

UNIX ガイド 追補版

京都産業大学
計算機センター事務室

'94.9.1

結 言

UNIX はオペレーティングシステム (OS) である。1970 年頃にベル研究所でミニコン用に開発された。当時は、IBM や GE 社の大型機発展時代で、MULTICS などというバッチ、TSS、オンライン何でもござれの巨大 OS が幅をきかせていた。

一方、DEC 社のミニコンが工業用や研究用に普及しつつあり、その OS として UNIX が注目されたのである。1970 年代にはベル研とその親会社である ATT が UNIX を増強し、そのシステムをソースコードまで含めて大学、研究機関などに頒布した。1980 年代には DEC 社をはじめとするミニコンメーカーが一齐に UNIX を OS としたコンピュータシステムを開発し、飛躍的に普及する。

UNIX が日本に上陸したのは、1970 年代の後半で、東京大学の石田晴久先生がベル研から DEC 社の LSI-11 上で持ち帰って、研究室で使ったのが最初とされる。1980 年には東京大学計算機センタの VAX-11 上で UNIX システムのサービスを開始している。その後の 2、3 年で、UNIX はミニコン用の標準 OS としての地位を確立し、日本のメーカーも争って UNIX ワークステーションを開発する。

本学には 1985 年にオムロン社 (当時立石電機) のスーパーメイトにより UNIX が計算機科学研究所に設置された。利用目的は画像処理でなんと FORTRAN と SPIDER という画像処理ライブラリを導入していた。当時の UNIX はシステム V のバージョン 1.0 で、その基本部は 10MB、アプリケーションを入れても全部で 20MB であり、メモリ 1MB、ディスク 80MB で同時に 10 本の TSS ユーザが可能であった。これを DEC の VT100 ターミナルにつないで、使っていたのである。このマシンは当然のことながら現役を引退しているが、今でも動く。

2 年後の 1987 年には、SUN3 や SONY の EWS などが設置され、ネットワークが組めるようになった。1989 年の工学部設置とともに、本学にもワークステーションが急増し、ウィンドウシステムやファイルシステムが本格化していく。

今回、本学計算機センタによって、UNIX サーバや多数のユーザマシンが整備され、一般ユーザが UNIX ワークステーションを利用する時代となった。そのガイドブックが発行される機会に、UNIX の誕生から本学へのかかわりまでを簡単に述べて、思いを新たにしたい。

工学部 黒住 祥祐

目次

第 1 章 続 UNIX もっともっと	1
1.1 ファイルの整理	1
1.1.1 ファイルの大きさを調べる	1
1.1.2 身に覚えのないファイル	2
1.1.3 ファイルを削除する	4
1.1.4 ディスクの利用量をチェックする	4
1.2 情報検索	5
1.2.1 学内の利用者情報の検索	5
1.2.2 学外の組織などの情報を調べる	6
1.3 機能追加、変更について	9
1.3.1 メールのリプライ、ニュースのフォローの時の引用記号について	9
第 2 章 Emacs	11
2.1 なにそれ?	11
2.2 Emacs の仕組み	12
2.3 これだけ覚えたら Emacs は使える!	12
2.4 もうちょっと覚えよう	13
2.5 日本語入力	14
2.6 ちょっと中級者	15
2.6.1 検索	15
2.6.2 picture モード	15
2.6.3 画面分割	15
2.6.4 バッファリスト	15
2.6.5 モードライン	15
2.7 上級者に向けて	16
2.8 とりあえずここまで	17
第 3 章 L^AT_EX	18
3.1 L ^A T _E X って何?	18
3.1.1 L ^A T _E X とは?	18
3.1.2 L ^A T _E X の特徴	18
3.1.3 L ^A T _E X の作業の進めかた	18
3.2 それでは使ってみましょう	19
3.3 L ^A T _E X におけるルール	22
3.3.1 最低限のルール	22
3.3.2 改行について	22
3.3.3 環境とコマンドの有効範囲について	23

3.3.4	ドキュメントスタイルについて	23
3.3.5	見出しの種類	24
3.4	自分の命令 (マクロ)	25
3.4.1	簡単な命令を作ってみよう	25
3.4.2	引数を持っている命令の作り方	26
3.4.3	マクロの名前の付け方	26
3.5	ファイルを分けて \LaTeX を使う方法	26
3.6	日本語 \LaTeX 定番スタイル集の使い方	27
3.7	いろいろなコマンドと環境	28
3.7.1	特殊文字	28
3.7.2	特殊文字でもそのまま出力する	29
3.7.3	文字の空白	30
3.7.4	水平方向と垂直方向の空白	31
3.7.5	改行と改ページ	32
3.7.6	引用	33
3.7.7	箇条書き	34
3.7.8	右寄せ、中央寄せ、左寄せ	37
3.7.9	文字の大きさ	37
3.7.10	書体	38
3.8	表の作り方	39
3.9	タイトルの作り方	41
3.9.1	概要の作り方	41
3.10	注釈と参照	42
3.10.1	傍注	42
3.10.2	脚注	42
3.10.3	相互参照	42
3.11	数式	44
3.11.1	数式的环境	44
3.11.2	添字	45
3.11.3	平方根	45
3.11.4	分数	46
3.11.5	括弧	46
3.12	記号など	48
3.12.1	雑記号	48
3.12.2	空白を空ける文字	48
3.12.3	アクセント	48
3.13	数式環境で使える記号	50
3.13.1	雑記号	50
3.13.2	ギリシャ文字	52
3.13.3	関数	53
3.14	例題	54
3.14.1	線形代数	54
3.14.2	微分積分学	59
3.14.3	ガウス積分	60

3.14.4	ベータ (B) 関数とガンマ (Γ) 関数	62
3.14.5	量子力学	64
3.14.6	電磁気学:マクスウェル方程式	65
3.15	エラーの対処	67
3.15.1	エラーの表示	67
3.15.2	エラーの種類	70
第 4 章	AUCTEX	71
4.1	AUCTEX で L ^A T _E X 生活が変わる	71
4.2	AUCTEX の起動	71
4.3	C-c	72
4.4	C-c C-e	72
4.5	C-c {	72
4.6	C-c C-c	72
4.7	C-c ‘	73
4.8	その他の機能	73
4.8.1	部分的なコンパイル	73
4.8.2	分割ファイルの管理	73
4.8.3	L ^A T _E X マクロの補間	74
4.8.4	複数行のコメントの付け外し	74
4.8.5	数式モードの支援	74
4.9	最後に	74
第 5 章	パソコンから UNIX	75
5.1	はじめに	75
5.1.1	パソコンで UNIX を使うと何が嬉しいか	75
5.1.2	予備知識	76
5.1.3	これ以降の読み方	78
5.2	FMV で telnet	80
5.2.1	端末ソフトの起動	80
5.2.2	ホストコンピュータへの接続	81
5.2.3	さあ、login しよう	82
5.2.4	ちょっとメッセージ	82
5.2.5	パスワードを変える	83
5.2.6	logout をしよう	83
5.2.7	端末ソフトを終了する	83
5.2.8	困った時の対処	84
5.3	マックで telnet	87
5.3.1	端末ソフトの起動	87
5.3.2	さあ、login しよう	87
5.3.3	ちょっとメッセージ	88
5.3.4	パスワードを変える	88
5.3.5	logout をしよう	88
5.3.6	端末ソフトを終了する	88

5.3.7	端末ソフトを起動する (その2)	88
5.3.8	少し高度な使い方	90
5.3.9	困った時の対処	92
5.4	FMRでtelnet	94
5.4.1	端末ソフトの起動	94
5.4.2	ホストコンピュータへの接続	95
5.4.3	さあ、loginしよう	96
5.4.4	ちょっとメッセージ	97
5.4.5	パスワードを変える	97
5.4.6	logoutをしよう	98
5.4.7	端末ソフトを終了する	98
5.4.8	cc2000以外のホストコンピュータへの接続	98
5.4.9	困った時の対処	99
5.5	Conturaでtelnet	102
5.5.1	端末ソフトの起動	102
5.5.2	ホストコンピュータへの接続	103
5.5.3	さあ、loginしよう	103
5.5.4	ちょっとメッセージ	103
5.5.5	パスワードを変える	104
5.5.6	logoutをしよう	104
5.5.7	端末ソフトを終了する	105
5.5.8	困った時の対処	105
5.6	FMVでX	107
5.6.1	端末ソフトの起動	107
5.6.2	loginする	108
5.6.3	ちょっとメッセージ	109
5.6.4	パスワードを変える	109
5.6.5	logout	109
5.6.6	端末ソフトを終了する	110
5.6.7	困った時の対処	110
第6章	MODEMからUNIX	112
6.1	計算機センター管理のターミナルサーバー	112
6.2	アクセス (接続) するまえに	112
6.3	アクセスしてみましょう!!	113
6.3.1	Annexターミナルサーバの場合	114
6.4	さらに詳しい使い方	114
第7章	NeXTを上手に使おう	115
7.1	NeXTファイル管理術	115
7.1.1	はじめに	115
7.1.2	ファイルビューアの各部名称	116
7.1.3	ファイルとは	116
7.1.4	フォルダで整理	116

7.1.5	コピー	117
7.1.6	移動	118
7.1.7	コピーと移動の違い	118
7.1.8	ファイルをフォルダにドラッグすると...	118
7.1.9	名称変更	118
7.1.10	削除	119
7.1.11	リンク	119
7.1.12	アクセス権	120
7.1.13	アプリケーションドック	121
7.1.14	シェルフ	121
7.1.15	フロッピーディスク	122
7.1.16	光磁気ディスク	123
7.1.17	おわりに	123
7.2	NeXTで使うネットワークサービス	124
7.2.1	最初に	124
7.2.2	文通しよう	124
7.2.3	ニュースしよう	128
7.2.4	最後に	133
第8章 Mathematica		134
8.1	Mathematicaってなあに?	134
8.1.1	ともかく起動、そしてやってみる!	134
8.1.2	その他の場所の Mathematica	136
8.2	ノートブックとコマンドラインについて	137
8.3	Mathematicaの簡単な命令	137
8.4	入出力一般	139
8.5	さらに進みたい人には	141
A	リファレンス	142
A.1	UNIX コマンド	142
A.1.1	ファイル管理に関するコマンド	143
A.1.2	ファイルに関する雑多なコマンド	144
A.1.3	プリンタに関するコマンド	146
A.1.4	プロセスに関するコマンド	146
A.1.5	アクセス権、アクセス制御に関するコマンド	147
A.1.6	マニュアルに関するコマンド	147
A.1.7	雑多なコマンド	148
A.1.8	現在使っているコンピュータに関するコマンド	149
A.1.9	利用者に関するコマンド	149
A.1.10	ネットワークサービスに関するコマンド	149
A.2	Emacs コマンド	151
A.2.1	絶対覚えておいた方がいいもの	151
A.2.2	必要に応じて覚えるもの	152

第 1 章

続 UNIX もっともっと

1.1 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、とにかくにも不要なファイルを消去しないとイケません。

1.1.1 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ありますが、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¹。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。

```
cc2000(84)% ls -l printscreen.ps
-rw-r--r--  1 worker    330481 May  6 15:01 printscreen.ps
cc2000(85)%
```

上記の例では printscreen.ps というファイルの大きさは 330481 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、330481 文

¹UNIX ガイドの 109 ページでアクセス権限を調べるために紹介しました。

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²。

`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³。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⁴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 Byte , 1 KiloByte = 1000 Byte (場合によっては 1 KiloByte = 1024 Byte) 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まので、`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。

1.1.2 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。

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 `ccns001~015` 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。

²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。

³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。

⁴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。

jsykojin.dic csosf01~41 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。item[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。

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 csosf01~41 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。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あります。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る。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ようであれば計算機センターの窓口か、sandai.question ニュースグループに相談してください。

1.1.3 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。

1.1.4 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   1000   1000
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 1433KiloByte 使っていることになります。
quota	利用できるファイルの量の上限を表します。例では 30,000KiloByte、つまり 30MegaByte 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 ファイルあることになります。
quota	利用できるファイルの数の上限を表します。例では 1000 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。

1.2 情報検索

1.2.1 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⁵。

`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の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.      ]
```

⁵ 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あります。

```
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。

1.2.2 学外の組織などの情報を調べる

1.2.1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。

1.3 機能追加、変更について

1.3.1 メールのリプライ、ニュースのフォローの時の引用記号について

cc 環境では `supercite` と呼ばれるツールを採用しました。これはメールやニュースの記事を書く時に使う引用記号について機能を強化したものです。

利点は「元記事を書いた人に応じて引用記号を登録することが出来る」という事です。

例えば `yasuda@cc.kyoto-su.ac.jp` さんが出したメールもしくは記事を引用する時は引用記号として「やっさん」を使い、`kaihara@cc.kyoto-su.ac.jp` さんのものに対しては「開原」を使うという様に出来ます。記事を引用する時の操作方法が `UNIX ガイド` の記述から若干変わりますので、注意して下さい。

これ以降に手順に沿って書きます。一度試して下さい。

MH-e ではリプライしたい記事を読んでいる状態から

```
a
to
C-c C-y
```

と入力すると、`supercite` が起動します。

次に GNUS ではフォローしたい記事を読んでいる状態から

```
F
y
```

のキー入力で、`supercite` が起動します。

さて、`supercite` が起動するとまず

```
Complete cite attribution string: (default masakazu)
```

で入力待ちになります。適当な引用記号を入力します。default の `masakazu` でよい時は `<Return>` キーを押すだけです。普通は引用元記事を書いた人の名字を漢字で使うので、

```
Complete cite attribution string: (default masakazu) 松浦氏
```

の様に入力します。すると、

```
登録しますか? (y or n)
```

と聞いてきます。ここで登録をすると次回からはそれが default となります。次に引用元の記事を表示している窓とは別の窓が開いて

```
>>>> On Tue, 16 Aug 1994 11:07:25 --900, matsuura@asca.kyoto-su.ac.jp \
(Masakazu Matsuura) said:
```

と表示されると同時にコマンドラインに

```
Showing reference header 1.
```

と表示されます。これは、「窓には supercite が用意している引用記事参照のためのヘッダ形式の 1 番目が表示されている」事を教えてくれているのです。 **n** を押す事でヘッダの形式の表示が

```
>>>> In article <9408160207.AA15861@asca.kyoto-su.ac.jp>, \
matsuura@asca.kyoto-su.ac.jp (Masakazu Matsuura) writes:
```

に変わり、コマンドラインには **Showing reference header 2.** と表示され、繰り返し **n** を押す事で 0 から 6 までのヘッダの形式が次々に表示されるので、適当なものが表示された所で **<Return>** キーを押します。

以上で引用記号の付いた引用元記事が出来るので、あとは切ったり貼ったり加えたりして自分の記事を作ります。その時注意すべき点は、文章の 1 行 1 行が短めになるようにこまめに改行する事です。

第 2 章

Emacs

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)

2.1 なにそれ?

さんだい君 (以下さ): コンプ博士、コンプ博士、Nemacsってなんですか?

コンプ博士 (以下博): うむ? ああ、それは UNIX で使われている多機能なスクリーンエディタじゃな。Emacs の日本語対応版の物をそう書く。普通は Emacs と書かれるな。

さ: えまくす? スクリーンエディタ?

博: ああ、すまん。さんだい君には難しかったか。Emacs は「イーマックス」と読む。スクリーンエディタとは画面全体に文章を表示させて編集できる簡単なワープロのような物じゃ。昔はラインエディタといって 1 行ずつしか扱えなかったのじゃが世の中便利になったもんじゃ。(しみじみ。) 思い起こせばあの頃は …

さ: (まずい。話をそらさねば。) コンプ博士、さっき多機能って言いましたが何ができるんですか?

博: そうじゃな。まず基本はワープロのような使い方なのじゃが、他にニュースやメールを読み書きしたりできる。この UNIX はインターネットという全世界に通じるネットワークに繋がっているから世界中の人と接することができるぞ。ニュースなんかは何千というジャンルに分かれておってな、真面目な話題から楽しいおしゃべりまで幅が広いからきっと自分の趣味にあった話題の場が見つかるじゃろう。

さ: おもしろそうですね。コンプ博士、使い方を教えて下さい!

博: 慌てるでない。まずは基本となる Emacs の使い方から覚えるのが先じゃ。

さ: 分かりました。じゃ、それを教えて下さい!

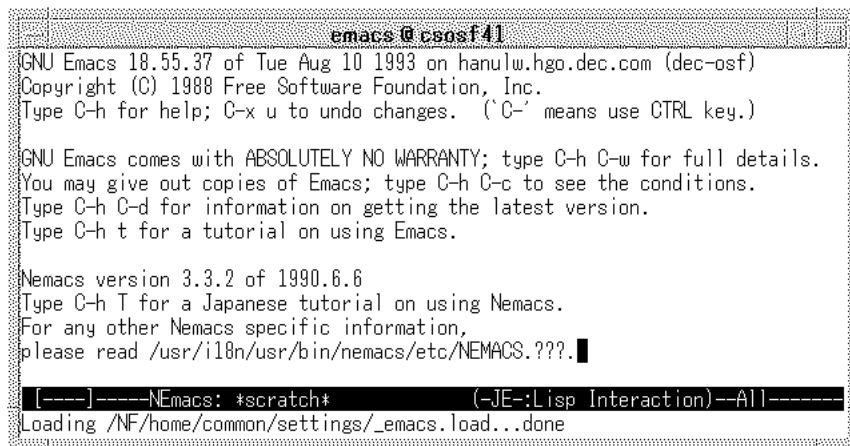
2.2 Emacs の仕組み

Emacs 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状態から始め、それをバッファ上で編集します。そしてディスクにファイルとして保存します。また、Emacs 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。Emacs 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 Emacs 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 Emacs のメッセージを表示する行が表示されます。

また、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バッファを幾つか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。

2.3 これだけ覚えたら Emacs は使える!

