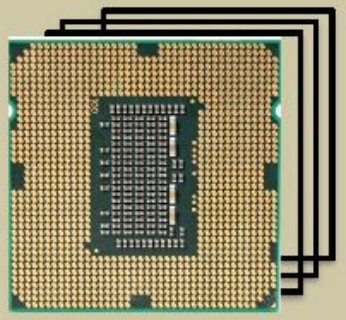
Computer Architecture

Central Processing Unit (CPU)



Random Access Memory (RAM)

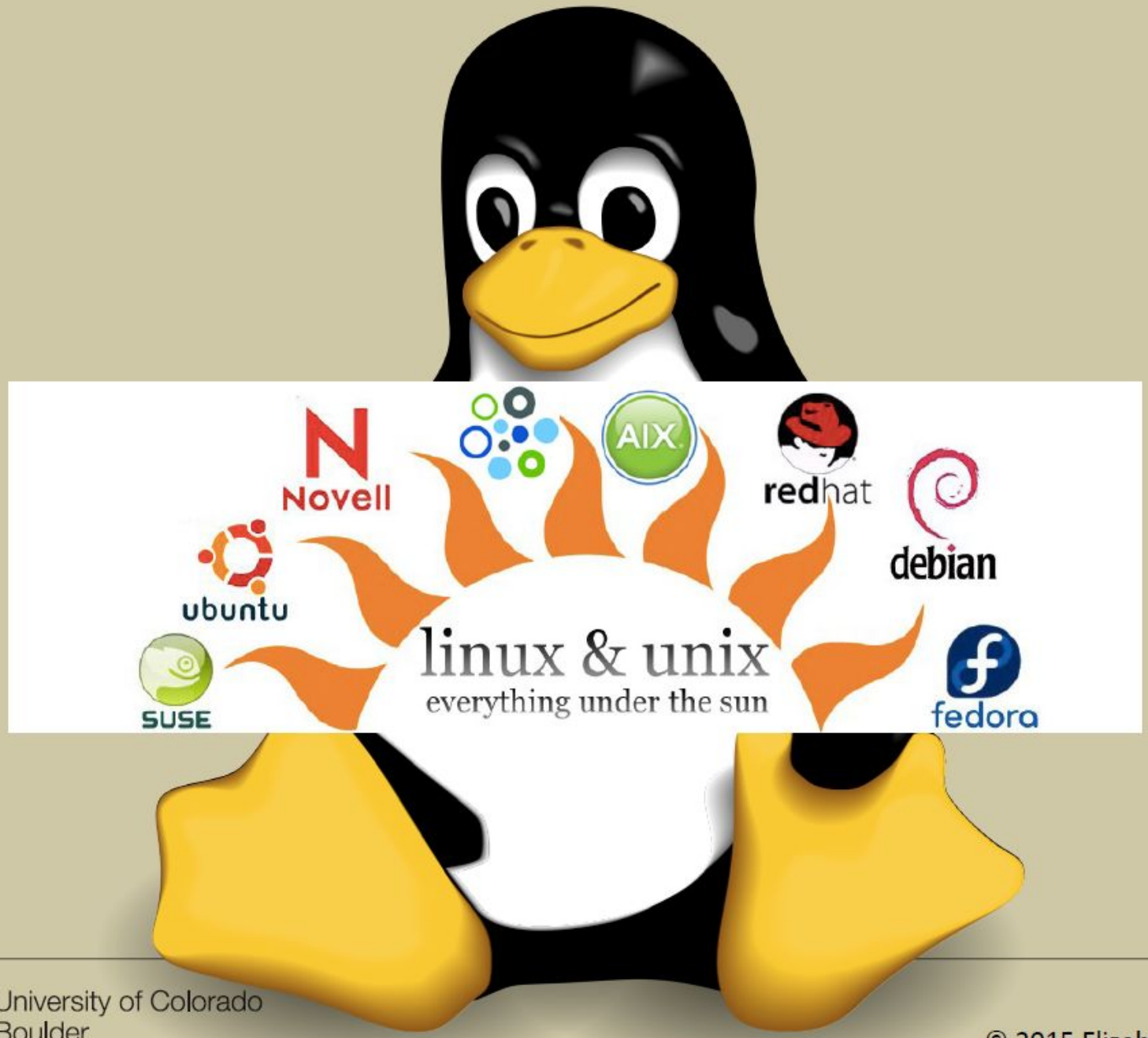
Input/Output Devices

Disk Storage

What is an Operating System?

