

# ワークステーションを使ってみよう

三木 邦弘

平成7年 1月 7日

## 目次

<b>1</b>	<b>ワークステーションとは</b>	<b>3</b>
1.1	ネットワークの時代	3
1.2	MS-DOSとUNIXの違い	3
1.3	文書清書システム	4
<b>2</b>	<b>利用開始と終了</b>	<b>5</b>
2.1	端末ソフトの起動と終了	5
2.1.1	学園センターの端末	5
2.1.2	004室の端末	5
2.2	パスワードの重要性	6
2.3	login	6
2.4	logout	7
2.5	パスワードの変更	7
2.6	演習問題	7
<b>3</b>	<b>テキストの編集</b>	<b>8</b>
3.1	漢字の入力方法	8
3.1.1	OAKによる漢字入力	8
3.1.2	ATOKによる漢字入力	8
3.2	テキストエディタ	9
3.2.1	使い方の基礎の基礎	9
3.2.2	使い方の基礎	9
3.2.3	知っているると便利な使い方	10
3.3	Unixのコマンド	11
3.4	演習問題	11
<b>4</b>	<b>電子メール</b>	<b>12</b>
4.1	メールアドレス	12
4.2	メールを扱うプログラム	13
4.3	メールを送る	13
4.4	メールを受け取る	13
4.5	保存していたメールの扱い方	15
4.6	その他の機能	15

4.7	演習問題	15
<b>5</b>	<b>電子ニュース</b>	<b>16</b>
5.1	様々なニュースグループ	16
5.2	fjの内容	17
5.3	学内独自のニュースグループ	19
5.4	ニュースの読み方	19
5.5	ニュースへの応答の仕方	21
5.5.1	記事に対してメールを送る	21
5.5.2	記事へのフォロー	22
5.5.3	記事の投稿	22
5.5.4	記事の取り消し	22
5.6	演習問題	22
<b>6</b>	<b>文書の清書</b>	<b>23</b>
6.1	L <sup>A</sup> T <sub>E</sub> Xとは	23
6.2	L <sup>A</sup> T <sub>E</sub> Xの動かし方	24
6.2.1	コンパイル	24
6.2.2	画面への出力	24
6.2.3	プリンタへの出力	25
6.2.4	L <sup>A</sup> T <sub>E</sub> X実行時に作られるファイル	25
6.3	基本的な書き方	25
6.3.1	使用する文字	26
6.3.2	特殊文字の表記	26
6.3.3	コマンドと環境	26
6.3.4	原稿の書き方	27
6.3.5	タイトルページ	27
6.3.6	文章構成	28
6.3.7	空白	28
6.3.8	行、段落	28
6.3.9	改行	28
6.3.10	文字の大きさ	29
6.3.11	文字のスタイル	29
6.3.12	行の離し方	29
6.3.13	中ぞろえ	30
6.3.14	右ぞろえ	30
6.3.15	引用	31
6.3.16	箇条書	31
6.3.17	そのままの形	32
6.3.18	要約	32
6.3.19	表	32
6.3.20	数式	33
6.3.21	様々な記号	33
6.4	演習問題	34

# 1 ワークステーションとは

まずワークステーションとは何か?となりますが、ちょっと強力なパソコンと思えばだいたい間違いはありません。以前はかなり処理能力などに差がありましたが、現在では能力的にはワークステーションもパソコンも同等なものが多いようです。ただ個々バラバラに使われることを想定されて作られてきたパソコンと、ネットワークを構成して利用することを想定されたワークステーションでは、特にネットワークへの対応の部分で大きな違いがあります。

パソコンでネットワークを構築しようとする、ネットワークの接続のための機器を別に購入する必要があります。さらにパソコンで標準的に使われている基本ソフトウェア (MS-DOS) はネットワークに対応していないので、別の基本ソフトウェアに交換しなければなりません。これに対してワークステーションならば、ネットワークに接続するにはケーブルを追加するだけです。基本ソフトウェア (UNIX) はネットワークに対応しているので、設定を行うだけで他のワークステーションのディスクを利用したりすることが簡単にできます。

パソコンも次第にネットワークを考慮したものに変わりつつあります。近い将来パソコンとワークステーションの目立った違いは無くなり、価格の高いワークステーションは消えていくと思われます。しかし、現在、ネットワークにつながったコンピュータの利用を学ぶには、ワークステーションを利用するのが一番の様です。

## 1.1 ネットワークの時代

高度情報化社会と言われて随分時が立ちますが、なかなか普通の人には縁がありませんでした。確かにワープロ、ファックス等は普及しました。しかしTV、ラジオ、新聞等のマスメディアは相変わらず送り手から受手のへの一方通行です。最近になってパソコン通信という形のメディアが使われるようになってきました。当初は一部のマニアばかり (野郎ばかり) だったのですが、次第に一般の人の利用が増えて、企業などでも社内のコミュニケーションの手段の1つとして認められて来ました。

近い将来、会社ではもちろん家庭に居てもコンピュータネットワークを利用して様々な情報を入手したり発信するのが普通になるでしょう。まあ、その練習?をしようと言うわけです。

## 1.2 MS-DOSとUNIXの違い

ワークステーションでは基本ソフトウェアとしてUNIXが使われています。これはパソコンで使われているMS-DOSと異なり、マルチタスク、マルチユーザーに対応しています。マルチタスクとは同時に複数のプログラムが実行できる事です。またマルチユーザーは同時に複数の利用者がシステムを利用できると言うことです。

マルチユーザーのシステムでは必ず、今使用しようとしている利用者は誰なのか明らかにする必要があります。またシステムの共有できる部分は勿論共有して構いませんが、個人的なデータを他の人が勝手に見たり変更できては困ります。そのためにシステムの利用開始時に利用者を確認する手続きと、全てのファイル等に誰にどのようなアクセスを許すかなどの設定をしておく必要があります。その点個人による個人利用しか考慮していないMS-DOSは気楽で、そのような手続きや設定は全く必要ありません。そのかわり電源を入れてくれた人に誰でもシッポを振ってしまいます。

逆にMS-DOSとUNIXには同じ所はないのか?と言えばかなりあります。UNIXの方が歴史が古く、MS-DOSは最近まで新しい版に変わる度にUNIXのコマンドや機能を取り込んできました。どちらかと言えばこの共通点ばかり勉強したほうが、大抵の基本ソフトに共通する点を学ぶことになり良いかもしれません。ですがここでは時間の都合もあり、パソコンの事は身近なパソコンで自分で勉強するということにして特に取り上げません。

### 1.3 文書清書システム

このテキストの後半は文書清書システムの話です。これはワークステーションとどんな関係があるのか？と言われると困るのですが、ワークステーションではもっぱらワープロ代わりに使われているものです。普通のワープロとは全く異なった視点から作られています。

普通のワープロでは、画面上にあるようにプリントされます。見たままが得られるので、わかりやすいのです。逆にここで紹介するソフトは文書の構造的情報を指示して、清書はシステム任せにしますので、わかりにくいです。慣れれば使いやすい物ですが、これは学生をいじめるためにやるのだと言われると否定できない所もあります。

ワークステーションの基本ソフトであるUNIXはパソコンよりも長い歴史を持っています。画面に表示して、と言う今では何の不思議もない事が、かつては不可能もしくは非常に高価な機器を必要としていました。プリントするときに綺麗に出れば良い、かつ普通の文字が入力できれば使えると言う文書清書システムがこのような状況で生まれました。

現在は時代が変わったのでもう不用？と言えば、

- ワープロがまともに使えない人が多い。表題を用紙の中央に持ってくるのに、せっせとスペースを入れて目で確認している人が居る。
- 章や節に分かれた長い文章（卒論など）を作成しようとすると、ワープロは章の番号の付替等をしてくれないので、編集が面倒になる。
- 文章の形式を登録する機能があるので、うまく利用すれば文書の作成に専念できる。

逆に特にワープロ専用機などは、短い文章やポスターのようなものを作成するには向いていると思います。パソコンの場合、ワープロソフトを購入すればワープロとして使えるし、ここで紹介する清書ソフトも利用できます。やはり個人用にはパソコンが一番でしょうか？

## 2 利用開始と終了

現在この椋山女学園大学にはワークステーションが7台しかありません。ワークステーションも正しい？利用形態では、1人の利用者が1台のワークステーションを使用するのですが、現状では無理です。そこで現在学園センターに設置されているワークステーション cc01 にはパソコンが16台端末として接続されていますので、この演習ではこれを通してワークステーションを利用します。なお生社棟の004室の約20台のパソコンも端末として利用可能ですが、少し操作が異なります。

学園センターの端末となるパソコンは学生情報処理実習室の入り口の側のFMRです。ちょうどカウンターの前、ワープロの隣に2列に並んでいるやつです。その一番奥にプリンターと並んでいるのが、cc01という名前が付いているワークステーションです。見た目は回りのFMRと変わりませんが、メモリーやディスクの容量、処理速度等はかなり違います。

### 2.1 端末ソフトの起動と終了

#### 2.1.1 学園センターの端末

まず起動方法です。

1. パソコンの電源を入れる。
2. メニューの左側から2番目の「OA Program」へ黄色い部分を動かして(  を押す )  を押す。
3. メニューの上から3番目の「WSの端末」へ黄色い部分を動かして(  を2回押す )  を押す。
4. 「cc01 login: 」と出たら準備完了。

次に終了方法です。

1.  キーを押しながら  キーを押す。
2. メニューの1番右端の「利用終了」へ黄色い部分を動かして(  を1回又は  を5回 )  を押す。

#### 2.1.2 004室の端末

こちらはノートパソコンを端末としています。LANで接続されているために応答が早く、かな漢字変換もATOK7なので005室のパソコンと同様に使えます。ただこの部屋は盗難防止のために通常は鍵が掛かっており、事務室で鍵を借りる必要があります。

004室に入ってすぐ左手にワークステーションとプリンターがあります。プリンターを利用する予定の人は、電源を入れておいて下さい。スイッチは本体の右側面後方にあります。

まず起動方法です。

1. 蓋を開ける。(自動的に電源が入ります。)
2. メニューが表示されるので、「ワークステーションの端末」を選択する。(  を押す。)
3. 「sss-1 login : 」と表示されたら、cc01 と入力する。たまに接続に失敗して「MS-Kermit> 」と出ることがあります。そのときには、c と入力します。
4. 「cc01 login: 」と表示されたら準備完了。

次に終了方法です。

1. ワークステーションの利用を終了すると、「MS-Kermit>」と表示されます。ここでq と入力する。
2. 「何かキーを押してください」と出るのでスペースバーでも押す。
3. メニューが表示されたら蓋を閉める。(数秒たつと電源が自動的に切れる。)

004室を退室するときは、空調や部屋の電気を消します。プリンターの電源も入っていたら消してください。ただしプリンターの隣にあるワークステーションの電源は絶対に切らないでください。

## 2.2 パスワードの重要性

ワークステーションは利用者の入力するユーザー名とパスワードで利用者の識別をします。あなたの銀行のカードの暗証番号と同様に他の人に教えたりしないで下さい。また安易な語をパスワードにしていると、見破られる原因になります。世の中にはそんなに悪い人は居ません。しかしごく少数の賢い悪人は、ネットワークと自分のマシンを使って自動的に世界中のシステムを荒らそうと狙っています。だから自分の身の回りには善良な人しかいない、そんなにコンピュータを使える人はいないと安心してはいけません。

