

UNIX システム利用入門 初級編

- 基本的なコマンドおよびエディターの使用法 -

システム運用掛 宮崎 洋
共同利用掛 森田 仁

1 システム概要

1.1 システム構成

東京大学情報基盤センターの大型計算機システムにはベクトル並列型スーパーコンピューター、超並列型スーパーコンピューター、汎用大型計算機があり、UNIX システムにおいては、それぞれ以下のように利用が可能です。

ベクトル並列型スーパーコンピューター (OS: HI-UX/MPP(UNIX))

HITACHI SR8000 (ホスト名 `sr8000-s`, `sr8000-p`)

総ノード数 : 128 ノード演算性能 : 8GFLOPS

主記憶容量 : 1 ノードにつき 8GB 拡張記憶容量 : 24GB (主記憶の一部を統合して構成)

言語プロセッサ : FORTRAN77, FORTRAN90, C, C++

エディター : vi

超並列型スーパーコンピューター (OS: HI-UX/MPP(UNIX))

HITACHI SR2201 (ホスト名 `scalar-mpp`)

総ノード数 : 1024 ノード演算性能 : 256MFLOPS

主記憶容量 : 1 ノードにつき 256MB

言語プロセッサ : FORTRAN77, FORTRAN90, C, C++

エディター : vi

汎用大型計算機 (OS: VOS3/FS, HI-OSF/1-MJ(UNIX))

HITACHI MP5800/320 (ホスト名 `m-unix`)

(UNIX 用として) 主記憶容量 : 2GB 拡張記憶容量 : 2GB

言語プロセッサ : FORTRAN77, C, C++

エディター : vi, nemacs

UNIX システム用入出力装置として

磁気テープ装置 : オープンリール型、カートリッジ型、DAT, QIC, 8mm

プリンター : ポストスクリプトプリンター (モノクロ 2 台、カラー 2 台)

汎用大型計算機 (MP5800/320) は計算機資源を VOS3 と OSF/1(UNIX) に分割して各々別のシステムとして運用しています。ベクトル並列型スーパーコンピューター (SR8000)、超並列型スーパーコンピューター (SR2201) は計算機資源をインタラクティブ処理サブシステム (TSS)、バッチ処理サブシステム (BATCH) に分割しており、ジョブの形態によって処理を分担するというサブシステム構成をとっています。現在、ログインセッションは TSS、バッチジョブは BATCH で実行されるよう以下のように構成しています。

MP5800/320		SR8000		SR2201
TSS/ BATCH	TSS/ BATCH	TSS (sr8000-s)	TSS (sr8000-p)	TSS (scalar-mpp)
(VOS3)	(m-unix)	BATCH		BATCH

1.2 接続方法

ネットワーク(インターネット)を経由して本センターの大型計算機システムを利用するには、telnet コマンド等の接続先として下表のホスト名を指定して下さい。

各ホストのホスト名の対応は次の通りです。

機種	OS	ホスト名
HITACHI MP5800/320	VOS3/FS HI-OSF/1-MJ (OPAC)	m-vos.cc.u-tokyo.ac.jp m-unix.cc.u-tokyo.ac.jp opac.cc.u-tokyo.ac.jp
HITACHI SR8000	HI-UX/MPP	sr8000-s.cc.u-tokyo.ac.jp sr8000-p.cc.u-tokyo.ac.jp
HITACHI SR2201	HI-UX/MPP	scalar-mpp.cc.u-tokyo.ac.jp

2 ログイン、ログアウト

2.1 ログイン

UNIX システム (m-unix, sr8000-s, sr8000-p, scalar-mpp) に接続すると、

```
HI-OSF/1-MJ (m-unix.cc.u-tokyo.ac.jp) (ttyp0)
login:
```

と表示され、ログイン名の入力待ちとなりますので、自分のログイン名(利用者番号)を小文字で入力して下さい。ログイン名を入力すると、

```
Password:
```

と表示されますので、パスワードを入力して下さい。このとき入力した文字は表示されませんのでご注意下さい。ログイン名あるいはパスワードが間違っている場合は、

```
Login incorrect
login:
```

と表示されますので、再度ログイン名から入力して下さい。なお、連続して5回以上まちがえるとログインできなくなりますのでご注意下さい。

ここで、ログイン名、パスワードとも正しければ、利用可能状態になりますが、初めてログインしたときに限り、パスワードの変更を要求してきます。これはセキュリティ上、より他人に知られにくいパスワードに変更してもらうものです。

```
Your passwd has expired.
Changing password for a30000.
Old password: _____          newuser で指定したパスワード
Last successful password change for a30000: NEVER
Last unsuccessful password change for a30000: NEVER
New password: _____          新パスワード
Re-enter new password: _____ パスワード再入力
```

このパスワード変更の後、(2回目以降のログインの場合は、パスワード入力のすぐ後)以下のようなメッセージが表示されます。これをログインメッセージといいます。

```
Last login: Thu Jul  8 11:11:07 from hostname
```

```
Welcome to MP5800 OSF/1 Unix at Computer Centre, the University of Tokyo (CCUT).
```

HI-OSF/1-MJ 05-03-/B (Mon May 31 17:00:00 JST 1999) / m-unix.cc.u-tokyo.ac.jp.

See /info/* files (in JIS-kanji) for getting further information.
("show-info" is the command to read these files.)

(Latest /info/* date: 1999.04.24)

Information --

For further information, see Announce with "show-info" command.

System will shutdown at Mon May 31 07:00:00 1999

System will start at Mon May 31 17:00:00 1999

TERM = (vt100)

%

ここで、1行目は最後にログインした日付、時間と接続先、3~7行目はセンターからのお知らせ、8~9行目はシステムの停止予定と再開予定日時です。システムに関する重要なメッセージが記述されますので、必ず読むようにして下さい。特にシステムの停止予定と再開予定等はスーパーコンピューティングニュース掲載内容が変更となった場合に最新の情報を記述しますので注意して見て下さい。

最後に、端末タイプの問い合わせで止まることがあります。端末タイプとしては通常 vt100 で問題ありませんが、本来は端末に合わせて設定する必要があります。

これらの操作を終了すると、プロンプト % が表示され、ログイン操作の完了となります。

2.2 ログアウト

TSS セッションを終了するには、logout または exit コマンドで終了します。

% logout

2.3 パスワードの変更

パスワードは、

% passwd

コマンドで変更します。指定できるパスワードは注意事項を参照して下さい。このとき入力した文字は表示されません。また、大文字と小文字は区別されます。なお、パスワードは定期的に変更していただくようお願いします。UNIX システムのパスワードには有効期間を設定しており、約3ヵ月以上変更がなかった場合、ログイン時にパスワードの変更が必要になります。またパスワードを変更せず1年以上経過した場合はそのアカウントでの利用ができなくなりますのでご注意ください。もしパスワードを忘れてしまった場合は、共同利用掛までご連絡下さい。

パスワード変更例

```
% passwd
Changing password for a30000
Old password:_____ 現在のパスワードを入力する
Last successful password change for a30000: Tue May 8 09:50:56 1996
Last unsuccessful password change for a30000: NEVER
New password:_____ 新しいパスワードを入力する
Re-enter new password:_____ 確認のため再度新しいパスワードを入力する
```

