UNIXの簡単な使い方

大阪大学 基礎工学部 情報工学科　藤川 和利、山口 英
大阪大学 大型計算機センター　馬野 元秀

1　はじめに

本稿では、UNIXでよく使われるviというエディタを用いてC言語によりプログラムを作成し、実行するという過程を通して、UNIXの簡単な使い方を紹介する。以下、viによるCのプログラムの作成・実行の方法と基本的なファイル管理機能の使い方について述べる。

2　大型計算機センターのワークステーションに接続するには

大型計算機センターのワークステーションを利用するには、センター内で使う方法とセンター外から交換回線(電話)または専用回線により使うことができる。

センター外からは現在のところ、ポートセレクタ経由でしか使うことができない。したがって、交換回線とポートセレクタ経由の専用回線から利用できる。

まず、センターと接続し、以下のように入力する。

```
<CR>                                ..... リターン・キーのみを押す

*** COMPUTATION CENTER OSAKA UNVERSITY ***

CLASS          S Y S T E M          BPS

  1          ACOS&SX(LEVEL0)        1200
  2          ACOS&SX(LEVEL0)        2400
  3          ACOS&SX(LEVEL0)        9600
  4          ACOS&SX(LEVEL2A)       2400
  5          WORKSTATION            1200
  6          WORKSTATION            9600

ENTER CLASS 5                       ..... 使っている回線速度に応じて、5 または 6
```

```
GO
<CR>                                    …… リターン・キーのみを押す
Machine-name Changed(88/7/13) sun260->ccsun01 sun50a->ccsun02...ccsun05

telnet> open ccsun01                    …… 使いたいワークステーションを選ぶ
Break-in character is BREAK
Trying to make connection...
[Open]

4.2 BSD UNIX (ccsun01)

login: a60000a                          …… 利用者番号
Password:                               …… パスワード
Last login: Wed Aug 17 15:51:16 from ccews01
Sun UNIX 4.2 Release 3.4EXPORT (GENERIC) #1: Thu Apr 30 09:36:18 PDT 1987

          ***********************************************
          ****  Computation Center, Osaka University  ****
          ***********************************************

   *** News from Center ***

    <センターからのニュースが表示される>

ccsun01%
```

このようにして、UNIXをつかう準備ができた。なお、プロンプトは上のように
ワークステーション名%
となり、使用するワークステーションごとに異なるので、以下では%だけとする。

センター内からはワークステーションのコンソールを直接使用することになる。通常 loginのプロンプトが出ているので、利用者名とパスワードを入力すればよい。

利用者がログインすると、UNIXはシェルと呼ばれるコマンドインタプリタを起動する。シェルにはいくつか種類があるが、普通はもっとも一般的なCシェル(csh)が起動される。なお、システムの使用を終えるには、コマンド logoutまたは exitを用いて、センター外からは、

```
% logout                                …… ワークステーションの利用の終了
[Remote Close]
[Closed]
telnet> q                               …… 接続を切る
```

とすればよい。センター内の場合は、logoutを入力するだけでよい。

3　C言語によるプログラミング

さて、実際に UNIX上で C言語の簡単なソースファイルを作成し、コンパイルして、実行してみよう。

3.1　ソースファイルの作成

ソースファイルを作成するには、エディタを用いる。UNIX上には、いろいろなエディタがあるが、ここでは最もポピュラーな viを用いよう。プログラムは、次のように出力するものを作ることにする。

```
hello, world
```

コマンドを入力して viを起動する。

```
%vi test.c
```

この場合作成するソースファイルの名前は test.cとする。UNIXにおいて C言語のソースファイルは、拡張子として最後に.cをつけておくのが普通である。すると、viの初期画面は、図 1のようになっている。

