平成１８年度　計算機数学　火・１

サテライト演習室 PC 上での Linux への login

PC 本体前面，ディスプレイの電源ボタンを入れる。数分間多数のメッセージが表示されるが、その間はキーボードの操作を行わないこと。

「Press Ctrl-Alt-Del の画面」が出たら指示にしたがってキーを押す。「認証をおこないます画面」で、アカウント名，パスワードを入力し，Vine Linux を選んでログオンボタンを押す。「VMware を起動しています」「Please wait till start X!!」が順に表示されて、x-window が立ち上がり 自動的に“kterm ウインドウ”がディスプレイ上に表示される。

まれに x-window が立ち上がらない端末がある。その場合は，character 画面上の

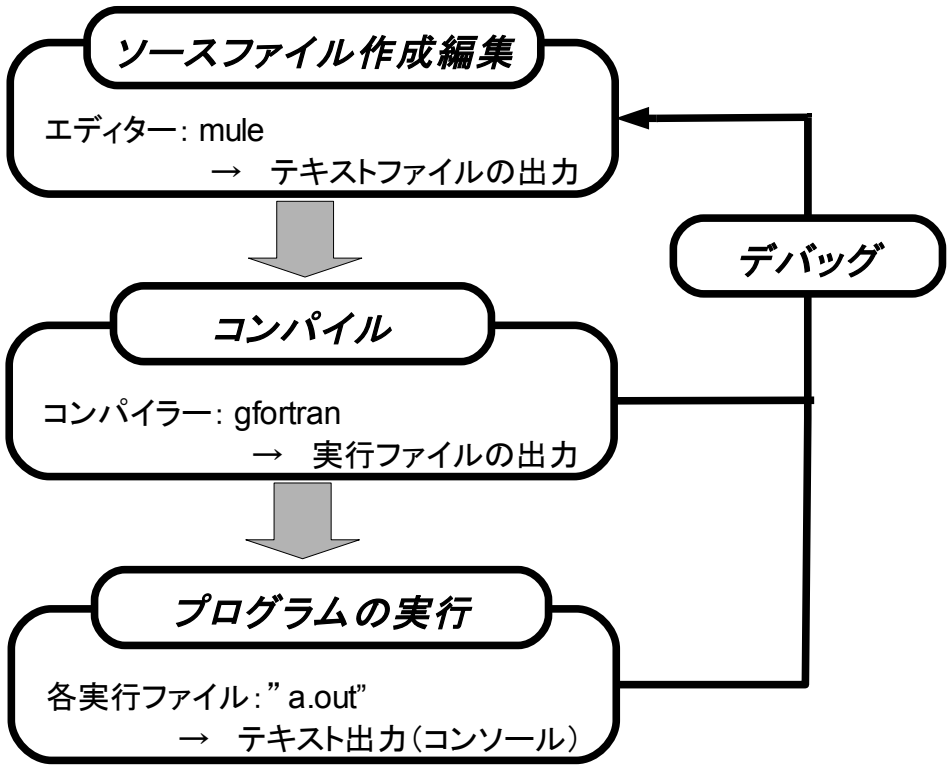
[....]$ **startx**

を入力する。

PC の logout

終了する際は、デスクトップでマウスを右クリックし、メニューから[Logout]を選択して右クリックを離す。電源を切断するには、「Press Ctrl-Alt-Del」画面が出たら指示にしたがってキーを押す。「認証をおこないます」画面で、シャットダウンボタンを押し、「シャットダウン」画面が出たら高速シャットダウンを選んで OK をクリックすると本体の電源が切れる。

作業の流れ



Mule(emacs, プログラム作成のためのエディタ)

1．Mule の起動

"kterm のウインドウ"内にマウスポインタを移動させてから、「$」に続けて

[….]$ **mule *filename.f90***

[….]$ **mule &**

と入力し、[**Enter**]キーを押す。あるいは、ルートウインドウ（kterm ウインドウの外側の、背景の部分）にマウスポインタをあわせ左クリックする。続いて、"**RootMenu**"の中の"**Mule**"にマウスカーソルをあわせ、左クリックする。すると、Mule ウインドウが現れ、Mule が起動する。"**&**" をつけて mule を実行するとバックグラウンドで作業が可能になる。ただし、編集したときにファイルをセーブしたか、確認を忘れないように。

