Shells

Program to execute

```
% gcc –o hello hello.c
```

Arguments

- The % is a shell prompt.
- Shells

  also execute config on launch

  – accept commands (programs) as input
  – finds the executable
  – interprets the arguments
  – starts executing the command
- Shells also have some “built-in” commands.

How?

Use PATH

Remember
public static void main(String args);

Which shell program?

- sh - Bourne Shell
  - the original shell
  - when we talk about shell programming we will use sh

- tcsh - C Shell
  - more C-like syntax
  - originally was better for command line

- bash - Bourne Again Shell
  - superset of sh
  - default on Linux mostly
  - default on new teach.cs accounts

Changing your shell

- chsh <username> bash

add or change user database info
-s newshell —— attempts to change user's shell to a new shell

Running a program

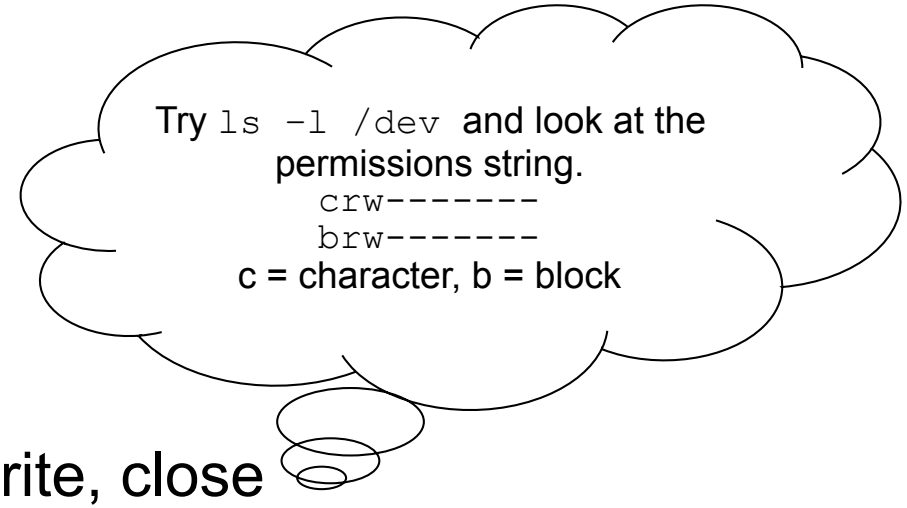
```
% gcc –o hello hello.c
% hello
```

- load a program into memory and hand it off to the OS to run the program.

Files and Directories

a file is a sequence of bytes

- "Everything is a file."
- Unix provides a file interface for all Input/Output.
  - regular files
  - directories
  - devices
    - video (block)
    - keyboard (character)
    - sound (audio)
    - network (block)
- File interface = open, read, write, close

