

# UNIX File Systems

System Administration Decal

Spring 2009

Intermediate Lecture #2

George Wu

Slides by: Joshua Kwan

# Last time...

- Commands: `cd`, `ls`, `pwd`, `mkdir`, `rmdir`, `cp`, `mv`, `rm`, `grep`, `find`, `locate`
- Pipes – chain output of one command into input of another.
- Input/output redirection with `>` and `<`
- Command substitution
- `/dev/null` – the ‘bit bucket’, or a file that’s always empty

# Today

- How are files, programs, and user directories organized in Unix?
- Types of files: it isn't just "files" and "directories"
- How do you do 'shortcuts' in Unix like aliases in OS X or shortcuts on Windows?

# Windows Filesystem Hierarchy

- Your data in “C:\Documents and Settings”
- Core system files, libraries in “C:\windows”
- Extra programs in “C:\Program Files”

# Mac OS Filesystem Hierarchy

- Your data in “/Users/yourname”
- Core system files in “/System”
- Extra programs in “/Applications”
- Configuration data in “/Library”
- Despite being much more concise/readable, a relative of...

# UNIX Filesystem Hierarchy

- / - root directory, generally contains nothing
- /home – personal user directories
- /etc – contains system configuration files
- /bin – contains core system programs
- /sbin – contains core system admin tools
- /dev – contains device files (more later)

# UNIX Filesystem Hierarchy

- /boot – contains the kernel image, etc.
- /lib – contains system libraries (think DLLs)
- /tmp – contains temp files, writable by all
- /usr – contains nonessential data and programs
- /var – contains “database” type information

# /proc and /sys

- Linux specific filesystems that are *really useful!*
- They contain meta-information about the computer, e.g.
  - /proc/cpuinfo – show CPU info
  - /proc/modules – show loaded kernel drivers
  - /proc/sys – kernel “tunables” (Advanced!)
- Won't study /sys in this class; similar



# UNIX FS in Practice: Ubuntu

- **/bin**/ls: Duh
- **/sbin**/shutdown: Power off/reboot machine
- **/var**/lib/dpkg: Contains information about the packages installed on your machine
- **/usr**/share/{doc,man}: Documentation, `man` pages for all packages
- **/etc**/cron.d: Configure scheduled tasks
- **/dev**/hda1: Device file for first partition of first IDE disk on system

# Administrivia

- Do your labs! Include your cs198-XX username, sorry I forgot to make that clear.
- Don't be afraid to ask questions – come to OH or email me with specific questions.
- Use `ilinux1.eecs.berkeley.edu` and `bash` for homework assignments if you don't have Linux at home.
- We *will* have a shell-scripting lecture!

# 4 Types of Files

- Should be familiar:
  - Regular files are exactly what you think; simply data files, e.g. text, music, programs
  - Directories: Simply contain other files.
- Probably not familiar:
  - Device files: look like normal files, but reading/writing them interacts with kernel, like `/dev/hda1` from before
  - Named pipes.. described by example!

# Just for Fun: Named Pipes

**mkfifo my\_pipe**

**gzip -9 -c my\_pipe > out.gz**

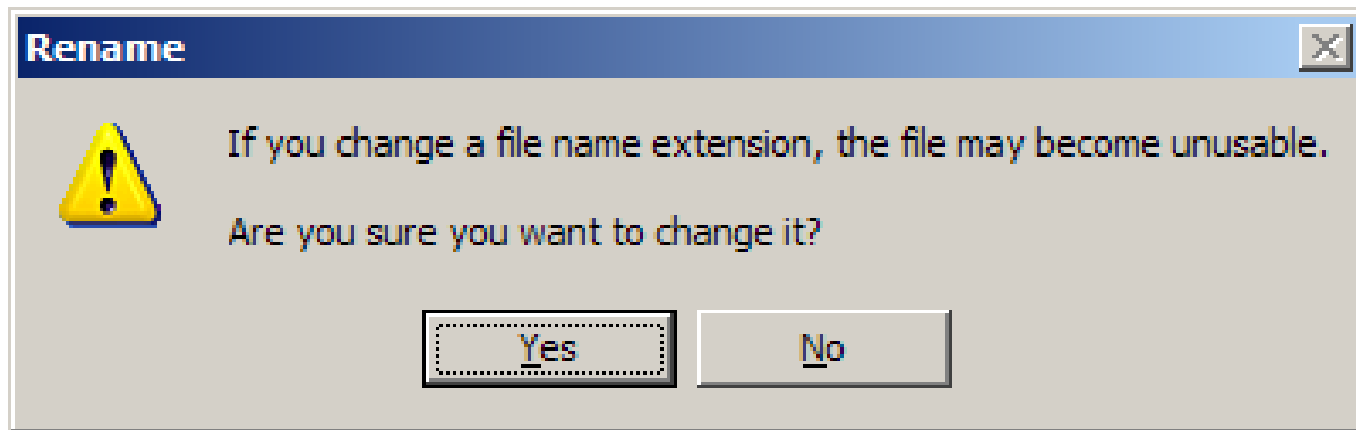
**Usage: cat file.txt > my\_pipe**

# Symbolic and Hard Links

- Like 'shortcuts'; use the **ln** command to make them.
- Symbolic links (**ln -s**): If you remove the file the shortcut points to, the link is broken!
- Hard link (**ln**): like an 'extra pointer' to the same data on disk. Looks like a copy, doesn't take up extra space.

# File Extensions

- On Windows, file extensions are very scary and meaningful!



- Unix? Not so much. “A rose by any other name smells just as sweet.”

# File Extensions

- You can use the **file** command to identify files based on their *content*, not their extension.

```
joshk@nigiri:~$ file music3.doc
music3.doc: Rich Text Format data, version 1, ANSI
joshk@nigiri:~$ mv music3.doc music3.mp3
joshk@nigiri:~$ file music3.mp3
music3.mp3: Rich Text Format data, version 1, ANSI
joshk@nigiri:~$ mv music3.mp3 music3
joshk@nigiri:~$ file music3
music3: Rich Text Format data, version 1, ANSI
```

- **Some** Linux programs will infer based on the extension, *but it's not central to the OS like on Windows.*