Linux HOWTO

F. Miyahara

2006 4/10

Linux の基本的な使い方を箇条書きで簡単に説明します。今まで UNIX をほとんど使ったことの無い人を対象にしています。知っておいたほうが良いと思うコマンド等をならべました。Linux の入門書等は詳しいものが多く出回っているのでここではコマンドの書式と例、簡単な説明のみを書いておきます。本文中で `$` がついている行がコマンドの実行例です。コマンドは `$` 以下を `kterm` や `xterm`、`gnome-terminal` 等の端末エミュレータ (以下ターミナルと呼ぶ) から入力し、実行します。各コマンドには様々なオプションがありますがここでは一部のオプションのみを紹介します。具体的にどのようなオプションがあるかは、コマンドのマニュアルを見て下さい。またここで紹介するコマンド等は一部の `PC` ではインストールされていない場合があるので、そのときは管理者に相談して下さい。

1. 基本的なコマンド

コマンドはターミナルから実行します。コマンドの実行結果をターミナル内に表示するコマンド (`ls` 等) の場合はそのコマンドを入力してエンターキーを押します。またウェブブラウザやメーラー等のアプリケーションを起動する場合はコマンドの後に `&` を入力してエンターキーを押して下さい。

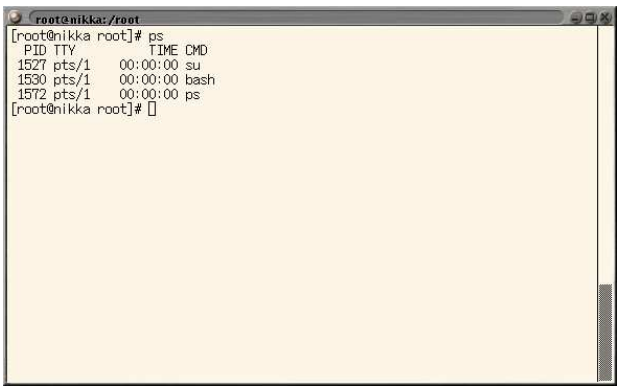

図 1: kterm で ps コマンドを実行した様子

◆ コマンドのマニュアルを読む

1）コマンドマニュアルを読む

書式: man [コマンド名]　　（日本語マニュアルは jman）

例 1: $ man ls

⇒ ls コマンドの使い方を読む

例 2: $ man 3 printf

⇒ C 言語の関数 printf のマニュアルを読む。ここで数字の 3 は参照するセクションを指します。man ページは複数のセクションに分類されており、セクションを指定することでそのセクションのみを参照します。セクションは以下のような分類がなされています。

1:　一般コマンド
2:　UNIX システムコール
3:　ライブラリ関数
4:　周辺装置関連
5:　ファイル形式
6:　ゲーム
7:　共有ファイル、マクロ関係
8:　システム管理コマンド

ほとんどのコマンドにはマニュアルが用意されています。コマンドの使い方が分からない場合はこのマニュアルを読んで解決しましょう。まずは man コマンドの使い方 man man を読んでみて下さい。特に -k オプションは便利なので意味を調べてみると良いでしょう。

◆ ファイル、ディレクトリーの操作

2）ファイルを表示

書式: ls [options]　[ファイル名 or ディレクトリのパス]

例 1: $ ls /etc

⇒ etc ディレクトリにあるファイルとディレクトリを表示
ディレクトリを指定しなければカレントディレクトリ（自分が現在いるディレクトリ）のファイルを表示します

例 2: $ ls -lta /tmp　（ls -l -t -a /tmp と同じ）

⇒ /tmp ディレクトリにあるファイルを表示

-l:　ファイルの詳細情報（アクセス権、所有者、file size, 作成時刻等）を表示
-a:　「.」から始まる .bashrc 等の隠しファイルを表示 。
-t:　更新時刻順に並べる（順番を逆にしたければさらに -r オプションをつける）。

また --color=auto オプションを付けるとディレクトリや特定のファイルの表示に色が付きます。

3）現在のディレクトリを表示

書式: pwd

例 1: $ pwd

⇒ 現在自分がいるディレクトリ (カレントディレクトリ) が表示される。ディレクトリとは Windows でいうフォルダのこと。

4）ディレクトリを作成

書式: mkdir [作りたいディレクトリのパス]

例 1: $ mkdir foo

⇒ カレントディレクトリにディレクトリ foo を作成

5）ディレクトリを移動

書式: cd [移動先のディレクトリ]