銀行のカードならば悪用されてもあなたが泣くだけで済みますが、大学や会社のシステムを利用する際のパスワードが漏れると、システムの全利用者に被害が及ぶことがあります。そしてその責任の大半はあなたが取らなければなりません。社会に出る前にきっちりパスワードが扱えるようになって欲しいと思います。

銀行のカードの暗証番号は、カードがなければ実際は使えないために数字4桁と言う簡単なものになっています。このシステムの様には誰でもパスワードさえ合えば利用できるような場合は、簡単に想像がつくようなものや辞書に載っている英単語等は駄目です。住所や家族から想像されるようなもの、誕生日や学籍番号から想像されるようなものが駄目な例です。適当な文字や数字を混ぜて適当に長いものがお勧めです。

## 2.3 login

システムの利用を開始する手続きです。「login:」と出ているところで、あなたのユーザー名を入力し、を押します。次に「passwd:」と出てきますので、あなたのパスワードを入力します。このときキーを押しても画面には何も表示されないの慌てないで下さい。最後にはやはり を押します。ユーザー名やパスワードに間違いがあると再び「login:」が出ますので、また最初から入れ直して下さい。

login が成功すると次のような感じのメッセージが出ます。

```
Lsst login: Tue Nov 1 10:54:10 from bach
SunOS Release 4.1.3_JLE1.1.3 (GENERIC) #3: Sat Sep 24 12:55:59 JST 1994
```

これは前回利用した時刻や基本ソフトの構成などを示しています。もしあなた宛の電子メールが来ていたらこの後に、

```
You have mail.
```

等のメッセージが表示される事があります。そして、

```
cc01[miwako]%
```

が表示されると、システムはコマンドを入力しても良い状態になったことが判ります。このようなコマンドの入力を促すメッセージの事をプロンプトと呼びます。

## 2.4 logout

システムの利用を終了する手続きです。プロンプトの出ているときに、logout と入力すると利用終了の処理がなされて、再び「login:」と表示されて、他の利用者が利用できる様になります。利用終了時には必ずこれを行わないと、次の利用者に不正に利用される可能性がありますのでご注意ください。

## 2.5 パスワードの変更

最初は全員同じパスワードがとりあえず設定されています。このままではお互いに相手になりすまして利用することが出来てしまいますので、必ずパスワードを変更してください。そのやり方は次のようになります。

1. プロンプトが出ている所に、passwd と入力する。
2. 「Changing password for ユーザー名 on cc01」と表示され、次に「Old password:」と出るので、現在のパスワードと を入力する。このとき画面には何も表示されないので慌てないように。以下の入力の際も同様です。
3. 「New password:」と表示されたら、新しいパスワードと を入力する。
4. 「Retype new password:」と表示されたら、もう一度新しいパスワードと を入力する。
5. 再びプロンプトが表示されたらパスワードの変更は成功。もし、「Mismatch – password unchanged.」と出た場合は、新しいパスワードの入力間違いですので、もう1回最初からやり直して下さい。

本当は毎月のようにパスワードを変更するべきだと言う意見もありますが、そこまでは必要ないでしょう。ただ変更したが、何に変更したか忘れてしまったような事はないようにお願いします。

## 2.6 演習問題

1. パソコンで端末ソフトを起動し、login をし、すぐ logout を試みよ。
2. 再びlogin を行い、パスワードの変更を行え。その後 logout し、すぐ login を試みて、パスワードが正しく変更されていることを確認せよ。
3. login を行い、プロンプトが表示されたら、letters と入力せよ。すると画面の上から英単語が降ってくるので、キーボードで入力する。正しく入力できれば得点となり、画面の下に達するまでに正しく入力出来なければ過失となり、3回失うと終了してしまう。最終得点の上位20位までは記録に残る。なお途中で終了したい場合には、**CTRL**キーを押しながら**C**キーを押すと「are you sure you want to quit?」と聞いてくるので y と入力すれば良い。
4. キーボードからの入力の練習ソフトとして typist というものがインストールされているので、キー入力の遅い人は練習しておくこと。typist と入力すればメニューが表示されるので、それに従えば良い。

### 3 テキストの編集

現在のコンピュータは本来の数値情報の処理だけでなく、文字、音声、画像、動画等を扱うことができます。処理する数値や文字の入力にはキーボードがよく使われ、他の入力にはそれぞれ専用の装置が使われます。

実際のデータ入力の際には必ずごみや誤りがまぎれ込みます。正しく処理を行うためには、これらを訂正しなければなりません。ここでは数値や文字データの入力や訂正に用いられるエディタの使用法について述べます。

#### 3.1 漢字の入力方法

英字や数字はキーボードから直接入力できますが、我々が通常使う漢字などの文字は直接入力することはできません。かつては大きな文字盤に全ての漢字の一覧を載せて、特殊なペンで入力したい文字を指示するようなものもありました。現在は通常のキーボードからローマ字またはカタカナで読みを入力し、それを漢字に変換する方式がよく使われます。必ずしも読みと漢字は1対1に対応していないことと、変換の際の様々なキー操作が変換ソフトによって異なる事が大きな2つの問題です。

##### 3.1.1 OAKによる漢字入力

FMRで利用できるかな漢字変換ソフトの名前がOAKです。本来は富士通の開発した親指シフトキーボードと共に用いられるのですが、本学では通常のキーボードと共に用いています。設計思想として「長い読みを変換するとどうしても誤変換が避けられないので、短い文節単位で変換を行う。」ことにしていますので、長い読みを入れると変換率は非常に悪くなります。その点に注意すればそれほど悪くはありません。

変換開始  キーを押す。(このとき画面の右下に「R英小」とあったのが「Rかな」に変わります。)

読みの入力 ローマ字で入力しますが、「～が」と言うような助詞までにとどめます。漢字の長い複合名詞も途中で変換したほうが確実です。促音の「ぁ」などはLと組み合わせて入力します。「ー」はXで入力します。

変換キー ひらがなやカタカナにする場合は、 キーを押します。2回押すとカタカナになります。

漢字にしたいときは、 キーを押します。

変換終了  キーを押す。(このとき画面の右下の「Rかな」が「R英小」にもどります。)

変換中に既に入力した文字が隠される事があります。句読点は単独で入れると半角のものが入るので注意が必要です。(次で説明するテキストエディタや後のほうで説明する文書清書システムは半角の句読点やカタカナが含まれるテキストは扱えません。)

##### 3.1.2 ATOKによる漢字入力

パソコンのワープロソフトのベストセラーの「一太郎」で用いられているかな漢字変換プログラムがATOKです。多くの他社のかな漢字変換プログラムがこれと同様のキー操作で変換を行うようになっていますので、これに慣れておけば応用が効きます。

変換開始  キーと キーを押す。(このとき画面の下に「連R漢」と出る。)

読みの入力 ローマ字で入力します。長い文節でもかなり正確に変換しますが、ほどほどの長さにとどめる方がよろしいようです。



変換キー ひらがなにする場合は、**F・6** キーを押します。カタカナにする場合は、**F・7** キーを押します。  
漢字にしたいときは、スペースバーを押します。

変換終了 **CTRL** キーと **XFER** キーを押す。ただし変換の途中では終了できないので注意。(このとき画面の下に再びファンクションキーの表示が復活します。)

## 3.2 テキストエディタ

ワークステーションでよく使われているエディタは、vi系とemacs系に分けられます。カーソルを動かすためのキー操作等が大きく異なりますので、どちらかの系統で慣れると他の系統のエディタは使いにくくなります。また同じ系統のエディタは基本的な操作はほとんど変わりません。

vi系の代表のviはUnixに標準装備のエディタです。ですからUnixの動くマシンではどこでも利用可能です。ただ操作に関して、コマンド入力状態とテキスト入力状態と分かれているので、初心者にとって分かりにくい点があります。

emacs系はUnixの標準装備ではありませんが、多くのマシンにインストールされています。基本的に入力した文字はテキストとして挿入されるので、ワープロに慣れた人には違和感が少ないようです。

ここではemacs系のngと言うテキストエディタについて説明します。

### 3.2.1 使い方の基礎の基礎

まず起動をするときは、プロンプトの出ているところで、ng ファイル名 と入力します。ngの後は必ず空白が必要です。

以下では次のような形でキー操作を説明します。

C-x **CTRL** キーを押しながら、x キーを押す。

C-x y C-x を押した後、y キーを押す。

C-x C-y C-x を押した後、C-y を押す。

ESC x **ESC** キーを押した後、x キーを押す。

ESC C-x **ESC** キーを押した後、C-x を押す。

編集を終えた後は保存しなければなりません。C-x C-s と入力すると最初に指定したファイルに保存されます。既に保存されていて新たな変更がなされていないときには、(No changes need to be saved)と表示されて保存は行われません。

ngを終了するには、C-x C-c と入力します。変更したのに保存をせずに終了しようとする、保存しませんがと聞いてくるので、必要があればyと答えます。

### 3.2.2 使い方の基礎

まずカーソルの移動です。一応キーボードの矢印キーでもカーソルが移動できるように設定してありますが、手がホームポジションから離れてしまいますので、できるだけこちらでやるようにしてください。

C-f カーソルの一文字前進 C-b カーソルの一文字後退

C-n カーソルの次行への移動 C-p カーソルの前行への移動

C-v 次の画面への移動 ESC v 前の画面への移動

通常の文字を入力するとそのままカーソルの位置に挿入されます。カーソルの位置の文字を消去する時は、C-dを入力します。入力中に今入力したばかりの文字を消したい場合には、**BS** (FMRでは $\leftarrow$ ) を押します。

ちょっと変わったコマンドとして1行の長さを揃えるものがあります。電子メールやニュースのテキストをエディタで入力すると、編集しているうちに1行の長さはふぞろいになります。1行の長さを揃えたい部分の前後を空行で分離し、その中にカーソルを移動してESC qを入力すると、各行はほぼ70字の所に揃えられます。

### 3.2.3 知っているると便利な使い方

1. まずはその他のカーソル移動のコマンドです。

C-e	カーソルの行の終わりへの移動	C-a	カーソルの行の先頭への移動
ESC >	カーソルのファイルの最後への移動	ESC <	カーソルのファイルの先頭への移動

この他に、C-gを入力すると移動したいのは何行目かを聞いてくるので、行数と を入力するとその行にカーソルが移動します。

2. カーソル以降の一行削除には、C-kを入力します。大量に削除する場合は、その先頭の位置でC-@を入力し印(mark)を付け、最後の次の文字のところでC-wを入力します。このときこの削除した内容はバッファと呼ばれる領域に保存されます。ここには1回の削除分しか入りませんが、C-yを入力するとその内容をカーソルの位置に挿入することができます。これを利用してテキストのファイル内での移動が実現できます。コピーをしたい場合には、C-wの代わりにESC wを使用すれば元の部分が削除されません。
3. 検索をしたい時には、C-sを入力して続けて探したい文字列を入力します。このとき最後に を入力する必要はありません。入力したものと同一文字列がC-sを入力した時のカーソルの位置以降にあればそこへカーソルが移動します。さらに一致する文字列の位置へ移動させたいときは、もう一度C-sを入力します。検索を止めるときは、ESCを押します。C-rは探す方向が現在のカーソル位置より以前になる他はC-sと同じ動作になります。
4. テキスト中もある特定文字列を全て、または部分的に置き換えるときには、ESC %と元の文字列と置き換える文字列 を入力します。すると該当する箇所にカーソルが移動しますので、

