Course Information

- **When**: September 8th - October 4rd; 3 times a week
- **Where**: MWF 12:20 - 1:10 in Phillips 203
- **Drop Deadline**: September 15th, one week into the course
- Passing grade for completing all assignments successfully

Hussam Abu-Libdeh  based on slides by David Slater
Course Information

- Office Hours: ??
- Email: hussam at cs.cornell.edu
- Website: http://www.cs.cornell.edu/courses/cs2042/
  - Still waiting on server permission
- CMS: http://cms.csuglab.cornell.edu
- If you preregistered you should be on CMS
Course Goals

- Overall Goal: Gain an understanding of the Unix environment and simple shell scripts
- More specifically
  - Files
  - Security
  - Text Processing
  - Regular Expressions
  - The Shell Environment
  - A variety of Shell commands
  - Basic Shell Scripting
Prerequisites

- Not assuming any previous experience with the UNIX environment
- Basic understanding of programming helpful
  - (but probably not even necessary)

Hussam Abu-Libdeh based on slides by David Slater
What Is Unix?

- One of the first widely-used operating systems
- Basis for many modern OSes
- Helped set the standard for multi-tasking, multi-user systems
- Strictly a teaching tool (in its original form)
'60s The ambitious project MULTICS (Multiplexed Information and Computing System) fails, but a number of seminal ideas (like pipes and shells) are proposed.

'69 Ken Thompson, Dennis Ritchie (et al.) start working on a file system, and name their system UNICS, which is later changed to UNIX.

- UNIX was “small, simple and clean”, and distributed freely to many universities, where it becomes popular.
Short history of UNIX

‘73 Thompson and Ritchie rewrote UNIX in C (while most of the operating systems at that time were written in assembly)

‘81 Berkley UNIX 4.1 BSD: vi, C shell, virtual memory

‘91 Linux, GNU, and others: similar to UNIX, but their source code rewritten, very popular and widespread, free
  ● Currently, X/Open is responsible for developing UNIX
  ● Many Linux Distributions: Ubuntu, Fedora, Debian, ...
Current UNIX Flavors

- Berkeley Software Distribution (BSD)
- Sun Solaris
- GNU/Linux
- Apple OSX
- Developed by students and faculty at UC Berkeley
- Forked from the proprietary version back in the 80s
- Has since split into many additional flavors - namely, NetBSD, OpenBSD, and FreeBSD
- Spawned a popular open-source software license (the BSD License!)
- Primary competitor to Linux among free OSes
Solaris

- Commercial offshoot of BSD
- Designed to run on Suns SPARC servers, since ported to x86
- Most of the source code was recently released for the OpenSolaris project
Pieced together by a Finnish guy named Linus Torvalds starting in 1991
Built over the internet using message boards (Usenet)
Designed to a UNIX-like standard, but not a direct descendant

Note:
Linux technically only refers to the OSs core, or kernel - without other programs it cant really do anything.
GNU = Gnu is Not Unix

- Movement in the 80s to build a free OS
- Created many very popular tools
- Unix like but uses no Unix code

Stallman says:

There really is a Linux, and these people are using it, but it is just a part of the system they use. Linux is the kernel: the program in the system that allocates the machines resources to the other programs that you run. Linux is normally used in combination with the GNU operating system: the whole system is basically GNU with Linux added, or GNU/Linux.
Like BSD, GNU/Linux has a variety of flavors called "distributions". These versions generally have different design goals (security, speed, desktop use) and package a unique set of tools with the kernel to achieve them.

