

# 計算機システムの使い方

ver. 2019/4/19

計算物理学 2 は、サテライト室の端末を使います。まず、パソコンの電源を入れ、ubuntu (Linux) OS を選択し、ログインウィンドウに[ログイン名]と[パスワード]を順に入力して端末にログインして下さい。

計算物理学 2 の授業に必要な計算機システムの操作は

- ・ 端末の起動、基本的なコマンド操作
- ・ プログラムのソースコードを書くエディタ操作
- ・ プログラムのコンパイル
- ・ プログラムの実行
- ・ プログラムの結果の出力(図作成)

です。それぞれを以下で説明します。

## ○端末の起動

- ・ ubuntu を起動したデスクトップ画面で[Ctrl]+[Alt]+T を押すと端末が起動します。
- ・ デスクトップ画面で右クリックして「端末を開く(E)」を選択しても起動できます。
- ・ クイック起動メニューの「コンピュータを検索」で terminal と検索しても端末が起動できます。

## ○端末の(Linux の)基本的なコマンド操作

Linux 端末では基本的にキーボードを使ってコマンドを入力することによって各種操作を行います。また、Windows のフォルダに相当するファイルシステムは、ディレクトリという一番上のルートディレクトリ"/"から始まるツリー構造で管理されています。端末を起動すると通常はホームディレクトリと呼ばれる各ユーザー専用のディレクトリから始まります。ここでは、ターミナル上での基本的なコマンドとディレクトリに対する操作を説明します。基本的に必要なコマンドは、ファイル・ディレクトリを見る・作成する・消去する・移動するの4つです。

以下で\$はコマンド・プロンプトなので入力しないでください(\$の後ろの単語はコマンドであるということを示しています)。またコマンドを入力したらEnterキーを押すことで実行されます。

(A) ファイル・ディレクトリを見る

lsはディレクトリの中にあるファイルの名前の一覧を表示するコマンドです。

```
$ ls
a.out*  example.f90  output.txt  work/
```

pwd は今いるディレクトリを絶対パス(/から始まるツリー構造)で表示します (print working directory)

```
$ pwd

/home/hinohara.nobuo.ga      (これは私のホームディレクトリです)
```

(B)ディレクトリを作成する

mkdir はディレクトリを作成(make directory)し、 rmdir はディレクトリを消去する(remove directory)コマンドです。

```
$ mkdir ディレクトリ名
$ rmdir ディレクトリ名
```

のように使います。rmdir でディレクトリを消去する場合は、そのディレクトリは予め空になっていないと削除できません (後述のrm -r [ディレクトリ名] を使えば中身も含めて全て消去できます)。

(C) rm はファイルを消去するコマンドです(remove)

```
$ rm [ファイル名]
```

のように使う。ディレクトリも消したい場合は-rオプションをつけます。

```
$ rm -r [ディレクトリ名]
```

いずれの場合も一度消してしまったファイルは復元できないため操作ミスに注意してください。

(D) cp はファイルを複製するとき(copy)、mv はファイル名を変更する (move) ときに使うコマンドです。mvで既に存在するファイル名を指定するとともにあったファイルは無くなるので注意。

```
$ cp [ファイル名1] [ファイル名2]      (ファイル名1をファイル名2にコピーする)
$ mv [ファイル名1] [ファイル名2]      (ファイル名1をファイル名2と名前を変える)
$ mv [ファイル名] [ディレクトリ名/]    (ファイル名をディレクトリ名の下に移動する)
```

(E) cd は別のディレクトリに移動するために使用する。(change directory)

\$ cd ディレクトリ名

ディレクトリ名を省略するとホームディレクトリに移動します。

カレントディレクトリは“.”で表します。

ひとつ上のディレクトリは“..”を表します。

ホームディレクトリは“~”で表します。

例：カレントディレクトリにあるcompphys2ディレクトリに移動

\$ cd compphys2/ あるいは \$ cd ./compphys2/ (最後のスラッシュはなくてもよい)