これ以降の記述は「UNIX ガイド」の第3章「UNIX それから」まで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モードを立ち上げて下さい。そしておもむろに **emacs** と打ち込んでリターンキーを押してみてください。いー、えむ、えー、しー、えす、りたーんです。すると英語ばかりの画面が現れます。



```
emacs @ csosf41
GNU Emacs 18.55.37 of Tue Aug 10 1993 on hanulw.hgo.dec.com (dec-osf)
Copyright (C) 1988 Free Software Foundation, Inc.
Type C-h for help; C-x u to undo changes. ('C-' means use CTRL key.)

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
You may give out copies of Emacs; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.
Type C-h t for a tutorial on using Emacs.

Nemacs version 3.3.2 of 1990.6.6
Type C-h T for a Japanese tutorial on using Nemacs.
For any other Nemacs specific information,
please read /usr/i18n/usr/bin/nemacs/etc/NEMACS.???.

[----]----NEmacs: #scratch#          (-JE:Lisp Interaction)--All-----
Loading /NF/home/common/settings/_emacs.load...done
```

図 2.1: Emacs 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物をいいます。例えば **C-h T** 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。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。チュートリアル の 終 わ り 方 は い く つ か あ り ま す が、Emacs 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 一 度 は や っ て み る こ と を お 薦 め し ま す。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。矢印の書いであるキーです。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ありますのでデリートキーでうまく行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。

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きそうな人に聞いて下さい。

話を戻しましょう。単に Emacs を立ち上げた場合 ***scratch*** というバッファが開きます。バッファというのは Emacs 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。Emacs 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。

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」の名のとおり落書き扱いですからこれを保存する時は何か名前をつけてあげないといけません。**C-x C-w** として下さい。すると **Write file** と出てきますので、ここにファイル名を書いてやればその名前で保存します。

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。とりあえず今後の説明のため **test** というファイル名で保存したとします。正常に動作すると **Wrote** とでます。バッファの名前が **test** に変わりましたね。

続けて何か書いて下さい。そして終わろうと **C-x C-c** をすると なにかメッセージがでてきます。

```
Save file ~/test ? (y or n)
```

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。

これだけで Emacs 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 Emacs 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。

2.4 もうちょっと覚えよう

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 Emacs 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 **emacs 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、Emacs が立ち上がった状態で **C-x C-f** とすると

```
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。また、この場合ファイル名の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>** 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。

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-l 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。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。

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っても構いません。

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** で挿入することができます。行の移動なんかが楽にできますね。

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-_** (アンダーバー) でいくらでも元に戻せます。**C-x u** でも同じです。

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。名前を変えたいなら **C-x C-w** ですね。

2.5 日本語入力

では次は漢字の入力にいきましょう。emacs は egg という日本語変換システムで日本語を入力します。egg については「UNIX ガイド」の第 3 章 9 節を参照して下さい。

C- で漢字入力モードになります。もう一度押すと元に戻ります。漢字入力はローマ字で入力し、変換はスペースキーです。

スペースキーを押すとシステムは入力した文字を解釈して幾つかの文節に分けます。その文節の分け方があっているなら、そのままスペースキーまたは **C-n** でどんどん次の候補を表示してくれるので目当ての漢字を選んで下さい。行きすぎた場合は **C-p** で前候補に戻ります。もし文節がおかしいなら **C-o**、**C-i** で文節を伸縮できます。文節の前後移動は **C-b**、**C-f** です。

打ち間違いに気付いたらバックスペースでひらがなに戻るのでカーソルキー (使えない機種もあるのでその場合は **C-b**、**C-f**) で移動して直すことができます。

覚えておくと便利なのが **M-h** (現在の文節をひらがなにする) と **M-k** (同じくカタカナにする) です。また、変換途中に **M-s** で候補の一覧を表示することができます。この場合は **C-n**、**C-p**、**C-f**、**C-b**、数字キーで選んでリターンで決定して下さい。記号の入力をしたいなら **C-^** で記号入力モードになります。そしてすべて思い通りに変換できたらリターンキーを押すと確定します。

尚、小さな文字、例えば「あ」ならローマ字入力は頭に **x** を付けて **xa** となります。

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。

2.6 ちょっと中級者

2.6.1 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。停止したい時は **<ESC>**(エスケープキー) です。

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いません。

他にも逆向きサーチなどありますがここでの説明は省きます。

2.6.2 picture モード

Emacs は通常挿入モードですが、これを上書きモードに変更することができます。**M-x picture-mode** で現在のバッファを一時的に **picture-mode** にします。**picture-mode** の終了は **C-c C-c** です。このモードは他にも文字入力後のカーソル移動方向の変更ができます。

2.6.3 画面分割

Emacs は画面を何分割かして幾つかのバッファを同時に表示することができます。**C-x 2** で上下に二分割します。バッファ間の移動は **C-x o** です。逆にバッファ表示を減らすときは **C-x 1** でカーソルのあるバッファ以外を隠しますし、**C-x 0**(ゼロ) でカーソルのあるバッファを隠します。隠すだけで完全に消すはしませんし、保存もしません。注意して下さい。

ウインドウサイズは **^** でカーソルのあるウインドウを縦方向に拡大します。ふたつ以上のバッファに同じファイルを読み込むことも可能です。その場合はそれら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れかを編集すると他のバッファにもそれが反映されます。

2.6.4 バッファリスト

Emacs 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。

2.6.5 モードライン

モードラインとは Emacs 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 Emacs を立ち上げたときは

```
[----]-----NEmacs: *scratch*          (-JE-: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

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。

-JE-

バッファの漢字コードを表します。左から順にファイル読み書き、キーボードからの入力、画面入出力をそれぞれ表していて、E:EUC N:NEC S:SJIS J:JIS、とそれぞれ頭文字で示されます。通常使うのは EUC と JIS でしょう。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、Emacs 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 DOS と PC の世界で使われています。

ファイルへの書き込みモードを変更するには **C-x C-k f** です。E、N、S、J の内、必要な物を選んで下さい。

Lisp Interaction

Emacs のモードを表示します。C モードや Fortran モード、TeX モード、gnus モード、mh-e 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。

2.7 上級者に向けて

Emacs 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。ちょっと自分のホームディレクトリで `cat .emacs<Return>` としてみして下さい。以前に何もしていなければ

```
cc2000(81)% cat .emacs
;;
;;      Emacs common settings.
;;

(load "/NF/home/common/settings/_emacs.load")
cc2000(82)%
```

と表示されると思います。**.emacs** は Emacs の環境設定をするファイルで、Emacs は立ち上がるときにこのファイルを読み込んで機能を拡張して始まります。

先程表示されたのは計算機センターが標準で用意してくれている物で、最初の3行は注釈ですので無視して構いませんが次の(load "/NF/home/common/settings/_emacs.load")の行は共通設定をしている大事な部分ですので消したりしてはいけません。この後に説明する事はこの行の後に付けるようにして下さい。

ただしこのファイルに不具合があると最悪、Emacs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いましょう。因みにemacs -qとすれば.emacsを無視して立ち上がりますのでどうしてもなくなったときはこれで.emacsを直して下さい。

簡単なカスタマイズから紹介しましょう。例えば他のワープロに慣れている人の中には「ん」を入力するのに「nn」とnを2回打ち込む癖のついている人がいると思います。そんな人は.emacsに

```
(defrule "nn" "ん")
```

の1行を追加して下さい。括弧も必要です。そして保存終了すると次にEmacsを立ち上げ直すと「nn」で「ん」と変換するようになります。

ちなみに.emacsはEmacs-Lispという言語を用いて記述されます。これを使って他にも色々設定できるのですがEmacs-Lispはここでは説明しきれないので幾つか簡単な例をA.2で挙げておきます。後は必要に応じて勉強して下さい。

2.8 とりあえずここまで

Emacsは言い出すときりがありません。実際、Emacs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.2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Emacs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 3 章

LaTeX

3.1 LaTeX って何？

3.1.1 LaTeX とは？

LaTeX とはみなさんがよく知っているワープロによく似た物です。しかしワープロと違い、LaTeX ではある手順を踏まないと望むものができません。

つまり LaTeX が使えるというのは「手順」を知っているということになります。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。

3.1.2 LaTeX の特徴

LaTeX の特徴をいくつかあげておきます。

1. UNIX ガイドも LaTeX で書かれている。つまりこのようなドキュメントを書きたい人には最適。
2. 機種に依存しない。パーソナルコンピュータでも UNIX でも同じようにできる。
3. 色々なフォント (文字の形) が使える。フォントさえあればいくらでも文字の形を変えられる。
4. 目次・索引・文献リストを簡単に作ることができる。
5. 世界中で標準化されており LaTeX 形式で出まわっている文書も多い。
6. 複雑な数式や表を扱うことができる。理科系の人にとっては必需品である。

など色々な特徴があります。

3.1.3 LaTeX の作業の進めかた

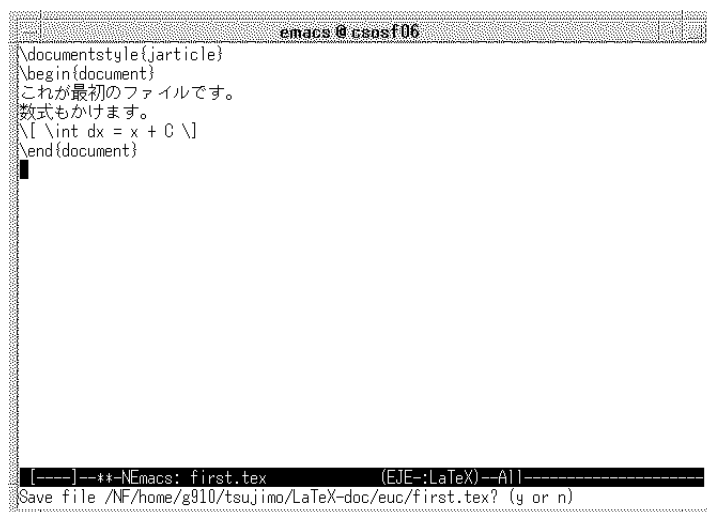
大まかに言うと次のようにして進めていきます。

1. エディタ (Emacs, Mule など) で文書ファイルを作成する。
2. 作成したファイルを LaTeX で処理 (コンパイル) する。
3. 処理したファイルをプリントアウトする。

3.2 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 `emacs first.tex &<Return>` と入力し Emacs 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 Emacs 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 Emacs 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 UNIX ガイド の 54 ページ、3.5.2 に詳しく説明してあります。)

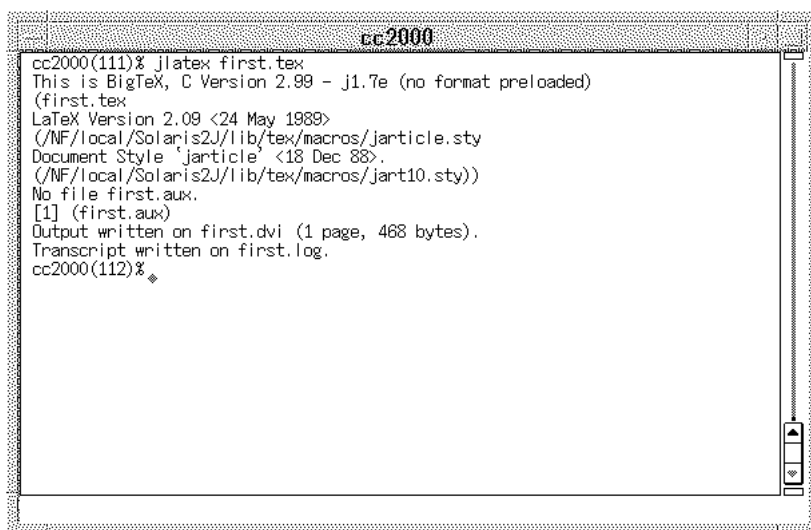


```
emacs @ csust06
\documentstyle{jarticle}
\begin{document}
これが最初のファイルです。
数式もかけます。
\[ \int dx = x + C \]
\end{document}

[---]--**NEmacs: first.tex (EJE-LaTeX)-A1 [---]
Save file /NF/home/g910/tsujimo/LaTeX-doc/euc/first.tex? (y or n)
```

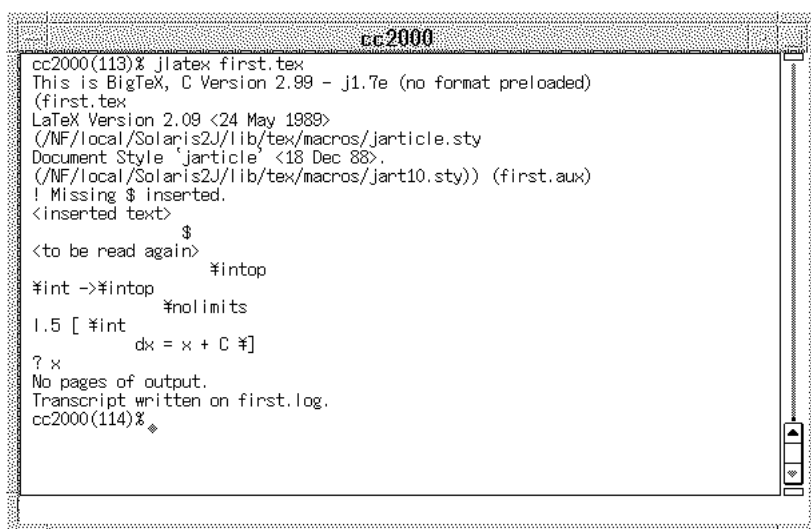
図 3.1: first.tex の画面

ファイルが出来たら Emacs 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 `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(111)% jlatex first.tex
This is BigTeX, C Version 2.99 - j1.7e (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))
No file first.aux.
[1] (first.aux)
Output written on first.dvi (1 page, 468 bytes).
Transcript written on first.log.
cc2000(112)%
```

図 3.2: うまくいった時の画面



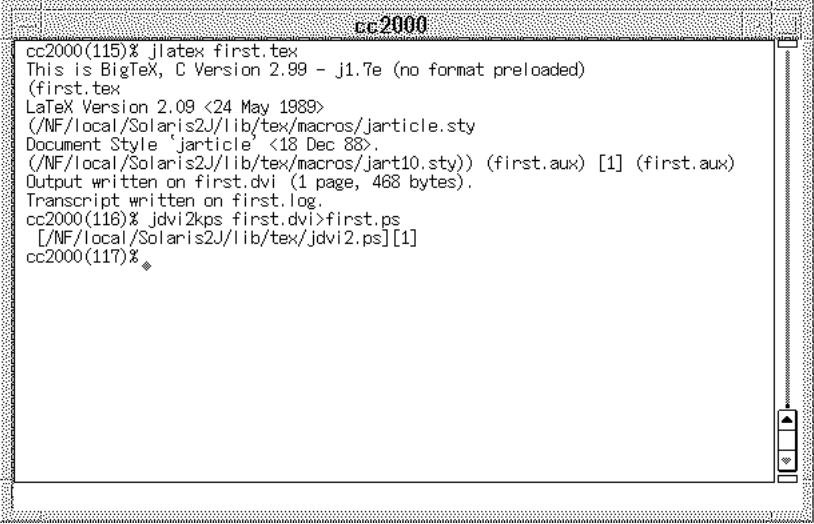
```
cc2000
cc2000(113)% jlatex first.tex
This is BigTeX, C Version 2.99 - j1.7e (no format preloaded)
(first.tex
LaTeX Version 2.09 <24 May 1989>
(/NF/local/Solaris2J/lib/tex/macros/jarticle.sty
Document Style 'jarticle' <18 Dec 88>.
(/NF/local/Solaris2J/lib/tex/macros/jart10.sty)) (first.aux)
! Missing $ inserted.
<inserted text>
$
<to be read again>
%intop
%int ->%intop
%nolimits
1.5 [ %int dx = x + C %]
? x
No pages of output.
Transcript written on first.log.
cc2000(114)%
```

図 3.3: エラーがあるときの画面

もしエラーがあった場合は `x` を入力し処理を中断した後、今までの所で間違いが無いか確認してください。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。

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、次のようにして変換してやります。

`jdvi2kps first.dvi > first.ps`<Return>と打ち込んでやります。これ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。

A terminal window titled 'cc2000' showing the execution of LaTeX and conversion commands. The output shows the successful compilation of 'first.tex' into 'first.dvi' and its conversion to 'first.ps'.

```
cc2000(115)% jlatex first.tex
This is BigTeX, C Version 2.99 - j1.7e (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(116)% jdvi2kps first.dvi>first.ps
[ /NF/local/Solaris2J/lib/tex/jdvi2.ps][1]
cc2000(117)%
```

図 3.4: 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。ここでは 21 情報処理教室のプリンターから出力します。

`lpr -Pcspr01 first.ps`<Return>と打ち込んで下さい。これに関するのことは UNIX ガイド 61 ページ「3.6 印刷」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。しかし手順はこれだけではありません。あなた自身を使い勝手のいいような仕方でもらえればいいのですから。

3.3 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 LATEX 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。 │
└───┘
```

3.3.1 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。)

3.3.2 改行について

Emacs で編集しやすくするため適当に改行を挿入しても出力結果は影響を受けません。段落を区切るときは空行 (改行だけの行) を入れます。また改行だけしたいときは `\\` を文の最後に入れます。

```
┌─── 入力 ───┐
│             │
│ こんにちは。 │
│             │
│ いいお天気ですね。 │
│             │
│             │
│ 明日はどうでしょうか？ │
└─── 出力 ───┘
│             │
│ こんにちは。いいお天気ですね。 │
│             │
│ 明日はどうでしょうか？ │
└───┘
```

```
┌─── 入力 ───┐
│             │
│ こんにちは。\\ │
│             │
│ いいお天気ですね。 │
└───┘
```

明日はどうでしょうか?

—— 出力 ——

こんにちは。

いいお天気ですね。

明日はどうでしょうか?

3.3.3 環境とコマンドの有効範囲について

前のセクションで紹介した`\begin{document}... \end{document}`のように`\begin{...}... \end{...}`といった形になっているものを「環境」といいます。

`\begin{...}`から`\end{...}`まではある一定の状態になっています。

「—— 入力 ——

```
\begin{center}
```

ここは何処\

私は誰

```
\end{center}
```

さー誰でしょう?

—— 出力 ——

ここは何処

私は誰

さー誰でしょう?

つぎに文書中で文字のフォントを変える`\bf`のようなものを「コマンド」といいます。

この「コマンド」には有効範囲が決められています。

`\bf`を例にとると、`\bf`と打っただけならばその場所以降すべての部分が太字体に変わってしまいます。望んでいる部分だけを変えたいのならば`{\bf}`のように範囲を指定しなければいけません。例をあげておきます。

「—— 入力 ——

もしかしてここは`{\bf 三叉路か}`? よくわからない?

もしかしてここは`\bf 三叉路か`? よくわからない?

—— 出力 ——

もしかしてここは**三叉路か**?

もしかしてここは**三叉路か**? よくわからない?

3.3.4 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。

また次のような時もあります。

```
\documentstyle[...]{jarticle}
```

ここで括弧 [] の中を「オプション」といって次のような指定ができます。「オプション」は必要がなければ書く必要はありません。

文字の大きさ .. 文書全体の文字の大きさを指定します¹。11pt や12pt といった指定ができます。

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。デフォルトではa4j になっています。

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。

3.3.5 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。

`\part{...}`... パート (部) の見出し

`\section{...}`... セクション (節) の見出し

`\chapter{...}`... チャプター (章) の見出し

`\subsection{...}`... サブセクション (小節) の見出し

`\subsubsection{...}`... サブサブセクション (小小節) の見出し

`\paraguraph{...}`... パラグラフ (段落) の見出し

`\subparaguraph{...}`... サブパラグラフ (小段落) の見出し

上の中で`\chapter{...}`はjreport jbook でのみ使用できます。

		jarticle	jreport,jbook
部	<code>\part</code>	Part 1	Part 1
	<code>\chapter</code>		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>\paraguraph</code>	[1.1.1.1]	[1.1.1.1.1]
小段落	<code>\subparaguraph</code>	[1.1.1.1.1]	[1.1.1.1.1.1]

¹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。

ために次のように入力します。

```
┌─── 入力 ───┐
\section{太陽系第3番惑星地球}
地球っていうんだよね。
\subsection{日本国}
JAPAN?
\subsubsection{京都産業大学}
環境抜群
\paragraph{計算機センター}
なぜか2階が入口
\subparagraph{3階 NeXT 部屋の住人}
変な奴が揃っているところ
└──────────┘
```

1 太陽系第3番惑星地球

地球っていうんだよね。

1.1 日本国

JAPAN?