例 1: $ cd /etc/

⇒ /etc ディレクトリに移動。
ディレクトリを指定しなかった場合はホームディレクトリに戻ります。
また、cd - で直前にいたディレクトリに戻ります。

Linux では個人のディレクトリやシステムの設定ファイル等が適当なディレクトリに納められています。ディレクトリは役割ごとに分類されており、 tree 構造をとっています。幹にあたるのが / ディレクトリで各個人のディレクトリ (ホームディレクトリ) は /home 以下に納められています。

6）ファイルをコピー

書式: cp [options] [コピーしたいファイル名 or ディレクトリ] [コピー先]

例 1: $ cp /etc/profile ./

⇒ /etc/profile をカレントディレクトリにコピー
(「 . 」はカレントディレクトリーという意味)

例 2: $ cp -r /home/miyahara/.xemacs ~/

⇒ /home/miyahara/.xemacs をディレクトリごとホームにコピー
(「 ~/ 」はホームディレクトリーという意味)

7）ファイルの移動 or ファイル名を変更

書式: mv [移動したいファイル or ディレクトリ] [移動先]

例 1: $ mv hoge.txt foo

⇒ hoge.txt を foo に移動

例 2: $ mv hoge.txt hogehoge.txt

⇒ hoge.txt を hogehoge.txt に変更 (ファイル名の変更)

8）ファイルの削除

書式: rm [options] [削除したいファイル]

例 1: $ rm hoge.txt

⇒ hoge.txt を 削除

例 2: $ rm -rf foo

⇒ ディレクトリ foo をまるごと削除。Linux は Windows のように一度ゴミ箱に移すという操作は行わないため消したファイルは簡単には元に戻せないので注意。すぐに対処（まずはディスクを umount）すれば復旧は可能だが面倒。大事なファイルはバックアップが肝心。

9）テキストファイルをターミナル上で読む or 出力を数行ごとに表示。

書式: less [ファイル名]

例 1: $ less hoge.txt

⇒ hoge.txt の中身を読む

例 2: $ ps -ef | less

⇒ プロセスを数行ずつ表示。
ここで | はパイプ（9 ページ）という機能で、ps の出力を less に渡す役割を担っている。

10）テキストファイルから特定の文字列を探す

書式: grep [options] ”[探したい文字列]” [ファイル名]

例 1: $ grep ”printf” main.c

⇒ main.c から printf を含む行を探し、表示する。
注:- や [等はメタキャラクタと呼ばれるそれ自体に意味を持つ文字なのでメタキャラクタを含む文字は直前にバックスラッシュを付けて \-、\[とする必要がある。（\ は環境によっては￥マーク になる）

11）ファイルを gzip で圧縮する

書式: gzip [options] [ファイル名]

例 1: $ gzip temp.txt

⇒ temp.txt を圧縮して temp.txt.gz に変換。

12）gzip で圧縮されたファイルを展開する

書式: gunzip [options] [ファイル名]

例 1: $ gunzip temp.txt.gz

⇒ temp.txt.gz を展開して temp.txt に戻す。

13）ファイルを bzip2 で圧縮（/展開）する

書式: bzip2(/bunzip2) [options] [ファイル名]

例 1: $ bzip2 temp.txt

⇒ temp.txt を圧縮して temp.txt.bz2 に変換。

14）アーカイブ（複数のファイルをまとめたファイル）を展開する

書式: tar [options] [ファイル名]

例 1: $ tar zxvf tmp.tar.gz

⇒ gzip で圧縮されたアーカイブ（tar.gz がつくファイル）を展開

例 2: $ tar jxvf tmp.tar.bz2

⇒ bzip2 で圧縮されたアーカイブ（tar.bz2 がつくファイル）を展開

15）ファイルのアクセス権を変更する

書式: chmod [options] [ファイル名]

例 1: chmod 755 a.out

⇒ 自分に「書き込み（2)+実行（1)+読み込み（4）」=7 を許可し、
同グループと他人には「実行（1)+読み込み（4）」=5 を許可
Linux のファイルにはそのファイルの所有者（ユーザー ID）とグループ（グループ ID）が与えられている（ls -l で確認）。各ユーザーは少なくとも一つのグループに所属しているはず。こうすることで個人、グループ、その他でファイルのアクセス権を区別できる。chmod の 3 つ並んだ数字の始めが所有者、次がグループ、最後がその他に対応し数字の値がアクセス権を表す。アクセス権は

読み込み=4
書き込み=2
実行=1

の合計で与えられる。各自のユーザー ID は /etc/passwd、グループ ID は /etc/group で確認できる。アクセス権の確認 ls -l の見方
-rwxr-xr-x 1 miyahara miyahara 6596 Mar 10 18:21 a.out
赤色の部分が所有者、青色がグループ、緑色がその他に対するアクセス権。r は read, w は write, x は execute を許可するという意味。

