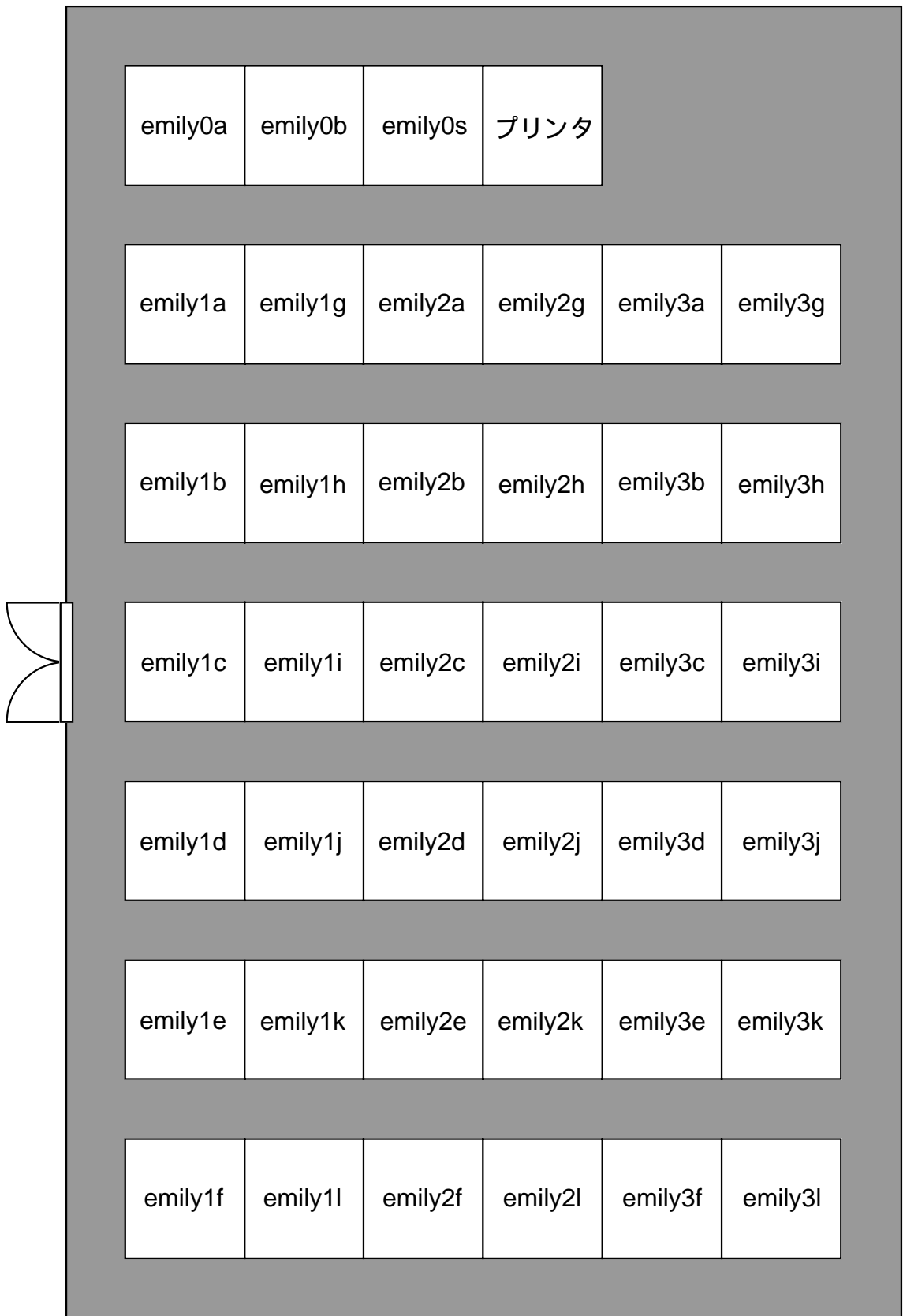
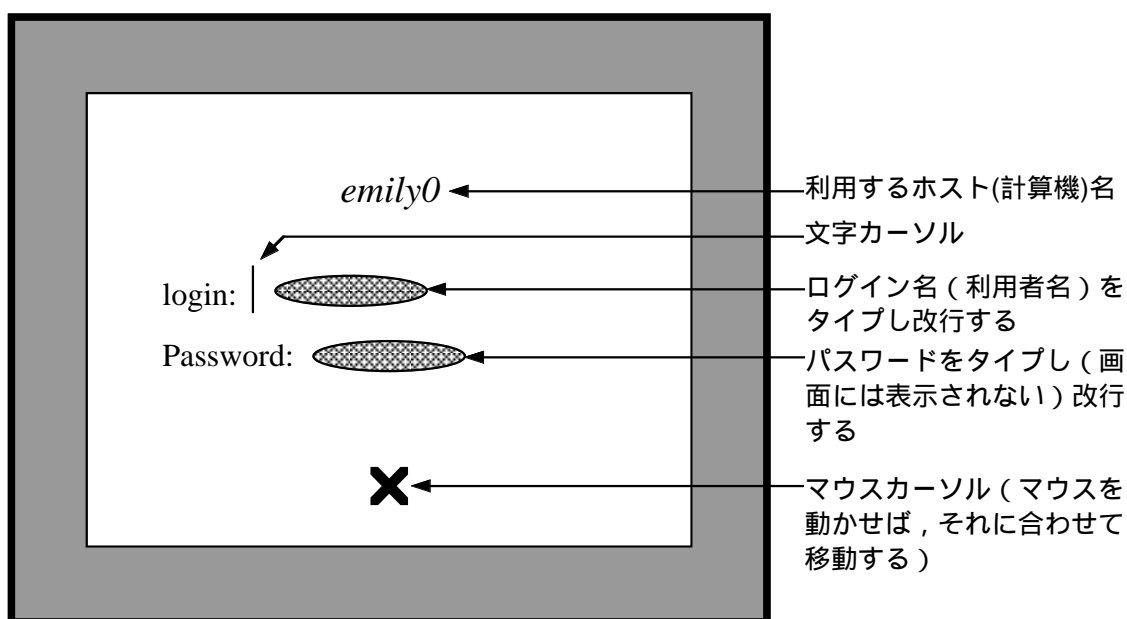