1.1.1 京都産業大学

環境抜群

計算機センター なぜか2階が入口

3階 NeXT 部屋の住人 変な奴が揃っているところ

図 3.5: 上の実行結果の画面

3.4 自分の命令 (マクロ)

\LaTeX では自分の命令 (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。

3.4.1 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`\koumei` という命令を作ります。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`\koumei` を定義しておきます。

```
\newcommand{\koumei}{\begin{center}京都産業大学\end{center}}
```

ファイルのこれ以降の場所で`\koumei` 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。ここで使った`\newcommand` という命令は新しい命令を定義するものです。

3.4.2 引数を持っている命令の作り方

さっきの`\koumei` 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²。ここで`[1]` は引数の数、`#1` は引数の入る場所です。

3.4.3 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³。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。

3.5 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げにくく、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`\koumei` 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くって、その中に`\koumei` 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。

² これってただの center 環境ですね

³ あまり漢字はお薦めできません

```
\input{macro}
```

そうすればその場所に`macro.tex`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`\koumei`のマクロが使えるようになります。

3.6 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}` のところです) が入っていますので、各自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。

3.7 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。一方、環境とは、

```
\begin{...}&\end{...}
```

の対で使うコマンドのことです。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。

3.7.1 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、記号の! ” ’ () = - ‘ @ + ; * : , . / [] ? 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。
- `$` は、数字やアルファベットを`$`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3.11節のところまで。
- `%` は、`%`をつけた後の部分からその行の終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3.8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3.11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3.11節で。

```
——— 入力 ———  
$ a_{11}+b_{11}=0 $
```

⁴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。

—— 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{"5E}	~
^	\symbol{"7E}	^
\	{\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 3.7.2 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 3.7.2 節で紹介する方法で出力して下さい。

3.7.2 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

⁵但し TypeWriter フォントの時は大丈夫です。これが TeX 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。

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	<code>\verb!%! </code>
出力	<code>%</code>

入力	<code>\verb+sample サンプル ;:@]/. , こんなぐあい+</code>
出力	<code>sample サンプル ;:@]/. , こんなぐあい</code>

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
<code>\begin{verbatim}</code>
たとえば <code>!@#\$\$^&*(<>;: なんかも</code>
<code>[]{}'"\ -もみーんなこんなもんさ</code>
<code>\end{verbatim}</code>
出力
たとえば <code>!@#\$\$^&*(<>;: なんかも</code>
<code>[]{}'"\ -もみーんなこんなもんさ</code>

このようになります。

3.7.3 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、スペースバーで空けます。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。半角空白を何個も出力するには「\」で区切ります。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。

入力	<code>Time_shock</code>
出力	<code>Time shock</code>
入力	<code>Time______shock(space key を 6 回押しました。)</code>
出力	<code>Time shock</code>

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。

入力 `Time______shock`

出力 `Time shock`

コマンドの後には必ず空白を入れて下さい。そうしないとエラーの対象となります。

3.7.4 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。

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、「を入れます。」という言葉が前に 0.5cm 移動したと考えて下さい。

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 `\hspace{3.5cm}` を入れます。

出力

垂直方向に 3.5cm 空白を入れます。

⁷括弧の中の数字です。

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

3.7.5 改行と改ページ

改行するには、ワープロのように 1 回 return key を押しても出力では改行されていません。L^AT_EX で は、改行したいところで一行空白を入れるか、「`\\`」というコマンドをつける必要があります。

一行空白を作ると、改行されて少し頭が引込みます。つまり、段落ができるということです。一方、「`\\`」は、ただ改行するだけで頭は引込みません。

入力

空白を一行作ると改行されます。

そして、頭が引込みます。「`\verb!\\!`」はただ改行するだけです。\\
頭は引込みません。\\
このように改行されました。\\

出力

空白を一行作ると改行されます。

そして、頭が引込みます。「`\\`」はただ改行するだけです。
頭は引込みません。

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。

このように改行されました。

改ページをするには、`\newpage` があります。もし、表や、図が 2 ページにまたがっていたとすると、それらをすべて強制的に次のページに送ります。

3.7.6 引用

引用を行うものとして、コマンドの `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-横町の借屋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-横町の借屋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。

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。

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・をつけた（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・をつけた（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、

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
  \begin{itemize}
\item 箇条書き
  \end{itemize}
\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。

3.7.8 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。

3.7.9 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>

ここでひとつ。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。

3.7.10 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	出力
<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>	<code>abcdefg 京都産業大学</code> <code>abcdefg 京都産業大学</code> <code>abcdefg 京都産業大学</code> <code>abcdefg 京都産業大学</code>

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。

```
{\huge {\bf unix}}
unix
{\bf {\huge unix}}
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。注意しましょう。

3.8 表の作り方

表を作るのが、`tabular`¹³環境です。

```
\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。最後の列にはいりません。くれぐれも列指定の数と表の要素の列の数が変わらないようにして下さい。

┌─── 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。先ほどは、最後の列に、「\\」をつけませんでした。罫線を入れる表には、必要となりますので注意しましょう。

┌─── 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}
```

¹³ 「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。

3.9 タイトルの作り方

タイトルをつけるには、`\maketitle` というコマンドがあります。使い方としては、
`title{タイトル名}` 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 にかかけられ、結果ファイルが作られた日になります。

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ください。

3.9.1 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}.....\end{abstract}
```

とします。.... には、概要の内容が入ります。

```
----- 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}
\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。....
\end{abstract}
```

以下のようになります。

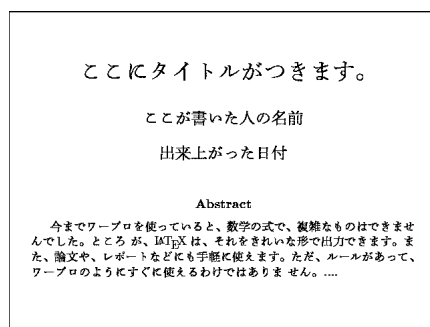


図 3.6: `\maketitle` による結果 (部分)

3.10 注釈と参照

本文外のところに書かれる注釈には、**傍注**と**脚注**があります。傍注は、本文の横に出力され、脚注は、本文の下に出力されます。

3.10.1 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。

3.10.2 脚注

脚注¹⁴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、`{...}`の部分がページの下に書かれます。

3.10.3 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くることがあります。そうすると、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¹⁵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。このラベルのところ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 3.10.3節参照...
...42ページ参照...
```

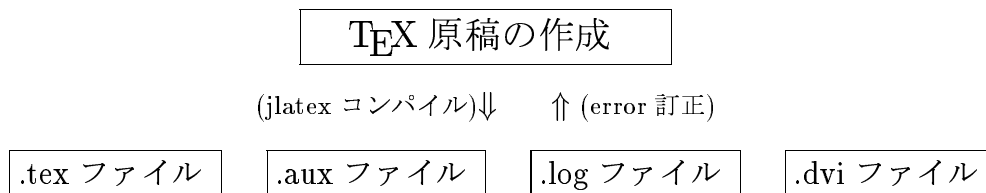
(復習)

`\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

¹⁴ここに出る注釈を脚注と呼びます。

¹⁵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。

イルの手順は、以下のような形になります。



.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。

3.11 数式

3.11.1 数式的环境

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 2 通りの方法があります。**displaymath** 環境とは、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** 環境とは、文中に数式を出力します。これも別の言葉で、**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。別の行に出力します。
- **math** か、`$. . .$` か、`\(...\)` のいずれか。本文中に出力します。

```
┌─── 入力 ───┐
\begin{displaymath}
a+b=b+a
\end{displaymath}
└──────────┘

─── 出力 ───┬──────────┘
               
$$a + b = b + a$$

```

```
┌─── 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、...
```

このように出力されます。

以上二つの環境は数式番号が振られません。また、2 つ以上の数式を用いるとエラーを起こします。

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¹⁶が付きませんでした。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}
└──────────┘
```

¹⁶ここでは、(第?章.1)となっていますが、一般的には、(1) とか (2) とかになります。このガイドでは、数式番号が変わるようなスタイルファイルを作ってしまったためにこうなっていました。

—— 出力 ——

$$a + b = c \quad (\text{第 3 章.1})$$

しかし、`equation` も一行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、すべてに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 ——

$$a + b = c \quad (\text{第 3 章.2})$$

$$c + d = e \quad (\text{第 3 章.3})$$

$$e = a + b + d$$

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。

3.11.2 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 ——

$$f(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n$$

3.11.3 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

$$\sqrt[3]{\sqrt{2}+1}$$

3.11.4 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

$$\frac{c_1}{b_1 + \frac{c_2}{b_2 + \frac{c_3}{b_3}}}$$

3.11.5 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¹⁷。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`、`\Bigg`

¹⁷大きさはポイント数によって違います。

をつけます。

```
入力
< \bigl( \Bigl( \biggl( \langle 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
```

$$\left\{ \left(\frac{ax+b}{cx+d} \right) + \left(\frac{ex+f}{gx+h} \right) \right.$$

3.12 記号など

これ以降は \LaTeX 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。

3.12.1 雑記号

入力	出力	入力	出力	入力	出力	入力	出力
<	i	>	i		—		
\lq	‘	\rq	’	\lbrack	[\rbrack]
\slash	/	\copyright	©	\aa	å	\AA	Å
\l	ł	\L	Ł	\S	§	\P	¶
\dag	†	\ddag	‡	\TeX	TeX	\LaTeX	LaTeX
\dots	...	-	-	--	-	---	—

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	\underbar{a}
出力	a
入力	ここで一発\underbar{下線}が欲しいなあ
出力	ここで一発下線が欲しいなあ

3.12.2 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 3.7.4 を参照して下さい。

入力	意味	入力	意味	入力	意味
\space	空白一文字	\empty	何もなし	\null	幅ゼロの箱
\thinspace	1/6em の幅	\negthinspace	-1/6em の幅	\enspace	1/2em の幅
\enskip	1/2em の幅	\quad	1em の幅	\qqquad	2em の幅
\,	3mu の幅	\>\<	4mu の幅	\>\<	5mu の幅
\!	-3mu の幅				
\smallskip	3pt の高さ	\medskip	6pt の高さ	\bigskip	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で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。

3.12.3 アクセント

入力	出力	入力	出力	入力	出力	入力	出力
\'a	à	\'a	á	\v{a}	ã	\u{a}	ä
\={a}	â	\^{a}	â	\.a	à	\H{a}	á
\~{a}	ã	\"a	ä	\d{a}	ą	\c{a}	ą
\b{aa}	aa	\t{aa}	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。)

3.13 数式環境で使える記号

これ以降は \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、本来使っていないものかどうか保証の限りではありません。

3.13.1 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 `\not\equiv`

出力 \neq

入力 `\not\supset`

出力 $\not\supset$

(筆者に数学的知識がありませんので、おかしい例示となっているかも知れませんが、勘弁して下さい。)

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>	\intop	<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>	\ointop
<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>\leftrighthrightarrow</code>	\leftrightarrow
<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>	$\left\{ \begin{matrix} abc \\ def \end{matrix} \right\}$

3.13.2 ギリシャ文字

ギリシャ文字の一覧表です。¹⁸

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	<code>\pi</code>
rho	P		ρ	<code>\rho</code>
sigma	Σ	<code>\Sigma</code>	σ	<code>\sigma</code>
tau	T		τ	<code>\tau</code>
upsilon	Υ	<code>\Upsilon</code>	$υ$	<code>\upsilon</code>
phi	Φ	<code>\Phi</code>	ϕ	<code>\phi</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>

¹⁸ 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。

3.13.3 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	$(\text{mod } n)$

3.14 例題

ここから、例題を通して、見ていくことにします。

3.14.1 線形代数

入力

```
\[ A =
\left( \begin{array}{ccc}
6 & -3 & -7 \\
-1 & 2 & 1 \\
5 & -3 & -6
\end{array} \right) \]
行列 A の固有値を求めてみる。固有方程式より\
\[ \Phi_A (t)=
\begin{array}{|ccc|}
t-6 & 3 & 7 \\
-1 & t-2 & -1 \\
-5 & 3 & t+6
\end{array}
\end{array}
= (t-1)(t-2)(t+1)=0 \]
となるのは、
\[ t=1,2,-1 \]
$t=1$ のとき、固有ベクトル $\vec{p}_1$ は、\
\[ \left( \begin{array}{ccc}
-5 & 3 & 7 \\
1 & -1 & -1 \\
-5 & 3 & 7
\end{array} \right) \left( \begin{array}{c}
x \\
y \\
z
\end{array} \right)
=\vec{0} \]
から、
\[ \vec{p}_1=
\left( \begin{array}{c}
2 \\
1 \\
1
\end{array} \right) \]
$t=2$ のとき、固有ベクトル $\vec{p}_2$ は、
\[ \vec{p}_2=
\left( \begin{array}{c}
\end{array} \right)
```

```

1\\
-1\\
1
\end{array} \right)\]

```

$t=-1$ のとき、固有ベクトル $\{\vec{p}_3\}$ は、

```

\left[ \vec{p}_3 =
\left( \begin{array}{c}
1 \\
0 \\
1
\end{array} \right) \right]

```

よって、対角化行列は、

```

\left[ P =
\left( \begin{array}{ccc}
2 & 1 & 1 \\
1 & -1 & 0 \\
1 & 1 & 1
\end{array} \right) \right]

```

その逆行列は、

```

\left[ P^{-1} =
\left( \begin{array}{ccc}
1 & 0 & -1 \\
1 & -1 & -1 \\
-2 & 1 & 3
\end{array} \right) \right]

```

となる。したがって、

```

\begin{eqnarray*}
P^{-1} A P & = & \\
\left( \begin{array}{ccc}
1 & 0 & -1 \\
1 & -1 & -1 \\
-2 & 1 & 3
\end{array} \right) & & \\
\left( \begin{array}{ccc}
6 & -3 & -7 \\
-1 & 2 & 1 \\
5 & -3 & -6
\end{array} \right) & & \\
\left( \begin{array}{ccc}
2 & 1 & 1 \\
1 & -1 & 0 \\
1 & 1 & 1
\end{array} \right) & & \\
& = & \left( \begin{array}{ccc}
\end{array} \right)

```

```

1 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & -1
\end{array} \right)
\end{eqnarray*}

```

と対角化された。

出力

$$A = \begin{pmatrix} 6 & -3 & -7 \\ -1 & 2 & 1 \\ 5 & -3 & -6 \end{pmatrix}$$

行列 A の固有値を求めてみる。固有方程式より

$$\Phi_A(t) = \begin{vmatrix} t-6 & 3 & 7 \\ -1 & t-2 & -1 \\ -5 & 3 & t+6 \end{vmatrix} = (t-1)(t-2)(t+1) = 0$$

となるのは、

$$t = 1, 2, -1$$

$t = 1$ のとき、固有ベクトル \vec{p}_1 は、

$$\begin{pmatrix} -5 & 3 & 7 \\ 1 & -1 & -1 \\ -5 & 3 & 7 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \vec{0}$$

から、

$$\vec{p}_1 = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

$t = 2$ のとき、固有ベクトル \vec{p}_2 は、

$$\vec{p}_2 = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

$t = -1$ のとき、固有ベクトル \vec{p}_3 は、

$$\vec{p}_3 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

よって、対角化行列は、

$$P = \begin{pmatrix} 2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

その逆行列は、

$$P^{-1} = \begin{pmatrix} 1 & 0 & -1 \\ 1 & -1 & -1 \\ -2 & 1 & 3 \end{pmatrix}$$

となる。したがって、

$$\begin{aligned} P^{-1}AP &= \begin{pmatrix} 1 & 0 & -1 \\ 1 & -1 & -1 \\ -2 & 1 & 3 \end{pmatrix} \begin{pmatrix} 6 & -3 & -7 \\ -1 & 2 & 1 \\ 5 & -3 & -6 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{pmatrix} \end{aligned}$$

と対角化された。

テクニック 其の一 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。`\begin{array}`で、行列を作ると宣言していて、次に、`{l}`は、2列の行列を作ると宣言しています。1は、左詰めにするということを表しています。1の他に、センタリングの**c**と右詰め**r**があるのは、**tabular** 環境と同じです。

```
—— 入力 ——  
  
x\sin \dfrac{1}{x} & (x\ne 0)\\  
0 & (x=0)
```

は、行列要素である。要素一個一個を区切るには、「&」を使います。もし、行列要素がなければ、空白で区切ればよろしいでしょう。例として、

```
—— 入力 ——  
  
\begin{math}  
\left( \begin{array}{ccc}  
1 & & 1\ \  
1 & 1 & 1\ \  
1 & & 1  
\end{array} \right)  
\end{math}
```

—— 出力 ——

$$\begin{pmatrix} 1 & & 1 \\ 1 & 1 & 1 \\ 1 & & 1 \end{pmatrix}$$

このように、(1,2) と (3,2) の要素が空白になっています。

テクニック 其の二 ‘=’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、

3.14.2 微分積分学

入力

```
\def\dfrac#1#2{{\displaystyle \frac{#1}{#2}}}  
\begin{eqnarray*}  
f(x) =  
\left \{ \begin{array}{ll}  
x\sin \dfrac{1}{x} & (x \neq 0) \\  
0 & (x=0)  
\end{array} \right.  
\end{eqnarray*}  
[ \left| x\sin \dfrac{1}{x} \right| < |x| < \varepsilon (x \to 0) \\  
[ \lim_{x \to 0} x\sin \dfrac{1}{x} = 0 = f(0) \\  
よって、$f(x)$は$x=0$で連続である。 \\  
[ \lim_{h \to 0} \dfrac{f(h)-f(0)}{h} =  
\lim_{h \to 0} \dfrac{h\sin \dfrac{1}{h}}{h} \\  
= \lim_{h \to 0} \sin \dfrac{1}{h} \\  
右辺の極限值が存在しないから$f'(0)$は存在しない。
```

出力

$$f(x) = \begin{cases} x \sin \frac{1}{x} & (x \neq 0) \\ 0 & (x = 0) \end{cases}$$
$$\left| x \sin \frac{1}{x} \right| < |x| < \varepsilon (x \rightarrow 0)$$
$$\lim_{x \rightarrow 0} x \sin \frac{1}{x} = 0 = f(0)$$

よって、 $f(x)$ は $x = 0$ で連続である。

$$\lim_{h \rightarrow 0} \frac{f(h) - f(0)}{h} = \lim_{h \rightarrow 0} \frac{h \sin \frac{1}{h}}{h} = \lim_{h \rightarrow 0} \sin \frac{1}{h}$$

右辺の極限值が存在しないから $f'(0)$ は存在しない。

3.14.3 ガウス積分

入力

```

\begin{eqnarray*}
I=\int_{0}^{\infty}e^{-x^2}\ dx\\
I=\int_{0}^{\infty}e^{-y^2}\ dy
\end{eqnarray*}
これから、\\
\begin{eqnarray*}
I^2=\int_{0}^{\infty}e^{-x^2}\int_{0}^{\infty}e^{-y^2}\ dx\ dy\\
\\
x=r\cos \theta , y=r\sin \theta
\end{eqnarray*}
とおくと、\\
\begin{eqnarray*}
I^2 &=& \int \int e^{-r^2} r dr d\theta \\
&=& \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{\pi}{4}
\end{eqnarray*}
よって、\\
\left[ I=\frac{\sqrt{\pi}}{2} \right]
したがって、\\
\left[ \int_{0}^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2} \right]

```

出力

$$I = \int_0^{\infty} e^{-x^2} dx$$

$$I = \int_0^{\infty} e^{-y^2} dy$$

これから、

$$I^2 = \int_0^{\infty} e^{-x^2} \int_0^{\infty} e^{-y^2} dx dy$$

$$x = r \cos \theta, y = r \sin \theta$$

とおくと、

$$\begin{aligned}
 I^2 &= \int \int e^{-r^2} r dr d\theta \\
 &= \int_0^{\infty} r e^{-r^2} dr \int_0^{\pi/2} d\theta \\
 &= \frac{\pi}{2} \left[-\frac{1}{2} e^{-r^2} \right]_0^{\infty} \\
 &= \frac{\pi}{4}
 \end{aligned}$$

$$= \frac{\sqrt{\pi}}{4}$$

よって、

$$I = \frac{\sqrt{\pi}}{2}$$

したがって、

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

3.14.4 ベータ (B) 関数とガンマ (Γ) 関数

入力

B 関数

```
\begin{equation}
B(x,y)=\int_0^1 t^{x-1}(1-t)^{y-1}dt \ , \ (x>0,y>0)
\label{siki:latex_1}
\end{equation}
```