- Hundreds of distributions
- Popular distributions include RedHat, Ubuntu, SuSE, Slackware, Gentoo

Saying "GNU/Linux" every time is tedious, so we will just refer to the entire system as "Linux".
Apple OSX

built using a BSD-based kernel which was renamed “Darwin”

- Arguably the most popular desktop version of UNIX
- A pretty, easy to use experience built on a powerful frame

Steve Jobs Says:

What can the fully compliant UNIX technology in Leopard do? It can run any POSIX-compliant source code. Help you make the most of multicore systems. Put a new tabbed-interface Terminal at your fingertips. Introduce a whole host of new features that make life easier for every developer. Really, what can’t it do?
Advantages/Disadvantages: BSD

Pros
- Reliable and very secure
- Usable on almost anything that uses electricity
- Most flexible license
- Free!

Cons
- Least community/professional support
- Many flavors to choose from
- You thought Linux was for nerdy outsiders?!
Pros

- Built specifically for the hardware it runs on
- Scales really well as system size/load increases
- Lots of support from Sun as well as the community

Cons

- You are paying for Sun’s support and probably the hardware
- Primarily for server use, not super desktop-friendly
Advantages/Disadvantages: Linux

Pros

- Huge community support base
- Free (unless you want professional support)
- Free software to do almost anything
- “wine” allows you to run almost any windows program (even WoW if you are into that sort of thing)

Cons

- Dizzying array of distribution choices
- Lacks some widely-used software (Office, Photoshop etc)
Advantages/Disadvantages: OSX

Pros
- User friendly and just works
- Fully-featured GUI with a powerful terminal
- Supports most of the software the others lack

Cons
- Definitely paying for this one!
- Closed-source, not as flexible as Linux
- Only runs on hardware purchased from Apple (without breaching the EULA)
Why Linux?

- IT’S FREE
- More widely used than BSD or Solaris
- Easy to find beginner’s guides online if you need them
- Basic tools are pretty much standardized
What is the course about?

Powerful text based program launcher. We will learn how to string together simple programs to perform powerful tasks.
A shell is a program that allows the user to interact with the UNIX system:

- read user’s input and parses it
- evaluates special characters
- setup pipes, redirections, and background processing
- find and setup programs for execution
Help, all my images are named wrong!

2007-09-24-picturename.jpg should be

24-09-2007-picturename.jpg

for fn in *.jpg
do mv $fn ‘echo $fn |\
done
What commands do I use most?

```
history | awk '{print $2}' | sort | uniq -c | sort -nr | head
```

229 screen
146 exit
136 ls
81 vi
64 w
47 math
43 cp
33 cd
25 who

23 history
There are primarily two “families” of Unix shells:

- Bourne shell (AT&T) $sh \Rightarrow ksh \Rightarrow bash$
- C shell (Berkley) $csh \Rightarrow tcsh$
- We focus on bash: easy syntax and default in many systems
If you are registered for the course, you have an account with the CS undergrad lab. To access it go to

http://www.csuglab.cornell.edu/userinfo/

You can ssh into the machines in the lab. Instructions are available at the csuglab webpage. For windows machines, can use Putty (free) or Secure Shell (not free) to connect. If you are using any other system you should have the simple tool ssh:

Example Login:

ssh ha232@linus.csuglab.cornell.edu
Installing Linux on your machine

Most modern Linux distros (distributions) are easy to install.

- Disk repartitioning and dual boot automatically set up for you (so you can have Windows and Linux side by side)
- To install Linux:
  1. Download a Linux iso file (CD image file)
     - www.opensuse.org
     - www.ubuntu.com
     - www.fedoraproject.org
  2. Burn the iso image into a CD
  3. Boot your computer from the CD and follow the simple install wizard
  4. Enjoy! :-)

Hussam Abu-Libdeh based on slides by David Slater
If you have a windows machine, there are a few other options:

- **cygwin**: a Linux-like environment for Windows (http://www.cygwin.com/)
- **wubi**: Ubuntu Linux installer for windows. Can dual boot windows and ubuntu without repartitioning (http://wubi-installer.org/)
- **any linux live cd** (http://www.livecdlist.com)
- **Linux on a flash drive!**
- **VMWare**: run any Unix environment within windows

None of the assignments “should” require you to be logged into the csug machines.