経済計測研究所 207 室配置図

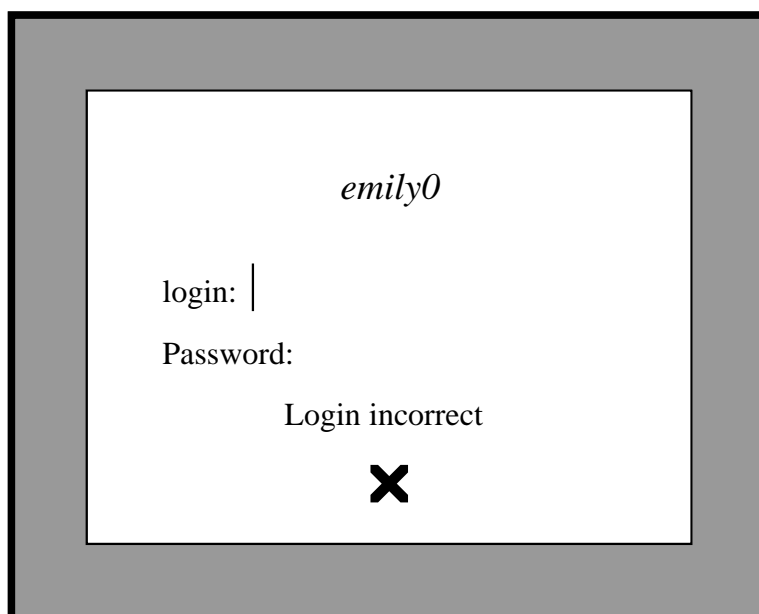


ログイン（利用開始手続き）



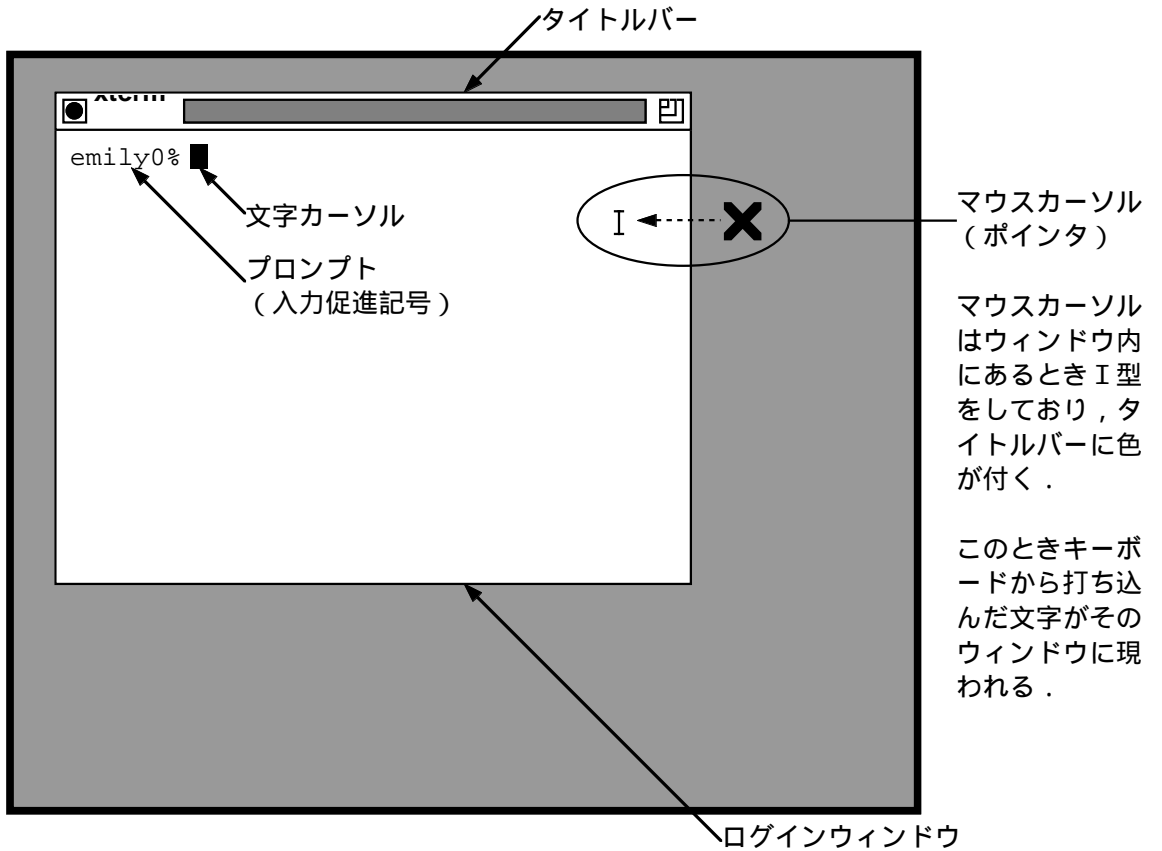
ログイン画面

- (1) login: のところに文字カーソルがあることを確かめる。
Password: のところにあれば、一度改行キーを押す。
Login: のところで文字をタイプしても現われないときは、一度「英数」キーを押してみる。
- (2) 割り当てられたログイン名(利用者名)をキーボードでタイプする。
打ち間違えたら Bs キーか DEL キーで修正する。
- (3) キーボードの改行キーを押すと、文字カーソルがPassword: のところに移る。
- (4) パスワードをキーボードでタイプする。
このときタイプした文字は他人に見られないように画面には表示されない。
- (5) 改行キーを押す。

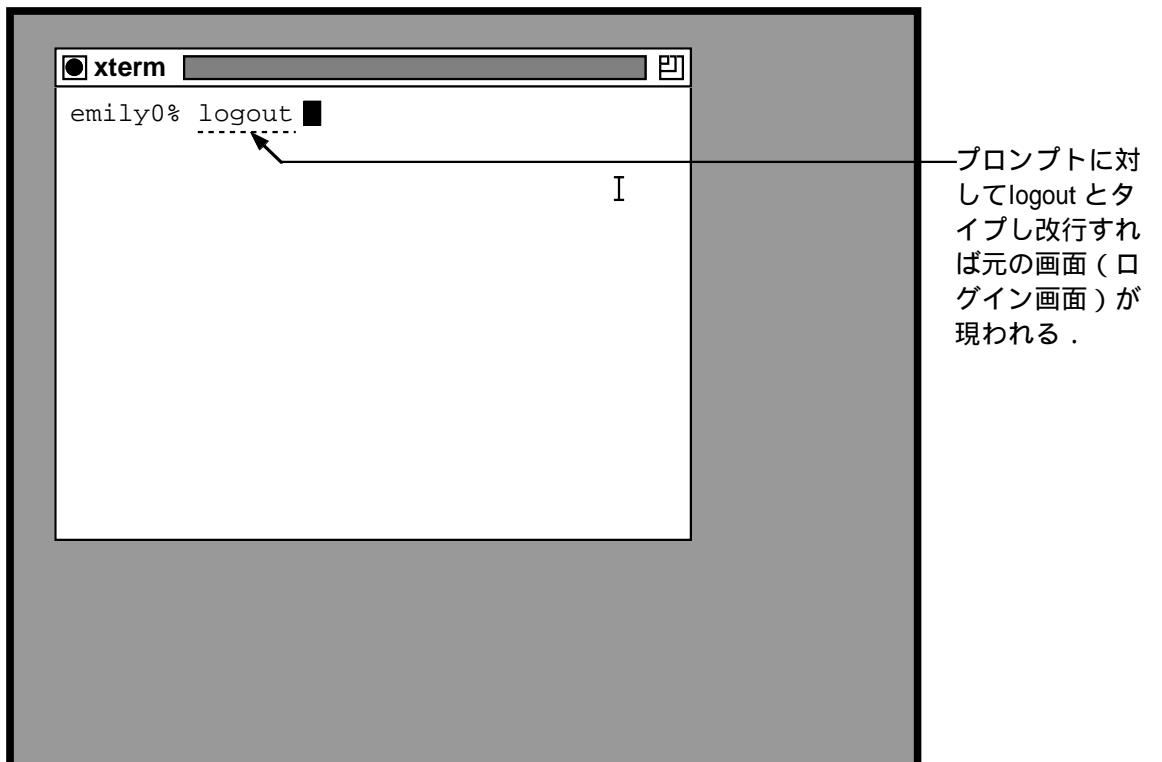


ログイン失敗(もう一度)

ログイン成功



ログアウト (利用終了手続き)



キーボード

Esc	1	2	3	4	5	6	7	8	9	0	-	=	Backspace
Tab	Q	W	E	R	T	Y	U	I	O	P	[]	\
CapsLock	A	S	D	F	G	H	J	K	L	;	'	`	Enter
Shift	Z	X	C	V	B	N	M	,	.	/			Shift
Ctrl		Alt									Alt		Ctrl