Γ 関数

```
\begin{equation}\label{siki:latex_2}
\Gamma(x)=\int_0^{\infty} e^{-t}t^{x-1}dt \ , \ (x>0)
\end{equation}
```

B 関数と Γ 関数の関係は、(\ref{siki:latex_1}) 式と (\ref{siki:latex_2}) 式において、
 $[t \to t^2 \ , \quad s \to s^2]$ と書き換えて、

```
\begin{eqnarray*}
\Gamma(x)\Gamma(y) &= & 4\int_0^{\infty} e^{-t^2}t^{2x-1}dt \\
& & \int_0^{\infty} s^{2y-1}e^{-s^2}ds \\
&= & 4\int_0^{\infty} \int_0^{\infty} t^{2x-1}s^{2y-1}e^{-(t^2+s^2)} \\
& & dt \ , \ ds \quad (t=r\cos\theta, s=r\sin\theta) \\
&= & 4\int_0^{\pi/2} \int_0^{\infty} r^{2(x+y)-2}e^{-r^2} \\
& & \sin^{2y-1}\theta \cos^{2x-1}\theta \ , \ r dr \ , \ d\theta \\
&= & 2\int_0^{\pi/2} \sin^{2y-1}\theta \cos^{2x-1}\theta \\
& & \ , \ d\theta \cdot 2\int_0^{\infty} r^{2(x+y)-1}e^{-r^2}dr \\
&= & B(y,x)\Gamma(x+y)=B(x,y)\Gamma(x+y)
\end{eqnarray*}
```

よって、

```
[ B(x,y)=\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} ]
```

出力

B 関数

$$B(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt \quad (x > 0, y > 0) \quad (\text{第 3 章.4})$$

Γ 関数

$$\Gamma(x) = \int_0^{\infty} e^{-t}t^{x-1}dt \quad (x > 0) \quad (\text{第 3 章.5})$$

B 関数と Γ 関数の関係は、(第 3 章.4) 式と (第 3 章.5) 式において、

$$t \rightarrow t^2, \quad s \rightarrow s^2$$

と書き換えて、

$$\begin{aligned} \Gamma(x)\Gamma(y) &= 4 \int_0^{\infty} e^{-t^2}t^{2x-1}dt \int_0^{\infty} s^{2y-1}e^{-s^2}ds \\ &= 4 \int_0^{\infty} \int_0^{\infty} t^{2x-1}s^{2y-1}e^{-(t^2+s^2)} dt ds \quad (t = r \cos \theta, s = r \sin \theta) \\ &= 4 \int_0^{\pi/2} \int_0^{\infty} r^{2(x+y)-2}e^{-r^2} \sin^{2y-1} \theta \cos^{2x-1} \theta r dr d\theta \end{aligned}$$

$$\begin{aligned} &= 2 \int_0^{\pi/2} \sin^{2y-1} \theta \cos^{2x-1} \theta d\theta \cdot 2 \int_0^\infty r^{2(x+y)-1} e^{-r^2} dr \\ &= B(y, x) \Gamma(x+y) = B(x, y) \Gamma(x+y) \end{aligned}$$

よって、

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

3.14.5 量子力学

入力

エネルギー固有値

$$\backslash [H u_n = E_n u_n \backslash]$$

$$\backslash [H | n \rangle = E_n | n \rangle \backslash]$$

Schrödinger 方程式

$$\backslash [\left[-\frac{\hbar^2}{2m} \nabla^2 + V(r) \right] u = E u \backslash]$$

完全性

$$\backslash [\sum_a | a \rangle \langle a | = 1 \backslash]$$

Klein-Gordon 方程式

$$\backslash [(\partial_\mu \partial^\mu + m^2) \phi = 0 \backslash]$$

$$\backslash [(i \partial\!\!\!/ - m) \psi = 0 \backslash]$$

出力

エネルギー固有値

$$H u_n = E_n u_n$$

$$H | n \rangle = E_n | n \rangle$$

Schrödinger 方程式

$$\left[-\frac{\hbar^2}{2m} \nabla^2 + V(r) \right] u = E u$$

完全性

$$\sum_a | a \rangle \langle a | = 1$$

Dirac 方程式

$$(i \partial\!\!\!/ - m) \psi = 0$$

Klein-Gordon 方程式

$$(\partial_\mu \partial^\mu + m^2) \phi = 0$$

3.14.6 電磁気学:マクスウェル方程式

入力

```

\newcommand{\bmD}{\mbox{\boldmath $D$}}
\newcommand{\bmE}{\mbox{\boldmath $E$}}
\newcommand{\bmH}{\mbox{\boldmath $H$}}
\newcommand{\bmB}{\mbox{\boldmath $B$}}
\newcommand{\bmr}{\mbox{\boldmath $r$}}
\newcommand{\bmi}{\mbox{\boldmath $i$}}
\def\rot{\mathop {\rm rot}\nolimits}
\def\div{\mathop {\rm div}\nolimits}
電場に関するガウスの法則\
\[ \div{\bmD}(\{\bmr\},t)=\rho(\{\bmr\},t) \]
磁場に関するガウスの法則\
\[ \div{\bmB}(\{\bmr\},t)=0 \]
電磁誘導のファラデーの法則\
\[ \rot{\bmE}(\{\bmr\},t)+
\frac{\partial {\bmB}(\{\bmr\},t)}{\partial t}=0 \]
変位電流のアンペール-マクスウェルの法則\
\[ \rot{\bmH}(\{\bmr\},t)-\frac{\partial {\bmD}(\{\bmr\},t)}{\partial t}
={\bmi}(\{\bmr\},t) \]

```

出力

電場に関するガウスの法則

$$\operatorname{div} \mathbf{D}(\mathbf{r}, t) = \rho(\mathbf{r}, t)$$

磁場に関するガウスの法則

$$\operatorname{div} \mathbf{B}(\mathbf{r}, t) = 0$$

電磁誘導のファラデーの法則

$$\operatorname{rot} \mathbf{E}(\mathbf{r}, t) + \frac{\partial \mathbf{B}(\mathbf{r}, t)}{\partial t} = 0$$

変位電流のアンペール-マクスウェルの法則

$$\operatorname{rot} \mathbf{H}(\mathbf{r}, t) - \frac{\partial \mathbf{D}(\mathbf{r}, t)}{\partial t} = \mathbf{i}(\mathbf{r}, t)$$

テクニック 其三 数式の書体

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

```

テクニック 其の四 新しくコマンドを作る

ここの例題で、ベクトルを表すのに太字体を使いました。しかし、ただ文字を太くする (`\bf` コマンド) だけでは、ちょっと感じが違います。そこで、新しいコマンドを作ってみましょう。それには、`\newcommand` というコマンドを使います。

例えば、数式中で太文字体にかえたい時には、

```
\newcommand{\bmD}{\mbox{\boldmath $D$}}
```

という新しいコマンドと作ってみました。これは、`\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。

`{\mbox{\boldmath D}}` で、新しいコマンドの内容を指定します。

`\mbox{\boldmath }` は、数式モードで、文字を太字体にするコマンドです。例題では、いくつもできてきたので、いちいち入力している手間を省くために作りました¹⁹。このようにしてやれば、新しいコマンド

```
\newcommand{\bmD}{\mbox{\boldmath $D$}}
```

ができ上がりました。

```
\newcommand{\bmE}{\mbox{\boldmath $E$}}
```

```
\newcommand{\bmH}{\mbox{\boldmath $H$}}
```

```
\newcommand{\bmB}{\mbox{\boldmath $B$}}
```

```
\newcommand{\bmr}{\mbox{\boldmath $r$}}
```

```
\newcommand{\bmi}{\mbox{\boldmath $i$}}
```

これらも同様にして作ります。

テクニック 其の五 コマンドの変更

`\def` は TeX 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。

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。ところが、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、

```
\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。(第 3.13.3 節参照)

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。

¹⁹ もちろん、こんな事をしなくても、太字体にしたいところで、設定をすればそれでも構いません。

3.15 エラーの対処

3.15.1 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になります。

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。

┌─── 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 ...

\]->\relax \ifmmode \ifinner \@badmath
                \else $$\fi ...
1.6 \end{array} \right \]

?
! Emergency stop.
<to be read again>
                \edef
\@latexerr #1#2->\edef
                \@tempc {#2}\expandafter \errhelp ...

\]->\relax \ifmmode \ifinner \@badmath
                \else $$\fi ...
1.6 \end{array} \right \]

No pages of output.
Transcript written on abc.log.
```

と、先程成功した時とは何やら違うものが表示されました。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、ひな形がおかしいという事です。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。

このようにエラーが表示されている行のところに戻って直していきましょう。

3.15.2 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。

第 4 章

AUCTEX

AUCTEX は Emacs 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、TEX 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 TEX を試してみると言うのも良いかも知れません。

但し Emacs に追加された機能を利用していますので、Emacs の操作に慣れている必要があります。

4.1 AUCTEX で L^AT_EX 生活が変わる

AUCTEX を使うとコマンド 1 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 1 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 Emacs 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 Emacs と kterm (または DECterm) を行ったり来たりする必要が無くなります。

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 1 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 TEX の英語のエラーメッセージの意味が全く理解できなかった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なる事が分かると思います。それではまず 4.2 で AUCTEX を起動してから、4.3 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。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 3 つ) です。その後、4.8 ではその他の便利なコマンドをざっと紹介します。

4.2 AUCTEX の起動

さて、AUCTEX モードを起動しましょう。本学の cc 環境では AUCTEX は cc2000 にインストールされていますので、基本的には拡張子が `.tex` のファイルを cc2000 の Emacs で読み込めば AUCTEX はドキュメ

ントのスタイルを調べて自動的に適切なモードが起動するようになっています。しかし新しいファイルを書き始める時等のためにその情報が無い場合は、defaultとして $\text{JL}\text{A}\text{T}\text{E}\text{X}$ モードが起動します。

4.3 C-c

まず覚えましょう。 $\text{AUCT}\text{E}\text{X}$ のコマンドは主にC-cで始まります。それと、新しい TEX のファイルを開いて最初に $\text{AUCT}\text{E}\text{X}$ のコマンドを入力する時にmaster fileの名前を聞いてきますが、これは分割編集の際に参照される変数です。詳しくは4.8に説明がありますが、初めて $\text{AUCT}\text{E}\text{X}$ を使う場合は単にreturnキーを押しておきましょう。

4.4 C-c C-e

任意の環境 $\backslash\text{begin}\{\}\backslash\text{end}\{\}$ を書いてくれます。eはenvironmentのeです。

Environment type: (default document)

と聞いてきます。ここで環境名を答えます。4.1ではdocumentの例を挙げましたが、その場合にはdefaultのdocumentを使います。returnキーを押せば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

label:

を聞いてきますし、table環境なら

Float to: tbp

Caption:

Label: tab:

Center: (y or n)

Pstition:

Format:

を聞いてきます。各項目については $\text{L}\text{A}\text{T}\text{E}\text{X}$ のtable環境を習得すればわかるでしょう。

4.5 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。

4.6 C-c C-c

様々なコマンドを実行します。cはもちろんcommandのcでしょう。例えば、

Command: (default jLaTeX)

の様聞いてきますので、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たければその通りにC-c C-l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調べて、**Command:** 行に入力して return を押します。また、default の値も AUCT_EX が適当に選んで表示してくれますので、空 return を押す事が 1 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。

4.7 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にトラブルが生じた際に AUCT_EX は親切にも“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。一つ注意しておくとはなく ‘です。

4.8 その他の機能

前節までの機能で十分 L_AT_EX 編集環境は向上します。ですから初めて AUCT_EX 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。

4.8.1 部分的なコンパイル

AUCT_EX 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。AUCT_EX 上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りますので、前述の手順でコンパイルします。AUCT_EX は `_region_.tex` というファイル名でコンパイルするので、コンパイル結果をプレビューするときはファイル名を `_region_.dvi` にします。

4.8.2 分割ファイルの管理

AUCT_EX 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}
\newcommand{\C}[1]{\rm C-{\tt #1}\ }
\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 AUCT_EX 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 AUCT_EX 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全体または部分的コンパイルが出来るようになります。また、子ファイルを編集集中に親ファイルを参照するときには C-c ^ と入力すれば親ファイルに移ることもできるようになります。

4.8.3 L^AT_EX マクロの補間

AUCT_EX を使うと様々な L^AT_EX 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。これで、L^AT_EX の教科書が横に無くても L^AT_EX のコマンドが使えるようになります。まずは tesh を使っている人には馴染みのある <Tab> キーによる補完機能があります。AUCT_EX では <Tab> の前に C-c が付いて C-c <Tab> になります。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 C-c <Tab> を入力します。候補が 1 つになるまでこれを続け、1 つになれば最後まで補完してくれます。次に、マクロコマンドの対話式入力法です。C-c C-m でマクロ名を聞いてきます。途中まで入力した所で先程の補完が使えます。ただしここでは <Tab> の前の C-c は必要ありません。<Tab> キーだけで補完してくれます。

4.8.4 複数行のコメントの付け外し

AUCT_EX を使うと連続した多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。

4.8.5 数式モードの支援

AUCT_EX を使うと数式モードが利用できます。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 1 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。数式モードの詳細は /NF/local/Solaris2J/src/auctex-9.1i/doc/math-ref.ps を印刷したものを参照して下さい。

4.9 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、カスタマイズ機能については説明しませんでした。それらについては /NF/local/Solaris2J/src/auctex-9.1i/doc/にあるマニュアル (auc-tex.ps, ref-card.ps, math-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。

第5章

パソコンから UNIX

UNIX コンピュータといえば大画面モニタでキーボードと三つボタンのマウスを使って、パソコンよりちょっと高価なワークステーションと呼ばれるコンピュータを想像しますね。すでに UNIX コンピュータを使っている人は 21 情報処理教室の DEC-3300 や C3 情報処理教室の NeXT Station で利用している場合が多いでしょう。しかしこれらのコンピュータは台数に限りがあるうえ、設置場所が限られていますから学内のどこにいても使えるというわけではありません。

京都産業大学には幾つかの情報処理教室に分散してさまざまなパソコンがあります¹。これらのパソコンから UNIX が利用できれば、日頃授業などで使い慣れたパソコンで利用したり、とりあえず今いるところから一番近くの情報処理教室に行って UNIX コンピュータを使えるので大変便利です。

この章ではパソコンから UNIX コンピュータを利用する例として、各情報処理教室のパソコンを通して SPARCcenter2000 (ホスト名 cc2000) を利用する手続きを説明します。

SPARCcenter2000 以外の UNIX コンピュータにも応用出来るので是非使ってみましょう。

5.1 はじめに

5.1.1 パソコンで UNIX を使うと何が嬉しいか

パソコンを通して UNIX コンピュータを使う利点は様々です。少し列挙してみると、

- 近くの情報処理教室を利用できる

UNIX でニュースや電子メールを利用している人にはとても嬉しいことだと思われま

す。授業の合間の休憩時間に近くの情報処理教室に飛び込んで、電子メールのお返事をかいたり、ニュースを読んだり。ネットワークでキャンパスライフを情報武装した京都産業大学生にはきっと嬉しいことでしょう。

- 慣れたキーボードを使える

コンピュータのキーボードには様々な種類があります。統一されているキーといたら A~Z のアルファベットキーぐらいのもので他のキー (例えば @ や & あるいは * など) は全くと言っていいほど統一されていません。少しでも慣れたキーボードを使えるのは嬉しいことです。

- 日本語入力にパソコンの FEP を利用できる

FEP とは Front End Processor の略で、日本語入力機能のことです。

¹どの部屋にどのような設備があるかについては計算機センターが配布している「パソコン利用ガイド」を参照して下さい

「UNIX ガイド」では日本語の入力に `wnn` を使うことで内容を統一しています。

日本語入力機能としての `wnn` は UNIX 上で働きます。ですから何処の情報処理教室からでも同じ操作で日本語を使っていたいという方は `wnn` を利用した方がいいでしょう。

でも、パソコンのワープロで慣れた日本語入力操作をそのまま使いたい人はパソコンの FEP を使うことが出来ます。

このガイドではそれぞれのパソコンの FEP の使い方を説明しません。しかし誤ったキー操作で日本語入力モードになってしまった方のために、FEP の終了方法をそれぞれ解説しています。

FEP の使い方を詳しく知りたい方は情報処理教室に設置してある FEP のマニュアルを一読してください。

- パソコン独自の機能を利用できる

パソコンの中にはマックや Microsoft Windows のように複数の機能を同時に利用できるようにしたものがああります。ワープロや表計算とお絵描きソフトを同時に動かして、ワープロ文書にグラフや絵を張り付けた経験を持つ方もあるでしょう。UNIX を使っているときにもこれらのことは可能です。

例えば、電子メールを送るとき、情報処理教室で最初から一気に書いてしまうのはなかなか大変です。ワープロの文書として少しずつ書いておき、メールの内容が完成してからワープロのカット & ペースト機能を使ってメールを送れば、あらかじめ文章の推敲といった作業を出来るので誤字脱字のままメールを送ってしまうといったことも最小限に出来ます。

コンピュータを使いこなす上で効果を上げる一番の方法は少しでも多くの時間コンピュータを使ってみることです。

これらのパソコンを通して UNIX コンピュータを利用する幾つかの利点は、初心者が UNIX コンピュータを使う際に直面する様々な障害を少しは緩和してくれるでしょう。

5.1.2 予備知識

ここではパソコンから UNIX コンピュータを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。また、この節で自分がどの部屋のどのパソコンから UNIX を利用するかを決めておくと良いでしょう。

パソコンから UNIX?

UNIX コンピュータを利用する場合の操作方法には、大きく分けて二通りあります。そのコンピュータの前に座って操作する方法と、目の前には座らずに遠隔操作する方法です。UNIX コンピュータはほぼ例外無く遠隔操作の可能性を持っています。

計算機センターが持っている UNIX コンピュータは幾つかあります。これらを利用するには大抵の場合そのコンピュータの前に座るのが最適です。しかし様々な事情²からそのコンピュータの前に座って使う状況を作りにくい場合があるでしょう。このような場合は自分にとって便利な部屋の設備から、遠隔操作で UNIX コンピュータを利用するという方法をお勧めします。

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りますが、リモコン³の再生ボタンを押す時もあるでしょう。

²いつも授業で使っていて空きが無い、部屋が遠い、階段を上がるのが嫌、など

³Remote Controler、リモート・コントローラの略。すなわち遠隔操作機器の事ですね。

京都産業大学の cc 環境では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「びっ」とやるだけで良いのですが、コンピュータの端末は現在のところ電線でつないでやらないといけません。この役割を果たす電線がすなわち京都産業大学のネットワーク⁴というわけです。計算機センターが管理しているパソコンは全てあらかじ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「びっ」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⁵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⁶。

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 telnet⁷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 X⁸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。

計算機センターでは幾つかのコンピュータを管理しています。京都産業大学の教員、学生が利用できるパソコンは以下のものです。

⁴このネットワークには KING（「きんぐ」と読んで下さい）という名前が付けられています。

⁵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。

⁶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。

⁷「てるねっと」と読んでください。

⁸単に「えっくす」と読んでください。

Compaq Contura	コンチュラと呼んでください。5号館2階52情報処理教室に45台設置してあるパーソナルコンピュータです。
FMR-280H	エフエムアールと呼んでください。5号館1階51情報処理教室に35台設置してあるパーソナルコンピュータです。
FMV-466D	エフエムブイと呼んでください。計算機科学研究所2階C1情報処理教室に32台設置してあるパーソナルコンピュータです。
Macintosh Centris 660AV	マックと呼んでください。3号館31情報処理教室に44台設置してあるパーソナルコンピュータです。

これらのパソコンの全てに 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。

5.1.3 これ以降の読み方

ここでは部屋ごとに用意されているパソコンによるUNIXコンピュータの使い方をそれぞれ以下のように並べて説明します。

- Mac で telnet
- FMR で telnet
- Contura で telnet
- FMV で telnet
- FMV で X

これらの節はパソコンを利用する為の基礎知識を前提に書かれています。つまりあなたは使いたいと思っている部屋のパソコンの基本的な操作方法を知っている必要があります。

もしも各情報処理教室のパソコンの基本的な使い方がわからない場合は、まず計算機センターが発行し無料で配布している「パソコン利用ガイド」を参照して、そこから慣れて下さい。「パソコン利用ガイド」にはそれぞれの部屋のパソコンの電源の入れ方と切り方、また基本的な操作方法について参照するべきマニュアルの部分などについて詳しく書かれています。

また、これらの節はUNIXコンピュータを利用する為の基礎知識も前提に書かれています。つまりあなたはUNIXコンピュータを利用する際に必要なユーザ名、パスワードなどの概念を理解していなければなりません。ユーザ登録も済ませている必要がありますし、そのような人はきっと計算機センターが発行し無料で配布している「UNIXガイド」を持っているはずで、そしてこの「UNIXガイド」の14ページの2.2までを実行している、ないしは読んでいることでしょう。

もしもこれらの条件を満たしていない場合は、まず計算機センターが主催している「UNIXオリエンテーション」に参加して下さい。そこではユーザ名やパスワードなどといったものが何を意味しているかなどを説明しており、ユーザ登録の手続き方法も教えてくれます。手続きが済んだ人には「UNIXガイド」を配布しています。

そしてこれらの条件を満たしている人に向けて、各部屋ごとに以下のような順にパソコンからUNIXコンピュータを利用する手続きを説明します。

1. 端末ソフトを起動する
2. UNIX コンピュータに接続する
3. login する
4. (その必要があれば) パスワードを変える
5. logout する
6. 端末ソフトを終了する

すなわち説明はパソコンの電源を入れた後から始まり、UNIX コンピュータのコマンドが利用できるところまでと、UNIX コンピュータを使い終わった後から電源を切る前までを対象に行なわれます。パソコンの電源を入れたり切ったりする手続きについては先述の「パソコン利用ガイド」を参照して下さい。UNIX コンピュータの使い方についてはやはり先述の「UNIX ガイド」の第3章、「UNIX それから」以降を参照して下さい。

5.2 FMV で telnet

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⁹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。

5.2.1 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 5.1 が現れます。

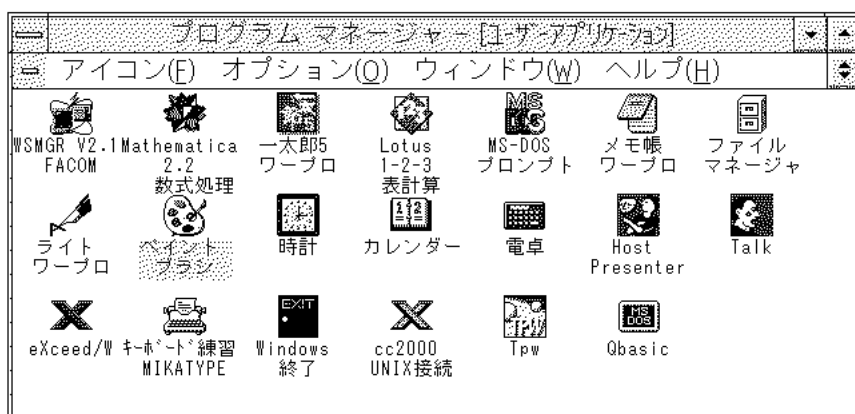


図 5.1: Windows 初期メニュー

⁹ キーボード真中やや右の大きなキー

図 5.1 の中の一つに「Host Presenter」図 5.2 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。

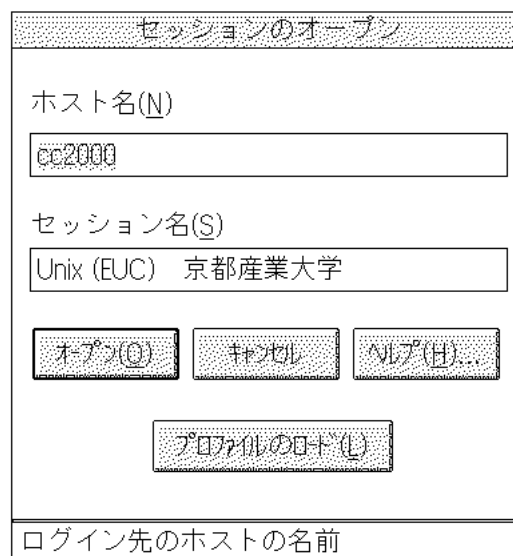


図 5.2: Host Presenter のアイコン

5.2.2 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 5.3 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 5.2 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 5.3 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。



セッションのオープン		
ホスト名(N)	<input type="text" value="cc2000"/>	
セッション名(S)	<input type="text" value="Unix (EUC) 京都産業大学"/>	
<input type="button" value="オープン(O)"/>	<input type="button" value="キャンセル"/>	<input type="button" value="ヘルプ(H)..."/>
<input type="button" value="プロファイルの読み込み(L)"/>		
ログイン先のホストの名前		

図 5.3: Host Presenter の初期画面

画面上には、図 5.3 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 5.4 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ソフト機能+(I) 設定(S)... ^L^P
UNIX(r) System V Release 4.0 (cc2000)
login: _
```

