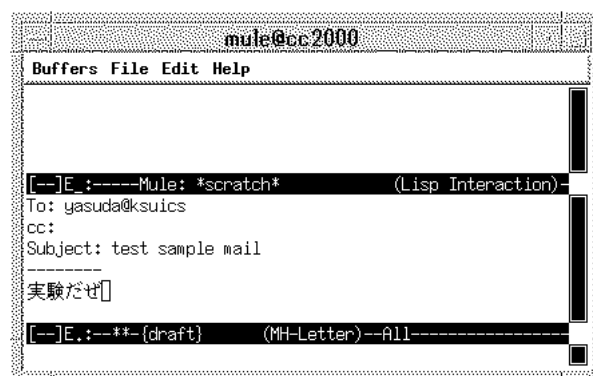


図 5.4 メールの内容を書く

この状態をレターモードと呼んでいます。 `To:`, `Cc:`, `Subject:` のすぐ下に ----- という行がありますが、これ以降に本文を書いてください。この --- 行は MHE システムが必要とするもので、削除しては

⁹ 最近 Subject にも漢字が利用できる場合がありますが、これは相手が Subject に漢字を適用できるシステムを持っているか、いかに依存しますからそれが確認できない限りは漢字は使わない方が無難です。因みに現在の cc 環境の MHE は漢字の Subject には対応していません。

いけません。この状態で普通に Mule によってファイルの内容を編集するときと同じ様にメールの本文を編集する作業が出来ます。Mule の操作に習熟すれば、メールの内容としてどこかのファイルの内容を取り込んだり、さまざまな応用が利くようになるでしょう。

そうやってメールの本文を書き終わったら、C-c C-c でメールが発信されます。反対にメールを書いている途中で、そのメールを出したくなくなった場合は C-c C-q とします。するとエコーラインに以下のような確認の為の質問をしてきますので、y と答えてください。

Kill draft message? (y or n)

これで、そのメールを破棄することが出来ます。

5.3.4 来たメールの返事を書く

来たメールに対する返事を書くには、返事を書こうとしているメールを図 5.3 のような状態で読んでいるときに a キーを押します。すると、そのメールに関係している人の誰に対しての返事を書くのかをエコーラインに以下のようなメッセージを表示して質問してきます。

Reply to whom:

この質問に対しては以下のいずれかで返事をします。

返答	意味
from	そのメールの差出人、つまり From: に書かれている人に返事を送る。
<Return>	from に同じ
to	そのメールの差出人、つまり From: に書かれている人を To: に、そのメールの To: に書かれている人を Cc: に指定する。
cc	そのメールの差出人、つまり From: に書かれている人を To: に、そのメールの To: 及び From: に書かれている人を Cc: に指定する。
all	そのメールの差出人、つまり From: に書かれている人を To: に、そのメールの To: 及び Cc: 及び From: に書かれている人を Cc: に指定する。

上記の表のいずれかによって返答すると図 5.4 に非常に良く似た状態になり、返事のメールの本文が書けるようになります。(これもレターモードです。)

まだ誰からもメールを貰っていない時にこの返事を書く練習をするには、先ほど自分宛に送ったメールに自分で返事を書くのが良いでしょう。この場合は書かれた返事は再び自分宛に送られることとなります。このように自分が出したメールに返事を書く場合には Reply to whom: の問い合わせに対しては必ず to で答えるようにして下さい。<Return>や、from で返答すると、返事を書く為の画面が正しく生成されず送信に失敗する時があります。

電子メールでの返事の作法

普通の手紙とは違って、電子メールでは返事を書くときに前回の手紙の内容を引用することが簡単にできます。以下の例のように引用と返答とを交互に混ぜながら返事を書くことと判りやすいメールとなることでしょう。

田中さん> 先日は飲み会お疲れ様でした。
いえいえ。ありがとうございました。

田中さん> そこで話しておりました原稿の件ですが、酒の上の事とは言え、冗談と言う

田中さん> わけでもなく、本当にお願いしたいと思います。

ぎょぎょ。やはり本気だったのですね。(^_^ ;)

ちょっと怖い気もしますが、了解しました。引き受けさせていただきます。

このような感じでしょうか。以下にこの種の引用を行なう手順を説明します。

返事を書こうとしているときに C-c C-y とすれば、そこに元となるメールの内容が引用されます。引用の際は自動的に各行初めの部分に「この行は引用だよ」ということを示すための記号を付けようとして、以下の質問をしてきます。

Complete cite attribution string: (default Hideki)

このの部分には元のメールを書いた人の名前の一部が表示されていると思います。例では田中秀樹（たなかひでき）さんの Hideki が表示されています。ここでは自分の好みの引用記号をしましょう。漢字も使えますのでここでは 田中さん<Return>としてみましよう。単に <Return>とすると最初に表示されていた Hideki が採用されます。

[あ]Complete cite attribution string: (default >) 田中さん

すると以下のように聞いてきます。

登録しますか? (y or n)

ここで登録すると次回からこの人から来たメールに対する引用記号として「田中さん」を最初に表示します。y と答えると、以下のように聞かれる場合があります。

前の登録が消えますがいいですか? (y or n)

これは以前「田中さん」ではない別の記号で登録されていたのだけれど、それを「田中さん」に置き換えて良いかという確認です。これにも y と答えれば良いのです。

さて、次に引用元の記事を表示しているウィンドウとは別のウィンドウが開いて

```
>>>> On Wed, 08 Mar 1995 21:09:34 +0900, Environment Test Worker <worker> said\  
:
```

と表示されると同時にエコーラインに

Showing reference header 1.

と表示されます。これは引用を開始する部分の先頭に、どのような記述を含めるかという質問です。n キーを押すことで **Showing reference header** の数字が 0~6 まで変わり、ウィンドウに表示されるスタイルも変化しているのが判るでしょう。気に入った表示が見つければそこで<Return>してください。すると返事を書こうとしているところに指定した引用記号を付けて元のメールの内容が現れるでしょう。但しそのままではメールヘッダも本文も全ての部分が引用されますので、かなり不要な部分も含まれている可能性があることに注意してください。

こうやって引用された元々のメールのうち、今回のあなたのメールでその返答となる部分以外の行を消去して、先に挙げた例のように引用と返答とを交互に混ぜながら返事を書くこと判りやすいメールとなることでしょう。

5.3.5 メールの整理

メールをしばらく利用していると、送られてきたメールがたくさん溜ってきて邪魔になります。不要なメールは削除することが出来ます。図 5.2 の状態で不要になったメールの行の左にカーソルを移動し、ここで「. (ピリオド)」ならそのメールの内容が読める、と言う状況でピリオドの代わりに **d** をタイプします。するとメールの一覧表示の番号のすぐ右に **D** がマークとして表示されます。この段階は単に削除すべきメールにマークを付けているだけで、実際の削除はまだ行われていません。どんどん削除したいメールにマークを付け続けて行くことが出来ます。マークを付け間違えた場合は、**D** マークの行にカーソルを移動して今度は **u** キーをタイプすれば **D** マークが消え、マークが外すことができます。

マークされたメールを実際に削除するには **x** もしくは **e** キーです。問い合わせなどはなく、すぐに削除を実行します。

5.3.6 メールが来ているかどうか確認する

ずーっと Mule を利用している人ならば、時々隙を見て `M-x mh-rmail<Return>` してメールを確認すれば良いのですが、メールが来ているかどうかを確認するためにわざわざ Mule を起動するのが面倒だと言う人のために、メールが来ているかどうかを簡単に確認する方法を紹介します。`from` コマンドです。`from<Return>` です。以下に例を示します。

```
csosf01(88)% from
From yasuda Tue Mar 22 18:51:30 1994
From yasuda Tue Mar 22 18:51:51 1994
csosf01(89)%
```

上記のように `from` コマンドは未読のメールがあれば、一通あたり一行で「誰からか」「いつ届いたか」を表示します。もしも未読のメールが一つもなければ `from` コマンドは何も表示せず以下のように終了します。

```
csosf01(88)% from
csosf01(89)%
```

`from` コマンドで未読メールが見つかった場合だけ、Mule を起動して MHE で読めば良い、と言う事ですね。

5.3.7 メールの実体はどこに？

MHE は各ユーザのホームディレクトリのすぐ下に `Mail` というディレクトリを用意し、メールをそこに保存しています。普通に操作をしていけば、MHE は全て `inbox` というところにメールを溜め込むのですが、その実体は `Mail` ディレクトリの下にある `inbox` というディレクトリです。そこに一通のメールを一つのファイルとして保存しています。ファイル名には順番に付けられた番号が使われています。

具体的には `~/Mail/inbox/1` などという名前でメールが残っているはずですが、自分で確かめてみると良いでしょう。

5.3.8 トラブルからの脱出

MHE を起動しようとするとき `.mh-profile` がない、というエラーメッセージが表示されるんですけど

`Cannot find MH profile /NF/home/..(略)../.mh_profile` というメッセージがエコーラインに表示され、MHE が起動できない場合があるかも知れません。このような場合は一度 Mule を終了してから、お

まじないとして `inc` コマンドを一度だけ実行してください。何か問い合わせをしてくるかも知れませんが、その時は `y` と答えてください。それでもう一度 Mule を起動し、それから MHE を起動してみてください。

5.3.9 MHE もつともつと

ここに紹介したのは MHE の機能のうちの基本的なものだけです。MHE にはもっと様々な機能がありますが、それについてはここでは説明しません。附録に参考文献を挙げておきますので、それらを参照して下さい。

5.4 メールを書くときの注意

ここでは操作方法ではなく、メールの中身の書き方について説明します。

初めてメールを出すときに

始めのうちはいきなり学外や海外にメールを出さずに、学内でメールの練習をしてから送って下さい。練習の相手が見つからない場合は計算機センターに相談してください。

あまり大きなファイルを送らないこと

メールによってファイルを転送することも出来ますが、、

- どうしてもメールによってファイル転送する場合は目安として 50 キロバイト¹⁰/メール以下にしましょう。あまり大きなファイルを送るのは、参加組織に迷惑がかかります。
- どうしてもメールによって大きなファイルを送らなければならないなら、1 メガバイトくらいまでなら分割して送っても大丈夫かもしれません。それ以上ならフロッピーやテープで送りましょう。

そのメールは相手を読めるものですか？

現在のメールがコンピュータによって処理されている限り、自分が利用しているコンピュータによって処理出来た文書が、相手が利用しているコンピュータによって復元出来るとは限らない事に注意しましょう。

- 漢字を含むメールが必ず相手に読めるとは限りません。相手はひょっとしたら漢字が表示出来ないコンピュータを使ってメールを読んでいるかもしれません。相手が漢字を読めるかどうか、まず最初に確認するのがよいでしょう。
- 基本的に Internet は JIS コード 漢字を利用していますが、あなたが Internet を利用する窓口になるコンピュータによって漢字コードはまちまちです。たとえば計算機センターが管理している cc2000, csosf01~40, ccns001~015 に限っては標準の漢字コードとして EUC コードを採用していますが、メールは JIS コードに変換されて送出されるように設定されています。どの漢字コードを適用すれば良いのか判らない場合はホストコンピュータの管理者に確認してください。
- Subject に漢字は使わないで下さい。
- 使ってはならない文字として、半角カナ（カナ・キーを押して入力したもの）やメーカー独自の文字（(株),I,II,III,IV,V 等の文字が 1 文字で表されているもの、丸の中に文字が書いてあるものなど）があります。また、使わない方が良い文字としては、罫線素片等があります。

¹⁰ バイトというのはコンピュータ上の資源の大きさを表す単位です。50 キロバイトと言えば漢字にして 25000 字、つまり原稿用紙 50 ページ強辺りとなります

- NeXT メールを送るときは相手が NeXT メールをそのメールアドレスで扱えることを確認してください。NeXT メールは音や絵をメールに含ませる事が出来ますが、相手も NeXT コンピュータを利用している必要が有ります。NeXT メール特有の注意事項については別紙に詳しく説明が有りますので、NeXT コンピュータでメールを利用される方は一度読んでください。

メールの信頼性について

送ったメールは相手に確実に届くことも、内容の完全な秘匿性も保証されていません。到着の確実性が要求される場合には、相手にそのメールが届いたら折り返し届いた旨をメールしてもらうように頼みましょう。ある程度の秘匿性が必要ならば暗号化するかまたは直接手渡しして下さい。

一行の長さについて

メール配送を行うコンピュータのシステム自体は一行 255 文字までサポートしています。しかし、そのメールの読み易さや返事を書く時に内容を参照することも考えて、漢字で約 35 文字以下、アルファベット(半角文字)で約 70 文字以下にするのが良いでしょう。

海外へのメールについて

国内以上にアドレスや送るメールの大きさの注意が必要です。また、海外のネットワークにはそれぞれのネットでの取り決めがあるので、それに従うようにしましょう。

メールアドレスに工夫が必要な場合もあります。一般的には宛先のアドレスは `user@aaa.bbb.ccc` のように `.jp` で終らないものの、国内と同様の書式です。(user には相手のユーザ名を、aaa,bbb,ccc 等には相手のアドレスを入れます)

著作権および責任の所在

書いたメールに関する責任は、書いた人の属する組織にあるのではなく、書いた人自身ににあります。また、著作権はそのメールを書いた人にあります。

違法行為の禁止

例えば、メールでのソフトの違法な流通などをやってはいけません。

5.5 GNUS : Mule によるニュースの読み書き

GNUS は Mule を利用してニュースを読み書きする機能を提供します。

Mule と共に働きますから、Mule の操作方法についてある程度理解していることを前提に説明します。

5.5.1 はじめに

ここでは以下の流れにしたがってニュースを扱う方法を説明します。

- GNUS の起動
- ニュースの記事を読む
- ニュースグループの選択
- ニュースの記事を投稿する
- ニュースの記事にフォローする
- 投稿した記事のキャンセル
- ニュースの記事にメールで返事をする
- 古い記事を読み返す
- ニュースの記事の保存

GNUS の全ての操作は Mule 上で行います。さあ、`mule` コマンドで Mule を起動してください。

5.5.2 GNUS の起動

ニュースを読むために、まず GNUS を起動します。Mule が起動されている状態で `M-x gnus<Return>` とします。

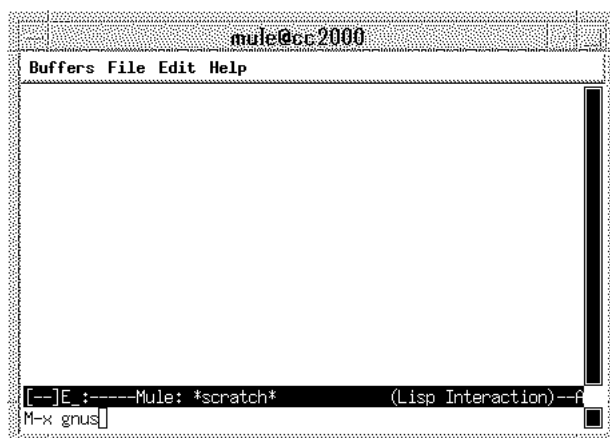


図 5.5 GNUS の起動

最初の GNUS の起動には 1 分以上かかる場合があります。これは全ての情報をゼロから構築しているためで、次回からはもう少し短い時間で起動できるようになるでしょう。以下のような状態になるまで、しばらくお待ちください。



図 5.6 ニュースグループ一覧

5.5.3 記事を読む

図 5.6 のように、一つのニュースグループが一行で表わされ、一覧として表示されているでしょう。カーソルは一覧表示のニュースグループの名前の左側にあるはずで、ここでカーソルを上下させて、自分が読みたいニュースグループの行にカーソルを移動させます。カーソルの上下は Mule 上でのファイルの編集の際のカーソルの上下と同じです。つまり上（一つ前の行）に移動したいときは上矢印（↑）もしくは C-p キー、下に移動したいときは下矢印（↓）もしくは C-n キーです。M-< や M-> で一番先頭や末尾の行への移動が出来ます。

このニュースグループの一覧が表示されている状態を「グループモード」と呼んでいます。

おそらく今は `sandai.` で始まる産業大学ローカルニュースグループだけが一覧に表示されるはずで、以下に'95.3 現在のローカルニュースグループの一覧を載せておきます。

内容	ニュースグループ名	期限
簡単な、もしくは一般的な質問	<code>sandai.question</code>	二カ月
娯楽及び趣味に関する話題	<code>sandai.rec</code>	二カ月
その他の、もしくは一般的な話題	<code>sandai.misc</code>	二カ月
下らない話題	<code>sandai.junk</code>	二週間
お知らせ、速報（一週間以内有効）	<code>sandai.news.week</code>	一週間
お知らせ、速報（一カ月以内有効）	<code>sandai.news.month</code>	一カ月
産業大学作品展示会場	<code>sandai.gallery</code>	
実験用ニュースグループ。貴方自身の練習用に	<code>sandai.test</code>	一週間
コンピュータに関する話題	<code>sandai.comp</code>	
学内コンピュータ環境に関するお知らせ	<code>sandai.comp.announce</code>	
コンピュータに関する管理者向けの話題	<code>sandai.comp.admin</code>	
授業に関する話題	<code>sandai.classes</code>	
授業に関する話題 (特に??クラスについて)	<code>sandai.classes.??</code>	
産業大学のニュース購読者全員へのお知らせ	<code>sandai.announce</code>	
産業大学の一般的な話題	<code>sandai.general</code>	
計算機センターからのお知らせ	<code>sandai.center.announce</code>	
計算機センターへの一般的な話題	<code>sandai.center.general</code>	

各ニュースグループに投稿された記事は、上記の期限を過ぎると自動的に消去されます。期限の書かれていないニュースグループの記事はそのまま残り続けます。この期限の設定を含めてニュースグループの設定は流動的なものです。最新の情報が知りたい場合は 4.7で紹介した `ccinfo` コマンドを利用するか、`sandai.question` に質問を投稿するなどして下さい。

さて、グループモードで、自分が読みたいニュースグループの行にカーソルを移動させて<Space>キーを押せば、そのニュースグループの記事の表題一覧と、そのなかの記事の内容を表示できる以下のような状態へと移行します。



図 5.7 記事の内容表示

この状態では Mule のウィンドウが二分割され、上半分に先ほど選んだニュースグループの記事の表題の一覧が、下半分にそのニュースグループの先頭の記事の内容が表示されています。下半分に表示されている記事の内容が長すぎて Mule のウィンドウに収まり切らないときは<Space>キーで一画面分スクロールさせることが出来ます。スクロールして過ぎてしまった部分の内容を巻き戻してみたい場合は<Delete>キーです。

次の記事を読む

今度は、ある特定のニュースグループの中の、一つの記事が一行で表わされ、一覧として表示されている訳です。カーソルは一覧表示の番号の右にあるはずですが、ここでカーソルを上下させて、自分が読みたいニュースグループの行にカーソルを移動させます。カーソルの上下は Mule 上でのファイルの編集の際のカーソルの上下と同じです。つまり上（一つ前の行）に移動したいときは上矢印（↑）もしくは `C-p` キー、下に移動したいときは下矢印（↓）もしくは `C-n` キーです。M-< や M-> で一番先頭や末尾の行への移動が出来ます。

自分が読みたい記事の行にカーソルを移動させて<Space>キーを押せば、記事の内容が Mule のウィンドウの下半分に表示されます。

また、記事の内容を読んでいるときに<Space>キーを押し続けて行けば、一つの記事を読み終った段階で自動的に次の記事に移動します。記事を読んでいる途中で `n` キーを押せば、今読んでいる記事の次の未読記事を表示します。`p` キーを押せば、今読んでいる記事の前の未読記事を表示します。

この、あるニュースグループを選択して、記事を次々と読んで行ける状態を「記事モード」もしくは「アーティクルモード」と呼んでいます。

`q` キーで記事モードから抜けてグループモードへ戻ります。図 5.6のような表示になってグループモードに戻った後は、また違うニュースグループを選んでその記事を読むことが出来ます。

5.5.4 GNUSを終了する、再起動する

記事を読み終って、普通の Mule の操作に戻りたいと思ったときはグループモードで `q` キーを押します。すると、エコーラインに以下のような表示が現れ本当に GNUS を終了するかどうか聞いてきます。

```
Are you sure you want to quit reading news? (y or n)
```

ここで `y` キーを押してやると GNUS を起動する前、つまり図 5.5 の状態に戻ります。Mule を終りたい場合はいつも通りに `C-x C-c` です。

再びニュースを読みたいと思ったときは単に `M-x gnus<Return>` とするだけです。

5.5.5 ニュースグループを選ぶ

読みたくないニュースグループを一覧表示から外す

グループモードでは、何も指示しなければ既に示したように `sandai` ニュースグループを全て表示します。その中にはあなたにとってほとんど意味のないニュースグループもあるでしょう。もう読まないと決めてしまったニュースグループはこの一覧に表示させないようにすることができます。図 5.6 のようなグループモードでカーソルを読みたくないニュースグループの行に移動し、そこで `u` キーを押します。するとその行の左端に `U` マークが付きます。`U` マークを付け間違ってしまった時は、もう一度その行で `u` キーを押すことによって外すことができます。幾つか `U` マークを付けていたら `l` (英小文字の `L`) キーを押してみてください。今まで `U` マークがついていた行が表示されなくなりましたね。

これで次から GNUS を使ってニュースを読む時に、全く読まないニュースグループの一覧が出てこなくなって随分楽になると言うわけです。ちなみに `u` は `Unsubscribed` の略で「購読しない」と言うことを意味しています。

一覧に出てこなかったニュースグループを一覧に加える

`cc` 環境ではあらかじめ多くのニュースグループが `Unsubscribed` されています。最初の利用であれば `sandai`. で始まる産業大学ローカルニュースグループだけが一覧に表示されるはずですが。

しかし挙げたものは `cc` 環境で利用できるニュースグループのほんの一部に過ぎません。もっと多くのニュースを読みたい! と思った人は一度 `Unsubscribed` されているニュースグループを全て表示させてみるのが良いでしょう。グループモードで `L` (今度は英大文字の `L`) キーを押してください。(少し時間がかかります。) 一覧に表示されている限り普通の操作でそのニュースグループの記事を読むことが出来ます。

全部で 6000 以上のニュースグループがありますので最初は何を読んでもいいのか迷うでしょう。そのうち `fj.` と `tnn.` で始まるニュースグループが日本語で書かれている主なものです。その他のほとんどのニュースグループは英語で書かれています。日本とアメリカ以外の各国のニュースを気が付いたところだけ並べてみました。参考にしてください。

<code>can.</code>	<code>uw.</code>	カナダ	<code>de.</code>	<code>zer.</code>	ドイツ	<code>dk.</code>	デンマーク
<code>fr.</code>		フランス	<code>il.</code>		イスラエル	<code>no.</code>	ノルウェイ
<code>nz.</code>		ニュージーランド	<code>swnet.</code>		スウェーデン	<code>uk.</code>	イギリス
<code>za.</code>		南アフリカ共和国					

ただこのままでは次に GNUS を起動した時にはまた一覧から外れてしまいますから、目的のニュースグループを一覧に表示させた状態で、そのニュースグループについている `U` マークを再度 `u` キーによって外してやれば良いのです。その効果はグループモードで `l` (今度は英小文字の `L`) キーを押すことで確認できますね。

Unsubscribed されているニュースグループの名前があらかじめ判っている場合は、もっと簡単に一覧に表示させることが出来ます。グループモードで j キーを押すと以下のようにニュースグループ名を聞きます。

Newsgroup:

ここで目的のニュースグループの名前をタイプして<Return>すると、それがニュースグループ一覧の一番上の行に表示されます。あとは普通の操作でそのニュースグループの記事を読むことが出来ます。次回の GNUS の起動後に再び一覧に表示させておきたい場合はそのニュースグループについている U マークを再度 u キーによって外してやる事を忘れずに。

5.5.6 記事を投稿する（けどちょっと待てよ）

せっかくニュースシステムが稼働しているのです。ただ読むだけで全く投稿しないのも面白くありません。ここは一つ何か書き込んでみましょう。

投稿する前にちょっと考えること

記事を投稿するのは GNUS で勿論可能なのですが、ニュースを読みはじめて間もなくの頃はとにかく読むのに徹して投稿は控えた方がいいと思います。これはネットワークニュースと言うものを良く理解しない内に「的外れな」「失礼な」もしくは「迷惑な」記事を投稿する危険があるからです。

ちょっと否定的に書きましたが基本的にはニュースも含めてネットワークサービスは参加しないと意味がありません。人の書いたものを読むだけでは面白さも半減です。あまり遠慮しすぎる必要もありません。投稿そのものには賛成します。ニュースを盛り上げるためにもぜひ投稿してください。

但しあなたがまだ投稿する事に慣れていない場合は、まず最初に `sandai.test` ニュースグループに投稿してみましょう。ここは投稿の実験や、練習の為のニュースグループですから操作に失敗しておかしなメッセージを投稿してしまっても誰の迷惑にもなりません。とりあえずここに自己紹介のメッセージでも投稿してみてもどうでしょうか。また、投稿はしばらくの間 `sandai` で始まるニュースグループに限定するのがお勧めです。つまり `fj` や `comp` など始まるインターネットを経由して外部に流れ出てしまうニュースグループには投稿しない方がいいかも知れません。そういうところで投稿に失敗したら、最悪の場合失敗した記事が世界じゅうを駆け巡ることになってしまい、大変迷惑です。しばらく投稿の操作や作法に慣れてからの方がいいでしょう。

何事も順を追ってやりましょう、と言う事です。

さあ投稿してみましょう

さてもう前置きは良いでしょう。何か記事を投稿してみましょう。記事の投稿には二つのケースが考えられます。

1. 全く新しい話題を投稿する
2. 人の記事に対する意見やコメントを投稿する。

ニュースをいくらか読んでいるうちに、この二種類の記事に遭遇すると思います。誰かが質問の記事を投稿し、他の誰かがそれに答えたりしているでしょう¹¹。

まず先に 1. のケースについて説明します。次に 2. のケースについて説明します。2. のケースのように人の記事に対して意見を加えて投稿することをフォローすると呼んでいます。

¹¹え？見たことがない？それは余りに経験が少なすぎます。もうちょっとニュースを読み込んでみてから投稿してはいかがでしょう？

5.5.7 新しい記事を投稿する

新しい記事を投稿するにはグループモードもしくは記事モードで **a** キーを押します。すると以下のようなメッセージをエコーラインに表示して確認をします。y で答えます。

Are you sure you want to post to all of USENET? (y or n)

もしグループモードで **a** キーを押した場合は、以下のようなメッセージをエコーラインに表示してどのニュースグループに投稿するのか聞いてきます。記事モードで **a** キーを押した場合は、キーを押した時のニュースグループが投稿先のニュースグループとなりますので、聞いてくることはありません。

Newsgroup:

これにはポストする記事のニュースグループをタイプします。するともう一度同じように **Newsgroup:** と聞いてきますので、単に **<Return>** とします¹²。

すると今度は以下のようなメッセージをエコーラインに表示して投稿する記事の表題を聞いてきます。

Subject:

これには何かわかりやすい表題をタイプしてください。Subject には漢字やかなはつかわず、アルファベットと数字程度で表現してください¹³。

すると今度は以下のようなメッセージをエコーラインに表示して投稿する記事の配布範囲を聞いてきます。

Distribution: local

この Distribution は、投稿する記事の配布範囲を示しています。以下のいずれかを答えるようにします。

配布範囲	ニュースグループ	意味
local	sandai.general など	京都産業大学内のローカルニュースグループ用。京都産業大学以外に配布されることはない。
fj	fj.jokes など	国内のニュースグループ用。fj など日本語のニュースの配送を受けている地域以外に配送されることはない。
world	comp.sys.sun など	国際ニュースグループ用。comp, sci, soc など世界じゅうに配布される。

あらかじめ local と表示されていますから、local で良い場合はそのまま **<Return>**、変更したい場合は local の後ろにあるカーソルを **<Delete>** キーで戻して fj などとタイプします。ここまで答えると Mule のウィンドウの状態が変わって、記事の内容を編集できるようになります。

¹² ここで複数回聞いてくるのはクロスポストという操作方法の為なのですが、ここでは説明しません。

¹³ 最近 Subject にも漢字が利用できる場合がありますが、これは相手が Subject に漢字を適用できるシステムを持っているか、いないかに依存しますからそれが確認できない限りは漢字は使わない方が無難です。因みに現在の cc 環境の GNUS は漢字の Subject には対応していません。



図 5.8 記事を書く

ここで記事を書くわけですから、**Newsgroups:**、**Subject:**、**Distribution:** のすぐ下に

```
--text follows this line--
```

という行がありますが、これ以降に本文を書いてください。この行はGNUSシステムが必要とするもので、削除してはいけません。この状態で普通にMuleによってファイルの内容を編集するときと同じ様にニュースの本文を編集する作業が出来ます。Muleの操作に習熟すれば、記事の内容としてどこかのファイルの内容を取り込んだり、さまざまな応用が利くようになるでしょう。

そうやって記事の本文が書き終わったら、**C-c C-c** で記事が発信されます。反対に記事を書いている途中で、その記事を投稿する気がなくなった場合は、**C-c C-k** とすれば、その記事を破棄することが出来ます。

5.5.8 記事にフォローする

誰かが書いた記事を読んでいて、その記事に対して意見を添えたり、質問をしたり、疑問に答えたりするような記事を投稿する事をフォローと呼んでいます。フォローを行うには、その元となる記事を読んでいる状態で**F**キーを押します。

すると、単に新たな記事を書いて投稿するときと同じ様に以下のようなメッセージをエコーラインに表示し確認してきます。

```
Are you sure you want to followup to all of USENET? (y or n)
```

これにも**y**と答えてやります。すると元となる記事の内容を引用するための記号を付けようとして、以下の質問をしてきます。

```
Complete cite attribution string: (default Hideki)
```

このの部分には元の記事を書いた人の名前的一部分が表示されていると思います。例では田中秀樹(たなかひでき)さんのHidekiが表示されています。ここでは自分の好みの引用記号をしましょう。漢字も使えますのでここでは田中さん<Return>としてみましよう。単に<Return>とすると最初に表示されていたHidekiが採用されます。

```
[あ]Complete cite attribution string: (default >) 田中さん
```

すると以下のように聞いてきます。

登録しますか? (y or n)

ここで登録すると次回からこの人から来た記事に対する引用記号として「田中さん」を最初に表示します。y と答えると、以下のように聞かれる場合があります。

前の登録が消えますがいいですか? (y or n)

これは以前「田中さん」ではない別の記号で登録されていたのだけれど、それを「田中さん」に置き換えて良いかという確認です。これにも y と答えれば良いのです。

さて、次に引用元の記事を表示しているウィンドウとは別のウィンドウが開いて

```
>>>> On Wed, 08 Mar 1995 21:09:34 +0900, Environment Test Worker <worker> said\  
:
```

と表示されると同時にエコーラインに

Showing reference header 1.

と表示されます。これは引用を開始する部分の先頭に、どのような記述を含めるかという質問です。n キーを押すことで **Showing reference header** の数字が 0~6 まで変わり、ウィンドウに表示されるスタイルも変化しているのが判るでしょう。気に入った表示が見つければそこで<Return>してください。すると返事を書こうとしているところに指定した引用記号を付けて元の記事の内容が現れるでしょう。但しそのままでは元の記事のヘッダ少しと本文など多くの部分が引用されますので、かなり不要な部分も含まれている可能性があることに注意してください。

あとはメールのときにそうしたように、引用のうちから不要な部分を削除し、残した必要な部分に対するコメントを書き足して行きます。

そうやって記事の本文が書き終わったら、C-c C-c で記事が発信されます。反対に記事を書いている途中で、その記事を投稿する気がなくなった場合は、C-c C-k とすれば、その記事を破棄することが出来ます。

5.5.9 記事のキャンセル

自分で投稿した記事については、その取り消しが可能です。この作業を記事のキャンセルと呼んでいます。例えば投稿した後で、書いた記事の内容に誤りを発見した場合や、投稿するグループを大きく外してしまった時などにキャンセルしたくなると思います。しかしキャンセルは非常手段で、多くの Internet 上の人々に迷惑をかけてしまいます。何よりもまず不注意な記事の投稿をしないように心がけることが大切だということを忘れないでおいて下さい。

記事をキャンセルするには、図 5.7 の状態のように、キャンセルしたい記事を読んでいる状態で C キーを押します。すると以下のようなメッセージをエコーラインに表示して本当にキャンセルして良いかどうか確認してきます。

```
Do you really want to cancel this article? (yes or no)
```

ここで yes<Return>と答えることによって記事がキャンセルされます。

5.5.10 メールで返事をする

ニュースの記事に対する返答というのは、同じく記事によって行なうフォローもあるが、時には記事の投稿者にメールによって返答したくなる場合もあるでしょう。

記事に対してメールで返事をするには、図 5.7 の状態のように、元の記事を読んでいる状態で **r** キーまたは **R** キーを押します。 **r** と **R** の違いは元記事の内容の引用にあります。 **R** は元記事の内容を引用してくれます。引用の際にはフォローの時と同じように引用記号を何にするか聞いてきます。いずれにしても以下のような状態になり、ここでメールを書くわけです。



図 5.9 メールで返事を書く

いつものように `--text follows this line--` の行から下にメッセージを書きます。この行を消してはいけません。内容の編集については普通の Mule のつもりで操作が可能です。メールの中身を書き終わったら `C-c C-c` でメールを送信します。反対にメールを書いている途中で、そのメールを送信する気がなくなった場合は、`C-x k` とします。するとエコーラインに以下のような確認の為の質問をしますので、`<Return>` とします。

```
Kill buffer: (default *mail*)
```

これで、そのメールを破棄することが出来ます。

5.5.11 古い記事を読み返す

GNUS は、未読記事が一つもないとそのニュースグループをグループモードでは表示しなくなり、記事モードでも普通では未読のものしか表示しません。一度読んでしまったニュースグループから、古い記事を再び読み返したい時は以下の手順を追うのがもっとも簡単です。

1. (もしあれば) そのニュースグループの記事を全部読んでしまう。未読の記事をゼロにする。
2. (もしグループリストに表示されていないが、ニュースグループの名前を覚えていれば) **j** キーでニュースグループ名を指定して表示させる。
3. (もしグループリストに表示されていなければ) **L** キーでそのニュースグループを表示させる。
4. グループモードで未読ゼロのニュースグループを選択し、読む。

要するに GNUS は未読記事ゼロのニュースグループを読むという指示をしてやると、以前の記事を読み返せるように働いてくれるのです。残っている記事が非常に多いと、その中から最近のものをいくつか選んで表示するかを以下のようなメッセージで問い合わせてきます。

```
How many articles from sandai.general (default 197):
```

例では sandai.general ニュースグループの記事が結構多いので、そのうちいくら取り出すかを聞いています。どうやら全部で 197 記事あるようで、黙って <Return>すると 197 取り出してくれます。つまり自分が読み返したいと思う記事の数をタイプしてください。すると指定した数だけの既読記事が取り出され、図 5.7 のように表示されるでしょう。あとは今まで通りの操作で同じように扱えます。

5.5.12 記事の保存

ほとんど全てのニュースの記事は時間が経てばシステムの中から自動的に削除されてしまいます。つまり残しておきたい記事に関しては自分の手元に保存しておく必要があるのです。

記事を保存するには図 5.7 の状態で記事を読んでいる時に、C-o キーを押します。するとエコーラインに以下のようなメッセージを表示して何と言う名前前で保存するかを聞いてきます。

```
Save article in Unix mail file: (default sandai.test) ~/News/
```

上の例ではたまたま sandai.test ニュースグループの記事を保存しようとしています。あなたのホームディレクトリの下に News というディレクトリの下に sandai.test という名前のファイルとして保存すると言っていますので、ここではそれに従うことにします。単に <Return>です。もしもファイルの名前を変更したければここでファイル名をタイプして<Return>です。ファイルが既存であればどんどん追加書きしていきます。

```
Loading rmailout...done
```

と言うメッセージが表示されて、記事が保存されます。

保存された記事をうまく読み返す方法もあるのですが、ここでは余り説明しません。保存したファイルは、特に指定をしなければあなたのホームディレクトリの下に News ディレクトリ以下にニュースグループごとの名前を付けて保存されているでしょうから、このファイルを直接見て下さい。

Mule の操作に慣れている人の為に

記事を保存する時に C-o ではなくて o キーで保存すると、記事はまるでメールのようにして保存されます。こうして保存された記事は MHE で読み返す事も出来ます。M-x mh-rmail で MHE を起動して、M-f で保存した記事のニュースグループをフォルダとして指定してやれば良いのです。MHE におけるフォルダの概念についてはここでは説明しません。

o で保存された記事は~/Mail/sandai.test/1 などという名前のファイルとして保存されています。

但しこうやって作成されたファイルは漢字コードが cc 環境標準の EUC ではなくて JIS になっている¹⁴ので、普通に cat コマンドや more コマンドでその中身（特に漢字の部分）を見る事が出来ません。中身を直接確認するには Mule で読み込んで見て下さい。メールが MHE によって保存されるときも、このようになっています。

¹⁴この表現が判らない人は無視して下さい。なにか普通ではないファイルになっているとだけ思ってください。

5.5.13 ちょっとしたテクニック

記事を選ぶ時に表題一覧を画面いっぱいに表示する

図 5.7 の様に、記事を表題を見ながら選んでいく時に表題一覧が数行分しか表示されなくて操作が面倒な時があります。そんな時は「= (等号、イコール)」キーが有効です。

図 5.6 の状態で、ニュースグループを選ぶ時に <Space> キーを押す代わりに = キーを押してやります。するとそのグループの表題一覧だけが画面いっぱいに表示されます。ここで記事を選んで <Space> キーを押せば普通に記事を読むことが出来ます。

図 5.7 の状態でも、= を押してやれば、画面いっぱいに表示されます。

あるグループの記事を全部読んだことにする

GNUS は記事を全て未読と既読に区別し、普通に記事を読もうとすると未読のものだけを表示します。そうすることによって GNUS を起動する度に、新たに届いた記事にだけ注目して読み続けることが出来るのです。

このような仕掛けは毎日ニュースを読んでいるような場合には有効ですが、しばらくニュースを読まなかったり新しいニュースグループを選んで読みはじめる時には困ることもあります。つまり既にそのニュースグループには記事が数百もたまっていて、とてもではないけれど読み切れないというような事が起こるのです。

そのために GNUS には「今ある記事を全部読んだことにする」という機能があります。グループモードでも記事モードでもいずれでも構いませんから c キーを押せば、今選んでいるグループの記事を全部まとめて読んだことにしてくれます。

c キーを押せばエコーラインに以下の様なメッセージを表示して確認してきます。

```
Delete all articles not marked as unread? (y or n)
```

まるで記事が全て削除されてしまうかのような表示ですが、単に「未読マークを消すよ」と言っているだけなので、安心して y キーを押してください。

5.6 記事を書くときの注意

ここでは操作方法ではなく、記事の中身の書き方やその周辺の事柄について説明します。

どこのニュースグループに投稿するか？

ニュースグループは大きく分けて、3 つあります。

sandai.*	京都産業大学内のみのニュースグループ 配布範囲 (以下 Distribution) は local を選択します。
fj.*	国内のニュースグループ Distribution は fj を選択します。
comp.*,rec.*,soc.*,sci.*など	海外のネットワーク (主にアメリカの USENET のニュースグループ) Distribution は world を選択します。

初めのうちは学内のニュースグループで練習をしてから学外のニュースグループを利用するようにしましょう。なお、上記の他にも海外のニュースグループがいくつかあります。

モデレータがいるニュースグループとないグループ

- モデレータ（そのニュースグループの管理人）がいないニュースグループは unmoderated と呼ばれています。すなわち、投稿すれば必ず全体に投稿されます。
- モデレータがいるニュースグループは moderated と呼ばれています。このニュースグループに投稿すると、記事はそのニュースグループを管理している人に自動的にメールとして送られ、その人が有益であると考えた記事のみが全体に投稿されます。どのニュースグループが moderated かどうかは、fj に関しては、ときどき fj.guide.general にポストされる fj のニュースグループに関する記事を参照して下さい。

sandai.test, fj.test の使いかた

sandai.test はポストやフォローのテストを行なうためのニュースグループです。fj.test もその為のものなのですが、fj.test を用いると、fj を購読している 1000 を越えるサイトに記事が流れることとなります。fj.test はシステム管理者が使うべきものですから、あなたは fj.test を利用してはいけません。

記事の内容について

- 特にこのように書けというものはありませんが、自分の書いた記事は他人が読むことを考えて、読む人の立場に立って書きましょう。ときどき fj.questions.*などで、質問の内容がつかめないものがあります。
- ニュースのような文字による通信では、普段相手と会って話す時や電話で話す時と異なり、微妙なニュアンスが伝わりません。時々これに注意しなかったばかりに Internet で喧嘩をしていることがあります。注意しましょう。
- こう言ったことにならない為にも、しばらく人の投稿を見てネットワークニュース上でのマナーを学ぶのがいいと思います。喧嘩の元になるようなニュース上での「ひどい」記事は fj.misc というニュースグループなどでよく取り上げられますので参考にしてください。
- Subject の内容は、その記事の内容を反映しているものにすべきです。また、Subject には、漢字を使ってはいけません。
- 投稿の内容に関する責任は、書いた人の属する組織にあるのではなく、書いた人自身にあります。また、著作権はその記事を書いた人にあります。
- ニュースにおける誰かの発言は、発言者個人のものであって決して組織の代表としての発言ではないことに常に注意しておいてください。例えば某メーカーの社員が自社の製品について批判的なことを書いても、それは個人の感想に過ぎず、それ以上の意味はありません。その発言を取り上げてその会社の他の人に文句を言ったりするのは「反則」です。自分が買った製品が気に入らなくて「〇×社の人説明して下さい」などと発言するのもナンセンスです。むしろこのような場合は「〇×社の製品を買った人、良い解決法を考えましょう」とやる方が余程建設的です。
- 違法な内容の記事（歌詞のポストなどによる著作権の侵害、公道を時速〇〇〇で走ったなど）をポストしてはいけません。当たり前の話ですが、反社会的な記事を投稿するべきではありません。それが問題として取り上げられ、Internet が社会から迫害されてしまう可能性だってあるのです。（現在の社会を構成する大部分の人にとって Internet なんてあっても無くてもどうでも良いものの一つなのです。）

フォローの方法

ニュースを読んでいるとわかりますが、相手の引用文の始めには「>」等をつけます。また、引用の前には、誰に対する引用なのかをはっきりさせます。引用の量はなるべく少なくなるようにして下さい。なお「>」で始まる行の行数は、自分がポストする記事全体の半分以下でなければなりません（ニュースリーダーのデフォルトではこうなっています）。つまり引用大量、意見少量という事はするべきでない、という訳です。

signature のつけかた

signature はあなたの顔みたいなものです。個性的で印象に残るものにしましょう。行数があまり多くなると記事自体が読みにくくなってしまいます、少ない行数に情報をきれいにまとめるところが腕の見せ所です。目安としては4行以下がいいでしょう。

キャンセルについて

自分のポストした記事のみキャンセルを行なうことができます。ただし、キャンセルを行なうと、キャンセルを行なうためのコントロールメッセージが fj.* のニュースグループならば、fj のすべてのサイトに送られます。キャンセルを行なう時は十分慎重に行なって下さい。また、最初からキャンセルを行なわなくも済むようなポストをすべきです。

引用のマナー

自分のところに送られてきたメールをニュースに引用する場合には、メールの送り手の許可を得てからにしましょう。また、ニュースを引用する場合には、その引用した記事を特定できるような情報を付けて下さい。

第 6 章

UNIX もっともっと

ここでは計算機センターが管理している UNIX 環境を例に取りながら、より進んだ UNIX 環境の利用方法の説明と、さまざまな機能の紹介をします。ここでの説明は網羅的なものではなく、部分を取り上げて曖昧に説明しています。これは本文の読者のコンピュータそのものについての知識のハードルを高くしすぎないためで済ませるためです。読者が UNIX、つまりコンピュータの利用に慣れて行くにしたがって自分でマニュアル、書籍、ネットワークなどから情報を常に取り込んで理解することが大切です。

6.1 ファイルの整理

cc 環境ではファイルは UNIX コンピュータのディスクと呼ばれる部分に保存されています。このディスクと呼ばれる装置に格納できるファイルの量には限界が有り、cc 環境で利用できるディスク装置の数も限られています。この限られた資源を千人を超える利用者で共同利用しているのですから、おのずと一人一人が利用できるディスクの量も限られてきます。これはすなわち利用者を作ることの出来るファイルの数やその量の合計には限りがあるということです。その限界を超えてファイルを作ろうとしてもエラーメッセージが表示され、ファイルを作成することが出来ません。

cc 環境では利用者それぞれによるファイルの整理が必要なのです。

ディスクを大量に使っている人は login したときに「ディスクを既定値を超えて利用しているので不要なファイルを消去せよ」という意味のメッセージを見ることがあるかも知れません。これはその人が、一人の利用者が使用する事の出来るディスクの上限値を超えている事を意味しています。

このメッセージが表示された場合は、とにもかくにも不要なファイルを消去しないとイケません。

6.1.1 ファイルの大きさを調べる

ファイルの整理をするには、どれでも不要なファイルを削除すれば良いのですが、とにかく闇雲に削除するのではなく、サイズの大きなファイルから削除する方が良いでしょう。ファイルのサイズを見る方法は幾つか有りますが、ここでは二つ紹介します。

ls -l: 個々のファイルの大きさを調べる

ls コマンドに引数として -l オプションを加えれば、ファイルの大きさが判ります。ファイルの大きさは Byte、バイトと呼ばれる単位で表現され、ls -l で表示される情報のうち、左から 4 番目にあるものです。以下に例を挙げます。

```
csosf41(93)% ls -l core
-rw-r--r--  1 root    daemon   153248 02月22日 22:32 core
csosf41(94)%
```

上記の例では core というファイルの大きさは 153248 バイトであることが判りました。ファイルの種類によって一概には言えないのですが、単純には 1 文字は 1 バイトで表現できますので、153248 文字くらいの内容がこのファイルには含まれていると考えられます¹。

ls コマンドはファイル名の指定をせずに、単に `ls -l<Return>` として実行した場合はカレントディレクトリのファイル全ての情報を表示します。

しかしカレントディレクトリに存在するディレクトリファイルについては、そのディレクトリの下にあるファイルの量をまとめて表示してくれるわけではありません²。ディレクトリファイル以下にあるファイルの大きさについては、そのディレクトリ以下に `cd` コマンドで移動して、もう一度 `ls -l<Return>` とするしかありません。それが面倒ならば `ls -lR<Return>` で階層的にファイルの情報を表示させることも出来ますが、これはかなり見にくいものです。

du -s : ディレクトリ以下に含まれるファイル大きさの合計を調べる

ls -l コマンドではディレクトリ以下に含まれるファイルの容量の合計を知ることは簡単には出来ません。そのような場合は `du` コマンドに `-s` オプションを付けて実行します。以下に Mail ディレクトリ以下のファイルの量の合計を表示する例を示します。

```
cc2000(82)% du -sk Mail
39      Mail
cc2000(83)%
```

勿論ファイル名に Mail ではなく * を指定すればカレントディレクトリ以下の全てのファイル³の量を計算した結果が表示されます。

注意が必要なのは `du` コマンドは、その結果をバイト単位ではなくブロック単位、もしくはキロバイト単位で表示することです。1 Block = 512 Bytes , 1 KiloBytes = 1000 Bytes (場合によっては 1 KiloBytes = 1024 Bytes) となります。一般的にはキロバイト単位の方が扱い易いと思われるので、`du` コマンドを利用するときには `-sk` と言うように表示をキロバイト単位で行うようなオプションを付けておくといいでしょう。

上記の例では Mail というディレクトリ以下のファイルの量の合計は 39 キロバイトという事を意味しています。

但し `ccns001~ccns015` では `-k` オプションは存在せず、エラーとなります。その代わりに `ccns001~ccns015` では `du -s` だけで表示はキロバイト単位となります。

6.1.2 身に覚えのないファイル

UNIX を利用していくと、様々なファイルを作って作業することになるでしょう。それらのうち、幾つかはあなたが作った覚えのないファイルもあるでしょう。

以下にそれらのうち代表的なものを示します。

¹ この解釈は余りにも単純すぎて、現実には妥当ではありませんね。計算機の内部構造に興味のある人はもう少し自分で調べて納得してください。ここでは説明しません。

² このディレクトリファイルの「大きさ」については無視してください。これまた計算機の内部構造に興味のある人は調べてみるといいでしょう。

³ 厳密には名前が . で始まるファイルについては表示されません。

Mail 電子メールを読むためにあらかじめ作られているディレクトリです。削除しない方がいいでしょう。

Apps , Library ccsosf001~015 などの NeXTStation を利用するためにあらかじめ作られているディレクトリです。削除しない方がいいでしょう。

jsykojin.dic ccsosf01~40 でのかな漢字変換のためにあらかじめ作られているファイルです。DEC-3300 のキーボードに付いている「変換」キーを使うかな漢字変換をしない人は削除してもいいでしょう。

core プログラムやコマンドが実行している途中で何等かの原因によって実行が中断されたときに強制的に作成されます。このファイルの意味、使い道を知らない人は無条件に削除して構わないでしょう。むしろ大きさが結構大きいものが多いので積極的に削除すべきです。

dead.letter メールを書いている最中に何等かの原因によって作業が中断したときに強制的に作成されます。中身はきっと最後に書き掛けていたメールの中途半端な中身だと思います。内容を **more dead.letter<Return>** などして確認した後、保存の必要がなければ削除すれば良いでしょう。

mbox メールを扱うには MHE だけでなく、mail というコマンドも利用できます。このファイルは mail コマンドが作成するものです。cc 環境では mail コマンドによる日本語メールの読み書きは、その動作を保証していません。MHE 環境への移行を薦めます。移行が済めば mbox ファイルは内容を **more mbox<Return>** などして確認した後、保存の必要がなければ削除すれば良いでしょう。

printscreen.ps ccsosf01~40 の「画面印刷」機能を利用した時に出来るファイルです。印刷が済めば削除すれば良いでしょう。印刷する以外で内容を確認するには cc2000 で **ghostview printscreen.ps<Return>** とすれば良いでしょう。この機能を利用した画面印刷は何かとトラブルを起こすので薦められません。現在は容易に使えないように隠されています。画面印刷をするのならば、xv による画面取込み機能を利用するのが最も安定しています。

名前は固定ではありませんが、ファイルの名前に特徴のあるものが勝手にできている場合が有ります。以下に代表的なものを示します。

#abc# など、# で囲まれたファイル

Emacs がファイルを修正したときに一時的に作成するファイルです。Emacs の作業が何等かの理由で中断した為に残ったものと思われまます。内容を確認した後、保存の必要がなければ削除すれば良いでしょう。

abc~ など、~ が末尾に付くファイル

Emacs がファイルを修正したときに、修正前のファイルの内容をそのまま「~」を付けた名前のファイルに残します。例えば abc というファイルを修正し、保存すれば、修正した後の内容は abc ファイルに残り、修正する前の内容は abc~ ファイルに残ると言うことです。内容を確認した後、保存の必要がなければ削除すれば良いでしょう。

.cshrc , .login など、. が先頭に付くファイル

これらのファイルは大抵の場合何等かの環境設定ファイルです。内容が判らない場合は不用意に変更したり削除したりしない方がいいでしょう。なお、これらの先頭が「.」（ピリオド）で始まる名前のファイルは普通に ls コマンドなどを実行してもその一覧には表示されません。ある意味では「隠された」ファイルとなっているのです。これらのファイルの一覧を表示させるには **ls -a<Return>** とするのが良いでしょう。

これら以外にも様々なファイルが知らない間に出来ているかも知れません。小さなファイルはまだ良いのですが、大きなものは注意して、不必要と思ったら削除しておく方がいいでしょう。どうしても何故出来たのか判らないようなファイルが出来るとなれば計算機センターの相談窓口か、sandai.question ニュースグループに相談してください。

6.1.3 ファイルを削除する

ファイルの削除には `rm` コマンドを利用します。`rm` ファイル名<Return>です。ディレクトリを削除するには `rmdir` コマンドを利用します。`rmdir` ディレクトリ名<Return>です。ディレクトリ以下にファイルが残っていて、それらのファイルごと削除する場合は `rm -r` ディレクトリ名<Return>とします。

6.1.4 ディスクの利用量をチェックする

ファイルの量が限界に到達してから仕方なくファイルを削除するよりは普段からディスクの使用量をチェックしておく方がいいでしょう。あなたが現在使っているファイルが占めているディスクの中での量を表示するには `quota -v` コマンドを利用します。以下に例を示します。

```
cc2000(88)% quota -v
Disk quotas for yasuda (uid 2126):
Filesystem      usage  quota  limit   timeleft  files  quota  limit   timeleft
/NF/home         1433  30000  30000           141  10000  10000
cc2000(89)%
```

ここで表示される内容の意味を左から順に説明します。

Filesystem	余りにしないで下さい。(^^;)
usage	今使っているファイルの量の合計を表します。例では 1433KiloBytes 使っていることとなります。
quota	利用できるファイルの量の上限を表します。例では 30,000KiloBytes、つまり 30MegaBytes を意味しています。
limit	直前の quota と同じです。違いは気にしないで下さい。(^^;)
timeleft	余りにしないで下さい。(^^;)
files	今使っているファイルの数の合計を表します。例では 141 ファイルあることとなります。
quota	利用できるファイルの数の上限を表します。例では 10000 ファイルを意味しています。
limit	直前の quota と同じです。違いは気にしないで下さい。(^^;)
timeleft	余りにしないで下さい。(^^;)

すなわちファイルについてはその量と個数のそれぞれの合計に独立に制限があるのです。しかし現実には個数が限界を超えることは殆ど無いと思われまますので、要は左から 2 番目の `usage` が 3 番目の `quota` を超えないように調整していれば良いわけです。

6.2 情報検索

6.2.1 学内の利用者情報の検索

電子メールやニュースなどを利用していると、あの人のユーザ名は何だったかな？と思うときがあるでしょう。UNIX ガイドの「UNIX それから」では `finger` コマンドによる利用者情報の取得方法を紹介しましたが、ここではさらに詳しく利用者の情報を得ることが出来る `whois` コマンドを紹介します。

但し注意が必要なのはここで紹介している `whois` サービスは主として `cc` 環境の為に提供されているもので、全ての学内の UNIX コンピュータで利用でき、全ての学内の UNIX コンピュータ利用者の情報が登録されているわけではないということです。`cc` マシン群ではこのサービスが利用できますが、他の UNIX コンピュータからどの程度利用できるかは保証できません⁴。

`whois` コマンドを利用すると利用者のユーザ名などが検索できます。`whois keyword<Return>` です。以下にキーワードとして `yasuda yutaka` を与えた場合を示します。キーワードには空白が含まれて居ますので、念のため引用符でかこんで与えています。

```
cc2000(81)% whois 'yasuda yutaka'
[ Whois service for Kyoto Sangyo University.      1994.7.7 serviced from ccnic.]
[ Please contact yasuda@ksuics.kyoto-su.ac.jp if you find some trouble.      ]
```

```
ユーザ名,uid : yasuda , 2126
氏名          : Yasuda Yutaka
              : 安田 豊
所属          : 計算機センター事務室 (事務職員)
アカウント    : yasuda@cc yasuda@ksuics yasuda@cs
cc2000(82)%
```

教員、学生など、その人の種別によって表示される項目は様々です。ユーザ名の横に表示されている `uid` は個人を識別するためにユーザ名と一対一に割り当てられている番号です。`whois` コマンドのキーワードとしても使えます。例えば上記の例では `whois 2126<Return>` としても同じ結果が得られるでしょう。最後にアカウントという情報が表示される場合があります。これはその人がそこに表示されているシステムにユーザ登録されている事を示しています。しかし登録されているからと言って、そのアドレスにメールを送れば必ず読んでくれるとは限りません。あくまでメールを送るときを目安程度に考えてください。

キーワードにはユーザ名、名前、`uid`、所属などが使えます。また、これらの文字列の一部分でも与えることが出来ます。先ほどの例ではキーワードにローマ字のフルネームを用いましたが、この一部分として `whois yasuda<Return>` などのようにする事も出来ます。以下に例を示します。但し `yasuda` というキーワードにマッチする利用者はちょっと多かったようです。このような場合はマッチした人の一覧を挙げるようになっていますので、ここから `uid` などを元にもう一度 `whois` を行い、目標の人の詳細情報を得ることが出来ます。

```
cc2000(82)% whois yasuda
[ Whois service for Kyoto Sangyo University.      1994.7.7 serviced from ccnic.]
[ Please contact yasuda@ksuics.kyoto-su.ac.jp if you find some trouble.      ]
```

⁴因みに `cc` 環境 `whois` サービスは `ccnic` マシンが提供しています。`cc` 環境マシン群以外から利用する場合は `whois -h ccnic keyword` などとしてください。

uid	ユーザ名	種別	名前
2126	yasuda	S	Yasuda Yutaka 安田 豊
10233	abcd	G	Yasuda Hisahiro 安田 久洋
10335	kazutaka	G	Yasuda Kazutaka 安田 一尊
13206	hayato	G	Yasuda Hayato 安田 勇人

種別 = G:学生 I:院生 K:教員 S:職員 M:その他

Many macthing. See the detail, use uid number for key.
cc2000(83)%

ただ、余りにも漠然としたキーワードを与えた場合はマッチする利用者が多すぎ、以下のようなメッセージを表示して情報を表示しない場合が有ります。

```
cc2000(81)% whois hide
[ Whois service for Kyoto Sangyo University.      1994.7.7 serviced from ccnic.]
[ Please contact yasuda@ksuics.kyoto-su.ac.jp if you find some trouble.      ]
```

Too Many macthing. Try again with more detail information.
cc2000(82)%

このような場合はキーワードを **hide** だけではなく、**hideyuki** などにするなどしてもう少し条件を絞り込んで **whois** を再実行してください。

6.2.2 学外の組織などの情報を調べる

6.2.1で紹介した **whois** コマンドは cc 環境で独自に提供されているものです。学内の他の UNIX コンピュータで同じようにこのコマンドが通用するとは限りません。

産業大学以外の組織でも **whois** サービスが行なわれている場合があります。ここではそれらのうち、Internet を通じて公開されている **whois** サービスの使い方を紹介します。以下に紹介する公開 **whois** サービスは主にネットワーク関係の情報を提供しています。サービスは具体的には Internet 上のあるマシンで行なわれており、そのマシンに目掛けて **whois** コマンドでキーワードの問い合わせを行なうのです。

whois.nic.ad.jp 国内のネットワーク関係の情報を提供しています。これは JNIC（日本ネットワークインフォメーションセンター）という組織が行なっているサービスです。

rs.internic.net Internet 関係の情報を提供しています。これは Network Solutions Inc. というアメリカにある組織が行なっているサービスです。

nic.ddn.mil MILNET 関係の情報を提供しています。MILNET はアメリカの軍事関係の組織がつながれたネットワークです。ですからこのサービスは国防省が行なっているはずですが。（ちょっと自信がありません。）

組織の情報を得る

例えば京都産業大学のドメイン名 (Internet 上の組織もしくは組織のネットワークの名前) を知りたい場合は、国内の情報ですから whois.nic.ad.jp に問い合わせます。このホスト名は whois コマンドに -h に続いてオプションとして与えます。whois -h whois.nic.ad.jp. 'kyoto sangyo'<Return>と入力して下さい。キーワードに空白が含まれるので念の為にキーワードを引用符で囲んでいることに注意して下さい。以下に例を示します。

```
cc2000(86)% whois -h whois.nic.ad.jp. 'kyoto sangyo'
[ JPNIC database provides information on network administration. Its use is ]
[ restricted to network administration purposes. For further information, use ]
[ 'whois -h whois.nic.ad.jp help'. To suppress Japanese output, add '/e' at ]
[ the end of command, e.g. 'whois -h whois.nic.ad.jp xxx/e'. ]
```

```
京都産業大学 (Kyoto Sangyo University)          KYOTO-SU.AC.JP
京都産業大学 (Kyoto Sangyo University)          133.101.0.0
```

To single out one record, look it up with '!xxx',
where xxx is a key shown on the right hand side.

```
cc2000(87)%
```

京都産業大学のドメイン名 kyoto-su.ac.jp とネットワークアドレス 133.101.0.0 が表示されました。詳細な情報を得る為に今度は キーワードに kyoto-su.ac.jp を与えて検索してみましょう。例は挙げません。試してみるといいでしょう。逆にドメイン名からその組織についての情報を得ることも出来ます。

勿論海外の組織やドメイン名などについての情報を得る為には whois -h に続けるホスト名を rs.internic.net. などに置き換えれば大丈夫です。

人の情報を得る

今度は例えば Internet 上の yasuda さんを調べてみましょう。

whois -h rs.internic.net. yasuda<Return>と入力して下さい。以下に例を挙げます。

```
cc2000(93)% whois -h rc.internic.net. yasuda
whois: rc.internic.net.: cc2000(94)% whois -h rs.internic.net. yasuda
YASUDA SEISAKUSHO CO.,LTD. (NET-YASUDANET) YASUDANET          202.23.121.0
Yasuda Computer Service Co.,Ltd. (NETBLK-YCS-NET) YCS-NET
                                                                202.19.214.0 - 202.19.215.0
Yasuda Women's College (NET-YUNET) YUNET                    160.192.0.0
Yasuda, Keiichi (KY14)          yas@FURUKAWA.CO.JP          +81 463 23 1045
Yasuda, Noriyuki (NY12)                          +81-52-604-1111 ext. 2486
Yasuda, Yoshinori (YY2)                          81-92-412-5996 ext. 402
Yutaka, Yasuda (YY13)          yasuda@KYOTO-SU.AC.JP          +81-75-705-1483
```

The InterNIC Registration Services Host ONLY contains Internet Information
(Networks, ASN's, Domains, and POC's).