修飾キーの使い方

- 「かな」キーを一度押せば、それ以降はかなが入力される（通常使わない）
- 「英数」キーを一度押せば、それ以降は英数字が入力される

SHIFTを押しながらタイプしたとき

通常（「英数」キーを押したあと）

SHIFTを押しながらタイプしたとき

通常（「英数」キーを押したあと）
は小文字が入る

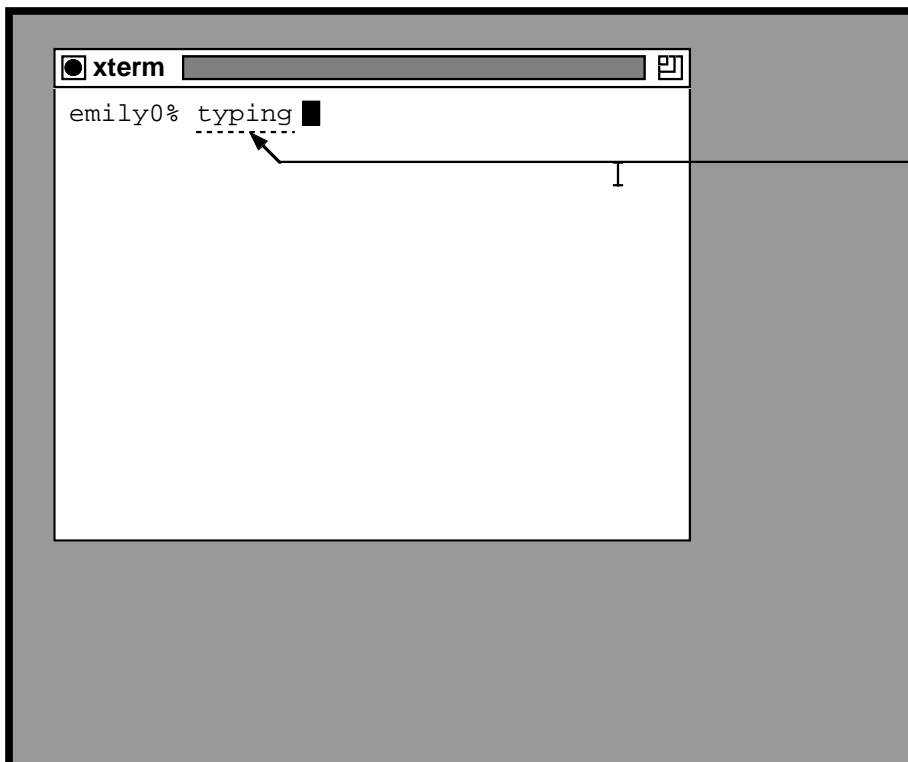
)	を
0	わ

「かな」キーを押したあと、SHIFTキーを押しながらタイプしたとき
「かな」キーを押したあと

A	
	ち

「かな」キーを押したあと

タイピングの練習

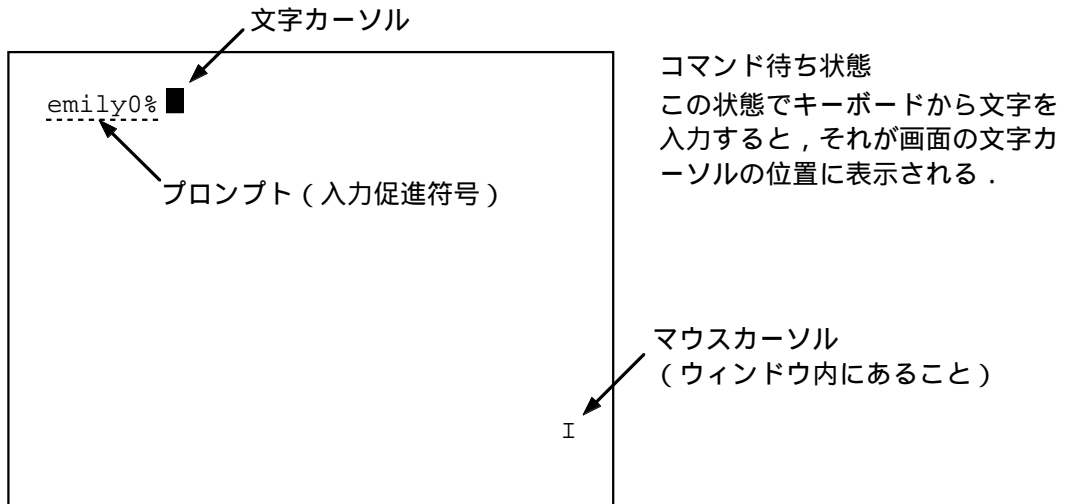


プロンプトに対して typing とタイプし改行する

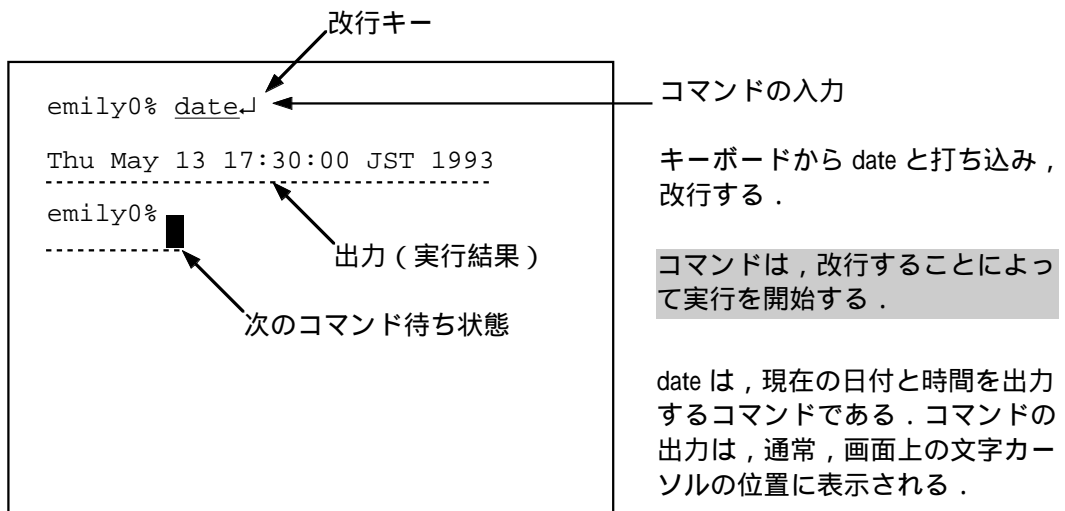
タイピング練習プログラムが起動するので、メッセージに従って操作する。

中断するときは Ctrl-C (Ctrl キーを押しながらCのキーを押す) をタイプする。

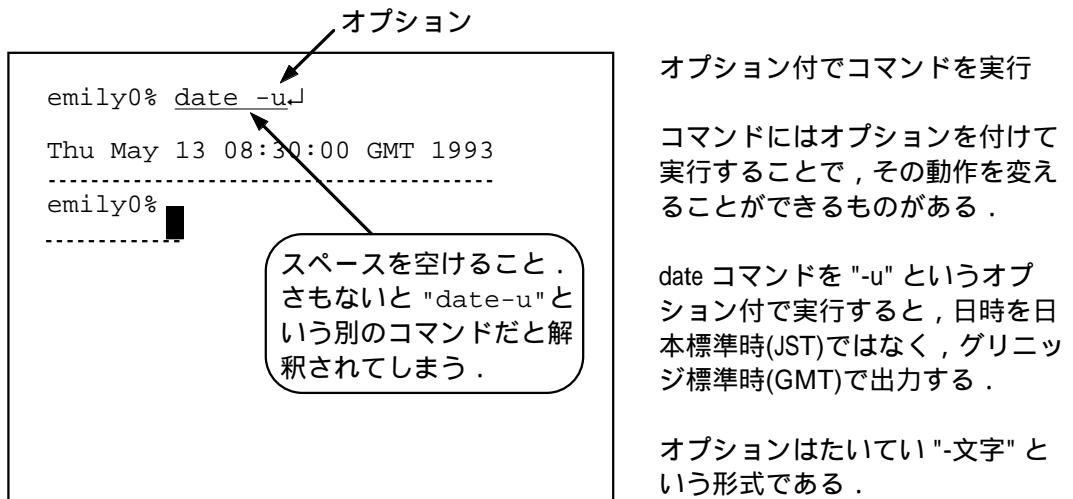
コマンド～コンピュータに対する指令



コマンドの実行



コマンドのオプション



コマンドの引き数

```
emily0% cal ↵
usage: cal [month] year

emily0% cal 7 1961 ↵
      July 1961
  S  M Tu  W Th  F  S
                1
  2  3  4  5  6  7  8
  9 10 11 12 13 14 15
 16 17 18 19 20 21 22
 23 24 25 26 27 28 29
 30 31
emily0%
```

引き数なしで実行

使い方を表示

"7 1961" という引き数付で実行

1961年7月のカレンダー

引き数付でコマンドを実行

コマンドには引き数を付けることができるものがある。引き数はコマンドの実行に必要な情報をコマンドに与えるときに使う。

calコマンドは、引き数なしで実行すると、引き数の与え方を表示する ([month] は月を指定する1つ目の引き数が省略できることを示している)。引き数として年や月を与えれば、その年や月のカレンダーを表示する。

オプションと引き数

オプションはコマンドに「作り込み」されていて、用意されたもの以外は受け付けないのに対し、引き数はコマンドに任意の情報を与えるのに用いられる。

課題(1) 自分の誕生月のカレンダーを表示し、自分の誕生日が何曜日だったのかを調べよ。

課題(2) `cal 1993` のように、引き数に月を指定しなかった場合は何が表示されるか。

コマンドへのデータ入力

```
emily0% bc ↵
2+3 ↵
5
Ctrl-D
emily0%
```

bc コマンドの実行

この状態で止まってしまう (データ入力待ちの状態)

ここに状態で数式を入力し改行すると、その計算結果が出力される。

でもまた止まってしまう (次のデータ入力待ち)