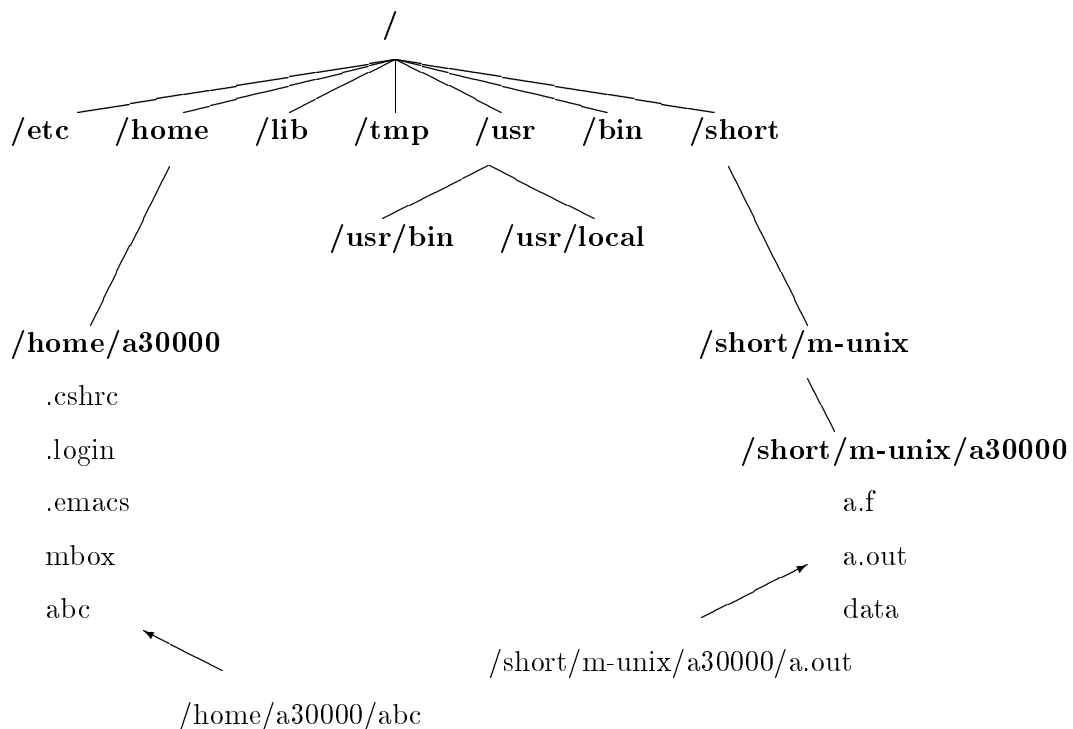
パスワード管理についての注意事項

- 8文字で英大文字、英小文字、数字、記号を適当に混在させる。
- 平易な語を使用しない。(氏名、生年月日、電話番号、利用者番号、辞書にある英単語など。)
- 回文(前後いづれから読んでも同じ)や単語の逆並びを使用しない。
- 定期的な変更を心掛け、以前と同じパスワードは設定しない。
- 以上を勘案して自分だけには記憶し易い文字列を設定し、どこかに書き留めておくようなことはしない。(パスワードを記したメモを第三者の目に留まる壁や端末等に張り付けることは厳禁。)

3 ファイルとディレクトリー

ファイル・ディレクトリー

UNIX システム上に情報を格納するためにはファイルを作成する必要があります。プログラムやデータ、文書、電子メールのメッセージ等は“ファイル”という単位でディスク上に保存されます。ファイルは下図のようなツリー構造を形成する“ディレクトリー”によってグループごとにまとめられており、必要なファイルの保存されている場所はディレクトリーによって指し示すことができます。



ディレクトリーの最上部は / (ルートディレクトリー) で表わされ、その下を各グループ (/home, /short, /usr, /tmp 等) に分けます。さらに階層が深くなるごとにスラッシュ記号 / で区切り、以下のようにグループごとに分けていくことでディスク上に保存された目的のファイルを指し示すことができます。

/	ルートディレクトリー
/short	短期ファイルディレクトリー
/short/m-unix	短期ファイル (m-unix) ディレクトリー
/short/m-unix/a30000	利用者 a30000 の短期ファイル (m-unix) ディレクトリー
/short/m-unix/a30000/a.out	ファイル a.out

利用者ディレクトリー

UNIX システムで利用者が使用できるディスクとして以下のディレクトリーを用意していません。利用者ディレクトリーは /home, /short, /tmp 等でそれ以外はシステムで使用します。

/home

利用者のホームディレクトリー。ここに作られたファイルは利用者が削除するまで保存されます。VOS3 の長期データセットに相当します。

/short

利用者が作業用に利用するディレクトリー。ここに作られたファイルは最終アクセスから 15 日後に削除されます。VOS3 の短期データセットに相当します。

/tmp

システムおよび利用者が短時間の作業用に利用するディレクトリーです。ここに作られたファイルは作成終了後しばらくして (平均 1 時間以内。最低 30 分、最大 1 時間 30 分以内で) 削除されます。

ホームディレクトリー

ログインしたとき、最初に参照できるディレクトリーが“ホームディレクトリー”で、各利用者ごとに区切られています。利用者は基本的にこのホームディレクトリー上にファイルを作成、保存します。例えば、利用者 a30000 の人がログイン直後にファイル abc を作成するとホームディレクトリー /home/a30000 にファイル abc が保存されます。このファイル abc はシステム全体から見たとき /home/a30000/abc と表現されます。

利用者 a30000 が使用できるディレクトリー

/home/a30000

ホームディレクトリー

/short/m-unix/a30000

短期ファイルディレクトリー (m-unix)

/tmp

一時ファイルディレクトリー

など

カレントディレクトリー

他のディレクトリーへは cd コマンドで移動することができます。コマンドを入力する状態にあるとき現在位置しているディレクトリーのことを“カレントディレクトリー”と呼びます。ログイン直後のカレントディレクトリーはホームディレクトリーです。またカレントディレクトリーを表示するためには pwd コマンドを使用します。

```
% pwd
```

```
/home/a30000
```

4 シェルとリダイレクション, パイプ

ログインするとログインメッセージのあとにプロンプト % が現れますがこれは C シェルというコマンド解析プログラムがコマンド入力要求のメッセージとして出力しています。つまり、UNIX コマンドを入力できる状態にあるということを意味しています。

```
% プロンプト
```

```
% ls
```

```
Mail      News      a.f      a.out    aaa      sample
```

ls というコマンドを入力すると C シェルは ls がコマンドであると解釈し、動作をコマンド実行に移します。ここで ls コマンドはカレントディレクトリーにあるファイルのリストを表示します。

```
% ls a*
a.f    a.out    aaa
```

ls コマンドの引数として a* を指定すると上記のようになります。これはアスタリスク * をワイルドカードとみなし、カレントディレクトリーにある a で始まるファイル名を全て表示するためです。C シェルが解釈する特定の文字は特殊な意味を持っています。例えば次のようなものがあります。

- * 任意の文字列にマッチする
- ? 任意の 1 文字にマッチする

```
% ls a??
a.f    aaa          ( 任意の 2 文字にマッチしています。 )
```