図 5.4: 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。

5.2.3 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ソフト機能+(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 5.5: cc2000 のログインの SAMPLE

図 5.5 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。

5.2.4 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。

```
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 5.6: 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。

5.2.5 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、あまり安全とはいえません。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「UNIX ガイド」の 10 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` コマンドを利用します。詳細は「UNIX ガイド」の 19 ページの 2.3.7 を参照してください。

5.2.6 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
```

5.2.7 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 5.3 が表示されます。その一部分を拡大したのが、以下の図 5.7 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。

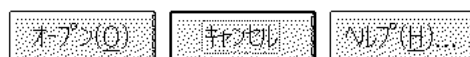


図 5.7: Host Presenter の終了ボタン

図 5.3 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」の位置を重ね合わせ、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 5.4 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 5.8 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」を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 5.1 に戻ります。



図 5.8: Host Presenter の終了

FMV から telnet 端末環境でホストコンピュータ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ください。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。FMV の電源の切り方に関しては別冊の「パソコン利用ガイド」を参照して下さい。

5.2.8 困った時の対処

ホストコンピュータに接続出来ない

- 図 5.3が表示されていない場合。図 5.2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 5.2 をダブルクリックする。
- 図 5.3は表示されているが、図 5.1 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 5.9 が画面に表示されて図 5.4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。

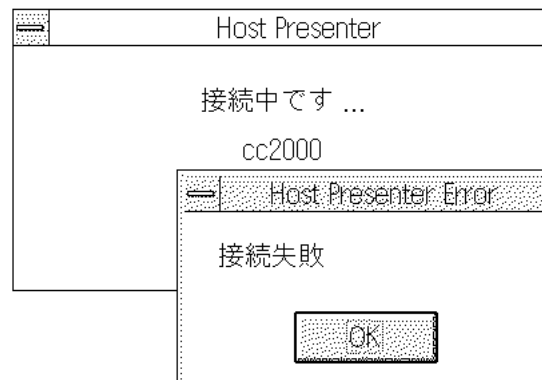


図 5.9: 接続失敗の画面

- 図 5.10 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。

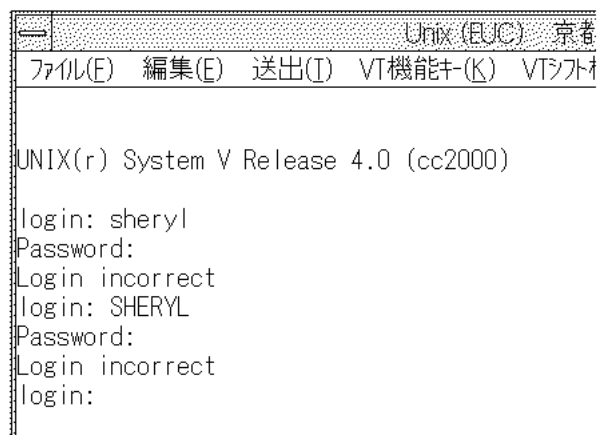


図 5.10: login 失敗の画面

文字が全角になる

- 図 5.11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。

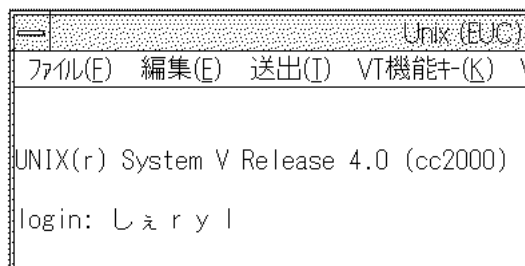


図 5.11: 文字が全角になる

5.3 マックで telnet

ここでは3号館2階の31情報処理教室のマックから telnet 環境で UNIX 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 UNIX コンピュータは cc2000 を例として示しています。

31 情報処理教室のマックでは端末ソフトとして「ASLTelnet」を利用します。これは telnet 環境を提供します。

5.3.1 端末ソフトの起動

まず端末ソフトを起動しましょう。

端末ソフト ASLTelnet の起動には二通りの方法があります。一つの方法はアップルメニューから起動する方法で、もう一つはマックのソフトの標準的な起動方法であるソフトウェアのアイコンを直接指定する方法です。まず、初心者にわかりやすいようにアップルメニューからの起動を先に説明します。もう一つの方法は少し複雑なので、後ほど「端末ソフトを起動する (その2)」で説明します。

マックのマウスを使ってアップルメニューから cc2000 を選びます。アップルメニューの cc2000 は ASLTelnet の環境設定がされたものです。

うまく起動できると、telnet ウィンドウが一つ開いて、

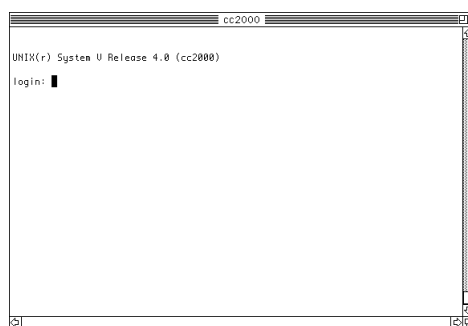


図 5.12: ASLTelnet の login 画面

と login の催促文字が出力されます。

5.3.2 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。

正しくユーザ名をキーインし<Return>キーを押すと、パスワードを聞いてくるのでこれも正しく入力し<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えて入力していませんか。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。

もう一度注意してやってみましょう。きつとうまく login 出来るはずです。

5.3.3 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 5.13: 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。

5.3.4 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、あまり安全とはいえません。是非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「UNIX ガイド」の 10 ページの 2.1.3 で既に述べたことに注意してください。

