

Lecture 2: The Filesystem

System Administration Decal

Spring 2010

Michael Gasidlo

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 OSX or shortcuts in Windows?

A Comparison: Windows

- User data in “C:\Documents and Settings”, or “C:\Users” on newer versions
- Core system files and libraries in “C:\Windows”
- User-added programs in “C:\Program Files”

A Comparison: OS X

- User data in “/Users”
- Core system files in “/System”
- User-added programs in “/Applications”
- Configuration data in “/Library”
- Despite being concise and readable, a descendant of. . .

UNIX Filesystem Hierarchy

- / - the root (usually contains nothing but other directories)
- /home – user data
- /etc – system configuration files
- /bin – core system programs
- /sbin – core system administrative programs
- /dev – device files (more on this later)

UNIX Filesystem Hierarchy

- /boot – files the system needs to boot, including the kernel
- /lib – system libraries (sort of like DLL's)
- /tmp – the temporary storage folder, writable by all
- /usr – non-system programs and data
- /var – database information and logs

/proc and /sys

- Linux-specific filesystems that are *really useful*
- They contain meta-information about the computer
 - /proc/cpuinfo – show CPU info
 - /proc/modules – show loaded kernel drivers
 - /proc/sys – kernel “tunables”
- Won't study /sys in this class

Some Examples

- **/bin/ls** – duh
- **/sbin/shutdown** – shutdown/reboot the system
- **/var/lib/dpkg** – lists information on installed packages
- **/usr/share/{doc,man}** – documentation and *man* pages for all packages
- **/etc/cron.d** – schedule tasks
- **/dev/hda1** – device file for first partition of first ide disk on the system

The 4 types of “files”

- Two of these should be familiar:
 - Regular files – data, text, music, programs, etc.
 - Directories – contain other files
- Two of these may be new to you:
 - Device files – look like normal files, but reading/writing to them interacts with the kernel (like `/dev/hda1` from before)
 - Named pipes

Named Pipes

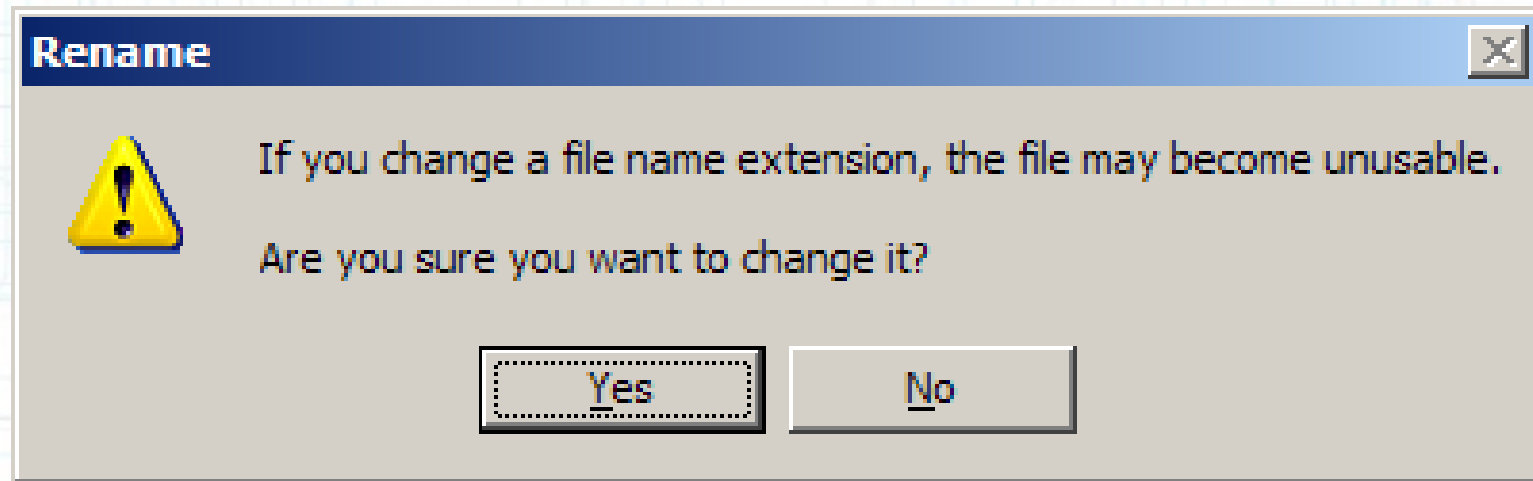
- We've already seen one type of pipe: “|”
- The other type of pipe: the named pipe
- Also commonly called “FIFO”- **F**irst **I**n **F**irst **O**ut
- Works just like an anonymous pipe, but has a name and location in the filesystem
 - eg: `$mkfifo my_pipe`
`$gzip -9 -c < my_pipe > out.gz`
`$cat file.txt > my_pipe`

Links

- Use “ln” to make links
- Symbolic links (**ln -s**) – like a “shortcut” to another file in the filesystem. If you remove the original file, the link is broken.
- Hard links (**ln**) – adds another pointer to the same data on the hard disk. Programs will see hard links as a regular file. If you remove the original file, the hard link remains unbroken, and the data is not erased from the disk.

File Extensions

- On Windows, file extensions are very important



- UNIX? Not so much. “A rose by any other name would smell as sweet.”

Administrivia

- The Sun computers in the lab may lack certain features that we'll be covering
- Use `tsunami.ocf.berkeley.edu` to do the labs if it doesn't work on the local machines
- We are still working the kinks out of **submit** and **glookup**; for now continue to e-mail lab reports

File Extensions

- You can use the **file** command to identify file types by their *content*, not their *extension*

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

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