コマンドの出力をファイルに保存すること、またファイルの内容をコマンドに渡すことができます。これをリダイレクションといいます。記号には > または < を用い次のように使います。

```
% ls > file          出力リダイレクション
% cat file          ...( file の内容を表示 )
Mail
News
a.f
a.out
aaa
file
sample
```

file という名のファイルに ls コマンドの出力結果が格納されているのがわかります。これを出力リダイレクションといいます。

```
% pwd >> file        出力リダイレクション ( 追加書き )
```

とすると先ほどの file の内容に pwd コマンドの出力結果、つまりカレントディレクトリー名が追加書きされます。

```
% cat file
Mail
News
a.f
a.out
aaa
file
sample
/home/a30000          ...( 追加書きされた行 )
```

このとき pwd > file とすると新規に file という名のファイルを作成しようとしていますが、ファイル file は既に存在するので次のようなエラーとなるでしょう。

```
% pwd > file
file: File exists.
```

次に入力リダイレクションです。これはファイルの内容をコマンドに与えることができます。

```
% grep a < file      入力リダイレクション
```

grep コマンドは文字 a を含む行を検索、表示しようと入力を待ちます。そこに file というファイルの内容が入力リダイレクションより入ってきたのでそれを処理します。

また、リダイレクションに似た機能でパイプがあります。前のコマンドの結果を次のコマンドに渡します。

```
% cat file | grep a      パイプ
```

file の内容をパイプ | で次のコマンド grep に渡します。パイプ経由でもらったファイルの内容は grep コマンドが処理し、文字 a を含む行を画面に表示します。結果は次のコマンドと同じです。

```
% grep a file
Mail
a.f
a.out
aaa
sample
```

5 主な UNIX コマンド

ls - ファイル情報の表示

```
% ls
Mail      News      a.f      a.out    sample
% ls -l
total 16
drwx--x--x  2 a30000      users          4096 May 10 22:43 Mail
drwx--x--x  2 a30000      users          4096 May 10 22:43 News
-rw-----  1 a30000      users           258 May 10 22:43 a.f
-rwx--x--x  1 a30000      users        430604 May 10 22:45 a.out
-rw-----  1 a30000      users           29 May 10 22:43 sample
% ls -a      (ピリオドで始まるファイル名も表示されます。)
.      ..      .cshrc  .login  Mail     News     a.f     a.out    sample
```

cat - ファイル内容の表示

```
% cat sample
sample file
abc abc abc
%
```

grep - パターン検索

```
% grep abc sample
abc abc abc      ( abc という文字を含む行を表示します。 )
%
```

cp - ファイルのコピー

```
% cp sample sample.org
% ls
Mail      News      a.f      a.out    sample   sample.org
```

rm - ファイルの削除

```
% rm sample
% ls
Mail      News      a.f      a.out    sample.org
```

mv - ファイルの移動(ファイル名の変更)

```
% mv sample.org sample
% ls
Mail      News      a.f      a.out    sample
```

pwd - 現在のディレクトリーの表示

```
% pwd
/home/a30000
```

cd - 現在のディレクトリー変更

```
% cd Mail
% pwd
/home/a30000/Mail
% cd                                     (ホームディレクトリーに戻る)
% pwd
/home/a30000
% cd ..                                  (ひとつ上のディレクトリーに移る)
% pwd
/home
% cd /short/m-unix/a30000 (短期保存用ディレクトリーに移る)
% pwd
/short/m-unix/a30000
% cd
```

mkdir, rmdir - ディレクトリーの作成、削除

```
% mkdir tmp                               (ディレクトリーの作成)
% ls
Mail      News      a.f      a.out    sample   tmp
% cd tmp
% pwd
/home/a30000/tmp
% ls
% cd ..
% ls
Mail      News      a.f      a.out    sample   tmp
% rmdir tmp                               (ディレクトリーの削除)
% ls
Mail      News      a.f      a.out    sample
```


more, less - ファイル内容の表示 (一画面ずつ表示)

```
% ls -l | more
total 7552
-rw----- 1 a30000 users 46 Jan 28 1999 STDIN.o18521
-rw----- 1 a30000 users 18 Feb 02 1999 STDIN.o18614
-rw----- 1 a30000 users 108 Jul 28 1997 a.csh
-rwx--x--x 1 a30000 users 457032 Jan 22 1999 a.out
-rw----- 1 a30000 users 143 Jun 12 1997 autodb1.f
-rw----- 1 a30000 users 66 Feb 03 1999 char.f
-rw----- 1 a30000 users 149 Feb 04 1998 clock.f
-rw----- 1 a30000 users 305 Jul 08 16:17 clock1.f
-rwx--x--x 1 a30000 users 455244 Sep 09 1997 dat
-rw----- 1 a30000 users 474581 Jul 09 17:10 data
-rw----- 1 a30000 users 76 Dec 17 1996 date.f
-rw----- 1 a30000 users 165 Mar 17 13:21 do.f
-rw----- 1 a30000 users 563 Jan 28 1999 es.f
-rw----- 1 a30000 users 65792 Jun 17 1998 file
-rw----- 1 a30000 users 111 Jun 11 1997 getarg.f
-rw----- 1 a30000 users 57 Jun 10 1997 getenv.f
-rw----- 1 a30000 users 54 May 19 1998 go1.csh
-rw----- 1 a30000 users 111 Dec 22 1997 hf_dcoma.f
-rw----- 1 a30000 users 187 Aug 23 08:52 hsrulm.f
-rw----- 1 a30000 users 169 Jun 17 1998 open.f
-rw----- 1 a30000 users 1595 Apr 16 15:59 write.f
-- More --
```

一画面で表示が止まり、スペースキーで次頁へ進みます。

```
% less /info/m-unix/centre_news/cnews9705-mt
```

```
UNIX システムにおける磁気テープ装置の使い方
- 8mm, DAT, QIC, Open MT, Cartridge MT -
```

システム運用掛

```
UNIX システムで利用できる磁気テープ装置として以前より設置していた
8mm, DAT, QIC に加えて 1/2 インチオープンリール磁気テープ装置、
1/2 インチカートリッジ磁気テープ装置のサービスを開始しました。
```

```
:
:
```

コマンドにより操作します。

```
/info/m-unix/centre_news/cnews9705-mt 13%
```

スペースキーで次頁へ、"q"キーで less を終了します。less 及び一部の more は 1 行単位で制御できます。"j"キーは画面を 1 行進め、"k"キーは画面を 1 行戻します。

f77 - 最適化 FORTRAN77 コンパイラーを起動する

```
% f77 a.f
```

cc - 最適化 C コンパイラーを起動する

```
% cc a.c
```

qsub - NQS ジョブをサブミットする

```
% qsub job.csh
Request 1280.m-unix.cc.u-tokyo.ac.jp submitted to queue: m-unix.
```

qstat - NQS のジョブ処理状況を表示する

```
% qstat
1999/09/02 (Thu) 14:23:51:   REQUESTS on m-unix.cc.u-tokyo.ac.jp
  REQUEST      NAME      OWNER      QUEUE      PRI NICE  CPU    MEM    STATE
16311.m-unix.c job.csh  a30000     A           31  20    120   1024  RUNNING
```

```
% qstat -b
1999/09/02 (Thu) 14:23:54:  BATCH  QUEUES on m-unix.cc.u-tokyo.ac.jp
QUEUE NAME      STATUS      TOTAL  RUNNING  RUNLIMIT  QUEUED  HELD   IN-TRANSIT
A                AVAILBL    1      1         3          0       0       0
B                AVAILBL    0      0         3          0       0       0
C                AVAILBL    0      0         2          0       0       0
D                AVAILBL    0      0         2          0       0       0
E                AVAILBL    0      0         1          0       0       0
F                AVAILBL    0      0         1          0       0       0
L                STOPPED    0      0         1          0       0       0
```

qdel - NQS ジョブをキャンセルする

```
% qdel -k 1280
Request 1280 is running, and has been signalled.
```

