CS2043 - Unix Tools & Scripting

Cornell University, Spring 2014[1]

Instructor: Bruno Abrahao

January 22, 2014

[1] Slides evolved from previous versions by Hussam Abu-Libdeh and David Slater

Administrivia

- **When**: January 22nd - March 5th; 3 times a week
- **Where**: MWF 11:15 - 12:00 in Hollister B14
- **Drop Deadline**: February 5th, two weeks into the course
- **Grade**: Successful/Unsuccessful
- **Workload**: 5 or 6 assignments

Administrivia

- Bruno's e-mail: abrahao at cs.cornell.edu
- Website: **http://www.cs.cornell.edu/courses/cs2043/**
- CMS: **http://cms.csuglab.cornell.edu**
- **TAs and office hours**: TBA
- **Piazza**: Ask questions, have discussions with course staff and classmates
- **Text**: No official text. Everything we need is in the Unix documentation: *man (together with the Web) is gonna be your best friend.*
- Try to give us feedback early: This is a short course!

Course Goals

Overall Goal: Intensive training on basic and advanced Unix tools 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 (pipe'ing)
- Advanced tools
  - Regular expressions
  - Stream manipulation
- Scripting
  - Shell scripting
  - Python scripting

What are scripts?

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

A simple example

We need to change the name convention of one million files:

24-09-2007-picturename.jpg

should be

2007-09-24-picturename.jpg

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

Prerequisites

- Not assuming any previous experience with the UNIX environment
- This course is a combination of the former CS2042 (tools) and CS2044 (scripting)
- Basic understanding of programming desirable
  - (but probably not even necessary)

Why you should 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

Why you should learn this stuff?

- Enables us to accomplish and automate complicated tasks that would require arduous manual labor.
- A rich set of small commands and utilities that can be combined in unlimited ways to perform complex custom tasks.
- Rewarding: Extremely useful computer skill that will be relevant many years from now.
- Not limited to preconfigured combinations or menus, as in personal computer systems. Unix is a well-stocked toolbox, not a giant do-it-all Swiss Army Knife.
- It is fun!
- Software Engineering interviews

