CS2043 - Unix Tools & Scripting

Spring 2015 [1]

Instructor: Nicolas Savva

January 21, 2015

[1] based on slides by Hussam Abu-Libdeh, Bruno Abrahao and David Slater over the years

Course Information

- **When**: January 21st - March 4th; 3 times a week
- **Where**: MWF 11:15 - 12:00 in Hollister B14
- **Add Deadline**: January 28th, one week into the course
- **Drop Deadline**: February 4th, two weeks into the course

Course Information

- **Grade**: SX/UX
- **Coursework**: 5 to 6 assignments
  (one of which can be a short final project)
- No textbook. Everything we need can be found in the Unix documentation `man` and the web will be your best friends.
- Ask questions, have discussions with course staff and classmates (use piazza).
- Passing grade for completing all the assignments successfully

Course Information

- **Office Hours**: 6pm-8pm on Friday (tentative)
  and by appointment
- **Email**: nss45 at cornell.edu
- **TAs and additional Office Hours**: TBD
- Website: **http://www.cs.cornell.edu/courses/cs2043/**
- Piazza: **http://piazza.com/cornell/spring2015/cs2043/**
- CMS: **http://cms.csuglab.cornell.edu**
- If you successfully registered you should be on CMS

Course Goals

Overall Goal: Gain an understanding of the Unix environment and scripting.

- We will learn how to string together simple programs to perform extremely powerful tasks.

Some of the topics include:

- Shell, Unix filesystem, basic tools
- Combining tools/commands (pipes)
- Advanced tools
  - Regular expressions
  - Stream manipulation
- Scripting
  - Shell scripting
  - Python scripting
- Potpourri
  - Automation, graphical plotting, VIM, software installation

Prerequisites

- We assume no previous experience with the UNIX environment
- Basic understanding of programming helpful
  (but not necessary)
- The course is a combination of the former
  CS2042 (unix tools) and CS2044 (scripting) courses

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)

What Is a Script?

According to Wikipedia:

- Scripts are programs written for a special run-time environment that can interpret and automate the execution of tasks which could alternatively be executed one-by-one by a human operator

Some characteristics:

- Interpreted (rather than compiled)
- Little use of data structures
- Mostly used for stream processing

Why should one learn this stuff?

"If (you've never used Unix before and) you think you're a power user now, just wait. You don't know what real power is."

-William E. Shotts, Jr.,
The Linux Command Line

A simple example

Task: Change the name convention for one million files:

01-21-2015-picturename.jpg

should become

2015-01-21-picturename.jpg

```
for fn in *.jpg
  do mv $fn ‘echo $fn |\
  sed ‘s/([0-9]+)-([0-9]+)-([0-9]+)/\3-\2-\1/’‘
done
```

Why should one learn this stuff?

- Accomplish and automate complex tasks that would otherwise require arduous manual labor
- Extremely useful computer skills that will remain relevant many years from now
- Unix is a well-stocked toolbox
  (not a giant do-it all Swiss Army Knife)
- We will explore a rich set of small commands and utilities that can be combined in many ways to perform complicated custom tasks.

Another Example

Write a sequence of commands that tells me what Unix commands you use most:

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

```
227 tmux
186 exit
136 ls
81 vi
64 w
57 ssh
43 cp
33 cd
25 who
23 history
```

Short history of UNIX

'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

- Many Linux Distributions: Ubuntu, Fedora, Debian, ...

- Currently, X/Open is responsible for developing UNIX

UNIX Flavors

- Berkeley Software Distribution (BSD)
- GNU/Linux
- Mac OS X
- Sun's Solaris
- IBM AIX
- HP-UX
- Silicon Graphics IRIX

Berkeley Software Distribution

- 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 the free OSes

Advantages/Disadvantages: BSD

Pros

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

Cons

- Conservative (slow progress)
- Least community/professional support
- You thought Linux was for nerdy outsiders?!