Please use the whois server at nic.ddn.mil for MILNET Information.

```
cc2000(95)%
```


組織や人やらの情報がごちゃまぜに表示されています。目当ての yasuda yutaka さんの詳細な情報を得るには、ローマ字の名前の後ろについている YY13 というキーワードを使ってもう一度 whois で問い合わせます。この YY13 などのキーワードは Internic で人の情報を整理する為に使っているキーらしく、このキーワードを与えることによって必ずその人の詳細な情報が表示されるようになっています。

ところで大抵の人は Intenret に yasuda さんがたったこれだけの筈はないと言う事に気がつくでしょう。Internic などに登録されている人というのは一般的にはネットワーク管理者である事が多いようです。それ以外の人あまり登録されていない感じです。しかし逆にネットワーク上での有名人というのは割合にネットワーク管理者だったりしますから、結構役に立つのです。

人の情報と違ってドメインの情報についてはほとんど完全に登録されているはずですが。

6.3 ファイルのアクセス権

誰か友達がファイルを作っていて、そのファイルをあなたが自分のホームディレクトリ以下にコピーしたいと思ったとします⁵。ファイルの階層構造を理解したあなたは早速以下のようなコマンドでコピーをしようとするでしょう。

```
csosf41(85)% cp ~/tanaka/sample.tex ~
cp: /NF/home/g840/tanaka/sample.tex: Permission denied
csosf41(86)%
```

多くの場合は上の例のように失敗してしまいます。このエラーメッセージ「Permission denied」は UNIX を使っていると、時々みかけるエラーですね。これは「あなたにはアクセス権が無いよ」ということを意味しています。

6.3.1 アクセス権

UNIX コンピュータはみんなで使うコンピュータであることは初めに説明しました。そのため、あなたのホームディレクトリのとおりには他人のホームディレクトリがあったりします。つまり誰でも他人のファイルの置き場所が大体わかっているわけです。これでは誰でも他人のファイルを覗いたり、書き込んだり、消去したりできてしまいます。

これは安全上（セキュリティ）の問題です。たった一人の人が専有して一台のコンピュータを使用し続けるのなら、そのコンピュータを自分の部屋に置いて、その部屋の入口に鍵を掛けておけば良いのです。でも cc 環境の UNIX のようにみんなで使うコンピュータは閉じ込めるわけにはいきませんからセキュリティを確保するためには何か別の対策が必要です。

そこでみんなで使うコンピュータの多くではアクセス権という考え方を採り入れています。つまりコンピュータ上の資源にはそれぞれ誰が使えるかという情報が書いてあるのです。逆にその情報から洩れた人は使えないということです。コンピュータ関係の世界では、この「(資源を) 使う」という事を「アクセスする」と表現することがあります。「アクセス権」とはつまり (資源に対する) 「使用権限」と言う意味です。

そして、cc 環境では安全のため（もしくはプライバシーのため）に「利用者のホームディレクトリ以下のファイルは、利用者自身しかアクセスできない」という設定になっています。先の「Permission denied」はこの制限に引っかかったためのエラーメッセージだったのです。

6.3.2 UNIX におけるアクセス権

UNIX では全てのファイルに常にアクセス権が設定されています。ファイルが利用者によって作成されると、利用者がまずファイルの所有者となり、自動的にアクセス権が設定されます。アクセス権の設定は後で所有者によっていくらかでも変更することが出来ます。

アクセス権は具体的には以下の 3 種類の項目に対してそれぞれ許可を与えることによって設定します。

⁵ もちろんその友達の了解を得て、ですよ！勝手に人のファイルをコピーしてはいけません。

種類 (略号)	ファイルの種類	設定することによって許可されるアクセスの内容
読み出し (r)	ファイル	そのファイルの中身を取り出す
	ディレクトリ	そのディレクトリ以下のファイルの一覧を表示する
書き込み (w)	ファイル	そのファイルの中身を書き変える、もしくは消去する
	ディレクトリ	そのディレクトリ以下にファイルまたはディレクトリを新たに作る、もしくはそのディレクトリを消去する
実行 (x)	ファイル	そのファイルを実行する
	ディレクトリ	そのディレクトリ以下に <code>cd</code> コマンドで移動する

上記の3種類のアクセス権の設定項目は「所有者 (user)」と「グループ (group)」と「その他の人 (other)」のそれぞれの対象に対して別個に割り当てられます。つまりあるファイルに対して「所有者は読み書きできて、グループは読むだけで、その他の人は読むことすら出来ない」という設定が出来るのです。

所有者とはそのファイルを作った人です。グループについてはここでは説明しません。自分がどのグループに属しているかは `id` コマンドで確認することが出来ますので、参考にして下さい。学生は全員 `student` グループの一員です。教員は全員 `teach` グループの一員です。その他の人と言うのは所有者でもグループのメンバーでも無い利用者のことです。

6.3.3 アクセス権限を調べる

`ls -lg` コマンドで、ファイルのアクセス権限を含めた詳細な情報を得ることが出来ます。書式は以下の通りです。

```
ls -lg [ファイル名...]
```

以下に実行例を示します。

```
csosf41(82)% ls -lg
total 20
drwxr-xr-x  2 yasuda  student    512 08月 29日 1992  Apps
drwxr-xr-x 11 yasuda  student    512 12月 17日 15:08 Library
drwxr-xr-x  5 yasuda  student    512 11月 02日 17:25 Mail
drwxr-xr-x  2 yasuda  student    512 03月 14日 18:26 Unix
-rw-r--r--  1 yasuda  system   4096 03月 08日 23:22 jsykojin.dic
-rw-r--r--  1 yasuda  student    523 03月 08日 23:24 log
drwxr-xr-x  2 yasuda  student    512 03月 14日 18:26 memo
csosf41(83)%
```

これ以降に以下の一行を取りだして、詳細に説明します。

```
-rw-r--r-- 1 yasuda student 523 03月08日 23:24 log
```

-rw-r--r--	ファイルの種類とアクセス権限についての情報。(さらに後述)
1	リンク数。ここでは説明しない。
yasuda	所有者
student	グループ
523	ファイルの大きさ。単位は Byte(バイト)。ここでは説明しない。
03月08日 23:24	作成年月日
log	ファイル名

この、ロングフォーマットで得られる出力の最初の部分 (-rw-r--r--) に注目して下さい。以下にそれぞれの桁について説明します。

桁位置 (例での値)	意味
1(-)	ファイルの種類を表す。d ならディレクトリ、- ならファイル。
2(r)	所有者に対する読みだし許可を表す。r なら許可、- なら禁止。
3(w)	所有者に対する書き込み許可を表す。w なら許可、- なら禁止。
4(-)	所有者に対する実行許可を表す。x なら許可、- なら禁止。
5(r)	グループに対する読みだし許可を表す。r なら許可、- なら禁止。
6(-)	グループに対する書き込み許可を表す。w なら許可、- なら禁止。
7(-)	グループに対する実行許可を表す。x なら許可、- なら禁止。
8(r)	その他に対する読みだし許可を表す。r なら許可、- なら禁止。
9(-)	その他に対する書き込み許可を表す。w なら許可、- なら禁止。
10(-)	その他に対する実行許可を表す。x なら許可、- なら禁止。

つまり以下のような表組を一列に引き延ばして書いたようなものです。

対象	読みだし	書き込み	実行
所有者	r	w	-
グループ	r	-	-
その他	r	-	-

6.3.4 アクセス権限を設定する

アクセス権限は所有者によって設定を変更することが出来ます。chmod⁶ コマンドを利用します。書式は以下の通りです。

```
chmod mode file...
```

chmod コマンドは指定の file のアクセス権を mode の指定に従って変更します。mode は3つの部分からなる文字列で、対象... オペレータ 内容 となっています。以下にそれぞれの部分に与え得る記号とその

⁶change mode の略のつもりなのです

意味を説明します。

対象	u	所有者
	g	グループ
	o	その他の人
	a	全ての人
オペレータ	+	追加
	-	取消
設定内容	r	読みだし
	w	書き込み
	x	実行

以下に `chmod` の具体例を挙げます。

1. `chmod a+r log` 全ての利用者に読みだし権限を与える
2. `chmod ug+rw log` 所有者とグループメンバーに読み出しと書き込み権限を与える
3. `chmod go-rw log` 自分以外の利用者から読み書き出来ないようにする

+ と - オペレータは元のアクセス権に新たな設定を「付加する」ように働きます。つまり元の設定で、影響を受けない部分はそのまま残ります。例えば 1. の例では書き込み権限や実行権限などには影響を与えず、もとの `log` ファイルが持っていたはずの書き込みに関するアクセス権限の設定はそのまま残ります。

いろいろ試して、`chmod` の振舞いを理解するのがいいでしょう。

ちょっとマニアックな `chmod` の使い方

もっと直接的にアクセス権の設定をするために、`chmod` コマンドにはもう一つの `mode` 文字列の与え方があります。アクセス権を表す `rw-r--r--` などの文字列を以下のようにして二進数に見立てて計算するのです。

```
rw-r--r-- → rw-,r--,r--  
110100100 → 110,100,100 → 6,4,4 → 644
```

つまり `rw` の部分で3つに分け、それぞれで二進数のつもりで計算するのです。`r` が 2^2 の桁、`w` が 2^1 の桁、`x` が 2^0 の桁、と言う訳です。計算すると、`rw-r--r--` は `rw-`、`r--`、`r--` と分割され、 $2^2 * 1 + 2^1 * 1 + 2^0 * 0$ 、 $2^2 * 1 + 2^1 * 0 + 2^0 * 0$ 、 $2^2 * 1 + 2^1 * 0 + 2^0 * 0$ となり、最終的に 6,4,4 となります。コマンドとしては `chmod 644 log` で完全に `rw-r--r--` を意味するアクセス権の設定が出来ます。

6.4 シェルよもう一度

6.4.1 ファイル名の補完

今までファイルを指定するときにはファイル名を全てタイピングしていたと思います。でも、ファイル名を全部タイプしなくてももっと簡単にファイルを指定することが出来る場合があります。その為にファイル名を記述する部分に用いる文字として* と ? があります。これらの文字を UNIX ではメタキャラクタと呼んでいます。以下にそれぞれ詳しく説明します。

説明はカレントディレクトリ以下のファイルの状況が以下のようにだと仮定して行います。

```
csosf01(88)% ls
bb      cc      log      log2     sample
csosf01(89)%
```

任意の文字列に適合するメタキャラクタ「*」

ここで「*」文字を紹介しましょう。例えば上記の例で `log` と `log2` の両方のファイルの中身を見たい場合は、以下のようにすれば良いのです。

```
csosf01(92)% more l*
```

つまり `l*` とは「`l` で始まり、それ以降はどんな文字列でも構わないファイル」を列挙することを意味しています。その結果 `more l*` は `more log log2` と解釈されて実行されます。

* が適用可能なのは何もファイル名の最後に限りません。

例えば上記の例で `more *2` とすればそれは「どんな文字列で始まっても構わないから、最後が `2` で終わるファイル」と解釈され、`log2` ファイルを差します。

`*o*` とすれば、それは「どんな文字列で始まっても構わないから、途中に `o` を含み、またどんな文字列で終わっても構わないファイル」と解釈され、`log log2` の二つのファイルを差します。

`*` とすれば、それは「どんな文字列でも構わないファイル」と解釈され、`bb cc log log2 sample` の五つのファイルを差します。

* 文字はファイル名の記述のどこに幾つ現れても構いません。

また、適合する文字が無くても適合したとみなします。例えば `log*` とすれば「`log` で始まり、それ以降はどんな文字列でも構わないファイル」と解釈され、`log log2` の二つのファイルを差します。

任意の一文字に適合するメタキャラクタ「?」

今度は「?」文字を紹介しましょう。例えば上記の例で `bb` と `cc` の両方のファイルの中身を見たい場合は、以下のようにすれば良いのです。

```
csosf01(92)% more ??
```

つまり `??` とは「どんな文字でも構わないから、二つからなるファイル」を列挙することを意味しています。その結果 `more ??` は `more bb cc` と解釈されて実行されます。

? が適用可能なのは何もファイル名の最後に限りません。例えば上記の例で `more ?b` とすればそれは「どんな文字でも構わないから一文字あって、次が `b` で終わるファイル」と解釈され、`bb` ファイルを差します。

? 文字はファイル名の記述のどこに幾つ現れても構いません。

ディレクトリに対してのメタキャラクタ

* と ? 文字はディレクトリに対しても適用可能です。例えば、`ls */*2` などとすれば「カレントディレクトリ以下の全てのディレクトリの下にある、どのような文字列で始まって構わないから最後は 2 で終わる名前を持つファイル」を見つけてその一覧を表示します。

6.4.2 対話的なファイル名の補完

シェルからファイル名を記述しているとき、シェルが提供している対話的なファイル名の補完機能が幾つか利用できます。

再び説明はカレントディレクトリ以下のファイルの状況が以下のようにだと仮定して行います。

```
csosf01(88)% ls
bb      cc      log      log2     sample
csosf01(89)%
```

ファイルの名前を途中までタイプしてくれる「<Tab>」

例えば `more sample` とタイプしたくて、`more s` までタイプしたとします。この状況で <Tab> キーを一度押せば `more sample` と、シェルの方で勝手にタイプしてくれて、`sample` から一つ離れた位置にカーソルが移動します。そこで <Return> を押せば良いと言うわけです。

つまりシェルは `more s` とタイプされた段階で、`s` に続くファイル名としてはこの状況では `sample` しか無いよ、と気を利かせてくれているのです。

今度は `more log2` とタイプしたくて、`more l` までタイプしたとします。この状況で <Tab> キーを一度押せば `more log` と、シェルの方で勝手にタイプしてくれて、`log` に続く位置にカーソルが移動します。ベルが一度鳴るかも知れません。

今度はシェルは `more l` とタイプされた段階で、`l` に続くファイル名としてはこの状況では `log` と `log2` しか無い事が判ります。だからとにかく一致している途中までタイプしてくれているのです。そこで残り不足している `2` をタイプして、<Return> を押せば良いと言うわけです。

ファイル名の一覧を表示する「C-d」

<Tab>による補完を行って、カーソルがファイル名の記述のすぐとなりに来て（ベルが鳴る）時は、まだ残りがあるかも知れないよ、という意味でした。この段階で、では一体どのような名前のファイルが可能性として残っているのかを確認するには C-d キーを押します。そうすれば以下のように、まだ可能性のあるファイル名の一覧を挙げてくれます。

```
csosf01(86)% more log ... ここで空白などあけずに<Control>-d
log  log2
csosf01(86)% more log
```

こうしてファイル名が長ったらしい場合でも、<Tab> キーと C-d を組み合わせて行けば、少ないタイピングで間違いなくそのファイルを指定することが出来ると言うわけです。

6.4.3 シェル変数と環境変数

シェルには⁷シェル変数と環境変数と呼ばれる、コマンドなどの動作に影響を与えるものがあります。これを設定し直すことによって、もう少しあなたにとって都合の良い UNIX 環境が得られるかも知れません。ここではシェル変数や環境変数の本質については触れずに、その設定例やヒントを紹介します。本質的なことについては附録の参考文献などからシェルに関する記述を参照してください。

シェル変数の表示、設定

現在設定されているシェル変数に何が有るかを調べるには `set` コマンドを利用します。また、シェル変数の内容を変更するにも `set` コマンドを利用します。`set` コマンドには以下の2通りの書き方があります。

1. `set`
2. `set var = string`

1. の書き方の場合、現在設定されている全てのシェル変数を表示します。

2. の書き方の場合、シェル変数 `var` の内容を `string` に設定します。

また、`echo` コマンドを使って特定のシェル変数の内容を表示することも出来ます。`echo $var <Return>` とすると、`var` という名前のシェル変数の内容を表示します。シェル変数は慣例として英小文字と数字で構成されています。

例えば `tcsh` ではしばらく使わないでほうっておいた場合、自動的にそのシェルを終了する機能があります。これを制御しているのはシェル変数 `autologout` です。`echo $autologout<Return>` とする事によって、現在設定されている放置時間が分単位で確認できると思います。これは不用意に切り忘れてしまったような不要なシェルを放置しないための処置なのですが、この時間が短すぎて困る環境の人もいるでしょう。そのような人は、以下のようにして放置時間を延長できます。

```
csosf01(81)% set autologout=300
csosf01(82)%
```

また、例えばカレントディレクトリの情報はシェル変数 `$cwd`⁸ に格納されていますので、`echo $cwd` などとして表示させることが出来ます。それからホームディレクトリの情報はシェル変数 `$home` に格納されています。これまた `echo $home` などとして表示させる事が出来ます。

`echo` コマンドなど、一般のコマンドでシェル変数を扱うときは常に `$` 記号がシェル変数の名前の前に付けます。`set` コマンドでシェル変数の名前を指定するときは `$` 記号を付けないことに注意してください。

環境変数の表示、設定

現在設定されている環境変数に何が有るかを調べるには `env` コマンド⁹を利用します。また、シェル変数の内容を変更するには `setenv` コマンドを利用します。

```
env
```

⁷特に `csh` と、それを模倣している `tcsh` には

⁸`current working directory` の略なのです

⁹UNIX の種類によっては `setenv` コマンドまたは `printenv` コマンドで行うというものもあります。

現在設定されている全ての環境変数を表示します。

```
setenv VAR string
```

環境変数 `VAR` の内容を `string` に設定します。

また、`echo` コマンドを使って特定の環境変数の内容を表示することも出来ます。`echo $VAR <Return>` とすると、`VAR` という名前の環境変数の内容を表示します。環境変数は慣例として英大文字と数字で構成されています。

例えばプリンタの操作をするときは `lpr -Pcspr01 sample.tex` などして常にプリンタ名を指定するようにしていました。環境変数 `PRINTER` が設定されている場合は、プリンタに関する各種コマンドは (`-P` オプションを省略した場合) 環境変数 `PRINTER` に指定されたプリンタに対して処理を実行しようとします。

```
csosf01(81)% setenv PRINTER cspr01
csosf01(82)% lpr sample.tex
csosf02(83)%
```

`echo` コマンドなど、一般のコマンドで環境変数を扱うときは常に `$` 記号が環境変数の名前の前に付けます。`setenv` コマンドで環境変数の名前を指定するときは `$` 記号を付けないことに注意してください。

6.4.4 リダイレクション

UNIX の標準的なコマンドには、標準入力と標準出力と呼ばれる入出力を処理対象としているものが多くあります。普通にシェルを使っているときは、標準入力はキーボード入力、標準出力はディスプレイ表示に割り当てられています。

この例として `bc` コマンドを挙げておきます。`bc` コマンドを実行して四則演算をキーボードからタイプすると、画面上に計算結果を表示します。これはつまり `bc` コマンドは標準入力から四則演算を受け付けて、計算結果を標準出力に流していると言うことの結果です。

さて、標準入力、標準出力はそれぞれ他のファイルに割り当てることが出来ます。この標準入出力のファイルへの割り当て直しを「リダイレクション¹⁰」と呼んでいます。

標準入力のリダイレクション

標準入力のリダイレクションは「`<`」記号で表現します。

`コマンド [オプション] [引数...] < ファイル名` と書いて、コマンドの標準入力を指定のファイルに割り当て直します。例えば `bc` コマンドの標準入力をキーボードではなくファイルに割り当てて見ましょう。`bc < ファイル名 <Return>` です。

まず `bc.in` というファイル名で、四則演算を並べたファイルを用意します。Emacs で新規に作成、編集して下さい。以下に出来上りを `cat` した例を示します。簡単な四則演算を 3 行用意しました。

```
csosf01(85)% cat bc.in
1 + 2
2 * 3
4 / 2
```

¹⁰ redirection 宛名を変える

```
csosf01(86)%
```

bc コマンドの標準入力に対するリダイレクションによってファイルの内容を四則演算の式の列として実行され、画面すなわち標準出力に結果が表示されます。

```
csosf01(86)% bc < bc.in
```

```
3
```

```
6
```

```
2
```

```
csosf01(87)%
```

標準出力のリダイレクション

標準出力のリダイレクションは「>」記号で表現します。

コマンド [オプション] [引数...] > ファイル名 と書いて、コマンドの標準出力を指定のファイルに割り当て直します。例えば bc コマンドの標準出力をディスプレイではなくファイルに割り当ててみましょう。bc > ファイル名 <Return>です。

```
csosf01(88)% bc > bc.out
```

```
1 + 2
```

```
2 * 3
```

```
quit
```

```
csosf01(89)%
```

cat コマンドでリダイレクションによってファイルに結果が残っていることを確認しましょう。

```
csosf01(89)% cat bc.out
```

```
3
```

```
6
```

```
csosf01(90)%
```

標準出力のリダイレクションで追加書き

標準出力のリダイレクションを既存のファイルに対して指定すると、単純に上書きしてしまい、そのファイルの元の内容は失われてしまいます。しかし追加書きするようなリダイレクションもあります。標準出力の追加リダイレクションは「>>」記号で表現します。

コマンド [オプション] [引数...] >> ファイル名 と書いて、コマンドの標準出力を指定のファイルに追加するように割り当て直します。例えば bc コマンドの標準出力をファイル追加するように割り当ててみましょう。bc >> ファイル名 <Return>です。

```
csosf01(90)% bc >> bc.out
```

```
5 * 2
```

```
quit
```

```
csosf01(91)%
```

cat コマンドでリダイレクションによってファイルにもとの結果に加えて新しい結果が追加されているのを確認しましょう。

```
csosf01(91)% cat bc.out
3
6
10
csosf01(92)%
```

標準入力、出力の両方のリダイレクション

先ほどの `bc.in` ファイルを利用して標準入力をそこから、標準出力をこれまた先ほどの `bc.out` ファイルに割り当てて見ましょう。以下にその例を挙げます。例では `cat` コマンドでファイルの中身を確認しています。

```
csosf01(92)% bc < bc.in > bc.out
csosf01(93)% cat bc.out
3
6
2
csosf01(94)%
```

6.4.5 パイプ

`bc` などのように、標準入力から何かを受取り、標準出力にその結果を返すようなコマンドをフィルタコマンド¹¹と呼んでいます。

先ほどの例で、`bc.out` というファイルが結果として出来上がっていると思いますが、例えばこのファイルの中身を数字の小さいもの順に並べ替える¹²為に、`sort` コマンドの利用を考えます。`sort` もフィルタコマンドですから、実行は以下のようにする事になるでしょう。`-n` オプションはソートの順番を数値の小さいもの順にするためのオプションです。

```
csosf01(94)% bc < bc.in > bc.out
csosf01(95)% sort -n < bc.out
2
3
6
csosf01(96)%
```

単純なコマンドでも、それを組み合わせて実行すると、なかなか便利な¹³ものです。ところで上記のコマンド 2 つの組み合わせを、もっと簡単に表現できるような仕掛けが UNIX には用意されています。以下の例を見てください。

```
csosf01(97)% bc < bc.in | sort -n
2
3
6
csosf01(98)%
```

¹¹ あたかもフィルターのように働くという意味ですね。

¹² 並べ替える作業を一般的に `sort` (ソート) と呼んでいます。

¹³ ちょっと意味のない例しか挙げられませんでした

2つのコマンドを一行で書いています。リダイレクションを使わないために中間的に発生していた `bc.out` ファイルも必要ありません。このような、コマンドとコマンドをつなぐ「|」記号を「パイプ」と呼んでいます。パイプがあれば、パイプの左のフィルタコマンドは標準出力をパイプの右のコマンドの標準入力につながわせて実行くれます。

例えば `cat` コマンドもフィルタコマンドですから、上の例は以下のようにも書き換えられます。

```
csosf01(98)% cat bc.in | bc | sort -n
2
3
6
csosf01(99)%
```

パイプは幾つでも重ねて使うことができます。

あまり実用的な例ではありませんが、`echo` コマンドを使えば簡単な計算を標準入力からキーボード入力しないで処理するような組み合わせも出来ます。

```
csosf01(99)% echo '3 * 4' | bc
12
csosf01(100)%
```

UNIX ではパイプとリダイレクションを応用することによって、各種のコマンドを柔軟に組み合わせて実行することが出来ます。

6.4.6 シェルの鬼へのヒント

以下に少しだけシェルの鬼¹⁴へのヒントをあげておきます。

キーボードからの標準入力の終了

今回用意した `bc.in` ファイルなどは Emacs を使わなくても以下のようなやり方で作成できます。

```
csosf01(100)% cat > bc.in
```

`cat` コマンドは引数としてファイル名が与えられていない時は、入力を標準入力から行ないますから、それをキーボードから入力して出力先を画面ではなくファイルにすると言うわけです。

上のコマンドを実行して、タイプする内容は `1 + 2<Return>2 * 3<Return>4 / 2<Return>C-d` です。最後の `C-d` はファイルの終了を意味します。キーボードからファイルの終了¹⁵を入力するのは、大抵の場合この `C-d` です。

今まで黙っていましたが、`bc` コマンドは、サブコマンド `quit` ばかりでなく、このファイルの終了を受けとつても終了します。だからこそ `cat bc.in | bc` で文句も言わずにちゃんと処理を終了したのです。

複数のコマンドを一行に書く

「;」を利用して、複数のコマンドを一行に書くことが出来ます。例えば以下のような感じです。

¹⁴ここでキックの鬼と言ってもわかる人は、、、

¹⁵一般的に end of file と表現します

```
csosf01(101)% date ; hostname ; whoami ; id
1994年03月12日(土)16時13分33秒
csosf01
yasuda
uid=2126(yasuda) gid=700(admin) groups=500(clerk)
csosf01(102)%
```

ところで上の4つのコマンドの結果をぜんぶ一つのファイルにリダイレクションしたい場合は、以下のようになります。

```
csosf01(102)% ( date ; hostname ; whoami ; id ) > hostinfo.txt
```

このような書き方で、別々に処理することももちろん出来ます。

```
csosf01(103)% ( date ; hostname ) > hostinfo1.txt ; ( whoami ; id ) > hostinfo2.txt
```

一連のコマンドをファイルに書いておく

シェルもフィルタコマンドです。ですからファイルにコマンド列を書いておいて、それをシェルの標準入力に与えて処理させることが出来ます。cc環境標準のシェルはtcshですから、シェルのコマンド名はtcshです。以下に例を挙げておきます。

```
csosf01(104)% cat batch
echo 'Here is my session information.'
date
hostname
whoami
id
csosf41(82)% tcsh < batch
```

対話形式ではないので警告のメッセージが表示される時がありますが、とりあえず結果は表示されると思います。単純な一連の手続きであればこれで十分ですね。

例のように短いものであれば問題ないのですが、長い複雑なものを作ったりしたときには何らかの覚え書きを入れておきたい時があります。そのような時には#を利用します。#より右に書いたものは覚え書き¹⁶とみなされて、その部分を一連の実行するべき手続きとして扱わず、無視されます。以下に覚え書きを含めて書いた例を挙げておきます。

```
csosf41(84)% cat batch
#
#   Sample shell program. '95.2 Yasu.
#
echo 'Here is my session information.'
date
hostname
whoami
id      # show my username and etc.
csosf41(85)%
```

¹⁶ コメントなどと呼ぶ時もあります。

6.4.7 シェルよ永遠に

シェルの説明をすると言うのにあの話（もしくはあの話）をしないとは何事か！と、お怒りの方もおられるでしょう。シェルには非常に多くの機能があります。さすがに全部書くのは無理ですので、これ以降は付録の参考文献を参照して下さい。

第 7 章

Mule

表記方法として

- C-h コントロールキーを押しながら h を押す。
- M-x ABC エスケープキーを押した後に x を押して普通に ABC と打つ。
- M-C-x エスケープキーを押した後にコントロールキーを押しながら x を押す。
- <Space> スペースキーを押す。
- <Return> リターンキーを押す。
- <Delete> デリートキーを押す。

などというものがあります。それを踏まえて以下を読んで下さい。

なお、機械の種類によってコントロールキーの機能がオプションキーに変更されている場合があります。(マッキントッシュ等)

7.1 なにそれ?

Mule とは、日本語、英語だけでなく、世界中の多くの言語を統一的に同時に扱える様に拡張された Emacs です。Emacs というのは多機能エディタで、単に文章を書くだけでは無く、インターネットのニュースを読んだり、メールを読み書きしたり、他にも色々な事が出来るエディタです。Mule は、その Emacs の機能を全て継承して、更に、多くの外国語を使えるようにしたエディタという訳です。本当にたくさんの機能があり、全てを説明すると本が一冊出来てしまいますので、ここでは基本的な Mule の使い方を説明します。

7.2 Mule の仕組み

Mule はバッファと呼ばれる作業場所を持ちます。流れとしては、まずディスク上に保存してあるファイルから文書を読み込んでバッファに記憶するか、もしくは新規に何も記憶していないバッファを用意して、それを編集します。そして編集した文章をディスクにファイルとして保存します。

また、Mule はバッファを一つだけでなくたくさん一度に持つことができ、それらを切替えて画面に表示することができます。Mule を起動した時にでてくる画面は「ウインドウ」と呼ばれる大きさが可変のもので、ここにバッファの内容が表示されます。例えば普通に Mule を起動した時はウインドウが一つと、下の方にモードラインという編集集中のバッファの情報が表示される行 (反転している行)、そして一番下にエコーラインという Mule のメッセージを表示する行が表示されます。

このウインドウは幾つにも分割することができ、それらのウインドウに幾つかのバッファを同時に表示

させることができます。また、同じバッファを二つ以上のウインドウに表示させることも可能です。その場合、どれかを編集すれば残りのウインドウにもそれが反映されます。

7.3 これだけ覚えたら Mule は使える!

これ以降の記述は UNIX の基本的な部分を一通り理解している事を前提として書きます。まず、UNIX のコマンドを入力できるターミナル画面がありますね。では、おもむろに **mule** と打ち込んでリターンキーを押してみてください。えむ、ゆー、える、いー、りたーんです。すると英語ばっかりの画面が現れます。

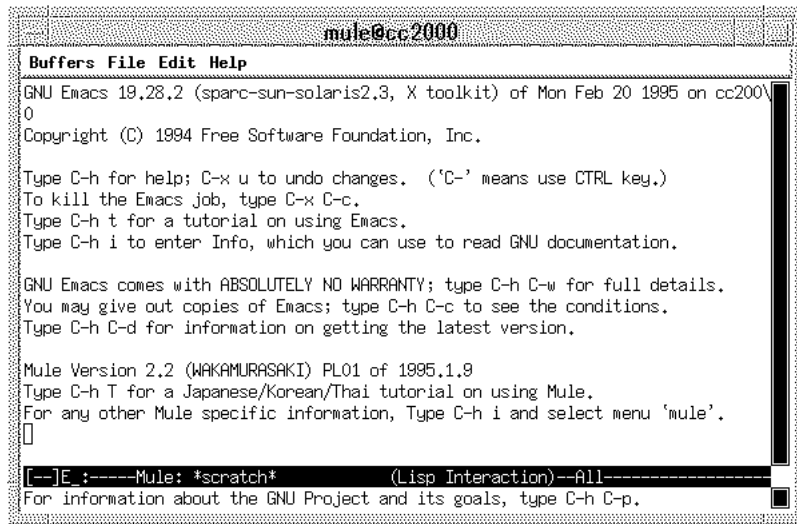


図 7.1 Mule 起動画面

ですが別に難しいことが書いてある訳ではありません。この画面は初心者の人にまず知っておいて欲しいことが書いてあります。ヘルプ、言い換えると簡単なオンラインの説明書の出し方やチュートリアルを始め方なんか書いてあります。

チュートリアル

チュートリアルというのは日本語にすると「個別指導」で、ここでは画面に従ってキー操作をしながら使い方を覚える入門用の練習のことをいいます。例えば **C-h T Japanese** で日本語のチュートリアルが起動すると書いてあります。チュートリアルをやりたい時はコントロールキーを押しながら **h** を押して、次に大文字の **T** ですからシフトキーを押しながら **T** を押します。そして **Japanese**¹ と入力してリターンキーを押すと始まります。チュートリアルの終わり方はいくつかありますが、Mule ごと終わらせるのが簡単でいいでしょう。C-x C-c です。チュートリアルは是非、一度やってみてください。

カーソルの移動

では本題のエディタの使い方に移ります。超基本部分から覚えていくことにしましょう。カーソルの移動はカーソルキーで行います。矢印の書いてあるキーです。間違えた場合はデリートキーで消します。普通はキーボードの上の段の一番右のキーですね。機械によってはバックスペースキーになっている場合も

¹ 先頭の J は大文字で後は小文字です。

ありますのでデリートキーでうまくいかないときは試してみてください。改行はリターンキーです。いろいろとキーボードを触って動作を確認して下さい。

変になったら

それから、まず覚えておいて欲しいことですが、この先の操作で間違えて、自分の意図しない変なメッセージが表示されてしまった場合は、**C-g** として下さい。これは指示途中の操作を取り消すコマンドです。変になったときは傷の浅いうちにこれをする方が無難です。下手に色々触るとどんどん変になることがよくあります。それでもダメなら周りのよく知っていそうな人に聞いて下さい。

名前をつけよう

話を戻しましょう。単に Mule を立ち上げた場合 ***scratch*** というバッファが開きます²。下の方に ***scratch*** と書かれていますね。これが今編集しているバッファの名前です。

scratch は「走り書き」の名のとおりに落書き扱いですからそのまま ***scratch*** バッファに何か書いても Mule を終了すると保存されずに消えてしまいます。ですからこれを保存するために、まず何か名前をつけてあげないといけません。後から名前をつける方法もありますが³、ここでは最初に名前をつけることにします。まず Mule が起動した時に **C-x C-f** とします。すると

```
Find File: ~/
```

と表示してファイル名を聞いてくるのでそこで何かこのファイルにつける名前を指定します。例えば **test** とでもしましょうか。するとエコーラインに **(New file)** と表示されてモードラインの ***scratch*** と書かれていた場所に **test** と表示されますね。

続けて何か書いてみてください。なにか書いてから終わろうと **C-x C-c** をするとメッセージがでできます。

```
Save file /NF/home/g910/ozaki/test? (y, n, !, ., q, C-r or C-h)
```

これは今のバッファが変更されているのでそれを保存するかどうかを聞いているのです。保存するなら **y**、しないなら **n** を押して下さい。**n** と答えたときは本当にしないのかどうか確認してくれます。親切ですね。ここでは保存を選びましょう。**y**、と。他にも選択肢が表示されていますが、今は気にしないでいいでしょう。どうしても気になるのなら **C-h** を実行して出てくるメッセージを読んで下さい。

これだけで Mule を使うことはできます。後は知れば知るほど便利に使うことができるのが Mule の特徴で、初心者から上級者まで幅広く自分のレベルにあった使い方ができます。

7.4 もうちょっと覚えよう

ファイルの呼びだし

当然、前に保存した続きを書きたい事があります。そういう時は Mule を立ち上げる時に続けて保存したファイル名を打ちます。さっき保存した **test** なら **mule test** と後ろに続けて **test** と打ち込みます。また、ファイルの読み込みにはもう一つやり方があり、Mule が立ち上がった状態で **C-x C-f** とすると

```
Find File: ~/
```

² バッファというのは Mule の作業を行う場所で、一度にいくつも開くことができます。Mule は文書をバッファで編集し、ディスク上のファイルと読み書きします。

³ **C-x C-w** で名前をつけて保存することが出来ます。

と表示してファイル名を聞いてくるのでそこで指定してやるとそのファイルをバッファに読み込みます。あれ、この C-x C-f というキー操作は先ほど新しいバッファに名前をつける時に行ったものと同じですね。実は C-x C-f は本来ファイルを読み込むための操作なのですが、そのファイルが存在しないと (**New file**) としてその名前で新しいファイルを作るのです。先ほどの場合は **test** というファイルが無かったから (**New file**) になった訳です。また、この場合のファイル名の入力には補完機能を使う事ができます。補完機能とはファイル名、例えば **test** という場合、他に **t** で始まるファイルがなければ、**t** だけを入力した時点で **<Space>**⁴ を打つと **test** とすべて補完されるという便利な機能です。また、他に **t** で始まるファイルがあればそれをバッファに表示しますので、お目当てのファイルの 2 文字目、3 文字目を入力して、他と区別できるところまで入力したら **<Space>** でまた補完します。同じ様に M-x *Command* によるコマンドにも補完機能が使えます。

他に、C-x **i** で前に作ったファイルをカーソル位置に差し込むことができます。C-x C-f と似たようなものですが、読み込んだファイルがカーソルの位置に差し込まれます。

大きなカーソル移動

ところで、慣れてきて文章が長くなるとカーソルを移動するのが面倒くさく感じてきます。そういう時は M-v と C-v でそれぞれ 1 ページ分上下に移動します。また、C-a と C-e は行頭と行末への移動です。もっと大きく動きたいなら M-< と M-> でバッファの先頭と末尾へ移動します。これだけ覚えておけば作業が大幅に楽になること請け合いです。

他にも M-x *goto-line* として行番号を入れるとその行へ飛びますし、今の行番号が知りたければ M-x *what-line* で表示してくれます。

C-1 も覚えておくと便利です。カーソルを画面の中央に移動するコマンドなのですが画面が乱れた時に書き直す役目も果たしてくれます。UNIX を良く使うようになると、Mule を使っている最中に talk コマンドや write コマンドで誰かから呼びかけられる事もあります。そうすると画面は相手の呼掛けのメッセージでぐちゃぐちゃになってしまい、折角編集していた貴方の文章が消えてしまいます。でも安心して下さい。これは画面上だけで実際には内容は変わっていないのです。C-1 の画面書き直し機能はこういう時に役に立ちます。後は余り使いませんから必要に応じて覚えるといいでしょう。

カット & ペースト

次にカット & ペーストを説明しましょう。文字のコピーですね。

まずコピー元を指定します。範囲の先頭で C-**<Space>** (コントロールを押しながらスペース) でマークをセットします。そして範囲の末尾へカーソルを移動します。このマークからカーソルまでの範囲を **リージョン** と呼びます。日本語で言うなら矩形 (くけい) ですね。セット位置からカーソル位置までを **リージョン** としますので別に末尾でセットして先頭へカーソルをもっていても構いません。

次に M-w でその範囲の文章を覚えます。また、C-w なら覚えると同時に削除する (カット) ので移動する時や単に消したい時に使えます。場合に応じて使い分けて下さい。

そしてコピーしたい先までカーソルを移動して C-y で覚えていた文章を挿入 (ペースト) します。マーク位置を忘れたときには C-x C-x でマーク位置とカーソルの場所を入れ替えるのでそれで確認して下さい。尚、C-x C-x はうまく使うと離れた場所から元の場所へ戻るときに使えます。

また、C-k でカーソル位置から行末までをカットします。C-k は行頭で 2 回連続して使うと 1 行消去することができます。これで消去した内容は (連続で消す限り) 覚えていますので C-y で挿入する事ができます。行の移動なんかが楽にできますね。

⁴**<Tab>**でも補完機能が働きます。**<Space>**はハイフンなどの区切り記号で一度止まるのに対して、**<Tab>**は可能な限り全てを補完するのでこちらの方も合わせて使うと便利です。

Undo

ところで色々編集しているうちに間違えてしまい、元に戻したい時があると思います。そういう時は C-u (アンダーバー) でいくらでも元に戻せます。**Undo**(アンドウ) という機能です。因みに、C-x u でも **Undo** になります。

保存

しかし、いつ何時、どんな理由で編集内容が壊れるか分かりません。一応ある程度書いたら内容を保存しておく習慣をつけておいた方がいいでしょう。精神衛生上もいいですね。C-x C-s で今のファイル名で保存します。

また、別の名前をつけて保存したいのなら C-x C-w として下さい。

```
Write file: ~/
```

このように表示されましたね。ここで好きなファイル名を書けばその名前で保存します⁵。

7.5 日本語入力

では次は漢字の入力にいきましょう。mule は Wnn(うんぬ) という日本語変換システムで日本語を入力します。

Wnn⁶ は 京都大学数理解析研究所、オムロン株式会社、株式会社アステックの3者によって開発されたかな漢字変換システムです。cc 環境ではこの Wnn を標準的な漢字変換システムとして採用しています。

EGG⁷ は電子技術総合研究所の戸村哲氏が中心となって開発した、Mule 上で Wnn を利用するためのシステムです。cc 環境ではこの EGG を Mule 上での標準的なかな漢字変換システムとして採用しています。

EGG のモード切り替え

さて、Mule 上で漢字をタイプするためには、まず EGG をローマ字かなモードにします。Mule が起動されている状態で、C-\ です。(キーボードによっては C-\ の代わりに、C-¥ かもしれません。) モードラインの左端に注目してください。以下のようにするのが判ると思います。

```
[あ]E_:-----Mule: *scratch*           (Lisp Interaction)--All-----
```

この状態から元に戻るためには、再び C-\ です。モードラインが元に戻りましたね。C-\ でモードラインの左端がくるくる変わるのを確認してください。

```
[--]E_:-----Mule: *scratch*           (Lisp Interaction)--All-----
```

モードラインの左端が [--] となっている状態を EGG の「透過モード」と呼んでいます。(透過モードと言うのは、つまり今までどおりアルファベットのタイピングがそのまま行える状態です。) [あ] となっている状態を「ローマ字かなモード」と呼んでいます。ローマ字かなモードにしておくと、タイプしたアルファベットはまず EGG に受けとられ、そこでローマ字として解釈されてひらがなが表示されます。それから漢字変換に関するキーを操作して目的の漢字かなまじり文へと変換するのです。

⁵ 但し既に存在するファイルを指定した場合、上書きしてしまうので注意して下さい。

⁶ 「うんぬ」と読みます。名前の由来は「私の名前は中野です」の略から来ているそうです。

⁷ 「えっぐ」と読みます。名前の由来は「たくさんまたせてごめんなさい」の略「たまご」から来ているそうです。

7.5.1 ためしに変換

ローマ字かなモードにして、アルファベットで「wata sinonamaehanakanodesu」とタイプしてください。打ち込んだアルファベットが縦棒にはさまれながら次々とひらがなに変換されて行くのが判るでしょう。

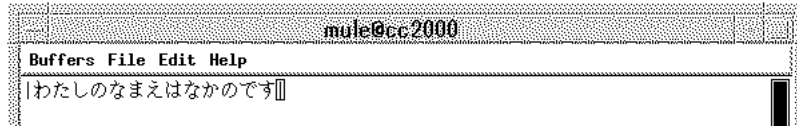


図 7.2 EGG のフェンスモード

この縦棒を EGG ではフェンスと呼び、変換途上のかな文字がフェンスにはさまれている状態をフェンスモードと呼んでいます。

フェンスの中の文字を編集するには普通の Mule の編集のときと同じくカーソルキー (←や→) と <Delete> キーです。カーソルキーが効かない場合は C-b でカーソル左、C-f でカーソル右と同じ動きをします。

ローマ字が全てかなに変換されたら <Space>⁸ キーを押すことによって今度はかなを漢字に変換する作業が始まります。

注意

あなたがそのコンピュータで初めてかな漢字変換を行うときは、以下のような問い合わせが行われます。

```
頻度ファイル "!/NF/home/g910/ozaki/Wnn/kihon.frq" が無いよ。作る?(y or n)
```

これらの質問には全て y と答えてください⁹。

再び注意

あなたがその Mule で初めてかな漢字変換を行うときは、エコーラインに「ホスト local の WNN を起動しました」などというメッセージが表示されて、かな漢字変換が開始されるのに若干時間が掛かるかも知れません。

かな漢字変換が始まると、モードラインの左端が以下ようになります。

```
[漢]E_:---*-Mule: *scratch* (Lisp Interaction)--All-----
```

フェンスの中は以下のようになっているでしょう。このモードラインの左端が [漢] となっている状態を「漢字変換モード」と呼んでいます。

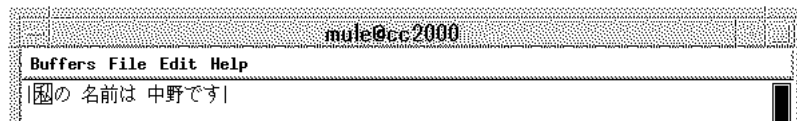


図 7.3 EGG の漢字変換モード

⁸ C-w でも構いません。

⁹ 結構何回も y と答えなければなりません。今試してみたら合計 12 回も必要でした！

ここで<Return>とすれば現在表示されている漢字かなまじり文の候補「私の名前は 中野です」が採用されて、Mule にタイプされます。フェンスがなくなって、漢字が Mule の普通の操作で編集できる状態になりましたね。

この作業を「確定」と呼んでいます。これでまず一つ漢字の文章がタイプされました。一安心です。

再変換

一度変換するだけで自分の目的の漢字に変換してくれれば良いのですが、先ほどのようにうまく行くことばかりではありません。そこで、EGG が最初に挙げてくれた漢字の候補を再変換しなければなりません。一度でうまく行かない例として、今度は「tonarinokyakuhayokukakikuukyakuda」とタイプして <Space> で変換してください。恐らく以下のようなになるでしょう¹⁰。

|戸なりの 客は 良くか 聞く 右-客だ|

おいしいですね。ここでは最終的に「隣の客は良く柿食う客だ」に変換し直す例をあげます。

かな漢字変換では変換は文節単位に行なわれます。はじめはローマ字でタイプされた一連のひらがなの列から EGG が自分で文節の区切りを見つけて変換します。上記の例ではその結果「戸なりの」「客は」「良くか」「聞く」「右」「客だ」と文節の区切りを付けました。EGG の漢字変換モードでは文節の区切りを一つの空白で表します。「戸なりの」と「客は」などの間にそれぞれ一つの空白があるのが判りますね。

次の候補を上げる

今、カーソルは「戸なりの」の上にあると思います。ここの文節の区切りはこれであっていますから、そこで<Space>を押すことによって次の候補が表示されます。この例では一度<Space>を押すと「隣の」が現れると思います。現れなかったら、現れるまで<Space>を押して下さい。あまり急いで<Space>を押すすぎると肝心の「隣の」が現れても勢いでもう一度<Space>を押してしまう時もあるでしょう。そういう時は C-p で一つ前の候補を再表示する事が出来ます。

さて、

|隣の 客は 良くか 聞く 右-客だ|

こうなりましたね。次の文節に移りましょう。C-f を押して下さい。「客は」はそのままで合っていますから、もう一度 C-f を押して「良くか」まで移動しましょう。もし行き過ぎてしまったら C-b で戻れます。

文節の区切りを変える

カーソルは「良くか」の上に移動しましたね。「良くか」は「良く」に文節の区切りを訂正しなくてはなりません。文節の区切りを縮めるには C-i です。一度押したら「良く」になりますね。さあ次の文節に移動しましょう。

|隣の 客は 良く 課 聞く 右-客だ|

「課」に移動しましたか。今度は文節を伸ばしてやる必要がありますね。C-o を押して下さい。「書き」になりますね。なったら<Space>で「柿」になるまで変換して下さい。先にも書きましたが、変換が行き過ぎてしまったら C-p で戻れます。

さて、今の「課」を「柿」にした時点で後ろが「食う客だ」になりましたね¹¹。さあ、これで全部が目的の漢字になりました。<Return>で確定しましょう。

¹⁰ 漢字変換を行う度に良く使う漢字が先に出てくるようになるので必ずしもこのような変換になるとは限りません。

¹¹ もしなっていないければ同じように変換して下さい。

7.5.2 特殊な文字列の入力

ちょっと普通でない文字を入力する必要がある場合の例をあげましょう。たとえば「今 America にいるタロウと〇〇〇しました。」¹²と入力したいとします。

次のように入力します。まず **ima** と打てば「いま」が入力されますね。次に入力モードをアルファベットに変更します。**q** を押して下さい。モードラインの[あ]が<aa>になりますね。この状態で **America** と入力します。できたら **C-q** で元の日本語入力モードに戻します。続いて **niirutarouto** で「にいたろうと」を入れて下さい。

まずここまでで一度変換しましょう。<Space>を押して下さい。すると

```
|今 Americaに いたろうと|
```

となりますね。最初の二つの文節はあっていますから **C-f** を二回押して「いたろうと」の文節まで移動して下さい。そして **C-i** を四回押します。すると

```
|今 Americaに いる 太郎と|
```

となります。C-fで「太郎と」に移動します。今は、「太郎」をカタカナで書きたい訳ですから変換し直しましょう。Spaceを押して出るまで待ってもいいのですがちょっと待って下さい。文節を「たろう」に縮めてカタカナにしましょう。C-iを一回押した後、M-kとしてみて下さい。M-kは文節をカタカナに変換するキー操作です。

```
|今 Americaに いる タロウ 戸|
```

こうなりましたね。あとは「戸」をひらがなにします。C-fで移動してM-hです。M-hは文節をひらがなに交換します。やっと目的の文字になりました。Returnで確定しましょう。

さて、次の「〇」ですが、どうやって入力しましょうか？もっとも簡単なのは「まる」で交換することです。しかし「〇」はたまたまこういう入力可能な記号で、普通このような読みがなによる入力はできません。そこで、C-^で記号入力モードにします。最初の一回は少し表示が出るまでに時間がかかります。エコーラインに **Garbage collecting...done** のような表示がつかたり消えたりしますね。点滅が止まったな、と思ったらC-1を押して下さい¹³。すると

記号入力: 0. JIS 入力 1. 記号 2. 英数字 3. ひらがな 4. カタカナ 5. ギリシャ文字

エコーラインにこのように現れますので、該当する番号を選んで下さい。なお、ロシア文字、罫線、部首入力、画数入力、第一水準、第二水準、補助漢字という入力モードもありますのでこれらを使いたい時はC-nやC-pで切替えて選んで下さい。「〇」は **1. 記号**の部類ですから **1** を押してReturnを押します。C-nを押して次候補を出していくと7回ほどで「〇」が出てくると思います。「〇」の左に出ている数字、もしくはアルファベットを押してReturnで確定して下さい。画面に「〇」が表示されましたね。もう一度C-^を押して下さい。先ほどの「〇」の位置にカーソルがありますから同じように「〇」を追加します。後は「simasita.」で入力できますね。

補足

C-^を押すと二度目からは前に入力した記号の位置から始まります。他の種類の記号を入力したい時はC-gで一つ前の選択画面に戻して下さい。

¹² 〇〇〇は伏せ字の意味です。多分「でんわ」とでも入るのでしょう。:-)

¹³ C-1 は実は画面再表示のコマンドなのですが、最初に記号入力画面にした時は **Garbage collecting...done** と表示したまま、次の画面を表示してくれないのでC-1を押します。

また、記号の入力は、例えば「○」なら **z1** のように **z** と何かのキーという形でいくつか登録されています。これは一覧として A.3 で紹介していますので、参考にして下さい。

どれでもいいですから一番入力し易いものを覚えて下さい。

7.5.3 まとめ

以下にキー操作のまとめを示しておきます。

C-\ 漢字モードと英語モードの切り替え

ローマ字かなモードでのキー操作

<Delete>	タイプミスの修正、一文字削除
←または C-b	カーソル一文字左
→または C-f	カーソル一文字右
<Space>または C-w	かな漢字変換の開始

漢字変換モードでのキー操作

←または C-b	カーソル一文節左
→または C-f	カーソル一文節右
C-o	文節を伸ばす
C-i	文節を縮める
<Space>または C-n	次候補の表示
C-p	前候補の表示
M-s	候補の一覧をエコーラインに表示する。数字による選択かカーソルキー（もしくは C-b ,C-f ,C-p ,C-n ）による選択を行い<Return>で確定。 候補一覧表時から抜けて元に戻るには C-g 。
M-k	文節をカタカナに変換する。
M-h	文節をひらがなに変換する。
C-^	記号入力モードにする。数字による選択かカーソルキー（もしくは C-b ,C-f ,C-p ,C-n ）による選択を行い<Return>で確定。 候補一覧表時から抜けて元に戻る時、及び前の選択画面に戻る時には C-g 。

7.5.4 ローマ字入力のヒント

「たんい」（もしくは「んお？」など）のように「ん」の次に母音が来るのです

「ん」を単独でタイプするために N (大文字) と n' が用意されていますので、それを利用してください。
「taNi」もしくは「tan'i」とタイプすれば「たんい」とローマ字変換されます。

「あっ」（もしくは「んあ？」など）のような小さい「っ」「あ」を入力したいのです

小さい「あいうえお」などは、x を前に付けてタイプします。「xa」とタイプすれば「あ」とローマ字変換されます。「xtu」で「っ」、「xyu」で「っ」です。

その他

変換せずにひらがなのまま確定してしまった時はその範囲をリージョン指定して `M-x henkan-region`、間違った漢字で確定してしまった時は、その範囲をリージョン指定して、`M-gyaku-henkan-region` とすると幸せになれることでしょう。

また、変換とは別に通常状態でリージョンを指定して `M-x zenkaku-region` や `M-x hankaku-region` でアルファベットや数字を全角、半角に変えられます。

`M-x upcase-region <Space>`や `M-x downcase-region <Space>`で大文字、小文字にすることも出来ます。

7.6 ちょっと中級者

7.6.1 単語登録

ちょっと一般的でない専門用語や人名、地名等はなかなか変換できず、漢字 1 文字ずつ入力しなければならないこともありますね。その単語を 1 回きりしか使わないのであればそれでもいいですが、何度もそういう単語を使う必要がある場合はちょっと面倒ですね。そういう時は単語を辞書に登録することができます。例えば「神山（こうやま）」に登録してみましょ。まず登録したい単語をリージョン指定します。「神」の位置にカーソルを合わせて `C-<Space>` でマークをセットして、「山」の次の文字の位置にカーソルを移動させます。ここで、`M-x toroku-region` として下さい。

[あ] 辞書登録『神山』 読み :

エコーラインにこの様に表示されますね。ここで「こうやま」と入力して`<Return>`を押して下さい。

登録辞書名: `0.private`

するとこのようになりますので、もう一度`<Return>`を押して下さい。そうすると品詞名を選択する画面になります。

品詞名: `0. 普通名詞/ 1. 固有名詞/ 2. 動詞/ 3. 特殊な動詞/ 4. 動詞以外の用言/`

これらの他に `C-n` を押せば

詞名: `0. その他の独立語/ 1. 接頭語, 接尾語/ 2. 単漢字 3. 疑似品詞/`

に切り替わりますので、該当する番号を選択して`<Return>`を押して下さい。「神山」は固有名詞ですから最初の **1. 固有名詞** を選択します。

品詞名: `0./ 1. 人名 2. 地名 3. 人名&地名 4. 固有名詞`

するとこのようになりますから、**2. 地名** を選択して`<Return>`を押します。これで「登録しました」という旨のメッセージが出て「神山」が辞書に登録されました。「こうやま」と書いて変換してみてください。「神山」に変換されましたね。他の場合、例えば動詞なども同様にして登録して下さい。

このように普段良く使う単語を登録することでどんどん漢字変換の効率が良くなります。また、それ以外にも **Wnn** は使用頻度を数えて、高いものを最初に変換するようにしますので使えば使うほど変換効率が上がっていきます。最初は変な変換をするかも知れませんが、どんどん辞書を鍛えて使い易いものにして下さい。

登録単語の抹消

M-x edit-dict-item として下さい。すると

[あ] よみ :

とエコーラインに現れますので消したい単語の読みがなを入力して<Return>を押して下さい。するとずらずらっとその読みがなで登録されている単語が出てきますので上下移動で消したい単語の行にカーソルを動かして **d** を押して下さい。その行の頭に **D** というマークがつきますね¹⁴。消したい単語の行、全てにマークをつけたら **x** を押して下さい。すると

いいですか? (yes or no)

と聞いてきますから **yes**<Return>か **no**<Return>で答えて下さい。元の画面に戻って **q** を押すと辞書編集を終ります。

7.6.2 検索

検索

検索は **C-s** とすると文字列を聞いてくるので探したい文字列を入力して下さい。すると入力した文字列の場所へジャンプします。見つけたい文字列は入力できたけれども、ジャンプした先が期待の物と違う場合は続けて **C-s** をすると同じ文字列で検索を続けます。行き過ぎた時や上方向に検索したい時は **C-r** です。目宛のものが見つけられたらカーソルを移動させれば検索モードから抜け出します。もし検索中に元の場所(検索を始めた場所)に戻りたくなったら **C-g** で戻る事が出来ます。

尚、検索を終えた後、続けて同じ文字列を検索したい時は **C-s C-s**、または **C-r C-r** とするとそれぞれ前の文字列で前後に検索をしてくれるので、再度文字列を入力する必要はありません。

置換

置換は **M-%** の後、検索文字列、置換文字列を入力した後、**y** で置換、**n** でパスします。また、**!**で残り全部を確認せずに置換を行ないます。

7.6.3 画面分割

Mule は画面を何分割かして幾つかのウインドウにバッファを同時に表示することができます。

C-x 2 でカーソルがあるウインドウを上下に二分割します。それらのウインドウ間のカーソル移動は **C-x o** です。また、**C-^** でカーソルのあるウインドウサイズを縦方向に拡大します。逆にバッファ表示を減らすときは **C-x 1** でカーソルのあるウインドウ以外を隠しますし、**C-x 0** (ゼロ) でカーソルのあるウインドウを隠します。ただし、隠すだけで完全に消去はしませんし、保存もしません。注意して下さい。

ふたつ以上のウインドウに同じバッファを読み込むことも可能ですし、違うバッファを **C-x C-f** で読み込んで同時に編集することも可能です。前者の場合はそれぞれのウインドウのバッファは違う場所を表示できますが実体は同じなのでどちらかを編集すると他のウインドウの同じバッファにもそれが反映されます。

¹⁴ただし削除できるのは **private** 辞書に登録したものだけです。

7.6.4 バッファリスト

Mule は同時に幾つものバッファを扱えます。C-x C-b で現在のバッファリストを表示させることができ、バッファを切り替えるには C-x b でバッファ名を続けて指定します。幾つものバッファを開いている時はバックグラウンドのバッファの状態 (保存したか否か) に注意して下さい。画面分割と併用すると違う文章からのカット&ペーストが簡単にできます。

7.6.5 モードライン

モードラインとは Mule の下の方にある ***scratch*** 等のバッファ名やそのバッファの状態を表示する行です。例えば普通に Mule を立ち上げたときは

```
[--]E_:-----Mule: *scratch*          (Lisp Interaction)--All-----
```

こう表示されます。これは前から順に説明すると、

[--]

日本語入力の際に使われる領域です。

E_

バッファのファイルの読み書きに使用する漢字コードを表します。色々ありますが、ここでの E_ は *euc-japan* を表しています。EUC は Extended Unix Code の略で、主として UNIX の世界で広まってきたコードで、Mule では何も指定しなければこれが用いられます。JIS は通信の方でよく用いられ、J_ と表されます。これはいわゆる日本工業規格ですね。中には絶対 JIS コードで書かなければいけないファイルなどもあります。何かパソコンとデータをやりとりする人は SJIS も使うことがあるかもしれません。これは Shift JIS の略で S_ と表されます。DOS と PC の世界で使われています。

ファイルへの書き込みの際の漢字コードを変更するには C-x C-k f です。<Space>を押すと一覧が現れますので目的のコードを選んで下さい。

バッファの変更状態を表す領域。変更されてまだ保存されていない時は --*-、書き込み不可の時は --%- となります。

scratch

バッファ名です。

Lisp Interaction

この位置には Mule の動作モードを表示します。Fundamental モードや C モード、Fortran モード、T_EX モード、gnus モード、mh-e モード、dired モード等、Emacs-Lisp と呼ばれる言語で用意された、たくさんのモードがあります。

All

All はバッファの全てが画面に表示されていることを示し、他に **Top** (バッファの始めが表示されている)、**Bot** (最後が表示されている)、**num%** (全体を 100% としてどのあたりが表示されているか) があります。

7.6.6 Dired モード

Mule のファイラー（ファイル操作）環境です。ファイルの呼びだしやコピー、リネームや消去などが簡単にできます。

M-x `dired` <Return> で **Dired モード** に移ります。（ファイル選択時にも起動します。）

カーソルキーで移動して目的のファイルまで移動します。ディレクトリを移動したい時はディレクトリにカーソルを合わせて **f** を押して下さい。ちなみに終了は C-x **k** です。

以下が目的のファイルの位置でできる機能一覧です。

- f カーソルのあるファイルを読み込む
- c カーソルのあるファイルを別ファイルにコピーする
- r カーソルのあるファイルを別の名前に変更する
- d ファイルに消去する印を付ける
- # # 印の付いた自動セーブファイル全てに消去印を付ける
- ~ ~ 印の付いたバックアップファイル全てに消去印を付ける
- u ファイルに付いた消去印を消去する
- x 消去印の付いたファイルを実際に消去する

便利な使い方としては `honyarara~` のようなバックアップファイル¹⁵をまとめて消したい時です。そのディレクトリで **Dired モード** の **Mule** を立ちあげて、`~` を押して、続けて **x** を押せば本当に消すかどうか、消すファイルの候補リストと共に確認をとってくるので、**yes** と答えればそれらを消してくれます。手で **rm コマンド** 等を使うより、速く、間違いなく消すことが出来ます。

7.7 上級者に向けて

Mule は個人個人が自分に使いやすいようにカスタマイズする事ができます。ただしカスタマイズに不具合があると最悪、Mule が動かなくなったりします。変更するときは十分な知識を持って自分の責任で行って下さい。ここでは詳しく触れずに幾つか簡単な例を A.3 で挙げておきます。後は必要に応じて勉強して下さい。

7.8 X 環境での Mule

X 環境での説明に入る前にマウスについて触れておきます。**Mule** はボタンが 3 つ付いたマウスで使えるように設計されています。その 3 つのボタンを、以後左ボタン、中ボタン、右ボタンと呼びます。さて、ボタンが 3 つ付いたマウスを使っている場合はそれで良いのですが、2 つ、ないしは 1 つのマウスもあります。ボタンが 1 つしかないマウスの場合は以下の説明の内、左ボタンの機能しか使うことが出来ません。しかしボタンが 2 つの場合は両方のボタンを同時に押すことで、中ボタンの代わりになります。

X 環境で **Mule** を使う場合、そうでない環境に比べて、少し便利に使うことができます。

一番使い勝手が違うのは文字カーソルの移動がマウスで出来ることでしょうか。移動先にマウスカーソルを合わせて左クリックでそこに文字カーソルを移動させることが出来ます。また、ボタンの役目は以下の通りです。（実はこれは X 環境の機能です。）

¹⁵ Mule はファイルを上書きする時に、前の古い内容のファイルを後ろに をつけたファイル名で置いておきます。これをバックアップファイルといいます。

- | | |
|--------------|-------------------------------------|
| 左ボタンでのドラッグ | その範囲をリージョンとして記憶します。(M-w) |
| 中ボタン | コピーします。(C-y) |
| 右ボタンのダブルクリック | 文字カーソルからマウスカーソルまでをリージョンとしてカット。(C-w) |

次に、X 環境の **Mule** は、ウインドウの右側にスクロールバーが付きます。ここにマウスカーソルを合わせて左クリックで下方向にスクロール (C-v)、右クリックで上方向にスクロール (M-v)、中ボタンでバーを掴んで上下スクロールができます。

もう一つ。ウインドウの上側にメニューがついていますね。Buffers,File,Edit,Help の4つです。ここにマウスカーソルを合わせて左クリックしてみるといろいろな項目が現れます。Open File(C-x C-f) 等、普段良く使うキー操作が登録されています。これを利用すればキー操作を覚える必要がなくなります。ワンタッチで日本語チュートリアル(練習)やニュースも立ち上がるので便利です。

他にも色々なモードの時の選択画面などの時にマウスで選択できたりします。

7.9 外国語の入力

Mule で使用できる言語は以下のようなものです¹⁶。

- アジア言語 (2 バイト文字言語): 日本語、中国語、朝鮮語、広東語
- ヨーロッパ言語 (ラテン文字に基づく言語): 英語、オランダ語、エスペラント語、フィンランド語、フランス語、ドイツ語、イタリア語、スカンジナビア語、スペイン語、スイス語、トルコ語、ノルウェー語
- その他の言語: ギリシャ語、ロシア語、ヘブライ語、タイ語、ベトナム語

日本語と中国語は **Wnn** で行ない、その他は **quail システム** で入力します。中国語を入力する時は C-x C-k m で出てくるメニューから PinYin を選択します。その他は **Mule** メニューの File 項目の一番上に **Multi-Lingual** というものがありますのでそれをクリックして下さい。そうすると M-[というキー操作で **quail システム** という外国語入力用のシステムが立ち上がります。続けて、M-s で言語の選択画面になります。何も書かずに <Space> を押せば一覧が現れます。そしてどれかを選択すると、その言語の入力モードになります。この時、M-z で入力の簡易ヘルプが現れます。

各言語の入力についてはここでは触れません。紹介にとどめておきます。必要がある場合は各々が調べて下さい。各言語のサンプルを載せておきます。

プリントアウト

多言語入力によって作られたファイルをプリンタから印刷するには **ct2ps** というコマンドを用います。例えば **nantoka** というファイルを **cspr01** から印刷する時は

```
% ct2ps < nantoka | lpr -Pcspr01
```

とします。

¹⁶ 但し、これらの外国語の入力は X 環境でのみ使用可能です。

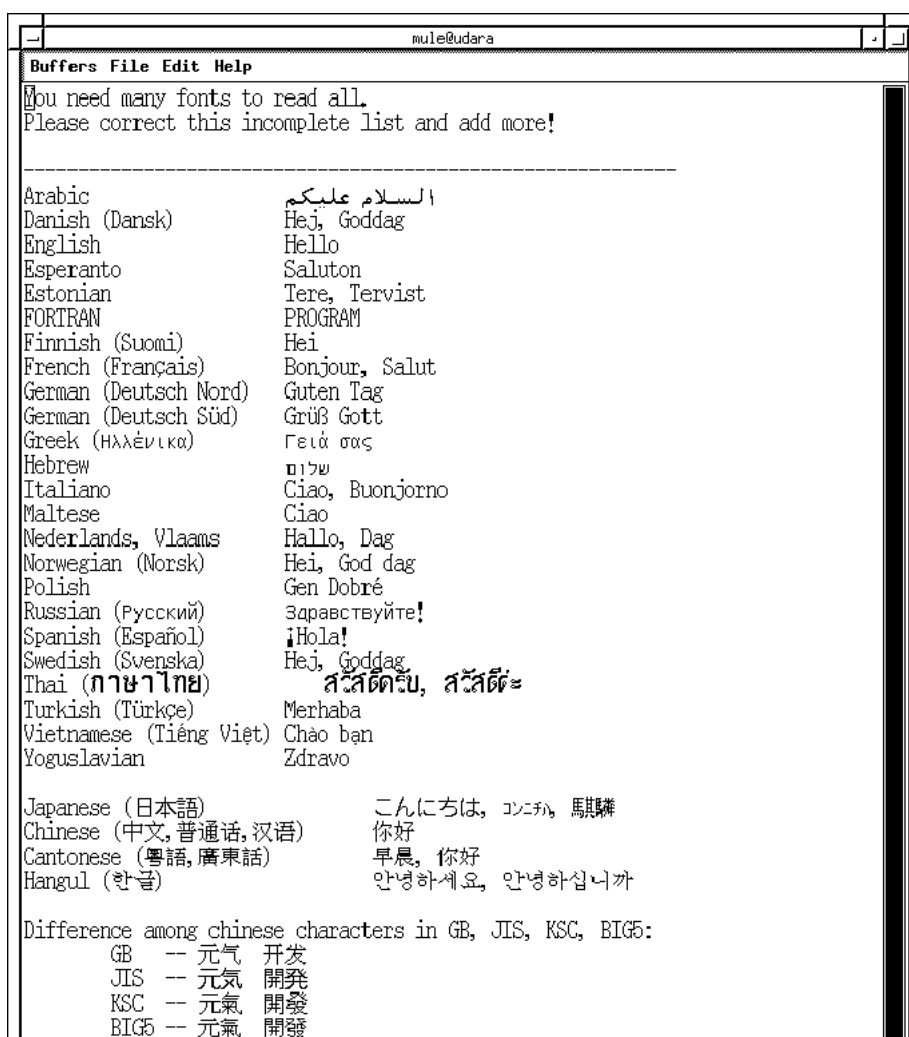


図 7.4 Mule による多言語入力

7.10 とりあえずここまで

Mule は言い出すときがありません。実際、Mule の解説書は何百ページもあります。しかし実際使う機能はここまでで取り上げた機能まで十分という方がほとんどだと思いますので、ここでの解説はこれぐらいで終わらせていただきます。A.3に代表的な機能の一覧を載せておきましたので参照してください。もっと Mule を使い込んでみたいという方は図書館にいろいろ本がありますのでそちらを参照すると良いでしょう。

第 8 章

LaTeX

8.1 LaTeX(らてふ)って何？

8.1.1 LaTeX とは？

今までワープロを使っていると、数学の式で、複雑なものではできませんでした。ところが、LaTeX は、それをきれいな形で出力できます。また、論文や、レポートなどにも手軽に使えます。ただ、ルールがあって、ワープロのようにすぐに使えるわけではありません。また普通のワープロに比べて自由度が大きく、さまざまなことが可能で、絵や図も簡単に文書中に埋め込むことができます。

レポートや卒論をワープロで書こうと思っている諸君、LaTeX を使って書いてみませんか。

8.1.2 LaTeX の特徴

LaTeX の特徴をいくつかあげておきます。

1. 複雑な数式をきれいな形で出力します。したがって、理科系の人にとっては必需品である。(論文など章節のはっきりした構造のある文章を作るのに向いている。)
2. お絵書きツールなどで作った図を簡単にはめ込むことができます。
3. 目次・索引・文献リストを簡単に作ることができる。

8.1.3 LaTeX の作業の進めかた

大まかに言うと次のようにして進めていきます。

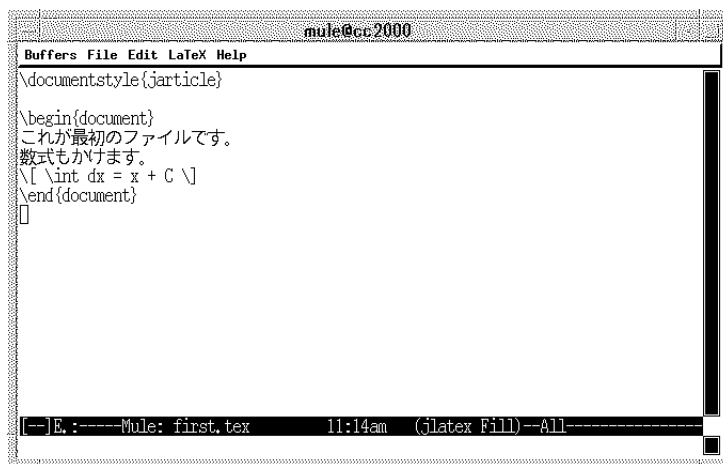
1. エディタ (Mule など) で文書ファイル¹を作成する。
2. 作成したファイルを LaTeX で処理 (コンパイル) する。
3. xdvi コマンドで、コンパイルされたものを見る。
4. 処理したファイルをプリントアウトする。

¹NeXT にある文机や Mac の EG-WORD、solo-writer や一太郎などで書いたものは使えません。

8.2 それでは使ってみましょう

ここでは DEC-3300 で \LaTeX を使う方法を紹介します。簡単な例を示しながら進めていきます。

まず `mule first.tex &<return>` と入力し Mule を立ち上げ、次のようなファイルを作ります。(ここでは DEC-3300、即ち X 環境で Mule を利用する為にコマンドの最後に `&` を付けています。非 X 環境で Mule を利用する場合はこの `&` は必要ありません。このことについては 4.5.2 に詳しく説明してあります。)



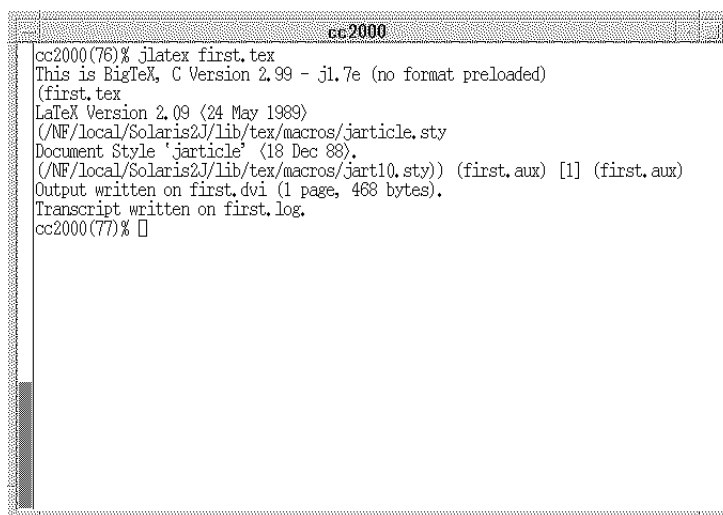
```
mule@cc2000
Buffers File Edit LaTeX Help
\documentstyle{jarticle}

\begin{document}
これが最初のファイルです。
数式もかけます。
\[ \int dx = x + C \]
\end{document}
[]

[-]E.:----Mule: first.tex      11:14am  (jlatex Fill)--411-----
```

図 8.1 first.tex の画面

ファイルが出来たら Mule を `C-x C-c` で終了します。first.tex というファイルがあることを確認して cc2000 のウィンドウの画面で、`jlatex first.tex<return>` と入力してやります。そうすると次のように表示されます。



```
cc2000
cc2000(76)% jlatex first.tex
This is BigTeX, C Version 2.99 - jl.7e (no format preloaded)
(first.tex
LaTeX Version 2.09 (24 May 1989)
(/NF/local/Solaris2J/lib/tex/macros/jarticle.sty
Document Style 'jarticle' (18 Dec 88).
(/NF/local/Solaris2J/lib/tex/macros/jart10.sty)) (first.aux) [1] (first.aux)
Output written on first.dvi (1 page, 468 bytes).
Transcript written on first.log.
cc2000(77)% []
```

図 8.2 うまくいった時の画面

```
cc2000
cc2000(82)% jlatex first.tex
This is BigTeX, C Version 2.99 - jl.7e (no format preloaded)
(first.tex
LaTeX Version 2.09 (24 May 1989)
(/NF/local/Solaris2J/lib/tex/macros/jarticle.sty
Document Style 'jarticle' (18 Dec 88).
(/NF/local/Solaris2J/lib/tex/macros/jart10.sty)) (first.aux)
! Missing $ inserted.
<inserted text>
$
<to be read again>
\intop
\int ->\intop
\nolimits
L.6 [ \int
dx = x + C \]
? x
No pages of output.
Transcript written on first.log.
cc2000(83)% □
```

図 8.3 エラーがあるときの画面

もしエラーがあった場合には、「?」マークが表示されますので、その後に、xを入力すると、コンパイルが中断されます。またどうしても処理が中断できない場合は C-c でプロセスを中断してください。その後、タイプミスの部分を探して直してからもう一度コンパイルの作業をして下さい。詳しいことは、第 8.18 節 (241 ページ) を参照して下さい。

さて、実際にコンパイルされたものがどのようなものに仕上がっているのか確認してみましょう。それには、プロンプトで `xdvi first.dvi &` と打ち込んでみて下さい。そうすると、次のような画面が現れます。

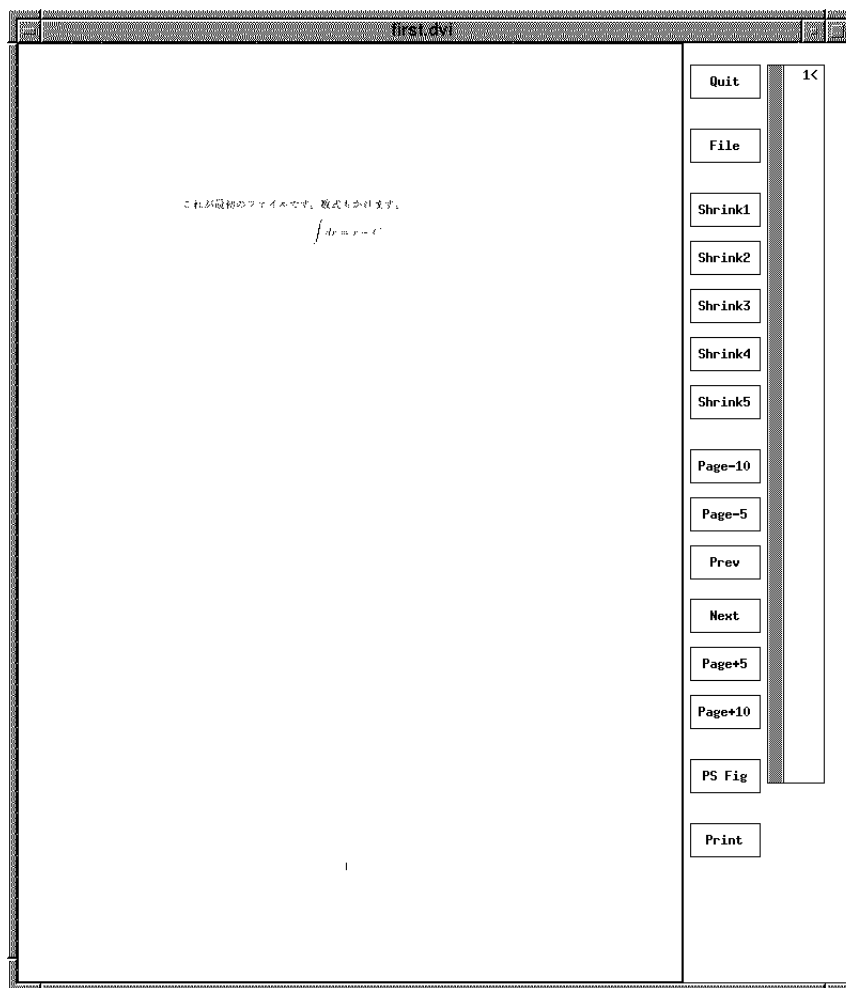


図 8.4 xdvi の画面

これが、コンパイルされて出来たものを表示した画面です。次にプリントアウトするのですが、このままではプリンターから出力できません。そのためプリントアウトできるようにして変換しておかなければなりません。

```
jdvi2kps first.dvi > first.ps<return>
```

と打ち込んでやります。この意味はプリンターが出力できるポストスクリプトという形式に変換する作業です。

```
cc2000
cc2000(79)% jlatex first.tex
This is BigTeX, C Version 2.99 - jl.7e (no format preloaded)
(first.tex
LaTeX Version 2.09 (24 May 1989)
(/NF/local/Solaris2J/lib/tex/macros/jarticle.sty
Document Style 'jarticle' (18 Dec 88).
(/NF/local/Solaris2J/lib/tex/macros/jart10.sty)) (first.aux) [1] (first.aux)
Output written on first.dvi (1 page, 468 bytes).
Transcript written on first.log.
cc2000(80)% jdvi2kps first.dvi > first.ps
[ /NF/local/Solaris2J/lib/tex/jdvi2.ps ] [1]
cc2000(81)% □
```

図 8.5 ポストスクリプトへの変換の画面

ようやくこれでプリントアウトできるようになりました。それでは 21 情報処理教室のプリンターから出力してみましょう。

`lpr -Pcspr01 first.ps<return>`

と打ち込んで下さい。これも作業しているディレクトリの中でどちらのウインドウでも構いません。

21 情報処理教室のプリンター以外から出力する方法に関しては 4.6 を参照して下さい。

これで一通りの \LaTeX を使う作業の手順を紹介しました。

また、dvi ファイルで出来具合を確認して、直接プリントアウトが出来ます。もう一度 `xdvi first.dvi &` と打ってみて下さい。

dvi ファイルが立ち上がりましたね。その画面の右下のところに、print ボタン

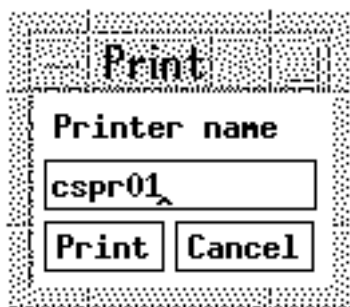


図 8.6 print ボタン

というのがあります。ここで、「print」と書いてあるところをドラッグして、(マウスのボタンを押しながら)「All pages」と書かれているところでマウスをはなすと、すべてのページがプリントアウトされま

す。また、ある部分だけ印刷をしたいときは、第 8.19 節を参照して下さい。ここで注意しておかなければならないのは、「プリント」のボタンを押した時、プリンターの名前が表示されますが、自分がいる情報処理教室のプリンターの名前になっているか確認して下さい。くれぐれもご注意を！

8.3 L^AT_EX におけるルール

例示の方法

これ以降では L^AT_EX の記述を例としてあげる場合に、以下のような記号を用いて表現します。

入力
この記号に続く記述は L ^A T _E X のソースを表しています。
出力
この記号に続く内容が、上記の記述によって出力されます。

8.3.1 最低限のルール

文書ファイルの形式には次のような最低限のルールがあります。

- 文書ファイル名の最後に「.tex」をつけます (例: first.tex)。
- 文書ファイルの最初と最後に半角で次の様に書きましょう。

```
\documentstyle{jarticle}
\begin{document}
:
:
:
\end{document}
```

`\begin{document}` から `\end{document}` までが文書の中身です。

- 使っている環境によって `\` (バックスラッシュ) が使えないときは半角の円記号 `¥` を使います。この 2 つは同じものだと思ってください。(文字コードが同じです。)

8.3.2 ドキュメントスタイルについて

文書ファイルのはじめに書いた

```
\documentstyle{jarticle}
```

というような行は文書のスタイルを指定したものです。

そしてこの中括弧 { } の中は実際のスタイルの指定で次のようなものがあります。

jarticle .. 論文・雑誌の記事・短いレポートに使う最も一般的なスタイルです。

jreport ... 長いレポートなどに使うスタイルです。

jbook 本などに使うスタイルです。見開きで見れるように奇数ページと偶数ページのレイアウトが違います。

これ以外にも、幾つかのスタイルファイルがありますので、第 8.22.2 節を参照して下さい。

オプションについて

括弧[]の中を「オプション」と呼びます。「オプション」は必要がなければ書く必要はありません。

```
\documentstyle[...]{jarticle}
```

以下、ここにはいるオプションをあげておきます。

文字の大きさ .. 文書全体の文字の大きさを指定します²。10pt と 11pt と 12pt のいずれかの指定ができます。10pt の時はここを省略します。

紙の大きさ 出力する紙の大きさを指定します。a4j(A4)、a5j(A5)、b4j(B4)、b5j(B5) といった指定ができます。普通 a4j を指定します。

紙の向き 紙の向きを指定します。縦にして使うときは portrait、横にして使うときは landscape と指定します。しかし縦書きはできません。デフォルトでは portrait になっています。

その他 二段組にするときは jtwocolum と指定します。

上のようなオプションを複数指定してやりたいときは

```
\documentstyle[a4j,12pt]{jarticle}
```

のように半角のコンマ(,)で区切り並べてやります。

8.3.3 見出しの種類

見出しの種類には次のようなものがあります。

種類	コマンド	jarticle	jreport,jbook
部	<code>\part</code>	Part I	Part I
章	<code>\chapter</code>	使えません	Chapter 1
節	<code>\section</code>	1	1.1
小節	<code>\subsection</code>	1.1	1.1.1
小小節	<code>\subsubsection</code>	1.1.1	1.1.1.1
段落	<code>\paragraph</code>	[1.1.1.1]	[1.1.1.1.1]
小段落	<code>\subparagraph</code>	[1.1.1.1.1]	[1.1.1.1.1.1]

スタイルファイルによって、出力される形が違いますので、それらを見ていきます。まず、jarticle の場合は、

┌─── 入力 ───┐

```
\documentstyle[a4j]{jarticle}
\begin{document}
\part{物理学}
\section{現代物理学}
\subsection{量子力学}
\subsubsection{量子力学の発展}
\paragraph{量子力学とは}
\subparagraph{量子力学の歴史}
\end{document}
```

²注意：指定した大きさのフォントがないときはエラーがでます。

—— 出力 ——

Part I

物理学

1 現代物理学

1.1 量子力学

1.1.1 量子力学の発展

量子力学とは

と出力されます。

`jreport` では、

—— 出力 ——

Part I

物理学

Chapter 1

物理学

1.1 現代物理学

1.1.1 量子力学

量子力学の発展

量子力学とは

と出力されます。ただし、**Part I** と **物理学** は、一枚の紙の中央に、そして、改ページをして **Chapter** 以

下のセクションが出力されます³。

最後に、jbook では、一ページ目に、**Part I** と物理学が出力され、2 ページ目は、何も出力されず、3 ページ目に、**Chapter** 以下のセクションが出力されます。

³ここでは、そのようになっていませんが、、、

8.4 いろいろなコマンドと環境

コマンドと環境の説明

このガイドで使われるコマンドはすべて、「\」で始まるもので、タイプライタ体で書かれています。ここでいうタイプライタ体とは、`typewriter style`のように書かれている書体です。一方、**環境**とは、`\begin{...}`と`\end{...}`の対で使うコマンドの一種みたいなものです。**環境**はすべて**太字体**で書かれています。よく現れてきますので、これから説明するコマンドと環境の使い方を是非覚えて下さい。

8.4.1 特殊文字

さて、`\begin{document}`の下に本文を書いていくわけですが、 \LaTeX にはいくつかの「ルール」があります。このガイドでは、この「ルール」の説明と解説を一通りしていきます。

本文で、そのまま使える文字や記号は、アルファベットの小文字と大文字、数字の0~9と、! " ' () = - ' @ + ; * : , . / [] ? の19個の記号だけです。

これ以外の記号で、`# $ % & ~ ^ _ \ { }`の10個の記号は、本文中で単独に使用すると、「ヤバイ」こととなります⁴。

というのも、これらの記号は、それぞれ意味を持っているからです。つまり、

- `#` は、0~9までの数字と組み合わせて、`parameter`として使います。詳しくは、8.20で。
- `$` は、数字やアルファベットを`$`ではさむと、そのはさんだ部分が数式モードになります。詳しいことは、第8.12節のところ。
- `%` は、`%`をつけた後の部分からその行の終わりまでを無視して出力します。つまり、自分がソースを書いている時の目印とか、その行のコメントとかに使います。

```
┌─── 入力 ───┐
What is done cannot be undone.%後悔先に立たず。
└─── 出力 ───┘
What is done cannot be undone.
```

このように、`%`以降の文章が出力されません。

- `&` は、表を作る時、列を区切るのに使います。詳しいことは、第8.10節で。
- `^` は、上つき文字にします。ただし、数式の環境で使います。詳しいことは、第8.12節で。

```
┌─── 入力 ───┐
$ a^{11}+b^{11}=0 $
└─── 出力 ───┘
 $a^{11} + b^{11} = 0$ 
```

- `_` は、下つき文字にします。ただし、これも数式モードで使います。詳しいことは、第8.12節で。

```
┌─── 入力 ───┐
$ a_{11}+b_{11}=0 $
```

⁴思った通りに出力してくれないばかりか、エラーになるかも知れない、ということです。

—— 出力 ——
 $a_{11} + b_{11} = 0$

- ~ は、一個分の空白を作ります。二個つければ 2 個分の空白を作ります。

「—— 入力 ——」
 What is do~ne cannot be undone.
 —— 出力 ——
 What is do ne cannot be undone.