2．新しいプログラムファイルの作成／既存のファイルを開く

「**Files**」をクリックすると、サブメニューが表示される。そこで、一番上の「**Open File**」をクリックすると、Mule ウインドウの一番下の行に"**Find File : ~/**"と表示されるので、キーボードから使用するファイル名（たとえば、"**sample.f90**"）を入力し、[**Enter**]を押す。"**.f90**"はフォートラン 90 のプログラムであることを表す拡張子である。ウインドウの下の方に黒い行がある。これを「モード行」といい、そこには現在の Mule に関する情報が表示されている。詳しいことはともかく、とりあえずモード行に（**sample.f90** という）ファイル名が表示されていることを確認すること。

3.プログラムファイルの編集

まず、最初のブランクを含め 7 文字目から（[**Tab**]キーにより自動的にカーソルが移動する）文字を入力し、フォートランプログラムを作成する。この時、マウスポインタがウインドウの中にないと(ウインドウがアクティブでないと)入力出来ないので注意する。

4．プログラムファイルの保存

「**Files**」をクリックするとサブメニューが表示されるので、そこで「**Save Buffer**」をクリックすると、作成したプログラム **sample.f90** が保存される。

5．Mule の終了

Mule を終了するときは、「**Files**」をクリックした後、「**Exit Emacs**」をクリックする（ショートカットキー Ctrl+x, Ctrl+c でも出来ます）。もし編集中のファイルがあるのに終了しようとした場合は、Question ウインドウで保存するかどうか聞いてくるので、通常は **Yes**、あるいは **Save All Buffers** を選ぶ。

Fortran の実行（コンパイルとプログラム実行）

他方の kterm のウインドウ内にマウスポインタを移動させてから、「$」に続けて

[….]$ **gfortran *sample.f90*** （←実行するプログラムファイル名）

と入力し、[Enter] キーを押す。この時、コマンド"f90"の"f "は、小文字であること。すると、mule で作成したファイル ***sample.f90*** を **gortran** がコンパイル(機械語に翻訳)し"***a.out***"という、実行形式のファイルを作る。次に「$」に続けて

[….]$ **./a.out**

と入力し、[Enter]キーを押すと、「プログラムに間違いがなければ」、プログラムが実行される。また、「$」に続けて

[….]$ **gfortran *sample.f90* -o *sample.out***

を入力して、実行形式の sample.out ファイルを作り、

[….]$ **./sample.out**

を実行しても良い。通常、カレントディレクトリにはパスが通っていない（ターミナルソフトが指定されたファイルを探しに行くべき場所ではないと認識している）ので、ファイルを実行する場合には、実行ファイルがカレントディレクトリにあることを "./file_name.out" のように明示する。ドット"."はカレントディレクトを意味する。参考までに親ディレクトリは"..".

また、これまで作成したファイルをチェックしたい場合は

[….]$ **ls**

を実行し、save されているファイルをリストする。ファイル名を sample.f90 から例えば reidai.f90 に変更する場合は、unix コマンド mv を使って、

[….]$ **mv *sample.f reidai.f***

なお、その他の unix コマンドは別紙を参照すること。

UNIX(linux)コマンド・アプリケーションの説明

CUI(Character User Interface)、コマンドラインステートメントの基本

端末マシンのターミナルソフトが起動するとコマンドラインにプロンプトが表示される。

[*user_name directory_name*]$

デフォルトでは *user_name* はアカウント ID@マシン名になり、*directory_name* は自分のホームディレクトリーを意味するチルダ"~"になっている（起動直後の場合）が、シェル定義ファイルを書き換えれば自分の好みに合わせて変更できる。プロンプトの表示はターミナルソフトがユーザーからの入力を待っている状態であることを示している。ユーザーが所定書式でコマンドを記述し、リターンを入力するとターミナルソフトはコマンドの実行を試みる。実行が終われば再びコマンドラインにプロンプトを表示するはずで、その表示がない間はコマンドを実行中ということを意味している。