FORTRAN、NQS 関係のコマンドについて詳細は UNIX システム利用入門 - 中級編 - を参照して下さい。

man - オンラインマニュアル

```
% man ls
```

```
ls(1)                                HI-OSF/1-MJ                                ls(1)
```

NAME

ls - Lists and generates statistics for files

SYNOPSIS

```
ls [-aAbcCdEfGgILmnpqrRstux1] [file ... | directory ...]
```

FLAGS

-a Lists all entries in the directory including the entries that begin with a . (dot).

ftp - ファイルの転送

anonymous ftp からファイル入手する方法の例

```
% ftp ftp.cc.u-tokyo.ac.jp
Connected to ftp.cc.u-tokyo.ac.jp.
:
Name (ftp.cc.u-tokyo.ac.jp:a30000): ftp          ftp または anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:                                       自分の E-mail アドレス
ftp> cd /pub/X11R6
ftp> ls
:
ftp> get README
200 PORT command successful.
150 Opening BINARY mode data connection for README (352 bytes).
226 Transfer complete.
352 bytes received in 0.019 seconds (18 Kbytes/s)
ftp> bye
221 Goodbye.
% ls
Mail      News      README    a.f      a.out    sample   sample.org
```

network 接続されたホスト oogata-lab.cc.u-tokyo.ac.jp にファイルを転送する方法の例 (次の例は架空のホストのため接続できません。)

```
% ftp oogata-lab.cc.u-tokyo.ac.jp
Connected to ftp.cc.u-tokyo.ac.jp.
:
Name (oogata-lab.cc.u-tokyo.ac.jp:a30000): oogata
331 Password required for oogata.
Password:
230 User oogata logged in.
ftp> put file.txt
200 PORT command successful.
150 ASCII data connection for file.txt (000.00.000.00,3742).
226 ASCII Transfer complete.
1448 bytes sent in 0.044 seconds (32 Kbytes/s)
ftp> bye
221 Goodbye.
%
```

ps - プロセスの確認

コマンド、プログラム等を動かすとその処理はプロセスという形で実行されます。現在自分の動かしている全てのプロセスを見てみましょう。

```
% ps a
  PID TT  STAT      TIME COMMAND
 11374 p1  R   +    0:00.01 ps          ( ps コマンド )
 12136 p1  S      0:00.20 -csh (csh)      ( C シェル )
```

他のプロセスが走っていると、以下のように 1 行 (またはそれ以上) 増えます。

```
% ps a
  PID TT  STAT      TIME COMMAND
  7176 p1  R   +   0:00.01 ps a
 12136 p1  S      0:00.21 -csh (csh)
 11309 p4  S   +   0:00.11 -csh (csh)
```

kill - プロセスの切断

正常に logout できなかった時などプロセスが残ってしまうことがあります。
(最後の行が残ってしまったプロセスとします。)

```
% ps a
  PID TT  STAT      TIME COMMAND
  7176 p1  R   +   0:00.01 ps a
 12136 p1  S      0:00.21 -csh (csh)
 11309 p4  S   +   0:00.11 -csh (csh)
```

kill コマンドには終了させたい PID (プロセス ID, ここでは 11309) を指定します。
(kill だけでは終了させられないプロセスがあるので kill -9 としています。)

```
% kill -9 11309
% ps a
  PID TT  STAT      TIME COMMAND
 10381 p1  R   +   0:00.01 ps a
 12136 p1  S      0:00.22 -csh (csh)
```

mail - メールを送信、受信

自分のログイン名(つまり自分自身)にメールを出してみましょう。
メールは実際に送信されますのでメールアドレス、カーボンコピーは間違いのない
よう十分確認して下さい。間違えて送信すると他のホストに迷惑がかかる場合が
あります。

例題のメールアドレスでは a30000 へ送信してしまいますので a30000 の部分を
自分のログイン名に置き換えて下さい。

```
% mail a30000@m-unix.cc.u-tokyo.ac.jp
Subject: mail test          ( サブジェクト )
mail no test dayo.        ( 本文 )
.                          ( ピリオドだけの行 )
Cc:                        ( カーボンコピー )
%
```

エディター(nemacs または vi)でメール本文を作成してみましょう。

```
% nemacs -nw file.m
```

メール用に作成したファイル file.m がある場合。

```
% mail a30000@m-unix.cc.u-tokyo.ac.jp < file.m
```

次に送ったメールを受信します。

```
% mail
Mail [5.2 UCB] [OSF 1.0] Type ? for help.
"/var/spool/mail/a30000": 1 message 1 new
>N 1 a30000          Mon May 12 22:22 11/336 "test mail"
? 1                  (メールの番号,ここでは 1 を指定します。)
Message 1:
From a30000 Mon May 12 22:22 JST 1997
Date: Mon, 12 May 1997 22:22:34 +0900      (日付)
From: NAME NAME <a30000>                   (差出人)
To: a30000@m-unix.cc.u-tokyo.ac.jp        (宛先)
Subject: test mail                         (サブジェクト)

mail no test dayo.                         (本文)

? q                                          ( "q" で終了します。)
Saved 1 message in mbox
% ls
Mail      News      a.f      a.out    file.m   mbox     sample
```

"mbox" というファイルができています。先ほど読んだメールがこのファイル "mbox" に保存されています。ファイルに保存したメールを読むためには次のようにします。

```
% mail -f mbox
Mail [5.2 UCB] [OSF 1.0] Type ? for help.
"mbox": 1 message
> 1 a30000          Mon May 12 22:22 12/347 "test mail"
? 1
Message 1:
From a30000 Mon May 12 22:22 JST 1997
Date: Mon, 12 May 1997 22:22:34 +0900
From: NAME NAME <a30000>
To: a30000@m-unix.cc.u-tokyo.ac.jp
Subject: test mail

mail no test dayo.

? q
%
```

6 vi エディターの使い方

1. vi エディターの起動

ファイル名(ここでは abc)を指定して vi コマンドを起動します。

```
% vi abc
```

■ カーソル

```
~  
~  
~
```

```
"abc" [New file]
```

vi エディター起動直後は "コマンドモード" になっています。この状態では文字は入力できません。

2. 文字の入力(カーソルの前に入力)

"i" と打って下さい。なにも変化はありませんが,"入力モード" になっています。以後キーボードから入力した文字が画面に表示されます。以下のように入力して下さい。

```
vi editor de abc wo input■
```

```
~  
~  
~
```

```
"abc" [New file]
```

3. 入力モードの解除

"コマンドモード" へ戻るためには ESC キー(エスケープキー)を押します。カーソルは行末の文字(t)の上に移動し,"コマンドモード" に戻りました。以下で説明する各コマンドが使えます。

```
vi editor de abc wo inpu■
```

```
~  
~  
~
```

```
"abc" [New file]
```