- \ は、バックスラッシュとって、コマンドの前につけます。前節で説明しました。
- { と } は、コマンドの後ろにつけます。前節で説明しました。

したがって、これらを単独で使うことは避けた方が良いでしょう。どうしても使いたいのであれば、以下のようにして下さい。但し、大きさが少し大きめの文字として出力されます。

特殊文字	出力する為の記述	その結果表示される文字
#	\#	#
\$	\\$	\$
%	\%	%
&	\&	&
~	\~{ }	~
^	\^{ }	^
-	_	-
\	\backslash	\
{	\{	{
}	\}	}

~、^、\ については以下のようにして出力することも出来ます。

特殊文字	出力する為の記述	その結果表示される文字
~	\symbol{"7E}	~
^	\symbol{"5E}	^
\	{\tt\symbol{"5C}}	\

これらの記号について上記の方法が気に入らなければ、第 8.4.2 節で紹介する方法で出力して下さい。

また、使ってもエラーを起こしたりする事はないのですが < > | の三つの記号はそれぞれどういうわけか ; ; — などという記号として出力されてしまいます⁵。< > | の記号を出力したい場合はやはり第 8.4.2 節で紹介する方法で出力して下さい。これらの記号については、第 8.4.2 節か、第 8.13 節で紹介する方法で出力して下さい。

⁵但し TypeWriter フォントの時は大丈夫です。これが T_EX の流儀だそうです。なぜこうしたのか想像もつきませんが。

8.4.2 特殊文字でもそのまま出力する

アルファベットの小文字、大文字や数字、! “ ’ @ ‘ () - = [] ; + : * , . ? / の半角文字はそのまま出力されました。これら以外の半角文字は各種のコマンドや意味を持った特殊文字として TeX に解釈されてしまい、そのままでは出力されませんでした。これらの記号を出力させるには `verbatim`⁶ という環境を使います。`verbatim` 環境で書かれた文字はコマンドや特殊な意味を持つとは解釈されません。`verbatim` 環境には二種類の使い方があります。以下にそれぞれ説明します。

記号一文字、もしくは一行におさまる文字列をそのまま出力する

`\verb` に続く記号でそのまま出力したい文字列をはさんでやります。どんな記号でも構いませんが、同じ記号ではさまないとエラーになりますので注意しましょう。以下に「% (パーセント記号)」を出力させる例を示します。

入力 `\verb!%!`

出力 `%`

入力 `\verb+sample サンプル ;:@]/., こんなぐあい+`

出力 `sample サンプル ;:@]/., こんなぐあい`

複数行にわたる文章をそのまま出力する

複数行にわたる文章をそのまま出力したければ、`verbatim` 環境を使います。

—— 入力 ——

```
\begin{verbatim}
```

```
たとえば !@#$$%^&*()<>;: なんかも
```

```
[]{}'"\|-もみーんなこんなもんさ
```

```
\end{verbatim}
```

—— 出力 ——

```
たとえば !@#$$%^&*()<>;: なんかも
```

```
[]{}'"\|-もみーんなこんなもんさ
```

このようになります。

8.4.3 文字の空白

文字による空白には、全角空白と半角空白があります。全角空白は、そのまま、スペースキーで空けます。半角空白は、何個空けても空白は1個分しか空けませんので注意しましょう。文字の空白は、`_`を使用します。`_`は、space key を一回押す、という意味です。そうすると、半角文字一個分だけ空白ができます。それ以上 space key をいれて空白を空けても1個分だけしか空白は空きません。半角空白を何個も出力するのは`_`を繰り返しておきます。

入力 `Time_shock`

⁶ 「ヴァーベイチイム」と読みます。日本語で「文字通り」という意味です。

出力	Time shock
入力	Time\shock(space key を 6 回押しました。)
出力	Time shock
入力	Time\\shock
出力	Time shock

8.4.4 改行と改ページ

\LaTeX で改行をするには、一行改行を入れるか、「`\\`」というコマンドをつける必要があります。一行改行を作ると、 \LaTeX のルールで改行されて少し頭が引っ込みます。つまり、段落ができるということです。一方、「`\\`」は、強制改行のコマンドでただ改行するだけで頭は引っ込みません。

「—— 入力 ——」

改行を一行作ると改行されます。

そして、頭が引っ込みます。「`\verb!\\!`」はただ改行するだけです。`\\` 頭は引っ込みません。`\\` このように改行されました。`\\`

「—— 出力 ——」

改行を一行作ると改行されます。

そして、頭が引っ込みます。「`\\`」はただ改行するだけです。頭は引っ込みません。このように改行されました。

「—————」

改ページをするには、`\newpage` という強制的に改ページするコマンドがあります。

8.4.5 水平方向と垂直方向の空白

まず、水平方向の空白は、`\hspace` があります。これは、引数⁷と一緒に使います。

「—— 入力 ——」

水平方向に 0.5cm 空白 `\hspace{0.5cm}` を入れます。

「—— 出力 ——」

水平方向に 0.5cm 空白 を入れます。

「—————」

このように、「空白」と「を入れます」の間に 0.5cm の空白ができました。

`\hspace` は、右の方をプラスとすると、マイナスの方向つまり、左の方向にも空白を入れることができます。空白を入れるというよりも、空白を「取る」という感じです。

「—— 入力 ——」

水平方向に -0.5cm 空白 `\hspace{-0.5cm}` を入れます。

⁷中括弧の中の数字です。

—— 出力 ——

水平方向に-0.5cm 空白を入れます。

このように、出力されます。よく見ますと、文字が重なってしまいました。これは、「`\hspace`」という言葉が前に 0.5cm 移動したと考えて下さい。文章中に`\hspace` を入れたらそこから水平方向にずれましたが、改行した後で、`\hspace` を使っても水平方向にはずれません。そこで、`\hspace*{...}` というコマンドを使います。たとえば、

—— 入力 ——

```
あいうえお\  
\hspace{2cm}あいうえお\  
\hspace*{2cm}あいうえお\  
あいうえおあいうえおあいうえお\  

```

—— 出力 ——

```
あいうえお  
あいうえお  
あいうえお  
あいうえおあいうえおあいうえお  

```

このようにすれば、行の始めから空白を入れることができます。

次に垂直方向の空白です。コマンドは、`\vspace` です。使い方は、`\hspace` と同じ使い方です。

—— 入力 ——

垂直方向に 3.5cm 空白`\vspace{3.5cm}`を入れます。

—— 出力 ——

垂直方向に 3.5cm 空白を入れます。

このように、3.5cm 垂直方向に空白ができました。

`\vspace` も同様に、マイナスの方向に空白を入れることができます。

—— 入力 ——

垂直方向に-0.5cm 空白`\vspace{-0.5cm}`を入れます。\
こんな感じになります。

—— 出力 ——

垂直方向に-0.5cm 空白を入れます。

御覧のように文字が重なってしまいました。これも「こんな感じになります」という文が、上に 0.5cm 移動したことになります。

長さの単位を以下に挙げておきます。

単位の名前	長さ	幅
cm	センチメートル	+ +
em	大文字の M の横幅	+ +
ex	小文字の m の横幅	+ +
in	インチ (約 25.4mm)	+ +
pc	パイカ (12pt, 約 4.21mm)	+ +
pt	ポイント (0.35mm)	++
mm	ミリメートル	++

出力に出ているのは、その単位で出力される間隔です。⁸

8.4.6 引用

引用を行うものとして、**quote** と **quotation** の二つの環境があります。**quote** 環境は、次のように、短い文章を引用するのに向いています。

—— 入力 ——

```
\begin{quote}
我々は、京都産業大学の学生である。
\end{quote}
```

—— 出力 ——

我々は、京都産業大学の学生である。

一方、**quotation**⁹ 環境は、比較的長い文章や、複数のパラグラフがあるような文章を引用するのに向いています。

—— 入力 ——

```
\begin{quotation}
7月初旬のおそろしく暑い時分のこと、とある夕方近く、一人の若い男が、C-横町の借家人からまた借りていた自分の部屋から街路に出て、なんとなく心のきまらないさまで、のろのろと K-橋の方へ歩いて行った。
\end{quotation}
```

—— 出力 ——

7月初旬のおそろしく暑い時分のこと、とある夕方近く、一人の若い男が、C-横町の借家人からまた借りていた自分の部屋から街路に出て、なんとなく心のきまらないさまで、のろのろと K-橋の方へ歩いて行った。

⁸ この原稿は縮小されて印刷されていますので実際はもう少し広いです。

⁹ 出力例を見ていただいたらわかるように左右に余白ができています。

8.4.7 箇条書き

箇条書きをするには、以下の3つの方法があります。

- **itemize** は、各々の箇条の先頭に・がついた (記号つき箇条書き) 形です。
- **enumerate** は、各々の箇条の先頭に番号がついた形 (番号つき箇条書き) です。
- **description** は、各々の箇条の先頭にラベルをつけた形 (見出しつき箇条書き) です。

箇条書きをするには、これらを環境として使い、それぞれの行の先頭に`\item`を付けます。

記号つき箇条書き

記号つき箇条書きとは、箇条の先頭に記号がついた形で出力します。

———— 入力 ————

```
\begin{itemize}
```

```
\item {\bf itemize}は、各々の箇条の先頭に・がつけた (記号つき箇条書き) 形です。
```

```
\item {\bf enumerate}は、各々の箇条の先頭に番号がついた形 (番号つき箇条書き) です。
```

```
\item {\bf description}は、各々の箇条の先頭にラベルをつけた形 (見出しつき箇条書き) です。
```

```
\end{itemize}
```

———— 出力 ————

- **itemize** は、各々の箇条の先頭に・がつけた (記号つき箇条書き) 形です。
- **enumerate** は、各々の箇条の先頭に番号がついた形 (番号つき箇条書き) です。
- **description** は、各々の箇条の先頭にラベルをつけた形 (見出しつき箇条書き) です。

のようになります。

番号つき箇条書き

これは、箇条の先頭に番号がついた形になります。

———— 入力 ————

```
\begin{enumerate}
```

```
\item {\bf itemize}は、各々の箇条の先頭に・がつけた (記号つき箇条書き) 形です。
```

```
\item {\bf enumerate}は、各々の箇条の先頭に番号がついた形 (番号つき箇条書き) です。
```

```
\item {\bf description}は、各々の箇条の先頭にラベルをつけた形 (見出しつき箇条書き) です。
```

```
\end{enumerate}
```

———— 出力 ————

1. **itemize** は、各々の箇条の先頭に・がつけた (記号つき箇条書き) 形です。
2. **enumerate** は、各々の箇条の先頭に番号がついた形 (番号つき箇条書き) です。
3. **description** は、各々の箇条の先頭にラベルをつけた形 (見出しつき箇条書き) です。

となります。

また、`\item`の後に、`[1a.]`などの番号を入れておきますと、

入力

```
\begin{enumerate}
\item[1a.] {\bf itemize}は、各々の箇条の先頭に・がつけた（記号つき箇条書き）形です。
\item[1b.] {\bf enumerate}は、各々の箇条の先頭に番号がついた形（番号つき箇条書き）です。
\item[1c.] {\bf description}は、各々の箇条の先頭にラベルをつけた形（見出しつき箇条書き）です
\end{enumerate}
```

出力

- 1a. **itemize** は、各々の箇条の先頭に・がつけた（記号つき箇条書き）形です。
- 1b. **enumerate** は、各々の箇条の先頭に番号がついた形（番号つき箇条書き）です。
- 1c. **description** は、各々の箇条の先頭にラベルをつけた形（見出しつき箇条書き）です。

このようになります。

見出しつき箇条書き

この環境は、前の二つと少し違っていて、以下のようにします。

まず、`\item`の後に、`[見出し]`をつけます。この見出しが前の二つでいう「記号」や「数字」に当たります。例で見てみましょう。

入力

```
\begin{description}
\item[itemize] 記号つき箇条書き
\item[enumerate] 番号つき箇条書き
\item[description] 見出しつき箇条書き
\end{description}
```

出力

itemize 記号つき箇条書き
enumerate 番号つき箇条書き
description 見出しつき箇条書き

これらを入れ子状にしても使えます。例えば、**itemize** 環境では、

入力

```

\begin{itemize}
\item 箇条書き
  \begin{itemize}
\item 箇条書き
  \begin{itemize}
\item 箇条書き
  \begin{itemize}
\item 箇条書き
  \end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}

```

— 出力 —

- 箇条書き
 - 箇条書き
 - * 箇条書き
 - 箇条書き

となります。 **enumerate** 環境では、

1. 箇条書き
 - (a) 箇条書き
 - i. 箇条書き
 - A. 箇条書き

となります。このような形で入れ子状の箇条書ができます¹⁰。

しかし、この入れ子状も3回までで限度になります。それ以上やるとエラーになります。注意しましょう。

¹⁰ 箇条書の記号や番号は変えることができます。例えば、数字の番号ではなく、アルファベットの番号にすることもできますが、ここでは説明しません。

8.4.8 右寄せ、中央寄せ、左寄せ

文章を右に寄せたり、中央に寄せたり、左に寄せたりするには、`flushright`、`center`、`flushleft` を環境として使います。

入力	出力
<code>\begin{flushright}</code> 文章を右によせます。 <code>\end{flushright}</code>	文章を右によせます。
<code>\begin{center}</code> 文章を中央に寄せます。 <code>\end{center}</code>	文章を中央に寄せます。
<code>\begin{flushleft}</code> 文章を左に寄せます。 <code>\end{flushleft}</code>	文章を左に寄せます。

という具合になります。

右寄せ、中央寄せ、左寄せは、他に、`\raggedleft`、`\raggedright`、`\centering`でも行えます。但しこれを書いた以降の全ての文に影響を与えますので、どちらかという文書全体の形を決める宣言として考えて下さい。どのような形に整形されるかは、既に出来上がっている複数行を含む文書で試して下さい。

8.4.9 文字の大きさ

基本となる文字のサイズは10ptです。その他に11pt,12ptが指定できましたね。これ以外で、文字の大きさを文中で指定することができます。以下にあげたように10個の文字の大きさがあります。`\tiny`は、ルビサイズです。`\footnotesize`は、脚注の文字の大きさです。`\normalsize`は、普通の文字の大きさです。`\large`からは、文字の大きさが少しずつ大きくなっていきます。(注意)¹¹

入力	出力	入力	出力
<code>\tiny</code>	<code>tiny</code>	<code>\large</code>	<code>large</code>
<code>\scriptsize</code>	<code>scriptsize</code>	<code>\Large</code>	<code>Large</code>
<code>\footnotesize</code>	<code>footnotesize</code>	<code>\LARGE</code>	<code>LARGE</code>
<code>\small</code>	<code>small</code>	<code>\huge</code>	<code>huge</code>
<code>\normalsize</code>	<code>normalsize</code>	<code>\Huge</code>	<code>Huge</code>

ここでひとこと。文中ではあまり文字のサイズを変えることはおすすめできません¹²。

¹¹ここで、印刷されているのは縮小して印刷していますので、若干、文字が小さくなっています。

¹²けっこう、読みにくいものです。注意しましょう。

8.4.10 書体

入力	欧文出力	日本語出力
<code>\rm</code>	roman	普通の欧文書体
<code>\bf</code>	boldface	太字体
<code>\it</code>	<i>italic</i>	斜体
<code>\sl</code>	<i>slanted</i>	傾斜体
<code>\sf</code>	sans serif	サンセリフ体
<code>\sc</code>	SMALL CAPS	大文字体
<code>\tt</code>	Typewriter	タイプライタ体
<code>\em</code>	<i>emphasize</i>	強調体

見て判るように、日本語は太字体にしか変わりません。また、強調体は、斜体と同じ文字となります。

これらの命令はひとたび使いますと、以降の文章をすべて、その書体にしてしまいますので、それを防ぐには、その書体にしたい範囲を{ ... }で囲っておきますと、その括弧の中だけが、コマンドの文体になります。

入力
<code>abcdefg 京都産業大学\\</code>
<code>{\rm abcdefg 京都産業大学}\\</code>
<code>{\tt abcdefg 京都産業大学}\\</code>
<code>{\bf abcdefg 京都産業大学}\\</code>

出力
abcdefg 京都産業大学
abcdefg 京都産業大学
abcdefg 京都産業大学
abcdefg 京都産業大学

文字の大きさと種類を同時に指定する時は、書体を先に指定し、その次に大きさを指定します。

	入力	出力
書体が先で大きさが後	<code>{\huge {\bf unix}}</code>	unix
大きさが先で書体が後	<code>{\bf {\huge unix}}</code>	unix

下の例では、太文字体の指定が無効になってしまいました。

さらに、`{\huge \bf unix}`のように中の括弧を取り除くことが出来ます。

入力	出力
<code>{\huge \bf unix}</code>	unix

8.5 表題

8.5.1 タイトルの作り方

タイトルをつけるには、以下にあげるコマンドを用います。

- `\title{タイトル名}`—ここにタイトル名を書きます。
- `\author{著者名}`—著者が複数の時は、`\and` コマンドを使って、連名にします。
- `\date{日付}`—`\date` が省略された時は、その原稿ファイルが \LaTeX にかけて、結果ファイルが作られた日になります。
- `\maketitle`—タイトルを出力します。

これらは、`\documentstyle{...}`と`\begin{document}`の後ならどこでも構いません。そして、`\maketitle` は、それらの後におきます。後の例を参考にして下さい。

8.5.2 概要の作り方

タイトルの次は、概要の作り方です。これは論文などではよく見かけられるものです。やり方は、

```
\begin{abstract}
  概要の内容
\end{abstract}
```

とします。

┌─── 入力 ───┐

```
\title{ここにタイトルがきます。}
\author{ここが書いた人の名前\and 共同者の名前}
\date{出来上がった日付}
\maketitle
\begin{abstract}
  今までワープロを使っていると、数学の式で、複雑なものはできませんでした。ところが、 $\text{\LaTeX}$  は、それをきれいな形で出力できます。また、論文や、レポートなどにも手軽に使えます。ただ、ルールがあって、ワープロのようにすぐに使えるわけではありません。....
\end{abstract}
```

┌─── 出力 ───┐

ここにタイトルがきます。

ここが書いた人の名前

共同者の名前

出来上がった日付

Abstract

今までワープロを使っていると、数学の式で、複雑なものはできませんでした。ところが、 \LaTeX は、それをきれいな形で出力できます。また、論文や、レポートなどにも手軽に使えます。ただ、ルールがあって、ワープロのようにすぐに使えるわけではありません。....

ここで、注意してほしいのは、ドキュメントスタイルを指定すると、それに合わせて、Part や Chapter が変わるのと同じように、表題もスタイルファイルによって変わります。つまり、上の例は、`jarticle` で書かれたものです。`jreport` と `jbook` では、一枚の紙に、タイトル、著者、日付を出力します。そして、ページをかえて、Abstract を出力し、またページをかえて、本文が始まります。ところが、`\jbook` では、Abstract が使えません。

ドキュメントスタイル	title	author	date	Abstract
<code>jarticle</code>	○	○	○	○
<code>jreport</code>	○	○	○	○
<code>jbook</code>	○	○	○	×

8.6 傍注

傍注として本文の横に「余白だ!」と出力するには、`\marginpar {余白だ!}`というコマンドを使います。余白だ!

傍注が本文の右に来るか、左に来るかは`\documentstyle{...}`で指定した本文のスタイルに依存します。どこに出るかはいろいろ試して見るのがいいでしょう。

`\marginpar[左余白]{右余白}`と書けば、右に余白があった場合にはそこに「右余白」、左に余白があった場合にはそこに「左余白」と出力されます。

8.7 脚注

脚注¹³を出力するには、`\footnote`というコマンドを使います。`\footnote{ここに出る注釈を脚注と呼びます。}`という具合です。

本文中の`\footnote`が現れたところのフットノートナンバーが振られ、括弧の部分がページの下に書かれます。

8.8 相互参照

論文や、レポートといった文章を書いている時、「第?節を参照して下さい」とか、「(6)式に(7)を代入して...」といったことがよくあります。これをそのまま本文に書いていると、文章を加筆したり、削除したりすると番号が違って来る時があります。

¹³ここに出る注釈を脚注と呼びます。

そうなる、いちいち番号をかえていかななくてはなりません。そこで、この番号振りを自動でやってくれるのが、`\label{...}`というコマンド¹⁴です。例えば、

```
\section{...}\label{bun}
```

としておきますと、`{...}`というセクションに「bun」というラベルがついたことになります。

このラベルには何でも構いませんが、`\`、`{}`、`,`、`}`、`...`といった特殊な記号は駄目です。

また、ダブって使われるラベルはエラーのもとになりなすので注意しましょう。ラベルを出力するには、相棒となる`\ref{...}`を使います。また、ページ番号を出力したい時には、`\pageref{bun}`とすると、出力されます。

入力例

```
\subsection{相互参照}\label{bun}
... 第\ref{bun}節参照...
... \pageref{bun}...
```

出力例

```
... 第 8.18.2 節参照...
... 243 ページ参照...
```

(復習)

`\label` コマンドを使った時は、必ず、コンパイルを 2 回以上行なって下さい。というのは、一連のコンパイルの手順は、以下のような形になります。

TEX 原稿の作成

(jlatex コンパイル)⇓ ↑ (error 訂正)

.tex ファイル

.aux ファイル

.log ファイル

.dvi ファイル

.tex ファイルを作って、一回でもコンパイルをすると、この残り 3 つのファイル¹⁵が出来上がります。

- .tex ファイル 原稿のファイルです。
- .aux ファイル は、補助ファイルです。ここに、相互参照のラベルのついたものが書かれます。
- .log ファイル は、エラーメッセージや実行状態が書かれます。ここに、エラーの原因が書いてありますので、もしコンパイルのときエラーを起こして、直したけれども、それでもまだエラーを起こすような時、原因がよくわからなければここを見ると解決することがあります。
- .dvi ファイル 出力結果のファイルです。

相互参照に利用されるファイルは上記の .aux ファイルです。これは 1 回だけでは、ただラベルが列挙されただけで、`\label` と `\ref` の対応関係がついていません。もう一度コンパイルの作業をすることによって、きちんと `\label` と `\ref` の対応関係がつかます。

つまり相互参照を含むドキュメントを修正した場合に、参照している章節の番号やページ位置が保証されるのは内容を修正せずに二回連続でコンパイルした時だけだということです。

¹⁴ラベルは本文中ならオールマイティにつけることができます。

¹⁵これ以外にも L^AT_EX で作られるドットファイルはあります。

8.9 箱

L^AT_EX では、いろいろな箱を作ることができます。枠付きの箱、点線の箱、枠なしの箱があります。

8.9.1 一行に収まる文字列を囲む

枠付きの箱 `fbox` と `framebox`

枠付きの箱を作るコマンドには、

- `\fbox`
- `\framebox`

があります。

`\fbox` は、一行分の箱を作ります。使い方は、

—— 入力 ——
`\fbox{ここに囲みたい文を入れます。}`
—— 出力 ——
ここに囲みたい文を入れます。

となります。この `\fbox` は、一行分の箱しかできません。つまり、2 行以上にわたる文もすべて一行にしてしまいます。例えば、

—— 入力 ——
`\fbox{あいうえおかきくけこさしすせそたちつてと\\ なにぬねのはひふへほまみむめもやゆよらりるれろわを}`
—— 出力 ——
あいうえおかきくけこさしすせそたちつてとなにぬねのはひふへほまみむめもやゆよらりるれろわを

のように、一行の箱になります¹⁶。`\fbox` は、キチキチの箱でしたが、`\framebox` というコマンドを使えばこの枠を自由に変えることができます。

`\framebox[長さ][位置]{囲みたい文}` とします。文の位置は、指定しないと中央になります。位置の指定には他に、

- `[r]` 右に合わせます。
- `[l]` 左に合わせます。

があります。いろいろな長さの箱ができますが、これも一行分の箱しかできません。例えば、

—— 入力 ——
`\framebox[10cm][l]{あいうえおかきくけこさしすせそ}`
—— 出力 ——
あいうえおかきくけこさしすせそ

となります。

¹⁶ 一行に収まらない文は、はみ出しても一行の箱にします。

もっとどかい箱を作るには、`\framebox` を、以下のようにして使います
`\framebox(横の大きさ, 縦の大きさ)[位置]{箱の中に入る文}`
のようにして使います。縦、横の大きさの単位は `pt`(ポイント) を使います。位置は、中央が基本で、

- `[t]` 上の中央に合わせます。
- `[b]` 下の中央に合わせます。
- `[r]` 右の真中に合わせます。
- `[l]` 左の真中に合わせます。
- `[tr]` 右上の角に合わせます。
- `[tl]` 左上の角に合わせます。
- `[br]` 右下の角に合わせます。
- `[bl]` 左下の角に合わせます。

があります。例えば、

```
—— 入力 ——  
\framebox(300,30)[bl]{あいうえおかきくけこさしすせそ}  
—— 出力 ——  
あいうえおかきくけこさしすせそ
```

となります。これも、一行分しか入りません。試してみてください。

点線の箱 `dashbox`

点線の箱を作るには、`\dashbox` というコマンドを使います。使い方は、
`\dashbox{点線の長さ}(横の大きさ, 縦の大きさ)[位置] {箱の中に入る文}`
のようにします。例えば、

```
—— 入力 ——  
\dashbox{1}(300,30)[bl]{あいうえおかきくけこさしすせそ}  
—— 出力 ——  
あいうえおかきくけこさしすせそ
```

となります。

枠なしの箱 `makebox`

今までは枠がありましたが、`\makebox` というコマンドを使えば枠のない箱を作ることもできます。使い方は `\framebox` と同じです。

`\makebox(横の大きさ, 縦の大きさ)[位置]{箱の中に入る文}`
例えば、

```

┌─── 入力 ───┐
\makebox(300,30)[t1]{あいうえおかきくけこさしすせそ}
└─── 出力 ───┘
                あいうえおかきくけこさしすせそ

```

となります。これも、一行分しか入りません。

8.9.2 複数行にわたる文の箱を作る

今までは一行分の箱しかできませんでしたが、複数行の箱を作るには、`\parbox` というコマンドと、`minipage` 環境があります。

`parbox` を使った箱

`\parbox` を使うには、以下のようにします。

```
\parbox[基準線]{横幅の長さ}{文章}
```

とします。とにかく例を見て下さい。●は、わかりやすくするために`\parbox` と `\parbox` を区切っています¹⁷。

```

┌─── 入力 ───┐
\parbox[t]{4.5cm}{あいうえおかきくけこさしすせそ}●
\parbox[t]{4.5cm}{あいうえおかきくけこさしすせそたちつとなにぬねのはひふへほ}●
\parbox[t]{4.5cm}{あいうえおかきくけこさしすせそたちつとなにぬねのはひふへほまみむめもやゆ
よりりるれろわゐゑを}
└─── 出力 ───┘
あいうえおかきくけこさしす●あいうえおかきくけこさしす●あいうえおかきくけこさしす
せそ                せそたちつとなにぬねのは   せそたちつとなにぬねのは
                ひふへほ                        ひふへほまみむめもやゆよら
                りるれろわゐゑを

```

また、

```

┌─── 入力 ───┐
\parbox[b]{4.5cm}{あいうえおかきくけこさしすせそ}●
\parbox[b]{4.5cm}{あいうえおかきくけこさしすせそたちつとなにぬねのはひふへほ}●
\parbox[b]{4.5cm}{あいうえおかきくけこさしすせそたちつとなにぬねのはひふへほまみむめもやゆ
よりりるれろわゐゑを}
└─── 出力 ───┘
                あいうえおかきくけこさしす
                あいうえおかきくけこさしす  せそたちつとなにぬねのは
あいうえおかきくけこさしす  せそたちつとなにぬねのは  ひふへほまみむめもやゆよら
せそ                ●ひふへほ                ●りるれろわゐゑを

```

¹⁷ この●は別のものも、なくても構いません。

となります。例で見ていただいたように、基準線とは、`\parbox` を上で揃えるか[t]、下で揃えるか[b]、ということです。枠の大きさを決めてしまえば、後はその大きさに沿って文章を改行していきます¹⁸。これは枠がありませんでしたが、枠をつけるには、`\fbox` と一緒に使います。つまり、

入力

```
\fbox{\parbox[t]{4.5cm}{あいうえおかきくけこさしすせそ}}●  
\fbox{\parbox[t]{4.5cm}{あいうえおかきくけこさしすせそたちつてとなにぬねのはひふへほ}}●  
\fbox{\parbox[t]{4.5cm}{あいうえおかきくけこさしすせそたちつてとなにぬねのはひふへほまみむめもやゆよらりるれろわゐゑを}}●
```

出力

あいうえおかきくけこさしす せそ	●	あいうえおかきくけこさしす せそたちつてとなにぬねのは ひふへほ	●	あいうえおかきくけこさしす せそたちつてとなにぬねのは ひふへほまみむめもやゆよら りるれろわゐゑを	●
---------------------	---	----------------------------------------	---	-------------------------------------------------------------	---

と枠ができ上がります。

8.9.3 minipage 環境

これは、`\parbox` を発展させたものです。以下のように使います。

```
\begin{minipage}[基準線]{横幅の長さ}
```

.....

```
\end{minipage}
```

入力

```
\begin{minipage}[t]{8cm}  
{\bf minipage}環境は、小さなページのようになります。そして、  
脚注もできます\footnote{footnote もつけることができます。}。\\  
\end{minipage}
```

出力

minipage 環境は、小さなページのようになります。
そして、脚注もできます^a。

^afootnote もつけることができます。

ここで[t] は、基準線を決めるものでした。そして、8cm の小さなページができあがりしました。さらに、このページを枠で囲ってみましょう。

入力

¹⁸改行するというより、文章をつめていく、と言った方がいいかも知れません。

```

\fbbox{
  \begin{minipage}[t]{8cm}
  {\bf minipage}環境は、小さなページのようになります。そして、
  脚注もできます\footnote{footnote もつけることができます。}。
  \end{minipage}
}

```

—— 出力 ——

minipage 環境は、小さなページのようになります。
 そして、脚注もできます^a。

^afootnote もつけることができます。

ここで[t] は、基準線を決めるものでした。そして、8cm の小さなページができあがりました。
minipage 環境をたくさん使うことも出来ます。

—— 入力 ——

```

\fbbox{
  \begin{minipage}{15cm}
  minipage 環境です。\\
  \fbbox{
    \begin{minipage}{10cm}
    minipage 環境です。\\
    \fbbox{
      \begin{minipage}{5cm}
      minipage 環境です。\\
      \fbbox{
        \begin{minipage}{4cm}
        minipage 環境です。\\
        \end{minipage}}
      \end{minipage}}
    \end{minipage}}
  \end{minipage}}
\end{minipage}}

```

—— 出力 ——

minipage 環境です。

minipage 環境です。

minipage 環境です。

minipage 環境です。

このような使い方も出来ます。

8.10 表

表を作るのが、`tabular`¹⁹環境です。表を作る環境は、`tabular`環境の他に、`tabbing`環境がありますが、ここでは、省略させていただきます。

```
\begin{tabular}{列指定}  
表全体  
\end{tabular}
```

このような形で作ります。

列指定とは、表の中にある要素一個一個の配置を指定します。列指定には以下のものがあります。

- l... 右寄せ (left)
- c... 中央 (center)
- r... 左寄せ (right)

そして、列の数だけこれを並べます。

表の要素一個一個を区切るには、「&」を入れます。そして、列の最後には、「\\」をつけます。くれぐれも列指定の数と表の要素の列の数を同じにしてください。

┌─── 入力 ───┐

```
\begin{tabular}{lcr|lcr}  
品名 & 値段 & 数量 & 品名 & 値段 & 数量 \\  
苺 & 200 円 & 2 ケース & 葡萄 & 400 円 & 3 ケース \\  
西瓜 & 1000 円 & 2 玉 & 檸檬 & 100 円 & 10 個 \\  
\end{tabular}
```

─── 出力 ───

品名	値段	数量	品名	値段	数量
苺	200 円	2 ケース	葡萄	400 円	3 ケース
西瓜	1000 円	2 玉	檸檬	100 円	10 個

これに罫線を入れてみましょう。縦の罫線は、列指定のところで、入れたいところに「|」を入れます。横の罫線は、`\hline`です。「\\」の後に置きます。「||」や`\hline\hline`とすれば、2重線になります。

┌─── 入力 ───┐

```
\begin{tabular}{l|c|r||l|c|r} \hline  
品名 & 値段 & 数量 & 品名 & 値段 & 数量 \\ \hline  
苺 & 200 円 & 2 ケース & 葡萄 & 400 円 & 3 ケース \\ \hline  
西瓜 & 1000 円 & 2 玉 & 檸檬 & 100 円 & 10 個 \\ \hline  
\end{tabular}
```

¹⁹ 「タビュラー」と読んで下さい。

出力

品名	値段	数量	品名	値段	数量
苺	200 円	2 ケース	葡萄	400 円	3 ケース
西瓜	1000 円	2 玉	檸檬	100 円	10 個

となります。

もう少し複雑な表を作ってみましょう。

入力

```
\begin{tabular}{|c|c|c|}\hline時刻 & \multicolumn{2}{c|}{平日} \\ \cline{2-3} & 神社発 & 本学発 \\ \hline 10 & 5,10,20 & 0,10,15 \\ & 25,30,45 & 20,35,45 \\ \hline\end{tabular}
```

出力

時刻	平日	
	神社発	本学発
10	5,10,20	0,10,15
	25,30,45	20,35,45

ここで使われている`\multicolumn`というコマンドは、表のある部分だけ、例えばこの表であれば、「平日」という項が、「神社発」と「本学発」の欄にまたがっています。つまり、複数の項目にわたって出力したり、その欄だけ他と違うような形で出力したい時に使います。使い方は、

`\multicolumn{またがる項の数}{列指定}{項目の内容}`

とします。ここでは、`\multicolumn{2}{c|}{平日}`としました。また、`\cline{2-3}`というコマンドは、2項めと3項めにだけ線を引きます。だから、ここの数字をかえれば、部分的に線を引くことができます。

これ以上の詳しい説明はやめにしておきますが、付録の参考文献にあげておきました \LaTeX に関する本を参考して、もっと複雑な表を作ってみてください。

8.11 絵

文章を書いていると、絵や図が必要になることがあります。そんな時役に立つが **figure** 環境です。ここでは、絵や図の作り方は説明しませんが、`xpaint` とか `tgif` などで作ったとします。ただし、図をセーブする時は必ず PostScript 形式で行なって下さい。

さて、注意しなければならないことがあります。それは、`\documentstyle` のところで、「`[epsbox]`」というオプションをつけなくてはなりません。これがないとエラーになります。

では、実際にやってみましょう。以下のようなファイルを作ります。

```
┌─── 入力 ───┐
\documentstyle[a4j,epsbox]{jarticle}
\begin{document}
\begin{figure}[htbp]
  \begin{center}
    \psbox[scale=0.5]{LaTeX-figure_1}\label{fig-1}
  \end{center}
  \caption{蝶}
\end{figure}
└─── 出力 ───┘
```



図 8.7 蝶

このようになりました。この蝶は `xpaint` で書きました²⁰。それでは、説明していきます。

- まず、**figure** 環境を書きます。これは単に絵や図をはめ込むための領域を確保するためのものです。
- オプションとして、`[htbp]` を指定しておきます
- **center** 環境は、単に真中に図を置きたかったから入れました²¹。
- これが重要なコマンド `\psbox` です。このコマンドが絵や図を文章の中にはめてくれます。
- `[scale=0.5]` は、図の倍率の指定です。ここでは、元の大きさの 0.5 倍に指定しています。

²⁰ 変な蝶でごめんなさい。

²¹ 別にあってもなくても構いません。

- `{LaTeX-figure_1}`は、ファイル名です。「.ps」は省略しても構いません。
- `\label` コマンドは、第 8.18.2節を参照して下さい。
- `\caption` コマンドは、絵や図の下に、表題をつけてくれます。別にいらなければ書かなくても結構です。

これらをとにかく呪文のように打って下さい。これで、図を文章の中にはめ込むことが出来ます。

コメント

オプションとして`[hbtpt]`と指定しましたが、これらは次のような意味です。

h here の頭文字で、「ここにはめ込む」という感じです。

b 「そのページの下にはめ込む」という感じです。入らなければ次のページの下にはめ込みます。

t 「そのページの上にはめ込む」という感じです。こちらも入らなければ次のページの上にはめ込みます。

p 独立したページにはめ込みます。つまり、その絵を出力し改ページをします。

それぞれの意味を書きましたが、 \LaTeX は、オプションの順番通りに作業してみて、一番いい状態のものを出力してくれます。しかし、おそらくあなたの希望通りには絵をはめ込んでくれないでしょう。つまり、「ここに絵を入れたい」といっても \LaTeX はやってくれるかどうか分かりません。そこで、`\newpage` コマンドを使って何とかしてみるしかありません。

もう一つ例を出しておきましょう。Mathematica で、グラフを書いて、xv を使って、グラフを切りとります。

┌———— 入力 —————┐

```
\begin{figure}[hbtpt]
  \begin{center}
    \leavevmode
    \psbox[scale=0.5]{LaTeX-figure_2}
  \end{center}
  \caption{トーラス曲線}
\end{figure}
```

————— 出力 —————┐

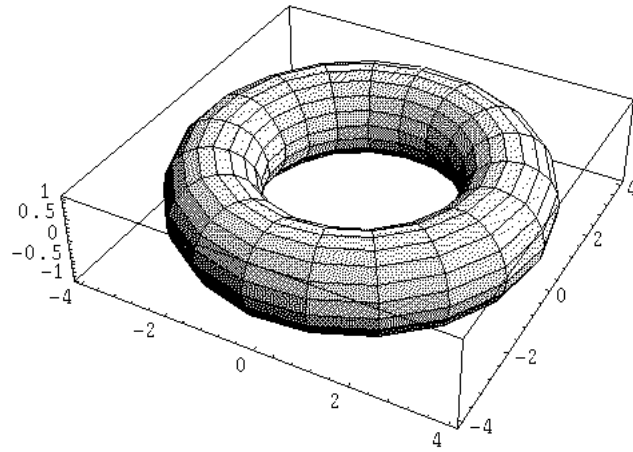


図 8.8 トーラス曲線

このように PostScript 形式のファイルであれば、何でも \LaTeX にはめ込むことができます。

8.12 数式

8.12.1 数式の種類

displaymath 環境と math 環境

数式を使う方法として、数式を単に表示する方法と、文中に数式を入れる二通りの方法があります。**displaymath** 環境は、本文とは別の行に、(次の行が多い) 数式を出力します。これを別の言葉で **displaystyle** といいます。**math** 環境は、文中に数式を出力します。これも **textstyle** と呼びます。しかし、「`displaymath`」と打ち込むのが大変だという人は、`\[...\]` という環境がありますので、こちらを使ってもよいでしょう。後者も`$...$`か、あるいは、`\(...\)` で挟んだ部分が数式として扱われます。

- **displaymath** か、`\[...\]` のどちらか。これがあると、改行して、数式を本文の中央に出力します。
- **math** か、`$...$` か、`\(...\)` のいずれか。本文中に続けて出力します。

本文中の一部分に数式が現れる時

本文中に現れる数式を出力するには、二通りの方法があります。

┌─── 入力 ───┐
`\begin{math}`
 ここで、 $x=r\cos \theta$, $y=r\sin \theta$ とおくと、...
`\end{math}`

─── 出力 ───

ここで、 $x = r \cos \theta$, $y = r \sin \theta$ とおくと、...

└───────────┘

あるいは、

┌─── 入力 ───┐
 ここで、`$x=r\cos \theta` , `$y=r\sin \theta` とおくと、...

─── 出力 ───

ここで、 $x = r \cos \theta$, $y = r \sin \theta$ とおくと、...

└───────────┘

どちらの方法でも同じように出力されました。

数式だけ出力する時

これも二通りのやり方があります。

┌─── 入力 ───┐
`\begin{displaymath}`
 $a(b+c)=ab+ac$
`\end{displaymath}`
 これは分配法則とよばれるもので、...

—— 出力 ——

ここで、以下のことが成り立つとします。

$$a(b+c) = ab + ac$$

これは分配法則とよばれるもので、...

あるいは、

—— 入力 ——

ここで、以下のことが成り立つとします。

```
\[ a(b+c)=ab+ac \]
```

これは分配法則とよばれるもので、...

—— 出力 ——

ここで、以下のことが成り立つとします。

$$a(b+c) = ab + ac$$

これは分配法則とよばれるもので、...

このように出力されます。

equation 環境と eqnarray 環境

displaymath 環境と **math** 環境は、一行の数式で、数式番号がつきませんでした。この数式番号をつける環境が、**equation** 環境と **eqnarray** 環境です。

—— 入力 ——

```
\begin{equation}  
a+b=c  
\end{equation}
```

—— 出力 ——

$$a + b = c \tag{8.1}$$

しかし、**equation** 環境も一行分の数式しか扱えません。

複数の数式を扱うには、**eqnarray** という環境を使います。**eqnarray** 環境を使うと、すべての行に番号が振られてしまいますので、番号を付けないようにするには、**eqnarray***としてやります。一部だけに付けるには、番号を付けない行の終わりに、**\nonumber** をつけます。

—— 入力 ——

```
\begin{eqnarray}  
a+b=c\\  
c+d=e\\  
e=a+b+d\nonumber  
\end{eqnarray}
```

—— 出力 ——

$$a + b = c \quad (8.2)$$

$$c + d = e \quad (8.3)$$

$$e = a + b + d$$

┌—— 入力 ——┐

```
\begin{eqnarray*}
a+b=c\\
c+d=e\\
e=a+b+d
\end{eqnarray*}
```

—— 出力 ——

$$a + b = c$$

$$c + d = e$$

$$e = a + b + d$$

これ以降に挙げる数式の様々な例については、全て数式環境の中で書くこととします。

8.12.2 添字

数式の上付き添字は、「`^`」を用いて、上にしたい添字の部分を`{...}`で囲みます。下付き添字も同様に、「`_`」を下にしたい部分を`{...}`で囲みます。

┌—— 入力 ——┐

```
f(x)=a_{0}x^{n}+a_{1}x^{n-1}+ \cdots +a_{n-1}x+a_{n}
```

—— 出力 ——

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$$

8.12.3 平方根

平方根をするには、

```
\sqrt{...}
```

としてやります。また、`n`乗根は、

```
\sqrt[n]{...}
```

としてやります。

┌—— 入力 ——┐

`\sqrt[5]{(x_1^2+y_1)^3}`

—— 出力 ——

$$\sqrt[5]{(x_1^2 + y_1)^3}$$

┌—— 入力 ——┐

`\sqrt[3]{\sqrt{2}+1}`

—— 出力 ——

$$\sqrt[3]{\sqrt{2} + 1}$$

8.12.4 分数

分数を表すには、
`\frac{分子}{分母}`
としてやります。

┌—— 入力 ——┐

`y=\frac{ax+b}{cx+d}`

—— 出力 ——

$$y = \frac{ax + b}{cx + d}$$

分数表示は、文中でも使えますが、 $y = \frac{ax+b}{cx+d}$ のように、小さく出力され、細かい字があると見えにくくなります。そういう場合は、 $y = (ax + b)/(cx + d)$ と普通の「/」を使えば、見栄えが良くなります。どうしても $y = \frac{ax+b}{cx+d}$ の形で見やすくするには、

`\displaystyle y=\frac{ax+b}{cx+d}`

と書きます。そうすると、文中でも、 $y = \frac{ax + b}{cx + d}$ のようになります。`\displaystyle` は、`displaymath` 環境のように、別の行に出力する数式を、本文中に同じような形で出力するコマンドです。

連分数も書けます。

┌—— 入力 ——┐

`\frac{c_1}{b_1 + \frac{c_2}{b_2 + \frac{c_3}{b_3}}}`

—— 出力 ——

$$\frac{c_1}{b_1 + \frac{c_2}{b_2 + \frac{c_3}{b_3}}}$$

以下のようにも書けます。

┌—— 入力 ——┐

`a_0 + \frac{b_1}{a_1} \atop +`

`\frac{b_2}{a_2} \atop +`

```
\frac{b_3}{a_3}{\atop +}
\frac{b_4}{a_4}{\atop +}
```

出力

$$a_0 + \frac{b_1}{a_1 + a_2 + a_3 + a_4 +}$$

ここで、`\atop` というコマンドは、`{上\atop 下}` のようにして使います。これは、横棒のない分数を出力するコマンドです。`\frac` のようになります。もし、上または、下に何もついていない場合は、上記の例のように `{}` としてやります。

8.12.5 括弧

`{}` を出力するには `\{ \}` などとします。それ以外の `() []` についてはそのまま書けば出力されます。即ち数式の中で `$(\{ (abc) \})$` と書けば `{(abc)}` と出力されます。但し括弧は対にして使わないとエラーになります。

括弧が全て同じ大きさでは見づらい場合があるかも知れません。大きくするには、括弧の前に、`\big` をつけます。

左括弧には、`\bigl`、右括弧には、`\bigr` とします。`\big` より更に大きくするには、`\Big`、`\bBig`、`\Bigl` をつけます。

入力

```
( \bigl( \Bigl( \biggl( \(\Biggl(
```

出力

```
(((((
```

`\big` などは括弧の大きさを絶対的に指定するものですが、次のように `\left`、`\right` とすると、括弧の大きさがその内容に合わせて相対的に選ばれます。

入力

```
\left \{ \left( \frac{ax+b}{cx+d} \right)+
\left( \frac{ex+f}{gx+h} \right) \right \}
```

出力

$$\left\{ \left(\frac{ax+b}{cx+d} \right) + \left(\frac{ex+f}{gx+h} \right) \right\}$$

基本的には括弧は左右の数が合っていないとエラーになります。もし片方だけの括弧を使いたいなら、以下のようにしてピリオド (`.`) を使います。(ピリオドは出力されません)

入力

```
\left \{ \left( \frac{ax+b}{cx+d} \right)+
\left( \frac{ex+f}{gx+h} \right) \right. .
```

出力

$$\left\{ \left(\frac{ax+b}{cx+d} \right) + \left(\frac{ex+f}{gx+h} \right) \right.$$

8.13 L^AT_EX で扱える記号

L^AT_EX で扱える各種の記号類を挙げます。内容は筆者が L^AT_EX の中身から抽出したものが多く含まれていますので、本来使っているものかどうか保証の限りではありません。

8.13.1 雑記号

入力	出力	入力	出力	入力	出力	入力	出力
<code>\S</code>	§	<code>\P</code>	¶		—		
<code>\lq</code>	‘	<code>\rq</code>	’	<code>\lbrack</code>	[<code>\rbrack</code>]
<code>\slash</code>	/	<code>\copyright</code>	©	<code>\aa</code>	ä	<code>\AA</code>	Å
<code>\dag</code>	†	<code>\ddag</code>	‡	<code>\TeX</code>	T _E X	<code>\LaTeX</code>	L ^A T _E X
<code>\dots</code>	...	-	-	--	-	---	—

独立した記号ではありませんが、望みの文字に下線を引くことも出来ます。

入力	<code>\underbar{a}</code>
出力	<u>a</u>
入力	ここで一発 <code>\underbar{下線}</code> が欲しいなあ
出力	ここで一発 <u>下線</u> が欲しいなあ

8.13.2 空白を空ける文字

ここに挙げるコマンドはそれぞれ一定の大きさの空白を、幅ならば右方向に、高さならば下方向に作ります。但し大きさが負の場合、幅ならば左方向に、高さならば上方向に空白が出来ることになります。長さの単位については 第 8.4.5 節を参照して下さい。

入力	意味	入力	意味	入力	意味
<code>\space</code>	空白一文字	<code>\empty</code>	何もなし	<code>\null</code>	幅ゼロの箱
<code>\thinspace</code>	1/6em の幅	<code>\negthinspace</code>	-1/6em の幅	<code>\enspace</code>	1/2em の幅
<code>\enskip</code>	1/2em の幅	<code>\quad</code>	1em の幅	<code>\qqquad</code>	2em の幅
<code>\,</code>	3mu の幅	<code>\></code>	4mu の幅	<code>\<</code>	5mu の幅
<code>\!</code>	-3mu の幅				
<code>\smallskip</code>	3pt の高さ	<code>\medskip</code>	6pt の高さ	<code>\bigskip</code>	12pt の高さ

`\>`と`\<`については`$`で囲まれていることから判るように数式中で使うべきものですが、入れる場所がなかったのここに採録しました。本来は数式中で微妙な隙間を作る為に、もともとあった`\,`に追加されたような気がします。なお、これらが使っている mu という長さの単位についてはその定義が判りませんでした。1mu で 1 ミリあるかないかの短い長さです。実際に試してみてください。

8.13.3 アクセントなど

入力	出力	入力	出力	入力	出力	入力	出力
<code>\'a</code>	à	<code>\'A</code>	á	<code>\v{a}</code>	ă	<code>\u{a}</code>	ǎ
<code>\={a}</code>	â	<code>\^{a}</code>	â	<code>\.a</code>	à	<code>\H{a}</code>	á
<code>\~{a}</code>	ã	<code>\"a</code>	ä	<code>\d{a}</code>	ą	<code>\c{c}</code>	ç
<code>\b{aa}</code>	aa	<code>\t{aa}</code>	aa				

例)

入力	Poincar\`e
----	------------

出力	Poincaré
----	----------

(筆者に言語学的知識がありませんので、おかしい例示があるかも知れませんが勘弁して下さい。)

\`を除いては{}は不要です。つまり、\`_a_でàになります。また、çは、c専用の記号です。

8.13.4 ヨーロッパ系言語特有の記号

入力	出力	入力	出力	入力	出力	入力	出力
<	ı	>	ı̇	\l	ł	\L	Ł
\ss	ß	\oe	œ	\ae	æ	\o	ø
\AE	Æ	\OE	Œ	\O	Ø		

(筆者に言語学的知識がありませんので、おかしい例示があるかも知れませんが勘弁して下さい。)

8.14 数式環境で使える記号

\LaTeX の数式で扱える各種の記号類を挙げます。内容は筆者が \LaTeX の中身から抽出したものが多く含まれていますので、本来使っているものかどうか保証の限りではありません。

8.14.1 雑記号

入力	出力	入力	出力	入力	出力	入力	出力
<code>\hbar</code>	\hbar	<code>\surd</code>	\surd	<code>\angle</code>	\angle	<code>\bowtie</code>	\bowtie
<code>\neq</code>	\neq	<code>\ne</code>	\neq	<code>\models</code>	\models	<code>\colon</code>	$:$
<code>\lhook</code>	\lhook	<code>\rhook</code>	ρok	<code>\lodotp</code>	\cdot	<code>\cdot</code>	\cdot
<code>\ldots</code>	\dots	<code>\cdots</code>	\dots	<code>\vdots</code>	\vdots	<code>\ddots</code>	\ddots
<code>\Vert</code>	$\ $	<code>\vert</code>	$ $	<code>\backslash</code>	\backslash	<code>\doteq</code>	\doteq
<code>\rangle</code>	\rangle	<code>\langle</code>	\langle	<code>\rbrace</code>	$\}$	<code>\lbrace</code>	$\{$
<code>\rceil</code>	\rceil	<code>\lceil</code>	\lceil	<code>\rfloor</code>	\rfloor	<code>\lfloor</code>	\lfloor
<code>\aleph</code>	\aleph	<code>\imath</code>	i	<code>\jmath</code>	j	<code>\ell</code>	ℓ
<code>\wp</code>	\wp	<code>\Re</code>	\Re	<code>\Im</code>	\Im	<code>\partial</code>	∂
<code>\infty</code>	∞	<code>\prime</code>	$'$	<code>\emptyset</code>	\emptyset	<code>\nabla</code>	∇
<code>\top</code>	\top	<code>\bot</code>	\perp	<code>\triangle</code>	\triangle	<code>\neg</code>	\neg
<code>\forall</code>	\forall	<code>\exists</code>	\exists	<code>\flat</code>	\flat	<code>\natural</code>	\natural
<code>\sharp</code>	\sharp	<code>\dagger</code>	\dagger	<code>\ddagger</code>	\ddagger	<code>\smallint</code>	\int
<code>\clubsuit</code>	\clubsuit	<code>\diamondsuit</code>	\diamondsuit	<code>\heartsuit</code>	\heartsuit	<code>\spadesuit</code>	\spadesuit
<code>\triangleleft</code>	\triangleleft	<code>\triangleright</code>	\triangleright	<code>\wedge</code>	\wedge	<code>\vee</code>	\vee
<code>\bigtriangleup</code>	\bigtriangleup	<code>\bigtriangledown</code>	\bigtriangledown	<code>\cap</code>	\cap	<code>\cup</code>	\cup
<code>\sqcap</code>	\sqcap	<code>\sqcup</code>	\sqcup	<code>\uplus</code>	\uplus	<code>\amalg</code>	\amalg
<code>\diamond</code>	\diamond	<code>\bullet</code>	\bullet	<code>\wr</code>	\wr	<code>\setminus</code>	\setminus
<code>\div</code>	\div	<code>\times</code>	\times	<code>\mp</code>	\mp	<code>\pm</code>	\pm
<code>\odot</code>	\odot	<code>\oslash</code>	\oslash	<code>\otimes</code>	\otimes	<code>\ominus</code>	\ominus
<code>\oplus</code>	\oplus	<code>\circ</code>	\circ	<code>\bigcirc</code>	\bigcirc	<code>\cdot</code>	\cdot
<code>\propto</code>	\propto	<code>\ast</code>	$*$	<code>\star</code>	\star	<code>\Box</code>	\square
<code>\sqsubseteq</code>	\sqsubseteq	<code>\sqsupseteq</code>	\sqsupseteq	<code>\parallel</code>	\parallel	<code>\mid</code>	$ $
<code>\dashv</code>	\dashv	<code>\vdash</code>	\vdash	<code>\leq</code>	\leq	<code>\geq</code>	\geq
<code>\succ</code>	\succ	<code>\prec</code>	\prec	<code>\succeq</code>	\succeq	<code>\preceq</code>	\preceq
<code>\supset</code>	\supset	<code>\subset</code>	\subset	<code>\supseteq</code>	\supseteq	<code>\subseteq</code>	\subseteq
<code>\in</code>	\in	<code>\ni</code>	\ni	<code>\gg</code>	\gg	<code>\ll</code>	\ll
<code>\approx</code>	\approx	<code>\sim</code>	\sim	<code>\simeq</code>	\simeq	<code>\perp</code>	\perp
<code>\equiv</code>	\equiv	<code>\asymp</code>	\asymp	<code>\smile</code>	\smile	<code>\frown</code>	\frown

独立した記号ではありませんが、等号などに斜線を入れて否定型を作ることできます。

入力 `a\not\in X`

出力 $a \notin X$

入力 `A\not\supset B`

出力 $A \not\supset B$

少し大きめの記号です。

入力	出力	入力	出力	入力	出力	入力	出力
<code>\coprod</code>	\coprod	<code>\intop</code>	\int	<code>\prod</code>	\prod	<code>\sum</code>	\sum
<code>\bigvee</code>	\bigvee	<code>\bigwedge</code>	\bigwedge	<code>\bigcap</code>	\bigcap	<code>\bigcup</code>	\bigcup
<code>\bigotimes</code>	\bigotimes	<code>\bigoplus</code>	\bigoplus	<code>\bigodot</code>	\bigodot	<code>\ointop</code>	\oint
<code>\bigsqcup</code>	\bigsqcup						

矢印です。

入力	出力	入力	出力
<code>\Longrightarrow</code>	\Longrightarrow	<code>\Longleftarrow</code>	\Longleftarrow
<code>\Rightarrow</code>	\Rightarrow	<code>\Leftarrow</code>	\Leftarrow
<code>\longrightarrow</code>	\longrightarrow	<code>\longleftarrow</code>	\longleftarrow
<code>\rightarrow</code>	\rightarrow	<code>\leftarrow</code>	\leftarrow
<code>\longmapsto</code>	\longmapsto		
<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\longleftrightarrow</code>	\longleftrightarrow
<code>\iff</code>	\iff		
<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\leftrightharpoonup</code>	\leftrightharpoonup
<code>\rightharpoonup</code>	\rightharpoonup	<code>\leftharpoonup</code>	\leftharpoonup
<code>\rightharpoondown</code>	\rightharpoondown	<code>\leftharpoondown</code>	\leftharpoondown
<code>\rightleftharpoons</code>	\rightleftharpoons		
<code>\Uparrow</code>	\Uparrow	<code>\Downarrow</code>	\Downarrow
<code>\uparrow</code>	\uparrow	<code>\downarrow</code>	\downarrow
<code>\Updownarrow</code>	\Updownarrow	<code>\updownarrow</code>	\updownarrow
<code>\nearrow</code>	\nearrow	<code>\searrow</code>	\searrow
<code>\nwarrow</code>	\nwarrow	<code>\swarrow</code>	\swarrow

続いて単独の記号ではなく、文字の上に付ける記号です。付ける元の文字をコマンドの後ろに{ }で囲んで与えます。

入力	出力	入力	出力	入力	出力	入力	出力
<code>\acute{a}</code>	\acute{a}	<code>\grave{a}</code>	\grave{a}	<code>\ddot{a}</code>	\ddot{a}	<code>\tilde{a}</code>	\tilde{a}
<code>\bar{a}</code>	\bar{a}	<code>\breve{a}</code>	\breve{a}	<code>\check{a}</code>	\check{a}	<code>\hat{a}</code>	\hat{a}
<code>\vec{a}</code>	\vec{a}	<code>\dot{a}</code>	\dot{a}				

再び単独の記号ではなく、文字の上に付ける記号です。今度は付ける元の文字の長さに合わせて記号の長さも変化します。同じく付ける元の文字をコマンドの後ろに{ }で囲んで与えます。

入力	出力	入力	出力
<code>\underline{abc}</code>	\underline{abc}	<code>\overline{abc}</code>	\overline{abc}
<code>\widetilde{abc}</code>	\widetilde{abc}	<code>\wdehat{abc}</code>	\widehat{abc}
<code>\overrightarrow{abc}</code>	\overrightarrow{abc}	<code>\overleftarrow{abc}</code>	\overleftarrow{abc}
<code>\overbrace{abc}</code>	\overbrace{abc}	<code>\undervrace{abc}</code>	\underbrace{abc}
<code>\sqrt{abc}</code>	\sqrt{abc}	<code>\sqrt[3]{abc}</code>	$\sqrt[3]{abc}$

何と言えは良いのでしょうか？とにかく見て下さい。

入力	出力
<code>\{abc\}choose{def}</code>	$\binom{abc}{def}$
<code>\{abc\}brack{def}</code>	$\left[\begin{matrix} abc \\ def \end{matrix} \right]$
<code>\{abc\}brace{def}</code>	$\left\{ \begin{matrix} abc \\ def \end{matrix} \right\}$

8.14.2 ギリシャ文字

ギリシャ文字の一覧表です。²²

ギリシャ文字				
文字の名前	大文字		小文字	
	出力	入力	出力	入力
alpha	A		α	<code>\alpha</code>
beta	B		β	<code>\beta</code>
gamma	Γ	<code>\Gamma</code>	γ	<code>\gamma</code>
delta	Δ	<code>\Delta</code>	δ	<code>\delta</code>
epsilon	E		ϵ, ε	<code>\epsilon, \varepsilon</code>
zeta	Z		ζ	<code>\zeta</code>
eta	H		η	<code>\eta</code>
theta	Θ	<code>\Theta</code>	θ, ϑ	<code>\theta, \vartheta</code>
iota	I		ι	<code>\iota</code>
kappa	K		κ	<code>\kappa</code>
lambda	Λ	<code>\Lambda</code>	λ	<code>\lambda</code>
mu	M		μ	<code>\mu</code>
nu	N		ν	<code>\nu</code>
xi	Ξ	<code>\Xi</code>	ξ	<code>\xi</code>
omicron	O		o	o(英語のoと同じ)
pi	Π	<code>\pi</code>	π, ϖ	<code>\pi, \varpi</code>
rho	P		ρ, ϱ	<code>\rho, \varrho</code>
sigma	Σ	<code>\Sigma</code>	σ, ς	<code>\sigma, \varsigma</code>
tau	T		τ	<code>\tau</code>
upsilon	Υ	<code>\Upsilon</code>	υ	<code>\upsilon</code>
phi	Φ	<code>\Phi</code>	ϕ, φ	<code>\phi, \varphi</code>
chi	X	<code>\Chi</code>	χ	<code>\chi</code>
psi	Ψ	<code>\Psi</code>	ψ	<code>\psi</code>
omega	Ω	<code>\Omega</code>	ω	<code>\omega</code>

²² 大文字で抜けているのは英語の大文字と同じだからです。

8.14.3 関数

入力	出力	入力	出力	入力	出力	入力	出力
<code>\log</code>	log	<code>\lg</code>	lg	<code>\ln</code>	ln		
<code>\lim</code>	lim	<code>\limsup</code>	lim sup	<code>\liminf</code>	lim inf		
<code>\sin</code>	sin	<code>\arcsin</code>	arcsin	<code>\sinh</code>	sinh		
<code>\cos</code>	cos	<code>\arccos</code>	arccos	<code>\cosh</code>	cosh		
<code>\tan</code>	tan	<code>\arctan</code>	arctan	<code>\tanh</code>	tanh		
<code>\sec</code>	sec	<code>\csc</code>	csc	<code>\max</code>	max	<code>\min</code>	min
<code>\sup</code>	sup	<code>\inf</code>	inf	<code>\arg</code>	arg	<code>\ker</code>	ker
<code>\dim</code>	dim	<code>\hom</code>	hom	<code>\det</code>	det	<code>\exp</code>	exp
<code>\Pr</code>	Pr	<code>\gcd</code>	gcd	<code>\deg</code>	deg	<code>\bmod</code>	mod

最後に挙げた mod に関してはもう一つ以下のような形が用意されています。

入力	<code>\pmod{n}</code>
出力	$(\text{mod } n)$

8.15 数学のテクニック

8.15.1 行列を作る。

行列を作るには、数式モードで、**array** 環境を使わなければなりません。**array** 環境は、**tabular** 環境と同じ使い方をします。

┌─── 入力 ───┐

```
\[ A =
\left( \begin{array}{ccc}
6 & -3 & -7 \\
-1 & 2 & 1 \\
5 & -3 & -6
\end{array} \right) \]
```

─── 出力 ───

$$A = \begin{pmatrix} 6 & -3 & -7 \\ -1 & 2 & 1 \\ 5 & -3 & -6 \end{pmatrix}$$

8.15.2 = の位置を揃える方法。

「=」の位置を揃えるには、**eqnarray** 環境で、揃えたい等号を、「&」で挟んでおけばよいでしょう。例えば、

┌─── 入力 ───┐

```
\begin{eqnarray*}
x &=& (y+z)^2 \\
&=& y^2+2yz+z^2
\end{eqnarray*}
```

─── 出力 ───

$$\begin{aligned} x &= (y+z)^2 \\ &= y^2 + 2yz + z^2 \end{aligned}$$

となります。

8.15.3 数式モードでの書体

数式モードでしか使えない大文字の筆記体というのがあります。

┌─── 入力 ───┐

`\cal ABCDEFGHIJKLMNOPQRSTUVWXYZ`

—— 出力 ——

ABCDEFGHIJKLMNOPQRSTUVWXYZ

8.15.4 新しくコマンドを作る

新しくコマンドを作るには、`\newcommand` というコマンドを使います。このコマンドは、 $\text{T}_{\text{E}}\text{X}$ がもともと持っているコマンド以外であれば、どんなものでも作ることができます。例えば、物理でベクトルを表すのに太字体の記号を用います。太字体にするには `\boldmath D` としてやればいいのですが、たくさんある時などはいちいち打つのは大変ですから、以下のようにして新しく太字体の記号をつくっておきます。

```
\newcommand{\bmD}{\mbox{\boldmath $D$}}
```

とすることによって新しくコマンドを作ることができます。では、説明していきます。

`\newcommand` で、新しいコマンドをつくる事を宣言します。`\newcommand{\bmD}` で、新しいコマンド `\bmD` をつくります。その新しいコマンドの内容として、`{\mbox{\boldmath D}}` としておきますと、数式モードで文字を太字体にするというコマンドができるのです。

—— 入力 ——

```
\newcommand{\bmD}{\mbox{\boldmath $D$}}
```

```
\[ \bmD=D_1+D_2+D_3 \]
```

—— 出力 ——

$$D = D_1 + D_2 + D_3$$

8.15.5 コマンドの変更

ところで、数式モードでのアルファベットは、斜体で出力されます。一方、演算子記号である `rot` や `div` といったものは、ローマン体で書かれています。しかし、斜体で書かれていると、演算子記号の意味をなさなくなってしまう恐れがあります。そこで、演算子記号を数式モードで、`rot` や `div` といったものをローマン体で出力するには、`\def` というコマンドを使います。`\def` は $\text{T}_{\text{E}}\text{X}$ が持っているコマンドを再定義するコマンドです。`\def` に続くものを定義します。そこで、

```
\def\mathop {\rm rot}\nolimits
```

とすると、数式モードでも、アルファベットをローマン体にすることができます。ここで、`\mathop` というコマンドは、数式モードで使うすべてのコマンドや環境を指します。`\nolimits` というコマンドは、`\lim` や `\sum` などは、上や下に記号が付きませんが、そうではなくて、その記号の上や下に何も記号がつかない、あるいは、つかないという意味です。したがって、`\mathop {\rm rot}\nolimits` は、数式モードで、`$_{\text{rm}} \text{rot}$` とすると、ローマン体の `rot` が出力されます。あとは、`\it` や `\rm` を組み合わせて、いろいろな文字や記号を作ってみましょう。

—— 入力 ——

```

\def\rot{\mathop {\rm rot}\nolimits}
\newcommand{\bmE}{\mbox{\boldmath $E$}}
\newcommand{\bmB}{\mbox{\boldmath $B$}}
\newcommand{\bmr}{\mbox{\boldmath $r$}}
電磁誘導のファラデーの法則\
\[ \rot{\bmE}(\bmr,t)+
\frac{\partial {\bmB}(\bmr,t)}{\partial t}=0 \]

```

—— 出力 ——

電磁誘導のファラデーの法則

$$\operatorname{rot} \mathbf{E}(\mathbf{r}, t) + \frac{\partial \mathbf{B}(\mathbf{r}, t)}{\partial t} = 0$$

8.16 複雑な数式を少々

入力

```

\def\dffrac#1#2{{\displaystyle \frac{#1}{#2}}}
\begin{eqnarray*}
f(x) =
\left \{ \begin{array}{l}
x \sin \dffrac{1}{x} \quad (x \neq 0) \\
0 \quad \quad \quad (x=0)
\end{array} \right.
\end{eqnarray*}

```

出力

$$f(x) = \begin{cases} x \sin \frac{1}{x} & (x \neq 0) \\ 0 & (x = 0) \end{cases}$$

入力

```

\begin{eqnarray*}
I^2 &=& \int\int e^{-r^2} r dr d\theta \\
&=& \int_{-\infty}^{\infty} r e^{-r^2} dr \int_0^{2\pi} d\theta \\
&=& \frac{\pi}{2} \left[ -\frac{1}{2} e^{-r^2} \right]_{-\infty}^{\infty} \\
&=& \frac{\sqrt{\pi}}{2}
\end{eqnarray*}

```

出力

$$I^2 = \int \int e^{-r^2} r dr d\theta$$

$$\begin{aligned}
&= \int_0^\infty r e^{-r^2} dr \int_0^{\pi/2} d\theta \\
&= \frac{\pi}{2} \left[-\frac{1}{2} e^{-r^2} \right]_0^\infty \\
&= \frac{\sqrt{\pi}}{4}
\end{aligned}$$

入力

Schrödinger 方程式

```
\[ \left[ -\frac{\hbar^2}{2m}\nabla^2+V(r) \right] u =Eu \]
```

出力

Schrödinger 方程式

$$\left[-\frac{\hbar^2}{2m}\nabla^2 + V(r) \right] u = Eu$$

8.17 さらに例題

数学でナブラ ∇ の 2 乗 (∇^2) をラプラシアンと呼びます。その記号を、 Δ で表します。

入力

```
\[ \nabla ^{2}=\triangle =\frac{\partial ^{2}}{\partial x^{2}}+\frac{\partial ^{2}}{\partial y^{2}}+\frac{\partial ^{2}}{\partial z^{2}}\]
```

出力

$$\nabla^2 \equiv \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

さらに、演算子記号として、ダランベルジャンというのがあります。これは、4元ベクトルのラプラシアンです。□ という記号で表します。数式で書くと、

入力

```
\[ \Box = \frac{\partial ^{2}}{\partial x^{2}}+\frac{\partial ^{2}}{\partial y^{2}}+\frac{\partial ^{2}}{\partial z^{2}}-\frac{\partial ^{2}}{c^2\partial t^2}\]
```

出力

$$\square = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} - \frac{\partial^2}{c^2 \partial t^2}$$

となります。

相対論では、 t, x, y, z の文字を使わずに、 $x^0 = ct, x^1 = x, x^2 = y, x^3 = z$ を使って座標を表しますので、先ほどの式は、

入力

```
\[ \Box =\frac{\partial ^{2}}{\partial {x^1}^2}+\frac{\partial ^{2}}{\partial {x^2}^2}+\frac{\partial ^{2}}{\partial {x^3}^2}-\frac{\partial ^{2}}{\partial {x^0}^2}\]
```

—— 出力 ——

$$\square = \frac{\partial^2}{(\partial x^1)^2} + \frac{\partial^2}{(\partial x^2)^2} + \frac{\partial^2}{(\partial x^3)^2} - \frac{\partial^2}{(\partial x^0)^2}$$

となります。

線形代数で、「直和」、「直積」を表すのに、 \oplus (`\oplus`) と \otimes (`\otimes`) を用います。

┌—— 入力 ——┐

```
\[ 直和:\; W=W_1 \oplus W_2 \oplus \cdots \oplus W_n\]
\[ 直積:\; W=W_1
\otimes W_2 \otimes \cdots \otimes W_n\]
```

—— 出力 ——

$$\text{直和: } W = W_1 \oplus W_2 \oplus \cdots \oplus W_n$$

$$\text{直積: } W = W_1 \otimes W_2 \otimes \cdots \otimes W_n$$

一般相対性理論では、Christoffel 記号 (Γ) というのがあります。これは、

$$\Gamma_{abc} = \frac{1}{2} (\partial_b g_{ac} + \partial_c g_{ba} - \partial_a g_{cb})$$

と定義される接続係数と呼ばれるものです。本文中、この接続係数を書くのは、めんどい、という時は、 $[_{abc}]$ とか、 $\left\{ \begin{smallmatrix} a \\ bc \end{smallmatrix} \right\}$ と書いて、接続係数を表すことにしています。

┌—— 入力 ——┐

```
\[ \left\{ \begin{smallmatrix} a \\ bc \end{smallmatrix} \right\} =\Gamma_{a b c}=\frac{1}{2}\left(
\partial_{b} g_{ac}+\partial_{c} g_{b a}-\partial_{a} g_{c b}\right)\]
```

—— 出力 ——

$$\left\{ \begin{matrix} a \\ bc \end{matrix} \right\} = \Gamma_{abc} = \frac{1}{2} (\partial_b g_{ac} + \partial_c g_{ba} - \partial_a g_{cb})$$

8.18 エラーの対処

8.18.1 エラーの表示

サンプルファイルとして、abc.tex を作ります。

┌─── 入力 ───┐

```
\documentstyle{jarticle}
\begin{document}
\[ \left\{ \begin{array}{c}
x'=ax+by\\
y'=cx+dy
\end{array} \right. \]
\end{document}
```

このファイルを jlatex にかけて、コンパイルしますと、

┌─── 出力 ───┐

```
jlatex abc.tex
```

```
This is TeX, C Version 2.99 - j1.7 (no format preloaded)
(abc.tex
LaTeX Version 2.09 <24 May 1989>
(/usr/local/lib/tex/macros/jarticle.sty
Document Style 'jarticle' <18 Dec 88>.
(/usr/local/lib/tex/macros/jart10.sty))[1] (abc.aux)
Output written on abc.dvi (1 page, 456 bytes).
Transcript written on abc.log.
```

と表示された時は、正しくコンパイルされたこととなります。

正しい例を出しましたが、どこか間違っている例を出しましょう。

ここからはまず読んで、それから実際にやってみて下さい。

┌─── 入力 ───┐

```
\documentstyle{jarticle}
\begin{document}
\[ \left\{ \begin{array}{c}
x'=ax+by\\
y'=cx+dy
\end{array} \right \]
\end{document}
```

さきほどの abc.tex のところで、6 行目の `\right.` の「`.`」をはずして見ます。そして、コンパイルします

と、

—— 出力 ——

```
jlatex abc.tex

This is TeX, C Version 2.99 - j1.7 (no format preloaded)
(abc.tex
LaTeX Version 2.09 <24 May 1989>
(/usr/local/lib/tex/macros/jarticle.sty
Document Style 'jarticle' <18 Dec 88>.
(/usr/local/lib/tex/macros/jart10.sty)) (abc.aux)
! Missing delimiter (. inserted).
<to be read again>
                \edef
\@latexerr #1#2->\edef
                \@tempc {#2}\expandafter \errhelp \expandafter {\@tem...

\]->\relax \ifmmode \ifinner \@badmath
                \else $$\fi \else \@badmath \fi \igno...
1.6 \end{array} \right \]
```

?

