

UNIX Lab Tutorial

CIS 15

Functionalialia

- Did everybody get my e-mail?
- Class cancelled on Feb 15 - Monday evening LAB TIMES (before projects are due)
- What did you think of InTheBeginning...

Summary

- UNIX Tutorial
- Download and compile a program!

UNIX Let's go

- Can you see the screen? Move closer.
- Can you log in?
- If not, pair up with someone who can... take extensive notes!



Using the Command Line is like
akin to tinkering under the
hood of a car.

It's good for you as Computer
Scientists, and approaching it
with a hack-tinker-attitude
will make it all that more
enjoyable!
Refreshing change to a GUI.

UNIX Shell

- The **Shell** is the Command Line Interface Environment that you are in when you log into the UNIX machine. It's your vehicle for driving around the UNIX machine.
- Shell's command prompt is denoted by a '\$' or '%'
- Just like there are many variants of UNIX, there are many variants of Shells. (These machines use the Korn Shell aka ksh)
- Just like cars, you can customize your shell by writing some config files (not necessary for this course)

UNIX: A collection of Tools

- UNIX is digital tool shop. There are a myriad of tools (usually really short words)

- man
- ls
- cd
- rm

- Usually the commands take this form:

`command - option(s) argument(s)`

- With the Shell prompt included, this is an example

`% ls -l project1`

- In documentation a greater-than (<) and a less-than (>) are used to describe the argument:

`% ls -la <directory name>`

Getting UNIX Help

- Buffet of UNIX tools for the SUN Lab:
- <http://acc4.its.brooklyn.cuny.edu/~help/>
- Use the **man <unix command>**
- Get “Just Enough Unix” - read up on the tools covered today
- Come to office hours (ask for help)

Home Directory

- Everything is File (Directories are special files)
- When you log-in, you are “in your home directory”
- This is your workshop, keep it tidy!
- Use **pwd** (Print Working Directory), to see where you are (shows a *path* more on that later)
- `% pwd`

ls

- List the contents of the current directory of where you are in (these are your “eyes”)
- Shows current directory
- `% ls`
- Shows ‘all’ - everything in current directory
- `% ls -a`
- Shows more information about things in the current directory
- `% ls -l`
- Shows the content of a particular directory
- `% ls <directory>`

cd

- Change Directory - Navigate about the UNIX system (these are your “legs”)
- Goes into a directory
- `% cd <directory>`
- Goes “up” a directory
- `% cd ..`
- No change (why?)
- `% cd .`
- Goes Home
- `% cd`

Text Editors

- Many options:
 - vi
 - emacs
 - pico
- Create a file with pico:

```
% pico <filename>
```
- Write something fun!
- ^x Exits ... (what is the '^' for?)

TAB Completion

- Type in `cd` and then hit the TAB key twice

`% cd [TAB]`

- Start typing out a directory name, half way through, hit the TAB key

more

- Show the contents of a file.

```
% more <filename>
```

- Space-bar moves a page
- 'q' quits
- **less** is **more**

mkdir

- Makes a directory

```
% mkdir <directory name>
```

mv

- Moves files around the system
- Moves a file into a directory

```
% mv <filename> <directory>
```

- Also used to RENAME a file

```
% mv <filename> <new filename>
```

cp

- Copies a file around (like mv)

```
% cp <filename> <copy-of-file name>
```

Wrecking Balls

- These are the basics. Now we are going to talk about how to delete files.
- In UNIX there is NO GOING BACK!
- Be CAREFUL! Once you delete a file or a directory it's GONE - no Trashcan, no Recycle Bin, no Dust Pan...
- Use **cp** alot!
- Be prepared to throw one away accidentally!

rm

- Removes a file

```
% rm <filename>
```

rmmdir

- What does this do?

history

- Shows a history of the last things you've typed to date

```
% history
```

- Shows the last 100 things you've typed

```
% history -N 100
```

- Too fast for you? Pipe it through more:

```
% history -N 100 | more
```

- This is where UNIX is like a glue-stick.