例：一つ上のディレクトリに移動

\$ cd ../

例：2つ上のディレクトリに移動

\$ cd ../../

例：ホームディレクトリの下にあるDesktopディレクトリに移動

\$ cd ~/Desktop/

### ○演習

- ・ホームディレクトリの絶対パスをpwdで表示してみましょう。
- ・計算物理学2での課題はホームディレクトリ下にcompphys2ディレクトリを作ってここで作業することにしましょう。本日は1回目なので1stというディレクトリを作りその中に移動してください。(名前はこれらである必要はありません)

## Linux のコマンド一覧

	コマンド	機能・操作
ディレ クトリ	pwd	カレントディレクトリの表示(print working directory)
	cd	カレントディレクトリの変更(ディレクトリの移動, change directory) <b>利用法</b> : cd 移動先ディレクトリ
	mkdir	ディレクトリの作成 (make directory) <b>利用法</b> : mkdir 作成するディレクトリ
	rmdir	ディレクトリの削除 (remove directory) 利用法 : rmdir 削除する <b>空</b> のディレクトリ
ファイ ル	ls	ディレクトリ情報の表示 (list) -l オプション : 詳細情報の表示

		<p>-t オプション：作成時間順でソート</p> <p>-S オプション：ファイルサイズ順でソート</p> <p>-r オプション：逆順にソート</p> <p>-h オプション：ファイルサイズを</p> <p>-R オプション：ディレクトリの中まで辿って表示</p> <p><b>利用法 1:</b> ls</p> <p><b>利用法 2:</b> ls ディレクトリ</p> <p><b>利用法 3:</b> ls -l ファイル 1 ファイル 2 ...</p> <p>詳細情報の見方</p> <pre>drwxr-xr-x 16 hinohara.nobuo.ga staff 512 2 10 06:08 data/</pre> <p>第一コラム d はディレクトリであることを表す。</p> <p>rwrxrwx はそのファイル・ディレクトリのパーミッションを表す。</p> <p>r: readable(読み込み可能) w: writable(書き込み可能), x: executable(実行可能)。はじめの3つはファイル所有者の権限、次の3つはグループの権限、最後の3つは任意のユーザの権限を表す。</p> <p>パーミッションは chmod コマンドで変更できる。</p> <p>第二コラムは</p> <p>第3・4コラムはファイル所有者名、グループ名、第5コラムはファイルサイズ、第6コラム以降は最終更新日時を表す。</p>
cp		<p>ファイル、ディレクトリの複製</p> <p>-r オプション：ディレクトリ全体の複製</p> <p><b>利用法 1:</b> cp コピー元ファイル コピー先ファイル</p> <p><b>利用法 2:</b> cp コピー元ファイル 1 コピー元ファイル 2 ... コピー先ディレクトリ</p> <p><b>利用法 3:</b> cp -r コピー元ディレクトリ コピー先ディレクトリ</p>
rm		<p>ファイル、ディレクトリの削除</p> <p><b>削除したファイルやディレクトリは復活できないので注意して使用すること。</b></p> <p>-r オプション(recursive)：ディレクトリも削除</p> <p>-i オプション(interactive)：削除を実行する前に確認。安全のためデフォルトに設定することも多い。</p>