◆ システムコマンド

1）プロセスを表示

書式: ps [options]

例 1: $ ps -ef

⇒ 全てのプロセスを表示
-e: 全てのプロセスを表示
-f: プロセスの詳細を表示

2）プロセス（プログラム）を終了させる

書式: kill [options] [PID(プロセス番号)]

例 1: $ kill -9 13630

⇒ PID 13630 を強制終了
-9: プロセスを強制終了
アプリケーションから応答が無くなった (通常の方法で終了できない) 場合は ps コマンドでそのプログラムの PID を調べ、強制終了する。

例 2: $ kill -s SIGSTOP 13631

⇒ PID 13631 を一時停止 (bash, zsh)。kill コマンドは実際にはプロセスにシグナル（シェルのマニュアルを参照) を送るコマンド。kill -s SIGCONT 13631 でプロセスを再開できる。SIGSTOP, SIGCONT は便利なので覚えておくと良い。

3）Linux のタスクを表示する

書式: top [options]

例 1: top

⇒ CPU、メモリの使用率等を表示。q で終了、h でヘルプ。

◆ bc で簡易計算

書式: bc -l
ターミナル上で簡単な計算ができます。 -l オプションを付けることで数学関数を使うことができます。

```
----- 実行例 ----
256^2
65536
sqrt(65536)
65536
4.0*a(1.0)
3.14159265358979323844
-----------------
```

使える関数は s(x):sin(x), c(x):cos(x), a(x):atan(x), l(x):log(x), e(x):exp(x), j(n,x):整数 n 次のベッセル関数。

◆ 印刷

Linux で何かを印刷するときは mozilla や firefox 等のアプリケーションから直接印刷するか、一度ポストスクリプトファイル（PS ファイル）を作ってから印刷する。慣れないうちは無駄な印刷をしないために gv（or ggv，evince）等で PS ファイルを表示、確認してから印刷して下さい。

1）テキストファイルの印刷

書式: a2ps [options] [ファイル名]

例 1: $ a2ps -Pprinter1 hoge.txt

⇒ hoge.txt をプリンタ printer1 に印刷
-P[プリンタ名] でプリンターを指定。-P の後はスペースを入れないこと。
プリンターの一覧は??ページ。

例 2: $ a2ps -4 hoge.txt > tmp.ps

⇒ テキストファイルから PS ファイル（a2ps 形式）を作る
-4: -n で 1 ページに n 枚をおさめることができる。デフォルトは -2
>: リダイレクトでファイルに出力。（9 ページ参照）

2）PS ファイルの表示

書式: gv [PS ファイル名]

例 1: $ gv tmp.ps & （gv がない場合は ggv または evince ）

⇒ tmp.ps を gv で開く。gv & で起動した後で開きたいファイルを選択しても良い。

3）PS ファイルの印刷

書式: lpr [options] [PS ファイル名]

例 1: $ lpr -Plp1 tmp.ps

⇒ tmp.ps をプリンター lp1 に印刷。
-P[プリンタ名] でプリンターを指定。-P の後はスペースを入れない。

4）使用できるプリンタの状態を表示する

書式: lpstat [options]

例 1: $ lpstat -t

⇒ 登録されているプリンターの全ての情報を表示

5）PS ファイルを数ページまとめて印刷

書式: psnup [options] [PS ファイル名]

例 1: $ psnup -4 tmp.ps | lpr -Pepson3f

⇒ 4 ページを 1 枚にまとめて印刷。-[数] でまとめるページ数を指定。

6）プリンタジョブを確認

書式: lpq -P[プリンタ名]

例 1: $ lpq -Pepson3f

⇒ プリンターのステータスを表示。印刷のジョブ番号等が表示される。

7）印刷を中止する

書式: lprm -P[プリンタ名]

例 1: $ lprm -Pepson3f [ジョブ番号]

⇒ 印刷を中止。ジョブ番号は lpq で確認。印刷を行なったのに印刷されない場合もジョブを取り消す。
印刷失敗の状態をそのままほったらかすと他の人に迷惑がかかるので気を付けましょう。失敗の原因が分からないときは管理者に聞く。

◆ ssh, scp（リモートログイン）

他の PC にリモートログインし、その PC を操作したりプログラムを走らせたりすることができます。自分の利用している端末が何かの原因でキー入力等を受け付けなくなった場合等は別の端末からログインして問題解決を試みて下さい。(電源ボタンは最終手段)

8）他の PC にリモートログインする

書式: ssh [ユーザー名]@[リモートホスト名または IP アドレス]