- Hardware components are only able to control themselves (CPU, RAM, DISK)
- Components do not know how to interact with each other.
- How do we get components to operate together?

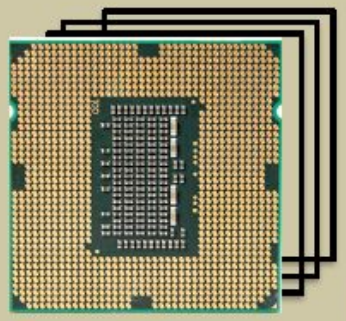
Add an Operating System



© 2015 Elizabeth S. Boese

Computer Architecture

Central Processing Unit (CPU)



Random Access Memory (RAM)

Insert **OS** Here

Input/Output Devices

Disk Storage

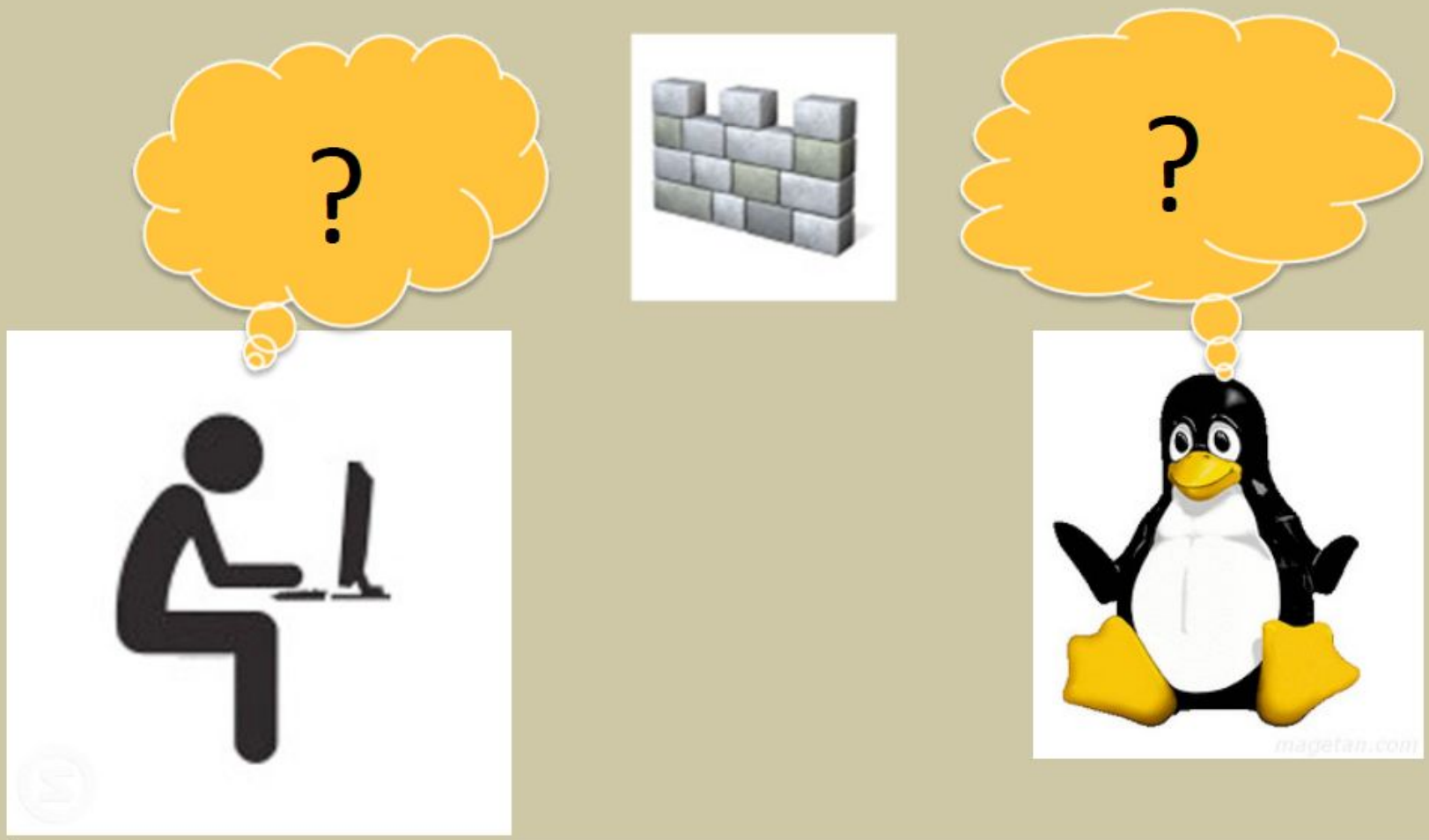
Is it Unix or is it Linux?