!	以下の候補を全て一気に置換する
スペース	置換実施後次の候補へ
BS	置換しないで次の候補へ
ESC	置換の終了

のいずれかのキーを押します。
5. 編集集中にちょっとngを中断して他のコマンドを実行したくなることがあります。そのような時には、C-zを入力するとngは一時停止してプロンプトが表示されてコマンドの入力が可能になります。逆に中断しているngに戻る時には、fg と入力します。中断しているのを忘れてlogoutしようとするとき、  
There are suspended jobs.  
と言われますので、このコマンドでngに戻ってngを終了して下さい。
6. 他のファイルの内容をカーソルの位置に取り込むには、C-x iとファイル名と を入力します。

### 3.3 Unixのコマンド

Unixのコマンドは標準でも1000に近い数のコマンドが使用可能です。その中でも良く使われるファイルに関するものをいくつかあげます。なお説明の最後に( )の中にかかれているのは、ほぼ同じ働きをするMS-DOSのコマンド名です。

**ls** ファイルの名前を表示するコマンドです。ファイルの名前だけでなく大きさなども知りたいときには、`ls -l` のように`-l`を付けます。(dir)

**cp** ファイルのコピーをするコマンドです。元のファイルとコピー先の指定が必要です。(copy)

**mv** ファイルの移動をするコマンドです。元ファイルと移動先の指定が必要です。cpと異なり元のファイルは消えます。単に名前を変更するのにも使われます。(ren)

**rm** ファイルを消去するコマンドです。消したいファイル名を指定します。(del)

**lpr** ファイルの内容をプリンターに印字するコマンドです。印字したいファイル名を指定します。(print)

**man** コマンドの説明(マニュアル)を表示するコマンドです。説明を見たいコマンド名を指定します。次の部分を見たいときには、スペースパー前に戻りたい時には`b`を入力します。見終わったら`q`を入力します。(対応するもの無し)

### 3.4 演習問題

1. プロンプトが出ているときに`ng Ng.txt` と入力せよ。すると`ng`の使い方の説明になるので、良く読んで書いてあるとおりにせよ。
2. プロンプトが出ているときに`ng aaa` と入力し`ng`を起動せよ。そしてこの章の最初の部分を入力し保存せよ。画面の横幅は狭いので適当な所で改行して行末の長さをできるだけ揃えること。また入力したテキストをコピーし、同じ文章を5回並べよ。正しく入力し保存できたら`ng`を終了し、できたファイル`aaa`の内容をプリンターに出力せよ。
3. プロンプトが出ているときに`man ng` と入力すると`ng`の説明が表示される。同様にして他のコマンドの説明を幾つか表示させて見よ。

## 4 電子メール

電子メールとは手紙の本文が電子的に送られるものです。基本的に1対1の通信に用いられるものですが、郵便、電話、ファックスなどと比べると様々な違いがあります。

郵便と比較すると、その伝達の速さが違います。電子メールはネットワークの結合状態によりますが、相手先が海外でも数十秒程度で相手に届きます。伝えられる内容は文字情報が基本ですが、音声や画像も送れない事はありません。封筒ですと手紙以外にお金を同封したりもできますが、電子メールではちょっと無理です。文章をワープロで作成し、それを紙に打ち出して封筒に入れるのならば、コンピュータに入力した文章を電子メールで直接相手に送る方が紙の節約になります。

電話はその場で相手と話ができますから、相談などをするのには便利です。しかし、相手が不在ではつながりませんし、音声によるものなので後でもめ事になった場合に証拠がありません。海外のように生活時間が異なる場合はお互いになりに気を使わなければなりません。電子メールだと相手は相手の都合の良いときに読む事ができます。逆に相手が、届いたメールを読むためにコンピュータにアクセスするまで、こちらのメッセージは伝わらない事になります。

ファックスは送った内容が紙に残ります。伝達速度も速いし時差に関係ない、紙に書いてある情報ならば絵でも文章でも同じ手間で簡単に送れるために広く会社関係で用いられています。ただファックスで得られたものは、あまり綺麗ではないし、これをさらに加工して送り返すとなると、ちょっと問題です。電子メールの場合、受け取ったものはまたそのまま次へ送る事が可能です。情報の追加や削除も容易です。例えば誰かが作った原案を、それを電子メールで受け取った人が添削してまた別の人に送るようなことができます。

電子メールは、郵便、電話、ファックスのように社会にそれを運用する組織があるとか、利用の仕方を常識としてみんなが知っていると言うような段階にはきていませんが、これから社会に広く普及していく技術である事は間違いありません。

### 4.1 メールアドレス

郵便物には宛名が必要です。電話も相手の電話番号が必要です。電子メールを送るのにもメールアドレスが必要です。かつてネットワークがごく小規模な範囲に留まっていた時代にはメールアドレスは、単にユーザー名のみ、またはメールの転送経路を直接示すものでした。今日ではインターネットを通して世界中の国々とメールの交換が可能になっています。これに対応して論理的な構成を反映したメールアドレスを利用するのが現在では普通です。

この学園センターのワークステーションの利用者のmiwakoさんのメールアドレスは次のようなものになります。

```
miwako@cc01.center.sugiyama-u.ac.jp
```

@の前がユーザー名です。その後はピリオドで区切りながら、マシン名(cc01)、部署名(center)、組織名(sugiyama-u)、組織分類(ac:学術)、国名(jp:日本)を示しています。このメールアドレスでインターネットにつながっているどのような組織からもmiwakoさんに電子メールが届きます。

アメリカのみ例外ですが他の国は国名を示すメールアドレスが決まっています。brがイギリスとかcaがカナダと言うような感じです。組織の分類としては、国内ではacの他に会社組織がco、政府関係がgo等に決まっています。組織名はそれぞれ様々なものがありますが、同名異組織では困りますので、JPNICと言う組織がみな異なるように管理しています。ただこの組織が管理しているのはこのレベルまでで、それより前に来る組織内のアドレスは各組織の自由になっています。

通常はこの長いメールアドレスをいちいち書かなくても良いように設定がされています。同じcc01に登録された利用者同志ならば、単にユーザー名だけで十分です。

## 4.2 メールを扱うプログラム

電子メール(以下メールと略す)を実現するためには多くのプログラムが働いています。まずメールの内容を編集するためのエディタ、メールを1つのマシン内で配送するプログラム、ネットワークを通してメールを送ったり受け取ったりするプログラム、受け取ったメールの本文を表示するプログラムなどがあります。通常の利用者はそれらの働きを全て知る必要はありませんが、最低限メールを送る方法と読む方法を知らなければなりません。

エディタにも何種類かあったように、利用者が直接触れるメールを送ったり読んだりするプログラムも何種類もあります。Unixに標準的に付いてくるmailの他、多くの人が様々なものを作っています。ここでは比較的初心者向けのelmを取り上げます。基本的にどのプログラムもできることは同じなので、後に別のプログラムを使うことになってもコマンドの対応さえわかれば問題無いでしょう。

なお、メールを扱うMS-DOS上で動くプログラムもあります。これを利用するとUnixの事は何も知らなくてもメールを扱うことができます。会社などの組織で本当にメールの機能だけ必要な場合にはよく使われます。

## 4.3 メールを送る

メールを送るにはまず相手のメールアドレスが判らなければなりません。今それがaaa@bbb.cccだったとします。メールを送るときには、プロンプトが出ている状態で、elm aaa@bbb.ccc と入力します。すると

```
Send only mode [ELM 2.4 PL23]
```

```
To: aaa@bbb.ccc  
Subject of message:
```

と表示されます。ここでSubject(見出し)を入力します。メールの内容の要約を簡単に英語かローマ字等に入れます。学内では問題ありませんが、学外に出す場合ここに全角の文字を使用するとその部分が文字化けする事がありますので、日頃からこのようにしてください。

この後はエディタが自動的に起動されます。そこで相手に送るメールの本文を入力します。このとき1行の長さに注意します。(長すぎると途中で切れてしまうことがあります。)画面の右の方が少し空くぐらいにします。この様にするときにはngのESC qを使うと便利です。保存してエディタを終了すると、

```
Please choose one of the following options by parenthesized letter: s  
e)dit message, edit h)eaders, s)end it, or f)orget it.
```

となります。通常はここで を押すとs)endと言うことで相手にメールとして送られます。本文の編集を再開する時はe、Subject等を修正したいときにはh、メールを送るのを中止するならばfを押します。

## 4.4 メールを受け取る

各利用者宛に送られたメールは通常ある特定のディレクトリ(/var/spool/mail など)にファイルとして保存されます。そしてloginの際にメールが着いていると、

```
You have mail
```

等のメッセージがプロンプトの前の行に表示されます。特定ディレクトリにあるファイルを直接読む事は可能ですが、通常は複数のメールが1つのファイルになっているために、メール用のプログラムで読みます。

メールを読むにはプロンプトが出ている時に `elm` と入力します。すると画面が変わり、中央に到着しているメールの一覧、下の所にコマンドの一覧が出ます。何もメールが来ていないときには中央部分は空白になります。

例えばメールがいくつか届いている場合には次のような感じになります。

```
Mailbox is '/var/spool/mail/miwako' with 2 messages [Elm 2.4PL23]
N  1  Apr 24  Larry Fenske      (49)  Hello there
N  2  Apr 24  jad@hpcnoe      (84)  Chico? Why go there?

You can use any of following commands by pressing the first character;
  d)elete or u)ndelete mail, m)ail a message, r)eply or f)orward mail, q)uit
To read a message, press <return>. j = move down, k = move up, ? = help
```

Command:

Nと言うのはまだ読んでいないメールです。読むと空白になります。また削除のコマンドを入れるとDになります。次がメールの番号です。その後に届いた日付、送り主の名前、行数、Subjectの内容と続きます。この後は色々なコマンドを入力してやってきたメールを片づけます。

- jやkのキーを押すか、矢印キーを押すと文字が反転している部分が上下します。反転しているのが現在指定しているメールです。
- 又はスペースを押すと、指定されたメールの内容を読む事ができます。長い内容のメールの場合、スペースを押すと続きを見る事ができます。前に戻るにはbを押します。読み終わったらqを押します。すると、

Command ('i' to return to index):

と出てきますので、このメールを保存する(ファイルにして取っておく)ならばs、このメールを消去するならばd、返事を出すのならばr、メールの内容を印刷したいときはp、単に最初の画面に戻るだけならばiを押します。

- dを押した場合には、他にメールが来ていればその内容が表示されます。
- sを押した場合には、

```
-----
Command: Save
Save message to: =相手の名前
```

が出ます。通常は を押して相手の名前のファイルにします。同じ相手からのメールは1つのファイルにどんどん追加される形で保存されます。何か別のファイルに保存したい場合にはそのファイル名を入力します。保存が終わると、最初の画面に戻りNだったものがDに変わります。

- rを押した場合には、

```
-----
Command: Reply to message      To: 相手の名前かアドレス
Subject of message: Re: 相手からのメールの見出し
```

と出ます。Subjectとして、相手からのメールのSubjectにRe:が付いた物がセットされているので通常は単に を押します。すると相手から着たメールの本文が各行の先頭に>が付いた形でコピーされますので、適当にそれを編集して返事が書けます。後はメールを送る時と全く同じ手順になります。

- dを押すと現在指定しているメールにDと言うマークが付きます。実際に削除されるのはelmを終了するときです。取り消すためにはuを押します。Dが付いたメールがあると終了時に、

