CS2204: Introduction to Unix

January 19\textsuperscript{th}, 2004
Class Meeting 1

* Notes adapted by Christian Allgood from previous work by other members of the CS faculty at Virginia Tech
What is Unix?

- A modern computer *operating system*
- Operating System
  - “a program that acts as an intermediary between a user of the computer and the computer hardware”
  - Software that manages your computer’s resources (files, programs, disks, network)
  - Examples: Windows, MacOS, Solaris, BSD, Linux (e.g. Mandrake, Red Hat, Slackware)
- Modern
  - Stable, flexible, configurable, allows multiple users and programs
Why Unix?

- Used in many scientific and industrial settings
- Open-source operating system (OS)
- Huge number of free and well-written software programs
- Excellent programming environment
- Internet servers and services run on Unix
  - Roughly 65% of the world’s web servers are Linux/Unix machines running Apache
Brief History of Unix

- Ken Thompson and Dennis Richie originally developed the earliest versions of Unix at Bell Labs for internal use in the 1970s
  - Simple and elegant
  - Written in a high-level language instead of assembly language
    - Small portion written in assembly language (kernel)
    - Remaining code written in C on top of the kernel
Unix Variants

- Two main threads of development
  - Berkeley software distribution (http://www.bsd.org)
  - Unix System Laboratories (http://wwwunix.org)
- Sun: SunOS, Solaris
- SGI: Irix
- FreeBSD, OpenBSD, NetBSD
- Hewlett-Packard: HP-UX
- Apple: OSX (Darwin)
- Linux (many flavors)
Brief History of Linux

- Andrew Tanenbaum, a Dutch professor developed MINIX to teach the inner workings of operating systems to his students.
- In 1991 at the University of Helsinki, Linus Torvalds, inspired by Richard Stallman’s GNU free software project and the knowledge presented in Tanenbaum’s operating system, created Linux, an open-source, Unix-based operating system.
- Over the last decade, the effort of thousands of open-source developers has resulted in the establishment of Linux as a stable, functional operating system.
Layers in a Unix-based System

Hardware
- CPU
- Memory
- Disks
- Terminals

Unix Operating System
- Process/memory management
- File system
- I/O

Standard Library
- open, close, read, write

Standard Utility Programs
- Shells, editors, compilers

Library Interface
- System calls

User Interface
- User

User mode

Kernel

Christian Allgood, Virginia Tech, 2004
Unix Structure

- The *kernel* is the core of the Unix operating system, controlling the system hardware and performing various low-level functions. Other parts of a Unix system (including user programs) call on the kernel to perform services for them.

- The *shell* accepts user commands and is responsible for seeing that they are carried out.
Unix Structure (cont.)

- Over two hundred utility programs or tools are supplied with the Unix system. These utilities (or commands) support a variety of tasks such as copying files, editing text, performing calculations, and developing software.

- This course will introduce a limited number of these utilities and tools, focusing on those that aid in software development.
Getting Started

- Logging in to a Unix machine requires an account on that system.
- After logging in, some information about the system will be displayed, followed by a *shell prompt*, where commands may be entered:
  - $
  - %
  - #
  - username@hostname>
  - hostname%
The Shell

- The *shell* is the program you use to send commands to the Unix system.
- Some commands are a single word:
  - `who`
  - `date`
  - `ls`
- Others use additional information:
  - `cat textfile`
  - `ls -l`
Command Syntax

- Commands must be entered exactly
  - `command options argument(s)`
- **Options** modify a command’s execution
- **Arguments** often indicate upon what a command should act (often filenames)
Example Command: ls

- ls -l
- ls -a
- ls -la
- ls -a; ls -l
- ls -al textfile1
- ls -al textfile1 textfile2
- ls -al directory
Always log out when you are done
Use the `exit` command to log out of a shell

Note: If you are running a window system, logging out of a shell only ends that shell. You must also log out of the window system, typically using a graphical menu.