コマンドの実行を終了する場合は、ここで Ctrl-D (Ctrl キーを押しながら D のキーを押す) をタイプする。

次のコマンド待ちの状態になる。

データ入力が必要なコマンド

コマンドのなかには実行後にキーボードからのデータ入力を必要とするものがある。

コマンドは入力されたデータを加工し、結果を出力する。結果は通常画面に表示される。

コマンドの実行を終了するには Ctrl-D (Ctrl キーを押しながら D のキーを押す) をタイプする。これはデータ入力の終りを意味する。

bc コマンドは入力データとして与えられた数式を計算して、結果を出力する電卓コマンドである。

コマンドの出力の保存

```
emily0% date > a↵  
(何も表示されない)
```

date コマンドを実行し、出力を a というファイルに保存する。

```
emily0% cal 5 1993 > b↵  
(何も表示されない)
```

cal コマンドを実行し、出力を b というファイルに保存する。

```
emily0% ls↵  
a b
```

a, b という2つのファイルができている。ls は現在所有しているファイルの名前をリストアップするコマンド。

```
emily0% cat a↵  
Thu May 13 17:30:00 JST 1993
```

a というファイルの内容を出力。

```
emily0% cat b↵  
    May 1993  
S  M Tu  W Th  F  S  
                1  
2  3  4  5  6  7  8  
9 10 11 12 13 14 15  
16 17 18 19 20 21 22  
23 24 25 26 27 28 29  
30 31
```

b というファイルの内容を出力。cat は引き数に指定したファイルの内容を出力するコマンド。

">" という記号の役割

コマンドの出力を、画面に表示する代わりにファイルに保存する。この操作を標準出力のリダイレクトという。

課題 (3) "cat a b" のように複数のファイルを引き数に指定して cat を実行したらどうなるか。

```
emily0% cat a > b↵
```

ファイル a の内容を画面に出力する代わりにファイル b に入れる。

```
emily0% cat a↵  
Thu May 13 17:30:00 JST 1993  
emily0% cat b↵  
Thu May 13 17:30:00 JST 1993
```

それまでのファイル b の内容 (カレンダー) は失われ、代わりにファイル a と同じ内容が入っている。

```
emily0% date >> b↵
```

">" の代わりに ">>" を使う。

```
emily0% cat b↵  
Thu May 13 17:30:00 JST 1993  
Thu May 13 18:07:25 JST 1993
```

それまでの内容の後ろに新しい内容が追加されている。

ファイルからの入力

```
emily0% cat↵
satou↵
satou
suzuki↵
suzuki
tanaka↵
tanaka
Ctrl-D ← Ctrl-Dを入力して、データの入力を終了する。
```

引き数なしでcatコマンドを実行すると、bc同様データ入力待ちになる。適当な文字をタイプすれば、改行を押す度にそれがオウム返しに画面に出力される。

```
emily0% cat > c↵
2+3↵
4*5+7↵
(3+4)*29↵
Ctrl-D
```

入力データとして数式を入れてみる。今度は改行を押してもデータは画面に表示されない(ファイルcに保存されている)。

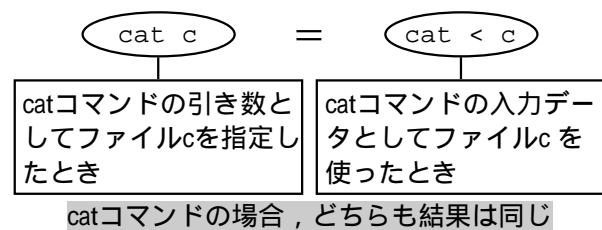
同様にcatを引き数なしで実行し、今度はその出力をファイルcに保存する。

```
emily0% cat < c↵
2+3
4*5+7
(3+4)*29
```

ファイルcの内容をcatコマンドの入力データに使う。ファイルcの内容がそのまま表示される。

```
emily0% bc < c↵
5
27
203
```

ファイルcの内容をbcコマンドの入力データに使う。ファイルcの内容の計算結果が出力される。



"<" という記号の役割

コマンドへのデータ入力をキーボードをタイプする代わりにファイルから行う。この操作を標準入力のリダイレクトという。

```
emily0% cat c | bc↵
5
27
203
```

2つのコマンドが同時に実行される

このようにしても同じ結果が得られる。この場合はcatコマンドの出力をそのままbcコマンドの入力に使っていることになる。これをパイプという。

課題(4) catコマンドを使って、yoshida, satou, suzuki, tanaka, aoyama という内容のファイル meibo を作り(1行に一人の名前を書くこと)、そのファイルを sort というコマンドの入力データとして与えると何が表示されるか。同様に、meibo を wc というコマンドの入力データとして使った場合は何が表示されるか。

コマンドの使い方を調べる ~ オンラインマニュアル

あるコマンドが何をするコマンドか調べる ~ `whatis`

```
emily0% whatis wc↵  
(wc コマンドの簡単な説明)  
emily0%
```

コマンドの使い方を調べる ~ `man`

```
emily0% man wc↵  
:  
(wcコマンドの使い方が表示される)  
:  
--MORE--
```

解説が画面に収まり切らない場合は[--MORE--]と表示して表示が一旦停止する。

- ・スペースキー 次の画面
- ・b キー 前の画面
- ・q キー 解説表示の打ち切り
- ・? キー 操作キーの説明表示

あるキーワードに関連したコマンドの一覧を表示する ~ `apropos`

```
emily0% apropos game↵  
( "game" というキーワードに関連するコマンド  
の一覧をwhatisと同じ形式で表示する )  
emily0%
```

課題 (5) `wc` コマンドの使い方を `man` コマンドを用いて調べ、課題 (4) で `wc` を使ったときに出力されたものが何を意味していたのか述べよ。

課題 (6) `ls` コマンドに `-l` (`L` の小文字) オプションを付けて実行したときに出力されるものは何か。また `ls` コマンドは通常ファイル名をアルファベット順に出力するが、これを逆順に出力するためには `ls` コマンドにどのようなオプションを付けて実行すればよいか。

パスワードの変更

```
emily0% yppasswd↵  
Changing NIS password for .. (略)  
Old Password: _____↵  
New Password: _____↵  
Retype New Password: _____↵  
emily0%
```

(パスワードは画面には表示されない)

古いパスワード (ログインした時に使ったもの) をタイプする。

新しいパスワードをタイプする。
確認のために2回タイプする。

ログインの時同様パスワードが他人に見られないよう、タイプしても画面には表示されない。

パスワードの変更には通常 `passwd` コマンドを用いるが、このシステムではパスワードの管理に `NIS` (Network Information System) と呼ばれる機構を用いているため、代わりに `yppasswd` コマンドを用いる。

電子メール～他の人にメッセージを送る

メールの送信

```
emily0% mail 相手のログイン名↵
How are you?↵
I'm fine.↵
Bye bye.↵
Ctrl-D ← 文章を書いたら、最後にCtrl-Dをタイプして送信
```

メッセージを送りたい相手のログイン名を引き数にして、mail コマンドを実行する。

```
emily0% cat > tegami↵
kono jugyou owattara,↵
ocha wo nomini ikou.↵
nanpa otoko yori.↵
Ctrl-D
emily0% mail 相手のログイン名 < tegami↵
```

catを使って、メッセージの本文をあらかじめファイルに保存する。

bcコマンドなどと同様、ファイルに保存されている内容をmail コマンドの入力データに使うこともできる。

場合によっては "Subject:" という表示が現われることがある。その場合はそこにメッセージのタイトルをタイプし改行したあと、その次の行から本文をタイプすればよい。

課題(7) 自分宛(自分のログイン名宛)にメッセージを送ってみよ。また自分の友人宛になにかメッセージを送ってみよ。

課題(8) 課題(1)～(6)の解答を担当教官のログイン名宛に電子メールで送れ。

メールの受信

```
emily0% mail↵
No mail. ← メッセージが届いていないとき。
emily0%
:
You have new mail. ← メッセージが届くと知らせてくれる。
emily0% mail↵
Mail version ..... (略)
"/usr/spool/mail/ログイン名
>N 1 e45kXXX Thu May 20 18:30 ..
N 2 s39aXXX Thu May 20 18:33 ..
&
mailコマンドのプロンプト
...
& 1↵
(メッセージの表示)
--MORE--
& r↵ ← 返事を書く
otegami arigatou.
Ctrl-D ← mailコマンドを終了する
& q↵
emily0% mail -f mbox↵ ← 保存されたメッセージを読む
```

引き数なしでmailコマンドを実行する。

メッセージが届いているときは、その差出人のログイン名、日付、メッセージに付けられたタイトルなどを表示したあと、&というプロンプトを表示してコマンド待ちになる。

新着メッセージは2通

1番のメッセージを読む。メッセージが画面に収まり切らないときは、manコマンド同様[--MORE--]と表示して表示が止まる。スペースキーを押せば次の画面に進む。

読み終えたメッセージはmailコマンドの終了時にmboxというファイルに保存される。これはcatでも参照できるが、mailコマンドでも読むことができる。

ファイル操作

ファイルの複写 ~ cp

```
emily0% date > a↵
emily0% cp a b↵
emily0% cat a↵
Thu May 13 17:30:00 JST 1993
emily0% cat b↵
Thu May 13 17:30:00 JST 1993
```