Command: Quit

Delete message? (y/n) y

と表示されますので、通常は を押します。

- qを押すとelmが終了します。もし読まなかったメールがあると、

Keep unread message in incoming mailbox? (y/n) y

と表示されますので、そのまま を押します。

#### 4.5 保存していたメールの扱い方

相手からのメールを保存しておく便利な事があります。メールを出すときに一番間違い易いのが相手のメールアドレスです。電子的に発送されるメールは郵便のように誤字脱字があっても配達人が機転を利かせて配ってくれるような事はありません。その割に1と1のような区別のつきにくい文字が良く使われます。相手からのメールが保存してあれば、それに対する返事を書くと言う操作でメールが送れますので相手のメールアドレスを入力するする手間が省けます。

既に保存してあるメールを見るためには、プロンプトが出ているときに、`elm -f +ファイル名` と入力します。このときファイル名は相手のユーザー名になります。後の操作は前節と全く同じです。よって保存していたメールの削除などもできます。

これまでに送ったメールの全てがSmailと言うファイルに保存してあります。よって`elm -f +Smail` でそれを呼び出して確認などができます。

ところで相手のユーザー名が思い出せない事もあります。そのような場合には、保存したファイルはすべてMailと言うディレクトリーの下にありますので、プロンプトが出ているときに、`ls Mail` と入力すれば保存してあるファイル名を知る事ができます。

#### 4.6 その他の機能

外国でタイプライターで書かれた手紙などでは最後に確かに本人からのものである証拠として手書きの署名を付けることがあります。それと同様に自分の発送するメールに決まった文字列を自動的に追加する機能があります。これを利用するにはプロンプトの出ているときに、`chgmsig`

#### 4.7 演習問題

1. 2人または数人でグループを作り、お互いにメールを送ってみよ。相手が居ない場合には自分自身に送れ。なお演習の仲間同志でメールを交換する際にはメールアドレスとしてはユーザー名のみを指定すれば良い。
2. 受け取ったメールを読み、保存し、返事を送れ。
3. 保存してあったメールを利用してメールを送れ。

## 5 電子ニュース

コンピュータネットワークでのコミュニケーションの方法として全章の電子メールと共に電子ニュースは大きな比重を占めています。前者が基本的に1対1のコミュニケーションを対象としているのに対して、後者は多対多のコミュニケーションを実現します。新聞等のマスメディアに対応する物ですが、従来のマスメディアが少数の送り手だったのに対して誰でも送り手になれるという特徴があります。

電子メールはその特徴の一つに即座に伝わると言うものがありました。電子ニュースはテレビほど早くは伝わりません。ネットワークの混み具合にもよりますが、数日の伝達の遅れはよくあります。パソコン通信等で用いられるBBS(Bulletin Board System:電子掲示板)もほぼ同じ様なものですが、これらが集中的な管理がなされているのに対して電子ニュースは特に管理者はなく多くの利用者の善意と努力で運用されています。

電子ニュースは多くのニュースグループにより構成されています。あるものは研究室内で、あるものは大学内で、あるものは世界中に流れています。現在インターネットで流れているニュースは1日あたり約7万記事、その大きさは約200MBになる日もあり、年々その流量は増大しています。

これだけの量になるととても一人の人が目を通す事はできません。ほとんどの人にとって、役に立たない情報が大部分を占めます。しかし、自分で必要とする情報を求めている事を記事にして投稿して、多くの人から情報の提供を受ける事が可能です。もちろん一方的に情報を得る事ばかりでは、響きを買う事はまちがいありません。自分も日頃から有益な情報を提供して、困ったときには助けてもらうような姿勢が大切だと思います。

### 5.1 様々なニュースグループ

現在電子ニュースを運用している組織にはまずまちがいなく組織独自のニュースグループがあります。一般的には外部の人がそれを知る事はできません。ここでは広く世界的に流通しているニュースグループについて紹介します。

次の表は94年12月11日に流れたニュースのグループ別の集計です。ここで説明するグループの部分だけ取り出しています。オリジナルはtnn.netnews.statsと言うニュースグループに毎日投稿されていますのでそちらをご覧ください。

group	article (%)	byte (%)	block (%)	byte/article
alt	17440 (23.6)	77261406 (39.3)	83918 (36.8)	4430
comp	11451 (15.5)	24997849 (12.7)	30089 (13.2)	2183
fj	416 ( 0.6)	944207 ( 0.5)	1127 ( 0.5)	2269
rec	23737 (32.1)	44268193 (22.5)	54863 (24.1)	1864
sci	1952 ( 2.6)	4230732 ( 2.2)	5068 ( 2.2)	2167
soc	5075 ( 6.9)	12782702 ( 6.5)	14996 ( 6.6)	2518
...				
total	73940	196508311	227839	

alt ここはなんでもありと言うグループです。非常に雑多な内容で流量も多いので残念ながら本学ではこのグループを取っていません。(要するに読めない。)

comp ここはコンピュータに関するグループで初歩的な内容から非常に高度な内容まで含んでいます。本学ではこのグループを取っていますが、cc01には取っていません。

fj ここはfrom Japanと言う事で日本語の記事が流れています。(実は他のグループの記事はほとんどが英文です。) fjの内部のグループについては後で述べます。



rec ここは趣味のグループです。様々なジャンルの話題を取り扱っていますが、これも流量が多いので本学では取っていません。

sci ここは科学技術に関するグループです。本学ではこのグループを取っていますが、cc01には取っていません。

soc ここは社会に関するグループです。その大部分は世界中の様々な民族の文化に関するグループです。その中にはsoc.culture.japaneseのような日本文化を扱う所もあり、海外の人が日本の文化をどのようにとらえているかを見る事ができます。

## 5.2 fjの内容

fjは日本発のニュースと言うことで大部分が日本語で書かれています。そしてそのサブグループとして、comp、rec、sci、soc等を持っています。そしてその中がさらにサブサブグループに分かれるような構造になっています。たとえばfj.rec.sports.soccerはfjの中のrecの中のsportsの中のsoccerと言うグループを示します。

fjの中にも約240のグループがあるので全てを紹介できませんがその一部を以下に紹介します。これのオリジナルは富士通の市川さんによるものです。

- +-ai AI、人工知能、認知科学などに関する議論
- +-announce fjの読者の全員に関係するような記事
- +-archives 文書、FAQなどを(将来はプログラムも)投稿するグループ
- +-binaries 実行形式のプログラムなどを投稿するためのグループ
- +-books 各種書籍について
- +-comp
  - | +-arch 計算機アーキテクチャについて
  - | +-comm 通信のハードウェアやソフトウェアなど諸般の事項の議論
  - | +-dev 各種計算機デバイスに関するグループ
  - | +-dsp デジタル信号処理に関する話題
  - | +-image 画像処理、コンピュータグラフィック、コンピュータアニメ等についての議論
  - | +-misc 計算機に関する様々な話題
  - | +-music コンピュータミュージック関連の話題
  - | +-oops オブジェクトオリエンテッドに関する話題
  - | +-parallel 並列処理に関して(システム、アルゴリズム等)
  - | +-printers プリンタについて
  - | +-speech 計算機を使った音声処理に関する議論と情報交換
  - | +-texhax TeXに関する議論、意見交換など
  - | +-text テキスト処理(含むワードプロセッサ)に関する話題
  - | +-theory 計算機科学の基礎理論。アルゴリズム、計算の複雑さ、形式モデル、意味論などを含む
- +-editor 計算機上で動く各種のテキストエディタに関する話題
- +-education 教育一般に関して
- +-followup 各種の記事に対する、特にfj.announceについての‘‘その後の議論’’をここで行う
- +-forsale 個人的な売り、または、交換しようとする物について
- +-guide fjの手引きなどについて
- +-jokes 冗談に関する情報(ではなくて冗談そのもの)
- +-kanakan 種々な漢字変換に関する話題
- +-kanji ネットワークや計算機における漢字の取り扱いなど
- +-lan ローカルエリアネットワークについて
- +-lang 様々なプログラミング言語について
- +-life
  - | +-children 子供、子育て、親の役割について
  - | +-in-japan 外国人の日本での生活に関する話題
  - | +-money お金との付き合い方に関する議論
  - | +-religion 宗教に関する議論
- +-living 家事・住環境・健康・家族の世話等の、日常生活に関わる話題と情報交換
- +-mail ネットワークの電子メールについて

```

+-misc 投稿すべきグループが他にないとき使う
|   +-handicap handicap 解消に関する議論、および行動
+-net ネットワーク関係の話題、情報、議論
+-news ネットワークニュースシステムに関する話題
+-os 各種オペレーティングシステムにかかわる話題
+-questions
|   +-unix UNIXに関する質問とその回答
|   +-misc 各種の質問とその回答
+-rec
|   +-aerospace 飛行機、飛行船、スカダイビングなど、空に関すること、物、話題
|   +-animation アニメーションに関する情報、議論
|   +-autos 自動車に関する記事
|   +-av AV(音響/映像機器)に関すること
|   +-bicycles 自転車の話題
|   +-bus バスについての議論
|   +-comics マンガについて
|   +-fine-arts 美術に関する話題、美術とは視覚的芸術のことである
|   +-fishing 釣りに関する話題
|   +-food 料理、うまい店、健康食品などについての情報、議論
|   +-games 各種ゲームについて
|   +-ham アマチュア無線(HAM)についての話題
|   +-idol いわゆるアイドルについて
|   +-marine ダイビング、水泳、ヨットなど海洋に関すること
|   +-misc その他の楽しみについて
|   +-models 模型全般に関する話題
|   +-motorcycles 自動二輪(原付を含む)の話題
|   +-movies 映画についての情報、感想、議論など。
|   +-music 音楽に関するあらゆることについての情報、感想、議論など
|   +-mystery ミステリについての話題
|   +-pachinko パチンコに関する議論
|   +-pets ペットの話題
|   +-photo 写真とカメラに関するグループ
|   +-play 演劇、劇場等に関する話題
|   +-radio ラジオ放送に関する話題
|   +-rail 鉄道に関して
|   +-sf SFについて
|   +-sports スポーツに関するさまざまな情報と議論
|   +-tokusatsu 特撮映像作品に関する話題
|   +-travel 旅行に関する情報
|   +-tv 各種TV番組(CMを含む)について
+-sci
|   +-astro 星と宇宙について
|   +-bio 生物学に関する議論を行う
|   +-chem 化学に関する議論を行う
|   +-geo 地球惑星科学に関する議論
|   +-human-factors 人間工学についての議論
|   +-informatics 情報学に関する議論
|   +-lang 自然言語について
|   +-math 数学について
|   +-medical 医学に関係する話題
|   +-misc その他の各種科学について
|   +-physics 物理学に関する議論を行う
|   +-psychology 心理学に関する議論と探求
+-soc
|   +-culture 社会と文化について
|   +-environment 自然環境と社会に関わる話題
|   +-history 歴史についての議論
|   +-human-rights 人権、生命倫理、差別問題などを議論する
|   +-law 著作権、生存権などなど。各種法律に関する議論
|   +-media 放送・出版などを含む通信媒体の社会的側面に関する議論
|   +-men-women 男女差別の議論など、男と女にかかわる社会的なトピックス

```

```

|  +-misc 社会現象、または、社会科学に関する議論その他
|  +-smoking 喫煙に関する問題やマナーについて
|  +-tech 科学技術と社会の関係についての議論
|  +-traffic 交通・運輸にかかわる、公害問題、行政やマナーなどの社会問題に関する議論
+-sources ソースプログラムを投稿する
+-sys 各社の計算機システムについて。
+-test テストメッセージを fj 全体に流すために用いる
+-wanted なんらかの情報について知りたいとき
+-windows 各種ウィンドウシステムについて。