		<p><b>利用法 1:</b> rm 削除ファイル 1 削除ファイル 2 ...</p> <p><b>利用法 2:</b> rm -r 削除ディレクトリ 1 削除ディレクトリ 2 ...</p>
	mv	<p>ファイル、ディレクトリの移動、名前の変更</p> <p><b>利用法 1:</b> mv 変更前ファイル名 変更後ファイル名</p> <p><b>利用法 2:</b> mv 変更前ディレクトリ名 変更後ディレクトリ名</p> <p><b>利用法 3:</b> mv 移動 file1 移動ファイル 2 ... 移動先ディレクトリ</p> <p>変更後ファイル名と同名のファイルがすでに存在する場合は上書きされてしまうので注意</p>
表示・編集・検索	cat	<p>ファイルの内容を出力</p> <p><b>利用法 :</b> cat ファイル名</p>
	less	<p>ファイルの内容をページごとに出力</p> <p><b>利用法 :</b> less ファイル名</p> <p>スペースでスクロール</p>
	vi	テキストエディタ
	emacs	<p>テキストエディタ</p> <p>-nw オプション : 端末内で起動(別 window が開きません)</p>
	grep	ファイル検索
プログラム	gfortran (g95)	<p>Fortran90/95 プログラムをコンパイルして実行ファイルを作成</p> <p>-o オプション : 作成する実行ファイル名の指定。指定しないと a.out という名前の実行ファイルが作成される。</p> <p>-O オプション等 : プログラムの最適化。計算機によって異なる。</p> <p><b>利用法 :</b> gfortran コンパイルするプログラム名 (ex: file.f) -o 作成する実行ファイル名 (ex: file.exe)</p> <p>実行法 :</p> <p>% ./file.exe</p> <p>(%はコマンドプロンプトなので、打ち込まない事)</p>
	gcc	<p>C 言語プログラムをコンパイルして実行ファイルを作成。</p> <p>-o オプション : 作成する実行ファイル名の指定。指定しないと a.out という名前の実行ファイルが作成される。</p> <p>-O オプション等 : プログラムの最適化。計算機によって異なる。</p> <p><b>利用法 :</b> gcc コンパイルするプログラム名(ex: file.c) -o 作成する実行ファイル名(ex: file.exe)</p>

		実行法 : % ./file.exe (%はコマンドプロンプトなので、打ち込まない事)
--	--	---

## ○プログラムの作成

Fortran90 のプログラムは半角の英数文字、一部の半角記号で構成されます。日本語入力モードではなく、英数入力モードを使用して下さい。

### エディタの起動

プログラムはエディタで書きます。メモ帳などどのエディタを使っても構いませんが、ここではテキストエディタ Emacs の使い方を説明します。プログラムを作成したいディレクトリに移動した後(先程の演習では~/compphys2/1st)でした)に、Emacs の起動は端末で

```
$ emacs example.f90 &
```

とすると emacs が起動し、この場合だと example.f90 という指定した名前のファイルの編集ができます。同名のファイルがディレクトリ内に存在しない場合は新規作成となります。最後の&についてですが、基本的にコマンドを実行(emacs を実行)すると emacs を終了するまでその端末を使えませんが、&をつけて実行することにより、別プロセスとして emacs が実行されるため、**emacs が起動中も端末も同時に使うことができます。**

Fortran 90 のプログラムファイル名は example.f90 などのように.f90 という拡張子にしてください。(emacs で Fortran90 プログラムであると自動判別されて、見やすく表示されます)  
**ファイル名の途中にスペースを挟まないようにしてください。** 端末(シェル)はスペースで単語を区切って1つ目がコマンド、2つ目がファイル名などとして識別しています。

### Emacs の使い方

Emacs では Ctrl キーを押しながらアルファベットキーを押すことで様々な操作ができます。例えば Ctrl キーを押しながら x、s を順番に押す操作を Ctrl-x, Ctrl-s と表記します。

Emacs がすでに起動した状態でファイルを開く(Ctrl-x Ctrl-f)

一番下の行でファイル名を尋ねられるので入力します。ホームディレクトリからのパスで表示されます。

### ファイルを保存(Ctrl-x Ctrl-s)

上書き保存されます。プログラム作成中はこまめに保存するようにしてください。実行すると  
Wrote /Users/hinohara.nobuo.ga/compphys2/example.f90  
などとファイルがどこに保存されているか一番下の行に表示されます。

## Emacs を終了 (Ctrl-x Ctrl-c)

ファイルを保存していない場合は保存するかどうか聞かれます。

## 検索 (Ctrl-s または Ctrl-r)

Ctrl-s と打つと一番下のラインに I-search: と出ますので検索する文字列を入力してください。自動的に現在のカーソルの位置から最も近い検索候補に移動します。次の検索候補に移動するには Ctrl-s を続けて打つと移動できます。Enter キーや矢印キーなどを打ったり Ctrl-g と終了します。