ファイルaをファイルbにコピーする（既にファイルbが存在した場合、以前の内容は失われる）。

ファイルaとbは同じ内容。

ファイル名の変更（ファイルの移動） ~ mv

```
emily0% ls↵
a b
emily0% mv a x↵
emily0% ls↵
b x
emily0% mv b x↵
emily0% ls↵
x
```

a, b 2つのファイルがあるとする。
aをxに移動する。

aが無くなり代わりにxができる。
bをxに移動する。

bが無くなる。

- ・ bの内容はxに移る。
- ・ 以前のxの内容は失われる。

ファイルの削除 ~ rm

```
emily0% rm x↵
emily0% rm a b↵
```

ファイルxを削除する。
引き数に複数のファイルを指定してもよい。

ワイルドカード ~ 存在するファイルの一括指定

ワイルドカードに使われる文字

- * 任意の文字列にマッチする
- ? 任意の1字にマッチする
- [] この中のいずれか1字にマッチする

当然これらの文字は（そのままでは）ファイル名に使えない。

組み合わせ例

- a* 先頭の文字がaのファイル
- a*c 先頭がaで最後がcのファイル
- a?c 先頭がaで最後がcの3字のファイル
- ??? 3字のファイル
- d[135] d1, d3, d5 の3つのファイル
- [a-z]* 先頭が小文字のファイル

使用例

```
emily0% ls -l a*
```

```
emily0% cat a*.doc
```

```
emily0% rm d[135].bak
```

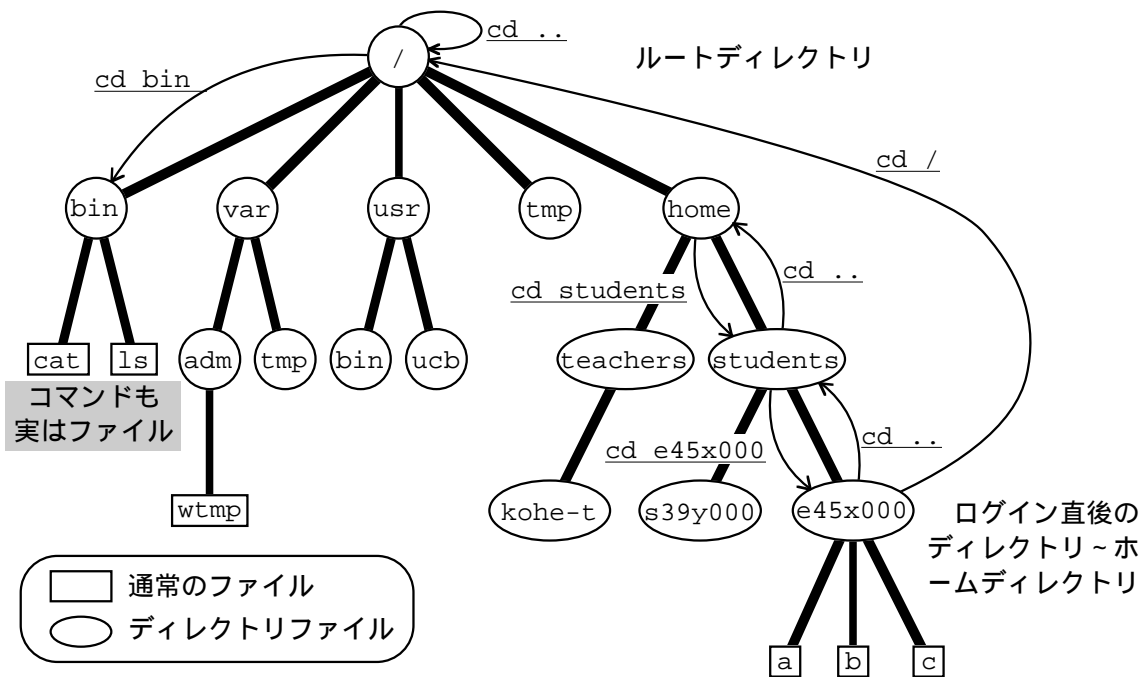
"[文字-文字]" は範囲を表す。
"[A-Za-z0-9]" のように複数指定可。

先頭の文字がaのファイルをリストアップする。

先頭の文字がaで、後ろに.docがついたファイルの内容を表示する。
d1.bak, d2.bak, d3.bak の3つのファイルを削除する。

課題(9) `echo *` を実行すると何が表示されるか。これに似た動作をする他のコマンドを一つ答えよ。

ディレクトリ～階層構造（木構造）によるファイル管理



ディレクトリの移動～cd

<pre>emily0% pwd↵ /home/students/e45x000 ← 自分のログイン名</pre>	現在いる場所（カレントディレクトリ）を調べる。
<pre>emily0% cd ..↵ emily0% pwd↵ /home/students</pre>	一つ上のディレクトリに移動する ～カレントディレクトリを移す。
<pre>emily0% cd e45x000↵ emily0% pwd↵ /home/students/e45x000</pre>	カレントディレクトリ内に存在する e45x000 というディレクトリに 移動する。
<pre>emily0% cd /↵ emily0% ls↵ a bin etc usr var vmunix ... (他)</pre>	ルートディレクトリに移動する。 ルートディレクトリにあるファイル をリストアップする。
<pre>emily0% ls -F↵ a/ bin/ etc/ usr/ var/ vmunix* ...</pre>	ls に -F というオプションを付けると、 ディレクトリファイルには /、 コマンドには * を付けて表示する。
<pre>emily0% cd bin↵ emily0% ls -F↵ cat* ls* ... (他いっぱい)</pre>	カレントディレクトリ内に存在する bin というディレクトリに移動する。
<pre>emily0% cd /usr/ucb↵ emily0% ls↵ rsh rwho ...</pre>	直接 /usr/ucb に移動する。
<pre>emily0% cd↵ emily0% pwd↵ /home/students/e45x000</pre>	cd を引き数なしで実行したときは ホームディレクトリに戻る。

ディレクトリの作成

ディレクトリの作成 ~ mkdir

```
emily0% ls↵
a b c meibo
emily0% mkdir d↵
emily0% ls
a b c d meibo
emily0% ls -F↵
a b c d/ meibo
```

これらのファイルが見つからないときは、`touch a b c meibo` を実行

d というディレクトリを作成する .

d というファイルができています . ls に -F オプションを付ければそれがディレクトリであることがわかる .

```
emily0% ls a↵
a
emily0% ls d↵
( なにも表示されない )
```

ls の引き数に普通のファイルを指定した場合はそれ自身が出力されるが、ディレクトリを指定した場合はそのディレクトリの中のファイルの一覧を出力する .

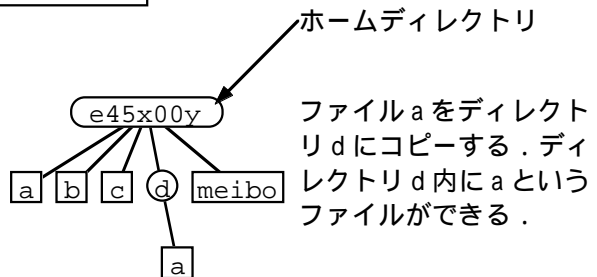
```
emily0% cd d↵
emily0% ls↵
( なにも表示されない )
emily0% cd ..↵
emily0% ls↵
a b c d meibo
```

d は作ったばかりなので、中には何もありません .

d に cd して確かめてみる .

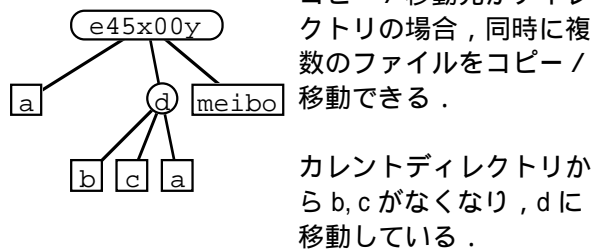
ディレクトリへの cp / mv

```
emily0% cp a d↵
emily0% ls d↵
a
```



ファイル a をディレクトリ d にコピーする . ディレクトリ d 内に a というファイルができる .

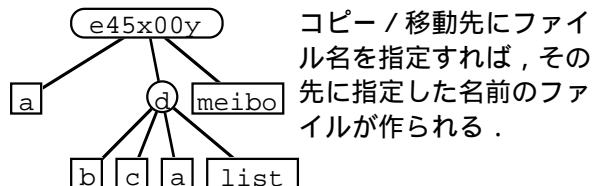
```
emily0% mv b c d↵
emily0% ls↵
a d meibo
emily0% ls d↵
a b c
```



コピー / 移動先がディレクトリの場合、同時に複数のファイルをコピー / 移動できる .