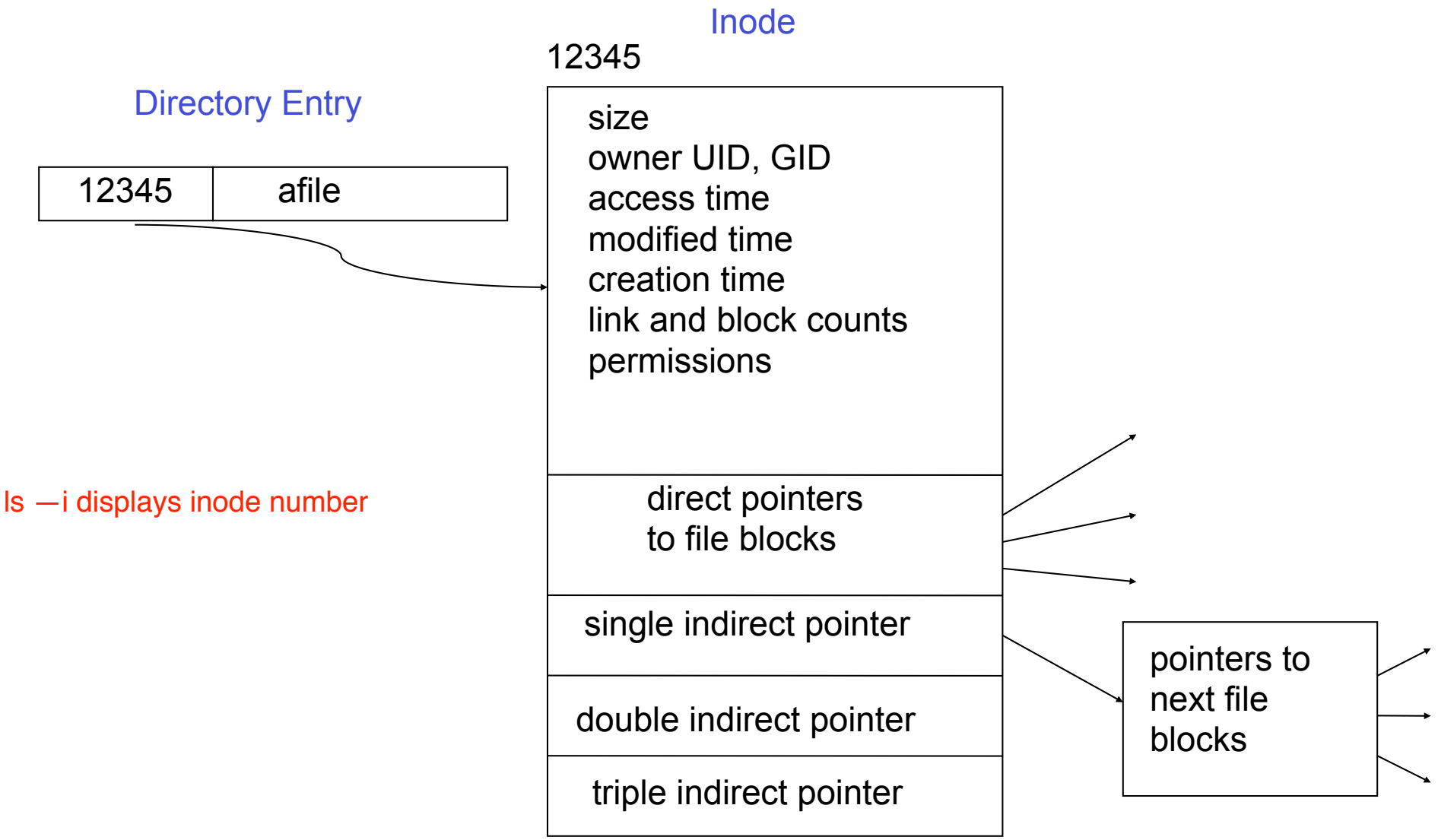


File System Hierarchy

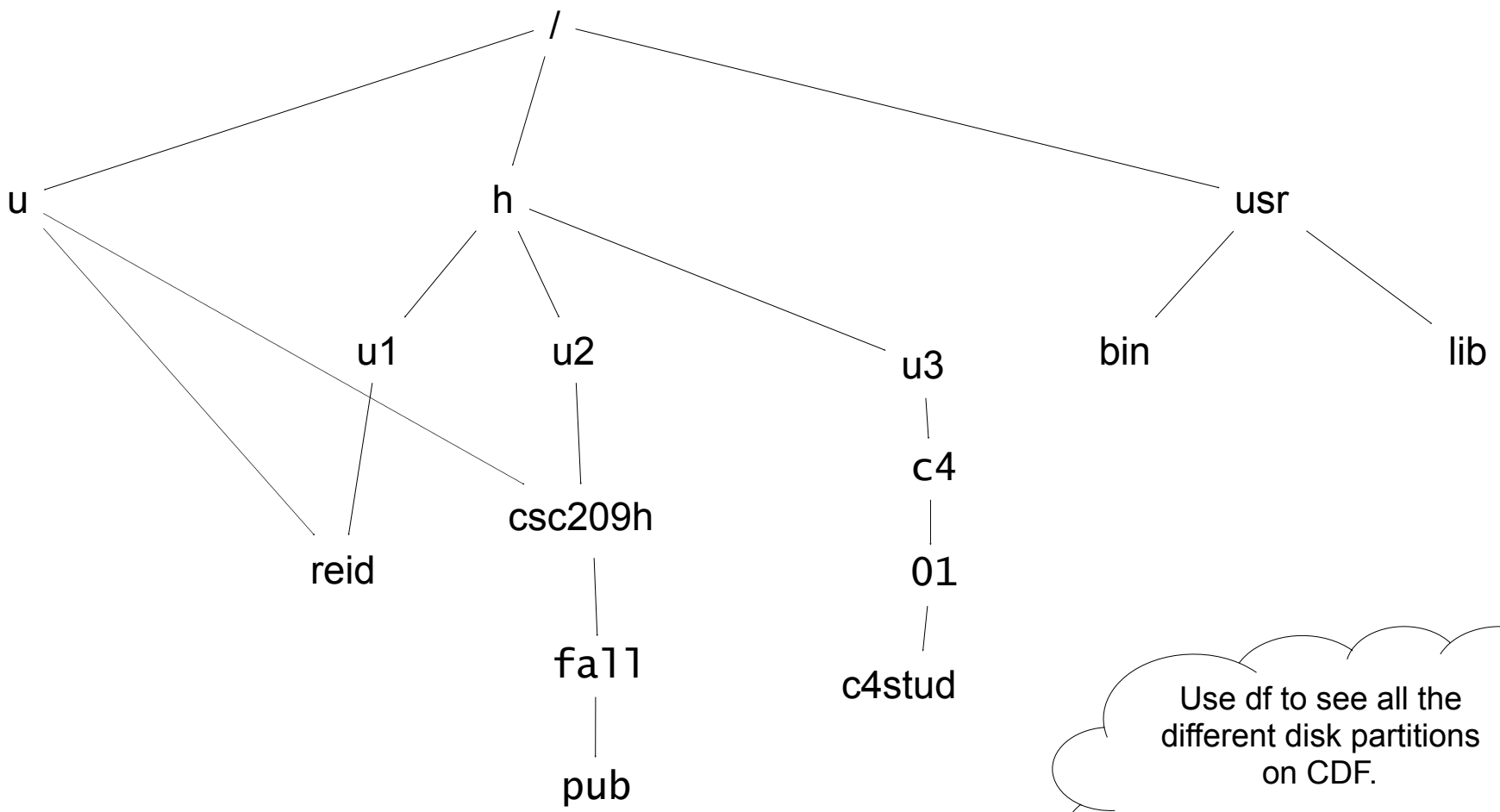
- Everything starts in the "root" directory whose name is "/"
- A directory is a file that contains directory entries.
- A directory entry maps a file name to an inode.
- An inode is the data structure that contains information about a file, including which disk blocks contain the file data.

he inode is a data structure in a Unix-style file system which describes a filesystem object such as a file or a directory. Each inode stores the attributes and disk block location(s) of the object's data. Filesystem object attributes may include metadata (times of last change, access, modification), as well as owner and permission data.

Inodes and Directory Entries



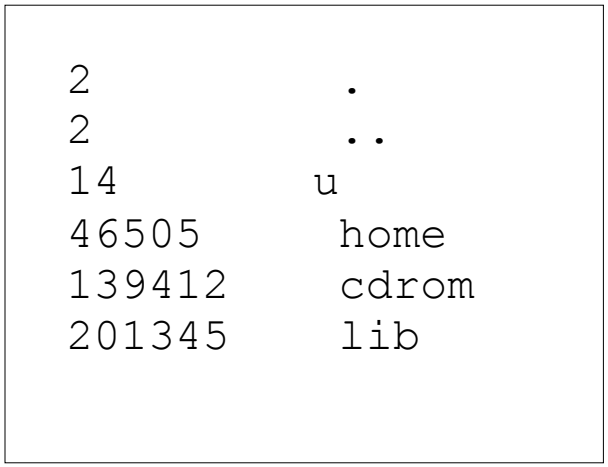
File System Hierarchy



Use df to see all the different disk partitions on CDF.