Typical software engineering interview question

Write a sequence of Unix commands that tells you what programs you use the 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
```

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

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)

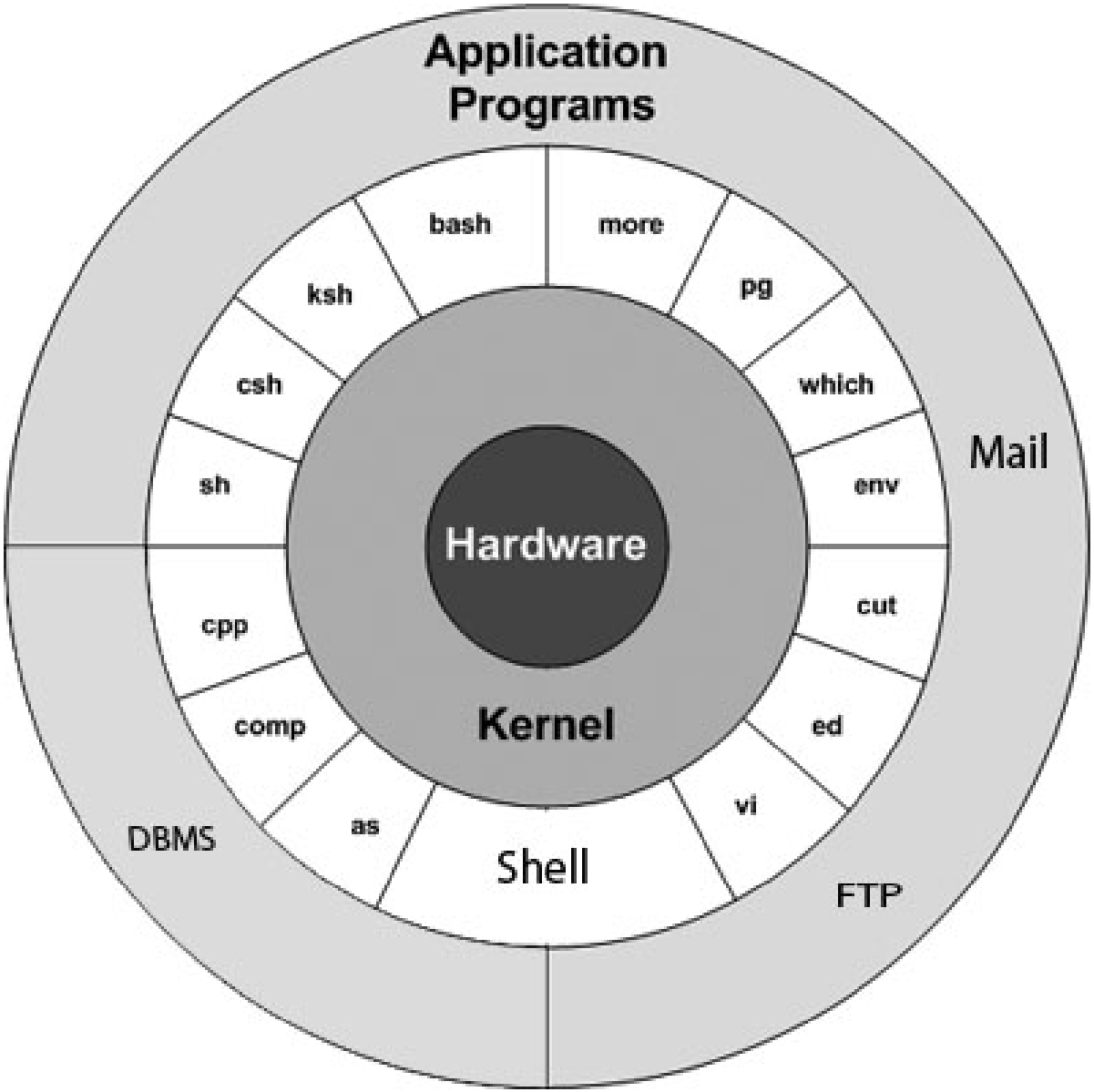
'81 Berkley UNIX 4.1 BSD: vi, C shell, virtual memory

'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 Architecture



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

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 free OSes

Advantages/Disadvantages: BSD

Pros

- Reliable and very secure
- Clean code
- Usable on almost anything that uses electricity
- 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: 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

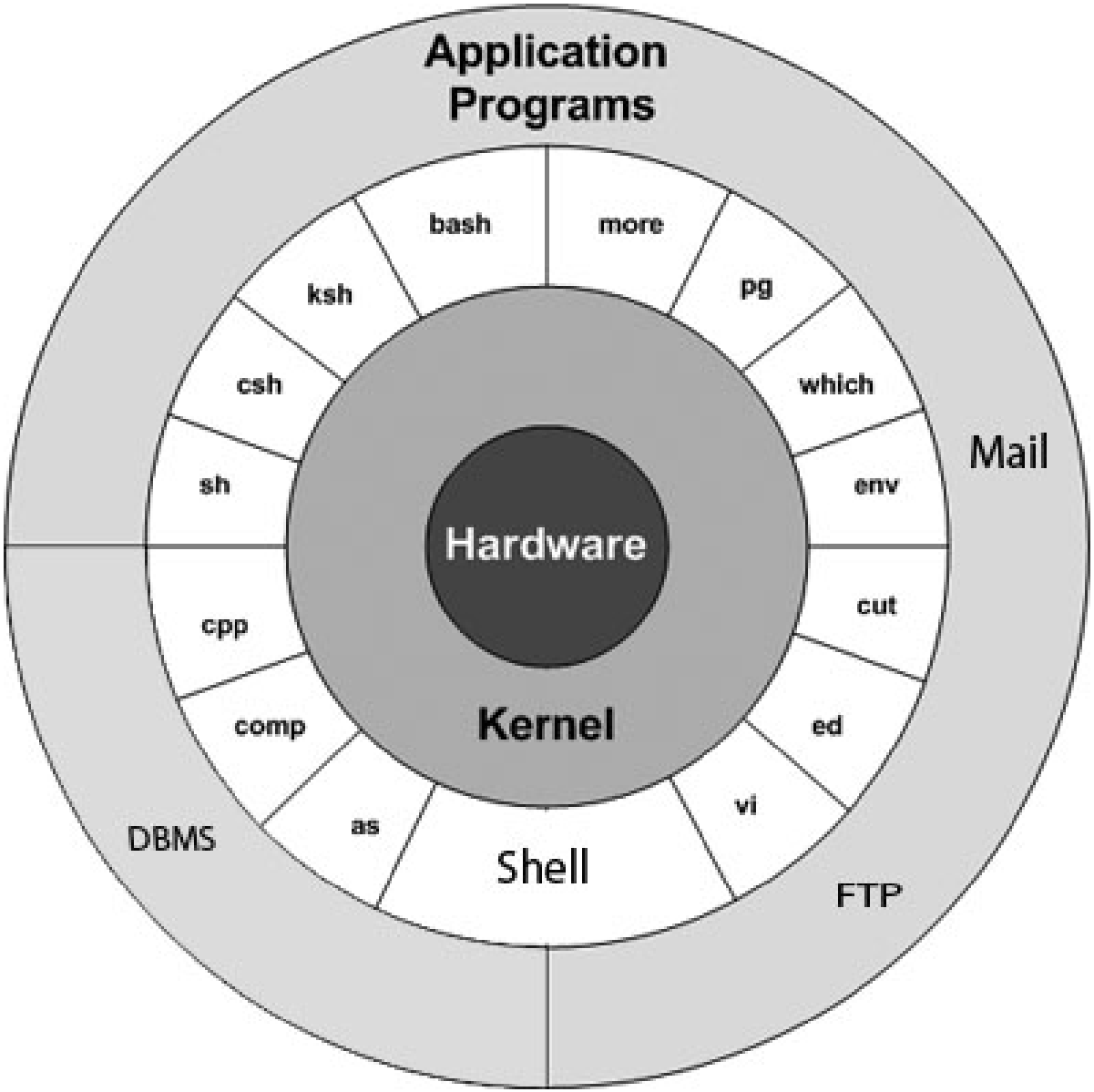
Linux!

- 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 OSs core, or kernel - without other programs it cant really do anything.

Unix Architecture



Free Software Movement

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.

Torvalds says:

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

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, such as RedHat, Ubuntu, SuSE, Slackware, Gentoo, etc.

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

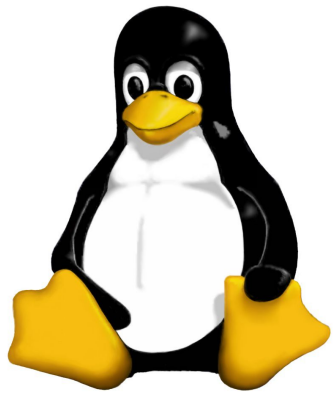
Advantages/Disadvantages: Linux

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
- Some distributions are privacy preserving

Cons

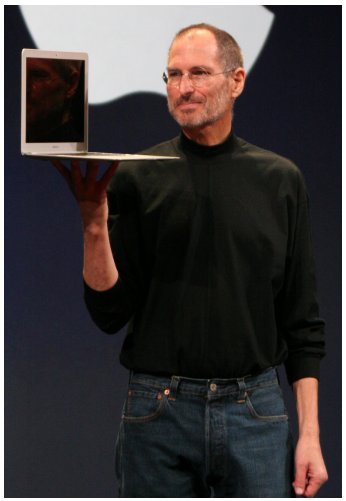
- Lacks some widely-used software (Office, Photoshop etc)



Mac OS X

Built on 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 cant 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

