

Getting started with Unix and computing

If you do not have UVa computing ID and Home Directory accounts, please apply at

<http://www.itc.virginia.edu/helpdesk/accounts/createacct.html>

You can connect to blue.unix

by using SecureCRT and transfer files using SecureFX – software that you can get from <http://www.itc.virginia.edu/central/>

For example, after you login to your UVa Home Directory using SecureCRT, you should be able to use Unix commands to manipulate files, create sub-directories, write computer codes, compile and run them.

For your term projects some of you may need more powerful computational resources – in this case you can request an additional account to use Research Computing Systems. Please read <http://www.itc.virginia.edu/research/hpc-account/> for instructions.

Some simple unix commands/help

File management

1) editing

Use vi, pico, jove, emacs, nedit, nano, xemacs,...

\$ pico filename

2) directories/files

ls : list directories and files

ls -la : list directories and files, how big they are, creation date, etc.

mkdir dirname: make directory named dirname

cd dirname : change to directory dirname

cd .. : go up one directory

cd : cd to your home dir

mv file1 file2 : move file1 to file2 (or rename file file1 to file2)

cp file1 file2 : copy file1 to file2

rm file1 : delete file1

rm file1 file2 ... : delete file1 file2 ...

rmdir dirname : delete directory dirname which is empty

mkdir dirname : make directory dirname

ftp machinename: to transfer files to remote machines

ssh machinename: connects to remote machines using secure shell (better than telnet)

Compilers

c/c++ compilers: cc, gcc, c++, CC, gCC, gc++,...

Fortran compilers: f90, gf90, xlf, ...

Generally one adds "flags" to compile a code:

\$ f90 -o execname file.f

-o execname: name the output executable "execname"

To run the executable:

\$ execname &

To be able to type other commands while the code is running (run in the background)

\$ execname &

To be able to log out and leave the code running:

\$ nohup execname &

To show your running processes

\$ps -fu loginid

That will show the process ID (PID). Then you can kill the process by typing

\$ kill -9 PID

Miscellanea:

"*" are wild cards. For instance "ls *.C" will list only files ending in .C.

"man command" gets the online help for command

"man -k topic" show me commands dealing with topic.

"Ctrl c" generally kills the current program being executed.

"more file" shows all file one screen at a time

"cat filename" shows the whole file

"head file" shows the top part of the file

"tail file" shows the last few lines of a file

"grep hello *.f90 " returns all lines containing the word hello in any file named *.f90

"command >> out.dat" sends output of command to out.dat

ITC (Computing Help Desk: 924-3731)

Web: <http://www.itc.virginia.edu/help.html>

e-mail: pc-consult@virginia.edu, unix-consult@virginia.edu, res-consult@virginia.edu

ITC UNIX pages: <http://www.itc.virginia.edu/desktop/unix/>

INTRO to unix: <http://www.itc.virginia.edu/desktop/unix/docs/u002.unix.intro.html>

Example Fortran code:

```
PROGRAM Test
! Lines that start with exclamation mark are ignored by Fortran90 compiler

IMPLICIT NONE
! By making this statement you are announcing that you will define types of all variables
! used in the code. An alternative is to use default type declarations.

REAL, PARAMETER :: PI = 3.141592653589
! The above line defines a named constant of type REAL.
! Named constants cannot be changed later in the code

REAL :: X, Y, dx, Xfinish
! This line declares several variables of type REAL

INTEGER step
! This line declares an integer variable step

OPEN (UNIT = 15, FILE='test.in')
! Open input file test.in

Read (15,*) Xfinish, dx
! Read the data from the input file

OPEN (UNIT = 16,FILE='test.out')
! Open output file

step=1
X=0

! We are starting a DO loop below
X_loop: DO

! Below we are performing some calculations
X=X+dx
step=step+1
Y=sin(X*PI)

! Writing data to the disc (to file test.out)
WRITE(16,*) step, X, Y

IF (X.GT.Xfinish) EXIT X_loop

End do X_loop

CLOSE (UNIT = 15)
CLOSE (UNIT = 16)
STOP
END
```

Instructions to compile and run the code

1. Create an input file: In the same directory where you are planning to run the code open new file test.in using your favorite editor (vi, nedit, emacs, pico, etc.). Enter values of the input variables and save the file.
2. Write a code using the same editor. Save it to a file with extension .f, e.g. test.f
3. Compile the code. To get an executable named test.x from file test.f:
f90 -o test.x test.f
4. To run the executable just type test.x or ./test.x
5. As a result of the execution, you should get new file test.out with the output data. You can plot the output data using your favorite graphics package, e.g. GNUPLOT. You can find instructions for GNUPLOT at <http://www.people.virginia.edu/~lz2n/mse524/Eduardo/>

You can also transfer this file to your PC using SecureFX and use your favorite graphics package in MS Windows or Mac environment.

If you have any problems, please do not hesitate to ask me.