

# Lecture 1 - Introduction

January 20, 2015

## Welcome

How many have used UNIX?

We're going to learn a lot about UNIX (or Unix-like systems).

- how it works (high level)
- how to program for it
- what makes it such a useful environment for programmers
- some of its weaknesses, and some things Windows gets right

You've probably already seen some UNIX

- using department servers
- Makefiles to build programs
- command line on your Mac

## Introducing UNIX

We log in or open a terminal.

- There are lots of GUI things on Unix.
- But the things that make it unique Unix are mostly at the command prompt.

This is the *shell prompt*

- how we interact with the system
- runs commands
- it's a program itself (you could write one)

Let's run a command:

```
$ ls
```

What does this do? It shows files.

So we have a few concepts:

1. Commands (mostly programs)
2. Files

Let's look at a file:

```
$ cat file.txt
```

A couple things going on here:

- File is full of text
- The command `cat` takes *arguments*

We can chain commands together:

```
$ cat file.txt | tr a-z A-Z
```

- Lots of little programs
- We can put them together

That is the essence of traditional Unix. # The Semester

Go over the syllabus.

- introduce myself when talking about instructor
- show TRACS

**Schedule:** A good chunk of our presentation is going to be driven by what we need for the main assignment sequence. We will slot in additional topics around that.

## How can you get a UNIX?

- use mac or linux
- connect to department servers (using [PuTTY](#) or SSH Secure Shell from Windows)
- VirtualBox (download appliance from TRACS)

We will mostly be using Linux; a lot will also work on Mac and other Unices. We may spend some time with a BSD.

## A bit of history

1. Originated in AT&T Bell Labs
2. Modified by Berkeley
3. Distributed by many (Sun, HP, DEC, Microsoft, IBM, SGI)
4. So did the modern BSDs (from 386BSD and 4.4BSD Lite)
5. Linux arose in the early '90s - kernel + GNU userland
6. Mac OS X came from NEXTStep + FreeBSD + Mach

The Unix system is layered - lots of little pieces that work together. We'll see the importance of this throughout the semester.

Key pieces:

- kernel (talks to hardware)
- userland (programs we run + libraries)
- GUI (a piece of userland, typically, but worth considering separately)

Mac OS X:

- Mach kernel + heavy additions (many from FreeBSD)
- FreeBSD userland (mostly - some GNU tools)
- Proprietary GUI derived from NEXTstep

Typical Linux (GNU/Linux) system:

- Linux kernel
- GNU userland
- X11 GUI (traditional UNIX GUI stack)

Android:

- Linux kernel
- BusyBox + Dalvik + custom stuff userland
- Custom GUI (SurfaceFlinger)

Router:

- Linux kernel
- BusyBox userland
- No GUI (maybe web interface)

FreeBSD:

- FreeBSD kernel
- FreeBSD userland
- X11 GUI

## Back to UNIX

The building blocks of a shell interaction:

1. Commands
2. Arguments
3. Pipelines
4. Redirections

Some useful commands:

1. cat
2. ls
3. grep
4. sort
5. sed
6. man
7. less

Let's look at this last one - `man`. How do we learn about commands?

```
$ man cat
```

We can search:

```
$ man -k sort
```

What does `man` do?

```
$ gunzip -c </usr/share/man/man1/cat.1.gz |nroff -man |less
```