例 1: $ ssh -Y hanako@ayame.local.hoge.ac.jp

⇒ アカウント名 hanako で ayame にログインする
-X で X forwarding を有効にする。OpenSSH のヴァージョンによってはでは -Y オプションが必要。X forwarding を有効にすることで接続先の PC のウィンドウを自分の PC の画面に表示できるようになる。(X Window System)

9）ファイルを別の PC に転送する

書式: scp [ファイル名] [ユーザー名]@[リモートホスト名]:[パス]

例 1: $ scp main.f hanako@ayame.local.hoge.ac.jp:/home/hanako/work

⇒ (ホームディレクトリを省略して scp main.f hanako@ayame:~/work としても良い。環境によっては接続先のドメイン名 (ここでは local 以下) を省略できる)
ファイル main.f をコピー先の PC ayame の /home/hanako/work に転送する。
ディレクトリを転送する場合は -r オプション。(cp とほとんど同じオプションが使える)。Windows の場合は WinSCP というプログラムを使えばクライアントとして利用可能です。

10）ジョブをバックグラウンドで走らせる

書式: [実行コマンド] >& [出力先のファイル名]
Ctrl+z を押してサスペンド (ターミナルにコマンド入力できる状態) にしてから
$ bg & を入力

例 1: $ ./sim >& log (この後 Ctrl+z)
$ bg &

⇒ 実行ファイル sim をバックグラウンドで走らせ、その出力をファイル log に渡す。
こうすることで接続先からログアウトしてもジョブが走り続ける。停止は kill コマンドを使う。

11) Windows PC にリモートデスクトップでログインする

書式: rdesktop [options] [リモートホスト名]

例 1: $ rdesktop -g 97% -a 24 [hogewin]

⇒ Windosw pc hogewin にログオンする。

◆ リダイレクト、パイプ

1) リダイレクト

UNIX では通常、プログラムの入出力が
標準入力 (stdin): キーボード
標準出力 (stdout): ディスプレイ
標準エラー出力 (stderr): ディスプレイ
で与えられています。リダイレクトを使うことで、これらの入出力を変更できます。リダイレクトの記号と機能は

記号	機能
<	標準入力をファイルにする (ファイルがキー入力の代わりになる)
>	標準出力をファイルにする (ファイルに出力結果が書き込まれる)
>>	標準出力をファイルに追加する
>&	エラー情報もファイルに出力

表 1:

2) パイプ (|)

パイプ (|) は ps -ef | less のようにある処理 (ps) の出力を別な処理 (less) に渡す役割を担います。

◆ テキスト処理

簡単なデータの処理やシェルスクリプト、プログラムの編集で役立つテキスト処理のコマンドを紹介します。テキストベースの処理がメインの場合は sed や awk，その他のコマンドの組み合わせのみで目的の処理を済ませられる事が多いです。より複雑な処理を行いたい場合は perl や ruby 等のスクリプト言語を使うと良いでしょう。

1) テキストファイル中の文字列の置換

書式: sed 's/置換したい文字列/置換後の文字列/g' [ファイル名]
(この形以外もあるが詳しくはマニュアル等を参照して下さい)

例 1: sed 's/windows/linux/g' samp.txt

⇒ samp.txt 中の文字列 windows を linux に置換し、標準出力に出力する。

2）awk

書式: awk [options] [ファイル名]

例 1: $ awk '$2>=70 {print}' score.dat
テキストファイル score.dat の 2 列目が 70 以上の行を標準出力に出力。
(列はオプションでセパレータを指定しなければスペースで区切られる)
テキストファイル score.dat が以下のとき

```
========== score.dat  ==========
Sato         58  95  79
Suzuki       75  90  71
Tanaka       65  88  83
Hashimoto    82  55  69
================================
```

コマンド awk '$2>=70 {print}' score.dat を実行した結果は

```
Suzuki       75  90  71
Hashimoto    82  55  69
```

となる。print の部分は C 言語の printf と同じ形をとることができる。

```
awk '{printf("%s %d\n$,$1,$2+$3+$4)}' score.dat
```

とすれば

```
Sato 232
Suzuki 236
Tanaka 236
Hashimoto 206
```

と表示される。

awk はコマンドと言うよりも pattern scanning and processing language と呼ばれる一つのプログラミング言語。より詳しいことは man page や awk に関する書籍、Web 上の解説等を参照して下さい。

◆ Tips

1）TAB 補完

TAB キーを押すことでコマンドの補完や、ファイル名の補完等ができます。例えば firefox を実行するときに fire まで打ち込んでから TAB キーを押すとコマンドが最後まで表示されます。補完候補が複数ある場合はもう一度 TAB キーを押すことでコマンド一覧が表示されます。