カレントディレクトリから b, c がなくなり、d に移動している .

```
emily0% cp meibo d/list↵
emily0% ls d↵
a b c list
```



コピー / 移動先にファイル名を指定すれば、その先に指定した名前のファイルが作られる .

ディレクトリの削除

ディレクトリの削除 ~ rmdir

```
emily0% rmdir d↵
rmdir: d: Directory not empty
emily0% ls d↵
a b c list
```

ディレクトリ d を削除する .

d が空でないので削除できない .

```
emily0% cd d↵

emily0% rm a list↵

emily0% mv b c ..↵

emily0% cd ..↵

emily0% ls d↵
(何も表示されない)
emily0% rmdir d↵
```

ディレクトリ d に移る .

ファイル a 及び list を削除する .
ファイル b 及び c を一つ上のディレクトリ (..) に移動する .

一つ上のディレクトリ (ホームディレクトリ) に戻る .

ディレクトリ d は空 .
削除成功 .

ディレクトリ単位の複写 / 移動

ディレクトリ単位の複写

```
emily0% mkdir d↵
emily0% cp a b c d↵

emily0% cp -r d e↵

emily0% ls d e↵
d:
a b c
e:
a b c
```

ディレクトリ d を作り , そこにファイル a, b, c をコピーする .

ディレクトリ d 全体を e というディレクトリにコピーする . ディレクトリ全体をコピーするときは cp に -r オプションを付ける .

2つのディレクトリ d, e の内容は同じ .

ディレクトリ単位の移動

```
emily0% ls -F↵
a b c d/ e/ meibo

emily0% mv e f↵
emily0% ls↵
a b c d f meibo

emily0% mv f d↵

emily0% ls↵
a b c d meibo
emily0% ls d↵
a b c f
```

ホームディレクトリには a, b, c, d, e, meibo の 6 つのファイルがありこのうち d, e はディレクトリ .

ディレクトリ e を f に移動する .
ディレクトリ名が変わっただけ .

ディレクトリ f を d に移動する .

既に d が存在するので , f はその中に入る .

エディタ Emacs ~ 文書を作る

```
emily0% emacs ↵
```

引き数なしで emacs を起動する .

```
GNU Emacs 18.55.2 of Wed Feb 10 1993 on wsre111 (berkeley-unix)
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/unsupported/lib/emacs/etc/NEMACS.???_

[-----]NEmacs: *scratch* (ESS: Lisp Interaction)~All-----
Loading sj3-egg...done
```

起動後解説が表示される . ヘルプ機能の使い方や , チュートリアルの実行方法などが書かれている . このメッセージは何かキーをタイプすると消える .

チュートリアルの起動方法

- (1) DEL キーをタイプした後 ,
- (2) シフトキーを押しながら T をタイプする .

emacs のキー操作の表記

C-x ----- Ctrl を押しながらか x をタイプする .

C-x C-c · C-x をタイプした後 , C-c をタイプする .

M-x ----- Esc キーをタイプした後 , x をタイプする .

課題 (1 0) チュートリアルを読みながら指示にしたがって操作の練習をせよ .

```
=====
日本語 GNUEMACS(NEmacs) 入門編
=====
注意: この入門編は、「習うより慣れる」をモットーに作成されています。
      >> から始まる行は、その時刻をすべきかを指示しています。

NEmacsのコマンドを入力するときには、一般的にコントロール・キー（キー・
トップに、CTRL あるいは、CTL と書いてある）やメタ・キー（普通 エスケープ・
キーを使う）が使われます。そこで、CONTROL とか META とか書く代わりに、次の
ような記号を使うことにします。

C-<文字> コントロール・キーを押したまま、<文字>キーを押します。例えば、
C-f は、コントロール・キーを押しながら f のキーを押すことを
意味します。

>> それでは、C-v (View Next Screen; 次の画面を見る) をタイプして
   みて下さい。次の画面に進むことができます。
[-----]NEmacs: NEMACS.tut (ESS: Fundamental)~Top-----
C-h (Type ? for further options) T
```

このシステムでは DEL キーの役割と C-h キーの役割を入れ替えているので、
C-h キー DEL キー
DEL キー Bs キー
のように読み替えること .

間違ったキー操作により先に進めなくなったときは、

C-g
をタイプしてみよ .

emacs を終るには **C-x C-c** をタイプする .

ファイルを保存するかどうかが聞いてくるので、保存する場合はここで y をタイプする .

ここで保存される NEMACS.tut というファイルは、オンラインチュートリアルで使った文書である . 不要なら削除して構わない .

```
[あ]~**NEmacs: sample (ESS: Fundamental)~All-----
Save file /home/teachers/kohe-t/nemacs.tut(y or n) _
```

Emacs の操作

文字の入力

入力 --- そのまま文字をタイプする

C-_ --- 取り消し(undo), 直前の状態に戻す(C-x u も同じ)

Bs --- 打ち間違えた文字を消すとき

C-g --- 中止(quit), 現在の操作を止める

DEL --- ヘルプメニューの表示

文字の削除と複写

C-k --- カーソル位置から行末まで削除

C-スペース

→ 《カーソル移動》

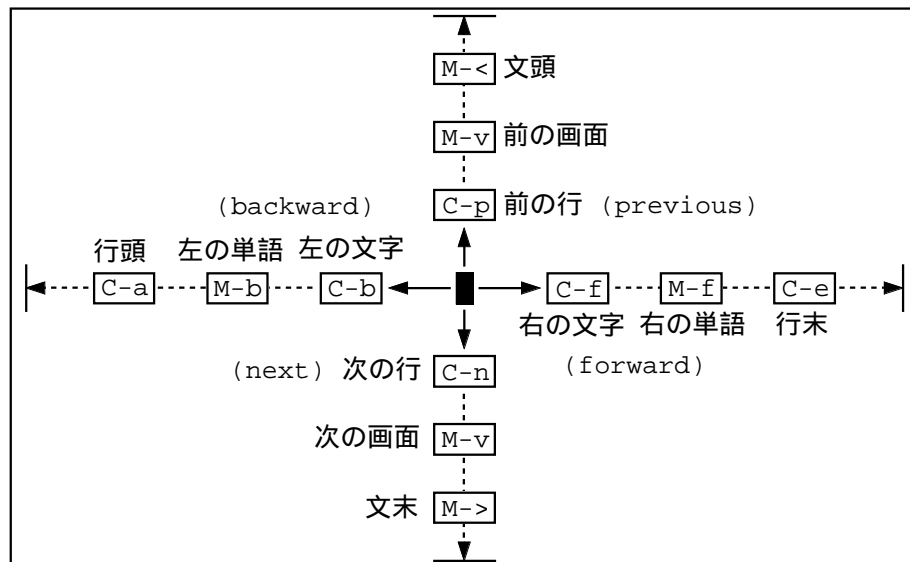
C-スペースでマークを付けたところからカーソル位置までを...

C-w --- 削除する

M-w --- 記憶する

C-y --- 削除した内容をカーソル位置に張り込む

カーソル移動



文字列の検索

文末方向に向かって探すとき **C-s** **C-r** 文頭方向に向かって探すとき

探したい文字列をタイプ

日本語を探すときはここで **Esc** をタイプし, 文字列の入力後改行する

文末方向に同じものを探すとき **C-s** **C-r** 文頭方向に同じものを探すとき

検索終了 **Esc** 検索取消 **C-g**

文字列の置換

M-% → "Query replace:" 検索する文字列をタイプ **⏎**

"with:" それを置き換える文字列をタイプ **⏎**

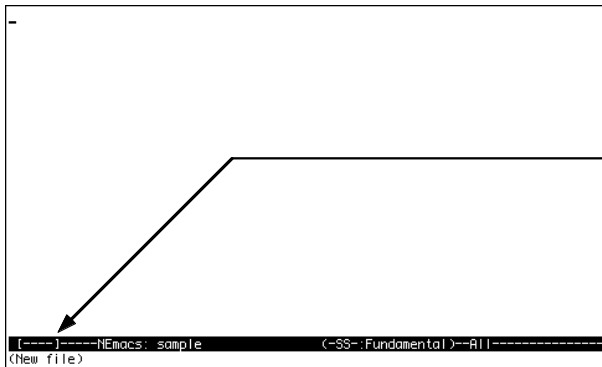
検索する文字列が見つかったとき

- 置き換える..... y (またはスペース)
- 置き換えない..... n
- 終了..... q (または **Esc**)

日本語入力～たまご

```
emily0% emacs sample ↵
```

引数にファイル名を指定する .

