Introduction to UNIX

CS 2204
Class meeting 1
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, ...)
  - e.g. Windows, MacOS
- Modern: features for stability, flexibility, multiple users and programs, configurability, etc.
Why UNIX?

- Used in many scientific and industrial settings
- Huge number of free and well-written software programs
- Open-source OS
- Internet servers and services run on UNIX
- Largely hardware-independent
- Based on standards
Brief history of UNIX

- Originally developed at Bell Labs for internal use in 1970s
- Borrowed best ideas from other OSs
- Meant for programmers and computer experts
- Two main threads of development:
  - Berkeley software distribution (BSD)
  - Unix System Laboratories System V
UNIX variants

- Sun: SunOS, Solaris
- GNU: Linux (many flavors)
- SGI: Irix
- FreeBSD
- Hewlett-Packard: HP-UX
- Apple: OS X (Darwin)
- ...
Parts of the UNIX System

- Kernel
- Shell
- Tools & Apps
Parts (Continued...)

- Kernel: heart of the OS.
- Shell: Interpreter between the user and the computer.
- Tools and applications: Accessible from the shell.
Getting started
Logins and passwords

- You must have an “account” on the UNIX machine you’re trying to use
- “login” is your user name (usually some variant of your real name)
- Your password will not echo as you type
- Remember good password practices
The shell prompt

- After logging in, you get some information about the system, then a shell prompt
- *shell*: The program you use to send commands to the UNIX system
- Prompts take many forms:
  - $ 
  - % 
  - *username@hostname>* 
  - *hostname # %*
Entering commands

- Anytime you see a prompt, you can enter a command for the shell
- Some commands are a single word
  - who
  - date
  - ls
- Others use additional information
  - cat textfile
  - ls -l
Command syntax

- Commands must be entered exactly. If you make a mistake before entering, delete/backspace to fix it. Be careful!

- **command** *options argument(s)*

- Options: modify command’s execution

- Arguments: often filenames that tell a command what to operate on
Example commands: ls

- `ls -l`
- `ls -a -l`
- `ls -al`
- `ls -a; ls -l`
- `ls textfile`
- `ls folder`
- `ls textfile1 textfile2`
- `ls -al textfile`
If you don’t get a normal shell prompt...

- A program is probably running
- If you see a special program prompt, try to quit the program (quit, bye, exit)
- If you see nothing, you can
  - Stop the program with CTRL-z (program will wait until started again)
  - Interrupt the program with CTRL-c (program will usually die)
Ending your session

- **Always** log out when you are done
- Use the command `exit` (also sometimes `logout` or `CTRL-d`)
- Note: if you are running a window system, logging out of the shell only ends that shell. You must also log out of the window system using a menu.