2）コマンド履歴

キーボードの [↑](or [Ctrl]+[P]),[↓](or [Ctrl]+[N]) キーを押すことで以前実行したコマンドを表示できます。

3）コピー＆ペースト

コマンドではありませんがマウスを使ったコピー＆ペーストの方法を覚えておくと便利です。方法はターミナルやエディター、Web ブラウザ等で表示されているテキストをドラッグして、テキストを張り付けたい位置でマウスの真中ボタンを押すだけです。

4）bash の設定
bash はコマンド言語インタプリタでコマンドを Linux に渡たす役割をしています。個人のホームディレクトリにある .bashrc ファイルを編集することでカスタマイズが可能です。(bash の他にも tcsh, zsh(高機能) があります。zsh は補完機能が bash よりも強力でホスト名の補完等 (ssh 等で使える) もやってくれるので興味があれば自分で開拓して下さい) デフォルトのシェルは chsh で変更できます。

4-1）パスの追加（bash, zsh）
bash はパスで指定されたディレクトリからコマンドを探してきます。パスが指定されていないディレクトリにある実行ファイルは絶対パスで直接指定しなければなりません。自分のディレクトリに何かをインストールしたときはパスを追加しておくと便利です。パスの追加は次のように行ないます。

```
PATH="$PATH:./"
PATH="$PATH:/usr/local/bin"
```

上記を .bashrc に追加。設定を反映させるには source ~/.bashrc を実行するか、ターミナルを再度立ち上げれば OK。

4-2）alias の追加（bash, zsh）
よく使用するコマンドに別名をつけることができます。

```
alias mule='xemacs'
alias rm='rm -i'
```

xemacs に mule という別名を付ける。rm コマンドを実行するときは常に -i オプションを使う。

5）ビープ音がうるさい
ビープ音を止めるには

```
$ xset b off
```

を実行して下さい。

6）前に実行したコマンドを呼び出す
前に実行したコマンドは [↑] キーで呼び出せるがこれだと履歴をさかのぼるのが面倒。ターミナル上で [Ctrl]+[R] を押したあと呼び出したい履歴の一部を入力すれば候補が表示される。例えば前に mplayer -fs -slang ja dvd://1 というコマンドを実行していた場合、[Ctrl]+[R] の後に探したいコマンドの一部 mpla と入力すれば候補が表示される。候補が複数ある場合はもう一度 [Ctrl]+[R] を入力するか、さらに入力を続け探したいコマンドの候補を絞る。探したいコマンドが見つかったらあとはエンターを押すだけ。長いコマンドを入力するのが面倒なときにも便利。(コマンドの履歴は複数のターミナルを起動していた場合は各ターミナルごとで記録されているので出てこないこともあります。zsh の場合はターミナルによらずコマンド履歴は共有されているので履歴の記録上限を超えない限り入力したコマンドを探すことが可能。)

2. エディターの操作

Linux で良く使われるテキストエディターは vi や emacs（or xemacs）です。ここでは emacs の簡単な使い方を紹介します。

新規にファイルを作成、または既存のファイルを開くときは
$ emacs［ファイル名］&
とします。xemacs のほうが emacs よりも多機能ですが、ほとんど好みの問題です。実行すると図 2 のように新しく emacs のウインドウが開きます。基本的な操作はウインドウの上にあるメニューから行なえます。emacs（xemacs）の操作は上のメニューをマウス