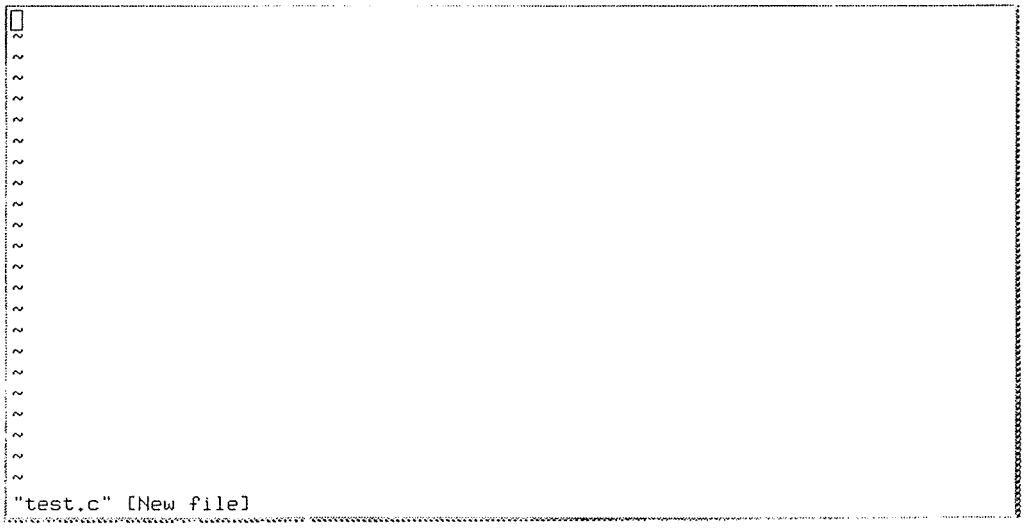

図1 viの初期画面

今、viの画面となっているので、早速プログラムを入力してみよう。しかし、viには、コマンドモードと入力モードの 2つのモードがあり、最初はコマンドモードになっている。したがって、まず iを押すと入力モードに変わる。では、次のようなプログラムを入力しよう。

```
main()
{
        printf("hello, world\n");
}
```

なお、入力ミスをした場合は、BACKSPACEキー(または、DELETEキー)で今入力した文字を削除できる。入力し終わったら、ESCキーを押す。すると、コマンドモードに戻る。このコマンドモードで大文字のZを2回押すと、ファイルをセーブして、viが終了する。

3.2 実行可能形式のファイルを得る

次に、このソースファイル test.cをコンパイルしよう。この場合、次のようにコマンドを入力する。

```
%cc test.c
```

すると、a.outという実行可能形式のファイルができる。ccは UNIX上の C言語のコンパイラである。そして、

```
%a.out
hello, world
```

のように、hello, worldと出力される。つまり、a.outという実行形式のファイル名を入力すると、それが実行されるわけである。

3.3 ソースファイルの編集

次に、既存のソースファイルを編集する場合を考えよう。ここでは、先ほど作成した test.cを次のように変更する。

```
main()
{
        printf("hello, ");
        printf("world");
        printf("\n");
}
```

このように変更するには、viのコマンドモードでのコマンドをいくつか知っておく必要がある。ここでは、簡単なコマンドを紹介しよう。入力モードからコマンドモードにするには ESCキーを押せばよい。

カーソル移動コマンド

h ： 1文字左へ移動
l ： 1文字右へ移動
j ： 1行下へ移動
k ： 1行上へ移動

これらのコマンドは、カーソル移動において最も基本的なものである。しかし、ファイルが大きくなって1画面では表示できない場合、h,j,k,lのみでカーソルを移動させるのはかなり面倒なものになる。このとき、便利なコマンドを紹介しよう。

カーソル移動コマンド

w ： 次の語の先頭へ移動
b ： 前の語の先頭へ移動
e ： 語の先頭へ移動
^ ： 行の先頭へ移動
$ ： 行の最後へ移動
H ： 画面の最初の行へ移動
M ： 画面の中央の行へ移動
L ： 画面の最後の行へ移動
CTRL-F ： 次の画面へ移動
CTRL-B ： 前の画面へ移動
数字+G ： ファイルの先頭からその数字行目へ移動

CTRL-Fは、CTRLキーを押しながらfを押す(CTRL-Bも同様)。数字+Gは、数字を入力してからGを押す。数字を入力しない場合は、ファイルの最後へ移動する。

文字の入力や消去といった編集用には、次のようなコマンドがある。

編集コマンド

x ： カーソルのある文字を消去
dd ： カーソルのある行を消去
i ： 入力モードになる(カーソルのある位置に挿入)
a ： 入力モードになる(カーソルの右に挿入)
o ： 入力モードになる(カーソルのある次の行に挿入)

これだけ知っていれば、ソースファイルを修正することができる。なお、入力モードでCRキーを押すとそこで行が分割される。

viにはコマンドモードと入力モードの2つのモードがあるため、初めて使う人は(かなり使っている人でも)、コマンドモードで文字を入力しようとしたり、入力モードでコマンドを使おうとすることがしばしばある。viを使うときは、現在どちらのモードであるかを覚えておく必要がある。(わからなくなったときは、ESCキーを押すとコマンドモードになる。ESCキーは何回押してもよい。)