UNIX コマンドの基本文法

UNIX ではいろいろな操作をする場合にコマンドを使う必要があり、このコマンドは基本的に英語の文法に準じた語順で指定される。すなわち、

コマンド　（目的語）　（目的語）

となる。例えば、

＜コマンド＞の例

カレントディレクトリを表示する：**pwd**

ファイルの一覧を表示する：**ls**

＜コマンド　目的語＞の例

sample.f90 というファイルを削除する場合：**rm** sample.f90

sample というディレクトリを作成する場合：**mkdir** sample

sample というディレクトリを削除する場合：**rmdir** sample

sample というディレクトリに移動する：**cd** sample

sample ディレクトリの中を表示する：**ls** sample

＜コマンド　目的語　目的語＞の例

sample.f90 を sample2.f90 にコピーする場合：**cp** sample.f90 sample2.f90

sample.f90 を sample2.f90 に名前を変更する場合：**mv** sample.f90 sample2.f90

sample.f90 と sample2.f90 を削除する場合：**rm** sample.f90 sample2.f90

のように用いる。また、目的語を省略した場合は、指定しない場合に解釈する目的語が決まっている場合が多い。例えば **ls** では目的語を指定しない場合には今作業をしているディレクトリを目的語として解釈する。くわしい日本語のマニュアルが **jman**（目的語）で表示される。ハイフンでオプションを設定する。

ファイルを複製する、削除する、名前を変える

コンピュータを使って作業を行うにしたがって、ファイルを複製したり、不要なファイルを削

除したりする必要が生じる。また、ファイルの名前をほかの名前にしたい場合もある。このようなときにファイルを操作する以下のようなコマンドを用いる。

ファイルの複製：**cp**
ファイルの削除：**rm**
ファイルの名称変更：**mv**

ディレクトリを作る、削除する

ファイルがたくさんできてくるとこれを整理するためのディレクトリが必要になったり、不要なファイルを削除した後の要らなくなったディレクトリを削除しなければならなくなったりする。このようなとき、UNIX では以下のようなコマンドを用いる。

ディレクトリの作成：**mkdir**
ディレクトリの削除：**rmdir**

ファイルを移動する

ファイルを整理していてあるディレクトリにあるファイルを別のディレクトリの中に移動させたいようなときにも mv コマンドを使用する。

ファイルの移動：**mv**

ファイルの一覧･ファイル情報を見る

たくさんのファイルを作っていくと、以前自分がどういう名前のファイルを作っていたのかわからなくなることがある。こういう場合に便利なファイルの一覧を見るコマンドがある。

ファイルの一覧：**ls**
オプションとして ls の後に-F, -al などを付けるとよりくわしい情報がでる。

例　[....]$ **ls -alF**

ピリオドで始まるファイルはプログラムの設定を記述しているので理解できないファイルや記述は原則として編集しない。拡張子が f90 となっているのがフォートラン 90 のプログラムファイル。拡張子にチルダが付いているのは mule が自動的に作成したバックアップファイル。拡張子が out となっているのが、コンパイルして作成された実行ファイル。実行可能なファイルにはアステリスクが付記される。

行の最初の１０文字はファイルまたはディレクトリの属性とアクセス許可情報。本人、グループ、その他の人に対してそれぞれ **r**ead（読込み）、 **w**rite（書込み）、e**x**ecute（実行）が許可されていることを示す。許可されてない場合はハイフン： "-"。

Windows では元来ファイルの属性管理が詳細でないので、Windows との間でファイルをやり取りすと属性が変わってしまう。属性変更には chmod コマンドを用いる。

ディレクトリを移動する

ある決まったディレクトリでファイルを操作し、これをほかのディレクトリに分類するのであれば特にこの操作は必要ではないが、最初から仕事に使うディレクトリを決めてその中でいろいろな操作を行う場合がある。このようなときにはディレクトリの移動という作業を行う。

ディレクトリの移動：**cd**

一つディレクトリの階層を上がるには，cd .. (cd スペース　ドット　ドット)とする

カレントディレクトリがドット:”.” で、親ディレクトリが２ドット:”..” で、ユーザーのホームディレクトリがチルダー:”~”で表わされる。

現在ひらいているディレクトリを知る

ファイル、ディレクトリを作成しながらディレクトリの移動を繰り返していると、現在作業しているディレクトリ（カレントディレクトリ）がわからなくなる場合がある。こういう場合にはカレントディレクトリを表示するコマンドを使う。