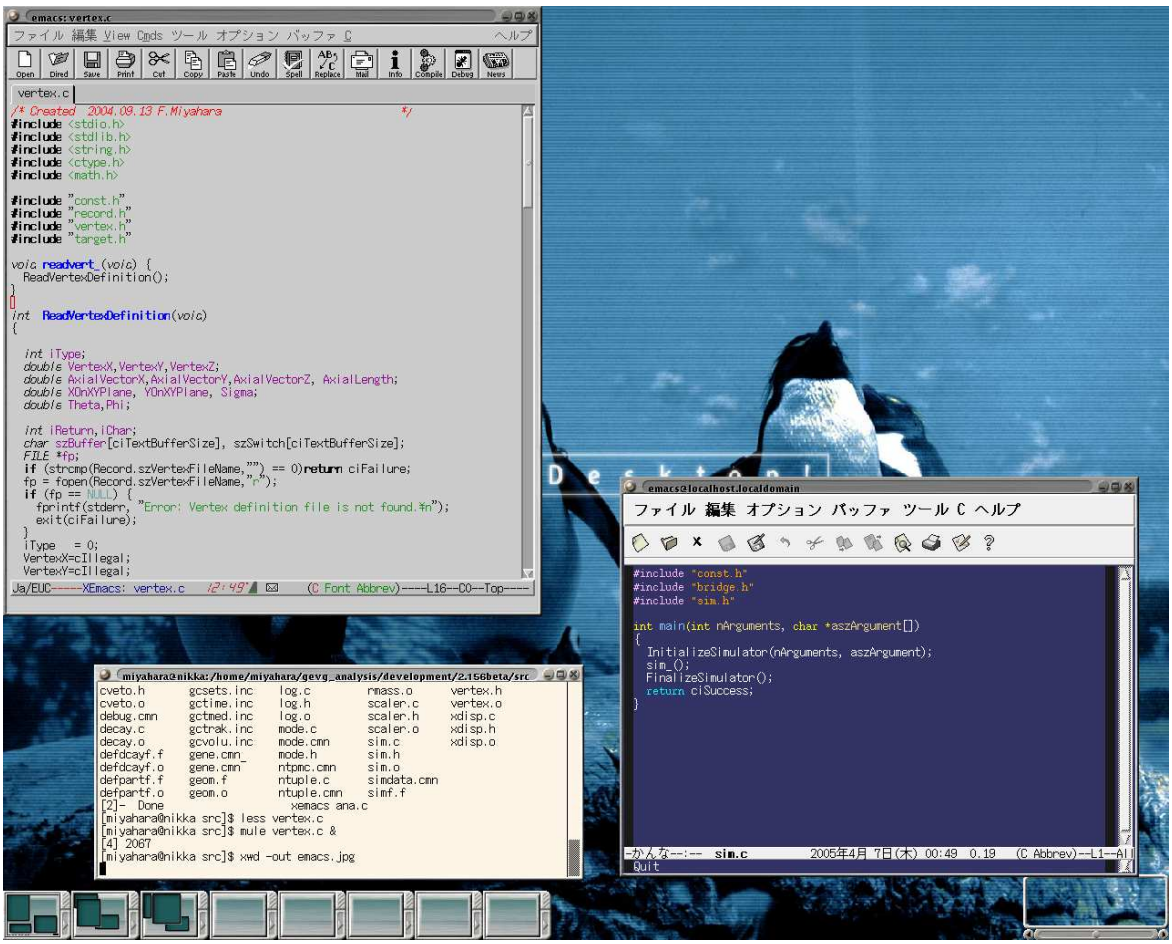

図 2: xemacs(左上) と emacs(右下) でファイルを開いた様子

でクリックすることで行なえますが、対応する機能は全てショートカットから実行できます。ショートカットを覚えればより効率的に作業が行なえます。GUI を使わないのであれば
$ emacs -nw［ファイル名］
とすることで、ターミナル内でファイルを開けます。-nw は ”no window” を意味します。こちらの方が動作が軽快です。