さらに、文字列を検索したい場合は、次のようなコマンドを用いる。

検索コマンド

f ：次に入力する1文字を前方(行の後ろの方)へ検索
F ：次に入力する1文字を後方へ検索
; ：fまたはFの繰り返し
/ ：次に入力する文字列を前方(ファイルの後ろの方)へ検索
? ：次に入力する文字列を後方へ検索
n ：/または?の繰り返し

/(または?)の場合は、画面の1番下に/(または?)が表示され、カーソルがそこに移動するので、検索したい文字列を入力してCRキーを押せばよい。
この他にも便利なコマンドがあるので、もっと知りたい人は各自で調べて頂きたい[1]。

3.4 viの中止

コマンドモードと入力モードを間違えて誤った操作をしてしまったりして、ソースファイルがおかしくなった場合、何の変更もしなかったものとしてviを終了することができる。その場合には、次のようにする。

まず、コマンドモードにする(ESCキーを押す)。そして：を押すと
画面の一番下の行に：が現われる。
ここで、q!と入力してCRキーを押す。

すると、viは終了しファイルは元の状態のままである。つまり、コマンドモードで

:q!⟨CR⟩

と打てばよいわけである。

3.5 実行結果を見てみよう

ソースファイルtest.cを変更できたと思う。では、コマンドモードにして、大文字のZを2回押して、viを終了しよう。
そして、

```
%cc test.c
```

で、実行可能形式のファイルa.outができ、

```
%a.out
hello, world
```

のように、hello, worldが実行結果として得られる。

ここまで述べてきたようにすると、プログラムを作成して実行結果を得ることができる。しかし、これではどのソースファイルに対しても実行可能形式のファイルは、a.outとなってしまう。そこでccには、実行可能形式のファイルを指定するオプション-oがあり、test.cの実行可能形式のファイルをtest1としたい場合は、次のようにすればよい。

```
%cc test.c -o test1
```

このようにするとプログラムごとに違った実行可能形式のファイルを作成することができる。

また、Cコンパイラ(cc)には、まだまだ多くのオプションがある。このオプションについては、文献[1]を参照されたい。

4 基本的なファイル管理

前節では、ソースファイルを作成・編集し、実行可能形式ファイルを得て、実行結果を見ることを行った。ここでは、ファイルを管理する方法について述べよう。

4.1 どのようなファイルがあるのだろう?

まず、どのようなファイルがあるかは、コマンドlsで調べることができる(lsはlistの略)。

```
%ls
a.out    test.c   test1
```

これより、a.out、test1、test.cの3つのファイルがあることがわかる。ファイルはアルファベット順に表示される。ファイルが少ない場合はこれでも良いが、テキストファイル、実行可能形式ファイルが入り混じって存在している場合は区別するのが大変になってくる。これを区別するために、-Fというオプションがある。

```
%ls -F
a.out*   test.c   test1*
```

test.cのようなテキストファイルはそのまま表示され、a.outのような実行可能形式ファイルはファイル名の最後に*が付く。以下で述べるディレクトリの場合は、最後に/が付く。

4.2 ディレクトリとは?

ディレクトリとは、ファイルをいくつか集めたもので、ACOS-6のカタログと同じ概念である。ログインした時点で利用者には、決められたディレクトリが割り当てられる。このディレクトリをホームディレクトリと呼び、利用者は、普通このディレクトリの下で作業を進めていく。あるディレクトリAには、ファイルの他にディレクトリBがあってもかまわない。この時、ディレクトリAからみると、ディレクトリBは「子ディレクトリ」となり、反対に、ディレク