カレントディレクトリの表示：**pwd**

ファイルの中味を特定のコマンドで見る

ファイルをたくさん作ってしまうと、ファイルの名称だけでは何のファイルだったかわからないことがある。こういう場合に、エディタで開いてみないと中味の確認ができないということになると非常に不便。UNIX ではファイルの中味を console 表示するコマンドとして cat が用意されている。また、ページャーと呼ばれるフィルターを介することでファイルの中味を一部づつ順を追って表示させることもできる。

全部を一度に表示：**cat**

１ページごとに表示：**less**

例　[....]$ **ls -alF | less**

[....]$ **cat *file_name***

console(画面)への出力を less をパイプして出力します。less の中ではリターンで続行、ｑで終了(quit)。

ワイルドカードで複数のファイルを一度に指定する

UNIX では複数のファイルを操作したい場合にワイルドカードを用いることができる。ワイルドカードは複数の文字を示すことのできる特別な記号で、これには大きく２つあって任意の１文字を示すものと、任意の文字列（１つ以上の文字のならび）を示すものがある。

任意の１文字：**?**

任意の文字列：*****

ワイルドカードを有効に使用することによって、多くのファイルを一度に削除したり、移動させたりすることができる。例えば file1, file2, file3, ..., file9 を全て消去したい場合は

rm file **?**

というふうに用いる。また現在のディレクトリの中にあるファイルを全て消去したい場合には

rm *****

というような使い方をする。ただし、いったん消してしまったファイルは元に戻らないので、注意すること。

テキストファイルを印刷する

ファイルをラインプリンタ（演習室で共同利用のため各部屋に設置されているプリンタ）で出力することが出来る。現在はポストスクリプトファイルの出力が標準に設定されている。くれぐれもバイナリーファイル（*.out の実行ファイルなど）の印刷命令を出さないように留意すること。直接アスキーファイルの内容を表示したいときは a2ps コマンドが準備されている。

印刷時に区別すべきファイルの種類

アスキーファイル（テキストファイル）

バイナリーファイル：実行ファイル、プログラムなど

ポストスクリプトファイル：印刷コマンドを特定の書式でテキスト記載したファイル

ファイルを印刷：**lpr**

例　[….]$ **lpr *file_name.ps***

例　[….]$ **a2ps *file_name.txt* | lpr**

ポストスクリプトファイルのコンソール表示

ghostscript を利用することができる。

例　[….]$ **gs *file_name.ps***

リダイレクションとパイプ

コマンドの出力はデフォルトではコンソールであるが、別途指定することにより、テキストファイルとして出力したり、別のコマンドに渡したりすることが出来る。

例　[….]$ **ls –alF | grep *f90***

ディレクトリーにあるファイルのリストを作成しその結果を grep で行検索して"f90"が含まれるものだけを表示する。

例　[….]$ **./sample.out>results.txt**

実行ファイル sample.out の実行結果をテキストファイル results.txt に書き込む。">"を">>"に変

えるとファイル末尾に追加書きとなる。

コマンドラインマニュアルを参照する

UNIX でコマンドラインから実行できるコマンドのほとんどはオンラインマニュアルで書式などを確認できる。

マニュアルの表示：**man *command_name***

日本語のマニュアルは jman で参照できるが、日本語マニュアルが準備されてないコマンドも多い。

グラフのプロット

gnuplot を利用することが出来る。通常のプロンプトで gnuplot を入力すると gnuplot のプロンプトが表示される。詳しくは別途マニュアルを参考に。

例　ある 2 次元データ（results.txt に保存されている）を x-y プロットをする方法

[….]$ **gnuplot**	gnuplot の起動
gnuplot>plot results.txt	コンソールに出力される
gnuplot>set terminal postscript	出力をポストスクリプトに変更
gnuplot>set output “results.ps”	出力先のファイル名を指定
gnuplot>replot	再出力される
gnuplot>quit	gnuplot の終了