

Unix File System



Class Meeting 3

* Notes adapted by Alexey Onufriev from previous work by other members of the CS faculty at Virginia Tech

Navigating through the maze: unix file system



- The **file system** is your interface to:
 - physical storage (disks) on your machine
 - storage on other machines (NFS)
 - input/output devices
- *Everything* in Unix is a **file** (programs, text files, peripheral devices, terminals)
- The filesystem provides a *logical* view of the storage devices
- **Directory** is a file to contain (references to) other files

Working Directory (review)



- **Working directory:** your current position in the file system
- `pwd` (print working directory) command outputs the **absolute path** (more later) of your working directory
- Unless you specify another directory, a command will assume that you want to operate within the working directory

Home Directory (review)

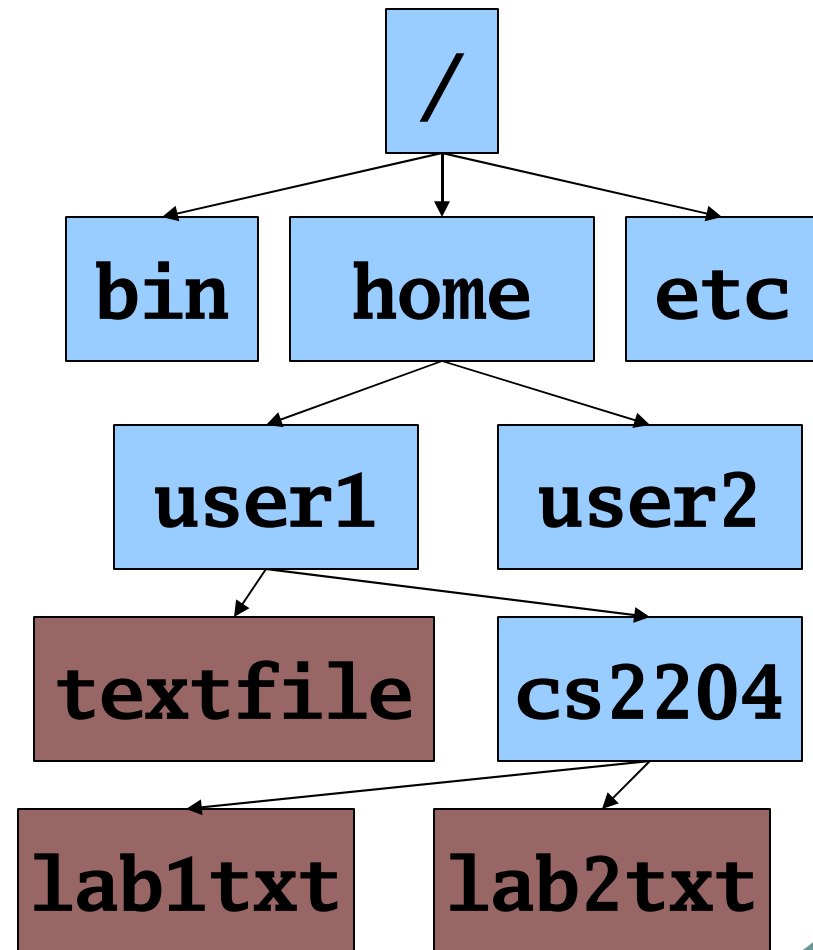


- **Home directory**: personal user space
- At login, your working directory will be set to your home directory
- The **path** (more later) to your home directory can be referred to by the ~ (tilde) symbol
- The home directory of `user1` can be referred to by `~user1`