と、プロンプトが入力待ちの状態です。これが間違った時に表示されるエラー表示です。それでは、重要な部分だけを解説していきましょう。

—— 出力 ——

```
jlatex abc.tex

This is TeX, C Version 2.99 - j1.7 (no format preloaded)
(abc.tex
LaTeX Version 2.09 <24 May 1989>
(/usr/local/lib/tex/macros/jarticle.sty
Document Style 'jarticle' <18 Dec 88>.
(/usr/local/lib/tex/macros/jart10.sty)) (abc.aux)
```

1 行目から 8 行目までは、エラーとは関係ありませんので飛ばします。

—— 出力 ——

```
! Missing delimiter (. inserted).
```

ここからが、エラーを表示しているところです。これは、区切り文字が見つからないという事です。つまり、対になるべきところがないという事です。次に、

—— 出力 ——

```
\@latexerr #1#2->\edef
```

これは、エラーの種類を表しています。つまり、`latexerr` とあるように、コマンドの使い方が間違っている、という事です。そして、その場所が、

—— 出力 ——

```
1.6 \end{array} \right \]
```

6行目のところにあります、という意味です。
最後の行に「？」が表示されて、コンパイルの作業が中断していますので、「x」と打ってみてください。そうすると、

—— 出力 ——

```
? x
```

```
No pages of output.
```

```
Transcript written on abc.log.
```

と表示されてるはずですが。この後は、Mule を立ちあげて、間違っているところを直していきましょう。

8.18.2 ちょっと違うやり方

先ほどは「？」の後に、「x」を打ちましたが、それ以外にも次のようなものがあります。

s, または **r** これは、エラーがあっても構わず最後までコンパイルをします。従って、いっぱいエラーがあると思われるファイルは使用しない方がいいでしょう。

h これは、英語で何が間違っているかを教えてくれます。

みなさん、それぞれ試してみてください。

8.18.3 エラーの種類

ここではいくつかのエラーの表示を見てください。

- `! Missing $ inserted.`
 1. 数式環境でエラーをしています。
- `! Undefined control sequence.`
 1. 定義されていないコマンドを使ったか、
 2. コマンドの後に空白がない、
 3. 余計なところで「\」がついてしまったためにコマンドとしてみなしてしまっている。
- `! \begin{eqnarray} ended by \end{document}.`
 1. `\begin`、`\end`の対応関係がおかしい。つまり、どちらかが多かったり、ぬけている。
- `! Missing } inserted.`
- `! Missing { inserted.`
 1. 括弧の対応が正しくありません。
- `Runaway argument?`
 1. `\section`において、閉じ括弧を忘れている。
 2. `\verb`で、はさんでいる記号が違う。など、コマンドの使い方が間違っている。
- `! Extra alignment tab has been changed to \cr.`
 1. 表を宣言した時に、指定した列の数よりソースにある列の数の方が多い。
 2. 列の最後に改行マーク「`\`」が抜けている。
- `! Environment ?????? undefined.`
 1. 環境の命令の綴りが違っている。

などなど、エラーはたくさんありますので列挙するのはやめます。とにかく、エラーを起こしている行番号を頼りに直して行って下さい。自分の勘違いやタイプミスがほとんどです。あきらめずに一つずつ直していきましょう。

もし、エラーの対処についてわからなければ、 \LaTeX の達人に作っている自分のファイルとそのファイル名の拡張子が`.log`のファイルの両方を見てもらいましょう。

8.19 部分印刷する方法

さて、完成した文章に少し手を加えて書き直した時など、最初から印刷するのは、時間と紙がもったいないと思います。そこで、`jdvi2kps` コマンドは次のように使います。

```
cc2000(3)% jdvi2kps -f 開始ページ -t 終了ページ foo.dvi > foo.ps
```

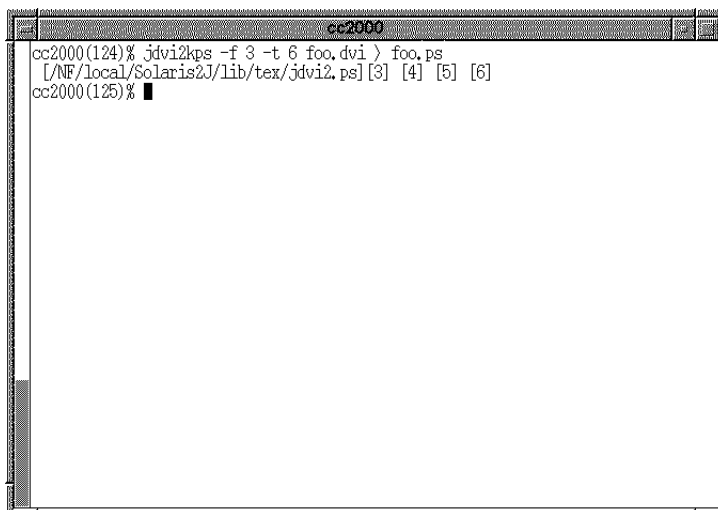


図 8.9 dvi ファイルから部分だけ取り出す方法

としますと、ある部分からある部分までの PostScript ファイル `foo.ps` が出来上がります。

ここで、PostScript ファイルを確認するのは、プロンプトで、
`\ghostview foo.ps` と打ってみて下さい。そうすると、

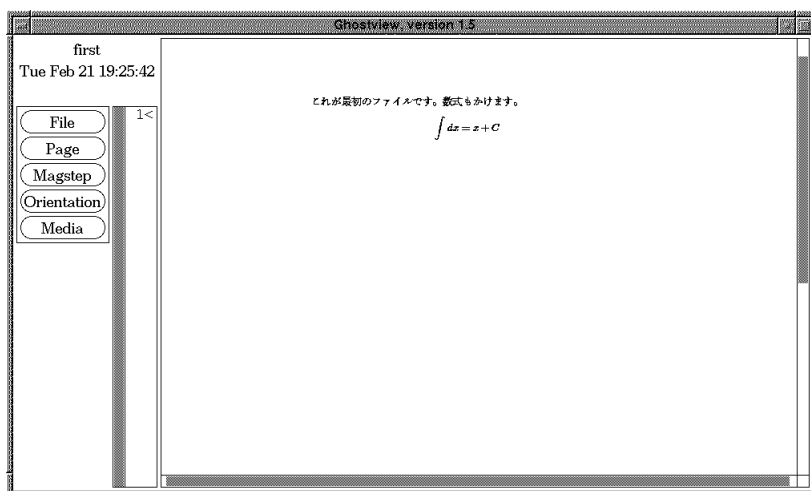


図 8.10 PostScript 形式のファイルの確認

このような画面が現れてきます。これで、PostScript 形式のファイルを見ることが出来ます。そして、
`lpr -Pcspr01 foo.ps`
とすれば、プリントアウトされます。
さらに、dvi 画面のプリントボタン



図 8.11 プリントボタン

そのところの **Print** のところをドラッグして、「Current page」のところでもうすボタンを離すと、そのページだけを印刷することができます。

8.20 自分の命令 (マクロ)

\LaTeX では自分だけの命令 (マクロ) を作ることができます。これは既存の命令を組み合わせて定義し作ります。

8.20.1 簡単な命令を作ってみよう

いま用紙の中央に次のように書こうとします。

京都産業大学

これを実現するためには

```
┌─── 入力 ───┐
\begin{center}
京都産業大学
\end{center}
└──────────┘
```

とします。

これと同じ働きをする `\KSU` という命令²³を作ります。そのためには文書ファイルにこの命令を使うより以前のところでつぎのように `\KSU` を定義しておきます。

```
\newcommand{\KSU}{\begin{center}京都産業大学\end{center}}
```

ファイルのこれ以降の場所で `\KSU` と書けば

```
\begin{center}京都産業大学\end{center}
```

と書いたのと同じ働きをします。 `\newcommand` については、第 8.15.5 節を参照して下さい。

8.20.2 引数を持っている命令の作り方

さっきの `\KSU` を「京都産業大学」をセンタリングするだけでなく、ほかの単語もセンタリングできるようにバージョンアップしてみましょう。

次のようにします。

```
\newcommand{\namae}[1]{\begin{center} #1 \end{center}}
```

このようにすればこれ以降、 `\namae{...}` と書けば (... のところに何か語を書くんですよ) 真ん中に単語がレイアウトされます²⁴。ここで `[1]` は引数の数、 `#1` は引数の入る場所です。引数とは、そのマクロに対応する文字列のことです。だから、... が引数に当たります。

8.20.3 マクロの名前の付け方

このときの名前は、 `[\namae]` のような英字も、 `[\名前]` のような漢字も使うことができます²⁵。しかし大文字と小文字は区別されます。

また記号や数字を含む名前も作ることができません。

²³ 何でも構いません。好きなものをつくって下さい。

²⁴ これってただの `center` 環境ですね

²⁵ あまり漢字はお薦めできません

8.21 ファイルを分けて L^AT_EX を使う方法

このガイドのように分量のあるものを作るとき、1つのファイルとして作業しているとエラーも見つけにくく、処理に時間がかかってしまいます。そこで「小さなまとまりごとに文書ファイルを作る」という考え方が出てきます。

L^AT_EX では `\input{...}` を使って実現することができます。

例として、

```
\input{first.tex}
```

とすると、その場所に `first.tex` のファイルの中身を書いたことになります。

これを使って先程の `\KSU` を自分専用のファイルに入れましょう。まずエディタを立ち上げ、`macro.tex` というファイル名のファイルをつかって、その中に `\KSU` の定義を書き込んでおきます。

そしてこの `macro.tex` というファイルを作業しているディレクトリに入れておきます。

そして使いたい文書ファイルの中で次のように `\input{...}` を使ってこのファイルを読み込みます。

```
\input{macro}
```

そうすればその場所に `macro.tex`²⁶ のファイルの内容を書いたことになり、これ以降の場所で `\KSU` のマクロが使えるようになります。

8.22 標準以外のスタイルファイル

普通スタイルファイルは、(j)article,(j)report,(j)book の3つでしたが、以下にあげるような方法で、いろいろなスタイルファイルが使えます。

8.22.1 日本語 L^AT_EX 定番スタイル集の使い方

京都産業大学がライセンス契約しているインプレスの日本語 L^AT_EX 定番スタイル集の使い方を紹介します。

このスタイル集を使うことによって、`jarticle` や `jreport` というスタイルだけでなく英文レター、はがき、時刻表、レポート用紙、アドレス帳、論文などのさまざまなスタイルの文書を作ることができます。

使い方の例として英文レターの場合は次のようにします。

```
┌─── 入力 ───┐
\documentstyle{iletter}
\begin{document}
\begin{letter}
:
:
\end{letter}
end{document}
└──────────┘
```

のようにドキュメントスタイルの指定を変えてやります。またそれだけではなく、そのスタイルで使いやすいような命令、環境(ここでは `\begin{letter}... \end{letter}` のところ)が入っていますので、各情報処理教室の棚やロッカーにあるマニュアルを見て有効に利用して下さい。

²⁶ `.tex` は省略しても構いません。

8.22.2 京都産業大学独自のスタイルファイル

ccreport.sty

これは主に、UNIX ガイドを書く時に、使われたドキュメントスタイルです。ただし、jreport 専用となります。このスタイルを使おうと思ったら、以下のようにする必要があります。

```
┌─── 入力 ───┐
\documentstyle[ccreport]{jreport}
\begin{document}
(内容)
\end{document}
└──────────┘
```

と、オプションのところに ccreport としておけば使えます。そうすると、次のようなスタイルにかわりま

```
┌─── 入力 ───┐
\documentstyle[a4j,ccreport]{jarticle}
\begin{document}
\chapter{ chapter のタイトル}
\section{ section のタイトル}
\subsection{ subsection のタイトル}
\subsubsection{ subsubsection のタイトル}
\paragraph{ paragraph のタイトル}
\subparagraph{ subparagraph のタイトル}
\end{document}
└──────────┘
```

```
─── 出力 ───
第 1 章
```

chapter のタイトル

1.1 section のタイトル

1.1.1 subsection のタイトル

subsection のタイトル

paragraph のタイトル

subparagraph のタイトル

これ以外にも「figure 1.1 崩壊曲線」が「**図 1.1 崩壊曲線**」に変わります。つまり、英語だったのが日本語に変わりました。一度お試しあれ。

8.23 さいごに

8.23.1 \LaTeX から \AUCTEX へ

次の章で、 \AUCTEX が紹介されていますが、これは、もういちいち打つのは面倒になってきて、何とか楽をしたい、という人にはお推めのもので。詳しいことは、第9章で。

さて、ここまで進まれてきた人は、一通りのことが出来るようになっているものと思われ。ここに書いたものだけでも、ある程度のレポートや卒論は書けるはず。さらに、ここを何とかしたいとか、こういう環境は使えないのか、といったことは後ろにあげておきました参考文献を片手に取り組んでいただきたいと思います。このガイドは、最低限の事しか書かれていません。というより、そうせざるを得なかったのです。世にはたくさんの \LaTeX に関する本が出ています。これをそっくり真似ることは著作権に触れてしまうのです。ある本のオリジナリティを真似ることが著作権に違反します。従いまして、 \LaTeX の機能をすべて余すことなくこのガイドで書くことは出来ません。足りないところは自分で補っていただけたら幸いかと存じます。

第 9 章

AUCTEX

AUCTEX は Mule 上で動作し、TEX を書く作業を支援してくれます。この機能を利用することによって、L^AT_EX の文書を編集する作業がずいぶん便利になります。特にエラー修正の効率はかなり上がると思われます。これらの機能は初心者ほど効果があると思いますので、まずいきなり AUCTEX を使って L^AT_EX を試してみると言うのも良いかも知れません。但し Mule に機能を追加して利用していますので、Mule の操作に慣れている必要があります。

9.1 AUCTEX で L^AT_EX 生活が変わる

AUCTEX を使うとコマンドひとつで

```
\documentstyle[a4j]{jarticle}
\begin{document}

\end{document}
```

等を書いてくれます。カーソルはもちろん `\begin` の行と `\end` の行の間に置かれます。もちろん `document` 環境以外の任意の環境もコマンドひとつで書いてくれます。これで `\begin{\end}` や `{}` の非対応のエラーから解放されます。

AUCTEX を使うと Mule の中からコンパイルとプレビューと印刷のコマンドを実行する事が出来るようになります。これで Mule と kterm を行ったり来たりする必要がなくなります。

AUCTEX を使うとエラーメッセージが日本語で表示されます。その上コマンドひとつで次々とテキストのエラー箇所にカーソルが移動します。これで意味が全く理解できなかった TEX のエラーメッセージからのデバッグ作業の苦痛は取り除かれます。

以上は AUCTEX の機能のほんの一部ですが、それらを使うだけでも L^AT_EX の編集作業が地獄から天国に一変する事でしょう。それではまず 9.2 で AUCTEX を起動してから、9.3 以降でこの節で紹介したすばらしいコマンド達の使い方を説明します。これらのコマンドはいわゆる必修項目です。といってもそのために覚える必要のあるコマンドはたったの 6 つ (実質 4 つ) です。その後、9.8 ではその他の便利なコマンドをざーっと紹介します。

9.2 AUCTEX の起動

さて、AUCTEX モードを起動しましょう。cc 環境では AUCTEX は cc2000 で使えるので、拡張子が `.tex` のファイルを cc2000 の Mule で読み込めば自動的に AUCTEX がドキュメントのスタイルを調べて適切

なモード¹が起動するようになっていました。新しいファイルを書き始める時など、その情報が無い場合は default として JLaTeX モードが起動します。

9.3 C-c

まずは AUCT_EX のコマンドは主に C-c で始まることを覚えてください²。新しい T_EX のファイルを開いて最初にこの AUCT_EX のコマンドを入力する時には master file の名前を聞かれます。これは分割編集の際に参照される変数で、詳しくは 9.8 に説明があります。初めて AUCT_EX を使う場合は <return> キーを押しておきましょう。

9.4 C-c C-e

このコマンドは任意の環境 `\begin{}``\end{}` を対話的にセットしてくれます。e は environment の e です。

Environment type: (default document)

と聞かれるので、使いたい環境名を答えます。9.1 では document 環境の例を挙げましたが、その場合には default の document を使います。ここで <return> キーだけを押しさえ default が採用されます。次に

Document style: (default jarticle)

と聞いてきます。ここで style 名を答えます。jreport が良ければそう書いて <return> キーを押します。最後に

Options:

と聞いてきますので A4 の日本語なら a4j、PostScript の図を取り込むなら epsbox もここで書いておきます。他の環境に対してはその環境に必要な項目を聞いてきます。例えば equation 環境なら

Environment type: (default itemize) equation

label: eq:abel

のように対話が進み、table 環境なら

Environment type: (default itemize) table

Float to: htbp

Caption: すごい表

Label: tab:wao!

Center: (y or n) y

Psition:

Format: |c||l||l||

のように対話します。この各項目の意味については 8.10 等を参照して下さい。

9.5 C-c {

C-c { で} がセットで挿入されます。これで中括弧の閉じ忘れのデバッグ作業から解放されます。

¹JLaTeX モード, JTeX モード, JSliTeX モード, TeX モード, LaTeX モード等があります。

²C-c で始まらないコマンドには例えばマクロの補間のコマンド (9.8.4) や C-c C-某 で入力されたコマンドを取り消すような場合に使われるコマンド等があります。

9.6 C-c C-c

このコマンドを使うことで Mule の中からコンパイルとプレビューと印刷のコマンドを実行する事が出来ます。c は command の c です。例えば、

Command: (default jLaTeX)

の様に聞いてきますので、default で良ければそのまま<return>を押します。コンパイルを実行すると

Type 'C-c C-l' to display results of compilation.

と教えてくれるので、コンパイルの様子を見なければその通りに C-c C-l と入力します。また、現在走らせているコンパイルなどを途中で中止させたい場合には C-c C-k と入力します。

コマンドは多数用意されていますが覚える必要はありません。スペースでコマンド一覧の表示と補完を同時に行なってくれます。その中から目的のコマンド名を探して、Command: 行に入力して<return>を押します。また、default の値も AUCTEX が適当に選んで表示してくれますので、単に<return>を押す事が一番多いでしょう。View や Print を選んだ時は続けて実行されるコマンドを確認してきます。View なら

View command: xdvi foo

です。ここで修正することもできます。問題なければそのまま<return>キーを押します。Print の場合はまずプリンターの名前を聞いてきます。

Printer: (default cspr01)

印刷させたいプリンターが cspr01 でない場合はプリンター名を入力します。すると実行されるコマンドを確認してきます。

Print command: jdvi2kps -d 400 foo | lpr -Pcspr01

問題なければそのまま<return>を押しましょう。

9.7 C-c ‘

さあ天国に行きましょう。デバッグコマンドです。このコマンドは覚える必要さえありません。なぜならコンパイル作業でエラーが見つかった場合、AUCTEX は親切なことに“エラーがあったよ。C-c ‘を押してね。”と教えてくれます。次のエラーに進むときも C-c ‘です。ここで’ではなく‘であることに注意しましょう。

9.8 その他の機能

前節までの機能で L^AT_EX 生活環境は格段に向上します。ですから初めて AUCTEX を使おうという人はこの節をざっと目を通すだけにするか、あるいは読み飛ばしてしまっても良いでしょう。

9.8.1 部分的なコンパイル

AUCTEX を使うと部分的なコンパイルができます。複雑な equation 環境や table 環境になるとコンパイルとプレビューを何度も繰り返しながら修正する事になります。そのために毎回ドキュメント全体をコンパイルするのはばかばかしいですね。AUCTEX では適当な領域だけを部分的にコンパイルすることができます。コンパイルしたい領域の先頭で C-<Space> を入力して、その領域の後尾の次の所にカーソルを置いて C-c C-r と入力します。するとコマンド入力待ちになるので、前述の手順 (9.6 参照) でコンパイルします。複雑な equation 環境や table 環境等を部分コンパイルしたいときには、部分コンパイルしたい環境の中にカーソルを置いた状態で C-c . と入力するだけで領域としてその環境が設定されるので、続けて C-c C-r と入力して下さい。他にも、編集している節を領域として設定するためのコマンド C-c * や、

編集しているバッファ全体を部分コンパイルするためのコマンド `C-c C-b` があります。部分コンパイルは `_region_.tex` というファイルに対して行なわれます。部分コンパイルの結果をプレビューするときは、`C-c C-c` で Region View を選択します (9.6参照)。

9.8.2 ドキュメントの分割編集

AUCTEXを使うと一つのドキュメントを複数のファイルに分けて編集しているときの管理が楽になります。分割編集する場合、一般には親ファイルと子ファイル(下の例では `file1.tex`, `file2.tex`, `file3.tex`) を作って親ファイルには

```
\documentstyle[a4j]{jarticle}
\begin{document}
\input{file1}
\input{file2}
\input{file3}
\end{document}
```

のように子ファイルを読み込むコマンドが書かれているのが普通です。さて、実際に AUCTEX で分割編集をするには、子ファイルを新しく開いた時の最初の AUCTEX のコマンド入力時に master file の名前を聞いてくるので親ファイル名(拡張子 `.tex` は付けません。)を入力します。それにより、子ファイルのバッファからでも `C-c C-c` で全体のコンパイルが出来るようになります。編集中の思考錯誤時には、分割しているメリットを生かして領域を指定すること無く `C-c C-b` で子ファイルのバッファだけの部分的コンパイルを行なうとよいでしょう (9.8.1参照)。子ファイルを編集集中に親ファイルを参照したくなった時には `C-c ^` と入力すれば親ファイルのバッファが開かれます。

9.8.3 アウトラインマイナーモード

9.8.2の様にドキュメントを分割して作成する方法を取らずにひとつのファイルに全ての文章を書くスタイルもあります。しかしそうすると、さて第2章第3節を校正しようかという時にそこを画面に表示させるために `C-v` を何回も押して…おととと、行き過ぎたゾ `<ESC> v`, なんてことになりかねません。AUCTEX でアウトラインマイナーモードを使えばそんな面倒を避けられます。まずは `<ESC> x outline-minor-mode` と入力して下さい。そしてまず `C-c C-o C-t` と入力して文章部分(ボディ)を隠して、`\chapter` や `\section` の行(ヘッダ行)だけを表示させておきます。この状態で、編集したい節の `\section` の所までカーソルを持っていておいて `C-c C-o C-e` と入力すれば隠されていた文章部分はその節の部分だけ表示されます。編集を終えたら `C-c C-o C-c` と入力して再び文章を隠しておきます。そして編集集中での各節を部分コンパイルする時は `C-c *` を使うことで領域を指定しましょう (9.8.1参照)。コマンドは上に挙げた他にも色々あります。その一覧を表 9.1に示しておきます。

表 9.1 アウトラインマイナーモードのコマンド一覧

key	binding
C-c C-o C-a	すべてを表示する
C-c C-o C-t	すべてのボディを隠す
C-c C-o C-o	カーソルの置かれている節以外のボディを隠す
C-c C-o C-q	カーソルの置かれている節のレベル以下をすべて隠す
C-c C-o C-e	カーソルの置かれている節を表示する
C-c C-o C-c	カーソルの置かれている節を隠す
C-c C-o C-l	カーソルの置かれている節以下のボディを隠す
C-c C-o C-k	カーソルの置かれている節以下のヘッダ行を表示する
C-c C-o C-d	カーソルの置かれている節以下をすべて隠す
C-c C-o C-s	カーソルの置かれている節以下をすべて表示する
C-c C-o TAB	カーソルの置かれている節直下のヘッダ行を表示する
C-c C-o C-b	同じレベルで一つ前の節のヘッダ行に移動する
C-c C-o C-f	同じレベルで一つ後の節のヘッダ行に移動する
C-c C-o C-u	一つ上のレベルの節のヘッダ行に移動する
C-c C-o C-p	カーソルの置かれている所より前方で、現在表示されている最も近いヘッダ行に移動する
C-c C-o C-n	カーソルの置かれている所より後方で、現在表示されている最も近いヘッダ行に移動する
C-c C-o C-h	コマンドヘルプを表示する

9.8.4 \LaTeX マクロの入力

AUCTeXを使うと様々な \LaTeX マクロを完全に覚えていなくても使う事ができるようになります。これによって、 \LaTeX の教科書が横に無くても \LaTeX のコマンドが使えるようになります。そのための機能に、tsshの機能として馴染みのある<Tab>キーによる補完があります。ただし、AUCTeXでは<ESC> <Tab>になっています。候補が複数あると補完は途中で止まりますので、その続きを何文字か入力してさらに<ESC> <Tab>を入力します。候補がひとつになるまでこれを続け、ひとつになれば最後まで補完してくれます。あるいはマクロを対話式に入力する方法を使うことで覚えていない \LaTeX マクロを入力することもできます。C-c C-mで対話的に \LaTeX マクロの入力を求められるので、マクロ名を途中までしか覚えてなくてもコマンドライン補完機能を使うことでマクロの入力ができます。<Tab>キーや<Space>キーを押せば、マクロ名を補完してくれたりマクロ候補一覧を表示したバッファを開いてくれたりします。`\chapter{}`や`\section{}`も \LaTeX マクロなのですが、AUCTeXではこれらの入力には別のコマンドが割り当てられています。C-c C-sがそれで、以下のような対話によって、`\chapter{}`や`\section{}`の入力を行ないます。

```
Select level: (default section) chapter
What title: すごい章
What label: cha:wao!
```

9.8.5 複数行のコメントの付け外し

AUCTeXを使うと連続した複数行にわたってコメント記号%を一度に付ける事ができます。コメント記号をつけたい領域の先頭でC-<Space>、後尾の次の行でC-c;とすればその領域にわたって行頭にコメント記号が付けられます。コメント記号を外したい場合も同様にその領域の先頭でC-<Space>、後尾の次の行でC-u C-c;とすればコメント記号が削除されます。

9.8.6 書体の指定

AUCT_EX を使って書体指定の一括入力ができます. 例えば `\it` についてはイタリック補正 (8.4.10を参照) もセットされます. 書体を設定した後, `C-u C-c C-f <KEY>` で書体設定の変更ができます. `<KEY>` には変更後の設定したい書体に対応する `C-b C-i C-r C-e C-t C-s C-c` のいずれかが入ります.

表 9.2 書体の指定

bold	C-c C-f C-b
<i>italics</i>	C-c C-f C-i
roman	C-c C-f C-r
<i>emphasized</i>	C-c C-f C-e
typewriter	C-c C-f C-t
<i>slanted</i>	C-c C-f C-s
SMALL CAPS	C-c C-f C-c
書体設定の消去	C-c C-f C-d

9.8.7 数式モードの支援

表 9.3 数式モードでの記号入力

ギリシャ文字

α (alpha)	a
β (beta)	b
δ (delta)	d
ϵ (epsilon)	e
ϕ (phi)	f
γ (gamma)	g
η (eta)	h
κ (kappa)	k
λ (lambda)	l
μ (mu)	m
∇ (nabla)	N
ν (nu)	n
ω (omega)	o
π (pi)	p
θ (theta)	q
ρ (rho)	r
σ (sigma)	s

記号

τ (tau)	t	\rightarrow (rightarrow)	C-f	\supseteq (supseteq)]]
υ (upsilon)	u	\leftarrow (leftarrow)	C-b	\backslash (backslash)	\ \
χ (chi)	x	\uparrow (uparrow)	C-p	\setminus (setminus)	/ /
ψ (psi)	y	\downarrow (downarrow)	C-n	\cup (cup)	+ +
ζ (zeta)	z	\leq (leq)	<	\cap (cap)	- -
Δ (Delta)	D	\geq (geq)	>	\langle (langle)	((
Γ (Gamma)	G	\sim (tilde)	~	\rangle (rangle)))
Θ (Theta)	Q	∞ (infty)	I	exp (exp)	C-e
Λ (Lambda)	L	\forall (forall)	A	sin (sin)	C-s
Ψ (Psi)	Y	\exists (exists)	E	cos (cos)	C-c
Π (Pi)	P	$\not/$ (not)	!	sup (sup)	C-^
Σ (Sigma)	S	\in (in)	i	inf (inf)	C-_
Υ (Upsilon)	U	\times (times)	*	det (det)	C-d
Φ (Phi)	V	\cdot (cdot)	.	lim (lim)	C-l
Ω (Omega)	O	\subset (subset)	{	tan (tan)	C-t
		\supset (supset)	}	$\hat{}$ (hat)	^ ^
		\subseteq (subseteq)	[\vee (vee)	v v

AUCT_EX を使うと数式モードで利用される記号の入力が簡単になります. `C-c ~` で LaTeX-math-mode になります. 再び `C-c ~` で LaTeX-math-mode から抜けます. 数式モードでは「`'`」(逆シングルクォート)

と一文字の入力で数式環境でよく使われる記号が展開されます。例えば ‘ a で `\alpha` と入力されます。表 9.3が数式モードのコマンド一覧です。

9.9 最後に

AUCT_EXにはまだ他にも幾つかのコマンドがあります。また、カスタマイズについては説明しませんでした。それらについては/NF/local/general/doc/にあるマニュアル(auc-tex.ps, tex-ref.ps)やinfoファイルを参照して下さい。英語はいやだ!という人には、UNIXmagazineの'93年9月号(基本的な使用方法)、10月号(様々な機能)、11月号12月号(カスタマイズの方法)が図書館にあります。また、X環境でMuleを使っている場合に限られますが、AUCT_EXはMuleの(というよりEmacs 19の)拡張機能の一つであるメニューバーに対応しています。(Japanese-)L^AT_EX-modeに入ると**LaTeX**という項目が追加され、AUCT_EXのコマンドのメニューが登録されています。

このマニュアルはWWWの学内の「各種ドキュメント、情報検索サービス」にも載せてありますので、併せて御利用下さい。

第 10 章

レポート システム

レポートシステムは、あなたが出そうと思っているレポートを紙に印刷して学部事務室や先生に提出する代わりに、電子メールで送るものです。電子メールを扱う方法さえ理解していればレポートシステムを使う事が出来ます。

10.1 レポート システムを使う前に

レポートをメールで提出するという事は、レポートの内容をファイルに作成して、そのファイルをメールとしてレポートシステム宛に送るという事を意味しています。

どのようなレポートでもいつでも闇雲にレポートシステム宛に送ればいいというものではありません。そのレポートが何という先生の何という課題のレポートなのかが判らないと意味がありません。すなわちあなたが出そうと思っているレポートにはあらかじめ担当の先生によって設定されたレポート名がある筈です。このレポート名を間違えたり登録されていないものだったりした場合、レポートシステムはそれを受け付けません。まずこれを注意して確認して下さい。

また、レポートには提出期限があります。この期限はあらかじめ先生によって課題ごとに決められたもので、この期限を過ぎた提出をレポートシステムは受け付けません。この期限も確認しておいて下さい。

提出期限内で先生による評価前であればレポートは何度でも訂正して再提出できます。逆に評価後の再提出をレポートシステムは受け付けません。

これらの条件を越えるような場合 (期限後の提出、評価後の再提出) については先生に相談して下さい。

10.2 レポート を提出するには

レポートメールの宛先は常に `report@cc.kyoto-su.ac.jp` です。レポートをメールで送る時にはレポート名を示す名前をメールのサブジェクトに付けてやります。

レポートはメールとして送られる為、一般のメールと同じ制限があります。例えば C や FORTRAN のプログラム演習で作成した実行モジュールなどのバイナリファイルはそのままでは送れません。レポートを読んで採点する先生の為に一行はある程度 (半角 70~80 文字) で改行した方がいいでしょう。(もちろん仕方なく長くなる場合を除きます。)

まずレポートの内容をファイルにまとめます。これ以降の例では `report.txt` という名前のファイルにレポートを書いたとしましょう。レポートファイルの先頭には誰のレポートか読んだ時にはっきり判るように学生証番号、氏名、課題の名前などを書いておく事を心がけましょう。

また、例では `class1/report1` というレポート名で提出する事にします。

また、今までメールは Mule の MHE などを利用して読み書きしてきたと思いますが、このレポートシステム宛にメールを送るには `mail` というコマンドで行ないます。このコマンドは UNIX コンピュータに `login` した状態で表示されるシェルプロンプト (`csosf01(81)%` のような表示) に続いてタイプするコマンドです。

10.2.1 そのレポートを初めて提出する場合

レポートのファイルをメールとして `report@cc.kyoto-su.ac.jp` 宛にサブジェクトをつけて送ります。サブジェクトに `send` 指定と、先生が指定するレポート名を書きます。具体的には以下のようにします。

```
mail -s "send, class1/report1" report@cc.kyoto-su.ac.jp < report.txt
```

レポートファイル指定の前の「<」を忘れないように注意して下さい！

提出すると受理されたかどうかのメールが返ってきます。

きちんと受理されたのなら「確かにあなたのレポートを受け取りました」とメールで返ってきますし、失敗した時は後述のエラーメッセージがメールで返ってきます。メールが返ってこない時は何らかのミスで受理されていません。何回やっても駄目な時は先生に相談して下さい。

10.2.2 そのレポートを再び提出する場合

一度出したレポートも、期限前でしかも先生がまだ採点していなければ、再提出して以前のものと置き換える事が出来ます。

レポートのファイルをメールとして `report@cc.kyoto-su.ac.jp` 宛にサブジェクトをつけて送ります。サブジェクトに今度は `replace` 指定と、先生が指定するレポート名を書きます。具体的には以下のようにします。

```
mail -s "replace, class1/report1" report@cc.kyoto-su.ac.jp < report.txt
```

レポートファイル指定の前の「<」を忘れないように注意して下さい！

初めての提出と同じように受理されたかどうかのメールが返ってきます。

10.2.3 提出したレポートの確認

提出したレポートがどんなものだったか確認する事も出来ます。以下のようにサブジェクトに今度は `check` 指定と、確認したいレポート名を書いて `report@cc.kyoto-su.ac.jp` 宛にメールを送って下さい。その際、メールの中身は何でも構いません。以下に中身が空メールで `check` する例を示します。

```
echo | mail -s "check, class1/report1" report@cc.kyoto-su.ac.jp
```

きちんと受理されたのなら以前に提出したレポートがメールで返ってきます。何かおかしければ後述のエラーメッセージがメールで返ってきます。

10.3 返ってくるメールのメッセージ

- このレポートは既に評価されています。

既にレポートに対して評価が行なわれてしまった為、受理できません。まだ期限以内なら担当の先生に相談して下さい。

- **確かにあなたのレポートを受け取りました。**

初めての `send`、及び `replace` の場合で、きちんと受理されました。

- **あなたが提出しようとしたレポートはすでに提出されています。置き換えるなら `replace` で再提出して下さい。**

間違って上書きすることのないよう、二度目以降の `send` をした時の警告です。(この `send` コマンドは受理されていません)

- **これがあなたの提出したレポートです。**

`check` で、前にそのレポートを出していた場合に現在受理しているものを送ります

- **あなたが確かめようとしているレポートは提出されていません。**

`check` をしようとしたレポートがまだ提出されていません。

- **あなたの提出しようとしたレポートのコマンドが判別できませんでした。Subject に `send`、`replace`、`check` のいずれかを指定して下さい。**

subject に `send`、`replace`、`check` 以外の物が指定してあります。確認してもう一度送り直して下さい。

- **あなたの提出しようとしたレポートは期限が過ぎていますので受け付けられません。**

`send`、及び `replace` をしようとしたレポートの提出期限が過ぎていますので受理できません。

- **あなたの提出したレポートの宛先が見つかりませんでした。サブジェクトをお確かめの上もう一度送って下さい。**

指定されたレポート宛先が見つかりませんでした。おそらく宛先を書き間違えていると思われるかもしれません。もしくは宛先が存在しません。

- **あなたの提出したレポートにはサブジェクトが付いていなかった為どこ宛のレポートか識別できませんでした。もう一度送り直して下さい。**

メールにサブジェクトが付いていませんでした。サブジェクトを付けてもう一度送って下さい。

特殊な場合

- **あなたの送られたメールからは日付が判別できませんでした。もう一度他のメールシステムからお送り下さい。**

メールから日付が判別できませんでした。特殊な環境から送っていませんか? cc 環境から送ってみて下さい。

- **レポートシステムがあなたのメールを宛先に書き込むことが出来ませんでした。担当の先生に御連絡下さい。**

メールをシステム上処理できませんでした。まず起こりませんが、起こった時は速やかに担当の先生に御連絡下さい。(ディスクスペースが無くなっていると思われます。)

メールが返ってこない場合

宛先を省略していませんか? `report` ではなく `report@cc.kyoto-su.ac.jp` として下さい。

第 11 章

Mathematica

11.1 Mathematicaってなあに？

Mathematica は Wolfram Research, Inc から発売されている数学とその応用のための汎用プログラムです。『数学を使って何かをやろう』と思っているけど『計算力に自信がない』というあなた。『数学を使って何かをやらなければならない』けど『面倒な計算は嫌だ』というあなた。そして、『数学を見てみたい』というあなたのためのプログラムです。

と、いっても何のことが解らない人も多いと思われるので、とにかく動かして試してみましょう。

11.1.1 とにかく起動、そしてやってみる！

とにかくやってみましょう。まず、2号館の4階の21情報処理教室の csosf01 ~ csosf40 でのやりかたを例にして説明します。

まず csosf01 ~ csosf40 のうちにどれかに login し、ルートメニューの中のアクセサリメニューから「Mathematica」を選択します。また、ターミナルからコマンドとして `mathematica &` と打ち込んでも構いません。すると、

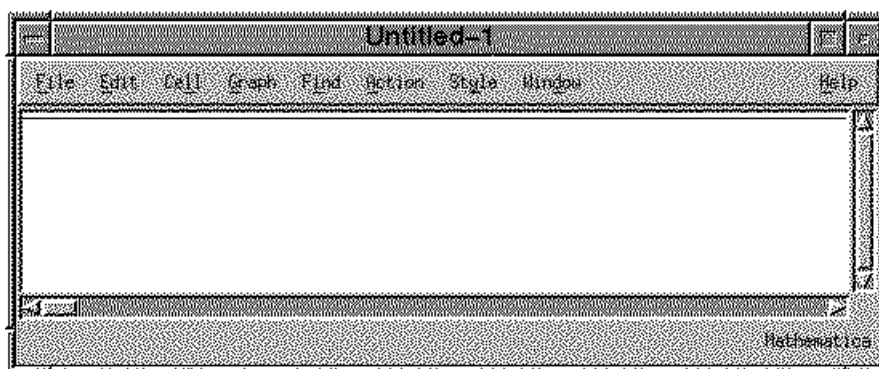


図 11.1 ノートブックのウインドウ

と、いうふうに、Mathematica のノートブックというウインドウが新しく出てきます。そのノートブックが何かという説明は後回しにしましょう。とにかくやるのです。

先ず試しに簡単なかけ算をやってみましょう。4 <Space>4 <Shift>+ <return>と入力します。しばらくするとウインドウは以下のようになります。

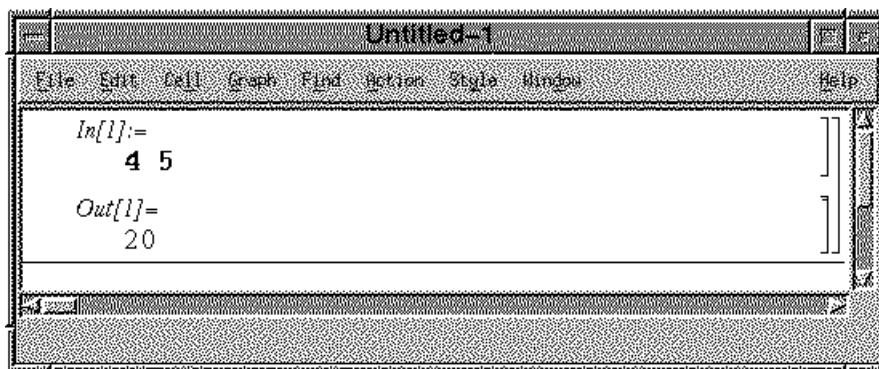


図 11.2 最初の簡単なかけ算

『こんな簡単な計算にこんなに時間がかかるのか!?』などとは驚かないで下さい。これは Mathematica を起動して最初の計算をやらせる時だけです。

さて、もう少し試してみしましょう。今度は $\sqrt{10}$ の計算をやってみましょう。

`N[Sqrt[10], 40]` <Shift>+ <return>と打ち込みます。入力する時は必ず、<Shift>+ <return>です。<return>だけでは駄目です。以下<Shift>+ <return>は省略します。つまりこれからは『 `N[Sqrt[10], 40]` と入力する』という表現を使います。

ともあれそうすると、

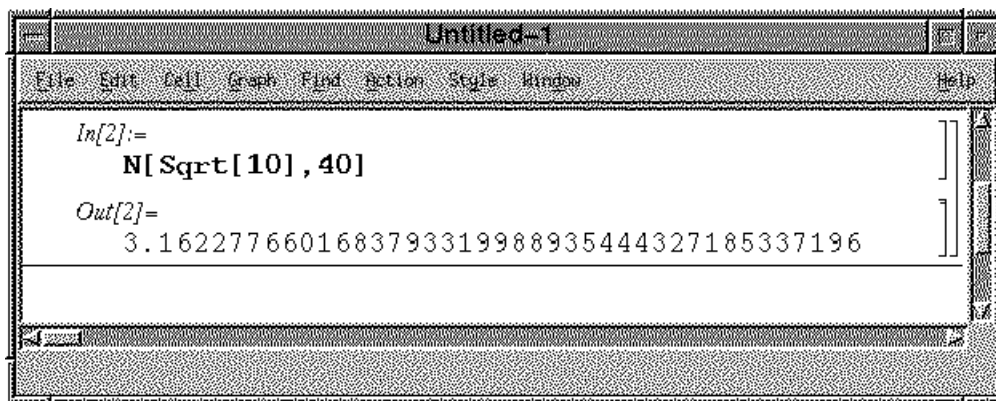


図 11.3 ちょっとした計算

となります。今度は素早く計算したはずですが。

また、以下のようなグラフィックを描くことも可能です。

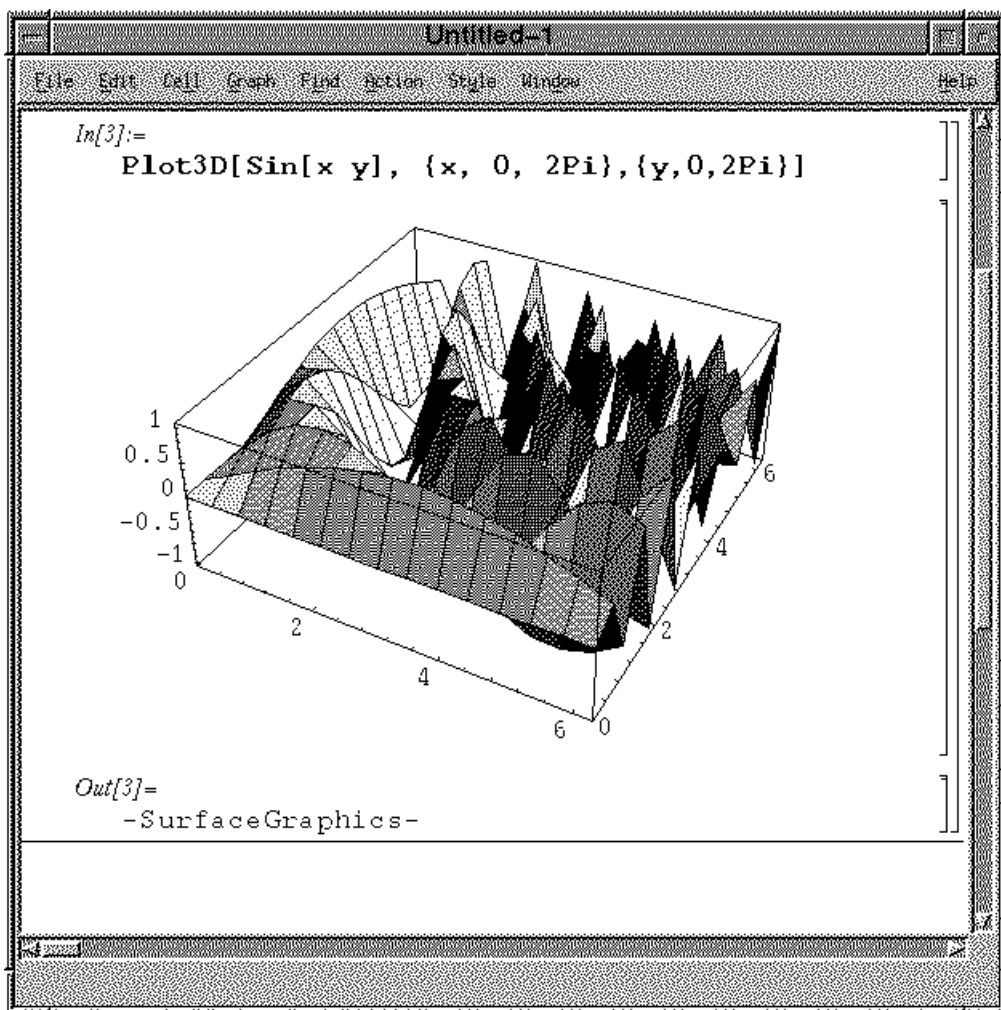


図 11.4 $y = \sin(xy)$ のグラフ

終了する時は mathematica の File メニューから Quit を選択します。

11.1.2 その他の場所の Mathematica

2号館 21 情報処理教室の他に産大では以下の場所に行けば Mathematica を使うことができます。

- 1) 計算機科学研究所棟 3 階の C2 情報処理教室
直接コマンドラインで `mathematica` と入力するだけです。すると、`csosf01` ~ `csosf40` と同様にノートブックウィンドウが出てきます。
- 2) 計算機科学研究所棟 3 階の C3 情報処理教室の NeXTStation 及びあちこちにあるマッキントッシュや Windows パソコン。
Mathematica のアイコンをダブルクリックするだけです。それでノートブックウィンドウが出てきます。

11.2 ノートブックとコマンドラインについて

さて、先に、

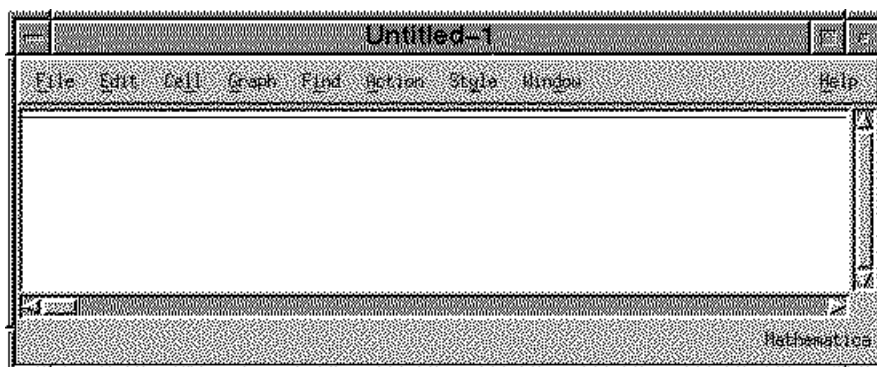


図 11.5 ノートブックのウインドウ

をノートブックウインドウと呼びました。これは実際のノートと同じような機能を持っていて Mathematica で行なったすべての入出力をまるごとファイルにしてセーブしておくことができます。もちろんそれをまたオープンすることもできます。

使い方も簡単でマウスでメニューから選択するだけです。

また、21 情報処理教室や C2 情報処理教室では、このようなノートブックではなくて直接コマンドラインで Mathematica を使うこともできます。その場合は `math` と入力します。ノートブックとは違って、最初の計算が極端に遅い、ということはありません。しかし 21 情報処理教室ではグラフィックが出ないという欠点もあります¹。

コマンドラインで終了する時は `Quit` と入力します。

11.3 Mathematica の簡単な命令

1) : 数値計算

- 足し算: $x+y$
- 引き算: $x-y$
- かけ算: $x*y$ または $x y$

```
In[n]:= 3 4   ← 入力
Out[n]:= 12   ← 出力
```

- 割算: x/y
- 乗巾: x^y

¹が、グラフィックは PostScript で書かれているので、ファイルにしてユニックスの他のコマンドを使って表示することはできません。

- 厳密な計算:

- 通常の電卓と違い、厳密な計算も行ないます。

- In[n]:= 3/7 ← 入力

- Out[n]:= 3/7 ← 出力

- In[n]:= 3/7 + 1/3 ← 入力

- Out[n]:= 16/21 ← 出力

- 近似値計算:

- //N を使えば近似値で出ます。

- In[n]:= 3/7 + 1/3 //N ← 入力

- Out[n]:= 0.781905 ← 出力

- 実数値で入力しても近似値となります。

- In[n]:= 3.0/7.0 ← 入力

- Out[n]:= 0.428571 ← 出力

2) : 数学関数

数学関数は、普通の英語の名前がついていて大文字から始まり引数は [] の中に入れるようになっています。また、複数の引数をとる場合は引数と引数の間を「 , 」で区切ります²。例えば以下のような関数があります。

- 平方根 (\sqrt{x}) : Sqrt[x]

- 自然対数 ($\log_e(x)$) : Log[x]

厳密な計算、近似値計算 //N の使い方は 1) : 数値計算と同じです。

数学関数は、普通の英語の名前がついている訳ですから、逆に名前から探すこともできます。つまり、Sin[x] って、定義されているかな、と思った時に。

- In[1]:= ??Sin

とすると、Sin[z] gives the sine of z と出てきます。³ つまり、Sin[] も数学関数として組み込まれていて、

- In[1]:= Sin[3] //N

- Out[1]:= 0.14112

というふうになります。

3) : 代数解析

- Expand[]: 式を展開

²[x y] では [x*y] というふうにかかけ算になってしまいます。

³これは次章でも説明します。

– In[1]:= Expand[(x + y)^3] ← 入力
– Out[2]:= $x^3 + 3x^2y + 3xy^2 + y^3$ ← 出力

● Factor[]: 因数分解

– In[1]:= Factor[x^3 + 3 x^2 y + 3 x y^2 + y^3] ← 入力
– Out[2]:= $(x + y)^3$ ← 出力

● Integrate[]: 積分

– どの変数について積分するか指定が、式の後に必要です。

– In[1]:= Integrate[2 x, x] ← 入力
– Out[2]:= x^2 ← 出力

● D[]: 微分

– どの変数について微分するか指定が、式の後に必要です。

– In[1]:= D[x^2, x] ← 入力
– Out[2]:= $2x$ ← 出力

4) : グラフィック

● Plot[f(x), {x, 0, 10}]: f(x) を x = 0 から 10 まで表示

● Plot3D[f(x, y), {x, 0, 10}, {y, 0, 10}]: f(x, y) を 3 次元で表示

例えば、 $y = \sin(x)$ というグラフを $x = 0$ から 2π まで表示したければ、Plot[Sin[x], x, 0, 2Pi] と入力します。 $z = \sin(x + y)$ というグラフを $x = 0$ から 2π まで $y = 0$ から 2π 表示したければ、Plot3D[Sin[x + y], x, 0, 2Pi, y, 0, 2Pi] と入力します。

11.4 入出力一般

この章では(いろいろな方法で) Mathematica を起動した後、どのように Mathematica を利用すれば良いか、について説明します。

直接入力する場合のコマンド色々

1) 前の結果の取り込み

- 最後に得られた結果: %
- k 回前に得られた結果: %%,,,,% (k 回)
- 出力 Out[n]:= で得られた結果: %n

– 以下のように使います。

```

In[1]:= 3/7 + 1/3 //N   ← 入力
Out[1]:= 0.781905     ← 出力
In[2]:= %             ← 入力
Out[2]:= 0.781905     ← 出力
- In[3]:= %% + 1      ← 入力
Out[3]:= 1.781905     ← 出力
In[4]:= %2 + 2       ← 入力
Out[4]:= 2.781905     ← 出力

```

2) 変数

数値計算とはいえ、やたら長い数値を入力するのは面倒ですよ。少しでも楽をしましょう。

- 変数に値を代入： $x = \text{値}$

– 変数は任意のアルファベットが使えますが、前章の組み込み関数とかち合わないよう、小文字を使いましょう。以下のようにして使います。

```

In[1]:= x = N[Pi, 10]  ← 入力
Out[1]:= 3.145926535   ← 出力
- In[2]:= x            ← 入力
Out[2]:= 3.145926535   ← 出力
In[3]:= x 2 3          ← 入力
Out[3]:= 18.8496       ← 出力

```

3) 情報を引き出す

前章でいろいろな関数、コマンドを列挙しました。しかしそれらを常に覚えていなければならない訳ではありません。

特に数学関数は Mathematica では $\sin(x)$ は `Sin[x]` などというように**普通**の英語の名前が付けられています。そこで、どんな関数があるか、どんな命令があるかは比較的調べ易くなっています。

例えば `N` の使い方を忘れてしまったとしましょう。そのときは、

- `In[1]:= ??N` ← 入力

と、すると、

- `Out[1]:= N[expr] gives the numerical value of expr. N[expr, n] does computations to n-digit precision`

などと出てきて、使い方、内容などを表示します。

ファイル

Mathematica はエディターを内蔵していません。よって、直接、`Plot3D[sin[x y], x, 0, 2Pi, y, 0, 2Pi]` と打ち込んでしまったら気が滅入って来ます⁴。そこで、普通にファイルをつくってそれを Mathematica に読み込ませましょう。

⁴sin ではなくて Sin と書かなければなりませんよね。

- Mathematica に file を読み込ませる : <<filename
- Mathematica から file に expr を書き込む : expr >> filename
- Mathematica で file の内容を表示 : !!filename
- エディタを立ちあげて、filename という名前の file をつくって、中に「N[Pi, 10]」と、書いてあるものとして説明します。

```
In[1]:= <<filename      ← 入力
Out[1]:= 3.145926535    ← 出力
- In[2]:= 2 4 >> filename ← 入力
In[3]:= !!filename      ← 入力
Out[3]:= 8              ← 出力
```

もちろん N[Pi, 10] のように簡単な命令ではなく、複雑な命令、打ち間違い易い命令を入力する時に、この方法は便利なものとなります。

何回も出したいグラフィック、式などはこのやり方で file の形で取っておくことをお進めします。

印刷

ノートブックで作った結果に関しては簡単に印刷する事が出来ます。Mathematica の File メニューの下の方にある「Print...」を選択します。すると以下のようなウィンドウが表示されます。

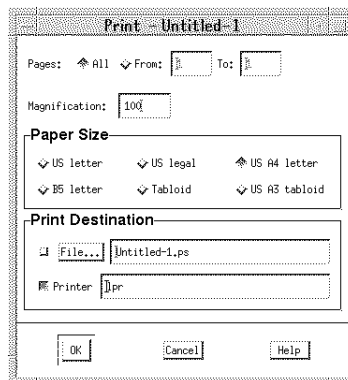


図 11.6 プリントウィンドウ

ここで「Printer」の設定に注意して下さい。まず「Printer」のすぐ左にある小さな四角のボタンが押されている状態⁵になっている事を確認して下さい。もしも押されていないようならマウスでこの四角のボタンをクリックして押して下さい。「Printer」のすぐ上の「File...」の左の小さな四角がもしも押されている状態ならばこれもクリックして今度は押されていない状態にしておきます。次に lpr と書いてある右側に以下のようにプリンタの名前を追加しましょう。

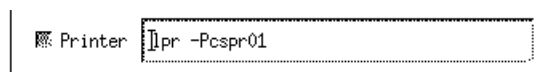


図 11.7 プリントウィンドウ (プリンタ設定の部分)

⁵判りにくいですが、まわりより暗くなっているのが押されてへこんでいるという意味です。

例えば 21 情報処理教室であれば `lpr` の右に空白を一つ開けて `-Pcspr01` と書き足してプリンタの指定をします。どの部屋でどのプリンタを指定するべきかは 107ページの 4.6を参照して下さい。上の図のようにできたら「OK」をクリックします。しばらくすればプリンタから結果が印刷されてくるでしょう。

ところでこれでは全文印刷されてしまいますが、ノートブックでは指定した部分だけを印刷する事も出来ます。*In* や *Out* などの行の右端に「`]`」などのようなカギ型の枠があると思います。印刷したい部分の右の枠をクリックすると、その黒くなった部が選択された事になります。

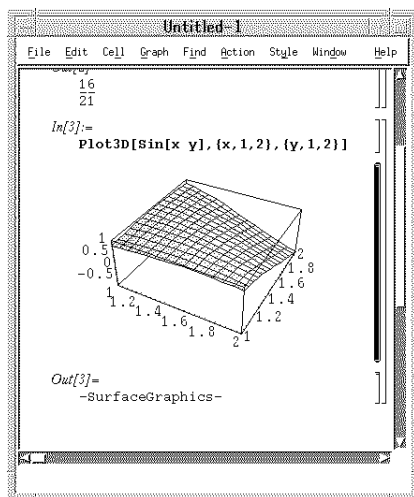


図 11.8 選択されたグラフ（グラフの右の黒い枠に注意）

Mathematica の File メニューの Print Selection... を選択すれば、図 11.6が表示されますので、先ほど紹介した印刷の手順にしたがって操作して下さい。今度は全文では無く選択した部分（上の例ではグラフ）だけが印刷されると思います。

11.5 さらに進みたい人には

このドキュメントは『ともかく Mathematica に触れてみよう』という人を対象に書かれています。ですから、ちゃんと Mathematica でプログラムを書こうという人や、実際に実験や研究で使おうという人は、Mathematica を作った本人、スティーブンウルフラムの書いた *Mathematica* というでっかい本がありますから⁶、それを参考に勉強を進めて下さい。

また、X ウィンドウ環境で利用している人だけに有効な手段ですが、ノートブックウィンドウにある「Help」というメニューで関数などの一覧を表示する事も出来ます。更に `mathbook` コマンドで組み込み関数のマニュアルが表示されます。

⁶ C2 情報処理教室に何冊かあります。

第 12 章

GNUPLOT

12.1 概要

GNUPLOT は対話式の操作によって数値データや数学関数のグラフを作ることができるプログラムです。GNUPLOT は誰でも自由に無料で使用できます。

GNUPLOT の最大の特徴は何といてもグラフを出力するために利用できるプリンタなどの出力装置の種類が非常に多いことでしょう。また、GNUPLOT はいろいろなコンピュータシステムの上で動き、16 ビットの小さなパソコンでも、もっと大きな UNIX マシンでも同じように使用することができます。広く使われている物であれば、GNUPLOT が動作可能なシステムと対応可能なプリンタのリストの中に、あなたが使っている物が含まれていると期待してよいでしょう¹。

GNUPLOT には、いろいろな形式のファイルとしてグラフィックスを出力する機能もあり、そうしたファイルを使ってたとえば L^AT_EX の文書の中に高品位のグラフを入れることができます。出来上がりはプロの仕事には及ばないかもしれませんが、数式やグラフの入った文書がかなり手軽に費用もかけずにできるのです。

GNUPLOT は線と点を使った 2 次元グラフと、メッシュや等高線を使った 3 次元グラフのいろいろなバリエーションを描くことができます。ただ、パイ・チャートや絵グラフを作ることはできませんし、スケールの異なる座標軸を使った複合グラフも直接には作れません。GNUPLOT は派手なものより、どちらかというと、地味なデータ分析のような実務に向いています。

GNUFIT というプログラムは GNUPLOT を拡張してフィッティングの機能を付け加えるものです。これは数式の中の未知パラメータを自動的に調整して与えられたデータに合わせる機能です。ただし、機能が低く限られた場合にしか使うことができません。GNUFIT が組み込まれている場合には、GNUPLOT を起動したときにオープニングメッセージの中にこのことが表示されます。

興味がわいてきたら、この後の説明に従って GNUPLOT をちょっと使ってみてください。ここでは cc2000 上の GNUPLOT をワークステーションやパソコンなどの X 端末から動かすことを想定していますが、21 情報処理教室にある DEC-3300 にも GNUPLOT が入っていて、cc2000 の GNUPLOT と同様にして利用できますから、21 情報処理教室があなたの行きつけの場所ならこちらを使ってください。その場合は、DEC-3300 の前に座ってマシンにログインして使います。ログインしたら自動的に開く DECterm もしくは kterm ウィンドウの中で作業してください。

これらの環境では、処理速度や操作性は十分ですし、かなり良い品質の作品を得ることができます。cc2000 や DEC-3300 で試した後、他のコンピュータでも GNUPLOT を使いたいと思ったら、`sandai.question`

¹GNUPLOT 3.5 は UNIX、VAX/VMS、OS/2、MS-DOS (PC/AT 機)、MS-Windows、Macintosh など動作する。NEC の PC-9801 で動作するバージョンもある。対応するプリンタや出力できるファイル形式については、GNUPLOT を起動してから `set term` と打てばリストが出る。

ニュースグループなどに投稿して情報を得ればよいでしょう。

12.2 GNUPLOT を初めて使う

では実際に GNUPLOT を使ってみましょう。X 端末からいつものように cc2000 にログインしてください。(DEC-3300 の GNUPLOT を使おうと思っている方は必要ありません。) エ? どういうことかわからない? それならとりあえず、明かりのついたどこかの情報処理教室に入って、そこにいる誰かに教えてもらってください。おっと! パスワードを忘れた? それなら計算機センターに行つてなんとかしてもらってください。それからここに戻つて先を続けてください。

12.2.1 GNUPLOT の起動

cc2000 (または DEC-3300) にログインした後、一度 `kterm&` と打つて (最後はいつもリターンキーを打つ。`kterm` の後に `&` を忘れないこと。)、もう一つのウィンドウを開いてください。そこで作業します²。元のウィンドウもいろいろ使いみちがありますから、そのままにしておいてください。新しいウィンドウで `gnuplot` と打つて下さい。これで GNUPLOT が動きだし、オープニング・メッセージと、`gnuplot>` というプロンプトが現れます。オープニング・メッセージの中にこの GNUPLOT のバージョン情報とグラフィックスを出力する端末の種類の設定が書かれています。`Terminal type set to 'x11'` と表示されているのが端末の種類です。

万が一そうになっていなければ、`quit` と打つて GNUPLOT を終了し、GNUPLOT の個人用の設定をチェックしなければなりません³。脚注のようにできたら、再び GNUPLOT を起動してください。

12.2.2 とりあえずグラフを!