Ctrl-r と打つと I-search backward: と出ます。現在のカーソル位置から逆向きに検索が実行されます。使い方は Ctrl-s と同様です。

## コマンド中断

### Ctrl-g

でコマンド入力を中断します。操作中コマンドに処理が移動してなんだかよくわからなくなったら Ctrl-g でファイル編集操作に戻ってください。

## 一行消去

Ctrl-k 行の中でカーソルよりも右側を消去します。消去された文字列はキルリング(クリップボードのようなもの)に保存されます。

## 元に戻す

Ctrl-x u 最後に行った動作をやめます。

## 貼り付け

Ctrl-y キルリングにあるデータを貼り付けます。

## 置換 (Esc を押して % を押す)

Query Replace : と聞かれるので置換したい文字列を入力します。

With: に置換する文字列を入力します。

現在のカーソルの位置から検索を開始し、一つずつ置換するかどうか聞かれるので y/n で処理をすすめます。



## ウィンドウ分割を解除

Ctrl-x 1 emacs を起動するとウィンドウが分割されていて下半分にメッセージが表示されていますが、ウィンドウ分割を解除して全ウィンドウを編集部分に使うことができます。

演習 以下のサンプルプログラム example.f90 を emacs を起動して入力、保存しましょう。C 言語で進めたい人は C 言語のサンプルプログラム example.c を入力します。

## Fortranのサンプルプログラム (example.f90)

!よりも右側はコメント文です。プログラムソースコードに入力する必要はありません。また、以下では日本語のコメントを書いています。自分でコメント文を作成する際には英語で入力することをおすすめします。(日本語の全角スペースなどがコメント文ではない部分に紛れ込むとコンパイルが通らなくなります)入力したあとは保存(Ctrl-x Ctrl-s)してください。

```
PROGRAM EXAMPLE          ! プログラム名の指定。
  IMPLICIT NONE          ! 暗黙型宣言を無効にします(毎回書いてください)
  REAL*8 :: a, b        ! aとbを倍精度実数型変数として定義

  a=1.0d0                ! aに1.0を代入
  b=2.0d0                ! bに2.0を代入
  WRITE(*,*) "sum of a and b =", a+b    ! aとbの和を標準出力に表示
  WRITE(*,*) "difference of a and b =", a-b ! aとbの差を標準出力に表示
  WRITE(*,*) "product of a and b =", a*b ! aとbの積を標準出力に表示
  WRITE(*,*) "quotient of a and b =", a/b ! aとbの商を標準出力に表示
END PROGRAM EXAMPLE
```

## C言語のサンプルプログラム (example.c)

```
#include<stdio.h>

int main(){

  double a, b;

  a = 1.0;
  b = 2.0;

  printf("sum of a and b = %lf\n", a+b);
  printf("difference of a and b = %lf\n", a-b);
  printf("product of a and b = %lf\n", a*b);
  printf("quotient of a and b = %lf\n", a/b);

  return 0;
```

```
}
```

## ○コンパイル

さて、プログラムが出来上がったら次はコンパイルを行います。プログラムにエラーがあるとコンパイルで失敗しますのでその場合は再びプログラムファイルを編集する必要があります。**&付きで実行した emacs は終了せずに端末に戻り**コンパイルを行ってみましょう。

作成したプログラム(ソースプログラム)をコンピュータが実行できる形式に変換する作業をコンパイルといいます。さらに、コンパイルによって生成された機械語のプログラムを実行ファイルと呼びます。コンパイルは保存されたファイルに対して行うため、コンパイルしようとするソースプログラムを必ずコンパイルの前に保存しておく必要があります。

プログラムのコンパイルと実行は端末で行います。まずは作成したプログラムがあるディレクトリに移動します。

```
$ cd compphys2/1st/
```

続いて ls でプログラムがディレクトリにあることを確認しましょう。

```
$ ls  
example.f90
```

Fortran90 プログラムのコンパイルは gfortran というコマンドで実行します。例えば、プログラムを example.f90 という名前で作成し、実行ファイルを example.out としたい場合

```
$ gfortran example.f90 -o example.out
```