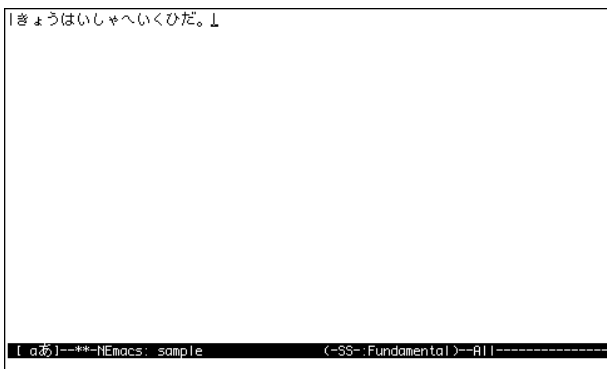


引数に指定したファイルが存在しない場合は (New file) が表示される .

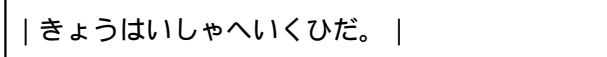
これからそのファイル (sample) の作成を行う .



`C-J` をタイプする . [aあ] が表示され、日本語入力モードになる . もう一度 `C-J` をタイプすると英字モードに戻る .



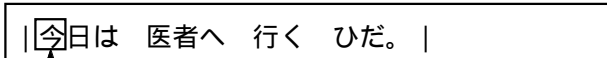
ローマ字で文章をタイプすると、それがひらがなで表示される .



入力中の日本語は |..| で挟まれている (フェンスモードという) . 変換はスペース .

↓ スペースをタイプする ↓

初めて emacs を使うときは、この後「ユーザー辞書」や「頻度ファイル」などを作成するかどうか聞いてくる . すべての質問に "yes" とタイプして改行すること .



漢字かなまじり文に変換される . 文節ごとに空白が入っている .

↑
カーソル

| 今日 歯医者へ 行く ひだ。 |

C-i をタイプして先頭の文節を縮めた場合

| 今日 歯医者へ 行く ひだ。 |

C-f を 3 回タイプした場合

| 今日 歯医者へ 行く 日だ。 |

目的の変換結果が出るまで
スペースか C-n をタイプする

今日歯医者へ行く日だ。

改行キーをタイプするか
次の文章を打ち始める

文節の長さを調節する .

C-o -- 文節を伸ばす

C-i -- 文節を縮める

変換対象の文節を移動する .

C-f -- 次の文節に進む

C-b -- 前の文節に戻る

正しい変換結果を出す .

C-n -- 次の候補を表示する

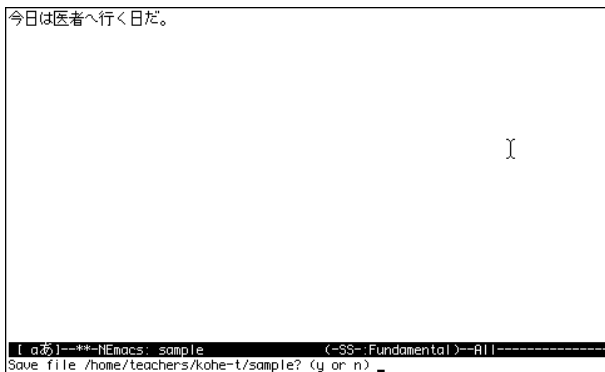
C-p -- 前の候補を表示する

M-s -- 候補の一覧を表示する

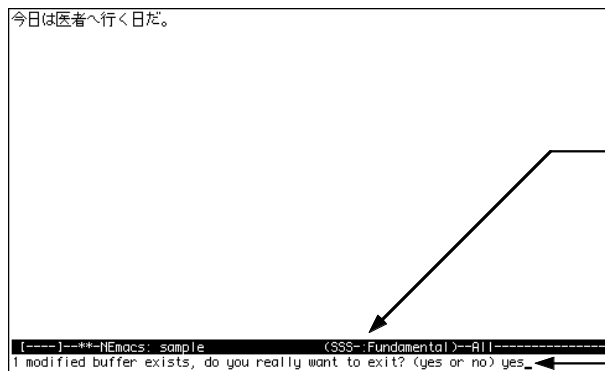
確定する .

↵ ---- 改行で確定 (次の文章
を打ち始めてもよい)

日本語入力のモードから抜け出るには、もう一度 **C-¥** をタイプする .



C-x C-c をタイプして emacs を終了する . 今回タイプした内容を保存するかどうか聞いてくる .



ファイルを保存する必要がなければ n をタイプする .

この場合は次にもう一度本当に emacs を終了してよいか確認のメッセージが出る .

ファイルを保存せずに emacs を終了するなら "yes" とタイプして改行する .

Emacs を使ってメールを読み書きする

Emacs を使ってメールを出す

```
M-x mail
```

Emacs の mail コマンドを実行 .

↓ 次の画面が現われる ↓

```
To: この部分に宛先を書く
Subject: ここはメッセージに付けるタイトル
--text follows this line--
ここから後に本文を書く . 1行の長さはアルファベットで60~70文字(漢字で30~35文字)程度にし, 1行ごとに改行すること .
```

} この部分(ヘッダ)には空行を入れてはいけない . また Subject: には日本語は使用できない .

この行には変更を加えてはいけない .

宛先はコンマ(,)で区切って複数書くことができる .

本文を書き終って, 宛先などに間違いがないかどうか確かめたら, `[C-c C-c]` をタイプする .

`[C-c C-c]` ...メールの送信

書いている途中のメッセージを破棄したいときは `[C-x k]` をタイプする .

Emacs を使ってメールを読む

```
M-x rmail
```

Emacs の rmail コマンドを実行 .

↓ 次の画面が現われる ↓

```
Date: Tue, 15 Jun 93 09:10:00 +0900
From: someone
To: me
Subject: greeting

Hello! How are you?
```

届いたメールの内容が画面に表示される .

この画面の時に, 下のキー操作が使える .

スペース	次の画面へ進む
Bs	前の画面に戻る (本当は DEL キーだが Bs キーと役割を交換している)
n	次の (削除マークのついていない) メールに進む
p	前の (削除マークのついていない) メールに戻る
M-n	次のメールに (削除マークの有無に関係なく) 進む
M-p	前のメールに (削除マークの有無に関係なく) 戻る
d	メールに削除マークをつける
u	メールの削除マークを消す
x	削除マークのついたメールを削除する
q	rmail を終る (削除マークのついたメールは削除される)
g	新しく届いたメールを取り込む
m	メールを書く (M-x mail と同じ)
r	返事を書く
h	一覧を表示する ~ 画面が2つに分割され, 一覧を見ながらメッセージの操作を行なうことができる (キー操作は少し変化する)

スペース	次の画面に進む	
Bs	前の画面に戻る	
n	次の（まだ読んでいない）記事に進む	
p	前の（まだ読んでいない）記事に戻る	
^	（フォロー記事の時）元記事を参照する	
C-n	次の記事に（読み終わっていても）進む（読むならスペース）	
C-p	前の記事に（読み終わっていても）戻る（読むならスペース）	
c	全ての記事を読んだことにする	
C	現在の記事が自分が投稿したもののとき，その記事を取り消す	
o	その記事を rmail 形式で保存する	
C-o	その記事を UNIX-mail 形式で保存する	記事を読んでいるときに q をタイプすれば，ニュースグループの一覧表示に戻る．ここでもう一度 q をタイプすれば，gnus を終ることができる．なお，この時 gnus の終了を確認するメッセージが表示されるので，y をタイプする．
u	その記事をとっておく	
M-u	その記事を読んでいないことにする	
m	メールを書く	
r	投稿者にメールを出す	
R	投稿者にメールを出す（元記事を引用する）	
a	記事を書く	
f	フォロー記事を書く	
F	フォロー記事を書く（元記事を引用する）	
q	ニュースグループ選択モードに戻る -----	

記事を投稿する

Emacs を使って記事を投稿する

```
M-x gnus-post-news ↵
```

```
6: 1: alt.technology.misc
1: alt.flame.faggots
1: wadai.talk.jokes
5: wadai.talk.misc
26: fj.ai
25: fj.binaries.mac
3: fj.binaries.misc
53: fj.binaries.msdos
35: fj.binaries.msdos.d
7: fj.binaries.x68000
126: fj.books
2: fj.comp.arch
36: fj.comp.comm
19: fj.comp.image
54: fj.comp.misc
53: fj.comp.music
5: fj.comp.oops
7: fj.comp.parallel
63: fj.comp.printers
10: fj.comp.speech
168: fj.comp.texhax
10: fj.comp.text
[-----]-- NGNUS: List of Newsgroups (C-SS--:Newsgroup (nyujin)-)~Top-----
SPC:Select n:Forward p:Backward q:Exit C-c TAB:Run Info ?:This help
```

```
Are you sure want to post to .. ? ↵
```

```
Newsgroups: wadai.test ↵
```