うまく起動できたら、GNUPLOT のプロンプト `gnuplot>` に対して次のように入力してみてください。大文字と小文字は区別されますから、注意して使い分けてください。また、式の中の括弧には小括弧 () を使います。

```
plot sin(x)/x
```

図 12.1 のように、ウィンドウが開いてグラフが描かれたでしょう。これは $y = \sin x/x$ のグラフです。初めは x 軸 (横軸) の範囲は -10 から 10 までになっています。 y 軸 (縦軸) の範囲は自動的に調整されます。この関数は $x = 0$ では 1 で、 x が 0 から離れるにしたがって振動しながら小さくなっていきます。今描かれたグラフでその様子がわかりますね。

この新しいウィンドウは決して勝手に閉じないでください。もし勝手に閉じてしまうと、GNUPLOT を起動しなすまでグラフが描けなくなります。GNUPLOT に次のコマンドを与えるために、マウスの左ボタンで、プロンプト `gnuplot>` が出ている GNUPLOT のウィンドウを選んでください。

x 軸の範囲を変更するには、`set xrange [-20:20]` というように命令を与えますが、まず `set` とだけ打つてみてください。これでは不完全なので、GNUPLOT は何が足りないかを教えてくれます。あらためて `set xrange` と打つと、今度は [が足りないと教えてくれるでしょう。: のところをわざと間違えて、`set xrange [-20,20]` と打つてみてください。間違いがちゃんと指摘されますね。では、今度こそ正しいコマンドを入力してください。GNUPLOT が何も言つてこなければ OK です。え? グラフが変わらない?... 少し待って下さい。範囲を変えて描いたグラフはまもなく見ることができます。

² さらに、もしあなたが UNIX に少し慣れていれば、GNUPLOT の練習をするためのディレクトリを `mkdir` で作つて `cd` でそこを作業ディレクトリ (カレントディレクトリ) とすることをお薦めします。

³ ホーム・ディレクトリに `.gnuplot` というファイルがあればそれを消去し、`GNUTERM` という環境変数が設定されていればこれを取り外す。

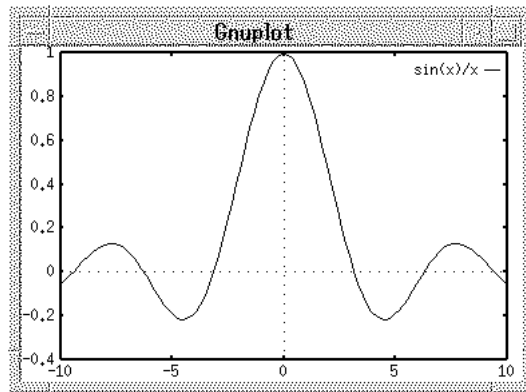


図 12.1 $y = \sin x/x$ のグラフ

x 軸の範囲がうまく変更されたかどうかは、`show xrange` と打ってやればわかります。`set` コマンドにはいろいろなバリエーションがあり、いろいろな設定を変更するのに使います。これは最も頻繁に使用される命令です。一方、`show` コマンドはいろいろな設定状態を調べるのに使います。

では、先程の関数をもう一度グラフにしてみましょう。`replot` と打つだけです。この命令は一番最近グラフにした関数をもう一度グラフにする命令です。縮めて `rep` でもだいじょうぶです。 x 軸の範囲が2倍になったのが確認できますね。

12.2.3 コマンドライン編集

ここで、我々のような面倒くさがり屋のために GNUPLLOT が持っているコマンドライン編集機能というものを紹介しておきましょう。これはタイプを楽にする機能ですが、`tcsh` シェルや `VMS` をお使いならお馴染みのものです。

まず `↑` キーを打つてみてください。前に入力したコマンドが呼び出されたのがわかりますね。もう一度打つとその前のコマンドに、さらにもう一度打つと前の前のコマンドに変わります。`↓` キーは逆向きの働きをします。前に入力した `set xrange ...` を呼び戻し、`←` キー、`→` キー と `delete` キーを使って範囲を `-50` から `50` に書き直してみましょう。`delete` キーは「左に向かって消していく」ことに注意してください。

書き直したら、それを GNUPLLOT に新しいコマンドとして与えるためにリターンキーを押します。その後 `replot` でまたグラフを描いてみてください。

12.2.4 サンプル数の変更

`set` 命令の別のバリエーションの一つを試してみましょう。さきほど描かれたグラフはちょっと荒っぽく描かれているでしょう？これは、式の値を計算する点の数が少なすぎるのです。`show samples` と打てば、恐らく、`sampling rate is 100,100` と表示されるでしょう。これは、 x 軸方向に 100 点しか計算していないことを表します。(前の方の 100 です。後の方の 100 は今は無視してください。)

では、計算する点の数を 200 にするために、`set samples 200` と打ってください。それからまた、`replot` とやってください。今度はずっとましになったと思います。

12.2.5 印刷しましょう

グラフを印刷する方法はシステムによっていろいろ変わってきます。ここでは、「PostScript」という形式のファイルに一旦出力してからそれをプリンタに送って印刷するという方法を説明しましょう。

まず、GNUPLOT で初めて作るグラフですからそれにふさわしいタイトルを入れた方がいいですね。ついでに目盛り線の描きかたもちょっと変えましょうか。次のように入力してください。

```
set title "Kinsan TOYAMA won the first output from GNUPLOT. Here it is!"
set grid
replot
```

タイトルはダブルクォート " で囲みます。タイトルの中身は好きなように書いてください。タイトル入りのグラフが描けましたね。目盛りは格子状 ... grid ... になったでしょう。

次に、グラフを PostScript ファイルに出力します。次のように入力してください。

```
set term postscript
set output "graph1.ps"
replot
set output
set term x11
```

最初の2行で、出力形式と出力先のファイル名を指定し、3行目の `replot` で実際に出力しています。後の2行は、出力先を元に戻し（つまり画面に）戻し、出力形式を `x` 端末に合わせ直しています。ちょっと面倒なようですが、ファイル名以外は `se te pos` というように縮めても認識してくれますし、間違えてもコマンドライン編集の機能で簡単に直せますからだいじょうぶです。

次は、いよいよ印刷です。一番始めに開けておいたウィンドウをマウスの左ボタンで選んでください。 `cc2000(81)%` のようなシェル・プロンプト が出ているウィンドウです。 `ls` コマンドで `graph1.ps` というファイルがあることを確認してください。 `ls` と打つだけです。

では、このウィンドウで

```
lpr -Pcspr01 graph1.ps
```

と打って、（`cspr01` は 21 情報処理教室にあるプリンタ。使おうと思うプリンタに置き換えること！）あなたの作品がプリンタから出てくるのを待ってください⁴。

`set output` コマンドでファイル名のかわりに UNIX の「パイプ」というものを指定して、GNUPLOT の中から直接印刷することもできます。

```
set term postscript
set output "| lpr -Pcspr01"
replot
set output
set term x11
```

この場合は `replot` と打った段階でグラフがプリンタに送られます。 `graph1.ps` のような PostScript ファイルは作られません。

⁴`lpr` コマンドの `-P` とプリンタ名の間は開けない。

12.2.6 オンラインマニュアル

GNUPLOT は操作画面上でマニュアルを読むことができます。(ただし、GNUFIT が組み込まれていてもフィッティングに関する項目は含まれていません。) 次は、このオンラインマニュアルの使い方を練習しましょう。印刷したマニュアルを作ったり、オンラインマニュアルを日本語にする方法については 12.5 節で説明します。

GNUPLOT のプロンプト `gnuplot>` に対して `help` と打って英語のオンラインマニュアルが出るのを確認してください。Help topics available: の下に、`copyright` とか `introduction` といった次に見ることのできる項目のリストも表示されているでしょう。

この中の `expressions` という項目を見るために `Help topic:` というプロンプトに対して `expressions` と打ってみてください。 `expr` というように始めの何文字かだけでも大丈夫です。これで、GNUPLOT の式についての説明が出ます。今は説明の中身まで読む必要はありません。さらっとながめるだけで結構です。

さらに、`Subtopic of expressions:` に対して `functions` と打つと関数の説明が出ます。? と打つと項目リストを再表示させることができるということも覚えておいてください。

項目名を入れずにリターンキーのみ打てば 1 つ前のレベルに戻ります。リターンキーを何度か打ってプロンプトを `gnuplot>` に戻してください。

次に、`help sin` と打ってみてください。いきなり `sin` 関数の説明が出たでしょう。また、`help functions` とやってみてください。(この場合最後の `s` まで必要です。 `function` では別の項目になります。) これで、オンラインマニュアルの使い方がわかりましたね。なお、`help` のかわりに `h` でもだいじょうぶです。

オンラインマニュアルは英語ですが、これが気に入ってきたら、複数のウィンドウを使って作業するといかにもスマートです。別のウィンドウでもう一つの GNUPLOT を起動して、片方にオンラインマニュアルを表示しながらもう一方で作業するのです。始めに開いたウィンドウが `シェル・プロンプト` を表示するだけで遊んでいます、そこに `kterm&` と打つてもう一つウィンドウを作り、そこで GNUPLOT を起動するのが良いでしょう。新しく起動した GNUPLOT でオンラインマニュアルを表示し、今使っている方でこの後の作業を続けてください。

12.3 GNUPLOT の終了と再開

ちょっと疲れましたね。(え? 疲れちゃいない? じゃあ疲れたということにしてください。) GNUPLOT を一度終了して一休みしたい ...

12.3.1 作業の保存

では GNUPLOT を終了しましょう。でも、ちょっと待った! 今まで行った作業は GNUPLOT を終了したら何も残らないのでしょうか? いいえ、ちゃんと残す方法があります。

`save "gplot1.plt"` と打ってください。 `gplot1.plt` という名前のファイルができて、この中に現在の GNUPLOT の状態が記録されます。この種のファイルを GNUPLOT のマクロファイルといいます。これでもう、いつ GNUPLOT を終了してもだいじょうぶです。本当にだいじょうぶかどうかは、次に GNUPLOT を起動したら最初に調べてみることにします。

GNUPLOT を終了するために `quit` と打ってください。 `exit` でも同じ意味です。 `Control-D` (コントロール・キーを押しながら `D` を押す) でも終了できます。終了したら、念のために `ls` コマンドで `gplot1.plt` ファイルがちゃんとあることを確認してください。

他の GNUPLOT も終了し、ウィンドウも全部閉じてログアウトしていただいて結構です。

12.3.2 GNUPLOT の再開

ゆっくり休みましたか？ では、再び cc2000 にログインして宿題をかたづけましょう。cc2000 にログインしたら、ls コマンドで前に作ったマクロファイル `gplot1.plt` があるのを確認してください。もしなければ、ディレクトリが違うのでしょうから、前に居たディレクトリに移る必要があります。カレントディレクトリ（作業ディレクトリ）に `gplot1.plt` があるのを確認したら、前と同じ様にして GNUPLOT を起動してください。

オンラインヘルプも見たいなら、ウィンドウをもう一つ開いて、そちらでも GNUPLOT を起動すればよいでしょう。

では、GNUPLOT のプロンプトに対して

```
load "gplot1.plt"
```

と打ってください。最後に描いたグラフが再現しましたね。試しに、

```
set xrange [-25:25]; replot
```

と打ってみましょう。

```
set nogrid; replot
```

というのもやってみてください。前の作業を継続できているのがわかるでしょう？

いま `set` コマンドと `replot` コマンドを一行に書いてしまいましたが、これについての説明はまだでした。実は、セミコロン ; で区切って複数のコマンドを一行に書いてもよいのです。ご想像のように、並べられたコマンドは左から順に実行されます。`set nogrid` は `set grid` の逆の働きをするコマンドです。（やってみればわかりますよね？）

12.4 数値データからグラフを描く

測定や調査結果のデータを GNUPLOT でグラフにするには、データを一旦ファイルに収めてからそれを GNUPLOT に処理させます。このファイルをデータファイルといいます。

データファイルは数値をただ並べただけのテキストファイルです。コンピュータで計算処理した結果を GNUPLOT でグラフにすることもできます。この場合は計算処理を行うプログラムは結果を数値として打ち出してデータファイルを作ることができればよいのです。

12.4.1 二次元データのプロット

x の値と y の値の組が複数並んだデータから作ったグラフを散布図といいます。各組みはデータ点と呼びます。データファイルには、1データ点あたり一組の x と y の値をこの順に 1 行に空白で分けて並べます。次のデータは 11 個のデータ点からなっています。

```
0.0    0.0
0.9    1.8
1.8    11.9
2.7    55.3
3.6    108.7
4.5    96.3
5.4    50.0
```

```
6.3    8.2
7.2    2.6
8.1    1.1
9.0    0.5
```

これをそのまま Mule のような テキストエディタ を使って打ち込み、ファイルに保存してください。GNUPLOT とは別のウィンドウで作業すればよいでしょう。ファイル名は `test1.dat` と付けてください。データを打ち込むときに、体裁を整えるために数値と数値の間の空白は適当に入れてかまいませんが、空行を入れるとグラフにしたときにその部分を線でつなぐことができなくなりますので注意してください。ファイルに保存したら、GNUPLOT で

```
plot "test1.dat"
```

と打ってください。マークだけのグラフが描けましたね。次に、

```
set data style linespoints
```

と打ってから `replot` で描きなおすと、マークの間に線が入ったでしょう。`set data style` の後に続けるパラメータにはほかにも、`lines` や `points` や `boxes` などがあります。これらをデータ・スタイルといいます。

`plot "test1.dat" with lines` というように、`plot` コマンドの中にデータ・スタイルを含めることもできます。`with` が必要です。この場合、指定したデータ・スタイルは以後の `plot` コマンドに影響を与えるわけではなく、その場限りの効果しか持ちません。

12.4.2 エラーバー付きの二次元プロット

次のデータは測定誤差を考慮し、 x 、 y の測定値、 y の下限、 y の上限を入れたものと思ってください。各データ点について4個の数値が入っていますが、一列目と二列目は前と同じです。

```
0.0    0.0    0.0    3.0
0.9    1.8    0.0    5.8
1.8    11.9   6.9    16.9
2.7    55.3   48.3   62.3
3.6    108.7  100.7  116.7
4.5    96.3   88.3   104.3
5.4    50.0   44.0   56.0
6.3    8.2    3.2    13.2
7.2    2.6    0.0    6.6
8.1    1.1    0.0    4.1
9.0    0.5    0.0    3.5
```

`test1.dat` を Mule で開いて書き直した後、同じ名前でも保存してください。

GNUPLOT で `plot "test1.dat"` とすれば前と同じグラフになり、データの3列目と4列目は使われません。3、4列目も使うために

```
plot "test1.dat" with errorbars
```

と打ってください。エラーバー（誤差の範囲を示す縦棒）がグラフに描き込まれたでしょう。`errorbars` もデータ・スタイルの仲間です。

12.4.3 列の選択など

少しトリックも紹介しておきましょう。

```
set data style points
plot "test1.dat" using 1:4
```

と打ってみてください。using の後の 1:4 はデータファイルの中の 1 列目を x 、4 列目を y として取り出すように指示しています。描かれたグラフを見ただけでそのことがわかるでしょう？ わからなければ、... 1:3 や ... 1:2 もやってみてください。

次に、

```
plot "test1.dat", "test1.dat" using 1:3, "test1.dat" using 1:4
```

と打ってみてください。plot コマンドではコンマ , で区切って複数のグラフを描くことができます。ただし、 x 軸と y 軸は共通です。つぎのように数式のグラフをいっしょに入れることもできます。

```
plot "test1.dat" with errorbars, 112 * exp(-(x - 4)/1.5)**2)
```

数式中の括弧は全部小括弧を使います⁵。

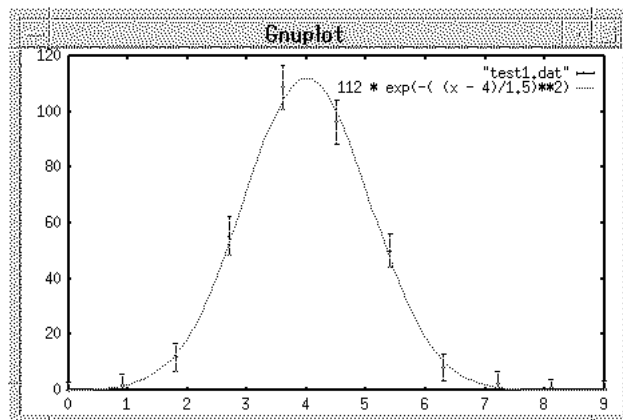


図 12.2 データと数式のグラフを一緒に描く

複数のグラフをいっしょに描く場合は x の範囲に注意してください。 x 軸の範囲が思ったようにならないときは、set xrange コマンドを使ってみればよいでしょう。

軸の目盛りを「対数目盛り」にすることもできます。

```
set logscale y
set yrange [0.1:200]
replot
```

とやってみてください。縦軸の目盛りが変わっていますね。対数目盛りでは、0 より大きい範囲を指定しなければならないことに注意してください。対数目盛では 100 と 10、10 と 1、1 と 0.1 は全部間隔が同じですから、縦軸の 0 の位置は無限に下方にあるのです。

では、

⁵exp は指数関数というもので、** はべき乗の演算子。

```
set nologscale
set autoscale y
```

と打って元に戻しておいてください。set autoscale y は、set yrange コマンドの効果を取り消します。

12.4.4 フィッティング

cc2000 の GNUPLOT には GNUFIT が組み込まれていて、フィッティングを行うことができます。GNUFIT は単純な計算法を使っているので、あらゆる場合に正しい答えが得られると期待してはいけません⁶。また、どのような数式でも正常に処理できるというわけには行かず、結果を見てうまくいったかどうかをチェックする必要があります。数式が未知パラメータに関して線形の場合は大抵うまくいくようですが、非線形の場合は駄目なことが多いようです。

ではちょっと試してみましょう。次のデータは、直線の方程式

$$y = ax + b$$

に従うはずの2つの量、 x と y を測定して得たものと思ってください。 a と b はある定数です。

1.6	4.8
1.0	3.8
1.8	4.8
2.9	6.7
0.2	2.8
0.5	3.3
0.5	3.1
2.5	6.3
3.3	7.6
3.4	7.6
2.2	5.7
2.4	6.2

これを test2.dat という名前のファイルに入れて、plot "test2.dat" でグラフを描いてみてください。データ点はだいたい一直線上に並んでいますね。この測定データから、秘密のパラメータ a と b を推定します。

まず、元の式を一発で呼び出せるようにするために、

$$f(x) = a * x + b$$

と打ちます。これで f という名前の関数を新たに定義して登録したことになります。このようなものを「ユーザー定義関数」といいます。show functions と打つと、ユーザー定義関数のリストを見ることができます。

次に、グラフを「目で見て」 a と b のごく大雑把な推定値を決め、 a と b に代入しておきます。 a は 1 程度、 b は 2 程度といったところでしょうから、

```
a = 1
b = 2
```

⁶GNUFIT は 最小二乗フィッティングを行う。これは異常データが部分的に含まれている場合にその影響を受けやすい。

と打ちます。そして、式とデータをグラフで比較するために

```
plot f(x), "test2.dat"
```

とやってみてください。だいぶん違いますねえ。

後はコンピュータで合わせてやりましょう。次のように打ってください。

```
fit f(x) "test2.dat" via a,b
```

GNUFIT は、 $f(x)$ とデータとの違いが最も小さくなるようなパラメータを探して計算を繰り返します。その途中経過が表示された後に最終の結果が出てきます。こんな具合です。

Final set of parameters		68.3% confidence interval
=====		=====
a	= 1.53444	- 0.0382937
b	= 2.37349	- 0.0820193

correlation matrix of the fit parameters:

	a	b
a	1.000	-0.868
b	-0.868	1.000

$a = 1.53444, b = 2.37349$ という結果が出ました。confidence interval というのは「信頼区間」といって、推定値がどれぐらいの不確定さを持っているかということをお知らせします。すべてのデータ点を一本の直線に乗せることはできないので、推定にはある程度の不確実性が伴うということはおわかりですね。correlation matrix については統計の理論を勉強してもらう必要がありますが、普通は無視してもよいでしょう⁷。

この段階で a と b の値は推定値に変わっています。show variables と打てば、 a や b のような「変数」とその値のリストを見ることができるようでしょう。あなたの知らない変数も出てくるかも知れませんが、それは無視してください。最後に、推定値に基づいた数式とデータをグラフにしてながめてください。前に一度やっていますから、replot と打つだけでOKです。

有用な結果が得られたら、save コマンドで GNUPLOT の状態を適当な名前のファイルに保存して置くのがよいでしょう。後でそうやって保存したものの中から使えるものを選んで、印刷などの処理に回すのです。

12.4.5 自作プログラムで GNUPLOT を利用する

自分で作るプログラムでグラフを出力したい場合に、描画処理を GNUPLOT に任せるようにすれば手間が省けます。グラフィックスの表示や印刷の処理は使用する端末やプリンタによって違ってきますが、GNUPLOT はその違いも吸収してくれますからあなたのプログラムは GNUPLOT が出力することのできたいへん多くの装置に出力できることとなります。

⁷推定計算の途中結果は fit.log という名前のテキストファイルとして残されます。

実際に試してみましょ。ごく単純な例を使いますので、UNIX やコンパイラの知識は必要ではありません。

漸化式

$$\begin{aligned}x_{i+1} &= -0.97 * x_i + y_i - 5 + 5/(1 + x_i^2) \\ y_{i+1} &= -0.995 * x_i\end{aligned}$$

と初期条件 $x_1 = 1, y_1 = 0$ で決まる x と y の値の組みの系列をグラフにしてみましょ。始めに $x = 1, y = 0$ という状態から出発してこの式で次の値を計算します。同じ計算を次々に繰り返して得られる x と y の列をグラフにするのです。10000 回ほど繰り返します。普通は FORTRAN などのプログラミング言語を使って計算するのですが、ここでは、AWK というのを使ってみましょ。これは C 言語に似ています。

```
BEGIN{ x = 1; y = 0; i = 10000;
      while( i-- > 0 )
      {   print x, y;
          nextx = -0.97 * x + y - 5 + 5 / (1 + x * x);
          y = -0.995 * x;
          x = nextx;
      }
}
```

テキストエディタでこれをそのまま打ち込んで `test3.awk` という名前のファイルに保存してください。括弧の種類と、コンマやセミコロンといった区切り記号に注意してください。短いので簡単ですね？

`test3.awk` ができたら、UNIX のプロンプトに対して

```
gawk -f test3.awk >test3.dat
```

と打ってください。これで AWK プログラムが走り `test3.dat` というファイルに計算結果が保存されます⁸。便りのないのはなんとやらで、プログラムに間違いがなければ何も表示されずに作業が終了します。念のために `test3.dat` の内容を見てみてください。

```
head test3.dat
```

で始めの数行が表示されます。

もし FORTRAN か Pascal を使うなら、図 12.3 や 図 12.4 のようなプログラムを書けばよいでしょう。`f77 test3.for` などとしてコンパイルした後、`a.out >test3.dat` で実行します⁹。

では、GNUPLOT でグラフにしてみましょ。GNUPLOT のプロンプトに対して

```
set data style dots
plot "test3.dat"
```

と打ってください。dots というデータスタイルは、データ点を小さい点でプロットするものですが、小さ過ぎたら `points` を使ってください。最後に前に説明したようにして印刷ましょ。何か芸術的なタイトルを入れるのもよいでしょう。

⁸ `>test3.dat` は、プログラムの「標準出力」を端末ではなくファイル `test3.dat` に書き込むように指示している。

⁹ cc2000 では、FORTRAN は `f77` コマンド、Pascal は `pc` コマンドでコンパイルできる。特に指定をしない場合、実行ファイルは `a.out` という名前になる。

```

REAL x, y, nextx
INTEGER i
x = 1
y = 0
DO i = 1, 10000
  WRITE(*,*) x, y
  nextx = -0.97 * x + y - 5 + 5 / (1 + x * x)
  y = -0.995 * x
  x = nextx
END DO
STOP
END

```

図 12.3 FORTRAN プログラムの例

```

program test3(input, output);
var
  x, y, nextx : real;
  i : integer;
begin
  x := 1; y := 0;
  for i := 1 to 10000 do
    begin
      writeln(x:8:2, y:8:2);
      nextx := -0.97 * x + y - 5 + 5 / (1 + x * x);
      y := -0.995 * x;
      x := nextx;
    end
  end.
end.

```

図 12.4 Pascal プログラムの例

12.5 マニュアル類について

ここでは、cc 環境から直接得ることのできる GNUPLOT のマニュアルなどの情報について説明します。

12.5.1 マニュアルを印刷する

印刷したマニュアルがほしい場合は、cc 環境にある次のファイルを自分のところにコピーしておいて後で印刷してください。やりかたについては UNIX ガイドの \LaTeX の項を参照するか、親切な友達にやってもらってください。

ディレクトリ	ファイル名
/NF/local/general/lib/gnuplot/	gnuplot-3.5.tex
	gnuplot-3.2j.tex
	gpcard.tex
	tutorial.dvi

`gnuplot-3.2j.tex` はオンラインマニュアルの日本語訳です。cc2000 では `jlatex` コマンドを使ってコンパイルします。コンパイルには、同じディレクトリにある `toc.entr.sty` と `titlepage.tex` が必要です。`gnuplot-3.5.tex` は英語版です。これらの内容はオンラインマニュアルと同じですが、全体を通して読むことができる点の違いがあります。

`gpcard.tex` はリファレンスカード (英語) で、要領良くまとめてあるので役に立ちます。これは `plainTeX` 用で、cc2000 では `tex` コマンドでコンパイルできます。このファイルの始めのほうにある `\columnspanpage=2` という行の数字を 2 ではなく 1 に直してから `TeX` にかけて印刷してください。2 のままだと、用紙からはみ出すおそれがあります。後で貼り合わせて縮小コピーすれば、りっぱなリファレンスカードになるでしょう。

GNUPLOT には `GnuTeX` というプログラムが組み込まれていて、その機能を使って \LaTeX の文書の中に GNUPLOT で描いたグラフを入れることができます。`GnuTeX` の特徴は、グラフ中のタイトルや目盛りラベルなどの文字に本文中と同じく `TeX` のフォントが使われることで、 \LaTeX の記法を使って数式やギリシャ文字、記号などを入れることができます。`tutorial.dvi` は、`GnuTeX` の使い方を実例をたくさん使って解説しています。これはコンパイル済みの `TeX` ファイルですから、そのまま `jdvi2ps` と `lpr` コマンドを使って印刷できます¹⁰。

なお、GNUFIT の使い方を簡単に説明したファイルも 同じディレクトリに入っています。名前は、`gnufit.doc` です。これは、Mule などで表示して読むことができますし、直接 `lpr` コマンドでプリンタに送って印刷することができます。

12.5.2 オンラインヘルプを日本語にする

オンラインヘルプが当面英語でもかまわない方や、UNIX のコマンドに慣れていない方はここは読み飛ばしてください。必要になってから読めばよいでしょう。

先程のマニュアル類の入っているディレクトリに `gnuplot.gih` というファイルと `gnuplot.gih.j` というファイルが入っています。前者は英語のオンラインマニュアルの「種になっている」ファイルです。しかし、次のようにして後者を「種」にすればオンラインマニュアルは日本語になります。

¹⁰ `TeX` の DVI ファイル (`.dvi` が付くファイル) は、`xdvi` コマンドで X 端末に表示できる。大きな画面なら十分きれいに表示でき操作も簡単なので、印刷したマニュアルのかわりとして使ってもよい。

空いているウィンドウで作業しましょう。まず `cd` と打って自分のホームディレクトリに移ってください。ここにある `.cshrc` というファイル（隠しファイルなので通常は気が付かない）を編集しますが、うまくいかなかったらすぐ元に戻せるように複製を作ります。

```
cp .cshrc cshrc.gplot
```

これで、`cshrc.gplot` というファイルに `.cshrc` の複製ができます。オンラインヘルプの日本語化がもしうまくいかなければ、後で

```
cp cshrc.gplot .cshrc
```

と打てば元に戻せます。

複製ができたなら Mule などのテキストエディタで `.cshrc` を開いてください。たとえば、

```
mule .cshrc
```

でOKです。`.cshrc` のいちばん下の行の次に一行追加して

```
setenv GNUHELP '/NF/local/Solaris2J/lib/gnuplot/gnuplot.gih.j'
```

と書いてください。クォーテーションマークは両方ともシングルクォート ' です。行の最後には必ず改行を入れてください。書けたら保存してエディタを終了してください。念のために、`cat .cshrc` と打って `.cshrc` を表示し、最後の行をよく見てください。

うまくいったかどうかを確かめるには、いったん `cc2000` からログアウトした後、再び `cc2000` にログインして `GNUPLOT` を起動し、`help` と打ってください。

12.5.3 デモンストレーション・ファイル

`GNUPLOT` は今まで説明したよりも、もっとももっとたくさんの機能を持っています。あなたが必要とする機能を探すために、デモンストレーション・ファイルを利用してください。デモンストレーション・ファイルは、ディレクトリ `/NF/local/general/lib/gnuplot/demo` に入っています。自分のところにコピーして使ってください。

たくさんのファイルの内どれが必要かはわかりにくいので、とりあえず全部のファイルをコピーしましょう。デモンストレーション・ファイルを収めるサブディレクトリを作って、そこにコピーするのがよいでしょう。たとえば、シェルプロンプトに対して、`mkdir gplotdemo` と打って、サブディレクトリ `gplotdemo` を作ってください。それから `cd gplotdemo` と打ってそこをカレントディレクトリにします。そうしてから、注意して、

```
cp /NF/local/general/lib/gnuplot/demo/* .
```

と打てば、すべてのファイルが手に入ります。最後のピリオド `.` はカレントディレクトリをあらわす記号で、必ず必要です。

`ls` コマンドでファイルのリストを見てください。ファイル名の最後が `.dem` となっているのが、`GNUPLOT` のマクロファイルで、`GNUPLOT` のコマンドが説明と共に入っています。Mule などでのぞいてみれば様子がわかるでしょう。説明文には各行の左端に `#` が付いています。`#` はその行の残りの部分を全部無視するように指示する `GNUPLOT` の命令です。

これらのマクロファイルは `GNUPLOT` の `load` コマンドを使って自動実行できるようになっています。一つのマクロファイルでたくさんのグラフを次々に表示しますが、次のグラフに切り替えるには `リターン・キー` を押します。マクロファイルの中に `pause -1` というコマンドがところどころ入っていて、これで自動実行を一旦止めてキー入力を待つようになっています。

これらのマクロファイルを実行するには、カレントディレクトリがデモンストレーション・ファイルのサブディレクトリになっていなければなりません。GNUPLOT を起動する前に `cd` コマンドでこのディレクトリに移っておけばよいでしょう。GNUPLOT のコマンドにも `cd` がありますが、ディレクトリ名をダブルクォートで囲って指定しなければなりません。GNUPLOT の `pwd` コマンドはシェルのコマンドの `pwd` と同じで、カレントディレクトリを表示します。

図 12.5 は、デモンストレーション・ファイル `world.dem` を元にして作った立体画像です。地図データから GNUPLOT で生成した図を `tgif` で合成した後、日本語 \LaTeX で書いたこの文章に取り込みました。目を回さないようにお楽しみください。

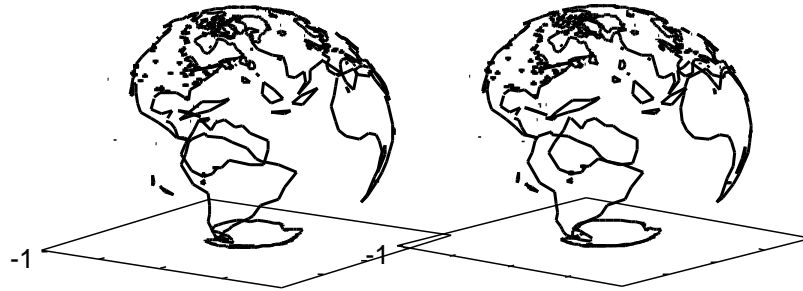
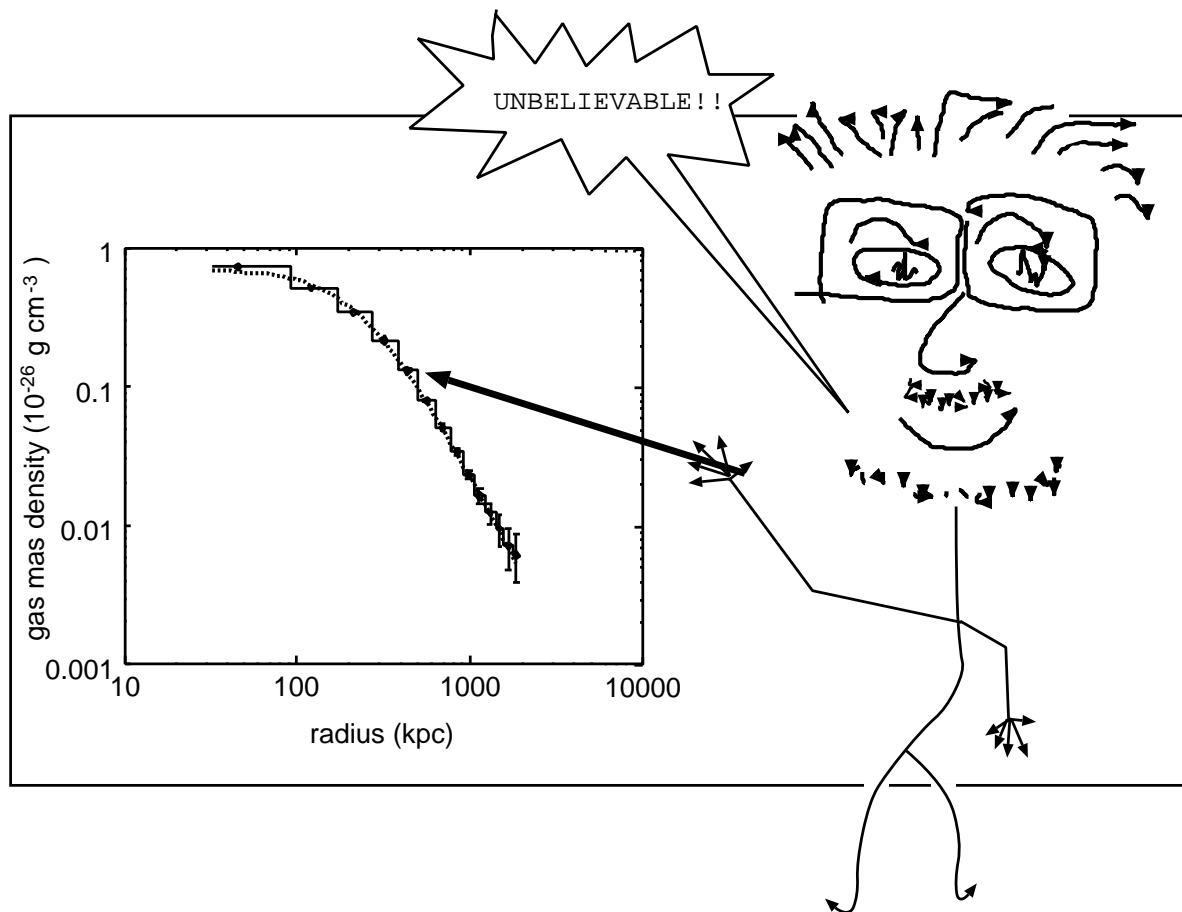


図 12.5 立体画像

第 13 章

tgif



tgif は X 上で作動する対話的な作図ツールです。tgif のようなグラフィカルユーザーインターフェースのアプリケーションは「習うより慣れよ」「百聞は一見にしかず」ですので適当に使ってみれば大体のことは出来てしまうことでしょう。

ではでは

matsu.

で終わってしまうわけにもいきませんので(というお約束のボケは置いといて)、真面目に説明を始めましょう:-)。ちなみにタイトル画(落書ともいふ)は全て tgif で作画したものです。cc 環境にはオリジナル版 tgif(tgif) と日本語化版 tgif(tgifj) がインストールされています。ここでは基本的にオリジナル版について説明していきます。オリジナル版と日本語化版の主な違いについては 13.2 で説明します。

13.1 tgif のできる事できない事

できる事

- テキスト, 四角, 楕円, ポリライン, フリーライン, ポリゴン, 円弧, 角丸四角形という多彩な図形を描画できます。
- 図形の拡大縮小ができます。
- XBM, XPM, EPS 形式の図形をファイルから読み込むことができます。
- 日本語化版 tgif(tgifj) では日本語入力が可能です。将来的には kinput2 による interactive な入力ができるようにする予定ですが、現状では X window の drag and paste を使って入力します。Mule からの利用がお薦めです。
- 作成した図形は tgif の専用形式以外にも XBM, PS, EPS, EPSI, TEXT 形式で出力することができます。
- 11 色(白黒含む) カラーをサポートしています。
- 階層的な図形単位管理ができます(が、私は使ったことがありませんので今回は図 13.12 で機能の一部を紹介するだけで説明はしません)。

できない事

- 90 度以外の図形の回転は出来ません。
- ファイルから読み込んだ PostScript 形式の図形の画面上での表示は出来ません。
- 日本語化版とオリジナル版の間の図形ファイルには互換性はないと思った方が良いでしょう。最初にどちらかを使わないことにしてしまうのも一つの手です。

13.2 日本語化版 tgif(tgifj) のオリジナル版 tgif との違い

オリジナル版がバージョン 2.15 パッチレベル 6 なのに対して日本語化版はバージョン 2.13 をベースに日本語化されていますので、2.15 にはあって 2.13 には無い機能が山ほどあります。目立つ所では日本語化版には **Status Window** と **Menubar Window**(図 13.2) がありません。オリジナル版 tgif で作ったファイルを日本語化版 tgif で読み込むことは出来ません。逆に日本語化版で作ったファイルは、日本語が使われている場合にはオリジナル版で読み込むことは出来ません。日本語化版 tgif(tgifj) で日本語を入力する方法は将来的には kinput2 から interactive に入力できるようにする予定ですが、現状では **X window** の **drag and paste** を使って入力します。Mule 上で日本語のテキストを書いて必要な部分をドラッグして反転させておいて、あとは tgifj で **Edit menu(13.4) の SelectionPaste** コマンドを選択します。コマンドの選択方法については 13.4 を参照して下さい。すると、四角い枠がマウスポインターについてくるので適当な場所でマウスの任意のボタンを押します。後は普通の図形単位と同じように扱えます。

図 13.1 tgif による日本語出力の例

オリジナル版がバージョン2.15パッチレベル6なのに対して日本語化版はバージョン2.13をベースに日本語化されていますので、2.15にはあって2.13には無い機能が山ほどあります。目立つ所では日本語化版にはStatus Window と Menubar Windowがありません。

オリジナル版がバージョン2.15パッチレベル6なのに対して日本語化版はバージョン2.13をベースに日本語化されていますので、2.15にはあって2.13には無い機能が山ほどあります。目立つ所では日本語化版にもStatus Window と Menubar Windowがありません。

13.3 tgif の起動と Window の名称

tgif を起動するにはターミナルから
オリジナル版

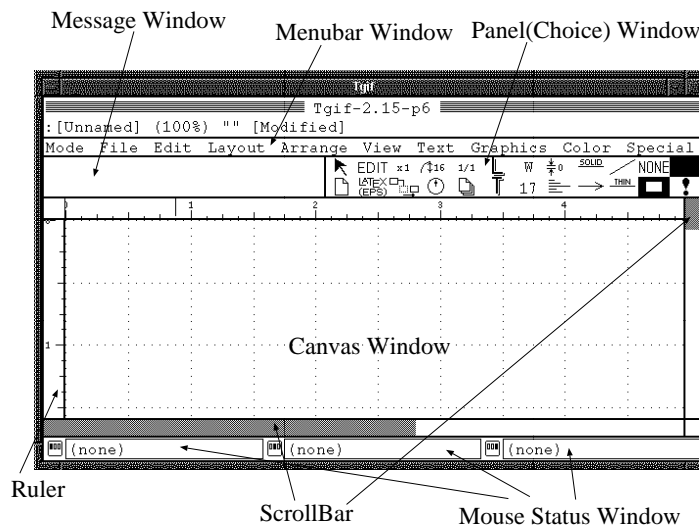
tgif [オプション] [ファイル名 (拡張子は obj で省略可)]

日本語化版

tgifj [オプション] [ファイル名 (拡張子は obj で省略可)]

と入力します。基本的にはオプションやファイル名を付ける必要はないでしょう。tgif で作図された図形は拡張子が “obj” のテキストファイルとして保存されます。tgif が起動すると図 13.2 の様なウインドウが現れます。以下では各 Window を図中の名称で呼ぶことにします。

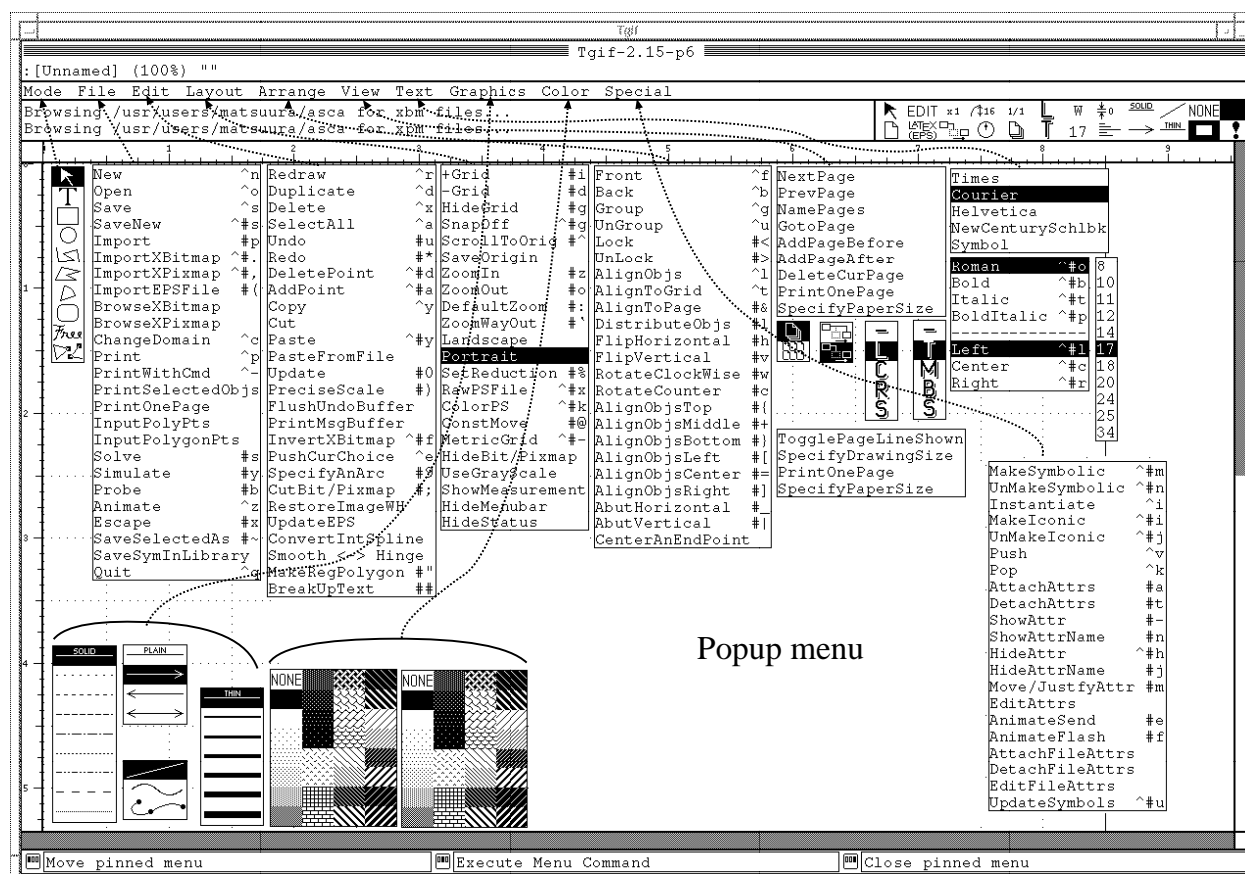
図 13.2 tgif の各 Window の名称



13.4 Popup menu

Menubar Window(図 13.2参照)の項目一覧(Main Menu)の所でマウスの任意のボタンを押すことで Popup Menu(図 13.3参照)が表示されます。その状態で上下にドラッグして Popup Menu に表示されている項目一覧(Sub Menu)の中の目的のコマンドの位置でボタンを離すことでそのコマンドを選択することができます。この時、ボタンを離す前に左右に大きくドラッグすることで表示されている Popup Menu を常時表示させる(Popup Menuの張り付け)ことができます。張り付けた Popup Menu の中のコマンドの選択はマウスの中ボタンで、移動させるには左ボタンでドラッグし、不要になれば右でクリックして消します。Canvas Window(図 13.2参照)でマウスの中ボタンを押すことでも Main Menu の Popup Menu が現れます。ここで項目を選ぶと、マウスポインターが \blacksquare に変わり、そこでもう一度 Canvas Window でマウスのボタンを押すことで先に選んだ項目についての Sub Menu の Popuo Window が現れます。Panel Window(図 13.2参照)内の各 Panel をマウスの左右のボタンでクリックすることでそれぞれの項目を変更することも出来ます。Panel 上でマウスの中ボタンを押しても各 Sub Menu の Popup Menu が現れます。ある状態でマウスの各ボタンにどんな操作が当てられているのかを覚えるのは最初のうちはややこしいものです。そういう人のために Mouse Status Window(図 13.2参照)にはその状態でマウスの各ボタンに割り振られている機能が表示されています。

図 13.3 Sub Menu の Popup menu 一覧



作図の手順は次のようになります。 Mode menu(図 13.4参照)で文字を入力するモードか図形を描くモー

ド(長方形, 円, ポリライン, ポリゴン, 円弧, 角丸長方形, フリーライン)を選択します。作図は **Canvas Window** 上で主にマウスの左ボタンを用いて行ないます。各図形の具体的な作図方法についてはそれぞれ冒頭でも言ったように「百聞は一見にしかず」ですので、マウスを動かしながら左ボタンを押したりして適当に使ってみれば分かります。

作図が終わったら、図形を選択するモードに設定して図形を選択します。1つの図形を選択するにはその図形をマウスの左ボタンでクリックします。すると図形の接点に四角い点が重ねて表示されます。これを以後操作点と呼ぶことにします。複数の図形を選択するにはキーボードの<Shift>キーを押えながら図形をマウスの左ボタンで次々とクリックしていきます。すでに選択されている図形をもう一度キーボードの<Shift>キーを押えながらマウスの左ボタンでクリックすることで選択されていない状態に戻すことができます。他にも、図形の無い部分から左ボタンによるドラッグを行なうことで、その始点と終点を対角線とする長方形の中に含まれる全ての図形を選択することが出来ます。以上のようにして選択された図形に対して操作点で左ボタンをドラッグすることで図形のサイズを変えることが出来、操作点以外の部分でドラッグすることでその図形を移動させることが出来ます。また、**Panel Window** や **Popup Window** のコマンドを使って選択されている図形に対して様々な加工を施すことが出来ます。

作画と加工が終わって File-Save コマンド (図 13.5参照) で保存したら、最後に図形の出力を行ないます。tgif はプリンターに直接出力する以外にも、XBM, XPM, PS, EPS, EPSI, TEXT 形式でファイルへ出力することができます。そこで、出力のコマンドを実行する前に出力形式の選択を行ないます。出力形式の変更は **Panel Window** の下の段の左から 2 番目の **Panel** の所でマウスの左右のボタンをクリックするか、Arrange menu(図 13.8参照) の下から 9 番目の項目を選択することで行ないます。出力形式を決めた後、File menu の Print コマンド (図 13.5参照) を選択することで図形の出力が実行されます。

以下に私が把握している Sub Menu のコマンド一覧を示しておきますので tgif 利用時の助けとして下さい。

図 13.4 Mode:描画モード

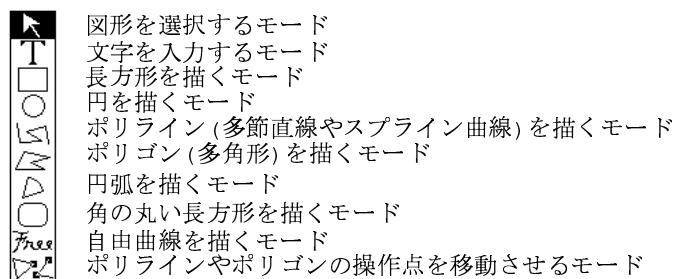


図 13.5 File:図形ファイルの入出力

New	^n	新しくobjファイルを開く
Open	^o	既存のobjファイルを開く
Save	^s	現在の図形をobjファイルに保存
SaveNew	^#s	現在の図形を新しくobjファイルに保存
Import	#p	objファイルを読み込む
ImportXBitmap	^#.x	xbmファイルを読み込む
ImportXPixmap	^#.x	xpmファイルを読み込む
ImportEPSFile	#{	epsファイルを読み込む
BrowseXBitmap		カレントディレクトリ内の全xbmファイルを読み込む
BrowseXPixmap		カレントディレクトリ内の全xpmファイルを読み込む
ChangeDomain	^c	
Print	^p	指定されているデバイスへ出力
PrintWithCmd	^-	commandを指定してPrint
PrintSelectedObjs		選択されている図形をPrint
PrintOnePage		このページをPrint
InputPolyPts		
InputPolygonPts		
Solve	#s	
Simulate	#y	
Probe	#b	
Animate	^z	
Escape	#x	
SaveSelectedAs	#~	選択されている図形を新しくobjファイルに保存
SaveSymInLibrary		
Quit	^q	終了

図 13.6 Edit:図形の編集

Redraw	^r	図形の再描画
Duplicate	^d	選択されている図形の複製
Delete	^x	選択されている図形の消去
SelectAll	^a	全ての図形を選択
Undo	#u	過去の操作の取消
Redo	#*	Undoの取消
DeletePoint	^#d	選択されている図形の制御点を消去
AddPoint	^#a	選択されている図形に制御点を付加
Copy	^y	選択されている図形をカットバッファにコピー
Cut		選択されている図形をカットバッファに移動
Paste	^#y	カットバッファの図形を読み込む
PasteFromFile		テキストファイルを読み込んで文字列として読み込む
Update	#0	
PreciseScale	#)	選択されている図形のサイズを数値入力に変更する
FlushUndoBuffer		Undoバッファを消去
PrintMsgBuffer		
InvertXBitmap	^#f	選択されているビットマップを白黒反転
PushCurChoice	^e	
SpecifyAnArc	#9	円弧を半径と二つの角度を入力する方法で描画
CutBit/Pixmap	#;	選択されているxbm/xpmから入力された幅と高さの領域の外側を刈り取る
RestoreImageWH		
UpdateEPS		
ConvertIntSpline		
Smooth <-> Hinge		
MakeRegPolygon	#"	
BreakUpText	##	

図 13.7 Layout:編集環境の変更

+Grid #i	+Grid #i	グリッド間隔を増やす
-Grid #d	-Grid #d	グリッド間隔を減らす
HideGrid #g	ShowGrid #g	グリッドを表示しない/表示する
SnapOff ^#g	SnapOn ^#g	グリッド単位で移動/グリッドによらず移動
ScrollToOrig #^	ScrollToOrig #^	
SaveOrigin	SaveOrigin	
ZoomIn #z	ZoomIn #z	拡大率を上げる
ZoomOut #o	ZoomOut #o	拡大率を下げる
DefaultZoom #:	DefaultZoom #:	拡大率をデフォルトに戻す
ZoomWayOut #'	ZoomWayOut #'	全体が表示される拡大率まで下げる
Landscape	Landscape	キャンバスを横長にする
Portrait	Portrait	キャンバスを縦長にする
SetReduction #%	SetReduction #%	
RawPSFile ^#x	LaTeXFig ^#x	Print の出力先を指定
ColorPS ^#k	BlkWhtPS ^#k	カラーPS/白黒PS
ConstMove #@	UnConstMove #@	
MetricGrid ^#-	EnglishGrid ^#-	グリッドの単位をcmにする/inchにする
HideBit/Pixmap	ShowBit/Pixmap	XBitmapXPixmapの内容を表示しない/表示す
UseGrayScale	NoGrayScale	
ShowMeasurement	HideMeasurement	ポインタの位置座標を表示する/表示しない
HideMenubar	ShowMenubar	コマンドメニューバーの欄を表示しない/表示する
HideStatus	ShowStatus	マウスステータスの欄を表示しない/表示する

図 13.8 Arrange:図形の整理整頓

Front ^f	選択されている図形を前に移す
Back ^b	選択されている図形を後ろに移す
Group ^g	選択されている複数の図形をグループ化する
UnGroup ^u	選択されている図形のグループ化を解く
Lock #<	選択されている図形を固定する
UnLock #>	選択されている図形の固定を解く
AlignObjs ^l	選択されている複数の図形の左上の位置を揃える
AlignToGrid ^t	選択されている図形の位置をグリッドに合わせる (SnapがOnの時のみ)
AlignToPage #&	選択されている図形の左上の位置をページの左上に置く
DistributeObjs #l	
FlipHorizontal #h	選択されている図形を水平方向に反転させる
FlipVertical #v	選択されている図形を垂直方向に反転させる
RotateClockWise #w	選択されている図形を時計回りに90度単位で回転させる
RotateCounter #c	選択されている図形を反時計回りに90度単位で回転させる
AlignObjsTop #{	選択されている複数の図形の上の位置を揃える
AlignObjsMiddle #+	選択されている複数の図形の上下の中央の位置を揃える
AlignObjsBottom #}	選択されている複数の図形の下の位置を揃える
AlignObjsLeft #[選択されている複数の図形の左の位置を揃える
AlignObjsCenter #=	選択されている複数の図形の左右の中央の位置を揃える
AlignObjsRight #]	選択されている複数の図形の右の位置を揃える
AbutHorizontal #_	選択されている複数の図形の水平方向の境を合わせる
AbutVertical #	選択されている複数の図形の垂直方向の境を合わせる
CenterAnEndPoint	選択されているポリラインの端を選択されている図形の中心に移動させる

図 13.9 View.Page.PageLayout.HoriArign.VertiArign.MoveMode:ページ管理

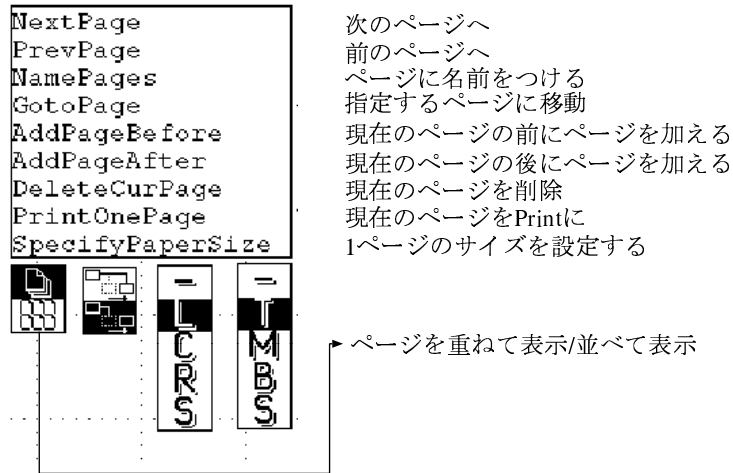


図 13.10 Text:Font.TextStyle.TextSize:文字種の変更

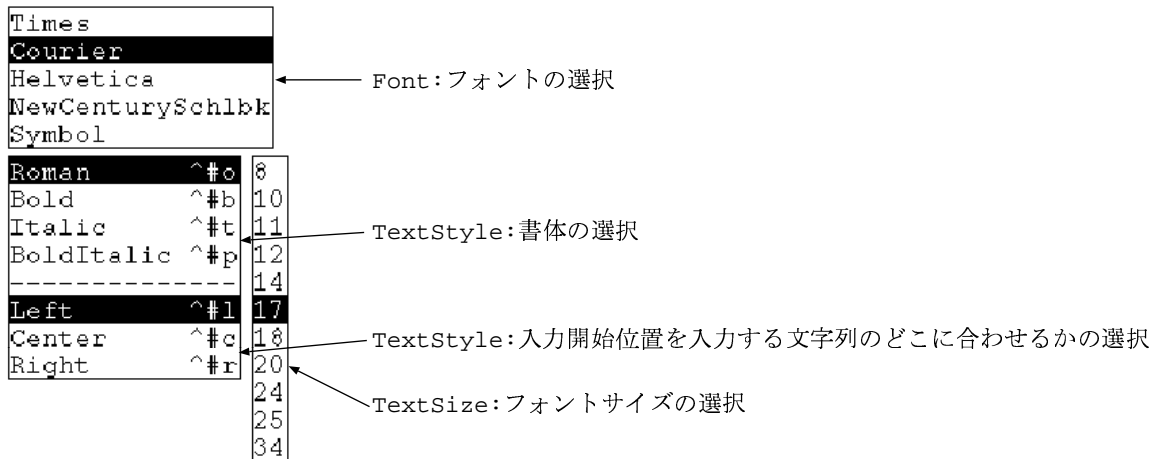


図 13.11 Graphics:LineDash.LineStyle.LineType.LineWidth.Fill.Pen: 線やのパターンの変更

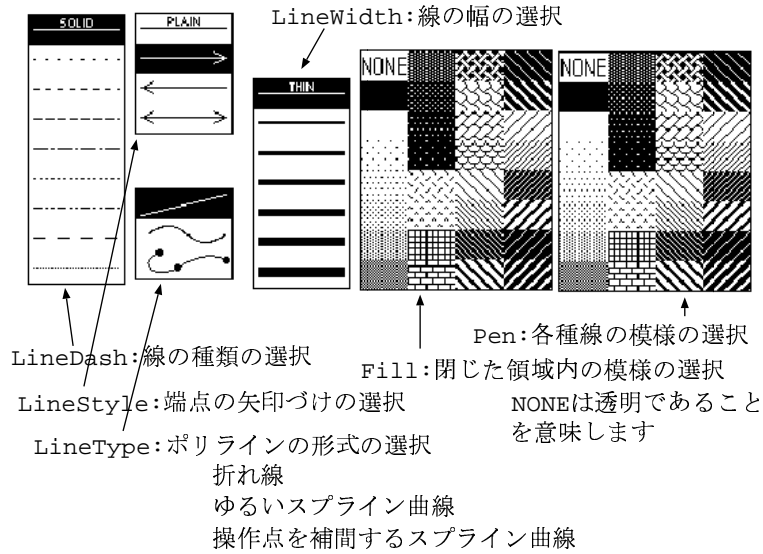


図 13.12 Special:その他の機能

MakeSymbolic	^#m	選択されているグループ化された図形をシンボル化
UnMakeSymbolic	^#n	シンボル化の解除
Instantiate	^i	シンボル化して保存されている図形の挿入
MakeIconic	^#i	選択されているグループ化された図形を部品としてアイコン化
UnMakeIconic	^#j	アイコン化の解除
Push	^v	部品の下位図形の編集
Pop	^k	部品の上位図形の編集
AttachAttrs	#a	
DetachAttrs	#t	
ShowAttr	#-	
ShowAttrName	#n	
HideAttr	^#h	
HideAttrName	#j	
Move/JustfyAttr	#m	
EditAttrs		
AnimateSend	#e	
AnimateFlash	#f	
AttachFileAttrs		
DetachFileAttrs		
EditFileAttrs		
UpdateSymbols	^#u	シンボル化している図形の更新

13.5 tgif 関連のツール

- **pstotgif** pstotgif は単純な線画からなる PostScript フォーマットを tgif フォーマットに変換します。使い方はコマンドの引数に PS ファイルを付けるだけです。

```
cc2000% pstotgif foo.ps
Tgif file written to foo.ps.obj
cc2000% tgif foo.ps.obj
```

成功率はあまり高くないようです。私は PostScript の事は分らないので、具体的にどのような PostScript ファイルが通ってどのようなのが通らないかは分かりません。

- **gnuplot** gnuplot はグラフの出力フォーマットとして tgif フォーマットをサポートしています。以下に使用例を示します。個々の gnuplot の説明については第 12 章を参照して下さい。

```
cc2000% gnuplot
```

```

G N U P L O T
unix version 3.5
patchlevel 3.50.1.17, 27 Aug 93
last modified Fri Aug 27 05:21:33 GMT 1993

*** GNUFIT 1.2                      22 Nov 93 ***
*** Nonlinear least squares fit added by C. Grammes ***
*** Send bugs regarding fit to cagr@rz.uni-sb.de ***

Copyright(C) 1986 - 1993  Thomas Williams, Colin Kelley

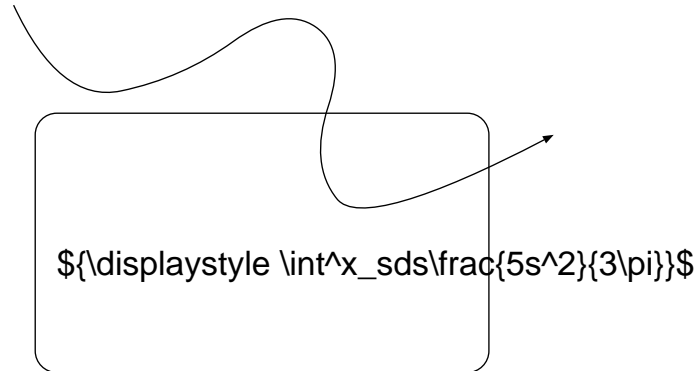
Send bugs and comments to bug-gnuplot@dartmouth.edu

Send bugs, suggestions and mods to bug-gnuplot@dartmouth.edu
```

```
Terminal type set to 'x11'
gnuplot> plot'foo.dat'
gnuplot> set xlabel'radius (cm)'
gnuplot> set ylabel'weight (g)'
gnuplot> replot
gnuplot> set term tgif
Terminal type set to 'tgif'
gnuplot> set output'foo.obj' <----- グラフの出力先のファイル名
gnuplot> replot                               を入力します。拡張子は''obj''
gnuplot> quit
cc2000% ls
... foo.obj ..
cc2000% tgif foo.obj &
...
```


- **tgif2tps tgif2tex** tgif では書けないけれど \LaTeX では書けるような文字列を図中に入れたい時には, tgif の文字入力モードで \LaTeX のコマンドを使って文字列を入力してから, この tgif2tps や tgif2tex を使って \LaTeX の出力に変換します. 例えば tgif で編集中の図形 (以降 foo.obj というファイル名だとします.) の中に $\int_s^x ds \frac{5s^2}{3\pi}$ という文字列を入れたいとします. その場合, 図 13.13 のように作図します.

図 13.13 tgif での \LaTeX コマンドの使用



これを EPS 形式で出力します (出力の仕方は 13.4 参照). そうして得られた EPS ファイル (foo.eps) を tgif2tps に掛けることで foo.tps という \LaTeX のソースファイルと foo.dps という EPS ファイルが作られます. 以下が tgif2tps の実行の様子です.

```
cc2000% tgif2tps foo.eps
Tgif2tps Version 1.10
Copyright (C) 1994, Koji Nakano (nakano@harl.hitachi.co.jp)
foo.tps and foo.dps are generated with
mag=1
cc2000% ls
... foo.dps foo.tps ...
```

foo.dps はもとの foo.eps の中から文字列の部分を取り去ったもので, もう一つのファイル foo.tps の中身は

```
\unitlength 1pt
\begin{picture}(388,192)
\put(0,0){\epsfile{file=foo.dps,scale=1}}
\put(128.2,111.8){\makebox(0,0)[lb]{\mbox{\LARGE ${\displaystyle \int^x_s ds \frac{5s^2}{3\pi}}$}}}
\end{picture}
```

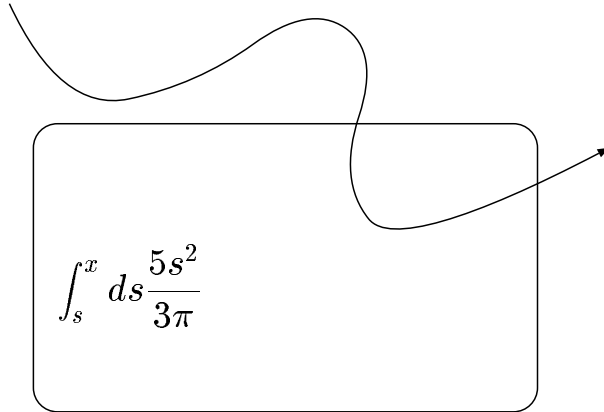
のように抜き取った文字列と foo.dps を適切な位置に記述しているだけです. あとは次のように, この foo.tps を \LaTeX のソースファイル (foo.tex) の中で呼び出します.

```
\documentstyle[epsbox]{jarticle}
```

```
\begin{document}
\input{foo.tps}
\end{document}
```

最後に `foo.tex` を `jlatex` に通せば出来上がり.

図 13.14 `tgif2tps` の出力例



13.6 終りに

私自身も完全に `tgif` の使い方を把握している訳ではありませんのでこの章でも `tgif` の機能の一部しか紹介できていません. 他の `tgif` のドキュメントとしては, `tgif2.2.4` をベースに日本語化された `tgif+` という `tgif` の仲間についてその作者が書いたかなり親切丁寧なものがあります. それを `/NF/local/general/doc/tgif+.ps` として置いてあります. 私達の使うバージョンとかなり開きがあるので細かい所は色々異なりますが使い始めの頃はとても参考になると思います. また, `UNIXmagazine` の 93 年 1 月号と 3 月号にやはり `tgif+` の使い方の記事が載っていてこちらも参考にされると良いかも知れません. あとの細かいところはオンラインマニュアルを参考にしてください.