パスワードを変更するには yppasswd コマンドを利用します。詳細は「UNIX ガイド」の 19 ページの 2.3.7 を参照してください。

5.3.5 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
```

5.3.6 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 UNIX 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。

5.3.7 端末ソフトを起動する（その 2）

これまではアップルメニューから端末ソフトを起動する方法を説明しました。実はこの方法が初心者には一番トラブルに遭遇しないで cc2000 を使う手段だとだと考えられます。cc2000 を使う限りこれまでの

方法が一番いいので、面倒なことは聞きたくないという方はこの節を無視してください。

さらにこの節を読み進むかたのために cc2000 以外の UNIX コンピュータとして csosf01 に telnet 接続する方法を説明します。

端末ソフトを起動する二番目の方法は ASLTelnet のアイコンをマウスカーソルでダブルクリックして起動します。

ASLTelnet のアイコンは students フォルダの中の Networking フォルダの中にあります。

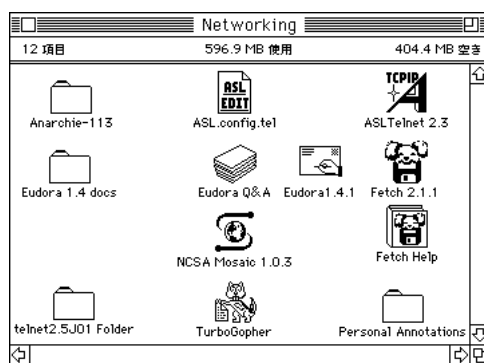


図 5.14: ASLTelnet のアイコン

このアイコンにマウスカーソルを重ねダブルクリックするとソフトウェアが起動しはじめます。メニューバーの右端に小さなアイコンが表示されたら ASLTelnet が起動された状態です。

この状態で「File メニュー」から「Open Connection...」を選択すると UNIX コンピュータを選択するウィンドウが表示されます。

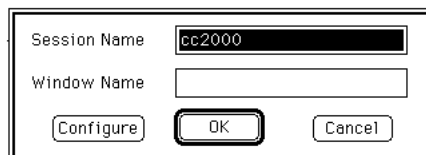


図 5.15: ホスト選択ウィンドウ：接続するホスト名を指定します

Session Name に目的の UNIX コンピュータのホスト名として csosf01 を入力し、OK を選択します。

OK を選択するとすぐに UNIX コンピュータの login ウィンドウが開かれます。もしウィンドウ開かれなかった場合、Session Name のホスト名を間違っていないかどうか確認しましょう。

ホスト名については「UNIX ガイド」の「第 2 章 UNIX はいかが」の「2.1.1 計算機センター管理の UNIX 環境」に掲載されていますのでそちらを参考にしてください。

Window Name は telnet ウィンドウを開いたときにメニューバーに書いたものを表示してくれます。ここに何も入れない場合は自動的に Session Name と同じ名称が適用されますので、使い勝手のいいように使い分けるといいでしょう。

5.3.8 少し高度な使い方

パソコンのカット & コピーを使う

最初にすこし触れましたが、授業のレポートなど少し大きな文章を電子メールで送るような仕事をしなければいけないことがあります。文章を入力するだけでも長い時間かかるうえ、誤字脱字を修正し、文章として推敲し、送信するというのは、一気にやっつけてしまおうとすれば大変ですね。

レポートの内容を作り上げる仕事とそれを電子メールとして送る仕事を別に行えれば嬉しいです。

パソコンのワープロを使ってレポートの入力から推敲までをやっつけてしまいます。これには空いている時間を充分に使って、何日もかけて納得がいくまで仕上げるのがいいでしょう。もし君が自宅や下宿にパソコンを持っていれば、この仕事はそちらでやってもいいでしょう。文章を入れるフロッピーを持ち歩いていれば、時間さえあれば情報処理教室に飛び込んで続きを仕上げるなんてことも可能ですね。

さて、そのようにしてレポートが完成したら、今度はそれを電子メールに張り付けて送信すればいいのです。それにはワープロと UNIX ウィンドウを同時に開いて、それぞれのウィンドウが重ならないように適当に配置して、ワープロの文章を電子メールの文章へと流し込む作業をします。

簡単に説明すると、まずワープロの文章をマウスで選択します。次に「Edit (編集) メニュー」から「Copy」を選択します。これでワープロの文章はマックのクリップボードというところに複製されました。

今度は電子メールのウィンドウに移動 (マウスで電子メールウィンドウを一回クリック) して、電子メールウィンドウの文章を張り付ける位置にカーソルを持っていき、Edit メニューの「Paste」を選択すると、クリップボードに複製されていた文章が電子メールに流し込まれます。電子メールのウィンドウの文章が流し込まれる位置は、カーソルがある位置ですから注意してください。

これらの操作は「Copy」は `command-c` で「Paste」は `command-v` のショートカットキーが用意されているので、慣れるとこちらの方が便利です。

最後に電子メールの文章を少し整えて、送信します。

複数のウィンドウを開く

ところでマックの ASLTelnet は telnet ウィンドウを複数開くことができます。

cc2000 に telnet ウィンドウを 2 枚開いて、csosf01 のウィンドウを 1 枚、そして VMS コンピュータである ksuvx1 のウィンドウを 1 枚といったことが普通に出来てしまうわけです。これらの telnet ウィンドウの間やワープロなどの文書との間でカット & ペーストといった操作も可能で、うまく使えばかなり便利なものです。

さてすでに cc2000 の telnet ウィンドウが一枚開いている状態で、二枚目以降のウィンドウを開く方法を二つ説明します。

まず一つの方法は、次に開く UNIX コンピュータが SPARCcenter2000 なら、最初のウィンドウと同じようにアップルメニューから cc2000 を選ぶといいのです。一番目のウィンドウと同じ位置に新しいウィンドウが現れますね。

一番目のウィンドウと重なって、いやだという人はマウスで新しいウィンドウのタイトルバーをつかんで適当な位置にドラッグして使いましょう。

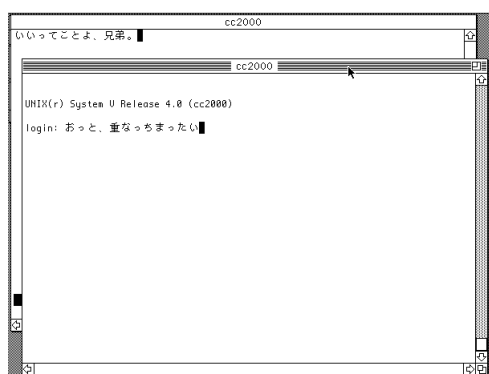


図 5.16: 二枚目の画面を開いて、重ならないように移動する

異なる UNIX コンピュータのウィンドウを開く

先ほどの cc2000 を選択する方法では開かれるウィンドウは SPARCcenter2000 のウィンドウです。他の UNIX コンピュータ、例えば DEC-3000/300 UNIX ワークステーション、つまり csosf01 の telnet ウィンドウを開くときにはどうすればいいのでしょうか。

ASLTelnet の「File メニュー」に「Open Connection (ショートカットキーはコマンド O : コマンドキーと英字の O キーを同時に押す)」があります。これを選択すると、次のウィンドウが表示されます。



図 5.17: ホスト選択ウィンドウ : csosf01 のウィンドウを開いてみましょう

Session Name に目的の UNIX コンピュータのホスト名、この場合は csosf01 を入れて、OK を選択します。OK を選択するとすぐに UNIX コンピュータの login ウィンドウが開かれます。もしウィンドウ開かれなかった場合、Session Name のホスト名を間違えて指定していないかどうか確認しましょう。

ホスト名については「UNIX ガイド」の「第 2 章 UNIX はいかが」の「2.1.1 計算機センター管理の UNIX 環境」に掲載されていますのでそちらを参考にしてください。

Window Name は telnet ウィンドウを開いたときにメニューバーに書いたものを表示してくれます。ここに何も入れない場合は自動的に Session Name と同じ名称が適用されますので、使い勝手のいいように使い分けるといいでしょう。

キーボードの最適化

アップルメニューの cc2000 は先ほども述べたように、cc2000 コンピュータ（つまり SPARCcenter2000）に telnet login 出来るように最適化された機能をサービスしています。このまま使っていく分には大きな問題は生じないでしょう（もっとも emacs を動かしたときのモードライン（反転文字）が少し右にずれています）。

ちょっと注意して欲しいのは、アップルメニューから cc2000 を選択して一番目の画面では emacs など問題なく動いてくれますね。ところが二番目の画面では C-c や C-s や C-q が入力できません。これはまだ二番目のウィンドウが UNIX コンピュータのために最適化されていないためで、以下の方法でしてください。

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。

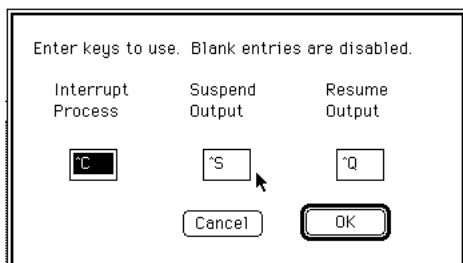


図 5.18: キーのセットアップ

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。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。

5.3.9 困った時の対処

文字を入力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。

入力すると日本語が出てしまう

最初に説明した FEP が日本語モードになっていることが考えられます。

日本語モードを止めるには command キーと Space バーを同時に押します。メニューバーの右から二つ目の小さなアイコンが◆になります。これが英文字入力モードです。

telnet 画面が表示されない

最初にアップルメニューから cc2000 を選択し ASLTelnet を起動したときに telnet 画面が現れなかったとしたら、おそらくあなたが手順を間違えたのではないと思われます。その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 UNIX 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。

もし ASLTelnet をアップルメニューの cc2000 以外の方法で起動したり、ホスト選択ウィンドウから接続ホスト名を指定して telnet ウィンドウを開けようとしたのであれば、ホスト名を間違えている可能性があります。

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 cc2000 で起動しているときは起こらない現象です。

マックの ASLTelnet が UNIX に送る日本語コードと UNIX から送られてくる日本語コードがうまくあっていないときにこの様な状況になります。

「Session メニュー」に「Transmit Code」と「Receive Code」があります。これをそれぞれ「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。

5.4 FMR で telnet

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。

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¹⁰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。

5.4.1 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5.19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。

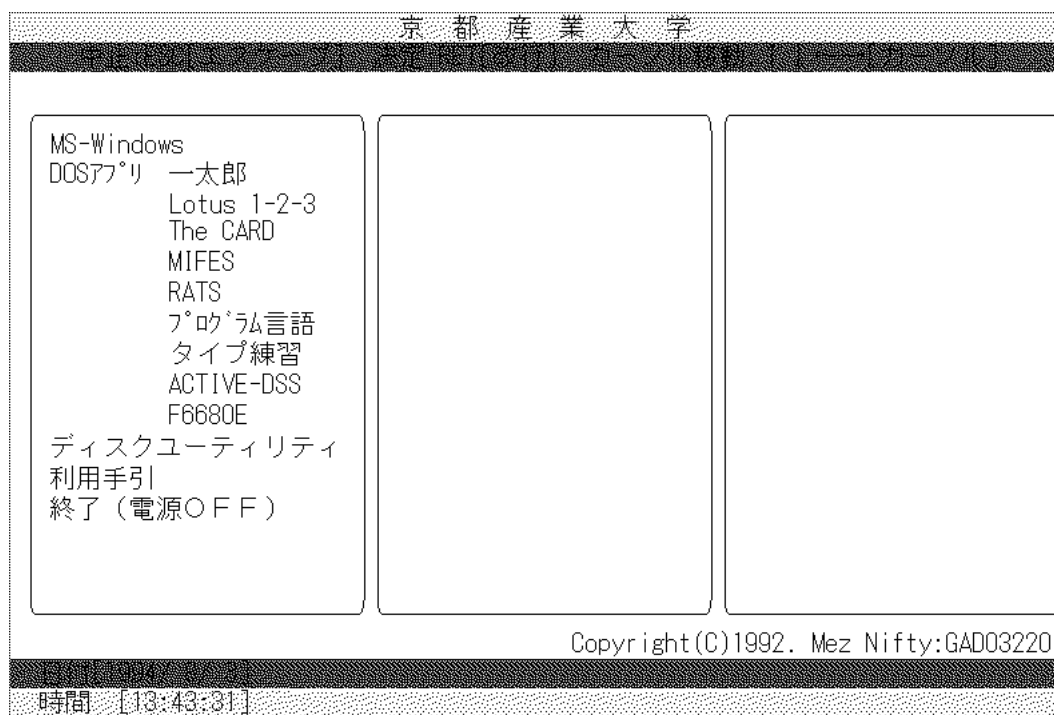


図 5.19: FMR の初期メニュー

¹⁰ キーボード真中やや右の大きなキー

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、図 5.20 が表示されます。

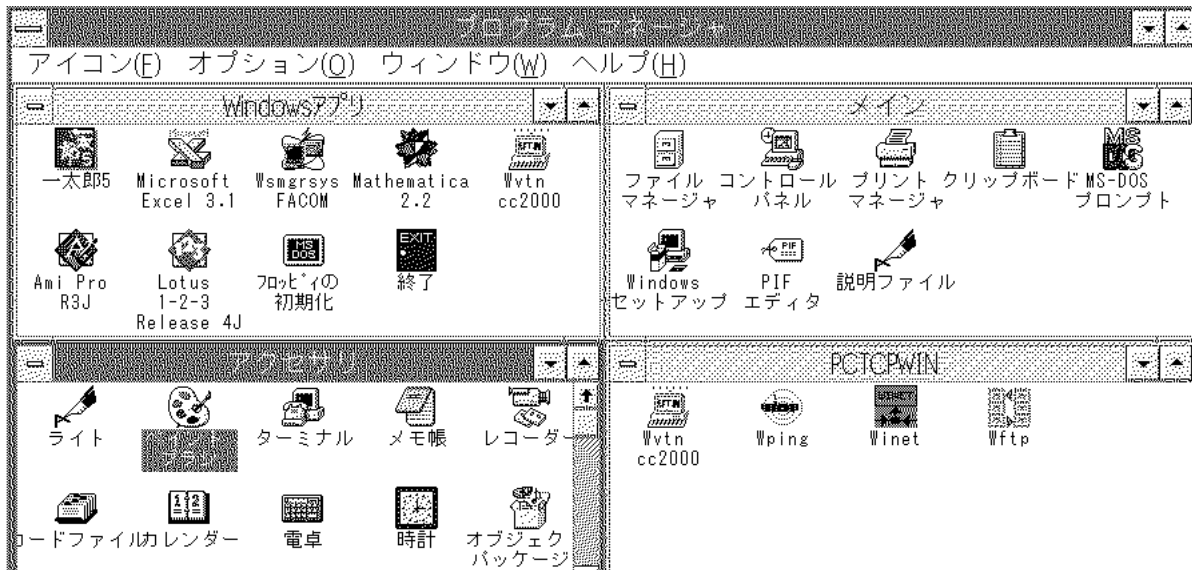


図 5.20: Windows 初期メニュー

図 5.20 の中の一つに「WVTN cc2000」図 5.21 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 5.21: WVTN のアイコン

5.4.2 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。もし、貴方が cc2000 とは違うホストコンピュータに接続したい場合は、この後の 5.4.8をお読みください。

WVTN が起動されると、図 5.22 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 5.21 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 5.22 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。

画面上には、図 5.22 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。

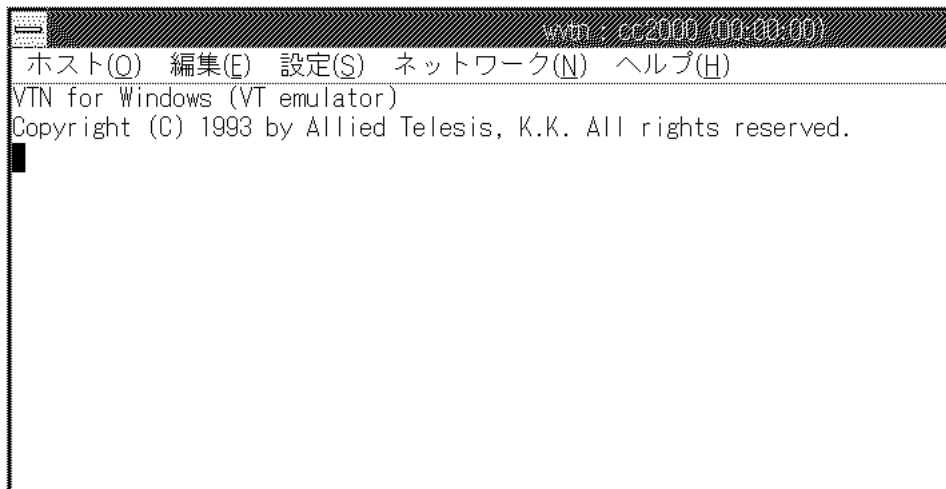


図 5.22: WVTN の初期画面

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 5.23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。

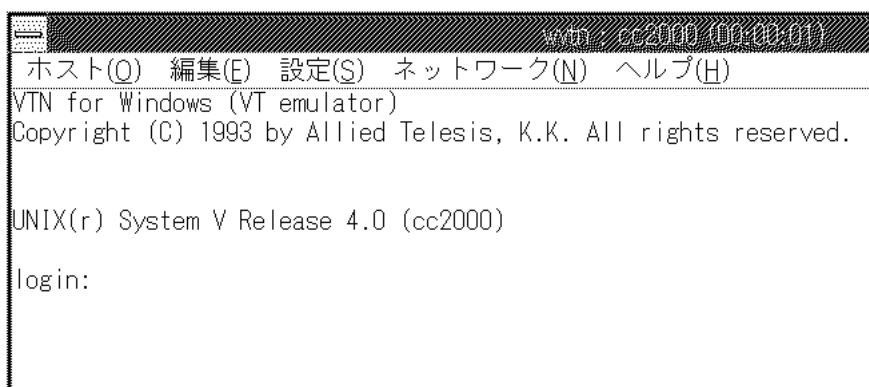


図 5.23: cc2000 のプロンプト

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。

5.4.3 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。

```
cc2000 (00:00:03)
ホスト(Q) 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 5.24: cc2000 のログインの SAMPLE

図 5.24 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。

5.4.4 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。

```
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 5.25: 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。

5.4.5 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、あまり安全とはいえません。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「UNIX ガイド」の 10 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` コマンドを利用します。詳細は「UNIX ガイド」の 19 ページの 2.3.7 を参照してください。

5.4.6 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
```

5.4.7 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 5.22 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 5.20) に戻ります。



図 5.26: WVTN の終了

図 5.22 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。

FMR から `telnet` 端末環境でホストコンピュータ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ください。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。FMR の電源の切り方に関しては別冊の「パソコン利用ガイド」を参照して下さい。

5.4.8 cc2000 以外のホストコンピュータへの接続

ここは `cc2000` 以外のホストコンピュータに接続する方を対象としています。 `cc2000` を利用される方は、読まなくて結構です。

画面上には、図 5.22 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「ホストの登録」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 5.27 のようになります。図 5.27 の左上部に「`cc2000`」が反転表示されています。ここを貴方が接続したいホストコンピュータの名前に書き換え、図 5.27 の右側にある「接続」ボタン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。貴方が接続したいホストコンピュータの **login:** が表示されます。login: が表示されているか確認して下さい。もし、画面上に現れない場合は、「困った時の対処」を参照してください。



図 5.27: 接続先ホストの変更

5.4.9 困った時の対処

ホストコンピュータに接続出来ない

- 図 5.22 が表示されていない場合。図 5.21 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 5.21 をダブルクリックする。
- 図 5.22 が表示されないまたは見えない場合。図 5.20 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 5.28が画面に表示され、その後図 5.22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 5.19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 5.28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。

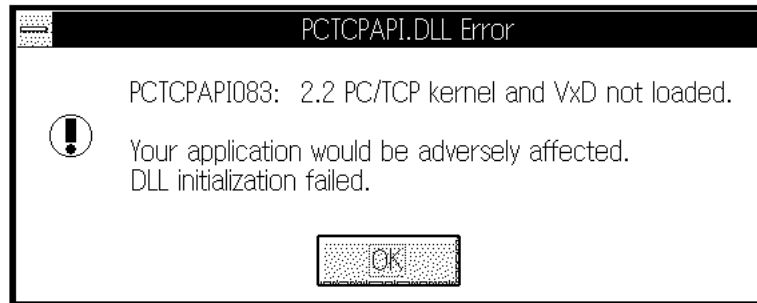


図 5.28: 端末ソフトの起動失敗画面

- 図 5.29が表示され、ホストコンピュータに接続できない場合。ホストコンピュータの名前（ホスト名）を間違えている場合または初期設定を変更されている場合が考えられる。5.4.8を読んで、ホストコンピュータの名前を正しく入力してください。



図 5.29: 接続失敗の画面

login ができない

- 図 5.30 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 5.30: login 失敗の画面

文字が全角になる

- 図 5.31 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、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: しえ r y l
Password: █
```

図 5.31: 文字が全角になる

5.5 Contura で telnet

ここでは5号館2階の52情報処理教室のConturaからtelnet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。

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Conturaではこの「Enter」キーを押して下さい。

5.5.1 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5.32が現れます。

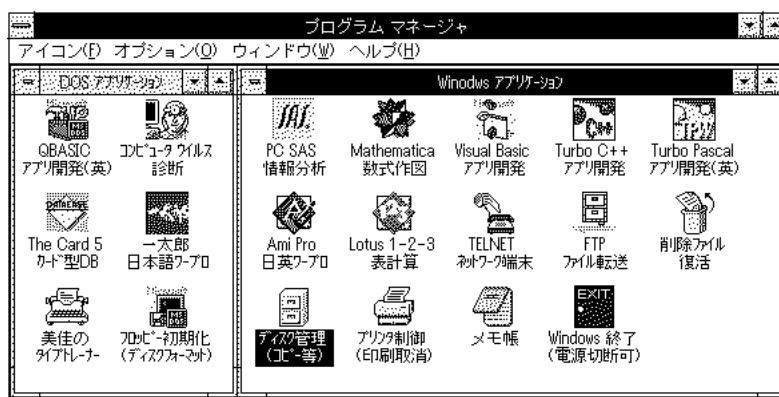


図 5.32: Windows 初期メニュー

図5.32の中の一つに「TELNET ネットワーク端末」図5.33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 5.33: Telnet のアイコン

¹¹ キーボード真中やや右の大きなキー

5.5.2 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 5.34 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 5.33 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 5.34 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 5.34: 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。

5.5.3 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 5.35: cc2000 のログインの SAMPLE

図 5.35 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。

5.5.4 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。

```
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 5.36: 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。

5.5.5 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、あまり安全とはいえません。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「UNIX ガイド」の 10 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` コマンドを利用します。詳細は「UNIX ガイド」の 19 ページの 2.3.7 を参照してください。

5.5.6 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
```


5.5.7 端末ソフトを終了する

logout すると図 5.34 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 5.37 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 5.32）に戻ります。

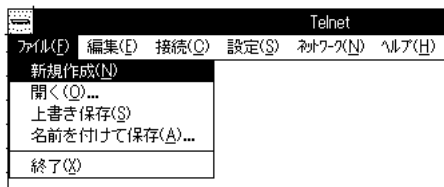


図 5.37: Telnet の終了

Contura から telnet 端末環境でホストコンピュータ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ください。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。Contura の電源の切り方に関しては別冊の「パソコン利用ガイド」を参照して下さい。

5.5.8 困った時の対処

ホストコンピュータに接続出来ない

- 図 5.34 が表示されていない場合。図 5.33 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 5.33 をダブルクリックする。
- 図 5.34 は表示されているが、図 5.32 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 5.38 が画面に表示されて図 5.34 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 5.38: 接続失敗の画面

login ができない

- 図 5.39 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。

```
Telnet - cc2000
ファイル(F) 編集(E) 切断(D) 設定(S) ネットワーク(N) ヘルプ(H)
UNIX(r) System V Release 4.0 (cc2000)
login: sheryl
Password:
Login incorrect
login: SHERYL
Password:
Login incorrect
login: █
```

図 5.39: login 失敗の画面

文字が全角になる

- 図 5.40 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。

```
Telnet - cc2000
ファイル(F) 編集(E) 切断(D) 設定(S) ネットワーク(N) ヘルプ(H)
UNIX(r) System V Release 4.0 (cc2000)
login: しえ r y l █
```

図 5.40: 文字が全角になる

5.6 FMV で X

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¹²。

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。

5.6.1 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 5.41 が現れます。

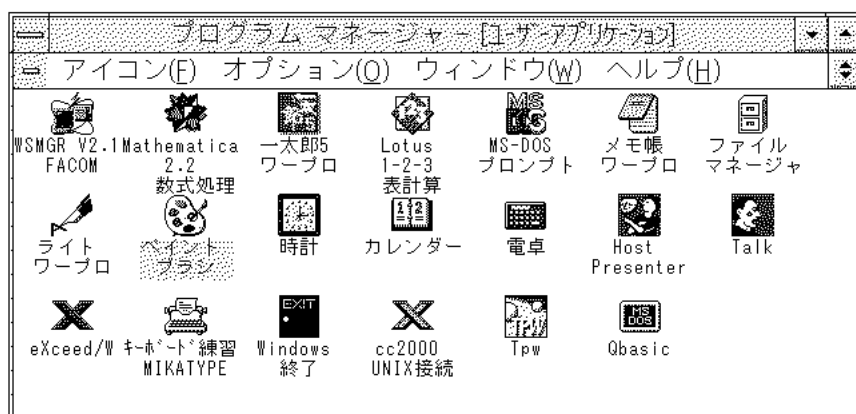


図 5.41: Windows 初期メニュー

¹² 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。

¹³ キーボード真中やや右の大きなキー

図 5.41 の中の一つに「cc2000 UNIX 接続」図 5.42 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。これと間違わないように注意して下さい。



図 5.42: eXceed のアイコン

上記の手続きで eXceed が起動されると、図 5.43 が画面に表示されます。もし画面に現れなかったら、あなたは端末ソフトの起動に失敗しています。もう一度図 5.42 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 5.43 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。

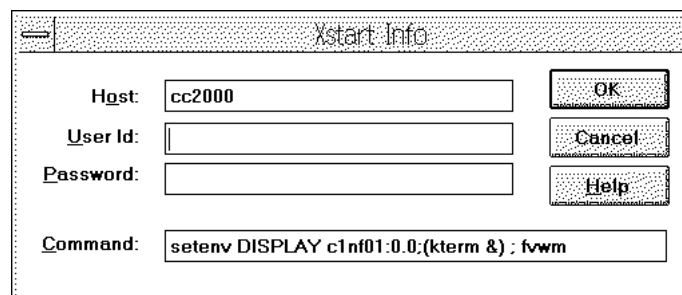


図 5.43: eXceed の初期画面

画面上には、図 5.43 が表示されており、「Host:」の右にホスト名として「cc2000」があらかじめ指定されているのが判りますね。

5.6.2 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。「User Id:」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、<Tab>キー¹⁴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、タイプした文字とは関係無く「*」記号が画面に表示されます。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。

画面一杯に灰色の背景が広がり、図 5.44 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。

¹⁴ キーボード左端の下から 4 つめにある「Tab」と左右の矢印が刻印されたキーです。

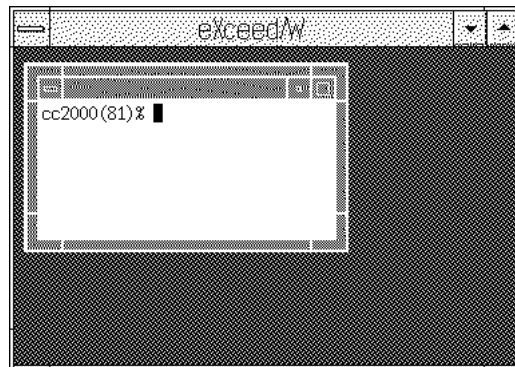


図 5.44: cc2000 の X 端末からのログインの SAMPLE

5.6.3 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。

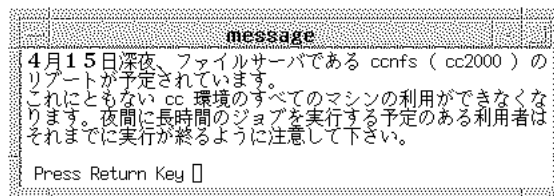


図 5.45: login 直後のメッセージ

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。

5.6.4 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、あまり安全とはいえません。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「UNIX ガイド」の 10 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` コマンドを利用します。詳細は「UNIX ガイド」の 19 ページの 2.3.7 を参照してください。

5.6.5 logout

次は logout です。この X 環境では logout 操作は特に必要無く、ただ eXceed を終了させれば良いのです。

5.6.6 端末ソフトを終了する

画面の一番左上端の小さな四角に注目してください。図 5.46 です。似たような形の四角は画面上に幾つかあると思いますので間違わないように注意して下さい。画面の最も上を見ると水色の「eXceed/W」と書かれた横棒があります。この棒の左端にある小さな四角です。

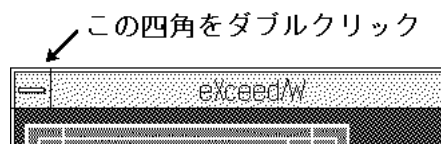


図 5.46: eXceed の終了

この小さな四角にマウスポインタを重ね合わせ、マウスの左ボタンをダブルクリックしてください。すると図 5.46 を表示して、本当に終了して良いかどうか確認してきます。



図 5.47: eXceed 終了時の確認問い合わせ

ここでリターンキーを押せば¹⁵電源を入れた最初の状態に戻ります。

FMV から X 端末環境でホストコンピュータ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ください。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。FMV の電源の切り方に関しては別冊の「パソコン利用ガイド」を参照して下さい。

5.6.7 困った時の対処

端末ソフトが起動しない

- 図 5.42 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 5.42 をダブルクリックする。
- 表示はされているが、図 5.41 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下さい。

¹⁵ もちろん「OK」ボタンをクリックしても構いません

login ができない

- 図 5.48が表示される。

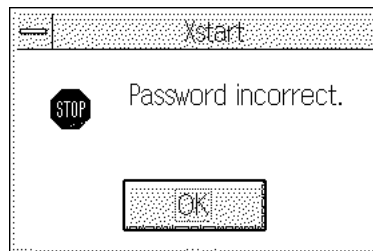


図 5.48: login 失敗の画面

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 5.6.1からやりなおしです。しかしこの段階では login に失敗したにもかかわらず、端末ソフトが起動されたままの状態になっています。まず 5.6.6に従い、端末ソフトを終了し、それから 5.6.1へと戻って下さい。

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。

第 6 章

MODEM から UNIX

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。第 5 章で説明した、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。

6.1 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	***_****.****
	MD144FB5V	4	***_****.****
	MD144XT10V	5	***_****.****
Annex	DMP	7	***_****.****

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。

6.2 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。

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 などを計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、計算機センターの相談窓口までお気軽にご相談ください。

6.3 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
```