```

### 5.3 学内独自のニュースグループ

本学ではsugiと言う名前のニュースグループを運用しています。まだ記事を投稿する人も無く、読む人もいないのでほとんどサブグループがありませんが、これから段々発展させていく予定です。グループの作成や内容についての検討もこれからです。

```

sugi.announce  sugiの読者の全員に関係するような話題
sugi.test       ニュースの投稿の練習用のグループ

```

sugi に投稿された記事は本学の外へは出ません。もし外部の人にも訴えたいのならばfjを利用することになります。しかしいきなり日本中を相手にするのはどうかと思われる方はtokaiと言う東海地区だけに流れているニュースグループがありますのでこれを利用すると良いでしょう。ただ現在のところあまりこのグループを利用している人はいないようです。

### 5.4 ニュースの読み方

ニュースシステムを運用するには、記事を配送するプログラムと利用者が記事を読むためのプログラムが必要です。どちらのプログラムも Unix に標準では付いて来ませんので、入手してインストールする必要があります。

ここではmnewsと言う松下電器の宅間顯さんが開発したニュースリーダー(ニュースを読むプログラム)について説明します。実際はこれ以外にも様々なニュースリーダーがありますが、mnewsは小型化、高速化、そして簡単に使用できることを目標に開発されています。多くの学生が少ないワークステーションを同時に使う事を考えてこれにしました。

まず起動の方法ですが、プロンプトの出ているところで、mnews と入力します。すると次のような感じの画面に変わります。

```

-----
Mini News Reader Version 1.17 Copyright(C) By A.Takuma [NSPL]cc01
ニュースカテゴリ：                               位置：All [EUC:m]
  最大   未読   モード  ニュースカテゴリ/ニュースグループ名
  8315   8315   [Y]    control
          219042 [Y]    fj
    0     0     [-]   general
  6864   6864   [Y]    junk
          430645 [-]    soc
          5     [-]    sugi
          140  [-]    tokai
          [+]    MAIL
-----

```

最大の所の数値は記事の通し番号の最後の値です。未読の所はまだ読んでいない記事の数です。モードの所の[Y]は投稿可能、[N]は投稿不可、[M]はモデレートグループ(投稿可能ですがモデレータのチェック

を受けます)、[-]はこの下にサブグループがあること、[+]は、メールグループ(ここではこれの説明は省略します。)を示しています。

- ニュースグループを選ぶには、ngと同じようにC-p、C-nか、elmと同じようにk、jが矢印キーで選択するグループを指定します。そしてスペース押すとそのグループに入ります。階層構造をなしているニュースグループの場合は同様の選択を記事の一覧が現れるまで繰り返します。もし、グループ選択時にソートする記事数を聞いてきた場合は、0を入力します。
- 間違ったニュースグループを選んだり元に戻りたい場合はqを押します。最初の画面でqを押すとmnewsは終了します。また記事の内容を読んでいる画面以外でQを押しても終了します。
- いつも読まないニュースグループにはUを押すと、ニュースグループにU印が付いて、次回からそのグループは表示されなくなります。

記事の一覧は例えば次のような感じで表示されます。

```
-----
Mini News Reader Version 1.17 Copyright(C) By A.Takuma [NSPL]cc01
ニュースグループ: music.juzz                      位置: 24% [EUC:m]
マーク 番号 日付 行数 差出人          サブジェクト (1205-2255)
R   1642  06/29   24 tanaka@dokok Which title do you like best?
R   1643  06/29   69 suzuki@kokok || Which title do you like best?
R   1644  06/29    9 miyazaki@aso || Which title do you like best?
    1645  06/29   13 yamagata@aci || Which title do you like best?
    1646  06/29   27 watanabe@sop || Which title do you like best?
      (中略)
    1657  07/01   86 ogawa@yumeno || Which title do you like best?
    1658  07/01   21 yamagata@aci || Which title do you like best?
    1659  07/01   40 ishida@haten || Which title do you like best?
    1660  07/01   48 kobayash@dea I listened "xyz".
    1661  07/01   19 koizumi@shib Question about jazz.
-----
```

マークの所にRとあるのは既に読んだ記事です。何も無いのがまだ読んでいない記事です。記事はできるだけ同じサブジェクトが続くように並び換えられています。サブジェクトが|で始まっている記事は別の記事をフォローしている記事です。ここで、

- グループを選択するのと同じようにして記事を選択します。さらにPやNを押すと未読記事だけを移動できます。
- スペースを押すと記事の内容を読む事ができます。記事の内容表示中は、スペースで次ページへ、bで前ページへ移動できます。途中でqを押せば記事一覧画面に戻れます。
- Dやdで記事にRマーク(既に読んだと言う印)を付ける事ができます。その後dならば次の未読記事へ、Dならば前の未読記事へ移動します。
- UやuでRマークを解除します。その後uならば次の記事へ、Uならば前の記事へ移動します。
- sを押すと記事をファイルに保存する事ができます。次にファイル名を力して下さい。もしファイルが存在する時は、旧ファイルに追加するならばy、中止するならばn、上書きをするならばoを押して下さい。
- cを押すと全記事にRマークが付きます。確認を求めてきますので、本当にそうするならばy、そうでなければnを押します。

例えばこんな感じで記事は表示されます。Pathの部分はこの記事がどのようなマシンを経由してきたものかを示しています。Fromに投稿者のメールアドレス、Organizationに投稿者の所属する組織の名称が示されます。Message-IDは全ての記事に互いに異なるものが付けられるようになっているので、特定の記事を指定するときにはこれを用います。

```
-----  
Path: alps!sonoue!saraniue!ichibanue!sonoshita!hassinmoto  
From: watanabe@sop.dokojja.soreha  
Newsgroups: music.juzz  
Subject: Re: Which title do you like best?  
Message-ID: <1234@sop.dokojja.soreha>  
Date: 29 Jun 92 12:35:55 GMT  
References: <920625@dokoka.wakarann>  
Sender: news@sop.dokojja.soreha  
Distribution: music  
Organization: Konnna oraganization ha arimasen.  
Lines: 13
```

In article <920625@dokoka.wakarann>,  
tanaka@dokokja.soreha (Tanaka Hiroshi) writes:

>あなたはどんな曲が好きですか。

私は「あれこれ」という曲が好きです。  
他にも「あれそれ」も好きです。

[1646] Line:1 byte 450/854 (56%)

## 5.5 ニュースへの応答の仕方

既にある記事に対して何かの応答を行いたい場合には2つの方法があります。一つは記事の投稿者にメールを送ること。もう一つはその記事に対する記事を投稿する(フォローすると言います)ことです。また全く新しい記事を投稿することもできます。

ニュースで答える場合には慎重にそのニュースグループの様子をよく理解してからにしてください。compのような英文ばかり流れている所に日本語で投稿すれば「おまえの記事は全然読めないぞ」という抗議のフォローやメールが世界中から殺到するでしょう。fjでも非常に多くの読者が居て様々な価値観をもって記事を読んでいます。自分勝手な内容や誤解されるような書き方にならないように注意して下さい。fjは世界中に流れています。アメリカでは全てのニュースグループの記事をCD-ROMの形で保存しているそうです。つまらない記事でも永久保存されますのでご注意下さい。

### 5.5.1 記事に対してメールを送る

ニュースへの返信は、まず記事の一覧の画面でその記事にカーソルを移動してから、rまたはRを押します。Rの場合には元記事の引用が行われます。エディタに入ったら--text follows this line--の次の行以降に本文を書き、保存して終了します。あらかじめ.signatureというファイルに署名の内容を入れておくと自動的に追加されます。記事全文が表示されるのでqで終了、送信の確認にyを押せばメールが送られます。中止するならばn、もう一度編集するならばeを押して下さい。送ったメールはelmで送ったものと同様にSmailというファイルに保存されます。

### 5.5.2 記事へののフォロー

記事へのフォローは、まず記事の一覧画面でその記事にカーソルを移動してから、fまたはFを押します。Fの場合には元記事の引用が行われます。後はメールを送る場合と全く同じです。ただし投稿した記事はNewsと言うディレクトリーの中のSnewsと言うファイルに保存されます。記事の投稿には少し時間がかかることがあります。投稿したままではmnewsは新しい記事を認識しないので、一度mnewsを終了して再起動するか、ニュースグループの選択画面で+gと押してmnewsに再認識をさせる必要があります。

### 5.5.3 記事の投稿

投稿したいニュースグループの記事の一覧の画面でa押すと記事を投稿する事ができます。すると配布範囲(Distribution)を聞いてきますので、無指定と言うことで のみを押します。最後にサブジェクト(見出し)を入力します。メールの場合と同様にサブジェクトに日本語は使用しないで下さいサブジェクトを指定しないとニュースの投稿は中止になります。またニュースグループの選択の画面でaを押すとニュースグループ名も入力する必要があります。後はフォローする場合と全く同じになります。

### 5.5.4 記事の取り消し

自分の投稿した記事は取り消すことが可能です。しかし一旦投稿してしまった記事はマシンからマシンへと流れており、この取り消しも「あの記事は取り消し」と言う特殊な記事が後から流れて行く形になるので、取り消される前に多くの人の目に触れる可能性はかなりあります。ですから投稿するまえによく確認をし、些細な誤りでの記事の取り消しは混乱を招くので避けるほうが良いようです。

記事の一覧画面で取り消したい記事にカーソルを移動し、C(大文字)を押します。すると確認を求めてきますのでyを押します。なお他の人の記事は取り消すことはできません。

## 5.6 演習問題

1. fj.comp、fj.soc、fj.sci、fj.recのそれぞれの中のサブグループから各一つずつ、それら以外から一つニュースグループを選んで全ての記事を読み、そこでどのような話題が取り上げられているか、200字以上でまとめよ。
2. .signatureに適当な自分用の署名を入力せよ。これにはプロンプトが出ている状態でchgnsig と入力すればエディタが起動されるので、入力し保存、終了すれば良い。
3. sugi.testに何か適当な記事を投稿してみよ。
4. sugi.testの適当な記事に何かフォローをしてみよ。

## 6 文書の清書

ワークステーション上で文書を清書する際に最もよく使われている  $\text{T}_{\text{E}}\text{X}$  についてここでは説明します。実際は  $\text{T}_{\text{E}}\text{X}$  自身を扱うのは難しいので、これを簡単に使えるようにした  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  を使用します。

なおこの章は鈴木裕信氏 (hironobu@sra.co.jp) が書かれた「ひろのぶの  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  入門 Version 0.23」を利用して作りました。このオリジナルはネットワークで公開されており、後に書籍として出版もされています。これを大幅に縮小割愛したのがこの章です。よって文章表現にやや不統一がありますし、例題等はそのまま頂いていることをあらかじめお断りしておきます。

### 6.1 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ とは

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  は本、レポート、記事、手紙などを書いて印刷するための仕組みです。 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  の読み方はラテックです。レスリー・ランポート (Leslie Lamport) が作りました。このテキストも  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  で書かれています。文章が中身はともかくとして、美しくならんでいますね。章や節に正しく数字を振ったり、1行の文字を揃えたりするのは  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  が行なっているのです。 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  は簡単に高品質の印刷物を作るのに便利です。