第 14 章

XV

`xv`を使うと、画像データを表示させる事が出来ます。cc環境では画面印刷に欠かせない道具でもありません。ある種の画像フォーマット¹から別のフォーマットへと変換する事も出来ます。`xv`は X ウィンドウ環境で働きますので、利用するためには X 環境が必要です。具体的には cc 環境では 21 情報処理教室の `csosf01` ~ `40` に `login` するか、C1 情報処理教室などの X 端末ソフトを用意しているコンピュータから `cc2000` に `login` する必要があるでしょう。

この章では `xv` の簡単な使い方を説明します。

14.1 はじめに

画像データの作成、変換といっても具体的にその必要性が思い浮かばないかも知れません。例えば何かの画像を \LaTeX の文中に張り付けたい時があったとしましょう。以下のような具合です。



図 14.1 意味もなく現れるユカリさん

このようなふざけた例ばかりでなく、`GNUPLOT`²で作成したグラフや `tgif`³や `xpaint`⁴で作成した絵なども画像のうちの一つです。さあ、あなたも画像データを扱いたくなってきたでしょう！

¹フォーマットの内容についてはここでは説明しません。

²`GNUPLOT`については第 12 章を参照して下さい。

³`tgif`については第 13 章を参照して下さい。

⁴`xpaint`については第 15 章を参照して下さい。

14.2 まずマニアックな人のために

最初に画像関係の事情をよく判っている人のための機能説明を専門用語でさささとやってしまいます。画像ファイルのフォーマットなどと聞いて良く判らない人は次の章まで読み飛ばして下さい。

xv は以下の画像フォーマットを扱う事が出来ます。

GIF	一般的に流通していますね。256色までしか表現できないのが今となっては辛い。
PM	良く知りません。
PBM	Portable Bitmap. 良く知りません。
X11 Bitmap	モノクロ二値画像のみです。X環境では <code>bitmap</code> コマンドで扱えます。
Sun Rasterfile	Sun Microsystems 社が規定したほとんどベタのフォーマットです。たしかモノクロだけが規定されていたような、、、
BMP	Microsoft Windows Bitmap. Windows パソコン標準の画像フォーマットです。
JPEG	写真などを綺麗に圧縮するにはこれが最高。互換性も高いですね。
TIFF	これもまた一般的に流通しています。NeXT は拡張して色数 256 以上に増やしています。機種依存が激しく互換性に乏しいフォーマットですね。
PGM	Portable Graymap. 同じく良く知りません。
PPM	Portable Pixmap. またまた良く知りません。
RLE	良く知りません。
RGB	Red Green Blue... でしょうね、、良く知りません。
PCX	良く知りません。
IRIS	良く知りません。

プリンタなどで流通している PostScript フォーマットを、*xv* は各種画像の出力フォーマットとしてだけサポートします。PostScript 画像を表示する事は出来ません⁵。

14.3 基本操作

まず基本操作を習得しましょう。

14.3.1 *xv* の起動と終了

xv を起動するには X 環境が利用できないといけません。cc 環境であればあなたは既に 21 情報処理教室の `csosf01~40` に login しているか、C1 情報処理教室などの X 端末ソフトを用意しているコンピュータから `cc2000` に login していますね。そこでターミナル (cc 環境であれば `Kterm` です) を一つ表示させて、そこにはコマンドプロンプトが表示されている事を確認して下さい。そこから以下のように `xv &<return>` で *xv* を起動します。以下の例ではコマンドプロンプトは `cc2000` マシンのものになっています。

```
cc2000(199)% xv &
[7] 23547
cc2000(200)%
```

すると以下のようなウィンドウが一つ現れると思います。

⁵PostScript 画像を表示するには `Ghostview` アプリケーションを利用して下さい。但し cc 環境では `Ghostview` が使えるのは `cc2000` だけです。

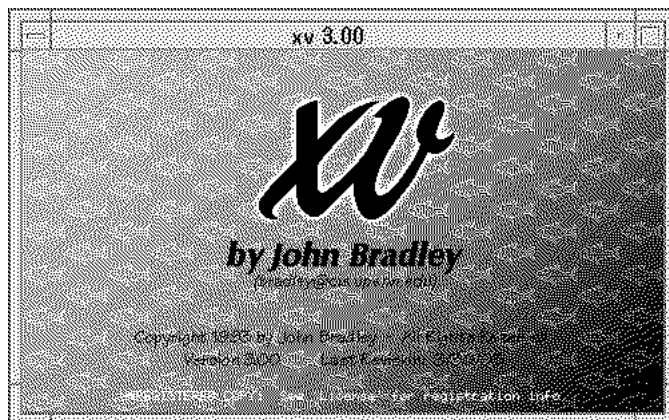


図 14.2 メインウィンドウ

このウィンドウをメインウィンドウと呼ぶことにします⁶。メインウィンドウが表示されたら *xv* の起動に成功しています。とりあえずいろいろな作業をやっていくわけですが、ここではまず一連の操作を覚えるという意味で何も作業しないで終了してみましょ。メインウィンドウはそこに画像を表示するためのものです。すなわち起動した直後はイニシャルの画像（「*xv*」と格好良く書かれた「絵」）が表示されていると言う訳です。このメインウィンドウでは *xv* を操作することは出来ません。メインウィンドウの絵をマウスの右ボタンでクリックし、コントロール用のウィンドウを表示させましょ。

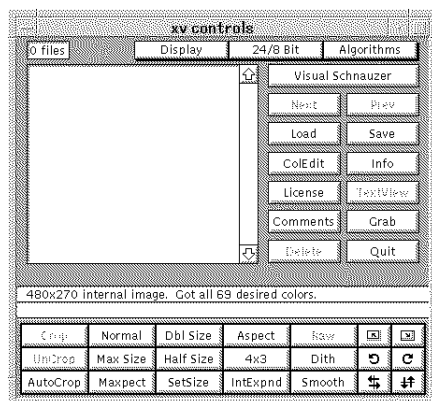


図 14.3 controls ウィンドウ

このコントロールウィンドウには数個のボタンがついていて、このボタンをマウスでクリックすることによって *xv* を操作するのです。 *xv* を終了する為にはコントロールウィンドウの Quit ボタンをクリックして下さい。

14.3.2 画像を表示する

xv の終了操作には成功したと思います。今度は画像を表示する為にもう一度 *xv* を起動して下さい。コントロールウィンドウも表示させておいて下さい。但し今度は、後の操作を簡単にする為に画像データが置いてあるディレクトリに `cd` コマンドで移動してから *xv* を起動して下さい。画像データを持っていない人

⁶正しくは何と呼ぶべきなのでしょう。私は知りません。

は/NF/local/general/lib/picture ディレクトリ以下にサンプルの画像が置いてありますのでこのディレクトリに移動してから起動して下さい。

画像ファイルを表示するにはコントロールウィンドウの Load ボタンをクリックして下さい。以下のウィンドウが表示されるでしょう。

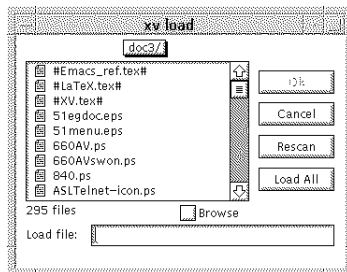


図 14.4 load ウィンドウ

このウィンドウの下側にファイル名を入力する為の枠が取ってありますね。ここにファイル名をタイプして下さい。この枠のすぐ上にはカレントディレクトリのファイルの一覧が表示されていますから、この中から表示させたいファイルをクリックすればいちいちタイプしなくても選んだものをタイプしたのと同じ状態にしてくれます。その状態で Ok ボタンをクリックするか<return>すれば、そのファイルがメインウィンドウに表示されます。

次の画像を表示させたいと思ったらまた Load ボタンを押して、新しい画像データのファイル名をタイプして読み込ませれば良いのです。xv は最後に読み込まれた画像だけを表示し、作業の対象とします。

ファイル名の代わりにディレクトリ名をタイプすることでディレクトリの移動が出来ます。「..」とタイプして一つ上のディレクトリ階層に移動することも出来ます。

14.3.3 画像を保存する

表示されている画像をファイルとして保存するには Save ボタンをクリックして下さい。以下のウィンドウが現れるでしょう。

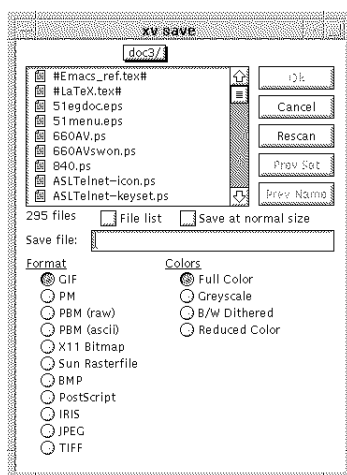


図 14.5 save ウィンドウ

ここでファイル名をタイプして Ok ボタンをクリックするか<return>です。表示されていた画像は指定された名前のファイルとして保存されます。出来上がった画像データファイルがどの程度の大きさになるのか `ls -l` コマンドで確認してみると良いでしょう。個々のファイルの大きさを確認する作業に関しては 154 ページの 6.1.1 を参照して下さい。

画像データをフォーマット変換する

保存の時の save ウィンドウのファイル名の為の枠の下に並んでいる丸いボタンの列に注目して下さい。そこには様々な画像ファイルのフォーマットが並んでいると思います。*xv* は標準的には読み込んだ絵のフォーマットで保存しようとはしますが、ここで違うフォーマットを選ぶ事も出来ます。例えば読み込んだ GIF フォーマットの画像データを保存する時に JPEG で保存するなどという状況です。これによって画像データのフォーマットを変換する事が出来るという訳です。

L^AT_EX など画像を扱う為には PostScript というフォーマットでなければいけません。PostScript での保存については 14.4.3 で詳しく説明していますので参照して下さい。

14.4 応用操作

基本を押えたら次は *xv* に満載された機能 (の一部) を見ていくことにしましょう。説明はかなりはしょって行かないしますので、自分で試しながら理解して行って下さい。

14.4.1 画像データを加工する

さて、*xv* を利用すると画像をいろいろ加工できます。一番簡単な加工方法はメインウィンドウの大きさを変えて絵全体の大きさ、縦横比率を変える事でしょうか。また、コントロールウィンドウの下側に並んだボタンの数々をクリックする事でさまざまな効果が得られます。それぞれどういう機能があるのか、各ボタンをクリックして確認しくと良いでしょう。ここではその中で比較的判りにくい Crop についてだけ説明します。

Crop とは切りとる、刈り取るというような意味で、一般に画像処理関係では要る部分だけを抜き出す作業⁷のことを指します。Crop するにはまずメインウィンドウの中から抜き出す領域を指定します。メインウィンドウの絵の抜き出そうと思う左上の地点から右下の地点までマウスの中ボタンでドラッグします⁸。これで四角い枠がメインウィンドウの画像の中に残ります。領域の指定に失敗したと思ったらもう一度やり直せば良いのです。枠が残っている間にコントロールウィンドウの Crop ボタンをクリックして下さい。メインウィンドウが枠のサイズにまで小さくなったのが判りますね。つまり画像の一部を切りとることが出来たのです。切り取りに失敗したら UnCrop ボタンを押す事で元に戻せます。

さて、様々な画像の変換作業をしたとしても、それは画面の上で行なっているだけで実際の画像データが入っていたファイルには何の影響もありません。加工した画像を画像データファイルに反映させるには現在の画像をファイルに保存してやらなければなりません。保存の仕方は既に説明しましたね。

14.4.2 画面に表示されているウィンドウを取り込む

画面に表示されているウィンドウを絵として画像データに残したいと思う時があるでしょう。例えばこのガイドのように、何かの説明の時に今画面に表示されているものを L^AT_EX に取り込んで例示としたいなどです。単純に画面に表示されているものをそのまま印刷したいと思う事もあるでしょう。

⁷同じ作業をトリミングなどと表現する場合があります。

⁸マウスの中ボタンを押して、押したまま右下に移動してから離す。

そのような場合に役に立つのがコントロールウィンドウにある Grab ボタンです。画像データにしたいウィンドウをよく見える位置に移動して、Grab ボタンをクリックします。「びっ」と音が鳴ったら目標のウィンドウをクリックします。その後数秒して「びびっ」と鳴ったら目標のウィンドウのそっくりさんがメインウィンドウに画像として取り込まれているでしょう。まさに「びっ」と「びびっ」の間に「ひつつかんで」来た訳ですね。あとは好きに加工、保存して利用すれば良いのです。

14.4.3 印刷する

cc 環境ではプリンタは PostScript フォーマットだけを受け付けます。画像データを印刷するにはとにかく PostScript にフォーマット変換して保存してやらなければなりません。印刷したい画像データがメインウィンドウに表示されているとします。この画像をまず保存する為に Save ボタンをクリックして、フォーマットの指定をします。私のお勧めは PostScript - B/W Dithered で保存する事です。Full Color や Grayscale などでも保存しても最終的に印刷できない事はありませんが、一枚印刷するのに数分以上掛かってしまいます。それでは他のプリンタ利用者に迷惑ですからね！フォーマットの指定が済んだら Ok ボタンをクリックするか、<return>すると以下のウィンドウが表示されます。

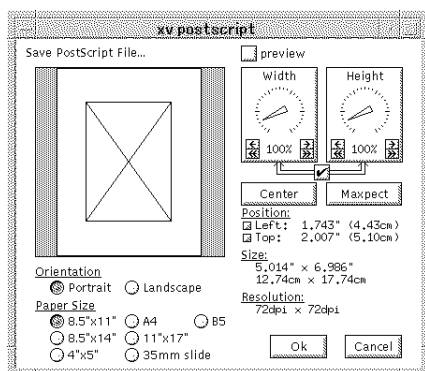


図 14.6 postscript ウィンドウ

ここで印刷する時のイメージ調整が出来ます。いろいろ試して下さい。大抵の事は気にしないで良いのですが、右半分に表示されているダイヤルで行なう倍率指定は有効です。これで印刷する時の縮尺を変えられます。ダイヤルの下にある小さな←ボタンをクリックする事で調整します。調整が決まれば Ok ボタンをクリックするか<return>です。

これで PostScript の画像データファイルが出来ました。プリンタで印刷するのに特殊なテクニックは必要ありません。以下のように lpr コマンドを普通に利用するだけです。

```
cc2000(257)% lpr -Pcspr01 window_title.ps
cc2000(258)%
```

lpr コマンドなど印刷については 107ページの 4.6を参照して下さい。

14.4.4 壁紙を貼る

コントロールウィンドウの上の方にある Display ボタンをプレスして下さい。するとそこから小さなメニューが下がってくるでしょう。そのままマウスを下にドラッグして、Root: tiled と書かれているところでボタンを離して下さい。メインウィンドウがなくなり、そこにあった画像が画面の背景いっぱいに表示

されたでしょう。このような画像の事を壁紙などと呼んでいます。Root: tiled 以外を指定したらどうなるか、試してみると良いでしょう。

このまま画像を新しく Load ボタンをクリックして表示しようとする、その画像は直接壁紙となって表示されます。また、面白い事に壁紙は *xv* を終了しても消えません。

壁紙を貼るもう一つの方法は、以下のようにして *xv* を起動する事です。

```
cc2000(274)% xv -root -quit heavymetal.jpg
cc2000(275)%
```

これでカレントディレクトリにある **heavymetal.jpg** という画像ファイルを壁紙として貼ってくれます。*xv* そのものは **-quit** オプションによってすぐ終了してしまうのですが、壁紙は残ってしまう性質を持っているから大丈夫と言う訳です。

14.4.5 色あいを変える

時々画像が全体的に暗いのもっと全体を明るくしたいとか、ある特定の色を別の色に変えてしまいたいと思う時があります。要は色調を変えようと言う事ですが、*xv* はある程度までならこれを可能にしています。コントロールウィンドウにある ColEdit (Color Edit の略でしょう) ボタンをクリックして下さい。新しいウィンドウが一つ開いて、そこで全体の色調、ある特定の色の変更、モノクロ化、色反転などが出来ます。この Color editor ウィンドウを消すにはもう一度 ColEdit ボタンをクリックすれば良いのです。

Color Edit 機能は説明し出すときりがないので止めておきます⁹。RGB, HSV, Intensity, Saturation と言った色編集についての基礎的な知識がある人には使い方は大体想像が付くと思います。適当に試しながら使ってください。

14.4.6 Visual Schnauzer を使う

何と説明したら良いのでしょうか。余り X ウィンドウ関係の操作に慣れていない人はやらない方が良いと思いますが、慣れている人にとっては結構便利な機能だと思います。

コントロールウィンドウにある Visual Schnauzer ボタンをクリックして下さい。新しく一つのウィンドウが開き、そこにはカレントディレクトリのファイルやディレクトリの一覧がアイコン (絵文字) になって表示されます。そのファイルのアイコンをドラッグしてディレクトリ移動するとか、ダブルクリックして表示させるとか、その種の操作に慣れた人なら違和感なく操作できると思います。Visual Schnauzer ウィンドウの右上にある Misc. Commands ボタンをプレスすれば、そこからメニューが表示されますが、その中でも Update Icons は面白い機能です。実行すると以下のような一段と判りやすい表示となります。

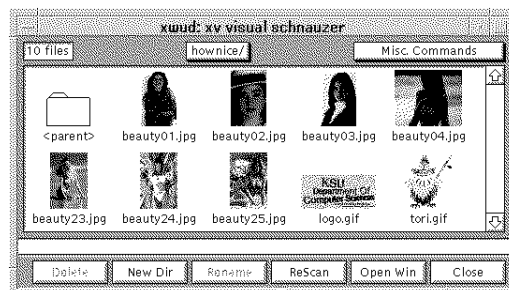


図 14.7 Visual Schnauze ウィンドウ

⁹ 実は私も良く判っていません。

Visual Schnauzer ウィンドウを閉じるには、Visual Schnauzer ウィンドウの右下にある Close ボタンをクリックします。

14.5 マニュアルなど

`xv` はまったくもって機能満載です。それに見合う非常に詳細なマニュアルが用意されていますが、それは `man` コマンドで表示されるものではありません。 `/NF/local/Solaris2J/lib/xv/xvdocs.ps` もしくは `/NF/local/OSF1/lib/xv/xvdocs.ps` というファイルに PostScript 形式で置いてあります。14.4.3で紹介したように PostScript ファイルを印刷するのに特にテクニックは必要ありません。 `lpr` コマンドで直接印刷すれば良いのです。 `lpr` コマンドなど印刷については 107 ページの 4.6 を参照して下さい。

但しこのファイルの印刷には非常に時間が掛かります。全部で 100 ページ近くありますし、一ページそのものを印刷するのも結構時間が掛かるものが含まれています。印刷する時はプリンタを長い間自分が占領してしまっても迷惑が掛からないかどうか確かめて行なって下さい。

14.6 法律に注意

画像を扱う時に常に注意すべき事に肖像権の問題があります。あなたが自分の作っているドキュメントに誰かのポートレート写真を使いたい時があるのは判ります。しかしその写っている相手はそうやって自分の写真があなたのドキュメントの中で使われる事を承知していますか？特にコンピュータで出版するような場合は簡単に複製、配布できたりします。作ったドキュメントがあなた以外の人の目に触れるような場合は肖像権の問題は非常に重要です。図 14.1 の写真は勿論このガイドに使って無料配布すると言う事を本人に確認し、承諾して貰ってから使っています。

同様の意味であなたが扱っている画像データはその絵の作者の著作権を侵害していませんか？自分が生み出したものだけを扱っている時はこのような問題はほぼ発生しないと言えますのですが、人が作ったものを扱う時には非常に大切な事です。注意して下さい。

第 15 章

xpaint

xpaint を使うと、画像データを作成、加工、表示させる事が出来ます。xpaint は X ウィンドウ環境で働きますので、利用するためには X 環境が必要です。具体的には cc 環境では 21 情報処理教室の csosf01~40 に login するか、C1 情報処理教室などの X 端末ソフトを用意しているコンピュータから cc2000 に login する必要があります。

この章では xpaint の簡単な使い方を説明します。

15.1 はじめに

画像データの作成、加工といっても具体的にその必要性が思い浮かばないかも知れません。例えば何かの画像を \LaTeX の文中に張り付けたい時があったとしましょう。以下のような具合です。

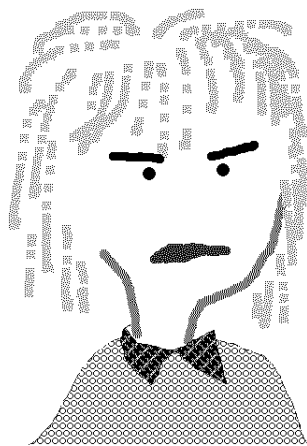


図 15.1 意味もなく現れる現在のユカリさん

このようなふざけた例ばかりでなく、tgif¹で作成した絵なども画像のうちの一つです。さあ、あなたも画像データを扱いたくなってきたでしょう！

¹tgif については第 13 章を参照して下さい。

15.2 まずマニアックな人のために

最初に画像関係の事情をよく判っている人のための機能説明を専門用語でさささっとやってしまいます。画像ファイルのフォーマットなどと聞いて良く判らない人は次の章まで読み飛ばして下さい。

xpaint は以下の画像フォーマットを扱う事が出来ます。

- TIFF これまた一般的に流通しています。NeXT は拡張して色数 256 以上に増やしています。機種依存が激しく互換性に乏しいフォーマットですね。
- PPM Portable Pixmap. またまた良く知りません。
- GIF 一般的に流通していますね。256 色までしか表現できないのが今となっては辛い。
- XBM X Bitmap. モノクロ二値画像のみです。X 環境では `bitmap` コマンドで扱えます。
- XPM 良く知りません。
- XWD X Window Dump. X 環境では `xwd`, `xwud` コマンドで扱えます。

プリンタなどで流通している PostScript フォーマットを、xpaint は各種画像の出力フォーマットとしてだけサポートします。PostScript 画像を表示する事は出来ません²。

15.3 基本操作

まず基本操作を習得しましょう。

15.3.1 xpaint の起動と終了

xpaint を起動するには X 環境が利用できないといけません。cc 環境であればあなたは既に 21 情報処理教室の `csosf01~40` に login しているか、C1 情報処理教室などの X 端末ソフトを用意しているコンピュータから `cc2000` に login していますね。そこでターミナル (cc 環境であれば Kterm です) を一つ表示させて、そこにはコマンドプロンプトが表示されている事を確認して下さい。

そこから以下のように `xpaint &` で xpaint を起動します。以下の例ではコマンドプロンプトは `cc2000` マシンのものになっています。

```
cc2000(199)% xpaint &  
[7] 23547  
cc2000(200)%
```

すると以下のようなウィンドウが一つ現れると思います。

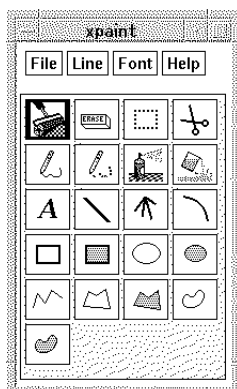


図 15.2 Toolbox ウィンドウ

²PostScript 画像を表示するには Ghostview アプリケーションを利用して下さい。但し cc 環境では Ghostview が使えるのは cc2000 だけです。

このウィンドウを Toolbox(工具箱) ウィンドウと呼びます。この Toolbox ウィンドウには多くのボタンがついていて、各ボタンをマウスでクリックすることによって xpaint を操作するのです。Toolbox ウィンドウが表示されたら xpaint の起動に成功しています。とりあえずいろいろな作業をやっていくわけですが、ここではまず一連の操作を覚えると言う意味で何も作業しないで終了してみましよう。

Toolbox ウィンドウの上方には File, Line, Font, Help のメニューボタンが用意されています。この File メニューボタンをマウスの左ボタンでプレスして下さい。以下のようなメニューが表示されます。

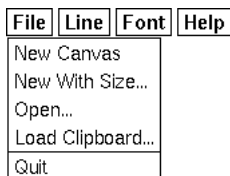


図 15.3 File メニュー

このメニューの一番下にある Quit の位置までマウスをドラッグし、Quit 項目の色が黒くなったらマウスのボタンを離して下さい。この一連の操作を「File メニューから Quit を選択する」と表現します。それで xpaint は終了します。

15.3.2 画像を新規に作成する

xpaint の終了操作には成功したと思います。今度は画像を作成する為にもう一度 xpaint を起動して下さい。

新規に画像を作成するには File メニューの New Canvas を選択して下さい。以下のウィンドウが表示されるでしょう。

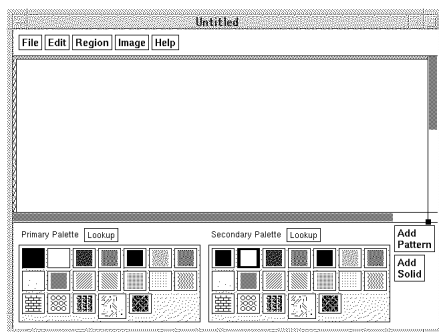


図 15.4 Painting ウィンドウ

さあ後はこのキャンバスはあなたのものです。自由に好きな絵を描きましょう。絵を描くには Toolbox ウィンドウから道具を選択し、キャンバスにマウスでクリックしたりドラッグしたりして行ないます。どの道具がどのような機能を持っているか一つ一つ試して行くのが良いでしょう。色は Painting ウィンドウの下半分に表示されているパレットをクリックして選択します。基本的には道具、色などを選択しておいてからキャンバスをクリックなどする事によってその色でその機能が働き出すようになっています。

直前の描画動作に関しては Edit メニューの Undo でやり直しが利きます。

15.3.3 画像を保存する

キャンバスの画像をファイルとして保存するには Painting ウィンドウの File メニューから Save を選択して下さい。以下のウィンドウが現れるでしょう。

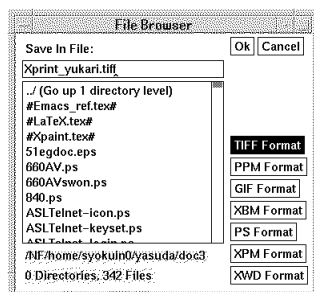


図 15.5 save ウィンドウ

ここでファイル名をタイプして Ok ボタンをクリックするか<return>です。表示されていた画像は指定された名前のファイルとして保存されます。出来上がった画像データファイルがどの程度の大きさになるのか `ls -l` コマンドで確認してみると良いでしょう。個々のファイルの大きさを確認する作業に関しては 154 ページの 6.1.1 を参照して下さい。

ファイル名の代わりにディレクトリ名をタイプすることでディレクトリの移動が出来ます。「..」とタイプして一つ上のディレクトリ階層に移動することも出来ます。

ファイル名を指定して、Ok ボタンを押す前に save ウィンドウの右に並んでいるフォーマットのボタンを押して保存する画像ファイルのフォーマットを選ぶ事も出来ます。

15.3.4 既存の画像ファイルを読み込む

既存の画像ファイルをキャンバスに読み込む事も出来ます。Toolbox ウィンドウの File メニューから Open... を選択します。図 15.5 の save ウィンドウに酷似したウィンドウが現れるでしょう。保存の時と同じようにここでファイル名をタイプして Ok ボタンをクリックするか<return>すればそのファイルがキャンバスに表示されます。

こうして既存の画像ファイルをキャンバスに読み込んで手を加える事により、画像の加工が出来るのです。加工されたファイルを既存のファイルに上書きする形で保存するのならば、Painting ウィンドウの File メニューから Save を、元のファイルはそのままに別のファイルに保存したければ Save As... を選択して下さい。

既存の画像ファイルが xpaint で扱えないファイルフォーマットであった場合は *xv*³を利用して xpaint が扱えるフォーマットに変換してから処理し、再び *xv* を利用して元のファイルフォーマットに戻せば大丈夫です。

15.4 応用操作

xpaint にはなかなかどうして多くの機能があります。書くとキリがありませんし、習うより慣れるというタイプのもでもあります。自分でいろいろ試していくのが良いでしょう。ここではとりあえず直感的にイメージできない操作方法によって実行される機能に付いてだけ紹介しています。


³ *xv* については第 14 章を参照して下さい。

15.4.1 道具の設定

線の太さ

鉛筆の太さや、四角や丸を書く時の枠線の太さを変えるには Toolbox ウィンドウの Line メニューで行ないます。



とにかくダブルクリック

Toolbox ウィンドウの道具ボタンをダブルクリック (二回続けてクリック) すると、幾つかの道具に関してはその道具の機能設定の為のウィンドウが表示されます。例えばローラー () をダブルクリックするとローラーの大きさを設定できるウィンドウが表示されます。

また、Painting ウィンドウの色パレットをダブルクリックすると今度は色を設定する為のウィンドウが開いて、あなたはそこで自由に色を作る事が出来ます。

15.4.2 Region

Region という概念があります。これからある操作をしようと思っている領域の事です。幾つかの機能はこの Region を設定してからでないと実行できません。

Region を設定するには矩形切り出しもしくはハサミ ( ) を選択して、Painting ウィンドウで指定したい領域をドラッグして決めます。うまく設定できたら目印として Region の上下左右、四隅の合計 8 点に黒い小さな四角が表示されます。

こうして Region 設定が出来たら、以下のような操作が可能になります。

移動 マウスの左ボタンで Region をドラッグしてください。

削除 Painting ウィンドウの Edit メニューから Cut を選択すると削除できます。

複写 Painting ウィンドウの Edit メニューから Duplicate を選択すると複写でき、それが新しい Region として扱われます。新しい Region を移動しても元の Region の図形はその場に残っています。

貼付 Painting ウィンドウの Edit メニューから Paste を選択すると直前に Cut もしくは Copy した Region を張り付ける事が出来ます。

回転 マウスの中ボタンで Region の端の方をつまんでドラッグして下さい。

更に Painting ウィンドウの Region メニューが選択できるようになり、そこでは反転、回転、色反転、エッジ抽出、エンボス加工、油絵風加工など色々な加工が出来ます。

15.4.3 拡大表示

Painting ウィンドウの Image メニューから Change Zoom... を選択すれば拡大倍数が選べます。ここでの数値は倍数でパーセントではありません。間違って 200% のつもりで 200 などと指定すると大変な事になりますので注意しましょう。

また、写真の加工などをしようとする時、一画素ごとに加工したいと思う時が多いでしょう。同じく Image メニューで Fat Bits を選択すると Painting ウィンドウとは別に Fat Bits ウィンドウが表示され、そこでは非常に大きく拡大された表示が行なわれます。Fat Bits ウィンドウでは Painting ウィンドウと同じ描画、加工の操作が行なえます。

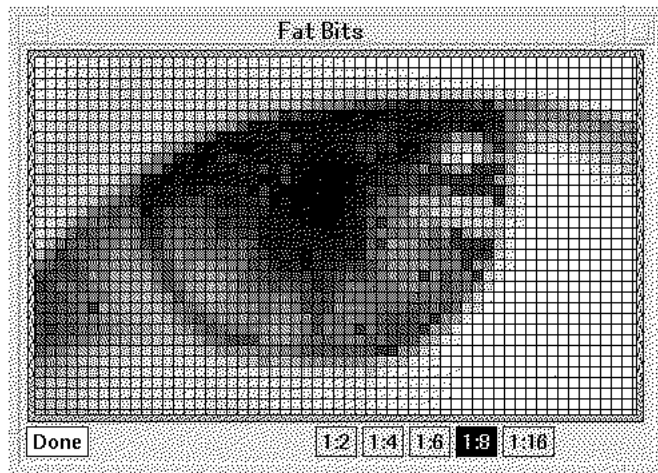


図 15.6 Fat Bits ウィンドウ (目を拡大表示したところ)

FatBits で表示されている領域が Painting ウィンドウの中に小さな四角い枠で表示されているのが判るでしょう。この枠をマウスでドラッグする事によって FatBits による拡大表示の場所も移動します。

15.5 マニュアルなど

シェルプロンプトから `man xpaint<return>` でマニュアルは表示されますが、そこでは xpaint の機能に付いてはほとんど何も教えてくれません。そのかわり Toolbox ウィンドウや Painting ウィンドウの Help メニューから以下のような Help ウィンドウを表示させる事が出来ます。

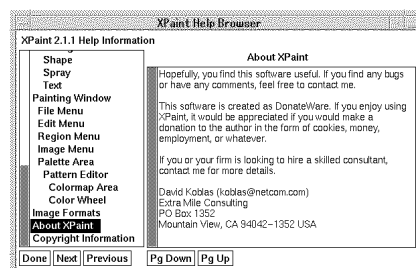


図 15.7 Help ウィンドウ

このウィンドウを閉じるには左下にある Done ボタンをクリックして下さい。

第 16 章

NQS

大勢の人たちで一つのコンピュータを共用していて、科学技術計算などの大規模な処理を実行したい人がたくさんその中に含まれていたとします。例えば 10 人の利用者がそれぞれ 3 時間かかるような大規模な計算を同時に同じコンピュータに掛けたとすると、全員の処理が終了するのが合計 30 時間よりも遥かに長くなったりする場合は往々にしてあります。これは一時に負荷のかかる計算をやらせた為に、計算機が過負荷状態になってしまうことから来ています。これを回避する為には、一つずつプログラムを実行し、先のプログラムの実行が終わったら次のプログラムの実行に掛かるようにするのが最善です。

このようなシステムは一般にバッチシステムと言われ、科学技術計算を良く行なうコンピュータ¹には大抵実装されているもので、そのような要求のある人たちにとっては大変便利なもので、VMS²や MSP³などの OS には実装されています。

しかし、その生まれの経緯から Unix にはもともとバッチキューという概念がありません。NQS はこのバッチキューを Unix 上で実現するソフトウェアです。cc 環境では cc2000 マシン (SPARCcenter2000) に実装されています。ここでは cc2000 上の NQS を例にとりて、NQS の使い方を説明します。

16.1 今どんなバッチキューがあるか？

バッチキューの確認には `qstat -b` とします。

```
% qstat -b
```

です。以下に例を示しておきます。

```
cc2000(411)% qstat -b
=====
NQS Version: 2.3  BATCH QUEUES on cc2000
=====
QUEUE NAME      STATUS   TOTAL   RUNNING  QUEUED   HELD  TRANSITION
-----
BATCH           AVAILBL  0       0/4/4    0        0      0
I               AVAILBL  0       0/2/2    0        0      0
cc2000(412)%
```

¹もしくは古典的なコンピュータ

²DEC-3500 などの OS です。

³FACOM の OS です。

STATUS のところが **available** になっていますから、利用可能なキューはこの時点で BATCH と I の二つですね。また、RUNNING 表示に出てくる A/B/C のそれぞれの数字は、各キューで

A : 実行されているジョブの数

B : 一人のユーザが同時に走らせることが出来るジョブの数

C : 同時に実行可能なジョブの数

をそれぞれ示しています。SPARCcenter2000 に限って言えば、このマシンは4つのCPUを同時に並行動作させられますから、同時に4つ以下の数のジョブが流れるぶんには全く負荷は掛かりません。(もちろんメモリを食いすぎると負荷になりますが。)

このうち BATCH は、デフォルトキューとなっていますから、これ以降の全てのコマンドでキュー指定をしなかった場合は、この BATCH という名前のキューが対象となります。(どのキューがデフォルトキューかという情報は表示されません。)

16.2 バッチジョブの投入

バッチジョブの投入には qsub コマンドを利用します。

```
% qsub [ -q QUEUE ] [ SHELLSCRIPT ]
```

です。QUEUE にはキュー名を、SHELLSCRIPT には、実行したいコマンド列を自分の login shell の文法で書かれたファイルの名前を書きます。

デフォルトキューで実行する場合は -q QUEUE を外してください。スクリプトが簡単に済むなら省略が可能です。その場合は標準入力にコマンド列を書くように要求してきますから、

```
% qsub [ -q QUEUE ]
```

だけで実行し、そこあとにずらずらとコマンド列を書くという形になります。

例として bb というファイルをデフォルトキューで実行します。以下の例を見て判るように、今回のスクリプトは非常に単純で、単に自分で作った resolv というプログラムをホームディレクトリのすぐ下の projectX という名前のサブディレクトリに置いていて、それにパラメタを与えて実行するだけです。date コマンドを前後に入れて時間を表示させるようにしています。

```
cc2000(360)% cat bb
cd projectX
date
resolv 100 x10 y20 z30
date
cc2000(361)% qsub bb
Request 16.cc2000 submitted to queue: BATCH.
cc2000(362)%
```

このように、16番と言うエントリ番号を貰いました。

16.2.1 ジョブの始まりと終わりのお知らせを貰う

自分が投入したジョブの前に、誰かが非常に時間の掛かるジョブを同じキューに投入していたりすると、自分のジョブはいつ実行されるか判りません。また、いつ終わるのかも判りません。

ジョブを投入するときに `-mb` , `-me` オプションを付けておけば、それぞれジョブが始まったときと終わったときに mail をくれます。両方のオプションを付けた例を示しておきます。

```
cc2000(402)% qsub -mb -me bb
Request 24.cc2000 submitted to queue: BATCH.
cc2000(403)%
```

こうしておくで、Subject が `24.cc2000 beginning.` などというメールをお知らせに貰う事が出来ます。

16.3 ジョブの標準出力

ジョブの結果がどのように残ってプログラマに与えられるのかはプログラム次第ですが、普通に shell script を実行したときに発生する標準出力と標準エラー出力は自動的にファイルとして生成され、残されます。投入したジョブが終わると、ジョブを投入したときのスクリプトファイルの名前の後に “.o” とエントリ番号を付けたファイルが標準出力の内容を残します。また、”.e” とエントリ番号を付けたファイルが標準エラー出力の内容を残しています。もしもスクリプトファイルを利用せずに標準入力からジョブの投入を行った場合は、STDIN に続いて “.o” と “.e” が付いたファイルが生成されます。

例を示しておきます。まずはエラーファイル。

```
cc2000(425)% more bb.e16
cc2000(426)%
```

空っぽでした。続いて標準出力。

```
cc2000(426)% more bb.o16
Sun Microsystems Inc.   SunOS 5.2       Generic March 1993
1994年01月26日(水) 19時43分01秒 JST
Start
End.
1994年01月26日(水) 19時43分19秒 JST
cc2000(427)%
```

ああ、うまく行ったようです。安心安心。

これらのファイルは自動的に、`qsub` でジョブを実行した時のディレクトリに生成されます。

16.3.1 標準出力、エラー出力のファイルを変更したい

もしもファイル名や生成される場所を変えたい場合は、`qsub` に `-o` , `-e` オプションを加えます。

```
% qsub [ -o STDOUT ] [ -e STDERR ] ...
```

です。STDOUT には標準出力の宛先ファイルを、STDERR には標準エラー出力の宛先ファイルを指定します。例として両方のオプションを付けたものを示します。

```
cc2000(431)% qsub -o cc -e ee bb
Request 29.cc2000 submitted to queue: BATCH.
cc2000(432)%
```

標準出力ファイルが `cc` に、エラーファイルが `ee` に作成されました。それぞれ `cc.o29` や `ee.e29` にはならない事に注意してください。

また、エラー出力と標準出力を同じファイルに出したいときは `-eo` オプションを使います。

```
% qsub -eo [ -o STDOUT ] ...
```

`-eo` オプションを付けて、なおかつ `-o` オプションでファイルを指定すると、エラー出力も、標準出力も共に `STDOUT` に指定したファイルに出力されます。

16.4 ジョブの状態表示

自分がエントリーしたジョブの状態を見るには、`qstat` コマンドを利用します。

```
% qstat { ENTRY | QUEUE }
```

です。`ENTRY` には上記のエントリー番号がはいります。`QUEUE` にはキュー名が入ります。例として先ほどの 16 番の状態を見てみます。

```
cc2000(362)% qstat 16
=====
NQS Version: 2.3 BATCH DEVICE PIPE REQUESTS on cc2000
=====
REQUEST      NAME          OWNER        QUEUE        PRI NICE CPU    STATE
   16.cc2000  bb            yasuda       BATCH        31  20 36000  RUNNING
cc2000(363)%
```

最後の `STATE` が `RUNNING` ですから、無事に今処理されている最中という事です。

ところで他の全てのジョブの状態を知るには `qstat -a` とします。

```
% qstat -a [ QUEUE ]
```

です。`QUEUE` にはキュー名が入ります。省略すると全てのキューの情報が表示されます。例を示しておきます。

```
cc2000(365)% qstat -a
=====
NQS Version: 2.3 BATCH DEVICE PIPE REQUESTS on cc2000
=====
REQUEST      NAME          OWNER        QUEUE        PRI NICE CPU    STATE
   17.cc2000  bb            yasuda       BATCH        31  20 36000  RUNNING
   18.cc2000  bb            yasuda       BATCH        31  20 36000  QUEUED
   28.cc2000  bb            yasuda       I             31  20 36000  RUNNING
cc2000(366)%
```

現在 17 番が `BATCH` キューで実行中で、18 番は待ち状態にあるということが判ります。また、28 番が `I` キューで実行中です。

16.5 流れているジョブの中身を確認する

qcat コマンドで、キューに入っているジョブの内容などの確認が出来ます。

16.5.1 ジョブの記述を確認する

既に投入したジョブの記述を確認するには qcat コマンドを利用します。

```
% qcat ENTRY
```

です。ENTRY には上記のエントリ番号がはいります。例として先ほどの 16 番の中身を見えます。

```
cc2000(363)% qcat 16
>>> Input file is /usr/spool/nqs/private/root/data/+++++F/++++F+++++0+++
#
date
cd projectX
resolv 100 x10 y20 z30
date
cc2000(364)%
```

という表示が出てきました。勿論 qcat コマンドは現在実行中、もしくは実行待ちのジョブに対してしか効きません。既に実行が終ってしまったジョブについてはどうにもなりません。

16.5.2 流れているジョブの途中経過を確認する

現在実行中のジョブの出力を実行の最中に確認するには qcat -o コマンドを利用します。

```
% qcat -o ENTRY
```

です。ENTRY には上記のエントリ番号がはいります。例として先ほどの 16 番の中身を見えます。

```
cc2000(364)% qcat -o 16
Sun Microsystems Inc. SunOS 5.2 Generic March 1993
1994年01月26日(水)19時43分01秒JST
Start
cc2000(365)%
```

という表示が出てきました。勿論 qcat -o コマンドは現在実行中のジョブに対してしか効きません。既に実行が終ってしまったジョブについては出力ファイルを見ることになります。

非常に長いジョブを実行する場合は処理の切れ目などで時々標準出力にメッセージを出すようにプログラムを書いておくと qcat -o 機能をうまく利用できて便利です。

なお、-o オプションの代わりに -e オプションを与えることによって (もしあれば) エラー出力の内容を実行途中に確認できます。

16.6 ジョブの実行を保留する

待ち状態にあるジョブの実行を一時保留するには `qhold` コマンドを利用します。

```
% qhold ENTRY
```

です。ENTRY には上記のエントリ番号が入ります。以下に例を示します。

```
cc2000(459)% qhold 33
Request 33 has been held.
cc2000(460)%
```

hold されたジョブを再び解放するには、`qrls` コマンドを利用します。

```
% qrls ENTRY
```

です。ENTRY には上記のエントリ番号が入ります。以下に例を示します。

```
cc2000(464)% qrls 33
Request 33 has been released.
cc2000(465)%
```

`qhold` は、現在実行中のジョブには適用出来ません。`qstat -a` でチェックしたときに、`QUEUED` になっているものだけです。以下に `qstat -a` の例を示します。

一番右の `STATE` の列を見てください。`RUNNING` が現在実行中で、`QUEUED` が現在実行待ちのものです。31番は `HOLDING` になっていますね。これは先ほど実行した `qhold` のせいで、実行が保留されているという事です。

```
cc2000(462)% qstat -a
=====
NQS Version: 2.3 BATCH DEVICE PIPE REQUESTS on cc2000
=====
REQUEST      NAME          OWNER        QUEUE        PRI NICE CPU    STATE
   32.cc2000  bb            yasuda       BATCH        31  20  36000  RUNNING
   33.cc2000  bb            yasuda       BATCH        31  20  36000  QUEUED
   31.cc2000  bb            yasuda       BATCH        31  20  36000  HOLDING
cc2000(463)%
```

16.7 ジョブの実行を停止、削除する

ジョブの実行を停止するには `qdel` コマンドを利用します。

```
% qdel [ -SIGNAL ] ENTRY
```

です。現在実行中のジョブの実行を停止させるには、`SIGNAL` に `9` を入れると良いでしょう。待ち状態のジョブの実行を取り消すだけなら、`-SIGNAL` は省略可能です。ENTRY には上記のエントリ番号が入ります。以下に例を示します。

```
cc2000(561)% qdel -9 54
Request 54 is running, and has been signalled.
cc2000(562)%
```

16.8 もっと詳しいキューの情報を調べる

最初に `qstat -b` コマンドによってキューの情報を調べる例を挙げましたが、もっと詳しく、それぞれのキューに割り当てられている最大メモリ量、最大 CPU タイムなどを調べるには `qstat` コマンドに `-bl` オプションを与えます。

```
% qstat -bl [QUEUE]
```

です。QUEUE にはキュー名が入ります。省略すると全てのキューの情報が表示されます。例を示しておきます。

```
cc2000(562)% qstat -bl BATCH
=====
NQS version: 2.3          BATCH QUEUE: BATCH.cc2000      status: AVAILBL
=====
                                Priority: 0
ENTRIES:
    Total: 0          Running: 0
    Queued: 0         Held: 0          Transiting: 0
COMPLEX MEMBERSHIP:
RESOURCES:
  Per-proc core file size limit= 1 Mbytes <DEFAULT>
  Per-process data size limit  = 10 Mbytes
  Per-proc perm file size limit= 10 Mbytes
  Per-proc execution nice value= 0 <DEFAULT>
  Per-process stack size limit = 10 Mbytes
  Per-process CPU time limit   = 300.0
  Per-process working set limit= 10 Mbytes
```

ACCESS

Unrestricted access

上記の `Per-proc data size limit` などが一つのジョブ当たりの最大メモリ量です。`Per-process CPU time limit` などが一つのジョブ当たりの最大 CPU タイムとなります。これらの制限を越えると、その時点でジョブは強制的に終了させられます。

16.9 エラーメッセージ

いつもの通りにジョブが動くはずのところ、以下のようなエラーが出てくる時があります。

16.9.1 あなたの所在地はどこですか？

これは `tty` 情報が取れない環境で `biff` を起動したときに表示されるエラーです。`biff` コマンドが `$HOME/.login` ファイルなどに含まれているのではありませんか？対処方法はありません。無視するか `biff` コマンドを使わないようにするしかありません。

16.9.2 警告: tty にアクセスできないため、このシェルでジョブ制御はできません ...

```
Warning: no access to tty; thus no job control in this shell...
```

もしくは

```
Warning: no access to tty (Bad file number).
```

```
Thus no job control in this shell.
```

これは tty 情報が取れない環境で `csch` もしくは `tcsh` を起動したときに表示されるエラーです。対処方法はあります。無視してください。

16.10 マニュアルなど

cc2000 にインストールされている NQS に関するドキュメントとしては `cc2000:/NF/local/Solaris2J/lib/nqs` 以下に `troff` の `mm` マクロ形式と、それを `PostScript` 形式に落したものが用意してあります。

また、一般的なオンラインマニュアルがインストールされているはずですので、`man -k nqs` などとして探せば大抵のコマンドの詳細情報が得られます。

第 17 章

MODEM から telnet

現在個人でパソコンを利用している方はモデムと呼ばれる製品をパソコンに接続して使うことで電話回線等を通して自宅ないしは研究室などから京都産業大学の UNIX コンピュータなどを利用することができます。2.6で説明した、telnet による UNIX コンピュータの遠隔操作が自宅や研究室から可能になるということです。ここでは接続する際の中継点ともいえるターミナルサーバーについて、その概要と利用法を説明します。

17.1 計算機センター管理のターミナルサーバー

現在、計算機センターでは以下のターミナルサーバーとモデムの組み合わせを提供しています。

ターミナルサーバー	モデム名	回線数	外線番号・内線番号
DecServer700	DMP	15	705-1500・2590
	MD144FB5V	4	705-1501・2591
	MD144XT10V	5	705-1489・2592
Annex	DMP	7	705-1488・2594

以下にそれぞれのモデムの性能を示します。(bps=ビット/秒)

DMP	2400bps MNP 無し
MD144FB5	14400bps V.42bis
MD144XT10V	14400bps V.42bis

学内の多機能電話を使用する場合は DMP の内線に対してのみ接続でき、その場合の転送速度は 9600bps となります。多機能電話は教員を対象に貸し出されています。貸し出しの相談は施設課までお願いします。

17.2 アクセス（接続）するまえに

アクセスするためにはパソコンの他に、モデムと通信ソフトが必要です。

モデム

モデムについては一般のパソコンなどを扱っている電気店で購入出来ます。14400bps V.42bis 対応の物がお勧めです。2400bps MNP5 のモデムなども使用できますが、通信速度は使い勝手に大きく影響するこ

と、値段もそれほど変わらないことから前記の物をお勧めします。現在では 28800bps のものも手頃な価格で出荷されていますので、検討されると良いでしょう。

通信ソフト

通信ソフトについては VT100 エミュレーションの出来るものがが必要です。市販のもの、フリーソフトウェアなどと呼ばれる無料で配布されているものなど幾らかありますのでパソコンに合わせて用意して下さい。代表的なものとして、hterm や termnec などを見算機センターで無料配布しています。これらの選定などについては、見算機センターの相談窓口までお気軽にご相談ください。

17.3 アクセスしてみましよう !!

さあ実際にアクセスしてみましよう。はじめに DecServer700 への接続を例に説明します。

1. 通信ソフトを使ってターミナルサーバーに接続します（電話をかけます）。
2. 接続が成功すると以下のメッセージがでます。

```
Connect (接続状況)
```

3. 接続が成功したら<return>です。画面がクリアされ、以下のようなメッセージと Local>プロンプトが表示されます。

```
Network Access SW V1.1 for DS700-08 (BL55-16)
(c) Copyright 1993, Digital Equipment Corporation - All Rights Reserved
Please type HELP if you need assistance
Local>
```

4. cc2000 に接続するには telnet コマンドを利用します。

```
Local> telnet cc2000
Trying...
Local -009- Session 1 to CC2000 established
```

```
UNIX(r) System V Release 4.0 (cc2000)
login:
```

5. login:プロンプトが表示され、cc2000 にアクセスされたことがわかります。実際に login してみましよう。

```
login: kosa
Password:
Last login: Tue Aug 2 09:41:32 from platinum.kyoto-s
Sun Microsystems Inc. SunOS 5.3 Generic September 1993
you have mail
cc2000(81)%
```

実際の操作では `login:` に続く `kosa` はあなたのユーザ名になり、その後で `<return>` キーを押すことになるでしょう。そして `Password:` に続くあなたのパスワードはいつものように表示されないまま打ち込まなくてはなりませんし、その後で `<return>` キーを押すのです。

- 上記のように `cc2000` のプロンプトが表示されたでしょうか。これで安心してはいけません。`cc2000` に対して `VT100` (エミュレーション) の端末を使用していることを教えて上げる必要があります。以下のコマンドを実行してください。`set term=vt100<return>` です。

```
cc2000(85)% set term=vt100
```

これで `login` に関する作業は終了です。

ターミナルサーバーとの接続を切るには以下のようにします。

- `cc2000` を `logout` します。
- Decserver のプロンプトが表示されますので、`logout` コマンドを実行してください。モデムの接続が切れます。

```
Local> logout
```

17.3.1 Annex ターミナルサーバの場合

次に Annex の使い方です。Decserver700 との違いのみを説明します。

- プロンプトが `Local>` ではなく `annex:` である。
- ターミナルサーバとの接続を切るには `logout` コマンドではなく `hangup` コマンドを使う。

17.4 さらに詳しい使い方

ここに挙げた以外にもターミナルサーバーには多彩な機能があります。各ターミナルサーバーのコマンド一覧は、それぞれのプロンプトから `help<return>` とすると確認できます。各コマンドの使い方は `help` コマンド名 `<return>` で確認できます。

第 18 章

NeXT

18.1 初めて NeXTStation を使う人へ

NeXTStation は計算機科学研究所棟 3 階の C3 情報処理教室に設置されているコンピュータです。NeXT では UNIX の環境をマウス操作やグラフィック (絵柄) を多用して、視覚的にわかりやすく利用することができます。使用できる主なソフトウェアとして、「Mathematica [マセマティカ]」(数式処理ソフト)「文机 [ふづくえ]」(日本語ワープロ)「Improv [インプロブ]」(表計算ソフト)があります。また「Objective C [オブジェクトティブ シー]」でオブジェクト指向のプログラミングでのアプリケーション開発ができます。

この節を読み進めるに当たって、はじめて使う人にもわかりやすいマニュアル「ようこそ NeXT...」¹をときどき参照してもらおうことになりますので手元においてから次へ進むとよいでしょう。

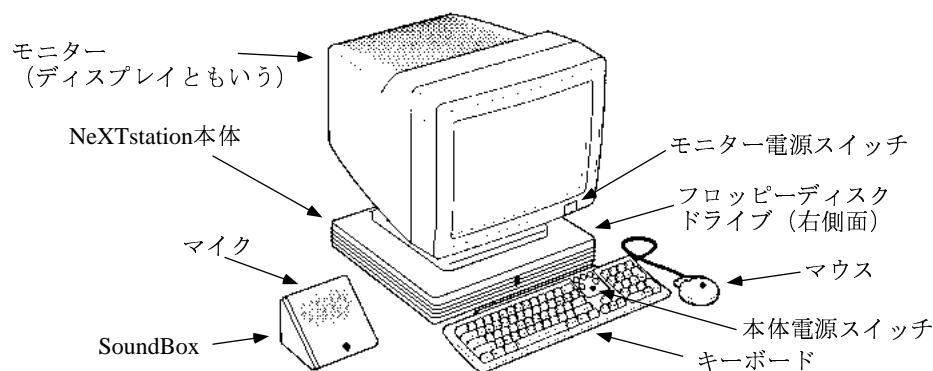


図 18.1 NeXTStation システム外観

18.1.1 まず部屋へ

まず、置いてあるところへ行ってみましょう。NeXTStation は計算機科学研究所棟の C3 情報処理教室に設置してあります。この教室は付録の「情報処理教室利用要項」に記載されている日・時間帯ならいつでも使えますが、次のことに留意してください。

部屋に入るときは、そこで授業をやっていないことを確認しましょう。もし授業中であれば入室して利用していいかどうか指導教員に確認を取るのが礼儀というものです。部屋に誰も居なかった場合は照明、空調機が切れている場合がありますので自分で適宜操作してください。照明スイッチは部屋に入っすぐの

¹C3 情報処理教室に入って左側の書架に並んでいる、60 ページほどの薄いマニュアルです。

右側、空調装置は入り口の反対側にある窓際のデスクの向こう側に設置されています。部屋そのものの管理ですが、計算機科学研究所事務室が行っていますので、何か部屋の運用について質問、要望があれば計算機科学研究所事務室までお願いします。また、コンピュータについての質問、要望については計算機センターの相談窓口で受けています。

18.1.2 電源を入れる

まず自分が利用するコンピュータを選びましょう。もし今回 NeXTStation を初めて使うのであれば、一連の操作に慣れるため、電源の入っていないマシンを選ぶのがいいでしょう。もし次の条件に当てはまっていれば、そのマシンの電源は入っています。

1. 画面に何か映っている。
2. 画面は暗いが、マウスを左右に転がすと急に画面が明るくなる。
3. モニターの電源が切れているが、次に述べている方法に従ってモニターの電源を入れて 10 秒程度待つと、何かが映る。もしくは 10 秒ほど待ってマウスを転がすと画面が急に明るくなる。

NeXTStation を使うためには NeXTStation 本体の電源とモニター²の電源を入れなくてはなりません。まず最初にモニターの電源スイッチを ON にしてください。画面の右下に「○」と「|」が付いているスイッチなのですが、「○」は「0」（ゼロ）を表し「OFF」を意味し、「|」は「1」を表し「ON」を意味します。今回は「|」の方にスイッチを押し込んでください。すると、スイッチの隣の緑色の電源ランプが点灯します。次に NeXTStation 本体の電源を入れます。キーボード上の図 18.2 の位置に「Power」と書かれた 1 つだけ緑色をしたキーがありますが、これが本体の電源スイッチです。これを押すと本体の電源が入り「カチッ・ブーン」という音とともに本体が起動しはじめます。

次の段階である login の準備を整えるまでしばらくかかりますので、そのまま数分間お待ちください。

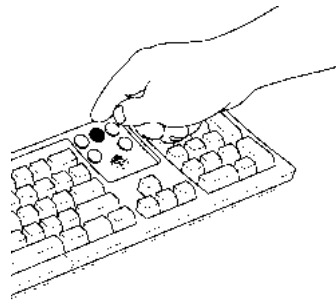


図 18.2 NeXTStation キーボード：本体電源を入れる

²いわゆるテレビの形をした画像を表示する装置のことです。

18.1.3 login する

画面中央には 10 センチ四方程度の枠が表示され、その中に更に二つの小さな枠が取られ、上の枠の左に「Name:」、下の枠の左に「Password:」と表示されていると思います。(図 18.3を参照) キーボードを使ってまずあなたのユーザ名をタイプします。ユーザ名をタイプし終り、³ ← キー³を一度押せば今度はパスワードを入力できる状態になります。ここでパスワードを入力するのですが、パスワードは横から見ている他の人にバレないように、タイプしてもその文字が画面に表示されません。キーを打ち間違えないように注意して入力し、もう一度 ← キーを押します。うまくユーザ名とパスワードを入力できれば画面表示が変わります。(図 18.4を参照) もし失敗すれば「いいえ」と首を振るように中央の枠が左右に移動します。

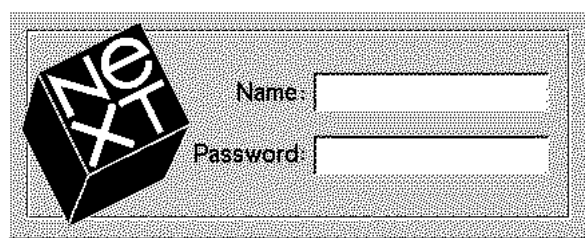


図 18.3 login 画面 :

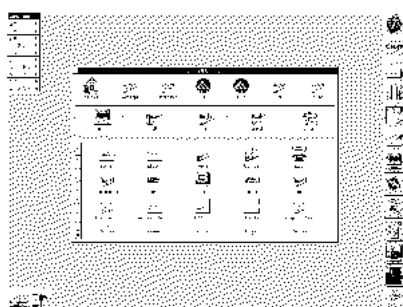


図 18.4 login が成功すると…

もしこれがあなたにとって Sign 登録後初めての login であればパスワードは Sign 登録時にあなたがメモしたものの筈です。何度試しても login 出来ないようであればユーザ名、パスワードのいずれかが間違っているか、まだあなたのユーザ登録作業が計算機センター側で完了していないのです。計算機センターに自分のユーザ名が既に登録されているかどうか確認してください。もしもユーザ名も正しく、また登録作業も完了しているなら、これはパスワードが違っているとしか考えられません。計算機センター相談窓口ご連絡してパスワードを強制変更して貰ってください。

³リターンキーと呼ばれます。

18.1.4 ちょっとメッセージ



図 18.5 login 直後のメッセージ

login すると計算機センターからのお知らせが表示されているかもしれません。よく読んで、自分に関係があるかどうか確認してください。もしも内容についてよく判らないことがあれば計算機センター相談窓口まで連絡するのがいいでしょう。

内容を読み終わったら ← キーを押すか、確認ボタンをクリックするかをすると、メッセージを表示しているウィンドウが消え、通常通り、図 18.5 の画面になります。

18.1.5 基礎知識

NeXT を便利に使う上で知っておいた方がいいことがいくつかあります。前述のマニュアル「ようこそ NeXT...」の「マウスの使用法」と「ウィンドウ」のところを読めば、この後の文章もわかりやすくなりますので、ぜひそちらの方を一度読んでみてください。

18.1.6 パスワード を変える (NeXT に初めて触れるなら...)

もしもこれがあなたにとって Sign 登録後初めての login ならば、ここでまずパスワードを変更しましょう。Sign 登録時のパスワードは機械的に決められたもので、余り安全とは言えません。ぜひパスワードをあなた自身が選んだ単語に変更してください。パスワードを決める際には 4 ページの 2.1.3 で述べている事に注意してください。また、より高い安全性のためパスワードを定期的に変更することをおすすめします。ただ、自分でパスワードをどのように変えたかだけは忘れないようにしてください。

NeXT でパスワードを変更するには Preferences アプリケーションのパスワードボタンを押して行います。詳しくはマニュアル「ようこそ NeXT...」の「パスワードの設定」のところを読んでください。



図 18.6 パスワード変更のウィンドウ

18.1.7 logout する

他にやりたいことも色々あるでしょうが今回は最初の利用でしょうし、まず一通りの操作をやるという意味で、logout をこの場でやってみることを勧めます。

まず画面の左上にあるメニュー（図 18.7）の一番下にある「ログアウト」をクリックしてください。⁴

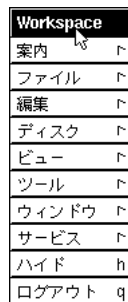


図 18.7 ログアウトのためのメニュー

すると logout パネルが画面に現れます。

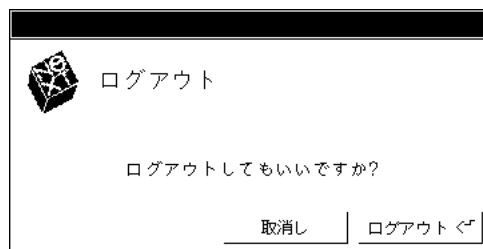


図 18.8 logout パネル

⁴もし、左上のメニューが「Workspace」ではない場合は、画面の右上にある



をクリックしてメニューを切替えてください。

logout パネルは本当に logout してよいかどうかを聞いてきますので、ここで logout してよいならマウスのポインタを「ログアウトする」ボタンの上に持って行ってマウスの左ボタンを押しましょう。

18.1.8 電源を切る

ログアウトしてしばらく待つと再び図 18.3 の login 画面が出てきます。次に、電源を切るために最初に電源を入れたのと同じ図 18.2 のキーボード上の緑の「Power」キーを押してください。

「Power」ボタンを押すと画面に電源 OFF パネル（図 18.9）が表示されますので「電源を Off にする」ボタンを押してください。しばらくすると「NeXTstation 本体の」電源が切れます（画面が真っ暗になる）。そして本体の電源が切れたのを確認したらモニターの電源スイッチを「○」側に押して、モニターの電源を切ってください。

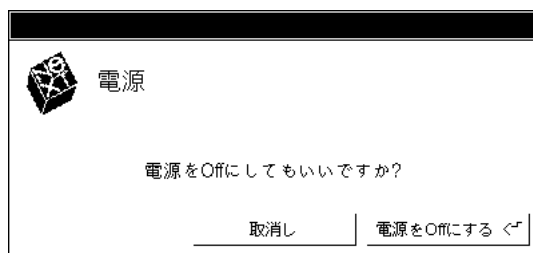


図 18.9 電源 OFF パネル

なお、部屋の入り口に入って正面にある ccns015 と、その左側の ccns013 の 2 台については本体の電源は常に入れっぱなしにしておいてください。その 2 台に関しては誤操作を防ぐため、キーボード上の「Power」キーを効かないようにしていますので、「Power」キーを押して図 18.9 の電源 OFF パネルが出てこなくても慌てないでください。

18.1.9 部屋を出る

マシンの電源が切れたら部屋を出ます。このとき、教室に誰もいなくなるようなら空調機、照明も切って退出してください。

18.1.10 マニュアルなど

NeXTStation に関するマニュアルは、教室に入って左側の書架に各アプリケーションのものも含めてすべて置いています。どれも台数分しかありませんので教室外への持ち出しは絶対しないでください。

18.1.11 さて、それから

NeXTStation は、UNIX コンピュータと呼ぶにはかなり異質な、NeXT 独自の環境が用意されています。勿論普通の UNIX コンピュータとして使うことも出来ませんが、ここでは説明しません。本来の NeXT コンピュータとしての利用方法については前の節で説明したマニュアルの中の、特に「ようこそ NeXT...」と「ユーザーズガイド」などを参照してください。

18.2 NeXT ファイル管理術

18.2.1 はじめに

あなたは NeXT を使っていて、次のようなことで困ったことはありませんか？

- ファイルの量が増えてしまって、すぐに制限を越えてしまう。
- 普段使うアプリケーションがあちこちのフォルダに分散しているので、起動するのが大変面倒である。
- 他のユーザーと同一フォルダを共有したい。
- パソコンとデータ交換をしたい。
- ファイルのバックアップを取りたい。

この章では、みなさんのこのような悩みの解決法を提供したいと思います。NeXT ではマウスとキーボードを使って、簡単にファイルの操作をすることができます。また、ファイルを上手に整理することによって、ディスクの量を節約できますし、セキュリティの確保やデータの有効利用にもつながります。

なお、ここでは NeXT の基本的な操作法（ウィンドウやアイコンの使い方）については述べません。これらは「ユーザーズガイド⁵」に詳しく書かれていますので、そちらを参照してください。

18.2.2 ファイルビューアの各部名称

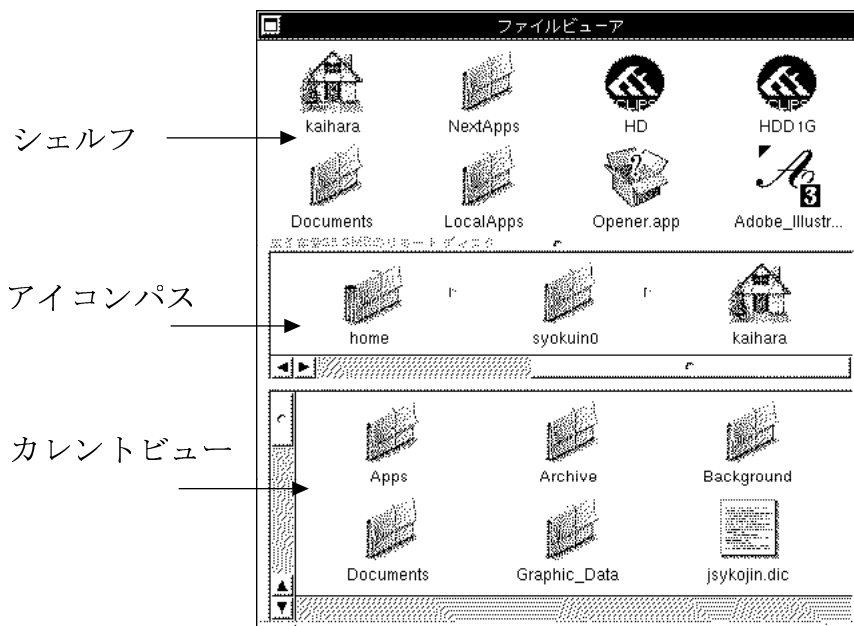


図 18.10 ファイルビューアの各部名称

⁵ C3 情報処理教室の書架に並んでいます。

18.2.3 ファイルとは

NeXT では情報をファイルとフォルダという 2つの形式で管理しています。

ワープロで作成した文章も、グラフィックツールで作成した画像も、そしてアプリケーションの本体もすべてファイルとして管理されています⁶。

18.2.4 フォルダで整理

あなたは、自分のホームフォルダ⁷に多数のファイルができてしまったらどうしますか？ 目的のファイルを探すのは大変でしょうし、関連のある複数のファイルを一括してコピーや移動、削除するにしても苦労をすることでしょう。

そういった場合、UNIX コンピュータではディレクトリというものを使ってファイルを階層的に管理します⁸。

関連するファイルを一つのディレクトリに入れておくことによって、ファイルの管理が大変楽になります。またディレクトリの中に別のディレクトリを入れることもできますので、大分類・中分類・小分類といったように階層的にファイルを管理することも可能です。

なお、NeXT の世界ではディレクトリのことをフォルダと呼びます。呼び方が違うだけで機能や意味するところはディレクトリとまったく同じです。すなわち、ディレクトリと書かれているところをフォルダと読み替えても解釈上は問題ありません。ただ、NeXT の世界ではディレクトリと呼ばずフォルダと呼ぶのが一般的なようなので、ここでも NeXT での話をするときは、フォルダで統一しようと思います。

ファイルビューアを複数開く

ファイルやフォルダの操作をするときに複数のファイルビューアを開けていると、操作がしやすい場合があります。

1. ワークスペースマネージャのメニューから「ビュー」を選択します。
2. 「ビュー」メニューから「新規ビューア」を選択します。

フォルダの新規作成

新しいフォルダを作成します。

1. 中に新しくフォルダを作りたいフォルダ（ややこしい書き方ですみません）をファイルビューア上にオープンします。
2. ワークスペースマネージャのメニューから「ファイル」を選択します。
3. 「ファイル」メニューから「新規フォルダ」を選択します。
4. そのままではフォルダ名が「新規フォルダ」になってしまいますので、適当な名前をキーボードから入力し、↵キーを押します。

ここでフォルダに付ける名前について注意すべき点があります。333ページの 18.2.9の注意事項を参照してください。

⁶詳しくは 96ページの 4.4を参照してください。

⁷家の形をしたアイコンで示されるフォルダ。login 直後にファイルビューアに表示されているフォルダで、あなたのユーザー名と同じ名前がついたフォルダのことです。

⁸詳しくは 111ページの 4.8を参照してください。

18.2.5 コピー

ファイルやフォルダの複製を別のフォルダに作ります。

1. コピー先のフォルダをファイルビューアに表示します。
2. **Alternate** キーを押したまま、コピー元のファイルまたはフォルダをコピー先のフォルダにドラッグします。

このとき、コピー元とコピー先が離れたフォルダ階層にある場合など、2つを同時にファイルビューアに表示できない場合があります。そういった場合は 18.2.4 の方法でもう一つファイルビューアを開いて作業をするとよいでしょう。

複製

「ファイル」メニューの「複製」コマンドを使えば、同じフォルダ内でファイルまたはフォルダのコピーをすることができます。何かファイルに変更を加える前に複製を作成しておけば、状況を変更前に簡単に戻すことができます。

1. 複製したいファイルまたはフォルダを選択します。
2. 「ファイル」メニューの「複製」コマンドを選択します。
3. 元のファイルまたはフォルダの名前の頭に「CopyOf」という文字列を挿入した名前で同じフォルダに複製が作られます。

18.2.6 移動

ファイルやフォルダを他のフォルダに移動します。

1. 移動先のフォルダをファイルビューアに表示します。
2. **Command** キーを押したまま、移動元のファイルまたはフォルダを移動先のフォルダにドラッグします。

このとき、移動元と移動先が離れたフォルダ階層にある場合など、2つを同時にファイルビューアに表示できない場合があります。そういった場合は 18.2.4 の方法でもう一つファイルビューアを開いて作業をするとよいでしょう。


18.2.7 コピーと移動の違い


言葉の意味の違いをそのまま解釈すればよいのですが、コピーの場合、元ファイルはそのまま残されますが、移動の場合には移動元のファイルは削除され、移動先にファイルが作成されます。

18.2.8 ファイルをフォルダにドラッグすると …

キーボードを何も押さずにファイルやフォルダを他のフォルダにドラッグすると、ドラッグ先のフォルダの位置から、自動的にコピーか移動かを選んで実行します。どちらを実行しているかはマウスカーソルの形で分かります。

自動的にコピーをする場合というのは、コピー元とコピー先が異なるディスクにある場合です。異なるディスクというのは、手元のコンピュータのハードディスクから見たときの、フロッピーディスクやネット

トワーク上の他のコンピュータのディスクなどがあたります。コピーをしているときは、コピーカーソル  が表示されます。

自動的に移動する場合というのは、移動元と移動先が同じディスクにある場合です。おそらく同じディスクに同一のファイルは2つもいないという理由からなのでしょう。移動をしているときは、移動カーソル  (緑の矢印型のカーソル) が表示されます。

18.2.9 名称変更

作成済みのファイルまたはフォルダの名前を変更します。

1. アイコンパス (図 18.10を参照) 内のファイルまたはフォルダアイコンの名前部分をクリックすると、「I」の形をした挿入ポイントが名前のところに現れます。
2. 変更したい名前に変更した後、↵キーを押すとその名前に変更されます。

漢字の使用について ファイルやフォルダの名称に漢字⁹を使った場合、cc 環境の他のコンピュータでそのファイルやフォルダの名称を表示させることができない場合があります。ファイルやフォルダの名称に漢字を使うのはやめた方がよいでしょう。

注意

ファイルやフォルダの名称変更をするときは、以下のことに注意してください。

1. ホームフォルダは名称変更できません。
2. アプリケーションを名称変更すると、ファイルをダブルクリックしたときに正常にアプリケーションを起動できなくなることがあります。
3. ファイルの拡張子¹⁰を変更すると、そのファイルをダブルクリックしたときに正常にアプリケーションを起動できなくなることがあります。

18.2.10 削除

リサイクラ

NeXT でファイルまたはフォルダを削除する場合には、リサイクラを利用します。リサイクラは画面右

下にあるアイコン  で示されています。

- 削除したいファイルまたはフォルダをアイコンパスからリサイクラにドラッグします。

注意 このままでは、ドラッグしたファイルやフォルダは、リサイクラ内に保存されたままになっています。削除したファイルやフォルダをいつでも復元できる半面、ディスクを消費しています。

リサイクラ内に格納されているファイルがすべて不要であることを確認できたらリサイクラを空にして、リサイクラ内のファイルやフォルダを抹消してしまいましょう。

⁹ ここでいう漢字とは2バイト文字 (いわゆる全角文字) 全般を示します。

¹⁰ ファイル名の最後につくピリオドとそれに続く文字のことで、ファイルの種類を表すのに使います。

リサイクルを空に

リサイクル内のファイルまたはフォルダが不要になったことが確認できたら、リサイクルを空にして、ファイルやフォルダをディスクから抹消します。

1. 「ファイル」メニューを選択します。
2. 「ファイル」メニューから「リサイクルを空に」を選択します。

復元

リサイクルを空にする前なら、いつでも一旦リサイクルにドラッグしたファイルを復元することができます。

1. 画面右下のリサイクルのアイコンをダブルクリックします。
2. リサイクルウィンドウから、復元したいファイルまたはフォルダをファイルビューア内のフォルダにドラッグします。

18.2.11 リンク

リンクとは

リンクとは、あるファイルまたはフォルダに対するポインタです。リンクにアクセスすると、自動的にリンク先ファイルやフォルダにアクセスします。リンクのアイコンとファイルの内容は、リンク先のファイルやフォルダとまったく同じです。

リンクを上手に利用するとファイルを指定する手間を大幅に省けます。フォルダ階層の離れたファイルやフォルダを指定しなければならないときに、リンクをフォルダ階層の比較的浅いところに作って、それを指定するようにすればいいのです。

また、リンクによってディスクを節約することもできます。気に入ったアプリケーションや画像ファイルなどがあって、自分のフォルダにおいておきたい場合などは、自分のホームフォルダにコピーを作らないでリンクを作ります。リンクは、リンク先のファイルやフォルダを示すだけのものなので、ディスクをほとんど消費しません。

注意 リンクは、リンク先のファイルやフォルダを名前と場所（パスと呼ぶこともあります）で指定します。リンクを移動したり名称変更してもリンクは成立したままですが、リンク元のファイルやフォルダを移動、名称変更、削除するとリンクは無効になります。

壊れたリンクは通常のアイコンにクエスチョンマークが追加されたアイコンで表されます。このようなアイコンが出来てしまった場合は、一旦このリンクを削除して新しくリンクを作り直すのがよいでしょう。

リンク作成

1. リンクを作成するフォルダをファイルビューアに表示します。
2. **Control** キーを押したまま、リンクを作成したいファイルまたはフォルダをリンクを作成するフォルダにドラッグします。

このとき、リンク元とリンク先が離れたフォルダ階層にある場合など、2つを同時にファイルビューアに表示できない場合があります。そういった場合は18.2.4の方法でもう一つファイルビューアを開いて作業をするとよいでしょう。

リンク削除

リンクを削除するときは、通常のファイルやフォルダを削除するのと同様にリサイクルを使います。この場合、リンク先のファイルやフォルダは削除されないで、リンクだけが削除されます。

18.2.12 アクセス権

これまで述べてきたファイルの各操作は、あなたのユーザアカウントがアクセス権を持つファイルやフォルダに対してのみ行うことができます。アクセス権に関する説明は 162 ページの 6.3 でされていますので、必ず参照してください。

アクセス権の確認と変更

NeXT でファイルまたはフォルダのアクセス権を確認、変更するには以下のようにします。

1. ワークスペースマネージャのメニューから「ツール」を選択します。
2. 「ツール」メニューから「インスペクタ」を選択します。
3. ファイルビューアでファイルまたはフォルダを選択します。
4. 「インスペクタ」パネル上部のポップアップリストから「アクセス管理」を選択します。

これでアクセス権を確認することができます。もし、アクセス権を変更したければ、「インスペクタ」パネルの中央にある表のチェックマークを操作して行います。該当する欄にアクセス権を与えたい場合はチェックマークに、アクセス権を与えたくない場合はチェックマークを 1 度クリックして「×」マークに変えてください。

18.2.13 アプリケーションドック

画面の右端にアイコンが一行に並んでいるスペースをアプリケーションドック（略してドックと呼ぶこともあります。）といいます。すでに Mail や Edit などのアプリケーションが登録されていると思いますが、このスペースには頻繁に使用されるアプリケーションのアイコンを置いておくことができます。開いているスペースには自分で任意のアプリケーションを置くこともできますし、不要なアイコンは外すこともできます。

アプリケーションドックへの登録方法

ファイルビューアで目的のアプリケーションを、ドックの任意の空き位置にドラッグします。

アプリケーションをドックから削除する方法

1. ドックから削除したいアプリケーションが実行中であれば、終了します。
2. 削除するアプリケーションのアイコンをドックからドック以外の場所にドラッグします。
3. ドックからそのアプリケーションのアイコンが消えたら、マウスボタンを放します。

ドックからアプリケーションを削除しても、アプリケーション本体は残っていますので、ファイルビューアから他のアプリケーション同様に実行することができます。

18.2.14 シェルフ

シェルフは簡単にファイルやフォルダにアクセスできるように、頻繁に使用するファイルやフォルダを格納しておくためのスペースです。(図 18.10を参照)自由にファイルやフォルダのアイコンを追加したり、削除したりすることができます。

シェルフの利用方法

- ファイルやフォルダをシェルフに登録するには、アイコンパスからシェルフにドラッグします。
- ファイルやフォルダをシェルフから削除するには、そのファイルまたはフォルダのアイコンをシェルフからファイルビューアのウィンドウの外にドラッグします。

シェルフのスペースを拡げるには…。

あれもこれもとシェルフにファイルやフォルダに登録していくと、シェルフのスペースはすぐに一杯になってしまいます。ファイルビューアのウィンドウ自体を横に拡げてよいのですが、それでもあまり多くのファイルやフォルダに登録することはできません。ファイルビューアをカスタマイズすることによって、シェルフのスペースを縦に拡げることができるようになります。

1. ワークスペースマネージャのメニューから「案内」を選択します。
2. 「案内」メニューから「プリファレンス」を選択します。
3. 「プリファレンス」パネル上部のポップアップリストから「シェルフ」を選択します。
4. 中央のやや下に「リサイズ可能シェルフ」という項目が出てきますので、その右側の□をクリックし、チェックマークを付けてください。
5. シェルフの中央下部に「○」の形をしたリサイズノブが表示されるようになります。このノブをドラッグすることにより、シェルフのスペースを調節することができます。

18.2.15 フロッピーディスク

NeXTには3.5インチフロッピーディスクドライブが1基搭載されています。このドライブでは以下のフォーマットのフロッピーディスクを使用することが出来ます。

- NeXTSTEP < 2ED(2.88MB),2HD(1.44MB),2DD(720KB) >
- MS-DOS < 2HD(1.44MB),2DD(720KB) >
- Macintosh < 2HD(1.44MB) >

フロッピーディスクを挿入すると、フロッピーディスクの形をしたアイコンがファイルビューア内に表示されます。ファイルビューアではフロッピーディスクがあたかも一つのフォルダであるかのように扱われます。