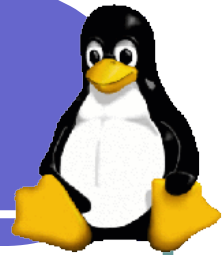
Unix File Hierarchy



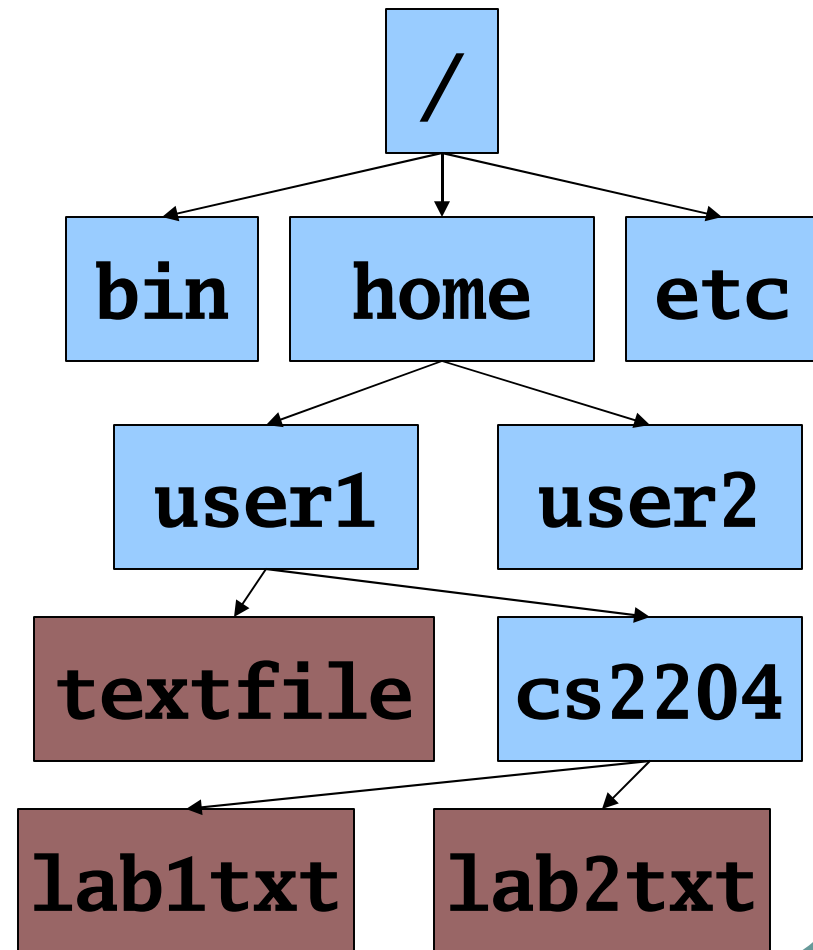
- **Root Directory:** /
- Directories may contain **plain files** and **other directories**
- Result is a tree structure for the file system



Unix Paths (review)



- Separate directories by the symbol /
- **Absolute Path**
 - start at the root and follow the tree
- Examples:
 - `/home/user1/textfile`
 - `~user1/textfile`
 - `~/textfile`



Unix Paths (cont)

