04-02-2015

Handout Computer Lab 1: Introduction to UNIX, FORTRAN, and FERRET

Log on to linux server:
On the Mac, launch X11 or Terminal (in Applications/Utilities/). (Windows users see note below.)
(If logging on from outside CEOAS type: ssh -X username@coas.oregonstate.edu. Hit return if prompted for window.)
Now type: ssh -X username@studentY (username is the student's CEOAS login name, Y is a number from 1 to 11). If you are prompted to accept an RSA key fingerprint, type 'yes'. Enter your CEOAS password. You are now remotely logged into one of the CEOAS student lab Linux systems.

Remote login from windows: tips from a student:
It also might be worth noting that if people are trying to use Cygwin or other similar unix-emulatorq from their windows machine to ssh, they have to install the OpenSSH package. It wasn't until after I did this that I was able to ssh to coas from my windows machine. Here is a tutorial that explains how to do it. Also, in the first step of the tutorial it says to open the Cygwin setup.exe file. It can be downloaded here.

Important UNIX commands:

bash	use bash shell (my recommendation because it has auto-completion with tab)
cd	change directory
ls	list
pwd	present working directory
mkdir	make new directory
rmdir	remove directory
rm	remove file (be careful, there's no undo once a file is removed)
cp	copy file (e.g. cp file1.f file2.f)
mv	move file
lpr	print (e.g. 'lpr -PmyBWprinter plot.ps' prints postscript file plot.ps on printer myBWprinter)
lpstat	print status (lpstat -a shows available printers)
man	manual pages (e.g. 'man man' or 'man ls')
more	look at file content
vi	vi-editor
exit	exit shell terminal
*	wildcard (e.g. 'rm *' will remove all files in a directory; be careful !)
grep	finds text in files (e.g. 'grep temp *.f' will look in all files that end with .f for the text 'temp')
|	pipe: diverts output from one command to input for next command (e.g. grep temp *.f | more)
arrow up key	previous command
tab key	auto-complete
ps	list processes
kill -9 PID	kill process # PID
CTRL z	halt process
bg	put process in background
fg	put process in foreground
CTRL c	abort process
emacs &	start emacs editor in background (&) (on Mac use xemacs &)
ssh	remote login (e.g. 'ssh -X myusername@student1.coas.oregonstate.edu')
scp	remote file copy (e.g. scp myusername@student1.coas.oregonstate.edu:~/test/test.f .)

Editors

editor: EMACS (C^ means CTRL)

C^x C^f	open file
C^x C^s	save file
C^x C^c	exit
C^s	search
C^r	search backward
Esc %	replace
C^k	delete line
C^d	delete character
C^y	paste (yank)
C^g	abort command
C^e	goto end of line
C^a	goto beginning of line
C^h	help
C^SHIFT_	undo
C^SPACE	set mark
C^w	delete region (from mark)

editor: vi (or vim)

ZZ	save and exit
hjkl	move cursor
$	goto end of line
dd	delete line
u	undo
i	insert mode
a	append (also enters insert mode)
ESC	exit insert mode
:w	write (save)
:q	quit
:help	help

editor: gedit

gedit is another user friendly editor

FORTRAN

Download and Installation

Mac: Download Xcode from https://developer.apple.com/xcode/ and install by double clicking on it (or insert OS X Install CD; click on Xcode Tools; double click on XcodeTools.mpkg). I also installed the command line tools. This should include gfortran. (Here is another free fortran compiler for high performance computing http://hpc.sourceforge.net/.)

Use: Create source code in text editor (e.g. emacs or vi)
first 6 columns are empty
test program test.f
$ emacs test.f&

```
      program EBM
      temp=0
      dt=1
      do i=1,10
           temp = temp + dt
           print*, temp
      enddo
      end
```

compile with
gfortran test.f (or f95 test.f)

run with
a.out (or ./a.out)

diverting output into file
a.out > file.dat

introduce in fortran
-functions and subroutines
-if statement

```
      program EBM
      do i=1,10
           temp = temp + dt
           print*, FLW(temp)
           call FSW(temp,x)
           print*, x
      enddo
      end

      function FLW(arg)
      A = -726
      B = 3.36
      FLW = A + B*arg
      return
```

```
      end

      subroutine calcFSW(in,out)
      S = 342.
      albedo = 0.3
c     this is a comment
      out = (1.-albedo)*S
      return
      end

c     mathematical operations
c     multiplication
            out=5*2.
c     division
            out=5./2.
c     exponentiation
            out=5**2.
```

An introduction to fortran is available here:
http://www.dur.ac.uk/its/software/application/?application=Nag+Fortran+90+Library
http://gcc.gnu.org/wiki/GFortran
http://gcc.gnu.org/fortran/

FERRET

Ferret is a free software for analysis and plotting of gridded data. Google “ferret noaa” for more info, documentation and download. As we will see it is particularly useful for analyzing large datasets such as climate model output.

copy my .bashrc and ferret_path files from /home/server/scratch/ATS421-521 to your home directory
if your running a bash shell type “source .bashrc”
now type “ferret”
this should start ferret and you should get the ferret “yes?” prompt

- Plot a function

define an axis:
yes? define axis/x=230:330:1 tempax

this could represent temperature in [K]. The last number is the interval, in this case 1 K. If you want a 0.2 K interval you'd use x=230:330:0.2

now you can define a variable that is the temperature like this:

yes? let temp = x[gx=tempax]

you can plot it

yes? pl temp

now you can create a function that depends on this variable:

yes? let FLW = 5.6e-8*temp^4

yes? pl FLW

- load ascii file:

yes? file/var=temp out.dat
show data:
yes? show data
or
yes? sh da (commands can be abbreviated)
you'll see the variable names and the number of grid points for each dimension

you can also read multiple columns
yes? file/var=time,temp out.dat

plot data
yes? plot temp
or

yes? plot/vs time, temp
you can create a time axis and a grid using this axis
yes? define axis/t/from_data/name=tax time
yes? define grid/t=tax mg
now re-read the data
yes? file/var=time,temp/g=mg out.dat
yes? plot temp

- you can also read multiple files
yes? file/var=time,temp/g=mg out2.dat
yes? sh da

and plot both
yes? pl temp[d=1], temp[d=2]

- load netcdf data:
yes? use levitus_climatology

plot data:
yes? shade temp[k=1]
yes? sha salt[x=180w]

yes? plot salt[x=18w,y=25n]
yes? pl/over salt[x=18w,y=55s]

averaging:
yes? pl temp[k=1,x=@ave]

differentiating:
yes? pl temp[k=1,x=@ave,y=@ddc] (centered difference)

integrating:
yes? pl temp[k=1,x=@ave,y=70s:70n@iin]
yes? list temp[x=@din,y=@din,z=@din]

show all transformations:
yes? show trans

show all functions:
yes? sh func

define new variables
yes? let dtdz = temp[z=@ddc]

save your new variable in a file
yes? save/file=my_dtdz.nc dtdz

- write your own ferret script:
open an empty file in an editor
enter ferret commands in the file and save it with a *.jnl extension e.g. myscript.jnl
to run it type
yes? go myscript

voila !

- Producing a figure:
yes? set mode metafile
yes? pl/vs/li {1,2},{2,3}
yes? q
now you have a file called metafile.plt. You can convert this to a postscript file using
$ Fprint -o myplot.ps metafile.plt
typing Fprint without arguments gives you help (try it!)
I usually use the -p portrait option and sometimes for color lines the -l cps option

once you have the ps file you can view it with “ghostscript” or convert it to a pdf file with “ps2pdf”