フロッピーディスクの初期化

NeXT 上で、NeXTSTEP フォーマット・MS-DOS・Macintosh の各フォーマットで初期化ができます。

1. フロッピーディスクを NeXT 本体右側面のドライブに挿入します。
2. まったく新しいディスクを挿入した場合、ワークスペースマネージャによって、そのディスクを初期化するかどうかの確認を求めるメッセージが表示されます。もし何らかの初期化がすでになされている場合に、そのディスクを再初期化したいときは、18.2.15を参照してフロッピーの再初期化を行ってください。
3. 確認メッセージの下部に表示された「初期化」ボタンをクリックしてください。
4. 「初期化」パネルのポップアップリストからフォーマットの種類を選択します。
5. ディスク名をキーボードから入力します。
6. 「消去」ボタンをクリックします。

フロッピーディスクの再初期化

1. フロッピーディスクを NeXT 本体右側面のドライブに挿入します。
2. ファイルビューアでフロッピーディスクを選択します。
3. ワークスペースマネージャのメニューから「ディスク」を選択します。
4. 「ディスク」メニューから「初期化」を選択します。
5. 「初期化」パネルのポップアップリストからフォーマットの種類を選択します。
6. ディスク名をキーボードから入力します。
7. 「消去」ボタンをクリックします。

注意

フロッピーディスクを一旦初期化すると、記録されていた情報はすべて消去されます。また、一旦消去された情報を復元する方法はありませんので、フロッピーディスクを初期化する際には十分に注意してください。

フロッピーディスクの取り出し

挿入したフロッピーディスクをドライブから取り出します。

1. ファイルビューアでフロッピーディスクを選択します。
2. ワークスペースマネージャのメニューから「ディスク」を選択します。
3. 「ディスク」メニューから「イジェクト」を選択します。

以上の操作をすると、フロッピーディスクがドライブから排出されます。稀に NeXT の不具合により、上記の操作をしてもフロッピーディスクが排出されないことがあります。そのときは計算機センターの相談窓口までご連絡ください。

18.2.16 光磁気ディスク

C3 情報処理教室にある 15 台の NeXT のうち、ホスト名 ccns015 (教室の入口を入れて正面にあるマシン) には 128MB タイプの 3.5 インチ光磁気ディスクが接続されています。NeXT では光磁気ディスクをフロッピーディスクと同じ操作法で利用することができます。また、NeXTSTEP フォーマットの他に Macintosh フォーマットのディスクも使用することができます。初期化やイジェクトについてもフロッピーディスクと同様に行います。

18.2.17 おわりに

ここではごく簡単に、NeXT でのファイル管理術について述べました。もっと多くの機能が NeXT にはありますので、それらについては「ユーザーズガイド」を見てください。

18.3 NeXT で使うネットワークサービス

18.3.1 最初に

これからあなたに NeXT での電子メールとニュースの簡単な使い方をご紹介します。NeXT を使えば他の UNIX マシンよりたいへん手軽にこれらのサービスを利用できるので、「ネットワークサービスってよくわからないけど、なんだかおもしろそうじゃん」という人には NeXT を使ってみることを勧めます。

ネットワークサービスに慣れる一番の方法を紹介しましょう。

1. 「UNIX ガイド」¹¹の以下の節を読む¹²。
 - (a) 124ページの 5.1
 - (b) 128ページの 5.2
 - (c) 151ページの 5.6
2. 自分がネットワークサービスを縦横無尽に駆使している姿を想像する¹³。
3. NeXT の基本的な使い方（メニューやウィンドウ、アイコンの操作法）を知らないのなら、324ページの 18.1に沿って、NeXT の雰囲気（作法？）を学ぶ。
4. この節を自分で実行しながら順番に読んでいく。
5. 実際に自分でバリバリ使っていく。
6. 使ってみてよかったら、友達に紹介して仲間を増やす¹⁴。


こう書いてあると、とっても大変そうですがそんなことはありません。”百聞は一見に如かず”です。あせらずゆっくり進んでいきましょう。

18.3.2 文通しよう

この節では電子メールソフトウェア「Mail」の使い方を紹介します。ここではメールのやりとりの練習と、C3 情報処理教室設置の NeXT 独自のメールに関するルールを中心に述べることにします。

始めよう

まずは「Mail」を起動¹⁵する方法を説明します。画面の一番右側にずらっとアイコン（絵柄）が一行並んでいるのがわかるでしょうか？

そのアイコン群の上から 3 番目にある  のアイコンをダブルクリックすると「Mail」が起動します。「Mail」の画面が現れるまでしばらくかかりますが、そのままお待ちください。

起動すると図 18.11 のようなメール箱（Active.mbox という名前が付いています）ウィンドウがオープンします。はじめて起動したときには、おそらく空っぽだと思いますが¹⁶、心配はいりません。これからどんどんたまっていくだろうあなた宛のメールに思いを馳せましょう。

¹¹ この本のことです。

¹² 量としてはほんの少しですからさっと読めることでしょう。

¹³ 秘訣は強く念ずることです。

¹⁴ 仲間は多ければ多い方が良いのは何をするのでも同じです。

¹⁵ ここでの起動とは、ソフトウェアを実際に使える状態にすることです。

¹⁶ もしかすると友達が先にあなた宛に送っていたメールが届いているかもしれませんね。

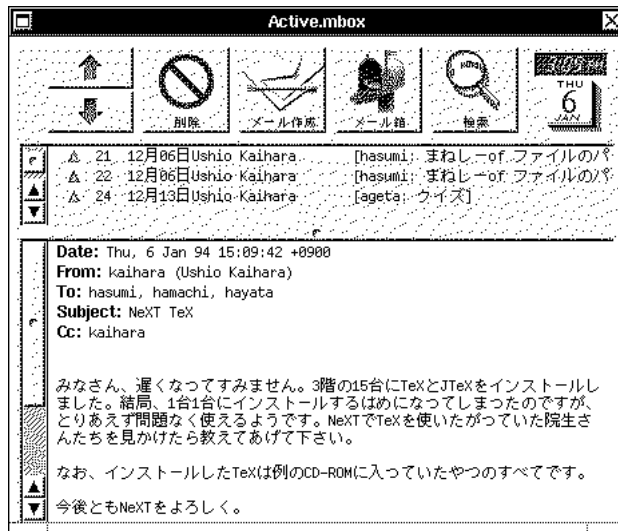




図 18.11 メール箱ウィンドウ

送ろう

まず試しに自分で自分宛に練習メールを送ってみましょう。

最初にメール作成ボタン  をクリックして、メール作成ウィンドウをオープンします（図 18.12 参照）。宛て先のメールアドレス（今は練習ですのであなたのメールアドレスですね）と表題、そして本文を書き込んで、最後に投函ボタン  をクリックします。詳しくは「ユーザーズガイド」¹⁷の 198～199 ページの「メッセージを送信するには」に説明がありますので、それを見ながらやってみてください。

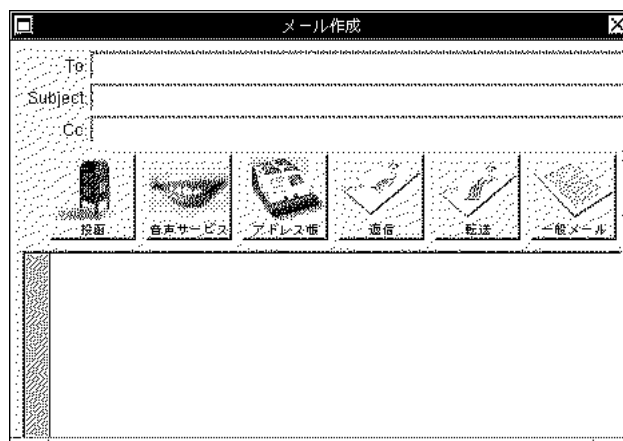


図 18.12 メール作成ウィンドウ

¹⁷ C 3 情報処理教室に入って左側の書架に並んでいます

NeXT ユーザーのメールアドレスについて

もし、あなたのユーザー名が `kaihara` であったとするとあなたのメールアドレスは `kaihara@ccnext.kyoto-su.ac.jp` となります。他の `cc` 環境のマシン群 (SPARCcenter2000 や DEC-3300) でのあなたのメールアドレス `kaihara@cc.kyoto-su.ac.jp` とまちがわないように気を付けましょう。前者のアドレスに送れば NeXT の方に届きますが、もし後者のアドレスに送ってしまえば他の `cc` 環境のマシン群の方に届けられてしまいます。

読もう

では次に、先の章で送った自分宛てのメールを読みたいと思います。おそらくまだメールがメール箱に届いていないので、次の手順で新着メールの取得をして、先ほどのメールを自分のメール箱に入れましょう。

1. 画面左上の Mail メニュー (図 18.13) から「ユーティリティ」をクリックする。
2. 隣に「ユーティリティ」メニューが出てくるので、その中の「新着メールを取得」をクリックする。
3. メール箱に新しいメールが入ってきたら、日付や差出人名などが書いてあるメッセージ行がメール箱ウィンドウに追加されることでわかります。もし、数秒待ってもメール箱の中に自分からのメールが入っていないようならあと数回、上の作業を繰り返してみてください。数回繰り返してみても駄目な場合は、宛先を間違えたか、投函ボタンをクリックし忘れたかのどちらかが考えられますので、もう一度 18.3.2 を読んでトライしてください。





図 18.13 Mail メニュー

先ほどの練習メールはメール箱に入ったと思います。それではそのメッセージ行のどこでも結構ですからクリックしてください。メール箱ウィンドウの下半分にあなたが先ほど書いたメッセージが表示されたでしょうか？表示されればまずは大成功。この練習メールと同じ要領で宛名の部分を他の人のメールアドレスにすると、その人にメールを送ることができます。どうです、試してみたくなってきましたでしょうか？

ところで NeXT でメールを送るときには、そのメールが一般メールか NeXT メールかに注意してください。NeXT メールに関する説明と 2 つのメールの種類の見分け方など、詳しくは 18.3.2 を参照していただくことになりますが、他の UNIX マシンに NeXT メールを送っても、受け取った人は判読ができませんので注意が必要です。

返事を出そう

返事といっても 18.3.2 と同じやり方で宛名と表題を付けて送っても構わないのですが、「Mail」には簡単に返事を出す機能として、自動的に宛名や表題を付けたり、もらったメールの中から引用をしたりする機能があり、それらを利用することによって、あまり手間をかけずに返事を出すことができるようになっていきます。

返信ボタン  と転送ボタン  を使います。

では先ほど自分宛に送ったメールに返事を出してみましょう。先のメールの内容を表示した状態で返信ボタンをクリックしましょう。「メール作成」ウィンドウが表示されますが、中にある「To:」欄に注目してください。すでに送り先のメールアドレスが書き込まれていると思います。ここに書かれたメールアドレスは元のメールから情報をもらって決められます。メールアドレスは長く間違いやすいものが多いので大変助かりますね。また、メールの題名も「Subject:」欄に元のメールの表題の先頭に「Re:」¹⁸を付けたものが自動的に書き込まれます。

もらったメールからの引用をしたい場合ですが、転送ボタンをクリックしてみてください。「メール作成」ウィンドウの中の本文の部分にもらったメール全部がコピーされているのがわかるでしょうか？この中から必要な部分以外を削除すれば引用の完成です。あとの操作は 18.3.2 のメールを送る場合と同じです。本文を完成させ投函ボタンをクリックすれば相手のところへ送られます。

NeXT メールをやりとりしよう

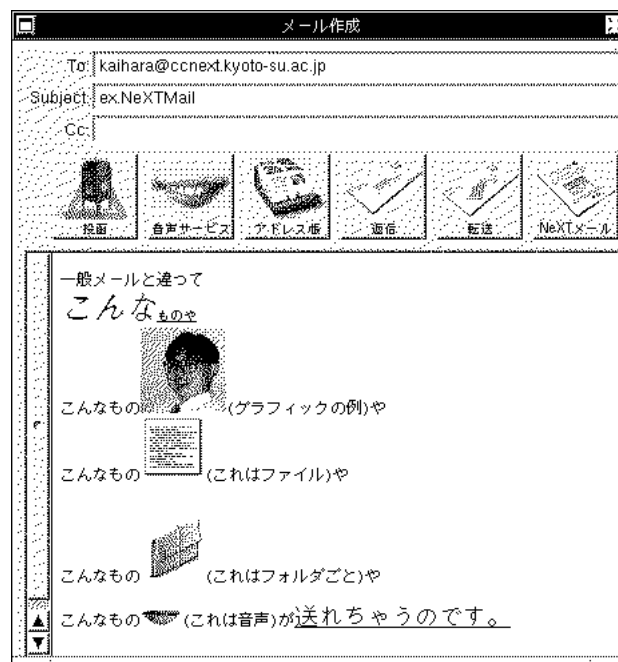



図 18.14 NeXT メール の例


NeXT メール形式のメールでは、文字はもちろん絵や音声を相手に送ることができます。しかもそれらを普通の文字だけのメールとほぼ同じ方法で扱うことができるのです。このように便利なメールですので、

¹⁸ response (返答) の意です

自分が NeXT メールを使える環境にあるということをメール友達に教えておくことをお勧めします。メールを出すときや、後の節で述べるニュースの記事を自分で投稿するときに、シグネチャ（署名）に「NeXT Mail is welcome」などと書いて自分が NeXT メールも使えるということを示しておく、そこから絵や音声を使ったメールのやりとりが始まるかもしれません。しかし便利なこの NeXT メールにもいくつか問題があります。詳しくは「ユーザーズガイド」の 198～199 ページを読んでもらいたいのですが、NeXT メールを NeXT 以外のコンピュータに送っても受け取った方には判読できません。

このことに気を付けて NeXT でメールを使ってほしいのですが、自分が今書いているメールが NeXT メールか、通常のメールかがときどきわからなくなることがあります。そういったときは「メール作成」

ウィンドウの右から 2 番目のボタンを見てください。この位置にあるボタンが  NeXTメール ならばあなたが今

作っているメールは NeXT メールです。もし、 一般メール ならば、他の UNIX マシンにも送ることができる一般メール形式です。また、このボタンを押すことによって一般メールならば NeXT メールに、またその逆といったように、作成中のメールの形式を変更することができます。ただし作成中の NeXT メールの中に絵や音声などが既にある場合は、それらを削除しないと形式の変更ができません。

もっと便利に使おう

「Mail」にはもっと便利な使い方がありますが、ここでは紹介はしません。興味のある人は「ユーザーズガイド」の第 13 章「メールの送受信」と第 14 章の「Mail アプリケーションの管理」をぜひぜひ、ぜひともごらんになってください。

また、電子メールを使っていく上で知っておいた方がよいルールやマナーを、139ページの 5.4で紹介していますので必ず読んでおいてください。

18.3.3 ニュースしよう

この節ではニュースソフトウェア「NewsBase」¹⁹の基本的な使い方を説明します。「NewsBase」は目次の役目をする「NewsBase」と、記事を読んだり書いたりするためのノートである「MMEdit」の 2 つのアプリケーションソフトから構成されています。この両方のソフトウェアを使ってニュースを読みます。

なお C3 情報処理教室では cc 環境でうまく動作するように、変更を加えた「NewsBase」を使っています。難しい変更作業をやり遂げてくれた本学理学部計算機科学科の吉信さんに感謝します。

NewsBase の起動

まず「NewsBase3.02」フォルダの中が見える状態にしましょう。ルートフォルダ²⁰→ LocalApps フォルダへ階層を下りていきます。(図 18.15を参照) このフォルダの中に、先程名前を出した「NewsBase」や「MMEdit」などのアイコンが置かれているのがわかるでしょうか。

次に「NewsBase」を起動します。「NewsBase.app」のアイコンをダブルクリックしてください。すると、図 18.16のパネルが出てきますので、「nntp サーバマシン名:」の欄が「cc2000」になっているかを確認してから²¹、OK ボタンをクリックします

読んでみよう

図 18.17のウィンドウが画面に表示されているはずです。

¹⁹ (株) リクルートのフリーソフトウェアです。

²⁰ ファイルビューアの一番左にある NeXT station のセットの形をしたアイコン

²¹ もし、「localhost」となっているのならば、cc 環境仕様の NewsBase ではなく、別の NewsBase を起動しています。もう一度

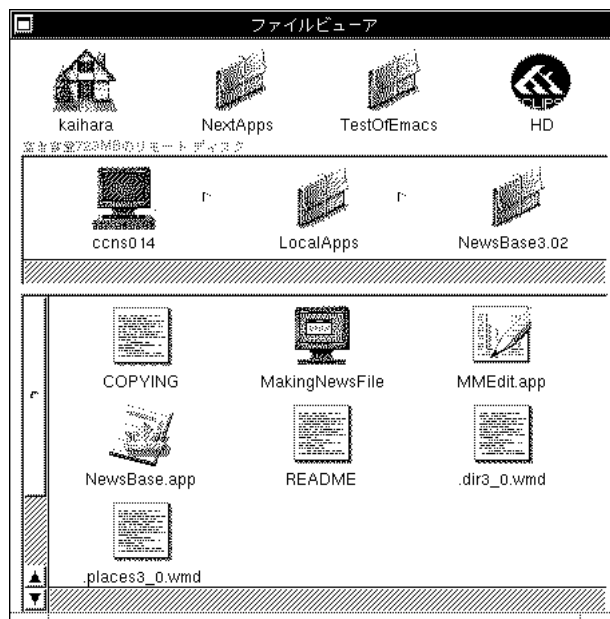


図 18.15 NewsBase の置き場所



図 18.16 サーバーマシン名の設定

ここで試しに「sandai.question」²²というニュースグループを覗いてみましょう。cc2000のアイコンの下に「sandai」という文字がありますね。「sandai.question」は「sandai」の下の階層に属しますので、まず「sandai」の方をクリックしてください。クリックすると右隣のリストの中から「question」を探してください。これが「sandai.question」というニュースグループを表しますので、これをクリックしてください。

パネルが現れて「××個を読み込みます」とたずねてくるかもしれません。もしそのニュースグループに含まれる未読記事の数が多ければ、このようにたずねてくるようになっているのです。

「OK」ボタンをクリックすると「sandai.question」の記事の見出しが図 18.17の NewsBase のウィンドウの右半分にはずらっと表示されます。

この中から自分の読みたいものを選び、見出しをクリックすることによって、自動で「MMEdit」が起動し、選んだ記事を表示します。(図 18.18参照)

他の記事を読みたい場合は、「MMEdit」のウィンドウの下に隠れている（でも一部は見えていると思いますが…）「NewsBase」ウィンドウの内をクリックをすると「NewsBase」のウィンドウが一番手前になりますので、次の読みたい記事をクリックしてください。

最初から丁寧にこの節を読み直して、やり直してください。

²² このニュースグループでは様々な質問がいつも飛び交っています。

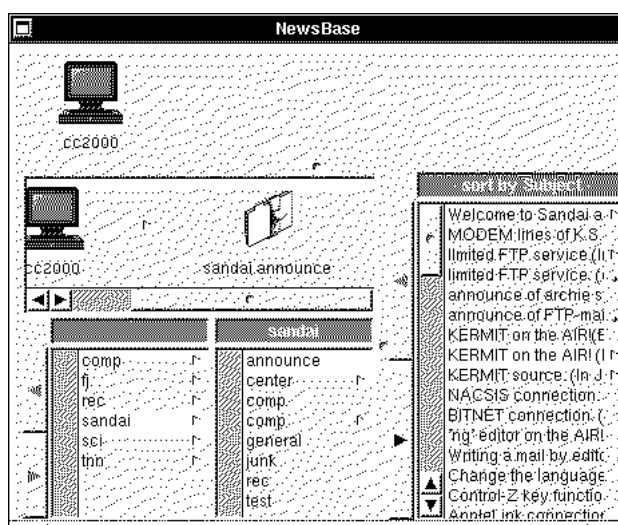


図 18.17 ニュースグループの表示

別のニュースグループを覗きたい場合も同様です。ニュースグループをクリックし直せば、そのグループの記事を表示してくれます。

発言しよう

人の書いているものを読んでいるうちに、自分でも投稿したくなつたのではないのでしょうか?でも、ニュースの雰囲気に慣れないうちに投稿するのはあまり勧められません。たくさんの記事を読んで、マナーやルールになじんでから投稿しましょう。京都産業大学内のニュースグループには投稿テスト用のニュースグループ「sandai.test」があります。投稿の練習にはここを利用するのがよいでしょう。

新しい話題を投稿しよう 画面左上にある「MMEditor」メニューから「ファイル」をクリックし、続いて現れたメニューから「新規作成」をクリックします。ウィンドウが開きますので各欄を設定します。Newsgroup 欄²³と Subject 欄²⁴それから Distribution 欄²⁵に記入してください。あとは記事本文をウィンドウ下半分の白いスペースに書き込んで「MMEditor」メニューから「ファイル」をクリックし、次に「ポスト」をクリックすると投稿がされます。

フォローしよう 他の記事に対して意見を投稿することをフォローといいます。投稿されている記事にフォローをするには、まず元の記事を「MMEdit」に表示させます。(つまり読むってことですね。)表示できたら、画面左上の「MMEditor」メニューから「ファイル」をクリックし、続いて現れたメニューから「フォロー作成」をクリックします。自動的にNewsgroup 欄と Subject 欄が設定されますので、あなたは Distribution 欄と記事本文を書くだけです。記事ができ上がれば、先程の「MMEditor」メニューからまた「ファイル」をクリックし、今度は「ポスト」をクリックすると投稿の完了です。

²³ 記事を投稿するニュースグループの名前(sandai.question や fj.sys.next)を書く欄です。

²⁴ 記事の表題を書く欄です。フォローの場合は自動的に元記事の表題の先頭に「Re:」(Response の略)を付けたものが設定されます。

²⁵ 記事の属性を設定します。sandai ニュースグループ群なら「local」、fj ニュースグループ群なら「fj」、その他のニュースグループ群なら「world」を設定します。

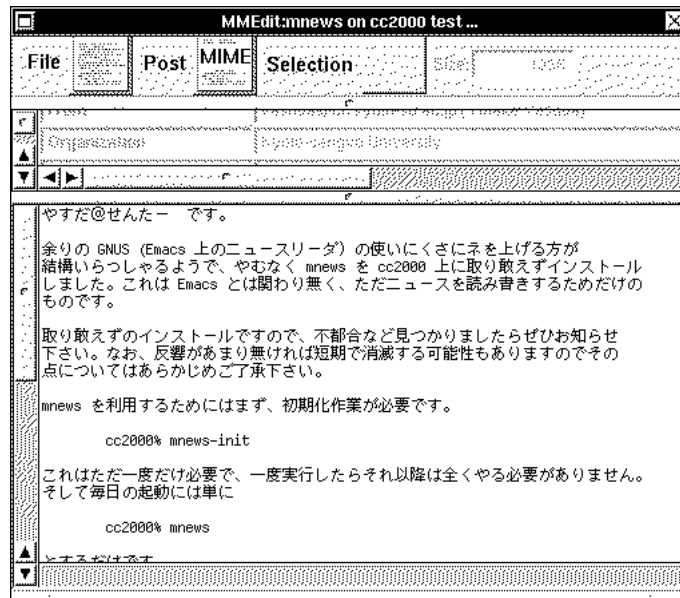


図 18.18 記事を読む

取り消したいときは… 自分で投稿した記事の取り消しをしたい場合があると思います。キーボードの [Control] キーと [Command] キーを押しながら取り消したい記事をクリックすると、自分が投稿した記事なら記事を取り消すことができます。

しかし、取り消せるからといって安易に投稿するのは困りものです。取り消さないで済むように慎重に投稿することを心がけてください。

GNUS と併用する人へ

NewsBase では記事の購読記録ファイルに、他の cc 環境マシンでの標準ニュースリーダーである GNUS の購読記録ファイルと同じものを使っています。従って NeXT で NewsBase を使って読んだ記事は、次に GNUS を使ったときでもちゃんと既読扱いになっていますし、その逆もまたそうです。ただ、NewsBase を使って記事を読むと、ニュースグループの順序がバラバラになってしまいます。(NewsBase はニュースグループを自動で並べかえて表示してくれるので、NewsBase だけを使っていると関係ありません。) 現在のところ、これを根本的に回避する方法はありません。ご了承ください。

また、NewsBase で投稿した記事は GNUS から取り消すことはできません。また、その逆もできませんので注意してください。

お願い

ニュースを使っていく上で知っておいた方がよいルールやマナーを、151ページの 5.6で紹介していますので必ず読んでおいてください。ニュースは一旦投稿してしまうと大変多くの人に読まれることになりま。思いがけない失敗をしないためにもルールやマナーには慎重になってほしいと思います。

18.3.4 最後に

さあ、これであなたもネットワークサービスの達人へ続く階段の一段目を登りました。たくさんの情報が洪水のようにやって来ることと思います。その洪水に溺れることなく上手に情報を使っていきましょう。

これから、もしわからないことができたら電子メールやニュースで質問を投げかけてみましょう²⁶。ネットワークにはおせっかいな人がたくさん住んでいますから、だれかがきっとあなたの質問に答えてくれることでしょう²⁷。

²⁶ でも見知らぬ人にいきなりメールを出すのは考えものですね。

²⁷ でも、必ず答えてくれるという保証ありませんが…。

付録 A

リファレンス

A.1 UNIX コマンド

ここでは UNIX のコマンドのうち、代表的なものを示します。例示は以下のような体裁をとります。

```
command : コマンドの働き (full spec of command)
  command [option] string...
    option      (full spec of option) オプションの働き
      |
      string                引数の意味
      |
```

一行目にコマンド名とその概略を、二行目に実際にコマンドを実行する時のスタイルを、三行目以降に二行目で使われた記号の説明をしています。

以下にそれぞれの意味を説明します。

command	まさにコマンド名です。
コマンドの働き (full spec of command)	そのコマンドがどのような働きをするかを簡単に書いています。 コマンド名はその働きを示す単語の省略形である事が良くあります。その元となった単語です。
option (full spec of option)	そのコマンドに適用できるオプションです。代表的なものだけを書いていきます。 オプション記号はその働きを示す単語の省略形である事が良くあります。その元となった単語です。
オプションの働き	そのオプションがどのような働きをするかを簡単に書いています。
<i>string</i>	そのコマンドに適用できる引数です。
引数の意味	与えられた引数がどのようなものとして解釈されるかを簡単に書いています。

オプションや引数の与え方のスタイルを示している二行目の部分では、[] | { } などの記号を使って表現しています。以下のルールに従って解釈して下さい。

- オプションや引数が [] で囲まれている場合がありますが、これは囲まれた部分はなくともいいと言う事を意味しています。
特にオプション記号が並んでおり、それが [] で囲まれていた場合は囲まれた部分から任意の記号を組み合わせて書いても良い事を示しています。例えば [-abc] と書かれていた場合は -a -b -ab -abc -cb などのいずれの書き方をしても構わないのです。

- オプション記号が並んでいる時に | で仕切られている場合がありますが、これは仕切られた両側の記号のいずれか一つを選ぶ事を示しています。
例えば [-a|-b] と書かれていた場合はオプション無しか -a か -b かいずれかで実行しなければなりません。もしも [-a|-b] ではなく {-a|-b} と書かれていた場合はオプションなしは許されず、-a か -b かいずれかを選ばなければなりません。
- 引数など斜体文字で書かれた部分については、その部分をファイル名やキーワードなど何か別の文字列で置き換えて与える事を意味しています。
特に引数の後ろに ... と書かれていた場合は引数を空白で区切りながら複数個書いても良い事を示しています。例えば *string...* と書かれていた場合は `sample test try` と書いても構わないのです。

A.1.1 ファイル管理に関するコマンド

ls : ファイルの一覧を表示する (list)

```
ls [-lagdF] [names...]
  -l      (long format)  詳細情報を表示する
  -a      (all)          隠しファイルも含めて表示する
  -g      (group)       グループ情報も表示する
  -d      (directory)   ディレクトリ以下をたどらない
  -F      ファイルの属性が判るようにディレクトリには/を、実行ファイルには*をつける
  names   ファイル名
```

pwd : カレントディレクトリの表示 (print working directory)

```
pwd
```

cd : カレントディレクトリの移動 (change directory)

```
cd directory
  directory   移動先のディレクトリ名
```

mkdir : ディレクトリの作成 (make directory)

```
mkdir directory...
  directory   作成するディレクトリ名
```

rmdir : ディレクトリの消去 (remove directory)

```
rmdir directory...
  directory   消去するディレクトリ名
```

cp : ファイルのコピー (copy)

```
cp [-i|-f] source-file destination-file
cp [-i|-f] source-file... destination-directory
cp -r[ilf] source... destination-directory
  -i      (inquiry)   コピー先でファイル名が既存の場合、上書きするかを問い合わせする
  -f      (force)     コピー先でファイル名が既存であっても問い合わせをせずに上書きする
  -r      (recursive) ディレクトリ構造ごと階層的にコピーする
  source-file         コピー元ファイル
  destination-file   コピー先ファイル
  destination-directory コピー先ディレクトリ
  source             コピー元ファイルもしくはディレクトリ
```

mv : ファイルの移動 (move)

```
mv [-i|-f] source-file destination-file
mv [-i|-f] source... destination-directory
  -i      (inquiry)   移動先でファイル名が既存の場合、上書きするかを問い合わせする
  -f      (force)     移動先でファイル名が既存であっても問い合わせをせずに上書きする
  source-file         移動元ファイル
  destination-file   移動先ファイル
  source             コピー元ファイルもしくはディレクトリ
  destination-directory 移動先ディレクトリ
```

rm : ファイルの消去 (remove)

```
rm [-[r] [i] |f]] [name...]  
-r (recursive) ディレクトリ構造ごと階層的に消去する  
-i (inquiry) 消去するかどうか確認する  
-f (force) 消去するかどうか確認しない
```

A.1.2 ファイルに関する雑多なコマンド

cat : ファイルの内容を表示する (catalog)

```
cat [-n] [name...]  
-n (number) 行番号をつける  
name ファイル名
```

more : ファイルの内容を一ページずつ表示する

```
more [name...]  
name ファイル名
```

head : ファイルの先頭を表示する

```
head [-number] [name...]  
-number 先頭から number 行めまで表示する  
name ファイル名
```

tail : ファイルの末尾を表示する

```
tail [-number|+number] [name...]  
-number 末尾 number 行を表示する  
+number number 行めから末尾まで表示する  
name ファイル名
```

file : ファイルの種類を表示する

```
file [name...]  
name ファイル名
```

touch : ファイルの更新日付を新しくする

```
touch [-c] name...  
name が存在していない場合にはその名前で空ファイルを作成する。  
-c name が存在していない場合にはファイルを作成しない。  
name ファイル名
```

od : ファイルの内容をダンプする (octal dump)

```
od [-bcx] [name...]  
-b 8 進数で表示する  
-c 可視コードは文字表示する  
-x (hex) 16 進数で表示する  
name ファイル名
```

split : ファイルを行単位で分割する

```
split [-number] [name [prefix]]  
number 行ごとに name ファイルを分割し、prefix に aa から zz までアルファベット順に合成した番号を付けた複数のファイルとして出力する  
-number 分割する行単位。デフォルトでは 1000 が与えられる  
name 元ファイル名  
prefix 出力ファイルの頭に付く名前
```

du : ファイルの量を表示する (disk use)

```
du [-sk] [name...]  
-s (size) name それぞれの総合計のみを表示する  
-k (kilobyte) 表示単位をキロバイト単位にする  
name ファイル名
```

ln : リンクを作成する (link)

```
ln [-fns] link [name...]  
-f      (force)          リンク先 link が書き込み禁止であっても確認の問い合わせをしない  
-n      (force)          リンク名 link が既存であれば上書きしない  
-s      (symbolic link) シンボリックリンクを作成する  
link    リンク名  
name    ファイル名
```

A.1.3 テキスト処理に関するコマンド

wc : ファイルの単語数などを調べる (word count)

```
wc [-clw] [name...]  
-c      (character)     文字数 (但しバイト数) を数える  
-l      (line)          行数を数える  
-w      (word)          単語数を数える  
name    ファイル名
```

diff : 二つのファイルの差分を表示する (difference)

```
diff [-biw] file1 file2  
-b      (ignore blank)  行末の空白を無視する  
-i      (ignore case)   大文字と小文字を区別しない  
-w      (word)          空白及びタブ文字を無視する  
file1   ファイル名  
file2   ファイル名
```

diff3 : 三つのファイルの差分を表示する (differences of 3 files)

```
diff3 file1 file2 file3  
file1   ファイル名  
file2   ファイル名  
file3   ファイル名
```

sort : ファイルの内容を行単位で順に並べ変えて表示する

```
sort [-cunfr] [name...]  
sort には非常に多くのオプションがある。man sort などして確認するのが良い。  
-c      (check)         正しくソートされているか否かのチェックのみ行なう  
-u      (unique)        同一行を出力しない  
-n      (number)        数値表現として解釈してソートする  
-f      (fold)          大文字と小文字を無視する  
-r      (reverse)       並び順を逆にする  
name    ファイル名
```

grep : パターンをファイルから検索して表示する (general regular expression)

```
grep [-[AB] number] [-cinlv] pattern [name...]  
pattern には正規表現が利用できる。  
-number  マッチした行の前後 number 行を含めて表示する  
-A number (after)   マッチした行の後ろ number 行を含めて表示する  
-B number (before)  マッチした行の前 number 行を含めて表示する  
-c      (count)      マッチした行数を数えるだけ  
-i      (ignore case) 大文字と小文字を無視する  
-n      (number)      マッチした行を行番号とともに表示する  
-l      (line)        少なくとも一つはマッチした行を持つファイルの名前を表示する  
-v      (invert)      マッチしなかった行を表示する  
pattern  検索するキーワード  
name     ファイル名
```

egrep : grep の完全版 (expression grep)

egrep [-cinlv] *pattern* [*name...*]

pattern には全ての正規表現が利用できる。

-c	(count)	マッチした行数を数えるだけ
-i	(ignore case)	大文字と小文字を無視する
-n	(number)	マッチした行を行番号とともに表示する
-l		少なくとも一つはマッチした行を持つファイルの名前を表示する
-v	(invert)	マッチしなかった行を表示する
<i>pattern</i>		検索するキーワード
<i>name</i>		ファイル名

fgrep : grep の高速版 (fast grep)

fgrep [-cinlv] *pattern* [*name...*]

pattern には正規表現は利用できない。簡略化された表現だけを提供する。

-c	(count)	マッチした行数を数えるだけ
-i	(ignore case)	大文字と小文字を無視する
-n	(number)	マッチした行を行番号とともに表示する
-l		少なくとも一つはマッチした行を持つファイルの名前を表示する
-v	(invert)	マッチしなかった行を表示する
<i>pattern</i>		検索するキーワード
<i>name</i>		ファイル名

tr : 文字を置き換える (translate character)

tr [-ds] [*string1* [*string2*]]

string1, *string2* には "\012" などとして 8 進数表記も可能。[a-z] などと a から z の連続した文字を意味する省略記法も可能。

-d	(delete)	標準入力から <i>string1</i> に含まれる文字を削除して標準出力に出す
-s		標準入力の連続する <i>string1</i> を一つの <i>string2</i> に置換して標準出力に出す

sed : ストリームエディタ (stream editor)

sed [-n][-e *script*][-f *scriptfile*] [*name ...*]

-n		デフォルトの出力を抑制する
-e		編集スクリプト <i>script</i> のバリエーションは多数あるのでマニュアルを参照の事
-f		編集スクリプトを <i>scriptfile</i> ファイルから読みとる
<i>name</i>		ファイル名

spell : 単語のスペルチェックをする

spell [-bvx] [*name*]

spell コマンドは非常に高機能だがここでは基本機能のみ載せる

-b	(British)	British 綴りをチェックする
-v	(verbose)	その綴りでスペリングリストに入っていないすべての単語を表示し、その単語から可能性のある派生語が示す
-x		可能性のある語幹を等号 (=) と共に表示する
<i>name</i>		ファイル名

A.1.4 プリンタに関するコマンド

lpr : プリンタに出力する (line printer)

lpr [-P*printer*] [*name...*]

cc 環境でプリンタに出力できる内容は単純なテキストファイルか、PostScript ファイルに限られる。

-P <i>printer</i>	<i>printer</i> で示されるプリンタに出力する
<i>name</i>	ファイル名

lpq : プリント待ちキューの内容を表示する (line printer queue)

lpq [-P*printer*]

-P <i>printer</i>	<i>printer</i> で示されるプリンタの待行列を表示する
-------------------	-----------------------------------

lprm : プリント待ちエントリを消去する (line printer remove entry)

`lpq [-Pprinter] [-] [job...]`

`job` は `lpq` コマンドであらかじめ確認する。

`-Pprinter` `printer` で示されるプリンタのエントリを処理対象にする
`-` 自分が最後に出力したエントリを消去する
`job` `job` で示されるエントリを消去する

A.1.5 アクセス権、アクセス制御に関するコマンド

whoami : 現在の利用者名を表示する (who am i)

`whoami`

id : 現在の利用者名、現在のグループ名などを表示する

`id`

`id` コマンドは各 OS によって相違が見られるので、`man id` などして確認するのが良い。

groups : 所属するグループ名の一覧を表示する

`groups [username...]`

`username` 表示させたい利用者名。デフォルトでは現在の利用者の利用者名が与えられる

newgrp : 新しいグループへのログイン (new group)

`newgrp [group]`

`group` グループ名。デフォルトでは現在の利用者のデフォルトグループが与えられる

chmod : ファイルのアクセス権を変更する (change mode)

`chmod [-R] mode name...`

`-R` (recursive) `name` がディレクトリであった場合は階層的に処理を行なう
`mode` 下部参照
`name` ファイル名

相対指定における `mode` は `{u|g|o|a}{+|-|=}{r|w|x}` となる。

対象	オペレータ	設定内容
u 所有者	+ 追加	r 読みだし
g グループ	- 取消	w 書き込み
o その他の人	= 強制	x 実行
a 全ての人		

絶対指定における `mode` は以下の数値の和となる。

値	対象	設定内容	値	対象	設定内容	値	対象	設定内容
0400	所有者	読みだし	0040	グループ	読みだし	0004	その他の人	読みだし
0200	所有者	書き込み	0020	グループ	書き込み	0002	その他の人	書き込み
0100	所有者	実行	0010	グループ	実行	0001	その他の人	実行

chgrp : ファイルのグループを変える (change group)

`chgrp [-R] group name...`

`-R` (recursive) `name` がディレクトリであった場合は階層的に処理を行なう
`group` 変更したいグループ名
`name` ファイル名

A.1.6 マニュアルに関するコマンド

man : コマンドなどのマニュアルを表示する (manual)

`man [-s section] title`

`man -k keyword...`

`-s section` `title` のセクションを `section` に限定する
`-k keyword` `keyword` にマッチするタイトルの一覧を表示する
`keyword` はコマンド名などの一部でも良い
`title` コマンド名など

whatis : コマンドなどの要約を表示する (what is)

`whatis title...`
title コマンド名などのキーワード

which : コマンドファイルの位置を表示する

`which command...`
command コマンド名

whereis : コマンドファイル、マニュアルファイルの位置を表示する

`whereis command...`
command コマンド名

A.1.7 雑多なコマンド

date : 現在の日付を表示する

`date`

sleep : 実行を保留する

`sleep [time]`
time 待ち時間 (秒)

cal : カレンダーを表示する (calender)

`cal [[month] year]`
オプションを全て省略すると今月のカレンダーを表示する。*year* だけを与えるとその年のカレンダーを表示する。*month* も与えるとその年のその月のカレンダーを表示する。
month 月の指定を 1 から 12 までで与える
year 年の指定を西暦で与える

echo : 引数を表示する

`echo [-n] [argument...]`
-n (no return) 出力に改行文字を加えない
argument 引数

banner : 引数を花文字で表示する

`banner string...`
string 10 文字までの花文字にしたい文字列

clear : 画面を消去する

`clear`

bc : 電卓

`bc [name...]`
name ファイル名。但しファイルの最後には `quit` を `bc` へのサブコマンドとして書く

look : 英単語を辞書検索する

`look [-df] [-tc] string`
-d (dictionary order) 文字、数字、タブ及びスペースが比較される
-f (fold case) 大文字と小文字を区別しない
-tc (termination character) *c* とその後続く文字を無視する
string 検索単語

tee : 標準入力を標準出力とファイルに書く

`tee [-ai] [name...]`
-a (append) ファイルへの出力を追加書きにする
-i (ignore interrupts) 割り込みを無視する
name ファイル名

script : 端末の操作をファイルに記録する

`script [-a] [name]`

`script` コマンドは新しくシェルを起動する。このシェルを終了することによって `script` コマンドも終了する。シェルを終了するには `exit` コマンドを利用すれば良い。

`-a` (append) *name* ファイルに対する出力を追加書きで行なう。デフォルトは上書き
name ファイル名。省略するとデフォルトとして `typescript` が与えられる

df : 利用可能なディスクブロックを表示する (disk free)

`df [directory...]`

`df` コマンドは各 OS によって相違が見られるので、`man df` などして確認するのが良い。

directory *directory* が存在するファイルシステムだけに限定して表示する

yppasswd : パスワードを変更する (YP password)

`yppasswd`

ypchfn : フルネームを変更する (YP change full name)

`ypchfn`

`ypchfn` コマンドは Sun OS 独自のコマンドで、`cc` 環境では `cc2000` でしか動かない。これによってメールの From: 行や `finger` などの表示に現れるローマ字の名前を変更できる。

ypchsh : login シェルを変更する (YP change shell)

`ypchsh`

`ypchsh` コマンドは Sun OS 独自のコマンドで、`cc` 環境では `cc2000` でしか動かない。これによって `login` した後に実行されるシェルを変更できる。この表現が理解できない場合はこのコマンドは実行しない方がよい。変更できるシェルには限りがあり、`cat /etc/shells <return>` などして確認できる。

A.1.8 ファイル圧縮などに関するコマンド

tar : テープ用集積ファイル (tarfile) を扱う (tape archiver)

`tar [-] [c|r|t|u|x] [v] [-b block] [-f device] [name...]`

テープにファイルを書き込む時などに使うが、ブロックサイズなどはテープドライブに依存するので各デバイスなどのマニュアルで確認する事。

<code>-c</code>	(creat)	tarfile を <i>name</i> から作成する
<code>-r</code>	(replace)	tarfile に <i>name</i> を追加する
<code>-t</code>	(title)	tarfile の内容一覧を表示する
<code>-u</code>	(update)	tarfile を更新する。最後に tarfile へ <i>name</i> ファイルを書き込んでから変更があれば tarfile に追加する
<code>-x</code>	(extract)	tarfile からファイル <i>name</i> を抽出もしくはレストアする
<code>-v</code>	(verbose)	現在の状態を表示しながら実行する
<code>-b</code>	(block size)	ファイルのブロック化係数を <i>block</i> にする
<code>-f</code>	(device file)	tarfile を格納するデバイスを <i>device</i> にする
<i>block</i>		ブロックサイズ。デフォルトは 20
<i>device</i>		デバイスファイル名。-で標準入出力、通常のファイル名の指定も可能
<i>name</i>		ファイル名もしくはディレクトリ名

uuencode : バイナリファイルを可視コード文字列に変換する

`uuencode [name] label`

結果は標準出力に出る。*name* を省略すると標準入力から読んだデータを変換する。

name ファイル名
label 変換したファイルを `uudecode` コマンドで戻す時のファイル名

uudecode : uuencode によって変換されたファイルをバイナリファイルに逆変換する

`uudecode [name]`

`uuencode` 時の *label* によって指定されたファイル名で結果が作成される。*name* を省略すると標準入力から読んだデータを変換する。

name ファイル名

compress : ファイルを圧縮する

compress [-cv] [name...]

圧縮されたファイルは *name.Z* という名前で作成され、元の *name* ファイルは消去される。

-c		圧縮結果を <i>name.Z</i> ファイルに作成せずに標準出力に出力する <i>name</i> ファイルは消去されない
-v	(verbose)	圧縮率を表示する
<i>name</i>		ファイル名

uncompress : **compress** コマンドで圧縮されたファイルを復元する

uncompress [-cv] [name...]

name は最後が *.Z* でなければならない。復元されたファイルは *name* から *.Z* が外された名前で作成され、元の *name* ファイルは消去される。

-c		復元結果を <i>name</i> から <i>.Z</i> を取り除いたファイルに作成せずに標準出力に出力する <i>name</i> ファイルは消去されない
-v	(verbose)	圧縮率を表示する
<i>name</i>		ファイル名

gzip : ファイルを圧縮する (GNU zip)

gzip [-cdhlv] [name...]

圧縮されたファイルは *name.z* という名前で作成され、元の *name* ファイルは消去される。

-c		圧縮結果を <i>name.z</i> ファイルに作成せずに標準出力に出力する <i>name</i> ファイルは消去されない
-d	(decompress)	圧縮ファイルを復元する
-h	(help)	オプション一覧を表示する
-l	(list)	圧縮ファイルの内容一覧を表示する
-v	(verbose)	圧縮率を表示する
<i>name</i>		ファイル名

gunzip : **gzip** および **compress** コマンドで圧縮されたファイルを復元する (GNU unzip)

gunzip [-cv] [name...]

name は最後が *.Z* もしくは *.z* でなければならない。復元されたファイルは *name* から *.Z* もしくは *.z* が外された名前で作成され、元の *name* ファイルは消去される。

-c		復元結果をファイルに作成せずに標準出力に出力する <i>name</i> ファイルは消去されない
-d	(decompress)	圧縮ファイルを復元する
-h	(help)	オプション一覧を表示する
-l	(list)	圧縮ファイルの内容一覧を表示する
-v	(verbose)	圧縮率を表示する
<i>name</i>		ファイル名

zcat : **compress** コマンドで圧縮されたファイルを表示する (cat Z file)

zcat [name...]

name は最後が *.Z* でなければならない。gzip 圧縮ファイルを扱える場合もある。gunzip 同様のオプションが使える場合もある。

<i>name</i>	ファイル名
-------------	-------

A.1.9 プロセスに関するコマンド

ps : 現在処理中のプロセス一覧を表示する (process)

ps [[-]aux]

ps コマンドはオプションもその表示も各マシン、その OS 種類によって大きく異なる。man ps などして確認するのが良い。

a	(all)	全ての利用者のプロセスを表示する
u	(user)	利用者情報指向に整形して表示する
x		端末制御を持たないプロセスも含めて表示する

kill : プロセスを終了させる

`kill [-l] [-signal] process-id...`

process-id についてはあらかじめ `ps` コマンドで確認しておく。 *signal* の種類によっては終了ではなくプロセスの再起動などが行なわれる場合がある。

<code>-signal</code>	<i>process-id</i> によって示されるプロセスに対して送られるシグナルの種類。
<code>l</code>	<i>signal</i> に利用可能な記号の一覧を表示する デフォルトでは <code>-15 (-TERM)</code> が与えられ、大抵これで終了させられる。強制終了の為に <code>-9 (-KILL)</code> を与える
<i>process-id</i>	終了させたいプロセスの番号

A.1.10 現在使っているコンピュータに関するコマンド

tty : 端末回線名を表示する (tele type terminal)

`tty`

hostname : ホスト名を表示する (host name)

`hostname`

uname : OS に関する情報を表示する

`uname [-apsv]`

<code>-a</code>	(all)	全ての情報を表示する
<code>-p</code>	(processor)	プロセッサ型を表示する
<code>-s</code>	(operating system)	OS 名を表示する。これはデフォルトで与えられる
<code>-v</code>	(version)	OS のバージョンを表示する

uptime : 起動されてからの時間と CPU 負荷率を表示する

`uptime`

A.1.11 利用者に関するコマンド

who : 現在利用している利用者の一覧を表示する

`who [-Hq]`

<code>-H</code>	(header)	見出しを出力する
<code>-q</code>		簡略化された形式で出力する

w : 現在利用している利用者と作業内容の一覧を表示する

`w [-hls] [username]`

見出しに `uptime` コマンドで表示されるのと同じ CPU 負荷率などが表示される

<code>-h</code>		見出しを表示しない
<code>-l</code>	(long format)	長い出力形式。これはデフォルトで与えられる
<code>-s</code>	(short format)	短い出力形式
<i>username</i>		利用者名 <i>username</i> に関する情報だけに限定する

finger : 利用者情報を表示する

`finger [-lms] [keyword...]`

`finger [-l] [username]@hostname...`

`finger` コマンドによって表示される最後にメールを読んだ日付については `cc` 環境では正しく表示されない場合がありますので無視して下さい。

<code>-l</code>	(long format)	長い出力形式
<code>-m</code>		<i>keyword</i> を利用者名に限定して検索する
<code>-s</code>	(short format)	短い出力形式
<i>keyword</i>		利用者名、氏名などの断片
<i>username</i>		利用者名。省略した場合は <i>hostname</i> コンピュータを利用している利用者一覧が表示される
<i>hostname</i>		リモートコンピュータのホスト名

whois : 利用者情報を表示する (who is)

whois [-h *hostname*] *keyword*

whois サービスの内容や使い方はそれが行なわれているサーバに大きく依存する。

-h *hostname* *hostname* コンピュータを whois サーバとする
keyword 検索するキーワード

A.1.12 ネットワークサービスに関するコマンド

telnet : TELNET プロトコルによるリモートログインを行なう

telnet [*hostname*]

hostname リモートログインするホスト名

rlogin : リモートログインを行なう (remote login)

rlogin [-8] [-l *username*] *hostname*

-8 (8 bit) 通信に 8 ビットを利用する。デフォルトは 7 ビットの可能性が高い。

-l *username* リモートログイン用の利用者名として *username* を使う

デフォルトでは現在の利用者名が与えられる

hostname リモートログインするホスト名

rsh : リモートマシンにコマンドを実行させる (remote shell)

rsh [-l *username*] *hostname* *command*

-l *username* リモートログイン用の利用者名として *username* を使う

デフォルトでは現在の利用者名が与えられる

hostname *command* を実行させるホスト名

command 実行させたいコマンド行

rpc : リモートマシンのファイルをコピーする (remote copy)

rpc [[*username*@]*hostname*:]*source-file* [[*username*@]*hostname*:]*destination*

rpc -r [[*username*@]*hostname*:]*source...* [[*username*@]*hostname*:]*destination-directory*

コピー元、先のファイル名の記述の先頭に *hostname*: を付加する事によって「:」以降に記述されているファイルはそのホストに存在する事を意味する。*hostname*: を省略すればデフォルトとして現在のホスト名が与えられる。更に *username*: を与える事によって「@」以降に記述されているホストに対するアクセスは利用者名 *username* で行なわれる事を意味する。*username*: を省略すればデフォルトとして現在の利用者名が与えられる。

-r (recursive) ディレクトリ構造ごと階層的にコピーする

username @以降に記述されるホストでの利用者名

hostname :以降に記述されるファイルが存在するホスト名

source-file コピー元ファイル

destination コピー先ファイルもしくはディレクトリ

destination-directory コピー先ディレクトリ

source コピー元ファイルもしくはディレクトリ

A.1.13 シェル (tcsh) のサブコマンド

cd, which, kill などシェルの子コマンドだが、これらは普通のコマンドとして紹介している。ここでの記述は tcsh 特有のものを含んでいる事に注意。

シェル変数、環境変数に関するサブコマンド

set : シェル変数を定義する

`set [variable[=string]]`

引数なしで現在定義されているシェル変数を表示する。*variable* を与えながら *string* を省略すると *variable* で示されるシェル変数を空文字列とする。

variable シェル変数名
string 文字列

unset : シェル変数の定義を解除する

`unset variable`

variable シェル変数名

setenv : 環境変数を定義する (set environment variable)

`setenv [variable [string]]`

引数なしで現在定義されている環境変数を表示する。*variable* を与えながら *string* を省略すると *variable* で示されるシェル変数を空文字列とする。

variable 環境変数名
string 文字列

unsetenv : 環境変数の定義を解除する (unset environment variable)

`unsetenv variable`

variable 環境変数名

ジョブ制御に関するサブコマンド

jobs : 現在実行中のコマンドの一覧を表示する

`jobs [-l]`

`l` (long) プロセス番号も表示する

fg : 停止中のジョブを再開する (fore ground)

`fg [%job]`

job jobs コマンドで表示されたジョブ番号

bg : 停止中のジョブをバックグラウンドで再開する (back ground)

`bg [%job]`

job jobs コマンドで表示されたジョブ番号

stop : バックグラウンドで実行中のジョブを一時停止する

`stop [%job]`

job jobs コマンドで表示されたジョブ番号

notify : バックグラウンドで実行中のジョブの状態変化を知らせる

`notify [%job]`

job jobs コマンドで表示されたジョブ番号

wait : 全てのバックグラウンドジョブの実況終了を待つ

`wait`

`%job` 番号の指定方法

<code>100</code>	プロセス番号 100 番
<code>%1</code>	ジョブ番号 1 番
<code>%</code>	直前に操作したジョブ
<code>%-</code>	一つ前のジョブ
<code>%cc</code>	実行コマンドが <code>cc</code> で始まるジョブ
<code>/?sort</code>	実行コマンドに <code>sort</code> を含むジョブ

プロセス番号は `ps` コマンドもしくは `jobs -l` コマンドで確認できる
ジョブ番号は `jobs` コマンドで確認できる

雑多なサブコマンド

alias : コマンドの別名を定義する

`alias [name [string]]`

引数なしで現在定義されているエイリアス一覧を表示する。`name` を与えながら `string` を省略すると `name` で定義されているエイリアスを表示する。

`name` エイリアス名
`string` 定義する文字列

unalias : エイリアスの定義を解除する

`unalias name`

`name` エイリアス名

rehash : コマンド参照の為の内部ハッシュテーブルを更新する (re-assign hash table)

`rehash`

unhash : コマンド参照の為の内部ハッシュテーブルを使わなくする (unuse hash table)

`unhash`

login : login シェルを終了し、新たに login する

`login`

logout : login シェルを終了する

`logout`

exit : シェルを終了する

`exit (expr)`

`expr` で与えられた数値は `$status` シェル変数に与えられる

`expr` 数値もしくは数値になる式。() は省略できる

exec : コマンドを実行する (execute)

`exec name`

`name` を実行する。実行が終っても制御は返ってこない。

`name` コマンド名もしくは実行可能なファイル名

source : 実行するコマンドの指定をファイルから読む

`source name`

ファイル名 `name` に書かれたコマンドを現在のシェルで実行する。

`name` ファイル名

history : 実行したコマンドの履歴を見る

`history [-hr] number`

ファイル名 `name` に書かれたコマンドを現在のシェルで実行する。

`h` イベント番号を表示に付けない

`r` (reverse) 履歴から最新の `number` 個のイベントを逆順に表示する

`number` 表示するイベントの数

A.1.14 索引

コマンド	頁 (章節)	コマンド	頁 (章節)	コマンド	頁 (章節)
alias	360 (A.1.13)	jobs	359 (A.1.13)	split	350 (A.1.2)
banner	354 (A.1.7)	kill	357 (A.1.9)	stop	359 (A.1.13)
bc	354 (A.1.7)	ln	351 (A.1.2)	tail	350 (A.1.2)
bg	359 (A.1.13)	login	360 (A.1.13)	tar	355 (A.1.8)
cal	354 (A.1.7)	logout	360 (A.1.13)	tee	354 (A.1.7)
cat	350 (A.1.2)	look	354 (A.1.7)	telnet	358 (A.1.12)
cd	349 (A.1.1)	lpq	352 (A.1.4)	touch	350 (A.1.2)
chgrp	353 (A.1.5)	lpr	352 (A.1.4)	tr	352 (A.1.3)
chmod	353 (A.1.5)	lprm	353 (A.1.4)	tty	357 (A.1.10)
clear	354 (A.1.7)	ls	349 (A.1.1)	unalias	360 (A.1.13)
compress	356 (A.1.8)	man	353 (A.1.6)	uname	357 (A.1.10)
cp	349 (A.1.1)	mkdir	349 (A.1.1)	uncompress	356 (A.1.8)
date	354 (A.1.7)	more	350 (A.1.2)	unhash	360 (A.1.13)
df	355 (A.1.7)	mv	349 (A.1.1)	unset	359 (A.1.13)
diff	351 (A.1.3)	newgrp	353 (A.1.5)	unsetenv	359 (A.1.13)
diff3	351 (A.1.3)	notify	359 (A.1.13)	uptime	357 (A.1.10)
du	350 (A.1.2)	od	350 (A.1.2)	uudecode	355 (A.1.8)
echo	354 (A.1.7)	ps	356 (A.1.9)	uuencode	355 (A.1.8)
egrep	352 (A.1.3)	pwd	349 (A.1.1)	w	357 (A.1.11)
exec	360 (A.1.13)	rcp	358 (A.1.12)	wait	359 (A.1.13)
exit	360 (A.1.13)	rehash	360 (A.1.13)	wc	351 (A.1.3)
fg	359 (A.1.13)	rlogin	358 (A.1.12)	whatis	354 (A.1.6)
fgrep	352 (A.1.3)	rm	350 (A.1.1)	whereis	354 (A.1.6)
file	350 (A.1.2)	rmdir	349 (A.1.1)	which	354 (A.1.6)
finger	357 (A.1.11)	rsh	358 (A.1.12)	who	357 (A.1.11)
grep	351 (A.1.3)	script	355 (A.1.7)	whoami	353 (A.1.5)
groups	353 (A.1.5)	sed	352 (A.1.3)	whois	358 (A.1.11)
gunzip	356 (A.1.8)	set	359 (A.1.13)	ypchfn	355 (A.1.7)
gzip	356 (A.1.8)	setenv	359 (A.1.13)	ypchsh	355 (A.1.7)
head	350 (A.1.2)	sleep	354 (A.1.7)	yppasswd	355 (A.1.7)
history	360 (A.1.13)	sort	351 (A.1.3)	zcat	356 (A.1.8)
hostname	357 (A.1.10)	source	360 (A.1.13)		
id	353 (A.1.5)	spell	352 (A.1.3)		

A.2 UNIX でよく使われる記号など

ここでの記述は `tcsh` 特有のものを含んでいる事に注意。

A.2.1 シェル変数の一覧

<code>argv</code>	シェルスクリプトに渡された引数の列。
<code>status</code>	直前のコマンド終了時の返り値を示す。
<code>cwd</code>	Current Working Directory. カレントディレクトリを示す。
<code>home</code>	ホームディレクトリを示す。
<code>path</code>	コマンドパスを示す。 <code>set</code> サブコマンドで設定可。
<code>user</code>	現在のシェルプロセスのユーザ名を示す。
<code>uid</code>	現在のシェルプロセスのユーザ id を示す。
<code>gid</code>	現在のシェルプロセスのグループ id を示す。
<code>term</code>	現在利用しているターミナル種別を示す。 <code>set</code> サブコマンドで設定可。
<code>tty</code>	現在利用しているターミナルの回線番号を示す。
<code>prompt</code>	シェルプロンプトの形式を示す。 <code>set</code> サブコマンドで設定可。
<code>autologout</code>	セットした場合、この秒数の間入力がなければ <code>tcsh</code> は自動的に終了する。
<code>ignoreeof</code>	セットした場合、端末から C-d を読んでもシェルは終了しない。終了には <code>exit</code> を使う。
<code>noclobber</code>	セットした場合、シェルは既存のファイルに出力をリダイレクトする事を許さない。この設定を無視してコマンドを実行するには <code>!</code> を使う。
<code>noglob</code>	セットした場合、 <code>*</code> , <code>?</code> などのワイルドカードによるファイル名の補間を行なわない。
<code>nomatch</code>	セットした場合、ファイル名の補間に失敗してもエラーとせずコマンドを起動する。
<code>verbose</code>	セットした場合、シェルはエイリアス、コマンド、ファイル名、変数などの置換えをした後のコマンドを表示しながら実行する。
<code>history</code>	コマンド履歴の最大数を示す。 <code>set</code> サブコマンドで設定可。
<code>histchars</code>	履歴置き換え文字を示す。未設定の場合 <code>!</code> を使う。
<code>savehist</code>	ファイルに残すコマンド履歴の最大数を示す。 <code>set</code> サブコマンドで設定可。
<code>shell</code>	現在のシェルを示す。
<code>tcsh</code>	<code>tsh</code> のバージョンを示す。
<code>version</code>	現在の <code>tcsh</code> のバージョンを示す。

A.2.2 環境変数の一覧

<code>PATH</code>	コマンドパスを示す。 <code>path</code> シェル変数と連動している。
<code>HOME</code>	ホームディレクトリを示す。
<code>PWD</code>	カレントディレクトリを示す。
<code>SHELL</code>	現在のシェルを示す。
<code>HOST</code>	ホスト名を示す。
<code>HOSTTYPE</code>	ホストコンピュータの種別を示す。
<code>LOGNAME</code>	現在のシェルプロセスのユーザ名を示す。
<code>USER</code>	現在のシェルプロセスのユーザ名を示す。
<code>LANG</code>	言語環境を示す。
<code>TERM</code>	現在利用しているターミナル種別を示す。 <code>term</code> シェル変数と連動している。
<code>MANPATH</code>	<code>man</code> コマンドが検索するマニュアルファイルの置き場所を示す。
<code>DISPLAY</code>	X ウィンドウアプリケーションの表示画面先を示す。
<code>EDITOR</code>	標準のエディタを示す。
<code>PAGER</code>	標準のページャを示す。
<code>PRINTER</code>	標準のプリンタ名を示す。
<code>TEXFONTPATH</code>	<code>T_EX</code> のフォントディレクトリを示す。
<code>TEXTFMPATH</code>	<code>T_EX</code> のフォントディレクトリを示す。
<code>ARCH</code>	ホストコンピュータのアーキテクチャを示す。 <code>cc</code> 環境特有。
<code>ENVIRON</code>	ホストコンピュータの環境種別を示す。 <code>cc</code> 環境特有。
<code>XENVIRON</code>	ホストコンピュータの X ウィンドウの環境種別を示す。 <code>cc</code> 環境特有。
<code>COMMON</code>	共通設定ディレクトリを示す。 <code>cc</code> 環境特有。

A.2.3 リダイレクション記号など

<	標準入力をファイルから読む
<< string	文字列 string が入力行の先頭に現れるまで標準入力を読む
>	標準出力をファイルに書く
>>	標準出力をファイルに追加書きする
>&	エラー出力をファイルに書く
>>&	エラー出力をファイルに追加書きする
>!	標準出力をファイルに書く (noclobber シェル変数による保護を無視)
>>!	標準出力をファイルに追加書きする (noclobber シェル変数による保護を無視)
>&!	エラー出力をファイルに書く (noclobber シェル変数による保護を無視)
>>&!	エラー出力をファイルに追加書きする (noclobber シェル変数による保護を無視)
	標準出力をパイプに書く
&	エラー出力もパイプに書く

リダイレクション記号などを使ったコマンド実行の例

command	通常のコマンド実行。
command &	バックグラウンドでのコマンド実行。
command1 ; command2	command1 の実行が済めば command2 を実行する。
(command1 ; command2)	command1 ; command2 に同じ。但し単一コマンドのようにシェルは扱う。
command1 command2	通常のパイプ付き実行。
command1 & command2	エラー出力を含めたパイプ付き実行。
command1 && command2	command1 の実行が成功すれば command2 を実行する。
command1 command2	command1 の実行が失敗すれば command2 を実行する。
(command > outfile) >& errorfile	標準出力とエラー出力を分ける。

A.2.4 ファイル指定のワイルドカードなど

*	任意のゼロ個以上の文字
?	任意の一文字
[characters]	[] に囲まれた文字列 characters に含まれる任意の一文字
[char1-char2]	文字 char1 から char2 までの範囲に含まれる任意の一文字
{string1,string2,...}	文字列 string1 もしくは string2 などのいずれか
.	カレントディレクトリ
..	カレントディレクトリの一つ上の階層のディレクトリ
~	自分のホームディレクトリ
~username	ユーザ名 username のホームディレクトリ

A.2.5 コマンド履歴を扱う為の表記法

!!	直前のコマンド行
!n	n 番目のコマンド行
!-n	n 番前のコマンド行
!string	string から始まる最近のコマンド行
!?string	string を含む最近のコマンド行
!\$	直前のコマンド行の最後の単語
!*	直前のコマンド行の 1 番めから最後の単語 (つまりコマンド以外の全ての引数)
!n:\$	n 番目のコマンド行の最後の単語
!n:^	n 番目のコマンド行の最初の単語
!n:m	n 番目のコマンド行の m 番目の単語
!n:m-1	n 番目のコマンド行の m 番めから 1 番目の単語
!n:*	n 番目のコマンド行の 1 番めから最後の単語 (つまりコマンド以外の全ての引数)
^str1^str2^	直前のコマンドの str1 を str2 に置き換える

A.2.6 正規表現

a	a (通常の文字) にマッチする
.	任意の一字にマッチする
^	行頭にマッチする
\$	行末にマッチする
^.....\$	5文字の行にマッチする
	前後の正規表現のいずれかにマッチする
ab cd	ab もしくは cd のどちらかにマッチする
ab cd ef	ab、cd もしくは ef のいずれかにマッチする
(ab cd)(12 34)	ab12 ab34 cd12 cd34 のいずれかにマッチする
[abc]	abc のどれか一字にマッチする
[^abc]	abc のどれか一字以外にマッチする
[a-z]	a から z までの範囲のどれか一字にマッチする
[a-hxyz0-9]	abcdefghxyz0123456789 のどれか一字にマッチする
?	?直前の正規表現のゼロ個ないしは一個にマッチする
ab?c	ac abc にマッチする。abbc などにはマッチしない
*	*直前の正規表現のゼロ個以上の繰り返しにマッチする
ab*c	ac abc abbbc などにマッチする。ab1c などにはマッチしない
a.*c	ac abc abbbc a123c などにマッチする。
+	+直前の正規表現の一個以上の繰り返しにマッチする
ab+c	abc abbc abbbc などにマッチする。ac ab1c などにはマッチしない
a.+c	abc abbc abbbc a1c などにマッチする。ac にはマッチしない
\{num\}	直前の正規表現の num 個の繰り返しにマッチする
\{num,\}	直前の正規表現の num 個以上の繰り返しにマッチする
\{num1,num2\}	直前の正規表現の num1 個から num2 個までの繰り返しにマッチする

\ ^ \$. [] () | * + ? は普通の文字ではなく、意味を持ったメタキャラクタである。メタキャラクタに使われている記号をそのまま表現したい時は \\$ などのように \ 記号に続けて書く。 \ 記号を表現したい時は \\ である。

A.3 Mule コマンド

A.3.1 絶対覚えておいた方がいいもの

mule <return>	Mule を起動する
C-x C-c	Mule を終了する
C-h T Japanese <return>	Mule(日本語) のチュートリアルを表示する
C-x C-f <i>filename</i> <return>	ファイルを読み込む
C-x C-w <i>filename</i> <return>	ファイル名を変更して保存する
C-g	指示途中のコマンド操作を取り消す
C-l	カーソルのある行を中央へ移動する
C-x u または C-_	直前の編集操作を取り消す (Undo)
M-x goto-line <return>	指定した行にジャンプする

カット & ペースト

C-<Space>(または C-@)	カーソルの位置にマークをセットする
C-x C-x	カーソルの位置とマークの位置を入れ替える
C-w	マークの位置からカーソルの前までを記憶して消去する (カット)
M-w	マークの位置からカーソルの前までを記憶する (コピー)
C-y	記憶した文字列をカーソルの位置に挿入する (ペースト)
C-k	カーソルの位置から行末までを消去する

検索

C-s <i>String</i>	カーソル位置より下方向に向かって検索する
C-r <i>String</i>	カーソル位置より上方向に向かって検索する
C-s	次を検索する
C-r	前を検索する
C-g	検索を終了しカーソルを検索開始前位置に戻す

Wnn

C-\	Wnn を起動/終了する
<Space>	変換する、次候補を表示する (C-n でも可)
C-p	前候補を表示する
<return>	変換文字を確定する
C-o	文節を伸ばす
C-i	文節を縮める
C-f	右の文節へ移動する
C-b	左の文節へ移動する

カーソル操作

		先頭	M-<			
		前ページ	M-v			
		1行上	C-p			
行頭	1語前	1字前	↑	1字後	1語後	行末
C-a	M-b	C-b	← →	C-f	M-f	C-e
			↓			
		1行下	C-n			
		次ページ	C-v			
		末尾	M->			

A.3.2 必要に応じて覚えるもの

起動時

mule <i>filename</i> <return>	Mule を起動して <i>filename</i> をバッファに読み込む
mule -q	.emacs を無視して Mule を起動する
mule -u <i>usr_name</i>	<i>usr_name</i> の人の .emacs の設定で Mule を起動する

ファイル操作

C-x i <i>filename</i> <return>	別ファイルをカーソルの位置に差し込む
C-x C-s	カレントバッファを保存する
C-x s	すべてのバッファを保存する
C-x k	カレントバッファを保存せずにクローズする
C-x C-v	カレントバッファにファイルを読み込む (カレントバッファの内容はクローズする)
C-x b	バッファを切替える
C-x C-b	バッファリストを表示する
C-x C-q	書き込みモードを変更する
M-x recover-file <return>	autosave された内容を読み込む

コードの変更

C-x C-k d	画面入出力コード変更
C-x C-k i	キーボードからの入力コード変更
C-x C-k f	ファイルの入出力コード変更

その他

全角、半角文字

M-x zenkaku-region リージョン範囲を全角にする

M-x hankaku-region リージョン範囲を半角にする

確定後の再変換

M-x henkan-region リージョン範囲を変換する。

M-x gyaku-henkan-region リージョン範囲を変換し直す。

文字の入れ換え

C-t カーソルの位置の文字とその左の文字を入れ換える

繰り返し

C-u *n Command* *Command* を *n* 回繰り返す

または M-*n Command* 例: C-u 5 C- アンドゥ5回

カーソル操作

M-a 文の先頭へ移動する

M-e 文の末尾へ移動する

カーソルの位置

M-x what-line <return> 今カーソルが何行目にあるか表示する

C-x l 全部の行数と現在のカーソル位置を表示する

改ページ

C-q C-l 改ページ文字[^] Lを入力する

消去、カット & ペースト

M-x kill-rectangle <return> マークセット位置からカーソル位置までのブロックを消去する

M-x clear-rectangle <return> マークセット位置からカーソル位置までのブロックを空白に置換する

M-x yank-rectangle <return> 消去したブロックをカーソル位置に挿入する

置換

M-% *search-string* <return>*change-string* <return>

検索文字列を置換文字列に確認しながら置換する

<Space>または y で置換を行なう

<Delete>または n で置換を行なわない

! で残り全部を確認せずに置換を行なう

^ で一つ前にもどる

M-x replace-string <return>*search-string* <return>*change-string* <return>

検索文字列を置換文字列にすべて置換する

画面分割

C-x 2	上下に二分割する
C-x 3	左右に二分割する (Emacs では C-x 5)
C-x o	カーソルを別ウインドウに移動する
C-x 1	カーソルのあるウインドウ以外のウインドウを隠す
C-x 0	カーソルのあるウインドウを隠す

ウインドウのリサイズ

C-x ^	カーソルのあるウインドウを縦方向に拡大する
C-x }	カーソルのあるウインドウを横方向に拡大する

バッファリスト

C-x C-b	バッファ一覧を表示する
?	バッファリスト簡易ヘルプを表示する
f	カーソルの行のバッファをウインドウに表示する
1	カーソルの行のバッファだけをウインドウに表示する
q	バッファ一覧を終了する

ヘルプ

C-h	ヘルプを呼び出す
C-h C-h C-h	ヘルプオプションとその説明を表示する
C-h k <i>Command</i>	<i>Command</i> の引数説明を表示する
C-h a <i>String</i>	<i>String</i> を含むコマンドの一覧を表示する
C-h b	現在のキー割当を表示する

オンラインマニュアル

C-h i	オンラインマニュアルを起動する
q	オンラインマニュアルを終了する
m	メニューを選択する
u	前のメニューに戻る
<Space>	続きを読む
<Delete>	前に戻る
n	次の項目に進む
p	前の項目に戻る
d	オンラインマニュアルの最初のメニューに戻る