Sun Solaris

- Commercial offshoot of BSD
- Designed to run on Sun's SPARC servers, since ported to x86
- Most of the source code was recently released for the OpenSolaris project

Advantages/Disadvantages: Sun Solaris

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

Linux!

- 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 OS's core, or kernel - without other programs it can't really do anything.

Free Software Movement (GNU)

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.

"Think of Richard Stallman as the great philosopher and think of me as the engineer"

-Linus Torvalds

GNU/Linux distros

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, etc

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

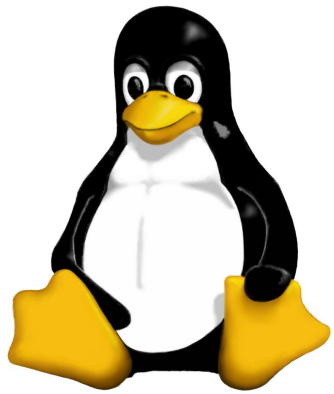
Advantages/Disadvantages: Linux

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
- Privacy preserving distributions are available

Cons

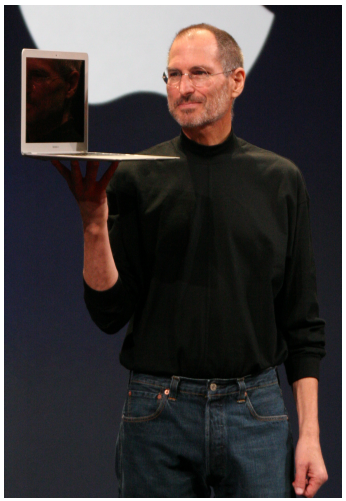
- Dizzying array of distribution choices
- Lacks some widely-used software (MS Office, Photoshop etc)



Mac OS X

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?