目的は同じでもワープロとはちょっと違います。ワープロは「入力すること = 清書すること」ですが、 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  はその規則に則って書かれた原稿をプログラムにかけて初めて清書される (美しく配置される) のです。 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  の規則は、文の論理的な構造をもとにして、全体の文字をまとめあげようとしています。要するに文章を書く時の論理的なブロックとして構成して行くのと同様な単位で処理していきます。

「 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  や  $\text{T}_{\text{E}}\text{X}$  はワープロみたいな WYSIWYG じゃないから難しい。」と言う指摘があります。“What You See Is What You Get” (見えた通りのものを得ることが出来る) というのは裏を返せば、“what you see is all you’ve got” (結局は見えた通りのものしか得られない) ということです。ですから、センスの無い人がワープロを使うとセンスの感じられない体裁のものができます。何十ページにもおよぶ文書を作成する際に、最初から終わりまで文章のスタイルを一貫させることができない人がワープロを使って作成すると、内容とその表現の不一致が読者を悩ませるようなものができるでしょう。

ランポートは  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  を設計するにあたり次のようなことを考えました。

- 印刷の活字を組んで、どのようにレイアウトするかは著者のアイデアを読者に理解させるような形ではなければならない。
- ドキュメントは読むのに楽でなくてはならない、視覚的な構造は論理的な構造を反映させていなければならない。

そして、書く時の心構えについてこう述べています。

あなたが、文書を書く時は、視覚的にどう表現されるかではなく、論理的構造に対して配慮しなさい。

文章とは視覚的概念ではなく論理的な概念であるべしというわけです。さらに、文章を論理的構造に従って書いていくと、形式を変更する際や、できた文章をデータベースに入れる際に便利などの利点もあります。

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  はドナルド・クヌース (Donald Knuth) の作った  $\text{T}_{\text{E}}\text{X}$  (テックと読む) を土台として作られているプログラムです。おいしいイチゴケーキを  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  とすると、土台となる軟らかいスポンジケーキが  $\text{T}_{\text{E}}\text{X}$  というような関係にあります。 $\text{T}_{\text{E}}\text{X}$  自体は表現力は豊かなのですが、初心者には使い難いという欠点があります。しかし、 $\text{T}_{\text{E}}\text{X}$  はプログラミング言語としてより多くの機能を持ったシステムであり、熟練者がそれを基礎にし新しい処理系を作れるように設計されています。

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  に関しては様々な図書が出版されていますが、一番頼りになるのは、 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  を作った本人による Leslie Lamport,  *$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ : A Document Preparation System*, Addison-Wesley, 1986. です。幸いにしてこの訳本がアスキーより「文書処理システム  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 」として出版されています。

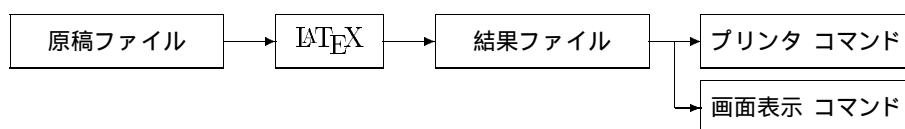
## 6.2 $\text{\LaTeX}$ の動かし方

$\text{\LaTeX}$ は現在では大型計算機からパソコンまで様々な機種で利用することができます。元来アメリカで作られたソフトウェアでしたので、当初日本語の文章には対応できませんでした。現在多くの人の努力により、日本語化された $\text{\LaTeX}$ には次の2種類があります。

- ASCIIの日本語 $\text{\LaTeX}$
- NTTの $\text{j\LaTeX}$

英語の部分に関してはオリジナルの $\text{\LaTeX}$ もASCIIの日本語 $\text{\LaTeX}$ もNTTの $\text{j\LaTeX}$ も同じ動作をしますが、日本語の部分は完全に互換ではありません。このテキストはASCIIの日本語 $\text{\LaTeX}$ を基に書いています。

処理の大まかな流れは、次のようになります。最初に $\text{\LaTeX}$ の形式で書いた原稿ファイルを作ります。 $\text{\LaTeX}$ のコマンドに原稿ファイルを与えると、結果ファイルが作成されます。結果ファイルをプリンタへの出力を行なうためにコマンドに与えて実際のプリントアウトを行ないます。



### 6.2.1 コンパイル

原稿ファイルを結果ファイルに変換することをコンパイルと呼んでいます。プロンプトが出ている状態で`platex` ファイル名 と入力します。原稿ファイル名には拡張子`.tex`がないと $\text{\LaTeX}$ に怒られますが、コンパイルする時のファイル名は、例えばファイルが`foo.tex`という名前だとすると、`foo`というように省略しても構いませんし、`foo.tex`と与えても構いません。

コンパイルをすると、原稿ファイル中の、相互参照、目次、索引などの情報をまとめておくファイルが自動的に現在実行中のディレクトリに作られます。原稿ファイルの中にこれらの機能を用いた場合、 $\text{\LaTeX}$ は参照できなかったラベルに出会った時と、すべての処理を終えた時に警告文を出します。よって目次などの機能を用いた場合にはコンパイルを2度実行する必要があります。

間違えてファイル名を指定する時がよくあります。そのような時 $\text{\LaTeX}$ は正しいファイル名を入力するようにしつこく聞いてきます。この状態の時、正しいファイル名を指定することなく抜きたい場合はファイル名のかわりに`null`と入力します。すると、 $\text{\LaTeX}$ がプロンプトを返してくるのですかさず、`\bye`か`CNTL-D`を入力して下さい。 $\text{\LaTeX}$ から抜けることができます。

処理途中でエラーが発生した場合は`x`を入力すると終了します。または`e`を入力するとエディタを起動してエラーの発生した行へカーソルを移動してくれます。エラーが発生した場合、 $\text{\LaTeX}$ はエラーメッセージと共にどうするかを聞いてきます(入力待ちになる)。なかなか判りにくいメッセージですが、良く見ながら`x`又は`e`を入力して誤りを訂正してください。

結果ファイルの名前は原稿ファイルの名前の拡張子`.tex`の部分が`.dvi`と替わったものです。原稿ファイルが`foo.tex`ならば結果ファイルは`foo.dvi`です。これは、プリンタに出力するためのコマンドを使って印刷するためのものであって、普通DVIファイルと呼ばれています。DVIは`DeVice Independent format`で、「出力すべき装置とは独立した形で情報を持っている」というような意味です。

### 6.2.2 画面への出力

ワープロと異なり $\text{\LaTeX}$ では原稿と出力の形態が全く異なります。ですから本当にきれいな出力が得られるかどうかは、出力して見なければ判りません。でも、毎回プリンターで印刷したのでは資源の無駄に



なってしまいます。

画像を出力できる端末ですと印刷するのと同じ様な感じで表示してくれるコマンドがありますが、現在利用できる端末は文字しか表示できない端末ですので、雰囲気程度しか判りません。それでも最初のうちはできるだけ画面で出力を確認してから印刷するようにして下さい。

プロンプトが出ている状態で、`jdvi2tty` ファイル名 と入力すると画面に出力に近いものが表示されます。スペースで続きを、`b` で前の方を見る事ができ、`q` でこのコマンドを終了する事ができます。

### 6.2.3 プリンタへの出力

学園センターのプリンタに出力する時は、`dviout` というコマンドを使います。004室のプリンターで出力する場合には `dviprt` というコマンドを使いますが、使い方は全く同じです。プリンターに出力するコマンド名は計算機の機種やプリンターの機種により様々な物があるので、他の環境ではおそらく異なるコマンド名でしょう。でもだいたい同じような機能を持っています。

`dviout` はDVIファイル(さっきまで結果ファイルとっていたもの)をレーザープリンタに出力します。プロンプトが出ている状態で、`dviout` オプション ファイル名 と入力します。

オプション<sup>1</sup>の部分には次のようなものを書くことができます。何も指定をしなければ、全てのページが印字されます。

<code>-f #</code>	#ページから印字する。	<code>-t #</code>	#ページまで印字する。
<code>-l</code>	用紙に対して横方向に印字する。	<code>-mf</code>	手差しした用紙に印字する。

ですから全てのページを印字する場合には、`dviout` ファイル名 と入力し、5 ページから8ページを出力する場合には、`dviout -f 5 -t 8` ファイル名 と入力することになります。

### 6.2.4 L<sup>A</sup>T<sub>E</sub>X 実行時に作られるファイル

L<sup>A</sup>T<sub>E</sub>X は入力や出力、自分自身が使う情報を蓄えておくファイルなどたくさんのファイルを使用します。実際に私達が扱うファイルは `foo.tex` か `foo.dvi` ぐらいですが、1度 L<sup>A</sup>T<sub>E</sub>X を実行すると色々な拡張子を持ったファイルが実行時のディレクトリに作られます。

例えば、拡張子 `aux` のついたファイルには相互参照表、索引、図目次、表目次などを作る時に必要な情報を書き出します。また、拡張子 `log` のついたファイルは L<sup>A</sup>T<sub>E</sub>X が実行中に画面に出力していた内容などを収めています。

これ以外には次のようなファイルが実行時の条件により作成されます。`\tableofcontents` のコマンドがあった時には、目次を作るために拡張子 `toc` のついたファイルが作られます。`\listoffigures` のコマンドがあった時には、図目次を作るために拡張子 `lof` のついたファイルが作られます。`\listoftables` のコマンドがあった時には、表目次を作るために拡張子 `lot` のついたファイルが作られます。`\makeglossary` というコマンドが使われた時には、`\glossary` というコマンドによって作られた情報が拡張子 `glo` のついたファイルに入り、用語索引ができます。`\makeindex` というコマンドが使われた時には、`\index` というコマンドによって作られた情報が拡張子 `idx` のついたファイルに入り、索引ができます。

## 6.3 基本的な書き方

ここでは L<sup>A</sup>T<sub>E</sub>X の書き方について説明を行いません。良く使う命令の書き方と、実際の出力結果のサンプルを載せます。細かい説明については省いています。

<sup>1</sup>`dviout` コマンドは内部的には `jdvi2kps` コマンドを呼び出しているの、これ以外に使える豊富なオプションに関しては `jdvi2kps` の `man` を参照してください。同じ物がそのまま使えます。

### 6.3.1 使用する文字

入力に使用できる文字は全角文字、半角の英数字、そして、次の16の特殊文字 (punctuation) です。

. : ; , ? ! ' ‘ ’ ( ) [ ] - / \* @

そして、次の10個の特殊文字は、 $\text{\LaTeX}$  の命令に使用されます。ですからそのまま使用するとどんな意味にも解釈される事があります。

# \$ % & - { } ~ ^ \

\ 以外は \#、\\$ とすることによって文字として表現できます。ただし、バックスラッシュ \ は次のようにキャラクタシンボルを表現する方法で行なうことが必要です。

$\backslash$  ⇒ \

$\text{\LaTeX}$  はテキスト中に空白があっても自動的につめてしまいます。意図的に空白を挿入する時は、\ する必要があります。改行の場合は \\ を使います。マイナス記号 ( - ) は特別な働きをします。連続して書くことによってハイフンなどの線 ( --- ) を引くことができます。例えば—このような時に便利です。注意しなければならないのは、マイナス記号が1つのときは、マイナス記号が2つ、3つ連続している時とはちょっと違うことです。次の例を良く見て下さい。

--- ---- ----- ⇒ - - - - -

### 6.3.2 特殊文字の表記

フランス語やドイツ語等の文字には、アクセント記号が付いたものがあります。以下はそれら文字の表し方です。例えば、*évêque* は  $\text{\e}v\text{\e}que$  と記述します。