4. カーソルの移動

キーの対応

キー	h	j	k	l
移動				

"h" と打つとカーソルが左へと動きます。abc の a の文字の上にカーソルを移動して下さい。

```
vi editor de ■bc wo input
```

```
~  
~  
~
```

```
"abc" [New file]
```

5. 文字の削除, 修正

"x" と打つとカーソル位置の 1 文字が消えます。3 回押して abc を消して下さい。

```
vi editor de ■wo input
```

```
~  
~  
~
```

```
"abc" [New file]
```

再び "i" と打って "入力モード" にします。text と打って ESC キーを押して下さい。

```
vi editor de tex █ wo input
~
~
~
"abc" [New file]
```

6. ファイルへの保存

”:” を押すとカーソルが画面左下に移動します。

```
vi editor de text wo input
~
~
~
: █
```

”w” と打ち Enter キー (Return キー) を押します。するとテキスト文字はファイル abc に保存されます。

```
vi editor de text wo input
~
~
~
:w
```

7. vi エディターの終了

vi エディターを終了するには ”:” を押してから ”q” リターンです。

```
vi editor de text wo input
~
~
~
:q
% █
```

No write と出て終了できない時は :w (ファイルへの保存) を忘れていました。:wq リターンと打てばファイルへの保存後, vi エディターを終了します。

```
:w ファイルへの保存
:q エディターの終了
:q! ファイルへの保存をしないでエディターを終了する
:wq ファイルへの保存, エディターの終了
```

以上でファイル ”abc” が作成されました。cat コマンドで確認してみましょう。

```
% cat abc
vi editor de text wo input
```

8. ファイルの編集

先ほど作成したファイル ”abc” を書き換えてみましょう。再び vi エディターを起動します。

```
% vi abc

█i editor de text wo input
~
~
~
"abc" 1/31
```

9. 文字の入力 (行の追加)

2 行目に行を追加してみましょう。"o" を押して下さい。カーソルが下へ移動し、"入力モード"になります。

```
vi editor de text wo input
█
~
~
"abc" 1/31
```

そのまま文字を入力していきます。2 行目を書くことができます。

```
vi editor de text wo input
 2 gyoume mo kaketa█
~
~
"abc" 1/31
```

ここで Enter キー (Return キー) を押すと 3 行目へと移ります。まだ "入力モード" は続いています。

```
vi editor de text wo input
 2 gyoume mo kaketa
 3 gyoume mo kakeru█
~
~
"abc" 1/31
```

ここで ESC キー (エスケープキー) を押すと "コマンドモード" になります。カーソルを動かすコマンド "h" "j" "k" "l" を使って移動して見ましょう。

```
vi editor de text wo input
 2 gyoume mo kaketa█
 3 gyoume mo kakeru
~
~
"abc" 1/31
```

10. 文字の入力 (カーソルの後に入力)

2 行目の末尾に文字を追加してみましょう。行末の文字 (a) の上にカーソルを移動し、"a" と打つと "入力モード" になるので yo とでも打ってみてください。

```
vi editor de text wo input
 2 gyoume mo kaketayo█
 3 gyoume mo kakeru
~
~
"abc" 1/31
```

なぜ "a" を使うのでしょうか。カーソルの前に文字を入力する "i" では a の文字の前から書き込まれるので次のようになってしまいます。

```
vi editor de text wo input
 2 gyoume mo kaketyoa█
 3 gyoume mo kakeru
~
~
"abc" 1/31
```

主な入力コマンドには次のようなものがあります。

- a カーソルの右側に文字を入力する
- i カーソルの左側に文字を入力する
- o カーソルのある行の下に文字を入力する
- O カーソルのある行の上に文字を入力する

11. 行の削除

3行目の適当なところで "dd" と打つとその行が消えます。

```
vi editor de text wo input
 2 gyoume mo kaketayo
 3 gyoume■mo kakeru
~
```

```
"abc" 1/31
```

"dd" と打つと

```
vi editor de text wo input
■2 gyoume mo kaketayo
~
```

```
~
```

```
"abc" 1/31
```

12. 変更の取消

誤って行を削除するなどをした時は "u" コマンドで最後に行なわれた変更を元に戻します。

```
vi editor de text wo input
 2 gyoume mo kaketayo
■3 gyoume mo kakeru
~
```

```
"abc" 1/31
```

ファイルに保存して vi エディターを終了しましょう。(:wq リターン)

7 主な vi エディターのコマンド

画面制御 コマンド	Ctrl-l	現在の画面を再表示する。
	Ctrl-y	画面を1行上にスクロールする。
	Ctrl-e	画面を1行下にスクロールする。
ページ コマンド	Ctrl-f	次の1画面を表示する。
	Ctrl-b	前の1画面を表示する。
	Ctrl-d	半画面下にスクロールする。
	Ctrl-u	半画面上にスクロールする。
カーソル移動 コマンド	j	カーソルを同じ桁で1行下に移動する。
	k	カーソルを同じ桁で1行上に移動する。
	h	カーソルを1桁左に移動する。
	l	カーソルを1桁右に移動する。
	Return	カーソルを次の行の先頭に移動する。
	0(ゼロ)	カーソルを現在行の先頭に移動する。
	\$	カーソルを現在行の最後に移動する。
	スペース	カーソルを1文字進める。
	nG	カーソルを <i>n</i> 行目の先頭に移動する。 <i>n</i> 省略のとき、ファイルの最後の行。
	/pattern	次の <i>pattern</i> が現れるところにカーソルを進める。
	?pattern	前の <i>pattern</i> が現れたところまでカーソルをもどす。
n	最後に行なったパターン検索 / または ? を繰り返す。	
テキスト入力 コマンド	a	カーソルの右側にテキストを入力する。ESC で入力を終了する。
	A	カーソルのある行の行末にテキストを入力する。ESC で入力を終了する。
	i	カーソルの左側にテキストを入力する。ESC で入力を終了する。
	I	カーソルのある行の行頭にテキストを入力する。ESC で入力を終了する。
	o	カーソルのある行の下にテキストを入力する。ESC で入力を終了する。
	O	カーソルのある行の上にテキストを入力する。ESC で入力を終了する。
	Backspace	テキスト入力中、最後に入力した文字を削除する。
	ESC	テキスト入力を終了する。
テキスト削除 コマンド	dw	カーソルのある文字からその語(ワード)の終わりまでを削除する。
	x	カーソルのある文字を削除する。
	dd	カーソルのある行を削除する。
	D, d\$	カーソルのある文字から行の終わりまでを削除する。
テキスト変更 コマンド	cw	カーソルのある文字からその語の終わりまでを、文字列で置き換える。
	C, c\$	カーソルのある文字から行の終わりまでを、文字列で置き換える。
	J	現在行と次の行をつなげる。
	rx	カーソルのある文字を別の文字 <i>x</i> で置き換える。
バッファ編集 コマンド	nY	現在行から <i>n</i> 行をヤंकする。 <i>n</i> が省略されると、現在行をヤंकする。
	p	ヤंकしたテキストをカーソルの後に挿入する。
	P	ヤंकしたテキストをカーソルの前に挿入する。
ファイル編集 コマンド	:w <i>name</i>	ファイルをファイル名 <i>name</i> で保存する。
	:r <i>name</i>	ファイル名 <i>name</i> のファイルを読み込む。
vi の終了	:wq, ZZ	ファイルを保存して、エディターを終了する。
	:q, :q!	保存せずに、エディターを終了する。