となります。すぐにコンパイル結果が表示されます。コンパイルが成功すれば通常はメッセージが表示されずにコマンドプロンプトがもう一行表示されますが(ls で実行ファイル example.out ができていることを確認してください)、プログラムに誤りがある場合にはエラーメッセージが表示されます。エラーがある場合はプログラムの実行ができませんので、プログラムを修正し、保存して再度コンパイルする必要があります。-o オプションで実行ファイル名を指定しなかった場合はデフォルトの a.out という名前で実行ファイルが作成されます。

C 言語の場合は gcc というコンパイラを用いてコンパイルします。

```
$ gcc example.c -o example.out
```

## ○コンパイル結果について

コンパイルに失敗した場合はエラーが表示されます。

ファイル名:先頭からの行数:

エラーが発生した行が表示され、原因と思われる箇所の下に「^」の文字または数字があります。その下にエラーの説明があります。このような場合は、プログラムに文法上の誤りがあり、コンパイルが正常に終了していません。プログラムを見直して、保存し、再度コンパイルして下さい。(あるいはプログラム修正後、保存を忘れていませんか?) エラーの表示された行(もしくはその1つ前の文)を確認して修正して下さい。(1つの誤りで多くのエラーが表示されることがあるので、先頭のエラーから1つずつ修正して、保存、コンパイルを繰り返してみてください)

例:

```
example.f90:6:2
      b = 2.0s0
      1
```

Error: Unclassifiable statement at (1)

ここではプログラムの6行目でbに2を代入するところでb= 2.0d0とすべきところをタイプミスしているためエラーが出ています。

## ○実行

正常にコンパイルが終了したら、そのプログラムを実行させます。ターミナルの画面で、

```
$ ./example.out
```

としてください。実行ファイル名を指定するのですが、カレントディレクトリにある場合は./をつける必要があります。

実際にプログラムを組んでみるとコンパイルには成功したのに実行結果が思い通りに得られないことがあります。その場合には、プログラムに何らかの問題(計算式の書き間違い、変数名の誤りやプログラムの構造上の問題)がありますので、プログラムを見直して修正し、再コンパイルしてください。プログラムはこまめに保存してください。

実行させたプログラムが何らかの原因で終了しない場合はCtrl-Cで強制終了させることができます。

## ○リモートアクセス

筑波大学の全学計算機システムにはサテライト室以外からでも、学内ネットワークに接続した他のコンピュータから、あるいは学外からアクセスすることもできます。自分のコンピュータを持っている人はぜひ設定してみてください。Linuxシステムについては

<https://www.u.tsukuba.ac.jp/remote/>

のLinuxデスクトップに接続する・UNIXシェルを利用する、を参照してください。

リモートデスクトップは通信速度に左右されるため、グラフを表示せずにプログラムの実行のみが目的であれば 4.全学計算機システムのUNIXシェルを利用する(SSH接続)をおす

すすめします。WindowsではPuTTY (<https://www.putty.org/>) などがSSHクライアントとして利用できます。ファイルの転送にはWinSCP (<https://winscp.net/eng/docs/lang:jp>) などが使えます。Macではターミナルでssh, scp/sftpが使えます(インストールする必要があるかもしれません)。学外からの接続ではVPNあるいは公開鍵認証を用いる必要があります。

### ○自分のコンピュータでの環境設定

自分のコンピュータですべてのコンパイル・プログラムの実行・グラフ作成の作業を行いたい場合、インストールは簡単ではありませんが

[http://wwwnucl.ph.tsukuba.ac.jp/comp\\_phys](http://wwwnucl.ph.tsukuba.ac.jp/comp_phys)

などを参考にしてください。

### ○まとめと演習

- ・Linuxを起動し、計算機システムにログインする。
- ・端末の使い方、emacsの使い方、コンパイルの方法、プログラムの実行の方法
- ・計算物理学2用のディレクトリを作成、その中にFortranあるいはCのサンプルプログラムをemacsで作成し、コンパイル、実行する。
- ・全学計算機システムへのリモートアクセスの方法を試みる。
- ・Linuxシステムにも慣れてください。計算物理学1の講義でも扱っていたので一度復習しておいてください。(II. Unixを使おう)

<http://www.u.tsukuba.ac.jp/~ishiki.goro.gm/CompPhys18/index.html>