

Lec 4. Remote Login, Send/Receive Files, Gnuplot

Remote Login and DISPLAY with X Windows: *One of the greatest innovations in computing in the late 80's was the invention of X Windows and client-server technology. Not only we have multiple graphics terminals on a screen to work in, but we can connect to another computer and have it display its GRAPHICAL output on the screen in front of us!*

- ▶ To login to your webcat/wxsession account
\$ ssh webcat.db.erau.edu
\$ ssh -XY wxsession.db.erau.edu
At the login prompt enter your username and password.
- ▶ To connect to your wxsession account from other's account
\$ ssh -XY username@wxsession.db.erau.edu
- ▶ To copy a file from this machine to your wxsession
\$ scp file username@wxsession.db.erau.edu:
- ▶ To copy a file from wxsession to this machine
\$ scp username@wxsession.db.erau.edu:file .

Type man ssh, man scp for more info!

Wildcards in Unix names

- ▶ * matches against none or more character(s) in a file (or directory) name.
\$ ls *py (lists all file names ending with 'py')
\$ ls lab* (lists all file names beginning with 'lab')
- ▶ ? matches exactly one character in a file (or directory) name.
Try: ls lab?.py, ls lab0.?, ls ?ab0.py

Redirecting the Output

- ▶ the symbol > redirects the output of a command to a file.
\$ cat lab0.py > test.py
\$ python3 test.py > aaa.txt
\$ cat lab0.c > test.c
\$ gcc test.c -lm
\$./a.out > bbb.txt
- ▶ the symbol >> appends standard output to a file.
\$ cat test.f90 >> out

Create 'typescript' file from screen output

Unix tool script keeps a record of everything you do (in the window where it's running) into a "log" file named "typescript", which you can edit and submit as your lab or classwork.

- ▶ In a big xterm, type:
\$ script (you'll see: "Script started, file is typescript")
\$ cat myscript.py (prints the file on the screen)
\$ python3 myscript.py
\$ chmod u+x muscript.py
\$./myscript.py
.....
\$ exit (you'll see: "Script done, file is typescript")
- ▶ Everything is saved in the file "typescript". In fact, more than you think is saved, like key codes from backspace, delete, enter, etc! Most likely it's a mess!
- ▶ Edit the file (\$ vi typescript) to clean up all the junk (great practice for vi !), make it readable and rename it as cw#script.txt or lab#script.txt!

Brief Introduction to Gnuplot

- ▶ **Gnuplot** is a portable command-line driven graphing utility, freely available for Linux and many other platforms. It can be used to draw many types of plots in 2 and 3 dimensions using lines, points, boxes, contours, vector fields, surfaces, etc.

To start **gnuplot** type “gnuplot” in the command line:

```
$ gnuplot
```

To exit type q in the **gnuplot** prompt:

```
gnuplot> q
```

- ▶ **2D-plot:**

```
gnuplot> plot cos(x)
```

```
gnuplot> plot [-10:10] cos(x)
```

```
gnuplot> plot [-10:10] cos(x), sin(x)
```

- ▶ **3D-plot:**

```
gnuplot> set contour
```

```
gnuplot> splot [-10:10][-10:10] sin(x*x+y*y)/(x*x+y*y)
```

► **Plotting data from a file named 'data.txt':**

```
# x      y
1    1.4
2    3.2
3    8.2
4   14.2
```

```
$ gnuplot
```

```
gnuplot> plot "data.txt" with lines
```

```
gnuplot> plot "dat.txt" with points
```

```
gnuplot> plot "dat.txt" with linespoints
```

► **Some useful options:**

```
gnuplot> unset key
```

```
gnuplot> set title "Plot of ..."
```

```
gnuplot> set xlabel "t(sec) ..."
```

```
gnuplot> set ylabel "v(m/sec) ..."
```

► **Printing and exporting the output**

First you prepare your plot as you want it in gnuplot. Then

- `gnuplot> set term postscript`
(will produce postscript output)
- `gnuplot> set output "printme.ps"`
(output to any filename you use)
- `gnuplot> replot`
(recreates plot but you don't see it, goes to file)
- `gnuplot> set term x11`
(resets the normal terminal so you can do more plotting)

Now you have a postscript file named `printme.ps` containing your plot. You can view, print or convert to other formats.

- `$ ghostview printme.ps`
- `$ lpr -Pprintername printme.ps`
- `$ ps2pdf printme.ps printme.pdf`
- `$ convert -quality 1000 printme.ps printme.jpg`

- ▶ Now, you should be able to complete your lab assignment.

Lab 1. Warmup On Unix, Python and Gnuplot

Due: Thursday, 09/13/2018

- ▶ Next class: **Introduction to Python Programming**