Intro To Unix

CS-121

What is Unix?

- Unix is a multi-user Operating System
- Pros
  - Powerful, reliable, stable, secure
- Cons
  - Designed for programmers (ie not regular people)
  - Difficult to learn at first (though not all versions)

Unix History

- First Version
  - Written in 1969 by Ken Thompson of Bell Laboratories
  - Called UNICS (Uniplexed Operating and Computing System)
  - Later shortened to Unix

Since Then Lot's of Versions!

<<See the Unix family tree>>

About Unix

- Designed from the ground up to be multi-user
  - Different users have different privileges
  - If a user's program crashes it should not affect other users etc.
  - Resources: Memory, CPU Time, Disk-space can all be managed between users

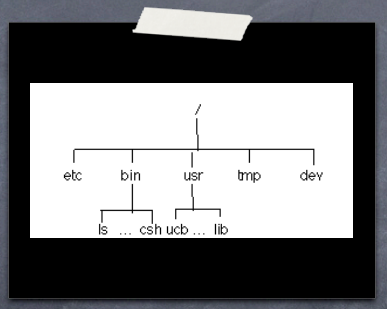
Components in Unix

- Kernel: The OS itself a program that manages resources and access to the hardware
- Shell: A program that allows the user to interact with the computer/OS
  - Graphics Shells
  - Command line shells

We'll be using these pretty standard across all versions of Unix

Interacting with Unix

- Things to know about Unix
  - Unix is case sensitive (for file names for commands for everything CaPiTaLiZaTioN Matters)
  - There's a singe file hierarchy (ie no A: B: C:)
    - Everything start at the root directory /.
    - The file separator is / (not \)



Example File Hierarchy

Important Directory Names

- . (dot) : The current directory
- .. (dot dot) : The parent directory
- ~ (tilde) : Your home directory. A directory which you own. The current directory when you log in.

Command Line Shells

- Different users can use different shells: You can switch at anytime.
  - sh : Shell (first shell ever written)
  - ksh : The Korn shell
  - csh : The sea shell
  - tcsh : The "terrific" C shell
  - bash : The Bourne-again shell

We'll be using this one in class. Default for Idaho Unix accounts. Lots of neat features. "easy" to use!

Common Unix Commands

Some commands are small programs located in /bin/

- ls : Shows files in the current directory
- cat file : Prints the specified file to screen
- cd dir : Changes the current directory
- pwd : Print the current directory
- cp file1 file2: Copy file1 to file2

Other commands are "built-in" to the shell.

These commands are described in detail in your jargon glossary

Common Unix Commands

- mv file1 file2 : Move (rename) file1 to file2
- rm file : Delete (Remove) a file
- mkdir dir : Make a new directory
- man command : Find out information about a specific command. ie: man ls

These commands are described in detail in your jargon glossary

Running a program from the current directory

- When you run a program from the current directory you have to specify the full path for security reasons.
- So to run: program in your current directory you type
  - ./program

Other Unix Programs:Editors

- An editor is a word-processor like program that allows you to edit text files.
- Many Editors Available in Unix
  - vi
  - pico
  - emacs

We'll be learning emacs -- very powerful: editor takes a while to learn.
We'll go over it in class.

Other Unix Programs: Compilers

- A compiler translates a description of a program in a text file into machine code.
- Different compilers for different programming languages: Pascal, Fortran, C, C++, etc.
- We'll be using g++, to compile a program in a file called program.cpp we input
  - g++ program.cpp -o program

Login in to your Unix Account

- We login to our uidaho Unix accounts using ssh (The secure shell)
  - Secure shell is a shell that works on your current computer and sends all commands to another shell (in our case bash) running remotely on another computer.
  - ssh is secure because everything is encrypted between both machines.

Login in to your Unix Account

- From Home
  - Download (Links are on the website)
    - Putty - Putty is a simple ssh client for windows use it to login remotely to Unix machines
    - Psftp - is used to transfer files between machines

Login in to your Unix Account

- From any ITs Windows Lab:
  - Start->Programs->SSH Secure Shell->SSH Client (this may very sightly from one machine to another).

No Graphics Shell with Putty!

- You can't point and click.
- All of your commands will have to be accessed from the keyboard.

If you want a graphics shell..

- If you run any kind of Unix OS: Linux, Mac OS X etc. Just open up a terminal and type:
  - ssh -X will7759@sunsol.uidaho.edu
  - here will7759 is your user name
  - You MUST be running X11 (which is essentially your graphics shell)
  - X11 comes free with all Unix OS

If you want a graphics shell on Windows

- If you run Windows you can use cygwin
  - Cygwin is a Unix emulation layer for windows.
  - It includes X11

When using a graphics shell..

- You need a very fast internet connection
- Cable and DSL may be Okay
- But not dial-up :-(

Login in Example

Working with Files

CS-120

Review: I/O using cin, cout

- cin, cout : Standard input and standard output.
- We can do simple I/O operations by simply redirecting input or output on the command line in Unix.
- ./my_program < input_file > output_file

cerr, the other standard stream

- Besides cin, and cout -- there is cerr
  - cerr is like cout except that it is meant to output error messages
  - cerr is useful : when you redirect output, you'll still see an error on the screen

Redirecting cerr

- To redirect both cout and cerr from Unix do
  - ./program &> output_file.txt

Redirecting stderror/ stdoutput

- ./program 2>err.out 1>output.out

Redirecting Output to Another program

- ./program1 | ./program2
- Examples
  - ls | less
    - ls lists all directories
    - less shows input one page at a time