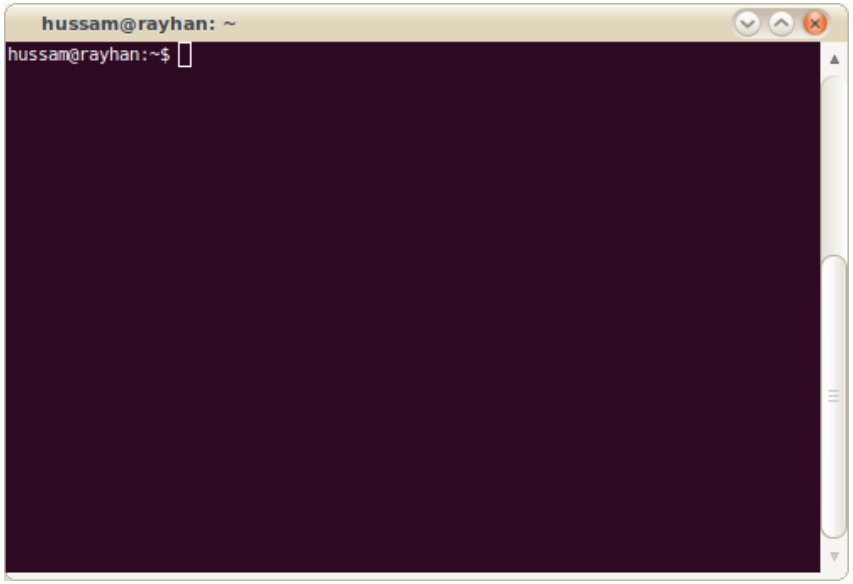
Cons

- 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 Solaris
- Easy to find beginner's guides online if you need them
- Basic tools are pretty much standardized

We're gonna live on a Unix terminal



- We'll interact with the Unix Shell: a text based program launcher.
- Mouse, windows, and clicks are from another (distant) world!

Internet Meme

Getting to a UNIX Shell

If you are registered for the course, you have an account with the CS undergrad lab.

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

You can ssh (secure remote login) into the machines in the lab. Instructions are available at the csuglab webpage. If you use MS Windows, download Putty (free) to connect.

Example Login:

```
ssh ha232@linus.csuglab.cornell.edu
```

Installing Linux on your machine

Preferred method as you get superuser powers and can 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 the Linux iso file (CD image file)
     - www.debian.org (stable, but slow updates)
     - www.ubuntu.com (User friendly, privacy problems)
     - www.gnewsense.org (only free software)
     - www.opensuse.org
     - www.fedoraproject.org
  2. Burn the iso image into a CD and boot your computer from the CD
  3. Follow the simple install wizard and enjoy! :-)

MS Windows

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: Unix environment in Virtual Machine within Windows

Apple

- **OSX**: Install xcode and the pagckage manager Macports (www.macports.org)
- **IPad or IPhone**: Download app, such as "SSH Term Pro" (commercial)

Next time

- We'll get our hands dirty
- **Reminder**: Homework (possibly a survey) will be posted in the course website. Stay tuned!