Directories and Links

directory file



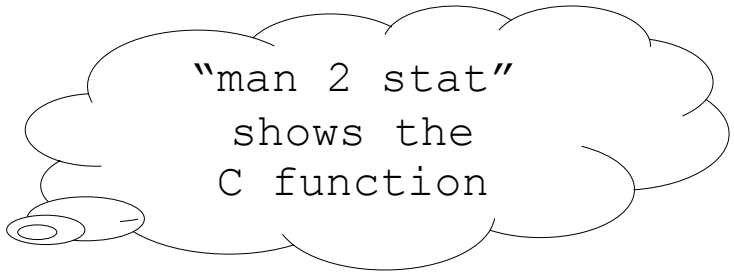

```
 % ls -l /
drwxr-xr-x    2 root   root   4096 Nov  8 17:56 bin/
drwxr-xr-x    2 root   root   4096 Aug 10 14:46 cdrom/
drwxrwsr-x    2 root   staff  4096 Feb  8  2002 home/
drwxr-xr-x    6 root   root   4096 Sep  2 15:26 lib/
lrwx------    1 root   root      6 Sep  2 15:32 u -> /cdf/u/
```

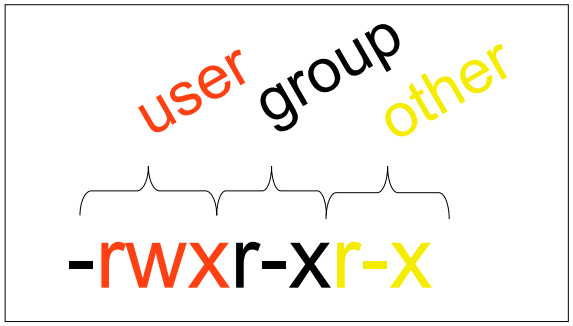
displays file status

Stat

stat -x display info more verbosely

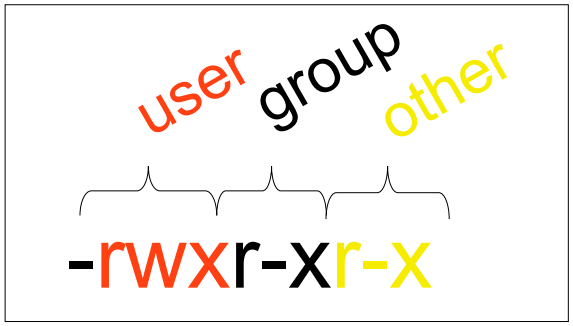
```
reid@wolf:/u$ stat /u/csc209h
  File: '/u/csc209h' -> '/h/u2/csc209h'
  Size: 13          Blocks: 0          IO
 Block: 4096   symbolic link
Device: 10301h/66305d   Inode: 959153
 Links: 1
Access: (0777/lrwxrwxrwx)  Uid: (    0/
 root)   Gid: (  517/ csc209h)
Access: 2017-01-10 19:59:16.041449720 -0500
Modify: 2016-09-08 15:37:43.513111065 -0400
Change: 2016-09-08 15:37:43.513111065 -0400
```





Permissions

- File permissions
  - read, write, execute – pretty much what you think
- Directory permissions
  - read – you can run ls on the directory
  - write – you can create and delete files in the directory
  - execute – you can “pass through” the directory when searching subdirectories.



chmod

1	1	1	1	1	0	1	1	0

- chmod 755 <filename>
  - 3 numbers between 0 and 7, the octal value for that category of user
  - Quiz — what is the command to set the permissions of the file classlist to be world readable but writeable only by the file owner and members of the group.
- Another approach
  - chmod u+rwx
  - chmod go-x
  - adds or removes permissions for those categories of users

Globbing

- A little like regular expressions but different syntax
- * matches any number of any character
- ? matches any one character
- [list of characters]
- [1-5] or [a-z] or [a-xz]