8 nemacs エディターの使い方

1. nemacs エディターの起動

ファイル名(ここでは abc)を指定して nemacs コマンドを起動します。

```
% nemacs -nw abc
(または % nemacs abc とすると新たなウィンドウが開かれる)
```

■ カーソル
[----]-----NEmacs: abc (-JJ-:Fundamental)--All-----
(New file)

2. 文字の入力

nemacs エディターは起動直後から文字を入力することが可能です。

以後キーボードから入力した文字が画面に表示されます。以下のように入力して下さい。

もし間違えた時は Delete キー(削除キー)を押して1文字削除して下さい。ここで Backspace キー(後退キー)を使うと C-h (Type ? for further options) というメッセージが出てきてしまいます。このときは control キー(コントロールキー)を押しながら "g" を押すことで元の状態に戻ります。

nemacs editor de abc wo input■
[----]---*-NEmacs: abc (-JJ-:Fundamental)--All-----

3. コマンドの入力

nemacs ではコマンド入用に Control キー(コントロールキー)を使います。例えば "C-g" という時は Control キーを押しながら "g" を押すことを意味します。

4. カーソルの移動

キーの対応	ctrl-b	ctrl-n	ctrl-p	ctrl-f

Enter キー(Return キー)を押し、次の行へ移ったら "C-p" "C-f" 等を使って abc の a の文字の上にカーソルを移動して下さい。

nemacs editor de ■bc wo input
[----]---*-NEmacs: abc (-JJ-:Fundamental)--All-----

5. 文字の削除, 修正

"C-d" を押すとカーソル位置の1文字が消えます。3回繰り返して abc を消して下さい。

nemacs editor de ■wo input
[----]---*-NEmacs: abc (-JJ-:Fundamental)--All-----

消えたところで text と打つと文字が挿入されていきます。

```
nemacs editor de text wo input
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
```

6. ファイルへの保存

"C-x" "C-s" と続けてを押すとテキスト文字はファイル abc に保存されます。

```
nemacs editor de text wo input
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
Wrote /home/a30000/abc
```

7. nemacs エディターの終了

nemacs エディターを終了するには "C-x" "C-c" を続けてを押します。内容が保存されていないときは確認のメッセージが出ます。

```
nemacs editor de text wo input
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
Save file /home/a30000/abc? (y or n) y
```

```
% ■
以上でファイル abc が作成されました。cat コマンドで確認してみましょう。
% cat abc
nemacs editor de text wo input
% ■
```

8. ファイルの編集

先ほど作成したファイル abc を書き換えてみましょう。再び nemacs エディターを起動します。

```
% nemacs -nw abc または nemacs abc
```

```
■nemacs editor de text wo input
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
```

9. 文字の入力 (行の追加)

"C-n" を押して下さい。カーソルが下へ移動します。そのまま文字を入力すると2行目以降を書くことができます。

```
nemacs editor de text wo input
2 gyoume mo kaketa■
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
```

ここで Enter キー (Return キー) を押せば3行目へと移りますので入力続けて下さい。

```
nemacs editor de text wo input
 2 gyoume mo kaketa
 3 gyoume mo kakeru█
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
```

カーソルを動かすコマンド "C-p" "C-n" "C-b" "C-f" を使って移動して見ましょう。

```
nemacs editor de text wo input
 2 gyoume mo kaketa█
 3 gyoume mo kakeru
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
```

10. 文字の入力

2行目の末尾に文字を追加してみましょう。行末の文字(a)の後ろにカーソルを移動し、yoとでも打ってみてください。

```
nemacs editor de text wo input
 2 gyoume mo kaketayo█
 3 gyoume mo kakeru
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
```

11. 行の削除

3行目の行頭で "C-k" と打つとその行が消えます。

```
nemacs editor de text wo input
 2 gyoume mo kaketayo
 █3 gyoume mo kakeru
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
```

"C-k" と打つと

```
nemacs editor de text wo input
 2 gyoume mo kaketayo
 █
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
```

12. 変更の取消

誤って行を削除するなどをした時は "C-x" "u" コマンドで最後に行なわれた変更を元に戻します。"C-x" "u" というのは Control キーを押しながら "x" キーを押した後、"u" キーだけを押してください。

```
nemacs editor de text wo input
 2 gyoume mo kaketayo
 3 gyoume mo kakeru█
[----]---*-NEmacs: abc      (-JJ-:Fundamental)--All-----
```

ファイルに保存してエディターを終了しましょう。("C-x" "C-s" "C-x" "C-c")

13. 漢字の入力

ところで漢字を入力したい時はどうするのでしょうか。お使いの端末にかな漢字変換機能 (ATOK など) がついている時はそれを利用することができます。ここでは Wnm (うんぬ) というかな漢字変換機能について解説することにします。

"C-¥" (円記号) または "C-\\" (バックスラッシュ) で変換モードに入ります。左下の [----] が [a あ] に変わります。そこでローマ字で「かんじ」と入力すると次のようになります。

```
nemacs editor de text wo input
2 gyoume mo kaketayo
3 gyoume mo kakeru
|かんじ|

[ a あ]-----NEmacs: abc      (JJJ-:Fundamental)--All-----
```

ここでスペースバーあるいは "C-w" を押すと漢字変換されます。

```
nemacs editor de text wo input
2 gyoume mo kaketayo
3 gyoume mo kakeru
|漢字|

[漢字]-----NEmacs: abc      (JJJ-:Fundamental)--All-----
```

スペースバー または "C-n" で次候補, "C-p" で前候補を選べます。次の候補を選ぶと「感じ」ができました。

```
nemacs editor de text wo input
2 gyoume mo kaketayo
3 gyoume mo kakeru
|感じ|

[漢字]-----NEmacs: abc      (JJJ-:Fundamental)--All-----
```

確定する時は Enter キー (Return キー) または "C-l" です。

```
nemacs editor de text wo input
2 gyoume mo kaketayo
3 gyoume mo kakeru
|感じ■|

[ a あ]-----NEmacs: abc      (JJJ-:Fundamental)--All-----
```

以上の機能を使えば基本的な編集はできるようになるでしょう。nemacs エディターにはまだまだ豊富な機能が備わっています。使い込んでいくうちに便利な道具の一つとなるでしょう。

9 主な nemacs エディターのコマンド

1. 主な nemacs エディターのコマンド

nemacs のコマンドは、一般的にコントロールキーやメタキー (エスケープキー) と組み合わせて使用します。nemacs の説明では、次のように表記します。

- 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。
- M-(文字) エスケープキーを押してから、(文字) キーを押します。