$\grave{o}$ \{o}	$\ddot{o}$ \~{o}	$\ddot{o}$ \v{o}	$\text{\c}{o}$
$\acute{o}$ \' {o}	$\bar{o}$ \={o}	$\text{\H}{o}$	$\text{\d}{o}$
$\hat{o}$ \^ {o}	$\acute{o}$ \. {o}	$\text{\t}{oo}$	$\text{\b}{o}$
$\text{\o}$ \ " {o}	$\text{\u}{o}$	$\text{\aa}$	$\text{\AA}$
$\text{\ae}$	$\text{\AE}$	$\text{\l}$	$\text{\L}$
$\text{\o}$	$\text{\O}$	$\text{\oe}$	$\text{\OE}$
$\text{\ss}$	$\text{\?}$	$\text{\!}$	

### 6.3.3 コマンドと環境

$\text{\LaTeX}$  にはコマンド (command) と環境 (environment) の考え方があります。コマンドは局所的、環境は広域的に影響を及ぼします。

コマンドは  $\sim$  や  $\backslash$  で始まります。コマンドは  $\{ \}$  や  $[ ]$  で囲まれた引数を取ることができます。たとえば、脚注を書くコマンドは  $\text{\footnote}$  であり  $\text{\footnote}$  { 脚注です。 } と書けば、実際には現在のページの下<sup>2</sup>に書かれます。

$\text{\begin}$  や  $\text{\end}$  もそのような意味ではコマンドです。注意しなければならないのは、コマンドを識別子としてあつかう必要があるので、 $\text{\LaTeX}$  などは  $\text{\LaTeX}$  というようにしなければなりません。

環境とは  $\text{\begin}$  { name } と  $\text{\end}$  { name } で囲まれた範囲が、name という規則にすべて従うという感じ

---

<sup>2</sup>脚注です。

### 6.3.4 原稿の書き方

原稿は必ず `\documentstyle[option]{style}` で始めなければなりません。そして文章の本体は `\begin{document}` と、`\end{document}` で囲まれた範囲内に書かなければいけません。

原稿のスタイル (*style*) の標準的なものには次のようなものがあります<sup>3</sup>

**jbook** これは、本の形式 (スタイル) です。奇数ページの上側 (ヘッダ) では節 (セクション) の番号とセクション名が左側、ページ番号が右側に出ます。偶数ページヘッダでは右側に章 (チャプタ) 名、ページ番号が左側に出ます。章の最初のページが奇数になるようにページが送られます。章立ては必ず `\chapter` から始めなければなりません。もし、`\section` レベルから始めた場合、エラーとなります。

**book** これは英文の本のスタイルです。

**jreport** これは報告書の形式です。ページ番号は用紙の下部の中心にくるような形になります。章立ては必ず `\chapter` から始めなければなりません。もし、`\section` レベルから始めた場合、エラーとなります。

**report** これは英文のレポートのスタイルです。

**jarticle** これは最もよく使われる論文の形式です。このテキストもこのスタイルを使っています。`\section` のレベルから書くことができます。

**article** これは英文の論文のスタイルです。

オプション (option) はスタイルの補助的な役割をします。基本となるフォントサイズの変更や見開きに変更する事ができます。

**11pt** 基本フォントサイズを 11 ポイントにします。標準のフォントサイズは 10 ポイントです。

**12pt** 基本フォントサイズを 12 ポイントにします。

**twoside** 見開きを行なうためのスタイルです。*book*, *jbook* スタイルは標準でこうなっています。

**twocolumn** 2 段組のページにします。

**titlepage** *article*, *jarticle* にのみ有効です。このオプションを指定すると、`\maketitle` や `abstract` 環境を使った結果が独立した紙に出されます。(要するに表紙ができる。)

$\text{\LaTeX}$  は指定したスタイルのスタイルファイルと呼ばれているもの (*style.sty*) を最初に読み込みます。

### 6.3.5 タイトルページ

タイトルページは `\maketitle` があらわれた時点でタイトルページが作成されます。

---

⇒

<pre>\title{HOW TO USE \LaTeX} \author{Hironobu Suzuki} \date{4 Oct 1988} ..... \maketitle</pre>	<p>HOW TO USE <math>\text{\LaTeX}</math> Hironobu Suzuki 4 Oct 1988</p>
--	---

<sup>3</sup>スタイルの名前で先頭に *j* がついているものは日本語  $\text{\LaTeX}$  専用のものです。オリジナルの英語の  $\text{\LaTeX}$  で使用すると、エラーとなりますので注意して下さい。

### 6.3.6 文章構成

文章の構成で、章や節を表すものとして、次のようなものが用意されています。

<code>\part</code>	<code>\section</code>	<code>\paragraph</code>
<code>\chapter</code>	<code>\subsection</code>	<code>\subparagraph</code>
	<code>\subsubsection</code>	

各々は1つの引数を取ります。通常のjarticleスタイルでは、節(section)の中が項(subsection)に分かれ、さらにその中が部分項(subsubsection)に分かれます。jbookやjreportは章(chapter)の中が節に分かれ、以下jarticleと同様に細かく分かれて行きます。この際引数として与えるのはそれぞれの見出しで、番号は $\LaTeX$ が自動的に付けてくれます。文章を考えながら作成する際には、章や節の順番が入れ替わる事はよく生じます。ワープロであればその度に章や節の番号を自分でつけ直す必要がありますが、その作業を行う必要がないために文章を考える事に専念できます。

よってこの項の原稿には、`\subsubsection{文章構成}`と書いてあるだけで、それにどんな番号が付くかはできてからのお楽しみであります。

### 6.3.7 空白

半角の空白は複数あっても、1つの空白と見なされます。逆に全角の空白は目に見えませんが、行の前後に残っていると歯抜けの段落ができたりするので注意が必要です。

意図的に空白を挿入したい場合は`\□`とします。これは $\LaTeX$  on UNIXのような書き方をしたとき $\LaTeX$  on UNIXというような結果になってしまいます。このような結果を避けたい時は`\LaTeX\ on□UNIX`とすれば $\LaTeX$  on UNIXとなります。

### 6.3.8 行、段落

原稿の文字列は自動的に詰められ改行されたりします。空行は段落の切れ目を意味しますが、複数の空行は1行分の空行と解釈されます。段落の先頭は1字下げがなされるので、原稿では字下げは不要です。

————— ⇒ —————

左につめて書きます。ただし、前の行は空白行です。

左につめて書きます。ただし、前の行は空白行です。  
連続した行です。連続した行です。連続した行です。  
連続した行です。連続した行です。連続した行です。  
連続した行です。連続した行です。連続した行です。

連続した行です。  
連続した行です。  
連続した行です。  
連続した行です。  
連続した行です。  
連続した行です。  
連続した行です。  
連続した行です。

### 6.3.9 改行

改行を行ないたい場合は`\\`か、`\newline`を使用します。その働きは正確に言えば段落の切れ目を示す事ですので、後で出てくる表での使用以外の本文では、これの代わりに空行を用いた方が良いでしょう。

逆に文字列や単語を途中で自動的に切り離されたくないとき、`\mbox`を使います。例えば、*Segmentation fault*の*Segmentation*と*fault*を離したくない場合は`\mbox{Segmentation fault}`とします。実際はコンパイルして、印刷して見て変な場所だけにこのテクニックは使ったほうが良いでしょう。

### 6.3.10 文字の大きさ

文字の大きさは`\Large`のようなコマンドを使って変更します。直接ポイント数を与えて文字の大きさを変えるような野蛮なことはしません。`large`や`small`のコマンドによって文字のポイント数が変わりますが、原稿の基準フォントのポイントサイズに合わせて相対的に`large`であり、`small`であります。このように総体的に指定する事により、後になって基本となるサイズを変更した場合にも、本文の指定を変更する必要がなくなります。

文字の大きさの有効範囲は同一ブロック内です。ブロックとは、環境の中や`{ }`で囲んだ範囲の事です。例えば、`{\Large abc}d`と行なうと`abc`が大きくなり`d`は元のサイズのままです。実際に行なってみると`abcd`となります。もちろん、入れ子にでき、`{\Large a {\LARGE b {\small c}}}`は`a b c`というようになります。

ワープロの初心者が作成した文書などで、頻繁に文字の大きさを変える例があります。必要以上の文字のサイズ変更は見やすさよりも見にくさを助長するだけなので避けるようにしましょう。

文字 <code>\Huge</code>	文字 <code>\huge</code>	文字 <code>\LARGE</code>
文字 <code>\Large</code>	文字 <code>\large</code>	文字 <code>\normalsize</code>
文字 <code>\small</code>	文字 <code>\footnotesize</code>	文字 <code>\scriptsize</code>
文字 <code>\tiny</code>		

### 6.3.11 文字のスタイル

英数字の文字のスタイルは全部で5つありますが、全角に関しては太字のみです。そのため、スタイルの指定を行なっているブロック内に全角文字があっても太字以外にはなりません。

影響を与える範囲などは文字のサイズの時と同じです。`\em`を用いて影響の及ぶ範囲の例を示します。

ここで`emphasize`の指定をスタイルと同等に扱っていますが、`LATEX`の作者であるランポートの本によれば、下線とイタリック文字というのは、視覚的な概念であるのに対し、強調(`emphasize`)は論理的な概念であると述べられています。

使える文字スタイルは次のようになります。

————— ⇒ —————

```
{\bf Bold type style. 全角は太字です。}
{\sf Sans serif type style. 全角は同じです。}
}
{\sl Slanted type style. 全角は同じです。}
{\tt Typewriter type style. 全角は同じです。}
}
{\sc Small caps type style. 全角は同じです。}
}
{\em A emphasized line. 全角は太字です。}
\underline{underlined text} \underline{架線付き文字です。}
```

**Bold type style.** 全角は太字です。 Sans serif type style. 全角は同じです。 *Slanted type style.* 全角は同じです。 Typewriter type style. 全角は同じです。 SMALL CAPS TYPE STYLE. 全角は同じです。 *A emphasized line.* 全角は太字です。 underlined text 架線付き文字です。

最後の下線の例はついでのおマケです。これはコマンドなのでちょっと形式が異なる所にご注意下さい。

### 6.3.12 行の離し方

段落の前後を離したいときや表とその説明の部分を空けたいときは、次のようなコマンド、`\smallskip`、`\medskip`、`\bigskip`を使用します。実際の間隔はページのレイアウトに左右されますので、同じコマンドでもいつも同じ間隔が空くとは限りません。



---

`\begin{flushright}`  
ここに配置しているものは、右に寄せられます。

ここに配置しているものは、右に寄せられます。  
右寄り

右寄り  
`\end{flushright}`

### 6.3.15 引用

`quotation` や `quote` は文章をまとめて引き下げることを行ないます。この環境で囲んだ部分は、前の行よりも字下げし、かつ行の長さも短くなります。前者と後者との差は段落の最初を字下げするかしないかです。

---

この例題は `quote` がどのようにインデントするかを示すものです。

```
\begin{quotation}
ここから、一段インデントしています。
この環境にいる間はズーッとインデントされています。

```

```
\begin{quote}
ここからまた、インデントされています。
通常環境は入れ子にしても大丈夫です。ただし、後に述べる verbatim だけは入れ子にできません。