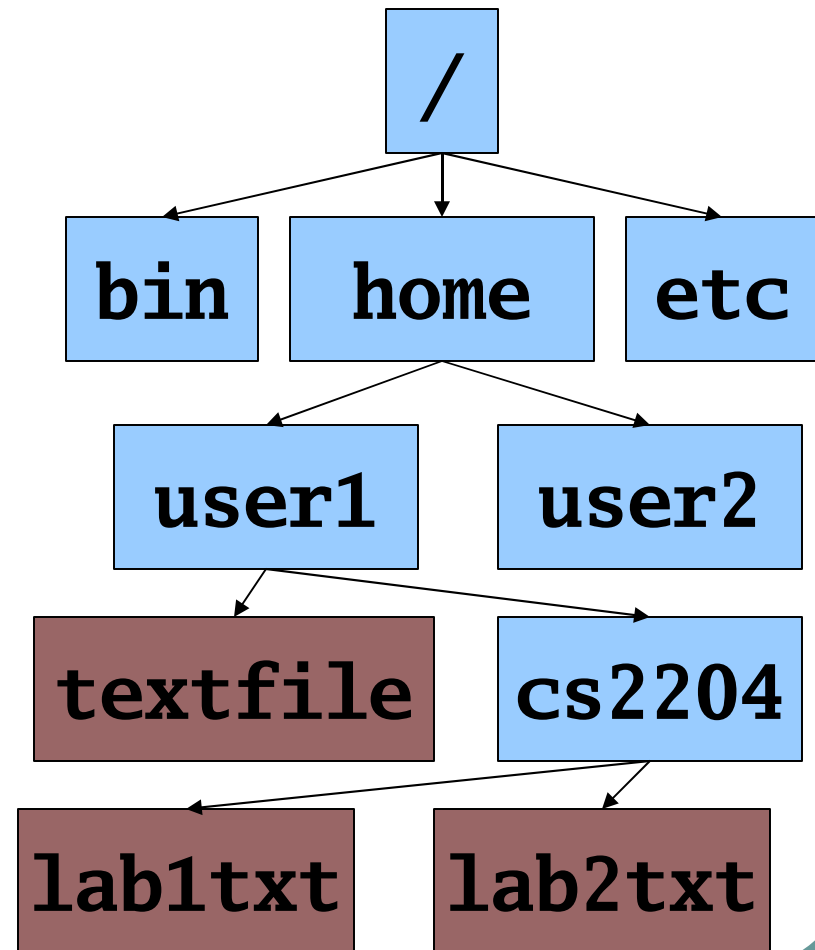


- **Relative Path**

- start at working directory
- .. – level above
- . – working directory

- **Examples:**

- `textfile`
- `cs2204/lab1txt`
- `../user2`



Some Standard Directories

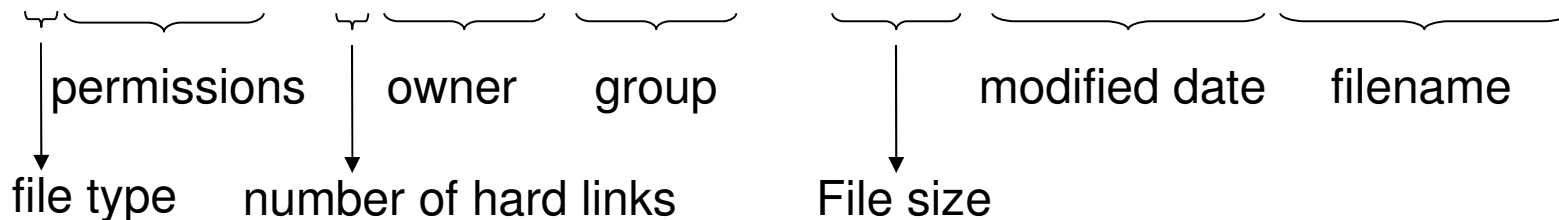


- **/** – root directory //generally need superuser access to change anything
- **/bin** – standard commands and utilities; **executables**
- **/dev** – block and character device directory
- **/etc** – host-specific **configuration**; host **services**
- **/home** – users' home directories
- **/lib** – library directory
- **/sbin** – system commands and utilities (needed to boot)
- **/tmp** – temporary files
- **/usr** – user utilities and applications; **/usr/local/**
- **/var** – system files that vary (logs, spools, email)

Partial Output of `ls -lFa`



```
drwxrwxr-x    2 cs2204  cs2204           64 Aug 29 11:13 ./
drwxr-xr-x    2 cs2204  cs2204           64 Aug 22 17:29 ../
-rw-r--r--    1 cs2204  cs2204        1994 Aug 27 10:42 announce.php
drwxr-xr-x    2 cs2204  cs2204           64 Aug 22 12:45 assignments/
-rw-r--r--    1 cs2204  cs2204        5773 Aug 22 08:23 calendar.php
lrwxrwxrwx    1 cs2204  cs2204           0 Aug 29 11:12 home ->
/home/courses/cs2204/
-rw-r--r--    1 cs2204  cs2204         475 Aug 22 12:38 index.php
drwxr-xr-x    2 cs2204  cs2204           64 Aug 24 16:24 labs/
drwxr-xr-x    2 cs2204  cs2204           64 Aug 24 16:07 notes/
-rw-r--r--    1 cs2204  cs2204       1210 Aug 25 09:20 readings.php
drwxrwxr-x    2 cs2204  cs2204           64 Aug 24 16:19 resources/
-rw-r--r--    1 cs2204  cs2204       18524 Aug 22 11:58 syllabus.php
```



Types of Files



- Plain (-)
 - Most files, binary or text
 - Note: Unix doesn't recognize any special filename extensions
- Directory (d)
 - Directory is actually a file
 - Points to another set of files
- Link (l): Pointer to another file or directory
- Special
 - b – block device (disks, CD-ROM)
 - c – character device (keyboard, joystick)

File Ownership



- Each file has a single **owner**
- `chown` command can be used to change the owner; usually only **root** can use this command
- Each file also belongs to a single **group**
- Groups may have different permissions than everyone else

File Permissions



- Permissions are used to allow or disallow access to files or directories
- Three types of permission:
 - Read (r)
 - Write (w)
 - Execute (x)
- Permission exists on three levels:
 - User (u)
 - Group (g)
 - World (o)

File Modification Date



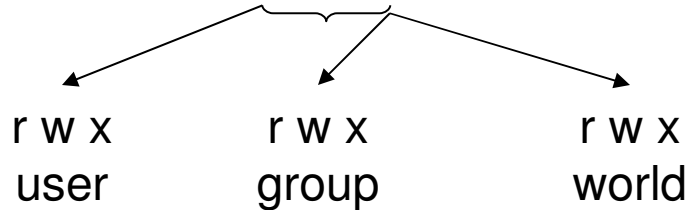
- Last time a file was changed
- Useful when . . .
 - there are many copies of a file
 - many users are working on a file
- `touch` command can be used to update the modification date to the current date (or to create a file if it does not yet exist)

File Permissions (cont)

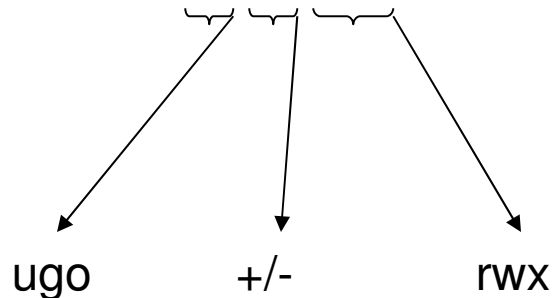


- `chmod <mode> <file(s)>`

- `chmod 700 textfile`



- `chmod g+rw textfile`



- `g+rw` changes permissions to 760 (octal)

More on “ls”. Regular expressions



- `ls -d */` // lists only names of the directories
- `ls -lh` // in human readable format, e.g Kb, GB, etc.
- `ls R*`
- `ls *R*`
- `ls ???R??` // list files and directories whose names are exactly 6 characters, the 4th being “R”.
- `ls [123]R` // list files and directories whose names are 1R, 2R or 3R.

Examples of “regular expression” - an extremely useful concept in UNIX (string matching)

Working with file names



- Most of the unix commands covered here that take file names as arguments can also use regular expression wildcards
 - * for any string, e.g. *.txt, obj*, a*.*
 - ? for any character, e.g. doc?
 - [] around a range of characters, e.g. [a-c]*
- Many systems provide filename completion - press the TAB key