ニュースグループには様々なものがあるが，このうち wadai.test は投稿の練習に使用できる．初めて記事を投稿するときは，まずここで練習しておいたほうがよい．

Emacs の gnus-post-news コマンドを実行．

あるいは

ニュースグループ選択画面の時に a をタイプする．

本当に記事を投稿するのか確認のメッセージが出る．y をタイプ．

投稿するニュースグループを指定する．

ニュースグループ名は，途中までタイプしてスペースをタイプすれば，残りを補完してくれるか，候補の一覧を表示してくれる．

Subject: test

記事のタイトルを入力する。日本語（漢字）は使用不可。

Distribution: wadai

記事の配布範囲を指定する。和歌山大学内なら wadai である。

配布範囲は基本的にはそのニュースグループが属するカテゴリ（"." で区切られた最初の単語）を指定する。但し指定しなくても問題はない。

Newsgroups: wadai.test

Subject: test

Distribution: wadai

--text follows this line--

ここから後に本文を書く。1 行の長さはアルファベットで 60 ~ 70 文字（漢字で 30 ~ 35 文字）程度にし、1 行ごとに改行すること。

この部分は既に入力されている（修正してもよい）。

この行には変更を加えてはいけない。

行末を揃えるには、事前に M-x auto-fill-mode

を実行しておくか、行末を揃えたい部分にカーソルを移動し、

M-q

をタイプする。

本文を書き終って、ニュースグループ名などに間違いがないかどうか確かめたら、C-c C-c をタイプする。

書いている途中のメッセージを破棄したいときは C-x k をタイプする。

```
0 21: [ 9:wakaba@ep.ee.] Hi
22: [ 4:y-furu@fumi.e] Re: Hi
23: [ 11:wakaba@ep.ee.]
24: [ 17:Terry@edu.wak]
[-----] NGNUS: wadai.test/21 (97 more) (EEE:Summary Thread)--Top-----
From: wakaba@ep.ee.saga-u.ac.jp (Hiromi WAKABAYASHI)
Newsgroups: wadai.test
Subject: Hi
Date: 21 May 93 15:06:27 GMT
Distribution: wadai
Organization: Dept. of Electric Power Eng., Saga Univ., Saga, Japan.
テストしてもいいですか？
--
&%%%%%%%%%%&
& 若林 裕望@ 佐賀大学大学院理工学研究科電気工学科1年 &
& E-mail:wakaba@ep.ee.saga-u.ac.jp &
&
& 谷村 有美は、いいですよ。 &
& p.s あなたは、cd "docile" を買いましたか？ &
&%%%%%%%%%%&
[-----] NGNUS: wadai.test/21 Hi (EEE:Article)--All-----
```

記事を読んでいる時に a をタイプすれば、そのニュースグループに記事を投稿できる。

現在読んでいる記事に対して、補足や反論などの付帯意見を述べたいときは、ここで

f

をタイプする。投稿する記事の中で元記事を引用したいときは、代わりに F をタイプする。

記事の投稿に関するマナー

投稿した記事は配布範囲に応じているいろいろな場所のコンピューターに配送される。comp などのカテゴリは世界中で購読されているため、ここに投稿した記事は全世界の何万台というコンピューターに配送される。このため、1 つの記事を投稿することは、膨大な量のコンピューターやネットワークという資源の消費を引き起こす。従って、積極的な投稿は歓迎されるが、できるだけ無駄を省くことが望ましい。

投稿のミス避けるため、事前に wadai.test で練習しておく。

引用は元の文章の意味が変わらない程度にできるだけ削る。

むやみに長い signature（署名）を付けない。

いくらコンピューターの上に構成された世界だと言っても、その向こう側にはやはり人間がいる。すなわち、このようなコミュニケーション形態でも、一般社会と同様な常識やマナーが適用されることを忘れないように。

課題（11） wadai.test に自己紹介の記事を投稿し、数分後その記事を読んでみよう。

清書印刷

```
emily0% emacs text.tex↵
```

something.tex のように .tex で終る名前のファイルを作成する .

```
\documentstyle{jarticle}
\begin{document}
UNIXはアメリカの電話会社AT&Tの
ベル研究所で開発された
オペレーティングシステムです .
```

最初に ¥documentstyle{jarticle} という 1 行を置く . ¥ と \ は同じ .

本文は ¥begin{document} と ¥end{document} の間に書く .

```
MS-DOSは木構造のファイルシステムや
入出力のパイプ, リダイレクトなど,
UNIXを真似ている部分がありますが,
マルチタスク, マルチユーザーの機能は
持っていないので,
複数の仕事を同時にさせたり,
複数の人が同時に使うことはできません .
\end{document}
```

段落を改める場合は, 空行を 1 行空ける .

行末はそろえる必要はないが, 1 行ごとに改行すること . 文や節単位で改行すれば, 後の編集が楽 .

C-x C-s でファイルを保存, あるいは C-x C-c で emacs を終了 .

```
emily% jlatex text.↵
```

今作成したファイルの名前を引数に指定して jlatex コマンドを実行 . 拡張子 .tex は省略できる .

エラーが起きて jlatex の処理が途中で止まる
ことがある .

....name: で止まったら Ctrl-D

....? で止まったら x↵

jlatex の処理が正常に終了すると元の .tex というファイルのほかに .log, .dvi, .aux という 3 つのファイルができる . このうち .dvi が目的のファイルである .

```
emily0% ls text.*↵
text.aux text.dvi text.log text.tex
```

jlatex による清書結果を画面上で確認する .

```
emily0% xdvi text.↵
```

UNIX はアメリカの電話会社 AT&T のベル研究所で開発されたオペレーティングシステムです .

MS-DOS は木構造のファイルシステムや入出力のパイプ, リダイレクトなど, UNIX を真似ている部分がありますが, マルチタスク, マルチユーザーの機能は持っていないので, 複数の仕事を同時にさせたり, 複数の人が同時に使うことはできません .

新しいウィンドウが開く . そのウィンドウ内にマウスを移動し,

マウスの右ボタンを押す

次のページの表示

マウスの左ボタンを押す

前のページの表示

Ctrl-C または q をタイプ

終了

画面上に先ほどの文章を清書した内容が表示される . 所望の結果が得られていれば, 印刷する .

```
emily0% lpr -d text.dvi
```

清書した結果を印刷する。 .dvi は省略できない。

```
emily0% lpr text.tex
```

ファイルの内容をそのまま印刷するには lpr コマンドを使う。

```
emily0% lpq  
emily0% lprm
```

印刷の順番を表示する。
印刷を取り消す。

レポートなどのまとめ方

(これらは省略可)
用紙はA4 本文 2 段組み タイトル, 要約を別のページに印刷

```
\documentstyle[a4j,twocolumn,titlepage]{jarticle}  
\begin{document}  
\title{この部分にタイトルを書く\\途中で改行をいれるには\\を  
入れる (普通に改行しても改行されない)}  
}  
\author{この部分に著者名を入れる \and 著者が複数のときは \and で区切る }  
\date{ July 13, 1993 }  
\maketitle  
\begin{abstract}  
この部分に要約を書く。  
\end{abstract}  
ここから本文  
\section{セクションの見出し, 番号は自動的にふられる}  
\subsection{サブセクションの見出し}  
本文その 1 . 1  
\subsection{サブセクションの見出し}  
本文その 1 . 2 \footnote{ここは脚注}  
\section{セクションの見出し}  
本文その 2  
\begin{itemize}  
\item 箇条書その 1  
\item 箇条書その 2  
\end{itemize}  
\begin{enumerate}  
\item 番号つき箇条書その 1  
\item 番号つき箇条書その 2  
\end{enumerate}  
\end{document}
```

この行を省略すると今日の日付になる。
これを忘れるとタイトルが付かない。
要約を付けないなら不要

セクション等を設けないなら不要。他に下のものがある。

```
\part  
↓  
\chapter  
↓  
\section  
↓  
\subsection  
↓  
\subsubsection  
↓  
\paragraph  
↓  
\subparagraph
```

忘れないように。

/usr/local/doc/latex というディレクトリにある latex.dvi というファイルを lpr -d latex.dvi で印刷すれば latex の入門書が得られる (用紙の節約のため, できるだけ xdvi を使用して画面上での確認にとどめること)。

課題 (12) 授業の感想文を jlatex を使って清書し, 印刷して提出せよ。タイトル, 著者, 日付を明記し (これらは同一ページに収めること), 要約は付けない。全部で 500 文字以上とする (ファイルの文字数の数え方は課題 (5) に示したコマンドによる)。