カーソル 移動	C-f	右に 1 文字
	C-b	左に 1 文字
	C-n	下に 1 行
	C-p	上に 1 行
	C-a	行の先頭に移動
	C-e	行の末尾に移動
	M-<	ファイルの最初に移動
	M->	ファイルの最後に移動
挿入・ 削除	DEL キー	1 文字削除 (カーソル前)
	C-d	1 文字削除 (カーソル位置)
	C-k	カーソル位置から行末削除
	C-o	1 行挿入
ファイル 操作	C-x C-f	ファイルの読み込み
	C-x i	ファイルの読み込み (カーソル位置)
	C-x C-s	ファイルにセーブ
	C-x C-w	指定したファイルにセーブ
取り消し ・中止	C-x u	削除した文字, 行を戻す
	C-g	コマンドの中止
	C-x l	ウィンドウを 1 つにする。
終了	C-x C-c	NEmacs の終了

2. nemacs の仮名漢字変換

ここでは、日本語変換システムに Wnn を使用しています。
まず、ホームディレクトリーに .emacs というファイルを作って、次の 1 行を書き加えます。

```
(load-library "nemacs-egg")
```

次に、NEmacs を起動してから C-\ を打ち、ローマ字仮名モードに入ります。これで日本語が使えます。通常文字を入力して、スペース・キーを打つと漢字変換モードになります。この 2 つのモード中の編集コマンドを紹介します。

ローマ字仮名モード・コマンド

開始・終了	C-\	ローマ字仮名モードの開始と終了
カーソル 移動	C-f	右に 1 文字
	C-b	左に 1 文字
削除	C-d	1 文字削除
中止	C-g	入力を中止
変換	スペース・キー	仮名漢字変換開始
	M-k	カタカナに変換

漢字変換モード・コマンド

開始	スペース・キー	仮名漢字変換の開始と次候補の表示
カーソル 移動	C-f	右に 1 文節
	C-b	左に 1 文節
	C-i	文節の長さを短く
	C-o	文節の長さを長く
中止	C-g	漢字変換モードを中止
変換	C-p	前候補を表示
	M-h	文節をひらがな
	M-k	文節をカタカナ

10 プリンターへの出力

本センター 1 階の利用者端末室には UNIX システムで利用できるプリンター RICOH ネットワークプリンター LP7200-UX、FUJIX 超高画質フルカラーデジタルプリンター PICTROGRAPHY3000 を設置しています。ここではファイルや標準出力の内容をプリンターへ出力させるコマンドについて説明します。

1. プリンターへの出力

各プリンターへの出力は `print` コマンドで行ないます。

• コマンド

```
print [オプション...] ファイル名..
print -q [オプション...]
print -rm [オプション...] ジョブ番号..
```

• オプション

<code>-bw</code>	モノクロ LP7200-UX
<code>-color -colour</code>	超高画質カラー PICTROGRAPHY3000 (用紙出力)
<code>-ohp</code>	超高画質カラー PICTROGRAPHY3000 (OHP 出力)
<code>-single</code>	片面印刷
<code>-both</code>	両面印刷 (LP7200-UX のみ指定可)
<code>-ps</code>	PostScript ファイル出力
<code>-text</code>	テキストファイル出力 (LP7200-UX のみ指定可)
<code>-a4 -a3 -b4 -b5</code>	用紙選択 A4/A3/B4/B5 (LP7200-UX のみ指定可)

省略すると アンダーライン のオプションが仮定されます。

<code>-q</code>	プリンターキューの状態を表示
<code>-rm</code>	プリンターキューのジョブを削除

2. プリンターへの出力例

PostScript ファイルを LP7200-UX (片面) へ出力します。

```
% print file.ps
```

テキストファイルを両面で出力します。

```
% print -both -text text-file
```

PostScript ファイルを PICTROGRAPHY3000 (用紙) へ出力します。

```
% print -color file.ps
```

PostScript ファイルを PICTROGRAPHY3000 (OHP) へ出力します。

```
% print -ohp file.ps
```

3. プリンターキューの状態を表示

次に `print` コマンドでキューに投入したジョブを確認します。
LP7200-UX (片面) のキューの状態を表示します。

```
% print -q
Printer: LP7200-UX-single side(s)
```

	:			
Rank	Owner	Job	Files	Total Size
1st	a30000	382	file.ps	14821 bytes

4. プリンターキューのジョブを削除

キューの状態表示で確認したジョブが、必要のないものなら削除をします。
LP7200-UX (片面) に投入したジョブ番号 382 のジョブを削除します。

```
% print -rm 382
```


11 利用者登録

ログイン名の登録は、newuser というログイン名でログインすることにより、利用者自身で行います。登録方法は以下のとおりです。

1. ネットワーク(telnet, rlogin 等)から登録しようとするホスト(m-unix または sr8000-s, scalar-mpp)に接続します。
2. "login:" がでたところで、"newuser" と入力します。(このときパスワードの入力要求はありません。)
3. "VOS3 ID" (これがそのままログイン名となります。ただしアルファベットは大文字ではなく小文字です。)と"VOS3 PASSWORD" を聞いてくるので答えます。
このときパスワードは、
 - 平成6年度から7年度へ継続した方は平成7年度運用開始日時点での文字パスワード
 - 平成7年度以降に新規登録した方は利用承認通知書に書かれている文字パスワードに設定されています。
4. 登録可能システムが表示されるので登録希望ホストを指定します。

以下に登録例(m-unix に telnet 接続の場合)を示します。

```
myhost% telnet m-unix.cc.u-tokyo.ac.jp      telnet で接続
HI-OSF/1-MJ (m-unix.cc.u-tokyo.ac.jp) (ttyp0)
login: newuser                               ログイン名 newuser でログインする
                                           System Registration Command

Enter Your (VOS3) Login Name : a30000       VOS3 ID を小文字で入力する
Enter Your (VOS3) Password :                VOS3 パスワードを小文字で入力する

You are authorized to have your account in following host(s).
In the host(s) marked with '*' your account is already created.
Choose (list of) Host Number(s) on which you want to create
your account. (If you want to choose several, type the all host
numbers with the delimiter of space or comma. a-b denotes a
range from a to b.)

Host      Host Name                               Host Description
Number
  1      m-unix.cc.u-tokyo.ac.jp                ... ( 選択するホスト )
  2      sr8000-s.cc.u-tokyo.ac.jp
  3      frontend.cc.u-tokyo.ac.jp
  4      x-controller1.cc.u-tokyo.ac.jp
  5      io-controller.cc.u-tokyo.ac.jp
  6      scalar-mpp.cc.u-tokyo.ac.jp
  7      mathematica.cc.u-tokyo.ac.jp
  8      macsymba.cc.u-tokyo.ac.jp
  9      cross-compiler.cc.u-tokyo.ac.jp

Host Number (or q for quit): 1              登録しようとするホストの番号を選択
Welcome to m-unix.cc.u-tokyo.ac.jp
Select an item number that you want to do.

1. User Account Registration                ... ( 登録 )
2. User Account Deletion                    ... ( 削除 )
3. Account Code Change                      ... ( 課金番号の変更 )
4. Increase File Size Limit                 ... ( /home ファイル使用量 増 )
5. Decrease File Size Limit                 ... ( /home ファイル使用量 減 )
```

Item Number (or q for quit): 1 登録は 1. を選択

User Account Registration

Available account list is the following.

acct	course	budget	fare	balance	cpu_used(HH:MM:SS)
A	1000	1000000	31680	968320	0:00:00

Specify account code [default: A] (or quit): A 課金番号を選択

Specified Account code is A. Is it ok? [default : y]: y 確認

Fee of your /home file is calculated as the followings;

File size up to 50MB	: Free
File size between 50MB and 1250MB	: 20 Yen/(MB*month)
File size more than 1250MB (max 4000MB)	: 24,000 Yen/month

Specify your /home size(MB)[default 50MB, max 4000MB]: 100 ファイル使用量上限値

Specified your /home size is 100MB. Is the value ok ? [default : y]: y 確認

Fee for this month (23 day(s) left) ... (今月の /home ファイル課金額)

ITEM	Fee(Yen)
/home	741
TOTAL	741

Fee for each month from next month ... (来月以降の /home ファイル課金額)

ITEM	Fee(Yen/month)
/home	1000
TOTAL	1000

Do you create your user account? (y/n) [default : y]: y 確認

Creating your user account a30000 in m-unix.cc.u-tokyo.ac.jp.