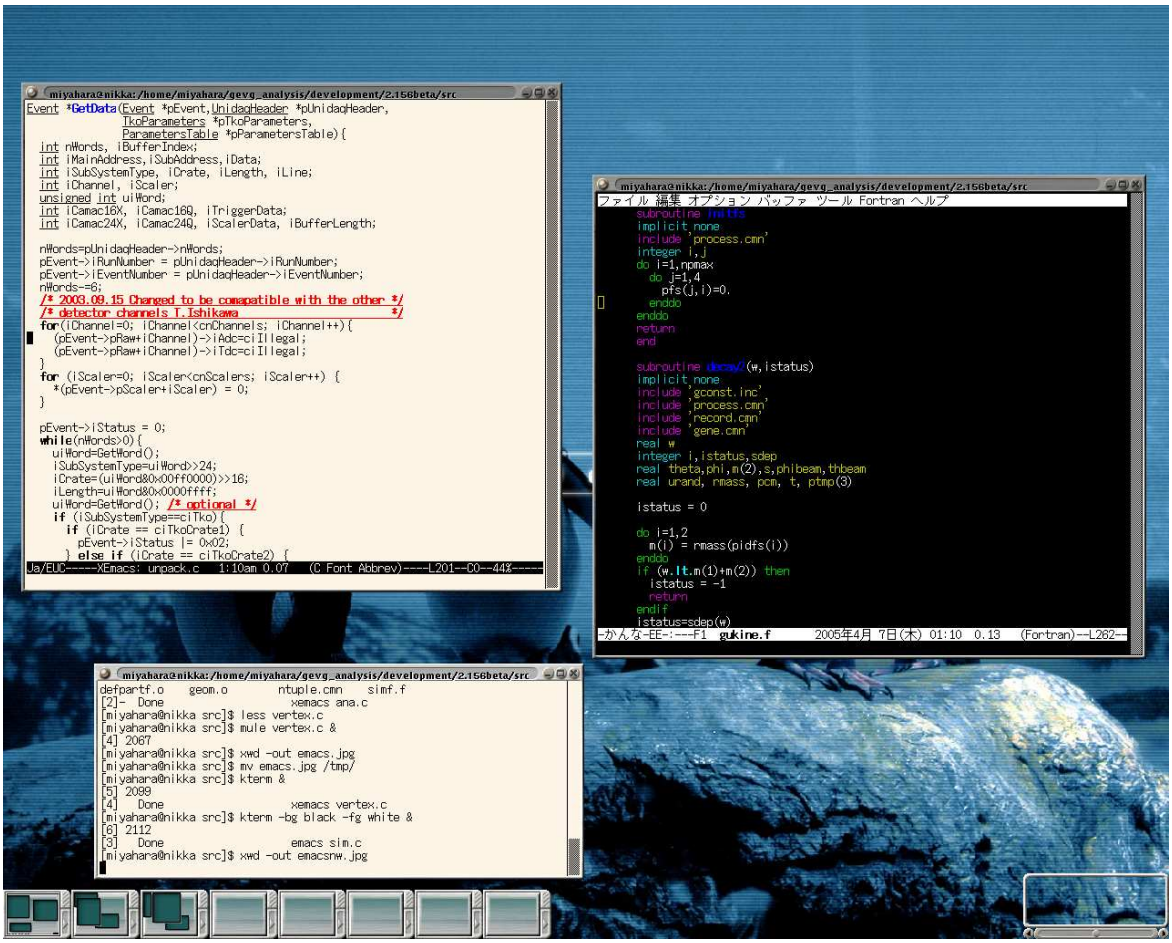

図 3: kterm 内で xemacs(左) と emacs(右) を -nw オプションで実行した様子。操作は全てキーボードで行なう (マウスによるコピー＆ペーストは可)。

良く使用するショートカットを表 2 にまとめました。ショートカットの一部 C-h 等は標準で使えない場合があります。この場合は emacs または xemacs の設定ファイルを編集する必要があります。xemacs の場合はホームディレクトリの .xemacs/init.el に次の行を追加すると C-h が BackSpace と同じになります (emacs なら .emacs 等)。

```
(global-set-key "\C-h" 'backward-delete-char)
```

設定ファイルにはショートカットキーの他にもいろいろな設定項目がありますがここでは触れません。また gnome-terminal 等の Alt キーに別な機能が与えられているターミナルでは "-nw" オプションで Alt キーを使ったショートカットが使えません。

また、 emacs のいくつかのショートカットはターミナル上の入力でも使う事ができるため、覚えておくとターミナルでの入力操作が楽になります。例えばコマンド入力中に Ctrl+A を押してコマンドラインの先頭にカーソルを移動したり Ctrl+D でカーソル上の文字を削除したりすることができます。Ctrl+R は過去のコマンド履歴から入力に一致したものを探してくれるのでいちいち履歴を遡ったり、同じコマンドを再び一から打ち直す必要がなくなるので便利です。

表記	意味
C- [キー]	Cntl を押しながら [キー] を押す。
M- [キー]	Esc を押した後に [キー] を押す。
M- [キー] [文字列]	M- [キー] の後、さらに Window 最下部の入力スペースに [文字列] を入力
Alt- [キー]	Alt を押しながら [キー] を押す。

ショートカット	ファイル操作
C-x C-s	ファイルの保存
C-x C-w	ファイルの別名保存
C-x C-c	emacs の終了
C-x C-i	ファイルの挿入
C-x C-f	ファイルを開く

ショートカット	カーソルの移動に関するショートカット
C-d	カーソル位置の文字を削除
C-h	Back Space と同じ (emacs or xemacs の設定ファイル依存)
C-a	行の先頭に移動
C-e	行の最後に移動
C-f	右に 1 文字移動
C-b	左に 1 文字移動
C-n	1 行下方に移動
C-p	1 行上方に移動
C-v	1 画面分カーソルを下へ移動
Alt-v	1 画面分カーソルを上へ移動
C-l	画面をクリアして再表示

ショートカット	その他
C-s	文字列の検索 (次の候補を探すときはさらに C-s)
C-r	文字列の検索 (逆向きに検索。(ファイルの先頭へ))
M-%	文字列の置換 (一語ずつ)
M-x replace-string	文字列の置換 (カーソルから後を一括で置換)
M-x replace-regexp	正規表現を用いた置換
C-k	カーソル位置から行末までをカット
C-x u or C-/	Undo (カットした後もとに戻すのによく用いる)
C-@ or C-SPC	mark を設定 (領域始点。この後にカーソルを止めた点が終点)
C-w	選択した領域をカット
C-y	貼り付け (← C-k, C-w に対応)
C-x r k	C-x → r → k の順にキーを押す。選択した領域を矩形カット (四角く切る)
C-x r y	矩形カットした領域を貼り付け
C-g	コマンドの取り消し (分けが分からなくなったらとりあえず C-g)
C-x 2	Window を縦に 2 分割
C-x 3	Window を横に 2 分割
C-x o	Window 間の移動
C-x 1	Window を 1 つに戻す
M-x indent-region	選択領域の C ソースコード自動インデント (行単位は Tab キー)