Wnn

変換

M-h	(変換途中で) ひらがなにする
M-k	(変換途中で) カタカナにする
q	アルファベット入力モードにする
C-q	アルファベット入力モードをやめる
C-k または C-c	変換をキャンセルする (ひらがなに戻る) (C-f C-b で前後して訂正可能)

変換候補の一覧

M-s	変換候補一覧をエコー行に表示する
C-n	次の一覧部分を表示する
C-p	前の一覧部分を表示する

記号の入力

C-^	記号一覧のメニューをエコー行に表示する
	0. JIS 入力
	1. 記号
	2. 英数字
	3. ひらがな
	4. カタカナ
	5. ギリシャ文字
	6. ロシア文字
	7. 罫線
	8. 部首入力
	0. 画数入力
	a. 第一水準
	b. 第二水準
	c. 補助漢字
C-n	次の一覧部分を表示する
C-p	前の一覧部分を表示する

単語登録

M-x toroku-region	リージョン (矩形) 指定した単語を登録する
M-x edit-dict-item	単語を登録した辞書を編集する

その他

x?	小さい文字を出す (例: 「xa」で「あ」)
z?	特殊記号を出す (例: 「z(」で「【」)

z を用いた記号の入力

	1	2	3	4	5	6	7	8	9	0	-	=	
	q	w	e	r	t	y	u	i	o	p			↵
	a	s	d	f	g	h	j	k	l	;	'		
		z	x	c	v	b	n	m	.	,	/		
													'

z+

	○	▽	△	□	◇	☆	◎	⊕	♂	♀	~	≠	
	《	》	々	/									↵
		,	,	◇	-	←	↓	↑	→				'
													'

z+Shift

	●	▼	▲	■	◆	★	£	×	⊞	.	.	±	
	<	>	全	§						↑		⊞	

図 A.1 z キーとの組合せによる記号

MHE

読む

M-x mh-rmail <return>	MHE メールリーダーを起動する
q	MHE メールリーダーを終了する
.	メールを読む
<Space>	続きの部分を読む
<Delete>	前の部分を読む

返事

a	メールに返事を書く
C-c C-y	メールの内容を引用する
C-c C-c	返事を送信する
C-c C-q	返事を送信するのをやめる

出す

M-x mh-smail <return>	メールを書く
C-c C-c	メールを送信する
C-c C-q	メールを送信するのをやめる

GNUS

M-x gnus <return>	GNUS を起動する
q	GNUS を終了する
ニュースグループ選択画面	
<Space>	カーソル位置のニュースグループを読む
c	すべて既読にする
u	次の起動から表示しないようにする
L	すべてのニュースグループを表示する
記事画面	
<Space>	カーソル位置のニュースを読む
<Space>	記事の続きを読む
q	ニュースグループ選択画面に戻る
<Delete>	記事の前の部分を読む
d	記事に既読マークを付ける
f	表示中の記事に対してフォローする
F	表示中の記事に対して引用付きでフォローする
a	投稿のための原稿を編集する
C-c C-c	(フォロー、投稿記事編集画面で) 記事を投稿する
o	記事を保存する

Cモード

M-x c-mode	Cモードにする
M- C-\ (または M-x indent-region)	リージョンの範囲をインデントする

コンパイル

M-x compile <return>	コンパイラを起動する
----------------------	------------

.emacs の設定例

```
(setq enable-double-n-syntax t)
```

「nn」で「ん」と変換するようにします。

```
(load "/NF/local/general/lib/mule/19.28/lisp/its/hira.el")
(its-defrule "string" "string2")
但し、emacs の場合は
(defrule "string" "string2")
```

string をローマ字入力すると string2 になるようにします。

例

```
(load "/NF/local/general/lib/mule/19.28/lisp/its/hira.el")
(its-defrule "dhi" "でい")
(its-defrule "thi" "てい")
```

```
(setq-default case-fold-search nil)
```

case-fold-search という検索時に大文字小文字を区別するかどうかのデフォルト値を設定します。この場合は全てのバッファにおいて区別します。

```
(global-set-key "\C-x@" 'compile)
```

C-x @のキー操作に対して compile のコマンド操作を割り当てます。因みに M- C-a なら” /e/C-a” と表します。

```
(autoload 'gnus "gnus" "Read Network News" t)
```

起動時に gnus 関数を自動的に読み込みます。

```
(setq kill-whole-line t)
```

ただし emacs の場合は

```
(defun kill-line-twice (&optional numlines)
  "Acts like normal kill except kills entire line if at beginning"
  (interactive "p")
  (cond ((or (= (current-column) 0)
             (> numlines 1))
         (kill-line numlines))
        (t (kill-line))))
(global-set-key "\C-k" 'kill-line-twice)
```

通常、一行削除は行頭で C-k を 2 回行なう必要がありますがこれで一回で済むようになります。

A.4 京都産業大学 FAQ(抄)

A.4.1 目次

- はじめに (374ページから)
 - FAQってなんですか？
 - このFAQリストの最新版(全部)はどうしたら得られますか？
 - FAQに載っていないトラブルなんです、どうすれば良いですか？
 - 補助員ってなんですか。
- UNIX 編
 - 各種のコマンドに関すること (375ページから)
 - 21 情報処理教室の csosf でフロッピーを使うには？
 - コマンドの実行結果をプリントアウトしたいのですが？
 - grep の使い方、正規表現が良く分かりません。
 - 実行中のプロセスの終了
 - LINKって何？
 - テキストのファイルがぐちゃぐちゃで読めません
 - Mule に関すること (378ページから)
 - mule でのかな漢字変換で「nn」で「ん」と変換するにはどうすれば良いのですか？
 - メールに関すること (378ページから)
 - メールを相手を読んだかどうかを確認したいのですがどうすればよいですか？
 - シグネチャを付けるにはどうすれば良いのでしょうか？
 - 相手のメールアドレスが判らないのですが、調べる方法がありますか？
 - ニュースに関すること (379ページから)
 - ニュースグループのソートの仕方 (GNUS)
 - ニュースを読んでいたら、ニュースグループの頭に「*」が付いてしまいました。偶然の産物なので、消し方を知りません。どうすれば消えるのでしょうか。
 - 時々相手が文頭や文末に自分のことを「安田@計算機センターです」などというように書いていますが、これはどういう意味ですか？
 - シグネチャを付けるにはどうすれば良いのでしょうか？
 - ニュースに記事を投稿したのですが、うまく投稿できたかどうかを確認するには、どうすれば良いですか？
 - いろんなニュースグループがありますが、それぞれどんなものなのですか？
 - クロスポストってなんですか？
 - Followup-To: ってなんですか？
 - X に関すること (382ページから)
 - リモートログインした機械でアプリケーションを立ち上げようとする Can't open display が出てきます。
 - それ以外のこと (382ページから)
 - パーミッションって何ですか？
 - ディレクトリを他の人からも見られるようにしたいんですが
 - キーボードを打っても文字が化ける、あるいは何も表示されずまともに動かないのですが
 - 印刷したいのですが、どのプリンタを指定したら良いのでしょうか？
 - 家のパソコンとデータをやりとりしたいのですが
 - 21 情報処理教室の DEC3300 でセッション休止から復帰できません

- Mac 編 (384ページから)
 - マックにリセットスイッチはついていないのですか？
- Program 編 (384ページから)
 - math.h を使ったらコンパイルできない。
- その他 (384ページから)
 - フロッピーディスクを買いたいのですが、

A.4.2 はじめに

Q. FAQってなんですか？

A. 何度も繰り返される質問をまとめたQ & A集です。

Frequently asked question の略です。何度も同じ質問が出ると答える方も疲れるし、質問する方も気が引けてくるので、まずこれを見ることでそれを解決しようという意図があります。ここでは学内にある計算機的环境に対するさまざまな質問に答えています。

Q. この FAQ リスト (の最新版) はどうしたら得られますか？

A. 次の内からお選び下さい。

- 計算機センター相談窓口 (計算機運用補助員待機場所) まで取りに行く。
- mosaic の各種ドキュメントにあるFAQを見る。
- sandai.question に定期的に投稿されるものを見る。

Q. FAQ に載っていないトラブルなんです、どうすれば良いですか？

A. 計算機について分からないことがあれば、各情報処理教室にあるコードレス電話で2578をダイヤルして下さい。計算機運用補助員というナイスガイ&ナイスギャル達がお答えします。また、お急ぎでない場合や、専門的な質問は京都産業大学ローカルニュースグループの sandai.question に投稿して下さい。誰か知っておられる方からフォローが入るでしょう。

Q. 補助員ってなんですか。

A. アルバイトの学生によるコンピュータ環境のお助け部隊です。

皆さんは学内のコンピュータ施設を使っていて、何か困った事があったことはないでしょうか。プリンタから印刷されてこない、コンピュータが使用中に止まってしまった、このアプリケーションの使い方が分からない、等々。そんなときにはぜひ補助員を呼んでください。

補助員とは正確には「計算機運用補助員」という名前で、計算機センターで学内のコンピュータのトラブルに対応するために待機している、その方面の知識を持った学生達のアルバイトです。

補助員は学内で授業が行われている間中、複数の人間が勤務しており、少なくとも一人は計算機センターが管理している11・C1・C3・21・31・51・52情報処理教室、それぞれの部屋で何かトラブルはないかと巡回を続けています。また少なくとも一人は計算機センターで常時待機しており、いつでもコンピュータ施設のトラブルに対する電話を受け付けています。あなたがもしどこかの情報処理教室でトラブルに遭遇した時、その部屋にタイミング良く補助員がいれば解決しますし、いなくても、各情報処理教室に設置されている内線電話を使って補助員に連絡すればかけて問題が解決してもらえるというわけです。

補助員を呼ぶには先ほども書いたように各情報処理教室に設置されている内線電話を使って「2578」をプッシュしてください。すると「計算機センター相談窓口」というところにつながり、補助員がトラブルの状況を聞いてきます。もしそれが簡単な対処法で解決するような問題ならば電話で補助員が解決方法の指示を出します。そうでなければ補助員が何らかの方法で解決しますので指示を聞いて下さい。

また補助員は年度の初めに新たにスタッフの募集をしております。コンピュータをやってみようという意欲のある方、応募をお待ちしています！

— 計算機運用補助員 —

業務内容：計算機センターが管理しているコンピュータ施設のトラブル全般の解決

業務期間：学内で通常講義のある期間全て

連絡方法：各情報処理教室に設置してあるコードレスホンで2578をプッシュしてください

A.4.3 UNIX 編

各種のコマンドに関すること

Q. 21 情報処理教室の csosf でフロッピーを使うには？

A. fdio コマンドを使用します。

メディアとしては 2ED, 2HD(1.44), 2DD(720) が利用できます。(2ED というのは最近東芝などが作っている 2.88MB 容量のものです。ワークステーションは最近良く使っていたりします。)
通常は 2HD と 2DD の DOS フォーマットを扱います。fdio コマンドを利用して下さい。

```
csosf03(81)% fdio
```

```
DOS Floppy Read/Write Utility  Version 1.2
```

```
fdio>
```

というところですかね。どのようなサブコマンドが利用できるかは help サブコマンドで調べることが出来ます。

```
fdio> help
```

```
binary    dir        cd          read
text      type      mkdir(md)  write
bye       pwd       rmdir(rd)  ren
quit      help(?)   find       del
```

```
fdio> help dir
```

```
dir -- ディレクトリの一覧を表示する.
```

```
使い方: dir [-awl] [dos-directory or dos-file ... ]
```

終了するには quit コマンドを使用します。

```
fdio> quit
```

UNIX ファイルをそのままフロッピーに詰めたい場合は tar コマンドを使うのがいいのでしょうね。デバイス名は /dev/rz5c です。

Q. コマンドの実行結果をプリントアウトしたいのですが？

A. いくつか方法があります。

97ページ 4.4.3に script コマンドが載っています。例えば、

```
cc2000(121)% script filename
```

とすると exit を実行するまでのことが filename という file に書き込まれます。これは画面に出力される事がそのまま書き込まれます。

169ページ 6.4.4にリダイレクトを利用する方法があります。

```
cc2000(127)% a.out > filename
```

とすると実行結果が filename という file に出力されます。

他には、既に実行されて画面に結果が表示されているならば、それを Cut&Paste で

```
cat > filename
```

を使って filename のファイルにする事も出来ます。

後は lpr コマンドでプリンターに出力して下さい。

Q. grep の使い方、正規表現が良く分かりません。

A. grep はある文字列を含む行の一覧を出力するコマンドです。たとえば、

```
% grep hello sample.txt
```

または

```
% cat sample.txt | grep hellow
```

とすれば、sample.txt ファイル中の”hello”という文字列が出現する行が全部画面（正確には標準出力）に出てきます。

文字列にはワイルドカード（「*」や「?」）も使用可能です。

サポートしている正規表現は

^ : 行頭に Match
 \$: 行末に Match
 . : CR (改行 Code) 以外の任意の 1 文字に Match、2 Byte Code も Document の Script に応じ 1 文字として認識

です。このように使います。

例：
 1: sample document 12345
 2: 12345
 3: 123456789 this is a test
 4: abcdefg9

grep '^123' で、2: 3: がヒット (先頭から 123 があるもの)
 grep '123' 1: 2: 3: (123 があるもの)
 grep '9\$' 4: (9 で終るもの)
 grep '9' 3: 4: (9 を含むもの)
 grep 'a...e' 1: 4: (a に続いてどんな文字でもいいから 3 文字あって、e があるもの)
 grep 'ae' 全くヒットしない (ae があるもの)
 grep 'a*e' 全くヒットしない (ae, aae, aaaa, ... を含むもの)
 grep 'a.*e' 1: 3: 4: (a に続いてどんな文字でもいいから 何文字かあって (ゼロ文字でも良い) e があるもの)

Q. 実行中のプロセスの終了

A. ps コマンドと kill コマンドを使用します。

とりあえず ps コマンドを実行します。すると以下のようなものがでできます。

```
cc2000(82)% ps
  PID TT      S   TIME COMMAND
  7977 pts/41  S   0:00 -tcsh
  8004 pts/41  0   0:00 ps
```

これで目的とする物がでてこないなら次のようにします。但しこれは cc2000 の場合ですのでそれ以外の機種は下記の表に対応するオプションを指定して下さい。

```
cc2000(83)% ps -aux | grep ozaki
ozaki      8012 80 1072 751 pts/25  S 18:06:34 0:06 emacs faq
ozaki     10034 50 380 317 pts/13  0 18:45:57 0:00 ps -aux
ozaki     3449243 259 236 pts/25  S 16:52:12 0:01 -tcsh
ozaki     10035 7 167 137 pts/13  S 18:45:57 0:00 grep ozaki
ozaki     9905189 259 230 pts/13  S 18:43:19 0:01 -tcsh
```

ホスト名	OS	オプション	備考
cc2000	Solaris2	-axu	/usr/ucb/ps の場合 /bin/ps の場合は -ef
csosf??	OSF/1	-ef	
ccns0??	BSD4.2	-axu	
--	標準的な SVR4	-ef	
--	標準的な BSD	-axu	SunOS なら -xu で良い

このようにすると全てのプロセスから ozaki さんのプロセスを表示します。パイプより前の部分でシステム中の全プロセスをリストして、パイプより後ろの部分で前半の結果から自分のユーザ名 (例では ozaki) の文字列を含む行だけ抜き出して表示させている訳です。これで自分のプロセスが表示されます。ですから上記の ozaki を自分のユーザ名に置き換えると自分のプロセスを表示してくれます。

そして一番左の数字がプロセス番号ですので、殺したい (終了させたい) プロセス番号 (PID) を指定した kill コマンドを実行します。

```
cc2000(83)% kill 8004
```

これで、もし止まらない場合はもう少し強力にして止めるオプションをつけます。


```
cc2000(83)% kill -1 8004
```

単なる kill (実は kill -15 と同じ), kill -1, kill -2, kill -9 の順に強力になりますので、順に試してください。パイプに関しては 171 ページ 6.4.5 を参照して下さい。

Q. LINK って何?

- A. LINK にはシンボリックリンクとハードリンクという 2 種類が存在します。シンボリックリンクとは一つ本物のファイル (ディレクトリ) があつた時、それを指し示すファイル (ディレクトリ) を作る事をいいます。ハードリンクとは一つのファイルにもう一つ名前をつけてやる事をいいます。通常使うのはシンボリックリンクの方です。リンクの利点は主に二つあります。一つはディレクトリ構造の離れた所にあるファイルを指定してアクセスし易くする事。もう一つはコピーと違ってファイルの実態は一つなのでディスク容量を節約することが出来ることです。後、プログラムのコンパイル等でライブラリの指定をリンクで行って行けば、それを交える事でライブラリをを簡単に切り換える事もできます。

```
% ln -s file1 file2
```

ここで file1: 元のオリジナルファイル、file2: 元ファイル (file1) を指し示すようにしたいファイル名です。この場合 file2 を編集しようとするとき自動的に file1 を編集している事になるという関係になります。

また、この時に file1 を消すと file2 は指し示すファイルが無くなるので名前だけ残って使えないファイルになってしまいます。

File の本体	file1 を消す	File の本体
↑	⇒⇒⇒⇒⇒⇒	
file1 ← file2		??? ← file2

それに対してハードリンクは

```
% ln file1 file2
```

と実行します。この場合、file1 と file2 の立場は同等で、実体は同じで名前を二つ持っている状態になります。この時は file1、file2 のどちらかを消してももう一つの側でファイルにアクセスすることが出来ます。

File の本体	File の本体
↑ ↑	↑
file1 file2	file2

Q. テキストのファイルがぐちゃぐちゃで読めません

A. ファイルの文字コードを変換します。

漢字やひらがな等のアルファベット 2 文字分の大きさのある文字 (全角文字) は規格によりその文字のコード (文字を記号とみなした通し番号) が違うので違う規格で読むと全然違う文字が表示されてしまいます。主に本学環境では EUC コードを用いていますがメールやニュースでは JIS コードも用いられています。cc 環境では nkf という Network Kanji code conversion Filter があります nkf コマンドを使ったコード変換は次のようにして行ないます。

1. ファイルを一旦別の名前にして、
2. nkf で 目的のコードに変換し、それを 元のファイル名に書き出し、
3. 別の名前にしておいたファイルを消去する

というところです。いかに具体的な手続きを。

```
1. cc2000(20)% mv filename1 filename2
```

2. 目的に合わせて次の何れかを選んで下さい。
それぞれ、j(JIS) e(EUC) s(ShiftJIS) に変換します。

```
cc2000(21)% nkf -j filename2 > filename1
```

```
cc2000(21)% nkf -e filename2 > filename1
```

```
cc2000(21)% nkf -s filename2 > filename1
```

```
3. cc2000(22)% rm filename2
```

Mule に関すること

Q. mule でのかな漢字変換で「nn」で「ん」と変換するにはどうすれば良いのですか？

A. .emacs に一行書き加えます。

まず

```
cc2000(01)% mule .emacs
```

として下さい。これで.emacs という mule の環境設定ファイルを読み込み mule が起動します。今まで何も変更していない人は

```
;;  
;;      Emacs common settings.  
;;
```

```
(load "/NF/home/common/settings/_emacs.load")
```

このような内容になっていると思います。

ここで、(load... の行の次の行に

```
(if (boundp 'MULE)  
    (progn  
      (setq enable-double-n-syntax t)))
```

を書いて下さい。

```
;;  
;;      Emacs common settings.  
;;
```

```
(load "/NF/home/common/settings/_emacs.load")
```

```
(if (boundp 'MULE)  
    (progn  
      (setq enable-double-n-syntax t)))
```

このようになりますね。これを保存して終了します。すると次に mule を起動した時には「nn」で「ん」が出るはずですが。

なお、emacs ならば上記の代わりに

```
(defrule "nn" "ん")
```

と書いて下さい。

メールに関すること

Q. メールを相手を読んだかどうかを確認したいのですがどうすればよいですか？

A. 自動的に相手を読んだら確認できるようなシステムはありません。

もし必要ならメールに読んだら返事をしてくれるよう書いておけば良いでしょう。

また、NeXT メールなら受領証という機能がありますのでそれで確認することができます。

Q. シグネチャを付けるにはどうすれば良いのでしょうか？

A. あらかじめファイルを作っておいてそれを挿入します。

前もってシグネチャをファイルに書いて作っておきます。そして必要に応じてその内容を取り込みます。そうすると毎回タイプする必要がなくなりますね。

mule を使ってメールを書いている人を対象に説明します。例えば sig というファイル名で 1～4 行程度のシグネチャを作っておきます。そしてメールを書き終わった時に C-x i と (コントロールキーを押しながら x キーを押して、コントロールキーを離して i キーを押す) すると、下に Insert file: ~/ の様なメッセージが出ると思います。出たら、そのメッセージに続けて sig とタイプします。

```
Insert file: ~/sig
```

となりましたか？ になったらリターンキーを押して下さい。無事にシグネチャがカーソルのあった位置へ書き込まれたと思います。なお、余り長いシグネチャはマナーに反しますので 4 行までにしましょう。

Q. 相手のメールアドレスが判らないのですが、調べる方法がありますか？

A. 基本的に調べる方法はありません。

ただし cc 環境ならば whois コマンドを利用して調べる方法があります。

ニュースに関すること

Q. ニュースグループのソートの仕方 (GNUS)

A. 直接順番を入れ換えるか、ファイルをソートします。

ニュースグループ選択画面のニュースグループを編集して行を入れ変えるとその順番になります。

```
:
9: sandai.comp.announce
4: sandai.comp
1: sandai.general
11: sandai.rec
:
```

例えばこのようなニュースグループの並びになっていて、sandai.rec を sandai.comp の上に持ってきたい、というような場合は、まずカーソルを sandai.rec の行の先頭 (1 カラム目) に移動させます。ここで C-k (コントロールキーを押しながら k) を二回実行すると sandai.rec の行が消えます。消えたら今度は sandai.comp の行の先頭にカーソルを移動させて C-y を実行します。

```
:
9: sandai.comp.announce
11: sandai.rec
4: sandai.comp
1: sandai.general
:
```

こうなりましたね。これで今後もこの順序でニュースグループが表示されます。他にはコマンドラインからソートする方法もあります。

`sort <元ファイル名> -o <出力ファイル名>`

でアルファベット順にソートできます。そしてニュースグループが書いてあるファイルは `~/newsrcc-cc2000` なので、自分のホームディレクトリで

```
cc2000(20)% sort .newsrcc-cc2000 -o .newsrcc-cc2000
```

というコマンドを実行してやればニュースグループをソートすることが出来ます。

!!! やってはいけないこと !!!

```
cc2000(20)% sort .newsrcc-cc2000 > .newsrcc-cc2000
```

こういう書き方は一見出来そうな気がするのですが、`> .newsrcc-cc2000` の部分で書き込み場所を先に確保してしまうので `.newsrcc-cc2000` が初期化され無くなってしまいます。無くなった物をソートしても何も残るはずは無く結果、`.newsrcc-cc2000` というファイルは中身なしのファイルになります。

Q. ニュースを読んでいたら、ニュースグループの頭に「*」が付いてしまいました。偶然の産物なので、消し方を知りません。どうすれば消えるのでしょうか。

A. gnus で読んでいる時に、u を押してしまったのではないのでしょうか？

記事を読んでいる時に u を押すと、mule の上の画面のその NG の記事一覧の一番左にあるマークが - になって、既読で記事を消去するのを一時保留します。(普通は、ここが D になって q で抜けると、D マークがついた記事は既読として消去されてしまいますよね。) それで、NG 一覧の画面では、そういうファイルが NG の中にあると、その横に*をつけて、以上の処理をした記事がありますよと、知らせます。だから、もういちどその記事を読んだことにしてやるために記事の場所で 'd' を押すと既読マークがつくので、すべて解決となるはずですよ。

Q. 時々相手が文頭や文末に自分のことを「安田@計算機センターです」などというように書いていますが、これはどういう意味ですか？

A.

これは「安田」さんが「計算機センター」の人ですという意味です。要は「@」マークが区切りとなっていて、個人@所属を表します。こういう書き方の元はメールアドレスからきています。メールアドレスは honyarara@cc.kyoto-su.ac.jp という風になっていますね。これは honyarara さんが jp(日本)の ac(大学・研究機関)の kyoto-su(京都産業大学)の cc(ホストマシン名)の人ですという意味です。きちんと個人@所属になっていますね。

尚、「個人%コメント等@所属」という表記もあるようです。これはパソコン通信(Nifty等の商用ネット)の方で使われていたものがインターネット側でも使い出されたものです。

Q. シグネチャを付けるにはどうすれば良いのでしょうか？

A. あらかじめファイルを作っておいてそれを挿入します。

前もってシグネチャをファイルに書いて作っておきます。そして必要に応じてその内容を取り込みます。そうすると毎回タイプする必要がなくなりますね。

muleを使ってメールを書いている人を対象に説明します。例えば sig というファイル名で1~4行程度のシグネチャを作っておきます。そしてメールを書き終わった時に C-x i と(コントロールキーを押しながら x キーを押して、コントロールキーを離して i キーを押す)すると、下に Insert file: ~/ の様なメッセージが出ると思われます。出たら、そのメッセージに続けて sig とタイプします。

Insert file: ~/sig

となりましたか？なったらリターンキーを押して下さい。無事にシグネチャがカーソルのあった位置へ書き込まれたと思います。なお、余り長いシグネチャはマナーに反しますので4行までにしましょう。

Q. ニュースに記事を投稿したのですが、うまく投稿できたかどうかを確認するには、どうすれば良いですか？

A. その記事を自分で読みましょう。

GNUS を使っているならニュースグループ選択画面で g キーを押す、または GNUS を一度終了して、もう一度起動すると新しく投稿された記事を読み込むのでそれで確認して下さい。ただし投稿後しばらくしないと記事が処理されません(最大15分)ので注意して下さい。

Q. いろんなニュースグループがありますが、それぞれどんなものなのですか？

A. 以下に一覧を挙げておきます。

京都産業大学に限ったニュースグループ

sandai 京都産業大学からのみ読み書き出来るニュースグループ。まずはここで慣れてから fj.* 等を読み書きすると良いでしょう。

京都産業大学以外の一般的なニュースグループ

fj 主に日本語によって論議をするためのもの。特定の組織や団体が管理・運営をしている訳ではなく、利用者全員の合意に基づいて運営されている。現在のところ営利目的の記事の投稿はできない。

tnn IJJ が主催する営利目的の記事を投稿できるニュースグループ

jp 日本の IP 接続組織用のニュースグループ

comp USENET のうちコンピュータに関する話題用

news USENET のうち NetNews に関する話題用

rec USENET のうち趣味に関する話題用

sci USENET のうち科学に関する話題用

soc USENET のうち社会問題に関する話題用

talk USENET のうち各種の論議用

misc 上記以外の USENET の話題用

alt USENET ニュースグループ群に代わる、各種の話題用。非常に緩いルールしかなく、かなり自由にニュースグループを作成して利用できる。

bionet 医学・生物学の話題用

biz ビジネスの話題用

gnu FSF(フリーソフトウェアファウンデーション)やその製品(GNUプロダクト)に関する話題用

k12 アメリカにおけるネットワークを使った教育の実験のためのニュースグループ

vmsnet DEC VAX/VMS OS に関する話題用

Q. クロスポストってなんですか？

A. 複数の NG に同じ記事を投稿する時に使用する機能です。

クロスポストとはニュースの記事をポストする時、ポスト先のニュースグループを複数指定（通常2つか3つ）して、同じ記事を各ニュースグループに投稿するものです。各ニュースグループに個別に投稿するのに比べて、記事の実体は一つで済みますので、ネットワーク資源の節約になります。

方法は、通常のポスト時に Newsgroup の指定行を

```
Newsgroups: sandai.test, sandai.junk
```

のようにカンマで区切って複数のニュースグループを書きます。この状態で投稿すると指定した各ニュースグループに投稿します。この状態の記事にフォローを入れた場合、Newsgroups の指定はそのまま受け継がれるので全てのニュースグループに同じフォローを入れる事になります。（このフォロー先を一意に決めたい時はフォローアップを使います。）

通常クロスポストを使うのは何箇所かのニュースグループで同時に議論したいときと、そのニュースグループで議論を続けるのにふさわしくない記事に対して、他のニュースグループにふる時につかいます。前者の場合は議題が幾つかのニュースグループにまたがっていて、どうしてもそれら全てのニュースグループの読者の意見を交えて議論したい、という場合です。通常のポストの際に Newsgroups に上記のように二つ以上のニュースグループを指定して下さい。

後者の場合は、フォローして、そのニュースグループと新しいニュースグループとのクロスポストにして、フォローアップを新しいニュースグループにします。例えば sandai.question のある記事に対して sandai.junk の方が適切だ、と思ったらフォローする際に次のようにします。まず F キーを押して引用符を決めるとこの様になりますね。

---ここから---

```
In-reply-to: ozaki@cc.kyoto-su.ac.jp's message of 26 Jan 1995 12:39:48 +0900
```

```
Newsgroups: sandai.question
```

```
Subject: Re: .emacs
```

```
Distribution: local
```

```
References: <TANIMA.95Jan26114834@csosf20.kyoto-su.ac.jp>
```

```
<OZAKI.95Jan26123945@cc2000.kyoto-su.ac.jp>
```

```
--text follows this line--
```

(引用した本文)

---ここまで---

通常、「-text follows this line-」より上の部分は書き換えないのですが、Newsgroups: 行の変更と、Followup-To: 行の追加を行ないます。

```
Newsgroups: sandai.question, sandai.junk
```

```
Followup-To: sandai.junk
```

この様に修正、追加します。次のようになりましたね。

---ここから---

```
In-reply-to: ozaki@cc.kyoto-su.ac.jp's message of 26 Jan 1995 12:39:48 +0900
```

```
Newsgroups: sandai.question, sandai.junk
```

```
Followup-To: sandai.junk
```

```
Subject: Re: .emacs
```

```
Distribution: local
```

```
References: <TANIMA.95Jan26114834@csosf20.kyoto-su.ac.jp>
```

```
<OZAKI.95Jan26123945@cc2000.kyoto-su.ac.jp>
```

```
--text follows this line--
```

(引用した本文)

---ここまで---

二行以外は変更ありません。これでいつものようにポストするとその記事は sandai.question と sandai.junk に投稿され、それぞれの記事を見た人がそれにフォローしようとするニュースグループに sandai.junk が選択されます。

そして、一言、「sandai.junk に振ります」と書き添えて下さい。こうすると sandai.question の読者は sandai.junk に議論の場が移るんだ、ということが分かりますし、sandai.junk の読者は sandai.question から移ってきた議論だ、ということが分かります。

Q. Followup-To: ってなんですか？

A. 記事に対してのフォロー先のニュースグループを決めるものです。

例えば sandai.test にある記事に

Followup-To: sandai.junk

となっていたらその記事のフォローは sandai.junk に投稿されます。

X に関すること

Q. リモートログインした機械でアプリケーションを立ち上げようとすると Can't open display が出て出ます。

A. 貴方が使っている機械の環境を変えなければいけません。

これはそのリモートログインしている機械が貴方の使っている機械に表示しようとしたけれども出来ませんでした、という事です。X ウィンドウはネットワークを介して繋がっている機械に画面を表示する為に作られたウィンドウシステムで、その為、何処に表示するのか、何処から自分の画面への表示を許すのか、を指定してやる必要があります。Can't open display とは表示しようとしている機械に表示出来ませんでした、ということです。他からの画面の書き込みを許すには xhost というコマンドを使います。

% xhost 機械名

とするとその機械からの表示を許可します。何処に表示するかは環境変数の DISPLAY で指定するのでログインした先のコマンドラインで

% setenv DISPLAY 自分の使っている端末の名前:0.0

と入力します。これで、目の前の端末が、ログイン先の機械の画面を表示出来るようになります。

例

```
csosf11(81)% xhost cc2000
~~~~~
```

```
cc2000 being added to access control list
```

```
csosf11(82)% rlogin cc2000
~~~~~
```

```
Last login: Thu Sep 22 11:00:56 from csosf41
```

```
tcsh: using dumb terminal settings.
```

```
Sun Microsystems Inc. SunOS 5.3 Generic September 1993
```

```
cc2000(81)% setenv DISPLAY csosf11:0.0
~~~~~
```

(下線部が入力部分)

setenv の 0:0 は 0 番目のディスプレイの 0 番目の画面という意味で、この意味のよく解らない人はおまじないとして打っていけば O.K.

それ以外のこと

Q. パーミッションって何ですか？

A. ファイルに対するアクセス権の事です。

パーミッションとはファイルに与えられるアクセス権で、オーナー、グループ、他人の三種類の人に対してリード権、ライト権、実行権をあるか無いかで決めます。詳しくは 162 ページ 6.3 を見て下さい。

Q. ディレクトリを他の人からも見られるようにしたいんですが

A. ディレクトリのアクセス権を変更します。

そのディレクトリに対するアクセス権を変えます。読めるようにするにはリード権と実行権を与えます。さらに書き込めるようにするにはライト権を与えます。あるグループに対して解放するなら所属グループを変更します。

この文章を読んで意味が分からない人はセキュリティ上、危険ですので少し勉強した方が良いと思われれます。例えば自分宛のメールを覗かれたり大切なデータをいじられたりする危険性があります。自信のない人は分かっている人に見て貰って下さい。

- Q.** キーボードを打っても文字が化ける、あるいは何も表示されずとも動かないのですが
A. 状況として3通り考えられます。

一つは何かバイナリデータなどの普通表示できないデータを画面に表示してしまった場合、端末設定が変わってしまい、以後の文字がすべて化けてしまう場合。そしてカナキーやCAPSキーなどの特殊なキーを押してしまった場合。もう一つはC-s(コントロールキーを押しながらsキーを押す)の場合です。順に対処法を。

- 一番目の場合
 ・ 端末状態をリセットする
 ・ 端末エミュレータを再起動してやる
- 二番目の場合
 カナキーやCAPSキーを解除する。(もう一度押す)
- 三番目の場合
 C-q (コントロールキーを押しながらqキーを押す) をする。

- Q.** 印刷したいのですが、どのプリンタを指定したら良いのでしょうか？
A. ccinfo コマンドで参照できます。

ccinfo の「cc 環境の設備について」の「プリンタの配置について」を見て下さい。
 NeXT についてはモノクロの印字は綺麗で速いレーザープリンタの TAKE や UME で、どうしてもカラーでないといけないものは遅いけどカラー印刷のできる MATSU で行って下さい。プリンタの場所は MATSU と TAKE が C 3、UME が C 4 情報処理教室です。

- Q.** 家のパソコンとデータをやりとりしたいのですが

- A.** フロッピーディスクのフォーマットには
- | | |
|--------------|--------------------------------------------------------------------------------------------|
| PC-9801,FM-R | 2DD(640KB),2DD(720KB),2HC(1.20MB),2HD(1.25MB),
2HD(1.44MB 但し最近の機種のみ) |
| DOS/V | 2DD(720KB),2HC(1.20MB),2HD(1.44MB),
2HD(1.25MB 但し一部の機種のみ) |
| Macintosh | 2DD(800KB Mac),2HD(1.44KB Mac)
※98などの同じ容量のフォーマットとは種類が
違うので互換性は無い。 |
| NeXT | 2DD(720KB),2DD(720KB NeXT),2DD(800KB Mac),
2HD(1.44MB),2HD(1.44MB Mac),2HD(1.44MB NeXT) |

などが存在します。目的とする機種間で共通のフォーマットを使って下さい。

対応表

情報処理教室 機種	9801	DosV	C1 FM-V	C3 NeXT	21 DEC	31 Mac	51 FM-R	52 compac
2DD(640KB)	○	×	×	×	×	×	○	×
2DD(720KB)	○	○	○	○	○	×	○	○
2DD(800KB Mac)	×	×	×	○	×	○	×	×
2HD(1.25MB)	○	△	○	×	×	×	○	×
2HD(1.44MB)	△	○	○	○	○	×	○	○
2HD(1.44MB Mac)	×	×	×	○	×	○	×	×

○可能 △機種により可能 ×駄目

使用できるフロッピーディスクを用意して、次の内、いずれかの手順で
 unix file system - floppy disk 間のやりとりをして下さい。

- 21 情報処理教室 (csosf) フォーマットされたディスクを挿入して fdio コマンドを使用する。詳細は fdio コマンドを実行して?でヘルプを出して読んで下さい。
- C 3 情報処理教室 (NeXT) フロッピーディスクを挿入するとワークスペースマネージャの上側 (シェルフ) にフロッピーディスクが現れ、

一般のフォルダと同じように扱えますのでそこと目的の
ファイルをやりとりして下さい。

C 1 情報処理教室 (Dos/V) フォーマットされたディスクを挿入して Windows のメニューより RapidFiler を起動して下さい。使い方はヘルプを読んで下さい。

Q. 21 情報処理教室の DEC3300 でセッション休止から復帰できません

A. Lock キーを調べて下さい。

login panel から login した後にルートメニューのアクセサリからセッション休止を選ぶと画面を真っ暗にして休止状態になりますが、このとき、Lock キー (CapsLock キー) を押して Lock 状態 (キーボード右上の Lock ランプが点灯) にしていても、休止状態になったと同時にキーボードの Lock ランプも消えてしまいます。(しかし Lock は効いたまま)

休止状態から復帰するにはパスワードの入力を必要としますが、このとき、Lock ランプは消えていますが Lock 状態のままです。小文字モードと思ってパスワードを入力してもアルファベットは大文字のままです。復帰に失敗します。

Lock キーはちゃんと効いていますので、Lock キーを一回押してもう一度パスワードを入力してみましょう。復帰に失敗した原因がこれであったならば、復帰に成功します。

A.4.4 Mac 編

Q. マックにリセットスイッチはついていないのですか？

A. キーボードによるリセットがあります。

ctrl+アップルマーク (花文字の奴)+キーボードの上にある三角のもの、です。(一部機種では使えません)

A.4.5 Program 編

Q. math.h を使ったらコンパイルできない。

A. UNIX 上で C 言語で、sin 関数などのいわゆる math.h を include するようなプログラムを cc しようとした時 NeXT ではそのままコンパイル出来るのですが、cc2000 や csosf や SUN では数学ライブラリを組み込む-lm オプションが必要です。

例: cc test.c -lm

#それでも出来ない時... プログラムミスかな?(^^;)

A.4.6 その他

Q. フロッピーディスクを買いたいのですが、

A. 基本的に電気屋さんやパソコンショップで売っています。種類は 3.5' 2HD (サンテンゴインチ ニイエイチディー) と呼ばれるものです。

学内では丸善などで取り扱っています。但し少し市場より値が高いので大量に買うのなら他のパソコンショップなどで買う方が良いでしょう。また、異なる機種間でデータのやり取りをされるのであれば 3.5' 2DD (ニイディーディー) と呼ばれる種類の方がいい場合もあります。

付録 B

情報処理教室利用要項

情報処理教室の利用について (学生便覧より抜粋)

計算機を利用する授業や自習のため、学内には9か所の情報処理教室を設置しています。この情報処理教室に設置している機器は、計算機システムの端末として利用できる他、パーソナル・コンピュータ（パソコン）としても利用できます。利用できるソフトウェアは各情報処理教室ごとに違いますので、その教室を管理する所管の事務室または計算機センター事務室で確認して下さい。

(1) 情報処理教室の利用機器等

建物	教室名	機器		所管
計算機科学研究所棟 2階	C1 情報処理教室	FMV-466D	32台	計算機科学研究所
計算機科学研究所棟 3階	C2 情報処理教室	PanaStation	22台	理学部
計算機科学研究所棟 3階	C3 情報処理教室	NeXT station	15台	計算機科学研究所
計算機科学研究所棟 4階	C4 情報処理教室	NeXT station	6台	理学部
1号館 2階	11 情報処理教室	PowerMacintosh6100	90台	一般教育研究センター
2号館 4階	21 情報処理教室	DEC-3300	40台	理学部
3号館 2階	31 情報処理教室	PowerMacintosh6100	44台	外国語学部
5号館 1階	51 情報処理教室	FMR-280H	35台	経済学部
5号館 2階	52 情報処理教室	Compaq Contura	45台	経営学部

(2) 利用資格

本学の学生であれば自由に利用することが出来ますが、利用機器によっては、計算機利用資格 (ID) が必要となります。計算機利用資格の申請は、計算機センターで行なって下さい。

(3) 利用時間

平日 午前8時45分～午後8時まで

土曜日 午前8時45分～午後5時まで

情報処理教室の利用は授業を優先しますが、授業のない時間帯は研究や自習として自由に利用できます。授業以外の目的で利用される場合は、「情報処理教室利用時間割表」を掲示しますので、授業有無を確認して下さい。なお、日曜日・祝日、夏期一斉休業期間および年末年始期間には使用できません。その他保守などによるシステム停止日、清掃などによる閉室日はその都度掲示で連絡します。

(4) 入室方法

入室は、学生証を情報処理教室出入口のカード読み取り装置に通し開錠のうえ、入室して下さい。なお、C2、C3、C4 情報処理教室の利用時間には開錠してありますが、平日の午後 4 時 30 分以降及び土曜日の正午以降の利用は、事前に計算機科学研究所事務室に届け出のうえ鍵を借用して下さい。また、52 情報処理教室への入室は常時鍵での開閉となりますので、経営学部事務室で鍵を借用して下さい。

(5) 利用心得

利用者は、次に掲げる利用心得を遵守して下さい。

1. 利用後は、機器の電源を切ること。
2. 最終利用者は、室内を消灯すること。
3. C2、C3、C4 および 52 情報処理教室での最終利用者は施錠し、鍵を借用した事務室の扉の郵便受けに返却すること。
4. 教室内のマニュアル及び備品の持ち出しの禁止
5. 教室内での飲食及び喫煙の禁止
6. その他所管の学部など事務室から特に指示のあった場合は、これに従うこと。

付録 C

著作権法 (抜粋)

※出典 ニフティサーブ 法令データベース

※タイプミスなどの誤りが含まれている可能性があることを予めご了承ください。

(昭和四十五年五月六日法律第四十八号)

昭和四十六年一月一日

平成五年一月一日法律第八九号

著作権法(明治三十二年法律第三十九号)の全部を改正する。

第一章 総則

第一節 通則

(目的)

第一条 この法律は、著作物並びに実演、レコード、放送及び有線放送に関し著作者の権利及びこれに隣接する権利を定め、これらの文化的所産の公正な利用に留意しつつ、著作者等の権利の保護を図り、もつて文化の発展に寄与することを目的とする。

(定義)

第二条 この法律において、次の各号に掲げる用語の意義は、当該各号に定めるところによる。

- 一 著作物 思想又は感情を創作的に表現したものであつて、文芸、学術、美術又は音楽の範囲に属するものをいう。
- 二 著作者 著作物を創作する者をいう。
- 三 実演 著作物を、演劇的に演じ、舞い、演奏し、歌い、口演し、朗詠し、又はその他の方法により演ずること(これらに類する行為で、著作物を演じないが芸術的な性質を有するものを含む。)をいう。
- 四 実演家 俳優、舞踊家、演奏家、歌手その他実演を行なう者及び実演を指揮し、又は演出する者をいう。
- 五 レコード 蓄音機用音盤、録音テープその他の物に音を固定したもの(音をもつばら映像とともに再生することを目的とするものを除く。)をいう。
- 六 レコード製作者 レコードに固定されている音を最初に固定した者をいう。
- 七 商業用レコード 市販の目的をもつて製作されるレコードの複製物をいう。
- 八 放送 公衆によつて直接受信されることを目的として無線通信の送信を行なうことをいう。
- 九 放送事業者 放送を業として行なう者をいう。
- 九の二 有線放送 有線送信のうち、公衆によつて同一の内容の送信が同時に受信されることを目的として行うものをいう。
- 九の三 有線放送事業者 有線放送を業として行う者をいう。
- 十 映画製作者 映画の著作物の製作に発意と責任を有する者をいう。
- 十の二 プログラム 電子計算機を機能させて一の結果を得ることができるようにこれに対する指令を組み合わせたものとして表現したものをいう。
- 十の三 データベース 論文、数値、図形その他の情報の集合体であつて、それらの情報を電子計算機を用いて検索することができるように体系的に構成したものをいう。
- 十一 二次的著作物 著作物を翻訳し、編曲し、若しくは変形し、又は脚色し、映画化し、その他翻案することにより創作した著作物をいう。
- 十二 共同著作物 二人以上の者が共同して創作した著作物であつて、その各人の寄与を分離して個別的に利用することができないものをいう。
- 十三 録音 音を物に固定し、又はその固定物を増製することをいう。
- 十四 録画 映像を連続して物に固定し、又はその固定物を増製することをいう。
- 十五 複製 印刷、写真、複写、録音、録画その他の方法により有形的に再製することをいい、次に掲げるものを

については、それぞれ次に掲げる行為を含むものとする。イ 脚本その他これに類する演劇用の著作物 当該著作物の上演、放送又は有線放送を録音し、又は録画すること。ロ 建築の著作物 建築に関する図面に従つて建築物を完成すること。

十六 上演 演奏（歌唱を含む。以下同じ。）以外の方法により著作物を演ずることをいう。

十七 有線送信 公衆によつて直接受信されることを目的として有線電気通信の送信（有線電気通信設備で、その一部の設置の場所が他の部分の設置の場所と同一の構内（その構内が二以上の者の占有に属している場合には、同一の者の占有に属する区域内）にあるものによる送信を除く。）を行うことをいう。

十八 口述 朗読その他の方法により著作物を口頭で伝達すること（実演に該当するものを除く。）をいう。

十九 上映 著作物を映写幕その他の物に映写することをいい、これに伴つて映画の著作物において固定されている音を再生することを含むものとする。

二十 頒布 有償であるか又は無償であるかを問わず、複製物を公衆に譲渡し、又は貸与することをいい、映画の著作物又は映画の著作物において複製されている著作物にあつては、これらの著作物を公衆に提示することを目的として当該映画の著作物の複製物を譲渡し、又は貸与することを含むものとする。

二十一 国内 この法律の施行地をいう。

※ 略 ※

第二章 著作者の権利

第一節 著作物

（著作物の例示）

第十条 この法律にいう著作物を例示すると、おおむね次のとおりである。

- 一 小説、脚本、論文、講演その他の言語の著作物
- 二 音楽の著作物
- 三 舞踊又は無言劇の著作物
- 四 絵画、版画、彫刻その他の美術の著作物
- 五 建築の著作物
- 六 地図又は学術的な性質を有する図面、図表、模型その他の図形の著作物
- 七 映画の著作物
- 八 写真の著作物
- 九 プログラムの著作物

2 事実の伝達にすぎない雑報及び時事の報道は、前項第一号に掲げる著作物に該当しない。

3 第一項第九号に掲げる著作物に対するこの法律による保護は、その著作物を作成するために用いるプログラム言語、規約及び解法に及ばない。この場合において、これらの用語の意義は、次の各号に定めるところによる。

- 一 プログラム言語 プログラムを表現する手段としての文字その他の記号及びその体系をいう。
- 二 規約 特定のプログラムにおける前号のプログラム言語の用法についての特別の約束をいう。
- 三 解法 プログラムにおける電子計算機に対する指令の組合せの方法をいう。

（二次的著作物）

第十一条 二次的著作物に対するこの法律による保護は、その原著物の著作者の権利に影響を及ぼさない。

（編集著作物）

第十二条 編集物（データベースに該当するものを除く。以下同じ。）でその素材の選択又は配列によつて創作性を有するものは、著作物として保護する。

2 前項の規定は、同項の編集物の部分を構成する著作物の著作者の権利に影響を及ぼさない。

（データベースの著作物）

第十二条の二 データベースでその情報の選択又は体系的な構成によつて創作性を有するものは、著作物として保護する。

2 前項の規定は、同項のデータベースの部分を構成する著作物の著作者の権利に影響を及ぼさない。

（権利の目的とならない著作物）

第十三条 次の各号のいずれかに該当する著作物は、この章の規定による権利の目的となることができない。

- 一 憲法その他の法令
- 二 国又は地方公共団体の機関が発する告示、訓令、通達その他これらに類するもの
- 三 裁判所の判決、決定、命令及び審判並びに行政庁の裁決及び決定で裁判に準ずる手続により行なわれるもの
- 四 前三号に掲げるものの翻訳物及び編集物で、国又は地方公共団体の機関が作成するもの

第二節 著作者

（著作者の推定）

第十四条 著作物の原作品に、又は著作物の公衆への提供若しくは提示の際に、その氏名若しくは名称（以下「実名」という。）又はその雅号、筆名、略称その他実名に代えて用いられるもの（以下「変名」という。）として周知のものが著作者名として通常の方法により表示されている者は、その著作物の著作者と推定する。

（職務上作成する著作物の著作者）

第十五条 法人その他使用者（以下この条において「法人等」という。）の発意に基づきその法人等の業務に従事する者が職務上作成する著作物（プログラムの著作物を除く。）で、その法人等が自己の著作の名義の下に公表するものの著作者は、その作成の時にける契約、勤務規則その他に別段の定めがない限り、その法人等とする。

2 法人等の発意に基づきその法人等の業務に従事する者が職務上作成するプログラムの著作物の著作者は、その作成の時にける契約、勤務規則その他に別段の定めがない限り、その法人等とする。

（映画の著作物の著作者）

第十六条 映画の著作物の著作者は、その映画の著作物において翻案され、又は複製された小説、脚本、音楽その他の著作物の著作者を除き、制作、監督、演出、撮影、美術等を担当してその映画の著作物の全体的形成に創作的に寄与した者とする。ただし、前条の規定の適用がある場合は、この限りでない。

第三節 権利の内容

第一款 総則

（著作者の権利）

第十七条 著作者は、次条第一項、第十九条第一項及び第二十条第一項に規定する権利（以下「著作者人格権」という。）並びに第二十一条から第二十八条までに規定する権利（以下「著作権」という。）を享有する。

2 著作者人格権及び著作権の享有には、いかなる方式の履行をも要しない。

第二款 著作者人格権

（公表権）

第十八条 著作者は、その著作物でまだ公表されていないもの（その同意を得ないで公表された著作物を含む。次項において同じ。）を公衆に提供し、又は提示する権利を有する。当該著作物を原著物とする二次的著作物についても、同様とする。

2 著作者は、次の各号に掲げる場合には、当該各号に掲げる行為について同意したものと推定する。

一 その著作物でまだ公表されていないものの著作権を譲渡した場合 当該著作物をその著作権の行使により公衆に提供し、又は提示すること。

二 その美術の著作物又は写真の著作物でまだ公表されていないものの原作品を譲渡した場合 これらの著作物をその原作品による展示の方法で公衆に提示すること。

三 第二十九条の規定によりその映画の著作物の著作権が映画製作者に帰属した場合 当該著作物をその著作権の行使により公衆に提供し、又は提示すること。

（氏名表示権）

第十九条 著作者は、その著作物の原作品に、又はその著作物の公衆への提供若しくは提示に際し、その実名若しくは変名を著作者名として表示し、又は著作者名を表示しないこととする権利を有する。その著作物を原著物とする二次的著作物の公衆への提供又は提示に際しての原著物の著作者名の表示についても、同様とする。

2 著作物を利用する者は、その著作者の別段の意思表示がない限り、その著作物につきすでに著作者が表示しているところに従って著作者名を表示することができる。

3 著作者名の表示は、著作物の利用の目的及び態様に照らし著作者が創作者であることを主張する利益を害するおそれがないと認められるときは、公正な慣行に反しない限り、省略することができる。

（同一性保持権）

第二十条 著作者は、その著作物及びその題号の同一性を保持する権利を有し、その意に反してこれらの変更、切除その他の改変を受けないものとする。

2 前項の規定は、次の各号のいずれかに該当する改変については、適用しない。

一 第三十三条第一項（同条第四項において準用する場合を含む。）又は第三十四条第一項の規定により著作物を利用する場合における用字又は用語の変更その他の改変で、学校教育の目的上やむを得ないと認められるもの

二 建築物の増築、改築、修繕又は模様替えによる改変

三 特定の電子計算機においては利用し得ないプログラムの著作物を当該電子計算機において利用し得るようにするため、又はプログラムの著作物を電子計算機においてより効果的に利用し得るようにするために必要な改変

四 前三号に掲げるもののほか、著作物の性質並びにその利用の目的及び態様に照らしやむを得ないと認められる改変

第三款 著作権に含まれる権利の種類

（複製権）

第二十一条 著作者は、その著作物を複製する権利を専有する。

(上演権及び演奏権)

第二十二條 著作者は、その著作物を、公衆に直接見せ又は聞かせることを目的として(以下「公に」という。)上演し、又は演奏する権利を専有する。

(放送権、有線送信権等)

第二十三條 著作者は、その著作物を放送し、又は有線送信する権利を専有する。

2 著作者は、放送され、又は有線送信されるその著作物を受信装置を用いて公に伝達する権利を専有する。

(口述権)

第二十四條 著作者は、その言語の著作物を公に口述する権利を専有する。

(展示権)

第二十五條 著作者は、その美術の著作物又はまだ発行されていない写真の著作物をこれらの原作品により公に展示する権利を専有する。

(上映権及び頒布権)

第二十六條 著作者は、その映画の著作物を公に上映し、又はその複製物により頒布する権利を専有する。

2 著作者は、映画の著作物において複製されているその著作物を公に上映し、又は当該映画の著作物の複製物により頒布する権利を専有する。

(貸与権)

第二十六條の二 著作者は、その著作物(映画の著作物を除く。)をその複製物(映画の著作物において複製されている著作物にあつては、当該映画の著作物の複製物を除く。)の貸与により公衆に提供する権利を専有する。

(翻訳権、翻案権等)

第二十七條 著作者は、その著作物を翻訳し、編曲し、若しくは変形し、又は脚色し、映画化し、その他翻案する権利を専有する。

(二次的著作物の利用に関する原著作者の権利)

第二十八條 二次的著作物の原著作物の著作者は、当該二次的著作物の利用に関し、この款に規定する権利で当該二次的著作物の著作者が有するものと同一の種類の権利を専有する。

※ 略 ※

(私的使用のための複製)

第三十條 著作権の目的となつてゐる著作物(以下この款において単に「著作物」という。)は、個人的に又は家庭内その他これに準ずる限られた範囲内において使用すること(以下「私的使用」という。)を目的とする場合には、公衆の使用に供することを目的として設置されている自動複製機器(複製の機能を有し、これに関する装置の全部又は主要な部分が自動化されている機器をいう。)を用いて複製するときを除き、その使用する者が複製することができる。

2 私的使用を目的として、デジタル方式の録音又は録画の機能を有する機器(放送の業務のための特別の性能その他の私的使用に通常供されない特別の性能を有するもの及び録音機能付きの電話機その他の本来の機能に附属する機能として録音又は録画の機能を有するものを除く。)であつて政令で定めるものにより、当該機器によるデジタル方式の録音又は録画の用に供される記録媒体であつて政令で定めるものに録音又は録画を行う者は、相当な額の補償金を著作権者に支払わなければならない。

(図書館等における複製)

第三十一條 図書、記録その他の資料を公衆の利用に供することを目的とする図書館その他の施設で政令で定めるもの(以下この条において「図書館等」という。)においては、次に掲げる場合には、その営利を目的としない事業として、図書館等の図書、記録その他の資料(以下この条において「図書館資料」という。)を用いて著作物を複製することができる。

一 図書館等の利用者の求めに応じ、その調査研究の用に供するために、公表された著作物の一部分(発行後相当期間を経過した定期刊行物に掲載された個々の著作物にあつては、その全部)の複製物を一人につき一部提供する場合

二 図書館資料の保存のため必要がある場合

三 他の図書館等の求めに応じ、絶版その他これに準ずる理由により一般に入手することが困難な図書館資料の複製物を提供する場合

(引用)

第三十二條 公表された著作物は、引用して利用することができる。この場合において、その引用は、公正な慣行に

合致するものであり、かつ、報道、批評、研究その他の引用の目的上正当な範囲内で行なわれるものでなければならない。

2 国又は地方公共団体の機関が一般に周知させることを目的として作成し、その著作の名義の下に公表する広報資料、調査統計資料、報告書その他これらに類する著作物は、説明の材料として新聞紙、雑誌その他の刊行物に転載することができる。ただし、これを禁止する旨の表示がある場合は、この限りでない。

※ 略 ※

第四十七条 美術の著作物又は写真の著作物の原作品により、第二十五条に規定する権利を害することなく、これらの著作物を公に展示する者は、観覧者のためにこれらの著作物の解説又は紹介をすることを目的とする小冊子にこれらの著作物を掲載することができる。

(プログラムの著作物の複製物の所有者による複製等)

第四十七条の二 プログラムの著作物の複製物の所有者は、自ら当該著作物を電子計算機において利用するために必要と認められる限度において、当該著作物の複製又は翻案（これにより創作した二次的著作物の複製を含む。）をすることができる。ただし、当該利用に係る複製物の使用につき、第百十三条第二項の規定が適用される場合は、この限りでない。

2 前項の複製物の所有者が当該複製物（同項の規定により作成された複製物を含む。）のいずれかについて滅失以外の事由により所有権を有しなくなった後には、その者は、当該著作権者の別段の意思表示がない限り、その他の複製物を保存してはならない。

(出所の明示)

第四十八条 次の各号に掲げる場合には、当該各号に規定する著作物の出所を、その複製又は利用の態様に応じ合理的と認められる方法及び程度により、明示しなければならない。

※ 略 ※

第四節 保護期間

(保護期間の原則)

第五十一条 著作権の存続期間は、著作物の創作の時に始まる。

2 著作権は、この節に別段の定めがある場合を除き、著作者の死後（共同著作物にあつては、最終に死亡した著作者の死後。次条第一項において同じ。）五十年を経過するまでの間、存続する。

※ 略 ※

第七節 権利の行使

(著作物の利用の許諾)

第六十三条 著作権者は、他人に対し、その著作物の利用を許諾することができる。

2 前項の許諾を得た者は、その許諾に係る利用方法及び条件の範囲内において、その許諾に係る著作物を利用することができる。

3 第一項の許諾に係る著作物を利用する権利は、著作権者の承諾を得ない限り、譲渡することができない。

4 著作物の放送又は有線放送についての第一項の許諾は、契約に別段の定めがない限り、当該著作物の録音又は録画の許諾を含まないものとする。

※ 略 ※

(侵害とみなす行為)

第百十三条 次の掲げる行為は、当該著作人格権、著作権、出版権又は著作隣接権を侵害する行為とみなす。

一 国内において頒布する目的をもって、輸入の時に国内で作成したとしたならば著作人格権、著作権、出版権又は著作隣接権の侵害となるべき行為によつて作成された物を輸入する行為

二 著作人格権、著作権、出版権又は著作隣接権を侵害する行為によつて作成された物（前号の輸入に係る物を含む。）を情を知つて頒布し、又は頒布の目的をもって所持する行為

2 プログラムの著作物の著作権を侵害する行為によつて作成された複製物（当該複製物の所有者によつて第四十七条の二第一項の規定により作成された複製物並びに前項第一号の輸入に係るプログラムの著作物の複製物及び当該複製物の所有者によつて同条第一項の規定により作成された複製物を含む。）を業務上電子計算機において使用する行為は、これらの複製物を使用する権原を取得した時に情を知つていた場合に限り、当該著作権を侵害する行為とみなす。

3 著作者の名誉又は声望を害する方法によりその著作物を利用する行為は、その著作人格権を侵害する行為とみなす。

※ 以下略 ※

付録 D

参考文献

ここにあげる文献にある記述が全て cc 環境で適用出来るとは限らないことに注意してください。

—— Unix 全般、シェル、コマンドなどについて ——

たのしい UNIX -UNIX への招待- 坂本 文著：アスキー出版局刊

月刊雑誌 UNIX Magazine の連載を集成した UNIX 初心者向けの入門書。

続 たのしい UNIX -シェルへの招待- 坂本 文著：アスキー出版局刊

上の一冊の続編。今度はシェルについて解説してくれる。

実用 UNIX ハンドブック 舟本 奨著：ナツメ社

UNIX コマンドの簡単なリファレンス。

UNIX C SHELL フィールドガイド G・アンダーソン、P・アンダーソン著：落水 浩一郎、大木 敦

雄訳：パーソナルメディア刊

csh のほぼ完全なガイド。

UNIX step++ シェルプログラミングのコツ 西沼 行博著：マグロウヒル刊

残念ながら csh ではなく、sh についての説明が主体。記述も少々古いがシェルを使いこなしたい人には便利。

The Unix Super Text 山口 和紀監修、于 旭・中村 敦司・新城 靖・西山 博泰・古瀬 一隆・石川 佳治・佐々木 重雄・林 謙一・萩原 一隆・金谷 英信・鈴木 孝幸・黒石 和宏 著：技術評論社刊

BSD と呼ばれるタイプの UNIX 主体に書いてあるため、cc 環境とは相違がある場合が散見されるが、一般的な概念からハウツーまで、広い分野に関して教えてくれる。上下巻の二冊組。上巻は一般的な UNIX の使い方、電子メール、ニュース、X ウィンドウなどについて。下巻は L^AT_EX、プログラミング、システム管理などについて。高価なのが難点か。

誰にでも使える UNIX 講座 安岡孝一著：ソフトバンク社刊、1992 年

yasuoka さんが root さんに色々教えてもらう対話形式の本。login, logout から始まって UNIX の基本的なツール sh, csh, sed, make, awk の使い方がこれ一冊でわかる。UNIX を使いこなしてみたい人の入門書。

MH & xmh Jerry Peek 著、倉骨彰 訳、砂原秀樹・鈴木麗 監訳：アスキー出版局刊、1994 年 11 月、5,800 円

MH に関するマニュアル。

—— Mule について ——

入門 Mule 大木 敦雄著：アスキー出版刊

まさに Mule の入門書。

—— Emacs について ——

Emacs は Mule の前のバージョンです。基本的な操作などに付いては殆どこの Emacs の文献が利用できます。参考にしてください。

GNU Emacs 入門 (株) 京都ソフトウェアリサーチメディアデザイン編：オーム社刊、1994

Emacs を非常に分かり安く解説しています。自習にも向いていると思います。付録としてついている切り取り式のコマンド一覧も便利。

GNU Emacs Debra Cameron and Bill Rosenblatt 著：ハイパーウェア監訳：ソフトバンク株式会社刊

Emacs のほぼ完全なガイド。NutsShell (ナッツ (どんぐり) のカラ?) シリーズと呼ばれる非常に詳細な Unix 関係のドキュメントのシリーズの一冊。

GNU Emacs 完全ガイド M.A. シュノーバー、J.S. ボウイ、W.R. アーノルド 著・小畑喜一、磯谷正孝、山野修、林秀幸訳：アジソン・ウェスレイ・トッパン刊 (情報科学シリーズ 30,35) (上巻：1992 年 11 月 3,400 円、下巻：1993 年 3 月 3,900 円)

本書が取り上げている GNU Emacs は Version が、18.57 です。紐解く時に「完全ガイド」の安心感があります。

GNU Emacs マニュアル Richard Stallman 著：竹内郁雄・天海良治訳：共立出版刊、1988 年 2 月、2,930 円

GNU の御大自らの解説です。《古典》の部類なのかもしれませんが。

入門 NEmacs 大木 敦雄著：アスキー出版刊

Emacs だけでなく、EGG, MHE, GNUS などについても説明してくれている。

—— TeX について ——

楽々 LaTeX 野寺 隆志著：共立出版刊、第二版、1994

LaTeX を用いた TeX の入門書。初心者には最適。

LaTeX 入門—美文書作成のポイント— 奥村 晴彦監修：技術評論社刊、1994

TeX についていろいろ丁寧に教えてくれる。

やさしい LaTeX のはじめかた すずき ひろのぶ著：オーム社刊、1991

表題通り LaTeX のやさしい入門書です。その割に、便利なことがチョコチョコと書いてある。

日本語 LaTeX 定番スタイル集 No.1, No.2 鷺谷 好輝著：インプレス刊

京都産業大学でキャンパスライセンスを取得している LaTeX のスタイルファイルの使い方解説書。きれいなスタイルファイルの見本としてもよい。

LaTeX エラーマニュアル 今井豊著：カットシステム (Tel.0423-94-2218) 刊、1994 年 6 月、2,300 円

マイナーなところからの出版のためか、話題になりませんでした。「エラーをなおし、エラーから学ぶ本格的解説書」ではあります。AUCTeX があれば、半減しているのでしょうか、TeX を使う楽

しきの半分は、「エラーとの格闘」ゲームではないでしょうか。その意味では、ゲーム本に分類すべきなのかもしれませんが。

てくてく TeX 阿瀬はる美著：アスキー出版局刊、(上巻1994年11月、2,000円、下巻1994年12月、2,000円)

語り口の楽しさが、坂本文著「たのしい UNIX」に似ているのは、「UNIX MAGAZINE」1989/04-90/07 連載だったせいでしょうね。

逆引き LaTeX D.J. バーガー著、引地 信之・引地 美恵子訳：マグロウヒル刊

「こんなことが出来るのではないか？」と思った機能から、その方法を調べる。

LaTeX トータルガイド 伊藤 和人著：秀和システムトレーディング刊

LaTeX 技能の華麗な一覧表。

文書処理システム LaTeX Leslie Lamport 著、Edgar Cooke・倉沢 良一監訳、大野 俊治・小暮 博道・藤浦 はる美著：アスキー出版局刊

他の LaTeX 解説書の説明では納得出来ない時の駆け込み寺。入門書なのに理論的。

改訂新版 TeX ブック Donald E. Knuth 著、斉藤信男監修、鷺谷好輝訳：アスキー出版局刊

教祖様直筆 TeXnician の Bible。頭から噛じるに困難でも、昼寝の枕に最適。

—— インターネットについて ——

ハッピー・ネットワークング 山本和彦著：アスキー出版局、1994年7月刊、1,500円

NEmacs, mail, news, FTP だけに焦点を絞った若者による新入生向けであることに好感が持てました。cc 環境では殆ど等価のドキュメントを `ccinfo` コマンドで印刷させることも出来ますが、綺麗に製本された物が欲しければやはりこれを買うことになるでしょう。

インターネット 情報生活入門 グループまたたび著：技術評論社、1994年10月刊、1,700円。 インターネットの全体的な紹介をビジュアルにしている点がよいと思います。

インターネット参加の手引き~1994年度版 WIDE Project 編・村井純、吉村伸 監修：共立出版、bit 別冊、1994年7月刊、4,800円

情報はタププリ詰まっていますけど、高価。1995年度版が出るそうです。

インターネット ユーザーズガイド Ed Krol 著・村井純 監訳：インターナショナル・トムソン・パブリッシング・ジャパン刊 1995年5月刊、4,400円

ブームに向けてタイミング良く出版されてスタンダードなものになったようですけど悪訳を批判されたりもしていました。今年の1月に「改訂版」が出ていましたが、訳の改善がはたしてどの程度行なわれたのか、確認していません。(これでは紹介文にならない!)

インターネット・ナビゲータ Paul Gilster 著・菱山博陸訳：丸善、1995年2月刊、6,592円

原書第2版が底本。「本書は常にモデムでインターネット利用する人を念頭においています。」(p.vii)ついで台詞に、コロっと心底、騙されました。インターネット論としても優れているものだと思います。

配布、改変

このドキュメントは非営利目的に利用する限り、自由に複写、改変、再配布して構いません。逆に営利目的に利用する事は許しません。この基本線を守る限り、あなたはこのドキュメントに対して何をしても自由です。ここではこれ以上述べません。あなたが著作者達の期待を裏切らないよう、信じています。

連絡など

訂正や誤りに対する連絡は京都産業大学の計算機センターまでお願いします。ここはまずい、こうした方が良いという相談は大歓迎です。我々の環境は時代に合わせてどんどん変化します。このドキュメントもまずいところはどんどん直して変えていって欲しいと思います。そうしてこのドキュメントが実際に役に立つものとして成熟して行けば良いなと考えているのです。

UNIX ガイド編集責任者

計算機センター事務室 安田 豊

編集委員

谷川 正幸 尾崎 孝治 山崎 英知 松浦 正和

小坂田 浩孝 辻本 将彦 笠 克明

計算機センター事務室 (坪内 伸夫 重田 裕之

土肥 順一 岡田 光博 開原 潮)

発行

京都産業大学 計算機センター事務室

〒 603 京都市北区上賀茂本山

電話 075-705-1483

謝辞

計算機センターは初めて使うところからメールやニュースが使えるようになるまでの一連のチュートリアルを書いています。残りの部分については cc 環境利用者の有志から原稿を頂きました。

まず Mule の章とそのリファレンス、レポートシステムと FAQ の部分を理学部計算機科学科の尾崎君が書いてくれています。L^AT_EX の章を理学部物理学科の山崎君¹が書いてくれました。L^AT_EX 前半の一部分は「UNIX ガイド 追補版」の為に同じく物理学科の辻本君が書き起こしてくれたものが元になっています。AUC_TE_X と TGIF の章については理学研究科物理学専攻の松浦君が、Mathematica の章については理学研究科数学専攻の笠君が書いてくれました。MODEM から telnet の章は経済学部の小坂田君が書いてくれています。GNUPLOT の章は理学部の谷川先生が書いて下さいました。NeXT の章は計算機センターの開原君が書いてくれました。彼はまた著作権法の部分をまとめてくれています。NQS、*xv*、*xpaint* の章と UNIX コマンドリファレンスは計算機センターの安田が cc 環境の一ユーザーとして書いています。今回も突然の依頼となりましたが工学部の黒住先生には再び巻頭を引き受けていただきました。ガイド執筆では直接現れませんが、cc 環境の Mule の設定には理学部の立木先生や他の方々から多くの御協力を得て何とかなりました。レポートシステムに関しては理学部の尾崎君の作品です。DEC-3300 の特製 xdm は工学研究科情報通新工学専攻の為永君の作品です。その他にも多くの方の協力を戴きました。

それら全ての人に感謝致します。

経緯

cc 環境は 1993 年の夏に計算機センターが導入した UNIX マシン群が核となったコンピュータ利用環境である。それまで大学の中に共用 UNIX 環境が存在しなかったと言う事もあって、当時大学の中での UNIX の利用者は少なかったが、1994 年の春から一般学生にも授業と関わりなくアカウントを出すという事態になった。そこで授業などとは関係なく UNIX 環境を利用することになる学生のために、私は簡単なガイドを書かなくてはならないと思った。そうして「UNIX ガイド」初版が書かれたのだ。

1994 年の夏には多くの人々の執筆協力を得て「UNIX ガイド 追補版」を出版する事が出来た。これは学内で UNIX 利用者がそれだけ育った事を象徴する出来事だった。そして今この二冊を合本して更に内容を加えた第二版を出版するに至ったのだ。一年半前に苦勞して cc 環境の設計を行っていたのが嘘のように昔の事に思える。最初の半年の利用者には何のドキュメントも用意してやれず苦勞を掛けたと今更ながらに思う。

1995 年の春から新入生を主体にもっと積極的に cc 環境へ誘導する事になった。これから cc 環境がどのように育っていくのか予想が付かない部分もあるが、このドキュメントがその助けになれば良いなとただそれだけを思う。

私が初めて UNIX を触ったのは 1987 年頃だったように記憶している。今から思えば素晴らしく低いスペックのマシンで、当然ながら漢字も出ないような状態で VT52 や VT100 をつないで使っていた。1987 年に私に「UNIX をやってみないか」と勧めてくださった黒住先生、後に導入された Sun3/50 を一緒になって使い込んでいった栗原、当時の特研の指導教員だった吉川先生。皆さんに感謝したい。

最後に。私はこの第二版を書き、まとめる為にまるまる二週間を費やした。この期間私を支えてくれたすべての人々に感謝を忘れてはいけない。

1995.3.23 UNIX ガイド 編集責任者 安田 豊

¹山崎君の名誉の為に書いておきますが、各種記号等の一覧表は計算機センターが書きました。言語学と数学の知識がないのは山崎君ではありません。