```

```
\end{quote}
\end{quotation}
```

この例題は `quote` がどのようにインデントするかを示すものです。

ここから、一段インデントしています。この環境にいる間はズーッとインデントされています。

ここからまた、インデントされています。通常環境は入れ子にしても大丈夫です。ただし、後に述べる `verbatim` だけは入れ子にできません。

### 6.3.16 箇条書

箇条書の書式として  $\text{\LaTeX}$  では3種類の方法があります。先頭に●をつけるもの、先頭に番号を振るものと、先頭にラベルを指定できるものです。それぞれ、`itemize`、`enumerate`、`description` です。

---

```
\begin{itemize}
\item 各々の行はあたまたにポチを付けられてならんでい
\item 書き出しは直接後ろからでなくても大丈夫です。
\begin{enumerate}
\item この行の頭には数字がついているはずで

```

```
\item ほらね、2行目でしょ。
\end{enumerate}

```

もちろん離れていてもきちんと対処してくれます。

```
\item 空白行は何の影響も与えません。
\begin{description}
\item [ラベル] この行の頭にはラベルがついてい
\item [Label] ラベルは bold 体になっています。

```

```
\end{description}
\end{itemize}
```

- 各々の行はあたまたにポチを付けられてならんでい
- 書き出しは直接後ろからでなくても大丈夫です。

1. この行の頭には数字がついているはず
2. ほらね、2行目でしょ。

もちろん離れていてもきちんと対処してくれます。

- 空白行は何の影響も与えません。

ラベル この行の頭にはラベルがついているはず

Label ラベルは bold 体になっています。

### 6.3.17 そのままの形

`verbatim` はテキストを入力したそのままの形で表すための環境です。例えば、`%\}` はそのまま打っても  $\text{\LaTeX}$  では正しく表示してくれません。C のプログラムのリストをそのまま引用するような場合、この環境を使用します。

---

<pre>\begin{verbatim} 次のキャラクターは通常、特殊 文字として扱われるので表示できない。バックスラ ッシュを使うかverbatimを使うかどちらかである。 \\ \ \ ??? ~~~ \$\$\$\$ ### \end{verbatim}</pre>	<p>次のキャラクターは通常、特殊文字として扱われるので表示できない。バックスラッシュを使うか <code>verbatim</code> を使うかどちらかである。</p> <pre>\\ \ \ ??? ~~~ \$\$\$\$ ###</pre>
---	---

行中にこれらの特殊文字を組み込みたいときは、`\verb` を使用します。`\verb+??##$$+` とすると、`??##$$` と出力されます。

### 6.3.18 要約

タイトルページに要約に付ける時はこの環境を使います。`\maketitle` コマンドより前にこれを使う必要があります。

---

<pre>\begin{abstract} This article is for the beginner. This abstract is printed onto top page.  \end{abstract}</pre>	<p style="text-align: center;"><b>Abstract</b></p> <p style="text-align: center;">This article is for the beginner. This abstract is printed onto top page.</p>
---	---

### 6.3.19 表

$\text{\LaTeX}$  では、表のようなものを作るのに、`tabbing`、`tabular`、`array` という環境が利用できます。それぞれ想定された用途があり、特徴がありますがここでは `tabular` のみ説明をします。

`tabular` の引数の中での命令の意味は、`l` は左詰め、`r` は右詰め、`c` は中央に各項目をそろえます。`p{wd}` は列の幅を `wd` に指定 (たとえば `p{1cm}`) することにより、固定することができます。

| をつけると垂直方向の線が書けます。`\\` の後の `\hline` は水平方向の線を引きます。`\cline{i-j}` は `i` 列から `j` 列までの水平方向の線を引きます。

---

<pre>\begin{tabular}{  l c r  } \hline sra &amp; sra-b1-net &amp; 12098 packets \\ &amp; \cline{2-3} &amp; sra-2-net &amp; 290873 packets \\ \hline ntt &amp; sun-loop-back &amp; 2839 packets \\ &amp; \cline{1-1} \cline{3-3} et1 &amp; &amp; 287 packets \\ \hline titech &amp; titech-a1-net &amp; 28329 packets \\ \hline \end{tabular}</pre>	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td rowspan="2">sra</td> <td>sra-b1-net</td> <td>12098 packets</td> </tr> <tr> <td>sra-2-net</td> <td>290873 packets</td> </tr> <tr> <td>ntt</td> <td>sun-loop-back</td> <td>2839 packets</td> </tr> <tr> <td>et1</td> <td></td> <td>287 packets</td> </tr> <tr> <td>titech</td> <td>titech-a1-net</td> <td>28329 packets</td> </tr> </table>	sra	sra-b1-net	12098 packets	sra-2-net	290873 packets	ntt	sun-loop-back	2839 packets	et1		287 packets	titech	titech-a1-net	28329 packets
sra	sra-b1-net		12098 packets												
	sra-2-net	290873 packets													
ntt	sun-loop-back	2839 packets													
et1		287 packets													
titech	titech-a1-net	28329 packets													



一つ一つの項目欄は `\multicolumn{n}{pos}{item}` によって調整することができます。n は使用する列数、pos は項目の水平方向の位置、item は項目の文字列です。引数である pos は現在すでに設定されているものを無効とし新たに設定します。この時有効となる引数 pos は l r c | p からなるものでなくてはなりません。

```
\begin{tabular}{|l|l|r|} \hline \hline
{\em team } &
\multicolumn{2}{c|}{\em driver} \\ \hline
マクラーレン & セナ & プロスト \\
ロータス & ピケ & 中島 \\ \hline \hline
\end{tabular}
```

<i>team</i>	<i>driver</i>	
マクラーレン	セナ	プロスト
ロータス	ピケ	中島

### 6.3.20 数式

数学モード (math mode) は数式を表現するために使用するモードです。T<sub>E</sub>X は元もと数学の本を美しく作るために生まれたために、数式の表現は得意です。しかし、一般的には数式を必要とすることは少ないのでここでは簡単な例を示すだけにします。

文中とは独立して数式を示したいときには、`\begin{math} ... \end{math}` を使います。文中に数式を入れる場合には `$...$` を使います。例えば、`$ \int \sin(2x+3)\cos(x)^{2} dx $` と記述すると、 $\int \sin(2x+3)\cos(x)^2 dx$  のように出力されます。

```
\begin{math}
\int \frac{x \cos(x)}{\sin(x)^2} dx
\end{math}
```

$$\int \frac{x \cos(x)}{\sin(x)^2} dx$$

数式中での英字は *xy* というように斜体になります。一方関数名などは例えば、log は通常の字体でなければなりません。そこで `\log` を使用します。例えば、`$\log x^y = y \log x$` とすれば  $\log x^y = y \log x$  となります。

$\cos(x)$	<code>\cos(x)</code>	$\exp(x)$	<code>\exp(x)</code>
$\gcd(x,y)$	<code>\gcd(x,y)</code>	$\log(x)$	<code>\log(x)</code>
$\max(x,y)$	<code>\max(x,y)</code>	$\min(x,y)$	<code>\min(x,y)</code>
$\sin(x)$	<code>\sin(x)</code>	$\tan(x)$	<code>\tan(x)</code>
$A^b$	<code>\$A^{b}\$</code>	$A_b$	<code>\$A_{b}\$</code>
$A_b^c$	<code>\$A_{b}^{c}\$</code>	$\frac{x}{y}$	<code>\(\frac{x}{y}\)</code>
$\sqrt{A}$	<code>\(\sqrt{A}\)</code>	$\sqrt[b]{A}$	<code>\(\sqrt[b]{A}\)</code>

その他に数学の記号として次のようなものも利用できます。

$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$\times$	<code>\times</code>	$\div$	<code>\div</code>
$*$	<code>\ast</code>	$\star$	<code>\star</code>	$\leq$	<code>\leq</code>	$\ll$	<code>\ll</code>
$\geq$	<code>\geq</code>	$\gg$	<code>\gg</code>	$\sim$	<code>\sim</code>	$\approx$	<code>\approx</code>
$\neq$	<code>\neq</code>						

### 6.3.21 様々な記号

文中では以下の様な記号を利用する事ができます。なお、これらも数学の記号の扱いなので、通常の文中では `\leftarrow` のように \$マークで囲む必要があります。

$\leftarrow$ <code>\leftarrow</code>	$\Leftarrow$ <code>\Leftarrow</code>	$\rightarrow$ <code>\rightarrow</code>
$\Rightarrow$ <code>\Rightarrow</code>	$\leftrightharpoonrightarrow$ <code>\leftrightharpoonrightarrow</code>	$\Leftrightarrow$ <code>\Leftrightarrow</code>
$\rightharpoonrightarrowdown$ <code>\rightharpoonrightarrowdown</code>	$\leadsto$ <code>\leadsto</code>	$\Uparrow$ <code>\Uparrow</code>
$\Uparrow$ <code>\Uparrow</code>	$\Downarrow$ <code>\Downarrow</code>	$\Downarrow$ <code>\Downarrow</code>
$\Updownarrow$ <code>\Updownarrow</code>	$\Updownarrow$ <code>\Updownarrow</code>	$\nearrow$ <code>\nearrow</code>
$\searrow$ <code>\searrow</code>	$\swarrow$ <code>\swarrow</code>	$\nwarrow$ <code>\nwarrow</code>
$\flat$ <code>\flat</code>	$\natural$ <code>\natural</code>	$\sharp$ <code>\sharp</code>
$\backslash$ <code>\backslash</code>	$\infty$ <code>\infty</code>	$\Box$ <code>\Box</code>
$\Diamond$ <code>\Diamond</code>	$\triangle$ <code>\triangle</code>	$\clubsuit$ <code>\clubsuit</code>
$\diamondsuit$ <code>\diamondsuit</code>	$\heartsuit$ <code>\heartsuit</code>	$\spadesuit$ <code>\spadesuit</code>
$\alpha$ <code>\alpha</code>	$\beta$ <code>\beta</code>	$\gamma$ <code>\gamma</code>
$\dagger$ <code>\dagger</code>	$\S$ <code>\S</code>	$\copyright$ <code>\copyright</code>
$\ddagger$ <code>\ddagger</code>	$\P$ <code>\P</code>	$\pounds$ <code>\pounds</code>

ちょっと特殊な記号?として`\LaTeX`、`\TeX`、`\ldots` や`\today`があります。これらはそれぞれ、`LATEX`、`TEX`、... や平成7年 1月 7日になります。

## 6.4 演習問題

1. 次のような文章を入力し、コンパイルし、プリンターに出力してみよ。なお、「ひろのぶ」の部分は自分の学籍番号と名前に直すこと。

```
\documentstyle{jarticle}
\title{SF 桃太郎}
\author{ひろのぶ}
\begin{document}
\maketitle
```

昔むかし、おじいさんと、おばあさんが住んでいました。

おじいさんは、クレーターヘリチューム結晶を掘りにいきました。  
おばあさんは、氷河に洗濯をするための水源となる氷をとりにいきました。

すると、空から氷河に救助用小型カプセルがおりてきました。  
表面は焼けただれ、一部は解けてさえいました。

おばあさんはそれを家に持ち帰り、おじいさんと二人で分解しました。  
すると、カプセルのなかから生まれてすぐの男の子が現れました。  
`\end{document}`

2. 「SF 桃太郎」を章分けして続きを追加せよ。前の課題の部分を「はじまり」として、「桃太郎の成長」、「桃太郎の旅立ち」と言う章を追加せよ。追加した章の内容は適当に作成せよ。全体として1ページ程度にすること。