-- Please wait a few minutes --

Finish creating your account a30000 in m-unix.cc.u-tokyo.ac.jp.

Select an item number that you want to do.

1. User Account Registration
2. User Account Deletion
3. Account Code Change
4. Increase File Size Limit
5. Decrease File Size Limit

Item Number (or q for quit): q 終了は q

You are authorized to have your account in following host(s).

In the host(s) marked with '*' your account is already created.

Choose (list of) Host Number(s) on which you want to create your account. (If you want to choose several, type the all host numbers with the delimiter of space or comma. a-b denotes a range from a to b.)

Host Number	Host Name	Host Description
1	* m-unix.cc.u-tokyo.ac.jp	... (登録完了なら *印)
2	sr8000-s.cc.u-tokyo.ac.jp	
3	frontend.cc.u-tokyo.ac.jp	
4	x-controller1.cc.u-tokyo.ac.jp	
5	io-controller.cc.u-tokyo.ac.jp	
6	scalar-mpp.cc.u-tokyo.ac.jp	
7	mathematica.cc.u-tokyo.ac.jp	
8	macsyma.cc.u-tokyo.ac.jp	
9	cross-compiler.cc.u-tokyo.ac.jp	

Host Number (or q for quit): q 終了は q

Connection closed by foreign host.

myhost% ... (回線が切断される)

利用負担金（1999年5月1日現在）

1. 汎用大型計算機 MP5800; m-vos, m-unix

● CPU 課金

利用者が使用する CPU 時間 / 月について以下の単価を適用します。

月額 1,000 円定額	月額 2,000 円定額	
～ 10 時間 / 月	～ 30 時間 / 月	0 円 / 秒
10 ～ 50 時間 / 月	30 ～ 40 時間 / 月	0.3 円 / 秒
50 時間 / 月～	40 時間 / 月～	0 円 / 秒

● ファイル課金

－ 長期保存ファイル (VOS3)、 /home(UNIX)

利用者が指定したファイル量の上限値まで使用しているものとして以下の単価を適用します。

月額 1,000 円定額	月額 2,000 円定額	
～ 50MB	～ 100MB	0 円 / (MB・月)
50 ～ 1,250MB	100 ～ 1,000MB	20 円 / (MB・月)
1,250MB～	1,000MB～	0 円 / (MB・月)

上限値の変更手続きは VOS3 は「大型計算機システム届出書」、UNIX は「newuser」により行います。上限値の変更が月の途中で増加の場合は差額分の日割計算とします。また、月の途中で削減はできず、翌月より反映します。

－ 短期保存ファイル (VOS3)、 /short(UNIX)

利用者が使用しているファイル量に対して以下の単価を適用します。
5 円 / (月・MB)

2. ベクトル並列型スーパーコンピュータ SR8000; sr8000-s, sr8000-p

● CPU 課金

利用者が使用する CPU 時間 / 月（または CPU 時間 / 月に相当する値）について以下の単価を適用します。

月額 1,000 円定額	月額 2,000 円定額	
～ 10 時間 / 月	～ 30 時間 / 月	0 円 / 秒
10 ～ 6,000 時間 / 月	30 ～ 5,000 時間 / 月	0.02 円 / 秒
6,000 時間 / 月～	5,000 時間 / 月～	0 円 / 秒

SR8000 の 1 ノードは 8 台のプロセッサから構成されています。各ノードには以下に示すノード属性（共有 / 占有）を設定しており、ログインセッションやバッチジョブが使用するノードによって課金方法が異なります。

－ 共有ノード

ジョブクラス A ～ F でバッチジョブを実行する場合、および sr8000-s にログインする場合に使用されるノードが共有ノードです。複数利用者でノードを共有するので他の利用者も同時に使用できます。課金は CPU 時間（スカラー計算時間）に対して行いません。

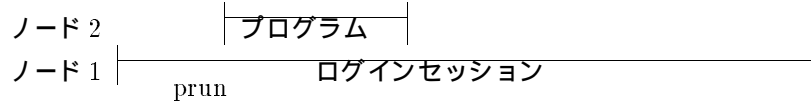
－ 占有ノード

ジョブクラス A-ES ～ F-ES, P001 ～ P016 でバッチジョブを実行する場合、および sr8000-p にログインする場合に使用されるノードが占有ノードです。1 ジョブがノードを占有するのでプロセッサが空いていても他の利用者は使用できま

せん。(要素並列ジョブはこのノードで実行します。)複数のプロセッサを使用するしないにかかわらず、スカラー計算時間の8倍のCPU時間を占有することになるので、課金はノード当たりの占有時間(経過時間)の8倍をCPU時間とみなして単価を適用します。

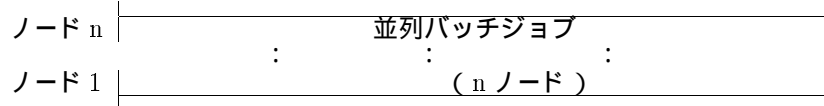
* インタラクティブジョブ

ログインノードについてはセッション時間(=ノードの占有時間)の8倍をCPU時間とみなして課金します。ここからに新たなノードを確保し、プログラムを実行(prun, mpirun等)した場合は確保したノードの占有時間合計の8倍をCPU時間とみなして追加分を課金します。



* バッチジョブ

バッチジョブの実行中は利用者が指定した数のノードを確保しますのでジョブの実行時間(=ノード当たりの占有時間)の8倍をノード数倍したものをCPU時間とみなして課金します。



• ファイル課金

上記 MP5800 のファイル課金と同様です。

3. 超並列型スーパーコンピューター SR2201; scalar-mpp

• 専有に係わる額

	月額 1,000 円定額	月額 2,000 円定額
(同時に占有可能な最大 PE 数)		
64PE まで	9,000 円/月	8,000 円/月
128PE まで	29,000 円/月	28,000 円/月
129PE 以上	69,000 円/月	68,000 円/月

• ファイル課金 /home, /short

利用者が指定したファイル量の上限值まで使用しているものとして以下の単価を適用します。

– 長期保存ファイル /home

(ファイル使用量上限値)	
~ 10MB	100 円 / (MB・月)
10 ~ 50MB	60 円 / (MB・月)
50MB ~	30 円 / (MB・月)

– 短期保存ファイル /short

10 円 / (MB・月)

利用負担金についての最新の情報は、スーパーコンピューティングニュースの表紙裏、または Web ページ <http://www.cc.u-tokyo.ac.jp> を参照して下さい。