表 2: emacs のショートカット

3. プログラムのコンパイルと実行

1) C 言語のコンパイル
　書式: gcc [options] [ファイル名]
　例: $ cc samp.c -lm
　　⇒ samp.c をコンパイルし、実行ファイル a.out を作る
　　-lm : math.h ライブラリーを使用。 プログラムの実行は
　　$ ./a.out
　　オプション -o で実行ファイルの名前を指定することが可能。
　　指定しなかった場合は通常 a.out が生成される。

```
/**********  以下サンプルコード samp.c  **********/
#include <stdio.h>
#include <math.h>
int main(void)
{
  float beta,p,m;
  printf("Enter Momentum[MeV/c] & Mass[MeV/c^2]\n");
  scanf("%f",&p);
  scanf("%f",&m);
  beta= p/sqrt(p*p+m*m);
  printf("p=%f m=%f\n",p,m);
  printf("beta=%f \n",beta);
}
/******************** 終わり **********************/
```

2) FORTRAN 77 のコンパイル
　書式: g77 [options] [ファイル名]
　例: $ g77 -o w2e samp.f
　　⇒ samp.f をコンパイルし、実行ファイル w2e を作る
　　-o : 実行ファイルを指定

```
***********  以下サンプルコード samp.f  ***********
      program w2e
      real e,m,w
      m=0.938
      write(*,*) 'Enter w [GeV]'
      read(*,*) w
      e=(w**2-m**2)/m*0.5
      write(*,*) 'Eg=',e , ' [GeV]'
      end
******************** 終わり **********************
```

4. アプリケーション、ライブラリ

便利なアプリ、ライブラリをいくつか紹介。

- Web ブラウザ
  - **firefox**
    ほとんどのディストリビューションに初めからインストールされている。
  - **w3m**
    テキストブラウザ。ターミナル内で実行できる。HTML ドキュメントをターミナル内で読むのにも便利です。
- メイラー
  - **sylpheed**
    日本産のメイラー。メイルは一通ごとに別々のファイルに保存される。
  - **thunderbird**
    firefox の兄弟。メイルは一つのファイルに保存される。
- PDF Viewer
  - **acroread**
    Adobe Reader（Acrobat Reader）の Linux 版
  - **xpdf、evince**
    どちらか一方はたいていのディストリビューションに標準でインストールされている
- 画像編集
  - **gimp**
    画像処理プログラム。Photoshop のようなもん。
  - **tgif**
    回路図や簡単な図を書くのに便利
- ライブラリ
  - **lapack**
    線形代数計算の Fortran77 用ライブラリ。コンパイルは g77 samp.f -llapack -lblas の形。

5. 便利な Web ページ

@ IT，Linux Tips Index： いろんな Tips を紹介
http://www.atmarkit.co.jp/flinux/rensai/linuxtips/tipsindex.html
IT Media： Tips for Linux： いろんな Tips を紹介
http://www.itmedia.co.jp/help/tips/linux/
日経 Linux，Linux コマンド集： コマンドの説明
http://itpro.nikkeibp.co.jp/linux/command/index.shtml

索 引

a2ps，7

bzip2，4

cd，3
chmod，5
cp，3

emacs，12
emacs のショートカット，14

grep，4
gsip，4
gunzip，4
gv，7

kill，6

less，4
lpq，7
lpr，7
lprm，8
lpstat，7
ls，2

man，2
mv，3

ps，6
psnup，7
pwd，3

rdesktop，9
rm，4

scp，8
ssh，8

top，6

アーカイブを展開する，5

印刷，7

簡易計算 (bc)，6

現在のディレクトリを表示，3

ジョブをバックグラウンドで走らせる，8

タスクを表示，6

ディレクトリを移動，3
ディレクトリを作成，3

特定の文字列を探す，4

パイプ，9

ファイルのアクセス権，5
ファイルの移動，3
ファイルの削除 rm，4
ファイルを圧縮する，4
ファイルをコピー，3
ファイルを展開する，4
ファイルを表示，2
プロセスの強制終了，6
プロセスを表示，6

リダイレクト，9
リモートデスクトップ，9