CS35L-5

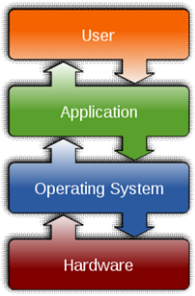
Week 1 - Day1

GNU/Linux

- Open-source operating system
  - **Kernel**: core of operating system
    - Allocates time and memory to programs
    - Handles file system and communication between software and hardware
  - **Shell**: interface between user and kernel
    - Interprets commands user types in
    - Takes necessary action to cause commands to be carried out
  - **Programs**

Operating System

- Interface between application and hardware
  - Process Management
  - Memory Management
  - File Systems
  - Hardware Drivers
  - Networking
  - Security



http://en.wikipedia.org/wiki/Operating_system

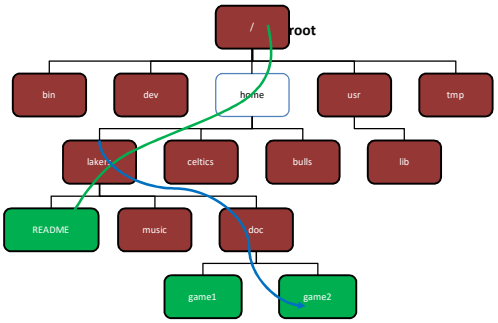
Why Linux?

- Cost
  - Free
  - Free applications
- Quality
  - Stability
  - Reliability
- Open Source
  - Good community, heavily documented
  - Widely used among researchers and developers
  - Customizable at OS level

Files and Processes

- Everything is a **file (or process)**:
  - **File**: collection of data
    - A document
    - Text of program written in high-level language
    - Executable
    - Directory
    - Devices
  - **Process**: an executing program identified by PID

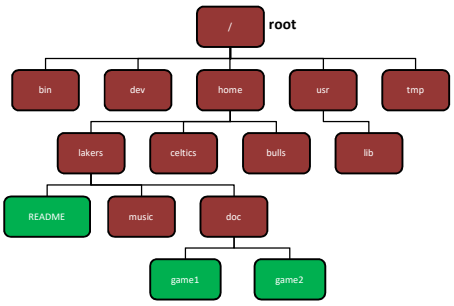
Absolute Path vs. Relative Path



Current directory: home

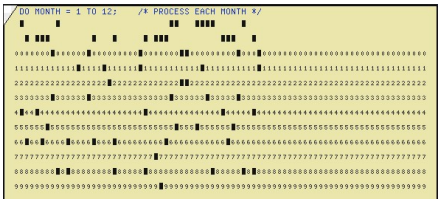
Unix File System Layout

- Tree structured hierarchy



Interacting with Computers - History

1) Switches and Lights
2) Batch Processing
3) Teletype - CLI
4) GUI



User Interfaces: CLI vs GUI

CLI - Command Line Interface
- Steep learning curve
- More control
- Cumbersome multitasking
- Speed with commands
- Low resources
- Power of scripting

GUI - Graphic User Interface
- Intuitive to use
- Limited by interface
- Easy multitasking
- Limited by pointing
- Graphic eats resources
- Difficult to automate

The Basics: Moving Around

- **pwd**: print working directory
- **cd**: change directory
  - **~** home directory
  - **.** current directory
  - **/** root directory, or directory separator
  - **..** parent directory

The Basics: Shell

- **<up arrow>**: previous command
- **<tab>**: auto-complete
- **!!**: replace with previous command
- **![*str*]**: refer to previous command with str
- **Format**: command -options arguments

  eg. ls -l prefix*

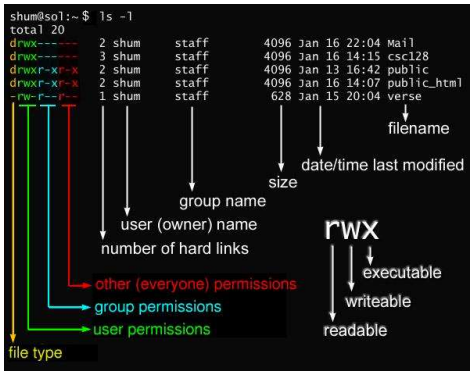
The Basics: Dealing with Files

- **mv**: move/rename a file
- **cp**: copy a file
- **rm**: remove a file
  - r: remove directories and their contents recursively
- **mkdir**: make a directory
- **rmdir**: remove an empty directory
- **ls**: list contents of a directory. Various useful options.
- **cat:** concatenates contents of file
- **head:** displays first n lines of file
- **tail:** displays last n lines of file

The Basics: Changing File Attributes

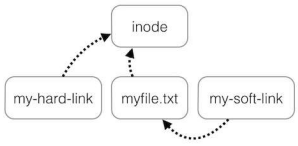
- **touch**: update access & modification time to current time
  - touch *filename*
  - touch -t 201101311759.30 *filename*
    - Change filename's access & modification time to (year 2011 January day 31 time 17:59:30)

Linux File Permissions



ln

- **ln**: create a link
  - Hard links: point to physical data
  - Soft links aka symbolic links (-s): point to a file



Linux File Permissions

- chmod
  - read (r), write (w), executable (x)
  - User, group, others

Reference	Class	Description
u	user	the owner of the file
g	group	users who are members of the file's group
o	others	users who are not the owner of the file or members of the group
a	all	all three of the above, is the same as *ugo*

The Basics: chmod (symbolic)

Operator	Description
+	adds the specified modes to the specified classes
-	removes the specified modes from the specified classes
=	the modes specified are to be made the exact modes for the specified classes

Mode	Name	Description
r	read	**r**ead a file or list a directory's contents
w	write	**w**rite to a file or directory
x	execute	e**x**ecute a file or recurse a directory tree

The Basics: find

- -type: type of a file (e.g: directory, symbolic link)
- -perm: permission of a file
- -name: name of a file
- -user: owner of a file
- -maxdepth: how many levels to search
- Examples
  - find . -name my*
  - find . -name my* -type f
  - find / -type f -name myfile -print

The Basics: chmod (numeric)

#	Permission
7	full
6	read and write
5	read and execute
4	read only
3	write and execute
2	write only
1	execute only
0	none

- Usage

– chmod [''references''][''operator''][''modes''] ''file1'' ...
Example: **chmod** ug+rw mydir, **chmod** a-w myfile,
Example: **chmod** ug=rx mydir, **chmod** 664 myfile

File Name Matching

- ?: matches any single character in a filename
- *: matches one or more characters in a filename
- []*:* matches any one of the characters between the brackets. Use '-' to separate a range of consecutive characters.

  ls -l [a-zA-Z]*

Lost? man

- Extensive documentation that comes preinstalled with almost all substantial Unix and Unix-like operating systems
- Usage
  - read a manual page for a Linux command
    - **man <**command_name**>**
    - Hit "q" to get out of man page

wh… Commands

- **whatis** <command>: returns Name section of man page
- **whereis** <command>: locates the binary, source, and manual page files for a command
- **who, whoami,which**

More Linux Commands

- **Pipeline**
  - Redirect the output of the first tool to the input of the following one.
  - Often useful with | grep
  - **man grep | grep pattern**

- **Redirection**
  > : write stdout to a file (creates if not present else overwrites)
  >>: append stdout to a file (creates if not present)
  < : use contents of a file as stdin

More commands: Manage Processes

- ps: report a snapshot of current processes
- top: dynamic view of linux tasks
- kill: send a terminate signal to a process