6.3.1 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。

6.4 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7章

NeXT を上手に使おう

7.1 NeXT ファイル管理術

7.1.1 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¹」に詳しく書かれていますので、そちらを参照して下さい。

¹C3 情報処理教室の書架に並んでいます。

7.1.2 ファイルビューアの各部名称

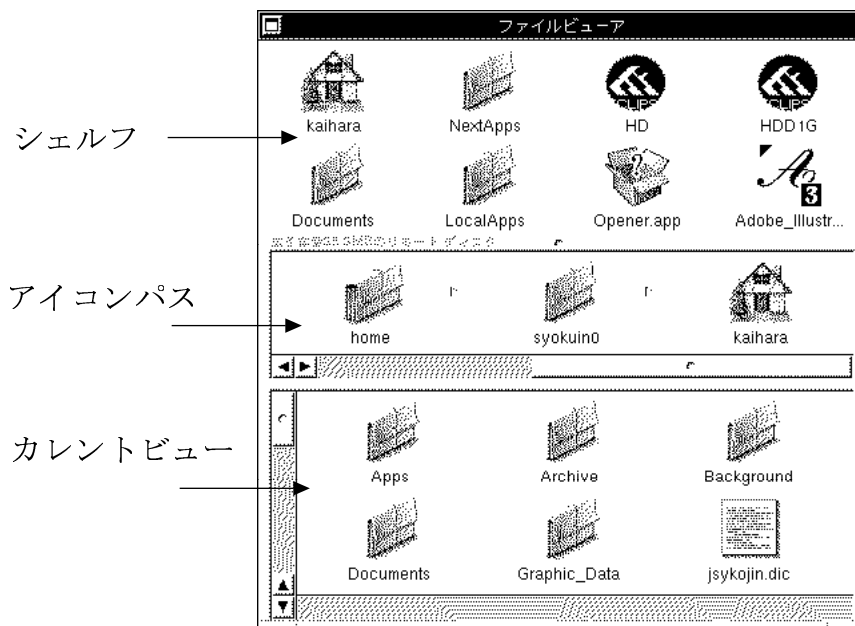


図 7.1: ファイルビューアの各部名称

7.1.3 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²。

7.1.4 フォルダで整理

あなたは、自分のホームフォルダ³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⁴。

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 での話をするときには、フォルダで統一しようと思います。

²詳しくは「UNIX ガイド」の「3.4 ファイル」を参照してください。

³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。

⁴詳しくは「UNIX ガイド」の「3.8 ファイルの階層構造」を参照してください。


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

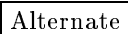
新しいフォルダを作成します。

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⁵。フォルダ名に漢字⁶を使うこともできますが、そのフォルダを cc 環境の他のコンピュータからは見ることができない場合があります。フォルダ名に漢字を使うのはやめた方がよいでしょう。

7.1.5 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. 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7.1.4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ういでしょう。

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。

⁵ファイル名についても同様です

⁶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。

7.1.6 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7.1.4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ういでしょう。


7.1.7 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。

7.1.8 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  (緑の矢印の形のカーソル) が表示されます。

7.1.9 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 7.1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。

注意

以下のような場合には、ファイルやフォルダの名称変更を行ってはいけません。

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⁷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。

7.1.10 削除

リサイクル

NeXT でファイルまたはフォルダを削除する場合には、リサイクルを利用します。リサイクルは画面右

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ルにドラッグします。

注意 このままでは、ドラッグしたファイルやフォルダは、リサイクル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。

リサイクル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ルを空にして、リサイクル内のファイルやフォルダを抹消してしまいましょう。

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。

7.1.11 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。

リンクを利用すると大幅にファイルを指定する手間を省けます。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。

⁷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、ファイル内の情報のタイプを表します。

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7.1.4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ラを使います。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。

7.1.12 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「UNIX ガイド」の「5.2 ファイルのアクセス権」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なければチェックマークに、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。

7.1.13 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。

7.1.14 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 7.1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を登録していくと、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もよいのですが、それでもあまり多くのファイルやフォルダを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。

7.1.15 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できます。

- NeXT < 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、7.1.15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。

7.1.16 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 (教室の入口を入っ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。

7.1.17 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。

7.2 NeXT で使うネットワークサービス

7.2.1 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」⁸の「ネットワークの世界へようこそ」の以下の節を読む⁹。
 - (a) 4.1 ネットワークサービス紹介
 - (b) 4.2 電子メール準備体操
 - (c) 4.5.13 記事を書く場合の注意
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¹⁰。
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、同じく「UNIX ガイド」の「NeXTStation を使ってみましょう」の章に従って、NeXT の雰囲気（作法？）に馴染む。
4. この章「NeXT で使うネットワークサービス」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¹¹。


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。

7.2.2 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、メールに関する C 3 情報処理教室設置されている NeXT 独自の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¹²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、そのままお待ち下さい。

起動すると図 7.2 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¹³、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。

⁸ 持っていない人は計算機センターの窓口に来て下さい。

⁹ 量としてはほんの少しですからさらっと読めるでしょう。

¹⁰ 秘訣は強く念ずることです。

¹¹ 仲間は多ければ多い方が良いのは何をしても同じです。

¹² 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。

¹³ もしかすると友達が先にあなた宛に送ったメールが届いているかもしれませんね。

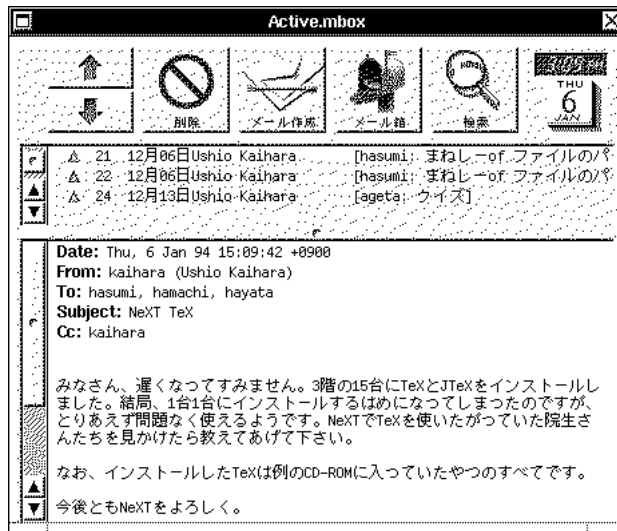




図 7.2: メール箱ウィンドウ

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 7.3参照）。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」¹⁴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。

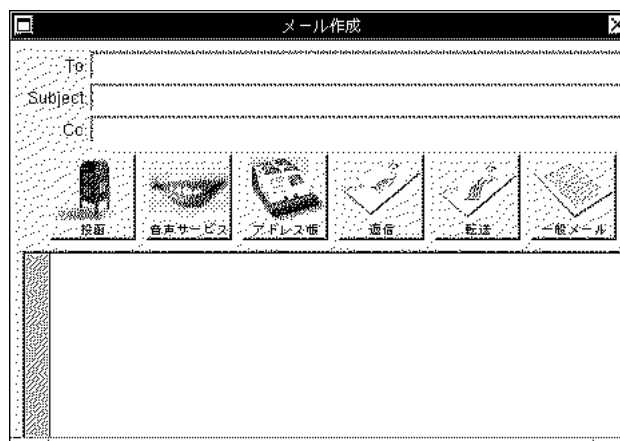


図 7.3: メール作成ウィンドウ

¹⁴ 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 マシン群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 マシン (SPARCcenter2000 や DEC-3300) 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んでみたいと思います。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 7.4) 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 7.2.2 を読んでトライして下さい。

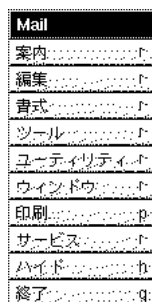




図 7.4: Mail メニュー

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下さい。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、試してみたくありませんか？

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下さい。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 7.2.2 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 7.2.2 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:」欄に注目して下さい。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:」¹⁵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 7.2.2 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

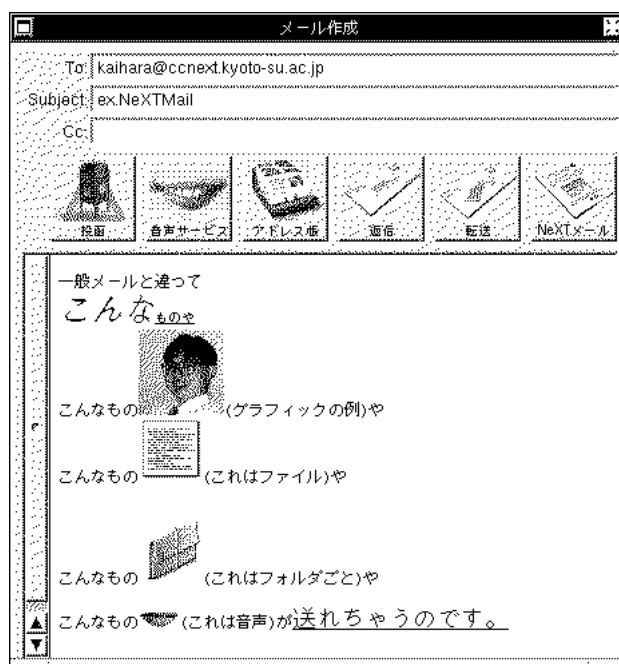



図 7.5: NeXT メール の例


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、

¹⁵ 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 1 3 章「メールの送受信」と第 1 4 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下さい。

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、「UNIX ガイド」の「4.4 メールを書くときの注意」で紹介していますので必ず読んでおいて下さい。

7.2.3 ニュースしよう

この章ではニュースソフトウェア「NewsBase」¹⁶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。

初めての NewsBase

最初の使用の際には若干自分で設定しなければならないことがありますが、これから書いてあることをちゃんと読めばまったく難しくありませんので、心配しないでついてきて下さい。

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（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）→ LocalApps フォルダへ階層を下りていきます。（図 7.6 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、購読記録を書き込むファイルを作りましょう。このファイルに今後自分がどの記事を既に読んでいるかやどのニュースグループを予約購読しているかなどが書き込まれることとなります。さきほどのフォルダに「MakingNewsFile」と書かれたアイコンがあるのですが、このアイコンをダブルクリックして下さい。ウィンドウが開いて何かメッセージを表示しますが気にしないで、ウィンドウのタイトルのところに「終了しました」というメッセージが出れば、購読記録ファイルの作成は完了です。

この作業は全くの最初の 1 回だけ実行すれば結構です。（次の日も次の年でもとにかく）2 回目以降ならば実行する必要はありません。

¹⁶ (株) リクルートのフリーソフトウェアです。

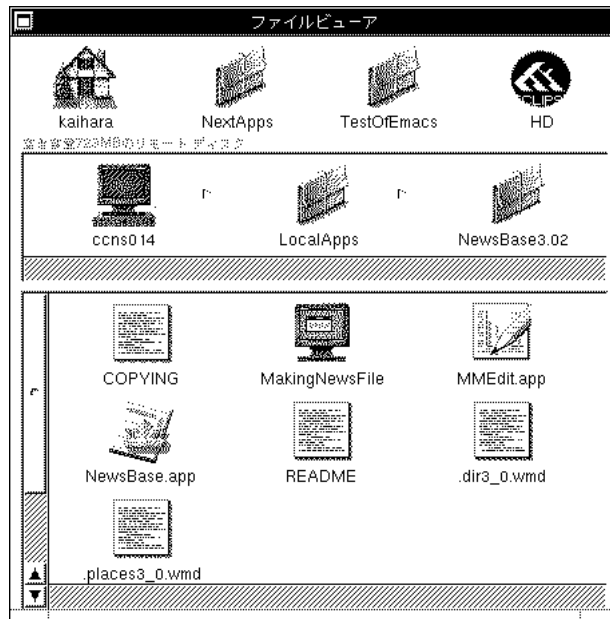


図 7.6: NewsBase の置き場所

そして次に「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。起動すると最初に「nntp サーバマシン名¹⁷」を入力するパネルが現れます(図 7.7を参照)。最初は「localhost」という言葉が入っていますが、ここを「cc2000」に書き換えて下さい。書き換えられたら「OK」ボタンをクリックして下さい。するとパネルが閉じてニュースグループの一覧が表示されます。



図 7.7: サーバマシン名の設定



図 7.8: NewsBase のメニュー

¹⁷ ニュースの記事を取りに行く先のマシンを指定します。cc環境では cc2000 という名前のマシンがそれに当たります



図 7.9: MMEdit のメニュー

次に、次回からいちいち「cc2000」と入力しなくてすむようにプリファレンスの設定をしておきましょう。画面左上にある「NewsBase」のメニュー（図 7.8）から「案内」をクリックして、それからその下の「プリファレンス」をクリックして下さい。すると、図 7.10のパネルが表示されますので、その中の「NNTP server」の枠で囲まれた中にある「マシン名」のところを先程と同じく、「cc2000」としてください。

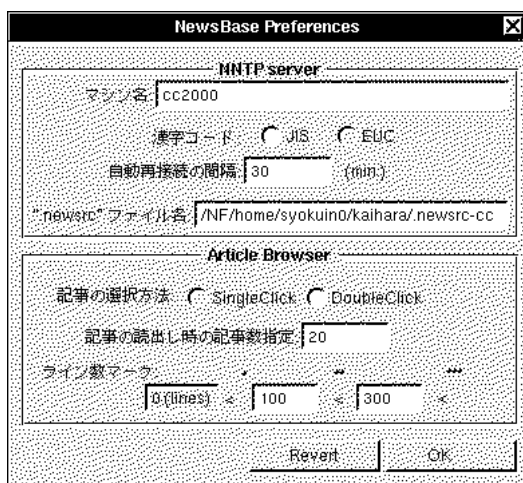


図 7.10: NewsBase 設定画面

書き換えが終わったら OK ボタンをクリックして下さい。続いて、「NewsBase」も終了させましょう。図 7.8の「NewsBase」メニューから今度は「終了」をクリックして下さい。

すぐにもう一度「NewsBase」を起動して記事を読みたくなるでしょうが、ちょっと待って下さい。もうひとつだけ設定してほしい項目がありますので、もうしばらくお付き合い下さい。

「MMEdit」は記事を投稿すると、その記事に自動的にあなたのメールアドレスを書き込んでくれます。そのメールアドレスにはマシン名も含まれるのですが、「MMEdit」が勝手につけてくれるそのままのマシン名を C3 情報処理教室の環境では使うことができませんので、自分で書き直す必要があります。

まず、「MMEdit」を起動して下さい。(アイコンは「NewsBase.app」の隣にあります。)次に画面の左上にある案内メニューをクリックして、その中の「プリファレンス...」の項目を選んで下さい。図 7.11のパネルが出てきましたか? そのパネル中の「From:」と「Reply-To:」の項目を自分のメールアドレスに変更して下さい。

たとえば、kaiharu さんでしたら、
kaiharu@ccnext.kyoto-su.ac.jp

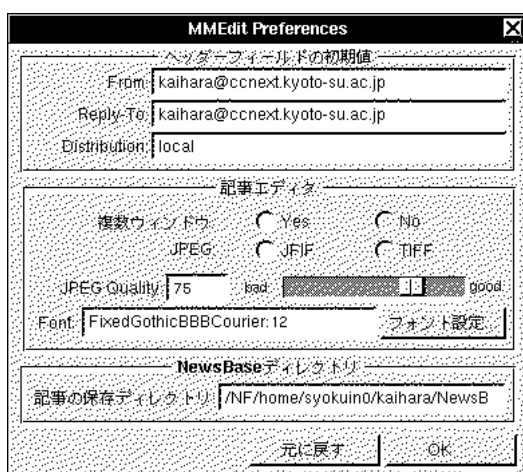


図 7.11: MMEdit 設定画面

とします。

書き換えが終わったら OK ボタンをクリックして、そのまま先程の「NewsBase」の終了と同じ要領で、「終了」メニューをクリックして「MMEdit」を終了させます。

これで準備万端です。さあ、たくさんの愛すべき？記事があなたを待っています。

「さあ、はりきってまいりましょう～～！」

読んでみよう

まず、記事を読むことから始めましょう。再び「NewsBase」を起動します。図 7.7 のパネルが出てきたら、「nntp サーバマシン名:」の欄が「cc2000」に変わっているかを確認してから、OK ボタンをクリックします¹⁸。

次に図 7.12 のウィンドウが画面に表示されます。

ここで試しに「sandai.announce」¹⁹ 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「fj」と「sandai」という 2 つの文字がありますね。「sandai.announce」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下さい。クリックすると右隣の一番上に「announce」が出てきます。これが「sandai.announce」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下さい。

パネルが現れて「××個を読み込みます」とたずねてくると思います。そのニュースグループに含まれる記事の数が少ない場合はたずねてこないのですが、「sandai.announce」にはたくさんの記事がありますので、パネルが開いてたずねてくるのです。「OK」ボタンをクリックすると「sandai.announce」の記事の見出しが図 7.12 の NewsBase のウィンドウの右半分に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 7.13 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下さい。