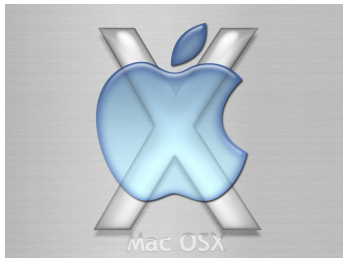
Advantages/Disadvantages: OSX

Pros

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

Cons

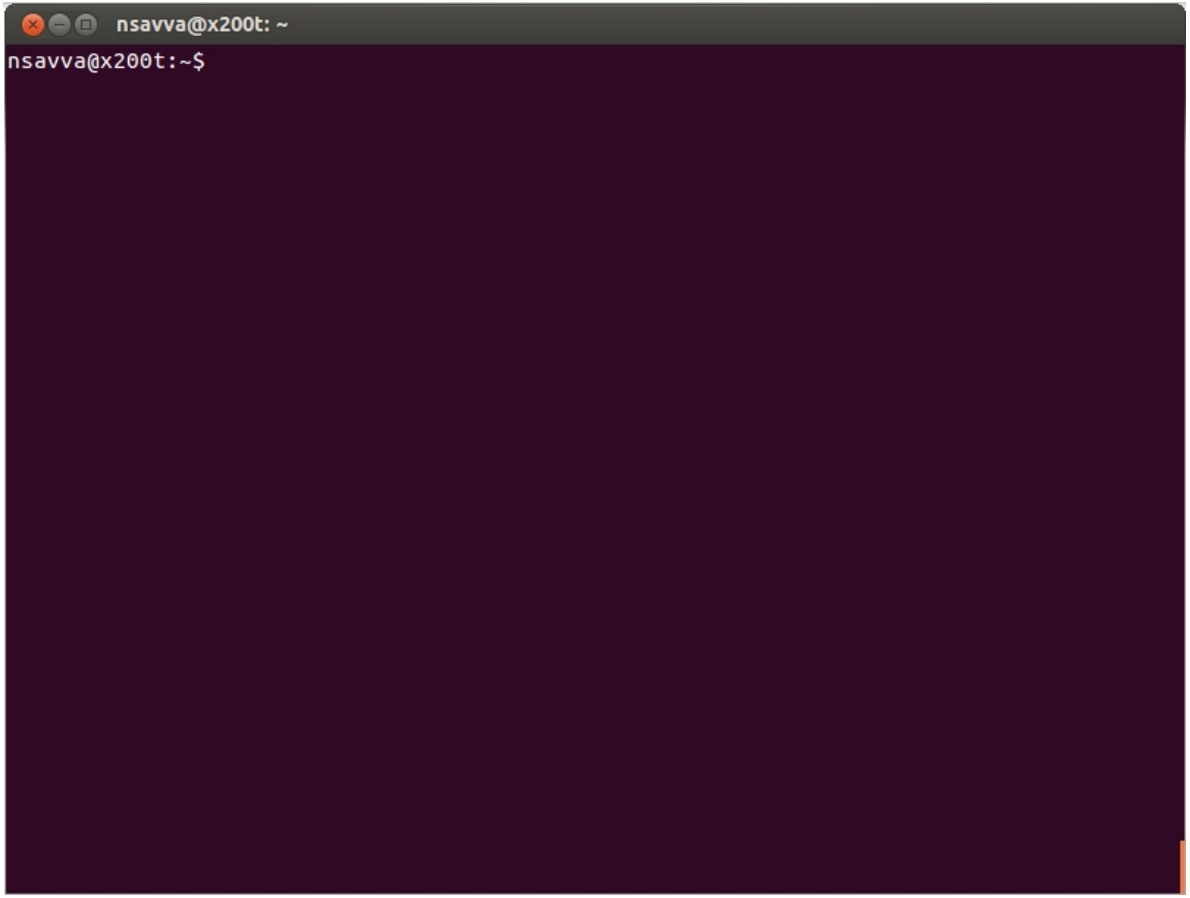
- Definitely paying for this one!
- Closed-source, not as flexible as Linux
- Only runs on hardware purchased from Apple (without breaching the EULA)



Why Linux?

- IT'S FREE
- More widely used than BSD or Sun Solaris
- Easy to find beginner's guides online if you need them
- Basic tools are pretty much standardized

Unix Terminal



- A Powerful text based program launcher
- We will mostly interact using the command line
  (windows and the mouse are from a distant and foreign world)

Unix shells

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

Unix Shells

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

Getting to a UNIX Shell

If you are registered for the course, you have an account with the CS undergrad lab. To access it go to

http://www.it.cornell.edu/support/coecis/cis/labs.cfm

You can ssh into the machines in the lab.
Instructions are available at the csuglab webpage.

Windows users can download Putty (free) to connect to the csuglab machines. People on other systems should be able to use `ssh` from the commandline:

Example Login:

```
ssh NETID@csug07.csuglab.cornell.edu
```

Installing Linux on your machine

Most modern Linux distros (distributions) are easy to install.
A lot of people prefer this method which gives you superuser powers and allows you to install programs using a package manager.

- Dual boot (set up automatically for you) so that you can have Windows and Linux side by side
- To install Linux:
  1. Choose a Linux distribution and download a Linux iso file (CD image file)
     - www.ubuntu.com (user friendly)
     - www.debian.org
     - www.opensuse.org
     - www.fedoraproject.org
  2. Burn the iso image into a CD / copy to a usb stick
  3. Boot your computer from the CD/USB, follow the simple install wizard and enjoy! :-)

Microsoft Windows option

If you have a Windows machine, there are a few other options:

- cygwin: a Linux-like environment for Windows (http://www.cygwin.com/)
- any linux live cd (http://www.livecdlist.com)
- Linux on a flash drive
- VMWare: run any Unix environment within Windows

Apple OSX options

- **OSX**: Install xcode and the package manager Macports www.macports.org
- **IPad/IPhone**: Download an app such as "SSH Term Pro" (not free)

Next lecture

- The first assignment will be out soon (due in a week)
- We are getting our hands dirty
- Learn about the Unix filesystem
- Try some basic commands