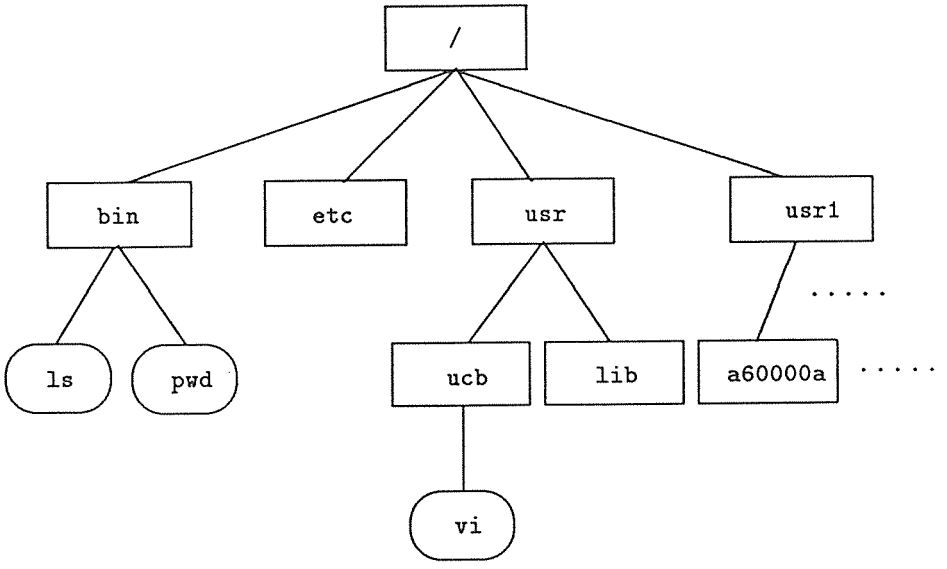

図2 ディレクトリの構造

トリ Bからみるとディレクトリ Aは「親ディレクトリ」になる。lsに-Fオプションに加えて-aオプションをつけてみよう。-aオプションはディレクトリの内容をすべて表示することを指定する。

```
%ls -aF
./      ../     a.out*  test.c  test1*
```

これを見るとディレクトリ.と..があります。.は自分自身を表す名前であり、..は親ディレクトリを表す。このことは、どのディレクトリでも同じである。また、ホームディレクトリは~で表すことができます。

すべてディレクトリから親ディレクトリへと順にたどっていくと、ディレクトリ/にたどりつく。/は、親ディレクトリがなく、このディレクトリをルートディレクトリと呼ぶ。つまり、UNIXでは/をルート(根)としたツリー構造がディレクトリによって構成されているわけである(図2)。

これをみると、UNIXのコマンドはすべて適当な場所にある単なる実行可能形式ファイルであることがわかる。

4.3 ディレクトリを作ろう

ファイルの数が増えて、自分のディレクトリにいろいろな種類のファイルができてきたとしよう。これは、自分の机の上に本や書類が氾濫した状態と似ている。では、机の上の本を整理す

るように、ファイルを整理して保存してみよう。

今、プログラムのソースファイル、実行可能形式ファイル、文書ファイル等が存在している場合を考え、これらを種類ごとに整理しておくことを考えよう。まず、ディレクトリを作る必要がある。これには、コマンド mkdirを用いる。

```
mkdir ディレクトリ名
```

実行可能形式ファイル用のディレクトリは、bin(binaryの略)という名前にし、ソースファイル用は src(sourceの略)、文書ファイル用は doc(documentの略)というようにしよう。

```
%mkdir bin
%mkdir src
%mkdir doc
```

これで、bin、src、docというディレクトリができた。実際にそれらがあるか確かめてみよう。

```
%ls -F
a.out*   bin/     doc/     src/     test.c   test1*
```

4.4 ファイルのコピー、移動、消去

では、実行可能形式ファイル test1をディレクトリ binに移してみよう。

```
%cp test1 bin/test1
```

このようにするとディレクトリ binに test1がコピーされる(cpは copyの略)。また、この場合は次のようにしても同じようにコピーされる。

```
%cp test1 bin
```

これは、binがディレクトリであるため自動的に同じファイル名でコピーするからである。では、ディレクトリ binに移って確かめてみよう。そのためには、cdコマンド(change directoryの略)を用いる。

```
%cd bin
%ls -F
test1*
```

このように UNIXでは、現在どのディレクトリにいるかということが重要である。このディレクトリをカレントディレクトリと呼ぶ。loginした時のカレントディレクトリは、ホームディレクトリになっている。

現在どのディレクトリにいるのかを知るには、コマンド pwdを使う。

```
%pwd
/usr1/a60000a/bin
```

なお、ディレクトリの指定ではカレントディレクトリからみた相対的なディレクトリを指定するので、binの前に/を付けないこと。
それでは、元のディレクトリに戻ってみよう。

```
%cd ..
%pwd
/usr1/a60000a
```