¹⁸ もし、localhost のままだったら、設定のしそこないです。「初めての NewsBase」を読んでもう一度トライしましょう。

¹⁹ このニュースグループはみなさんに有用な情報がいろいろ流されるので今後も定期的に読むとよいでしょう。

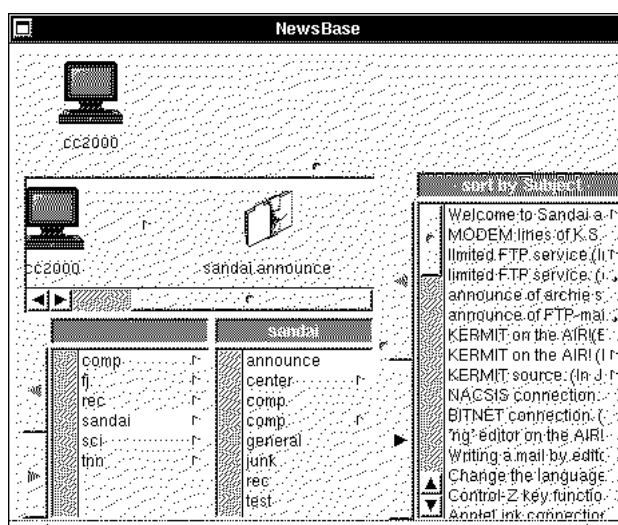


図 7.12: ニュースグループの表示

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なおせば、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っ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²⁰と Subject 欄²¹それから Distribution 欄²²に記入して下さい。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 他の記事に意見を述べることをフォローといいます。フォローを投稿するには、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。

²⁰ 記事を投稿するニュースグループの名前 (sandai.comp や fj.sys.next) を書く欄です。

²¹ 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。

²² 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。

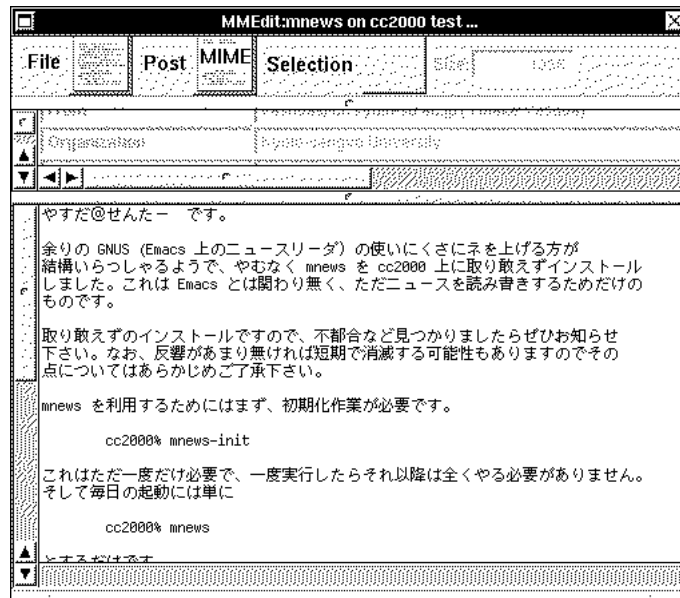


図 7.13: 記事を読む

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ります。「NewsBase」の機能として、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キャンセルできるというものがありますが、現在の C3 情報処理教室の設定では正常に動作せず、記事をキャンセルすることができません。

キャンセルしないで済むように慎重に投稿することはもちろんですが、もし自分の本意でない記事を投稿してしまった場合は訂正記事を書いて対処して下さい。

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、「UNIX ガイド」の「4.5.13 記事を書く場合の注意」で紹介していますので必ず読んでおいて下さい。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となります。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。

7.2.4 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なら電子メールやネットワークニュースで質問を投げかけてみましょう²³。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²⁴。

²³ 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。

²⁴ でも、必ず答えてくれるという保証もありませんが…。

第 8 章

Mathematica

8.1 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、いっても何のことも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。

8.1.1 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、アプリケーションで cc2000 のコマンドツールを呼び出します。そして mathematica と打ち込みます。すると、

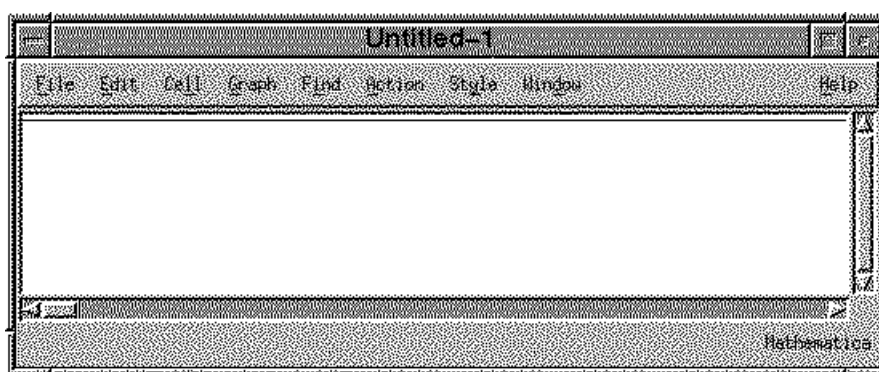


図 8.1: ノートブックのウインドウ

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。

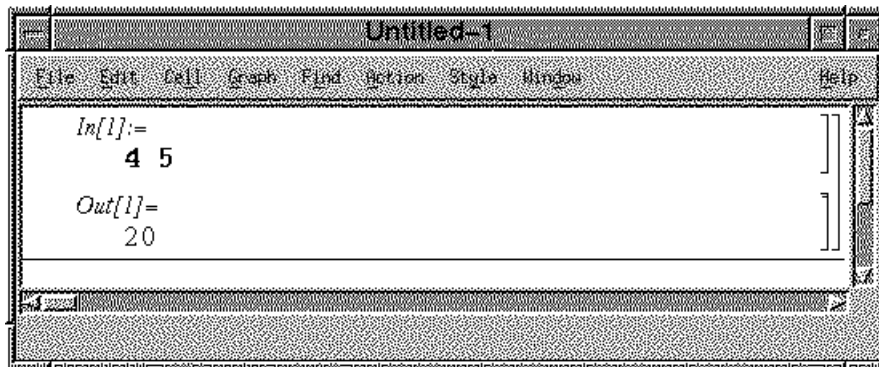


図 8.2: 最初の簡単なかけ算

『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、

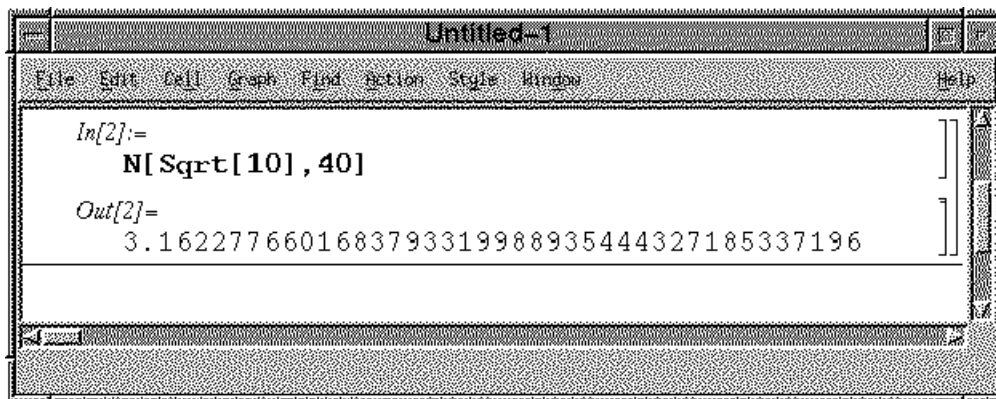


図 8.3: ちょっとした計算

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。

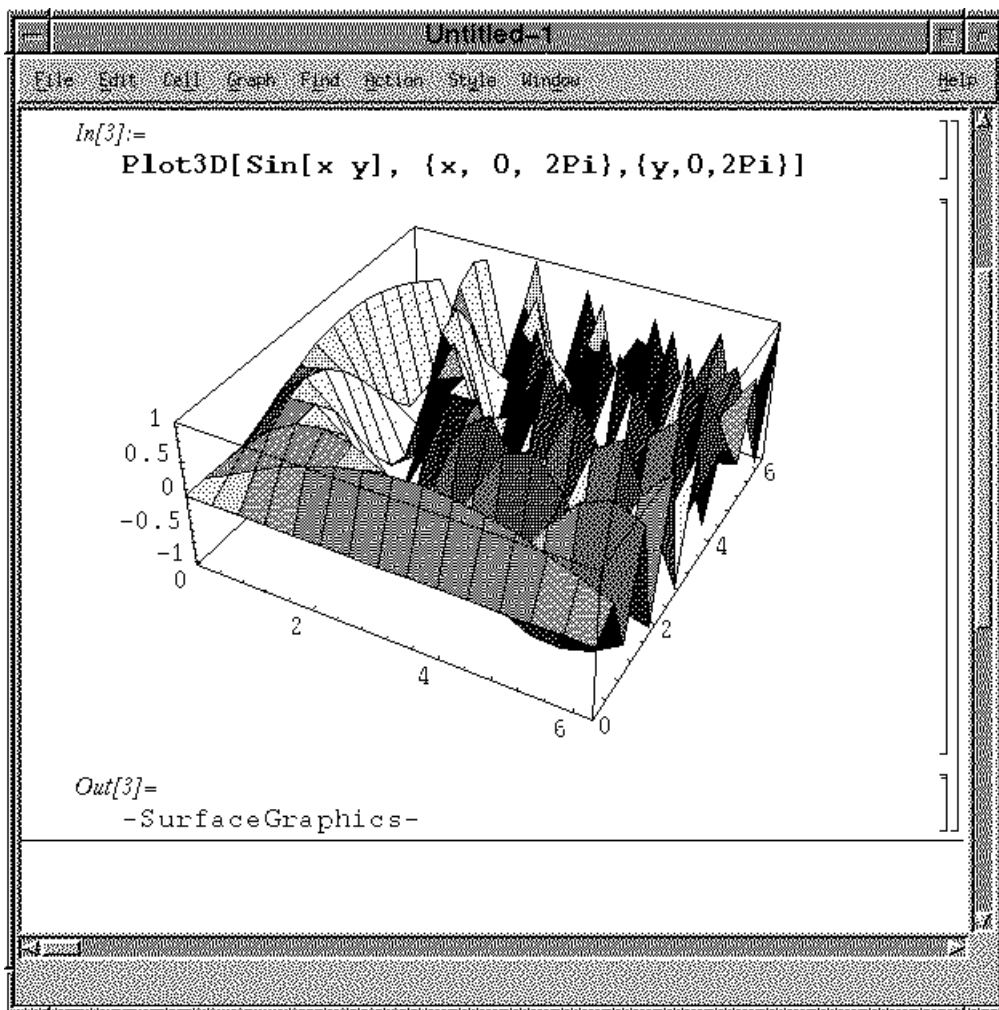


図 8.4: $y = \sin(xy)$ のグラフ

終了する時はメニューで Quit を選択します。

8.1.2 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 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 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。

8.2 ノートブックとコマンドラインについて

さて、先に、

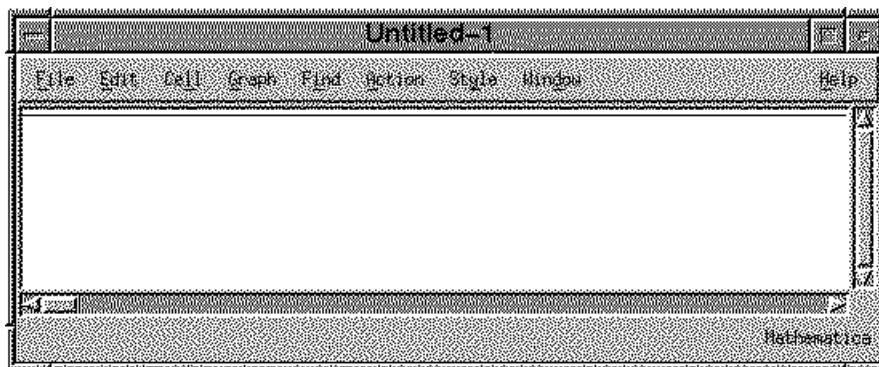


図 8.5: ノートブックのウインドウ

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。

8.3 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す。

- 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。

8.4 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
- エディタをたちあげて、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             ⇐ 入力

```

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。

8.5 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⁵、それを参考に勉強を進めて下さい。

⁵C2 情報処理教室に何冊かあります。

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。
(full spec of command)	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。
コマンドの働き	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。
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。
(full spec of option)	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      (directory)   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
```

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
```

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
```

diff : 二つのファイルの差分を表示する (difference)

```
diff [-iw] file1 file2  
-i (ignore case) 大文字と小文字を区別しない  
-w 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
```

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] [-cinv] 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
-Anumber        (after)      マッチした行の後ろ number 行を含めて表示する
-Bnumber        (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
-v              (invert)      マッチしなかった行を表示する
patter         検索するキーワード
name           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      (flag)    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
```

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
```

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 name.Z ファイルに作成せずに標準出力に出力する
        name ファイルは消去されない
-v      (verbose)  圧縮率を表示する
name    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 name から.Z 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
        name ファイルは消去されない
-v      (verbose)  圧縮率を表示する
name    ファイル名
```

gzip : ファイルを圧縮する

gzip [-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 *name.z* ファイルに作成せずに標準出力に出力する
 name ファイルは消去されない
-v (verbose) 圧縮率を表示する
name 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

gunzip [-cv] [name...]

name は最後に *.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 *name* から *.Z* もしくは *.z* 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
 name ファイルは消去されない
-v (verbose) 圧縮率を表示する
name ファイル名

A.1.3 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 [-Pprinter] [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。

-Pprinter *printer* で示されるプリンタに出力する
name 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 [-Pprinter]

-Pprinter *printer* 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)

lprm [-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.1.4 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 [-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。

-signal *process-id* 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。
 デフォルトでは -15 (-TERM) が与えられ、大抵これで終了させられる。強制終了の為に
 は -9 (-KILL) を与える
process-id 終了させたいプロセスの番号

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

A.1.7 雑多なコマンド

date : 現在の日付を表示する

```
date
```

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

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 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]
-a (all) 全ての情報を表示する
-p (processor) プロセッサ型を表示する
-s (operating system) OS 名を表示する。これはデフォルトで与えられる
-v (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.1.9 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]
-H (header) 見出しを出力する
-q 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
-h 見出しを表示しない
-l (long format) 長い出力形式。これはデフォルトで与えられる
-s (short format) 短い出力形式
username 利用者名 username 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...
-l (long format) 長い出力形式
-m keyword を利用者名に限定して検索する
-s (short format) 短い出力形式
keyword 利用者名、氏名などの断片
username 利用者名。省略した場合は hostname コンピュータを利用している
利用者一覧が表示される
hostname 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.1.10 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

rcp : リモートマシンのファイルをコピーする (remote copy)

rcp [[*username*@]*hostname*:]*source-file* [[*username*@]*hostname*:]*destination*
rcp -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.2 Emacs コマンド

A.2.1 絶対覚えておいた方がいいもの

emacs <Return>	Emacs を起動する
C-x C-c	Emacs を終了する
C-h T	Emacs(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>	バッファのカーソル以降で検索する
C-s	次を検索する
<ESC>	検索終了

egg

C-\	egg 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.2.2 必要に応じて覚えるもの

起動時

<code>emacs filename</code>	<code><Return></code>	Emacs を起動して <code>filename</code> をバッファに読み込む
<code>emacs -q</code>		.emacs を無視して Emacs を起動する
<code>emacs -u usr_name</code>		<code>usr_name</code> の人の .emacs の設定で Emacs を起動する

ファイル操作

<code>C-x i filename</code>	<code><Return></code>	別ファイルをカーソルの位置に差し込む
<code>C-x C-s</code>		カレントバッファを保存する
<code>C-x s</code>		すべてのバッファを保存する
<code>C-x k</code>		カレントバッファを保存せずにクローズする
<code>C-x C-v</code>		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)
<code>C-x b</code>		バッファを切替える
<code>C-x C-b</code>		バッファリストを表示する
<code>C-x C-q</code>		書き込みモードを変更する
<code>M-x recover-file</code>	<code><Return></code>	autosave された内容を読み込む

コードの変更

<code>C-x C-k d</code>	画面入出力コード変更
<code>C-x C-k i</code>	キーボードからの入力コード変更
<code>C-x C-k f</code>	ファイルの入出力コード変更 (E:EUC N:NEC S:SJIS J:JIS)

その他

全角、半角文字

<code>M-x zenkaku-region</code>	リージョン範囲を全角にする
<code>M-x hankaku-region</code>	リージョン範囲を半角にする

文字の入れ換え

<code>C-t</code>	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

<code>C-u n Command</code>	<code>Command</code> を <code>n</code> 回繰り返す
または <code>M-n Command</code>	例: <code>C-u 5 C-_</code> アンドゥ5回

カーソル操作

<code>M-a</code>	文の先頭へ移動する
<code>M-e</code>	文の末尾へ移動する

カーソルの位置

<code>M-x what-line</code>	<code><Return></code> 今カーソルが何行目にあるか表示する
----------------------------	---

改ページ

<code>C-q C-l</code>	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

検索

逆向き検索

- C-r *String* カーソル位置より上方向に向かって検索する
- C-r 次の見つける
- <ESC> 検索を終了しカーソルを最後に検索をした文字列のある位置にとどめる
- C-g 検索を終了しカーソルを検索開始前位置に戻す

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 5 左右に二分割する
 - 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消去する
 - C-x 0 カーソルのあるウインドウを消去する
- 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
- l 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

egg

変換

M-h	(変換途中に) ひらがなにする
M-k	(変換途中に) カタカナにする
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
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
-------------------	------------------------

その他

x?	小さい文字を出す (例: 「xa」で「あ」)
----	------------------------

MHE

読む

M-x mh-rmail <Return> MHE メールリーダを起動する
q MHE メールリーダを終了する
. 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

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

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

```
(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 (defrule "nn" "ん")

```
(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 version-control t)
```

```
(setq kept-old-versions 0)
```

```
(setq kept-new-versions 2)
```

```
(setq trim-versions-without-asking t)
```

ファイル名.~バージョン番号~ というバックアップファイルを作り、最新の 2 つのバージョンをバックアップファイルとして残し、古いバージョンを自動的に削除します。

```
(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。

付録 B

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 ——

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・荻原 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 \LaTeX 、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。

—— TeX について ——

逆引き \LaTeX D.J. パーガー著、引地 信之・引地 美恵子訳；マクロウヒル刊

「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、齊藤信男監修、鷺谷好輝訳；アスキー出版局刊

教祖様直筆 \TeX nician の Bible。頭から噛じるに困難でも、昼寝の枕に最適。

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。

謝辞

「UNIX ガイド」の出版から半年が経過し、当時書き切れなかった多くの事柄について、今回多くの方が書き起こしてくれました。

計算機センターは「続 UNIX もっともっと」「パソコンから UNIX」と UNIX コマンドのリファレンスを書きました。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 Emacs の章とそのリファレンスを理学部計算機科学科の尾崎君が、 \LaTeX の章の前半を理学部物理学科の辻本君が、後半を山崎君¹が書いてくれました。AUCT \E X の章については理学部研究科物理学専攻の松浦君が、Mathematica の章については理学部研究科数学専攻の笠君が再び書いてくれました。MODEM から UNIX の章は経済学部の小坂田君が書いてくれています。突然の依頼にも拘らず快く巻頭に書いて頂いた黒住先生、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。

1994.8.27 計算機センター事務室

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。