E-Mail me your history

- Two steps (using re-directs):

1. First create a file with your history

```
% history > bubba
```

2. Then, send me your file

```
% mail chipp@sci.brooklyn.cuny.edu < bubba
```

- `bubba` in this case is a text file containing your history. Go ahead and take a peek using **more** (i.e. `more bubba`)

Compile a Program

- Download some code
- Edit it
- Compile it
- Run it
- Debug it

Download some code

- Use **lynx** the text based web-browser!
- In your Home Directory
- `% lynx http://www.sci.brooklyn.cuny.edu/~chipp/cis15/`
- Press the DOWN arrow to high-light the “Warm-up’ link
- Press ‘d’ to download.
- Choose “Save to disk”, hit enter to accept the filename.
- Press ‘q’ to quit.

View/Edit the Code

- Use **pico** to open the file

```
% pico warmup.cpp
```

- Just look at it now, you can edit it later

Compile the Code

- Use the GNU C++ compiler

```
% g++ warmup.cpp -o warmup.out
```

- g++ works like this:

```
% g++ <source file> -o <output filename>
```

Run the Code

- Need to use a `./` in front of the code (this has to do with your *PATH* environment variable)

```
% ./warmup.out
```

- OOOH! Output!

Debug Your Code with dbx

- First you need to re-compile (that is compile again) your code with **debug flag** set

```
% g++ -g warmup.cpp -o warmup.out
```

- Your old warmup.out is **overwritten** by the new one (no questions asked!)

Debug Your Code with dbx

- Start **dbx** on your program

```
% dbx warmup.out
```

- Now you are in the **dbx** program (observe the **dbx**:

```
(dbx)
```

- Run program

```
(dbx) run
```

- Observe all of the output at once!

Debug Your Code with dbx

- View your code (choosing lines 1 - 34):

```
(dbx) list 1, 34
```

- See the line numbers?

```
(dbx) stop at 15
```

- Now run your program

```
(dbx) run
```

- You're paused at the beginning of the code, now incrementally step through the code

```
(dbx) step
```

- Observe the lines and the output that is generated.

Debug Your Code with dbx

- Now, set a break point at line 8

```
(dbx) stop at 8
```

- Continue to run the program

```
(dbx) cont
```

- When at the break point at line 8, observe the value of variable k

```
(dbx) print k
```

- Step a few more times

```
(dbx) step
```

- Print k, and observe a change in the printed value of k.

Remote Access from Home

- First, I strongly recommending scheduling time for yourself to come into the LAB and work (supportive environment with your classmates, and use of the SUN GUI)
- Need a listing of lab machines:
- PC: Use 'puTTY'
- Mac OS X: Use 'Terminal'

PUTTY for PC's

- Download putty.exe at:
- <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- Put putty.exe on your desktop, run it by double-clicking
- Use 'ssh'
- Server: one of the lab-machines
- Login / Password: same as what you used today

Terminal on MAC OS X

- The program Terminal is available in the Utilities Folder in your Application Folder.
- Terminal is UNIX SHELL in itself!
- Run Terminal - use the ssh command:

```
Chipp:~ chipp$ ssh acc86.its.brooklyn.cuny.edu -l chipp
The authenticity of host 'acc86.its.brooklyn.cuny.edu (146.245.252.86)' can't be
established.
DSA key fingerprint is e9:5f:72:d8:72:a0:27:ba:67:71:26:64:63:04:4a:73.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'acc86.its.brooklyn.cuny.edu,146.245.252.86' (DSA) to
the list of known hosts.
Access to Brooklyn College computers is restricted to authorized users
and approved educational and research purposes, only.
chipp@acc86.its.brooklyn.cuny.edu's password:
```

Summary

- In UNIX, you can:
 - login
 - move around
 - manipulate files
 - compile/run/debug code
- **ASSIGNMENT:** Log in remotely, hack around a bit (do a little bit of everything you did today), and send me your history via e-mail. *Don't put this off till the last minute!*