では、次のようにしてみよう。

```
%ls -F bin
test1*
```

先ほどと同じ出力が得られる。このようにすると、ディレクトリを移らずに別のディレクトリの内容を確かめることができる。もし、binの下にディレクトリ prog1があれば、ls -F bin/prog1とすればよい。
cpは元のファイルをコピーするので、元のファイルはそのまま残っている。元のファイルは、もう不要なので消去しよう。ファイルを消去するコマンドは rmで、

```
%rm test1
```

とすればよい。rmで1度ファイルを消去したらそのファイルは2度と回復することができない。rmを使う時は注意する必要がある(特に、今どのディレクトリにいるかに注意しよう)。
今の例では、ファイルを移すのにコピーして消去するという手段をとった。ここでは、本当にファイルを移すことのできるコマンド mvを使ってみよう。では、test.cをディレクトリ srcに移してみる。

```
%mv test.c src/test.c
```

この mvも cpと同様に次のように行なっても同じである。

```
%mv test.c src
```

今はホームディレクトリにいるから、

```
%ls -F
a.out*   bin/     doc/     src/
```

となって、test.cと test1はもうこのディレクトリにはない。a.outは不要だから消しておこう。

```
%rm a.out
```

また、ディレクトリを消去する時は、コマンド rmdirを用いる。では、ディレクトリ binを消してみよう。

```
%rmdir bin
rmdir:  bin:  Directory not empty
```

この時は、binに test1というファイルがあるので、このようなメッセージが表示された。rmdirは、ディレクトリが空でなければ実行できない。したがって、次のようにすればよい。

```
%rm bin/test1
%rmdir bin
```

rm bin/test1は、ディレクトリ binの下の test1というファイルを消すことを意味する。

ここまで述べてきたことで、ファイルの作成、編集、消去、コピー、移動、ディレクトリの作成、消去、カレントディレクトリの変更が行える。

4.5 ファイルの内容の表示

ファイルの内容を見るにはどうすればよいか？。前節で述べたスクリーンエディタ viを用いることが考えられる。しかし、ファイルを編集しないなら viを用いる必要はない。コマンド catを使って、ファイルの内容を見ることができる(catenateの略。もともとはファイルを結合するコマンド)。

```
%cat src/test.c
```

これにより、先ほど作成した test.cの内容が表示される。しかし、量の多いファイルをこのコマンドで見ようとすると、ファイルの内容が一度に表示されて、始めの方はスクロールされて画面の上の方に消えていってしまう。ファイルを順を追ってゆっくり見るには、コマンド moreを用いる。moreでは、1画面分を表示すると停止するので、1画面進むためには spaceキー、1行進むためには CRキー、表示を中断するには qを用いればよい。

ファイルの最初または最後の部分だけを見るコマンドもある。これらは、headと tailで、headはファイルの最初の 10行を、tailは最後 10行を表示する(表示行数はオプションで変更できる)。

以上挙げたコマンドについてまとめると次のようになる。

vi : スクリーンエディタ、テキストファイルの作成編集
cc : Cコンパイラ
ls : ディレクトリの内容表示
cd : ディレクトリの移動 (カレントディレクトリの変更)
rm : ファイルの消去
mkdir : ディレクトリの作成

rmdir : ディレクトリの消去
cp : ファイルのコピー
mv : ファイルの移動
cat : ファイルの内容の表示
more : ファイルの内容の表示 (1画面ずつ)
head : ファイルの最初の部分の表示
tail : ファイルの最後の部分の表示

これらのコマンドには、それぞれいろいろなオプションがある。UNIXでは、端末上でコマンドのマニュアルを見ることができる。lsのマニュアルを見てみよう。

```
%man ls
```

manのマニュアルも見ることができ、

```
%man man
```

とすればよい。

5 おわりに

以上、UNIXのもっとも基本的な使い方について述べた。とりあえずCのプログラムを作成し、実行するという点に重点をおいたため、UNIXの特徴である入出力の切り換え、パイプラインについては述べなかった。さらに詳しくは、マニュアルや参考文献を参照されたい。

参考文献

[1]栗原、藤崎、小幡:マイコン UNIXの使い方 廣済堂産報出版(1983)。
[2]村井、井上、砂原:プロフェショナル UNIX アスキー(1986)。
[3]B.W.Kernighan,R.Pike,石田訳:UNIXプログラミング環境 アスキー(1985)。