- Unix is a trademarked word
- The operating system originally designed in the late 60's was reborn again as open source
- Linux contains many of the original features and quirks of the Unix operating system

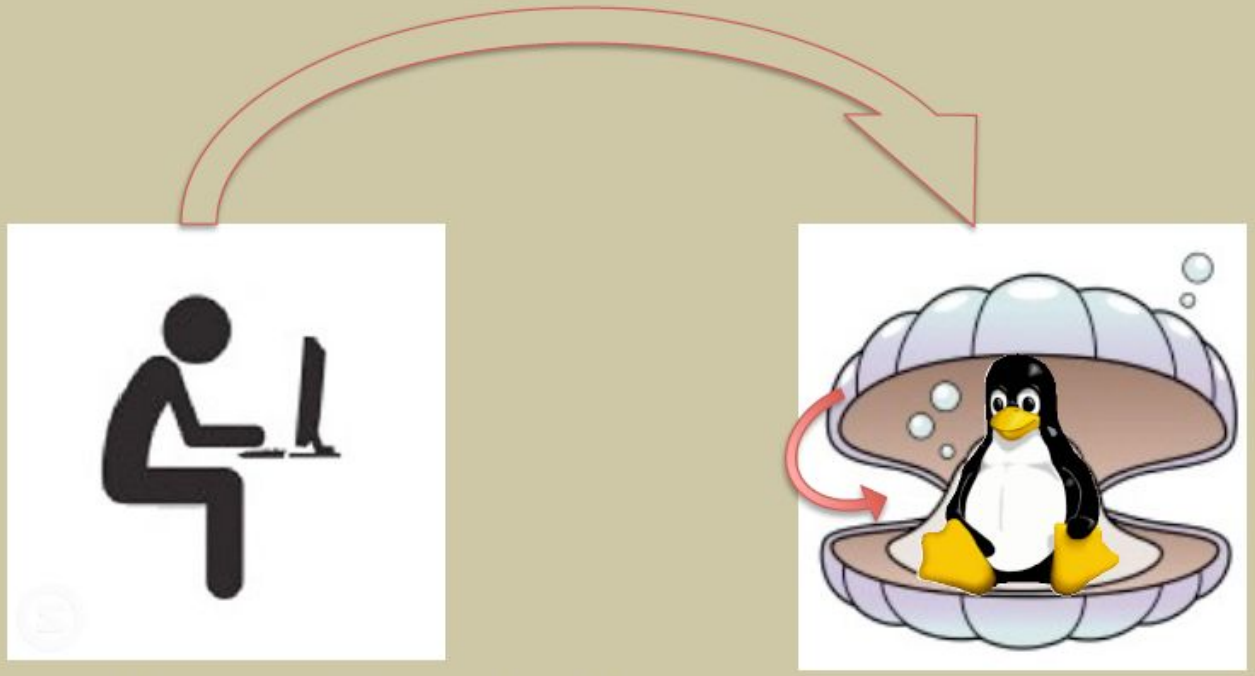
UNIX®

Linux

How can a user communicate with the Operating System?

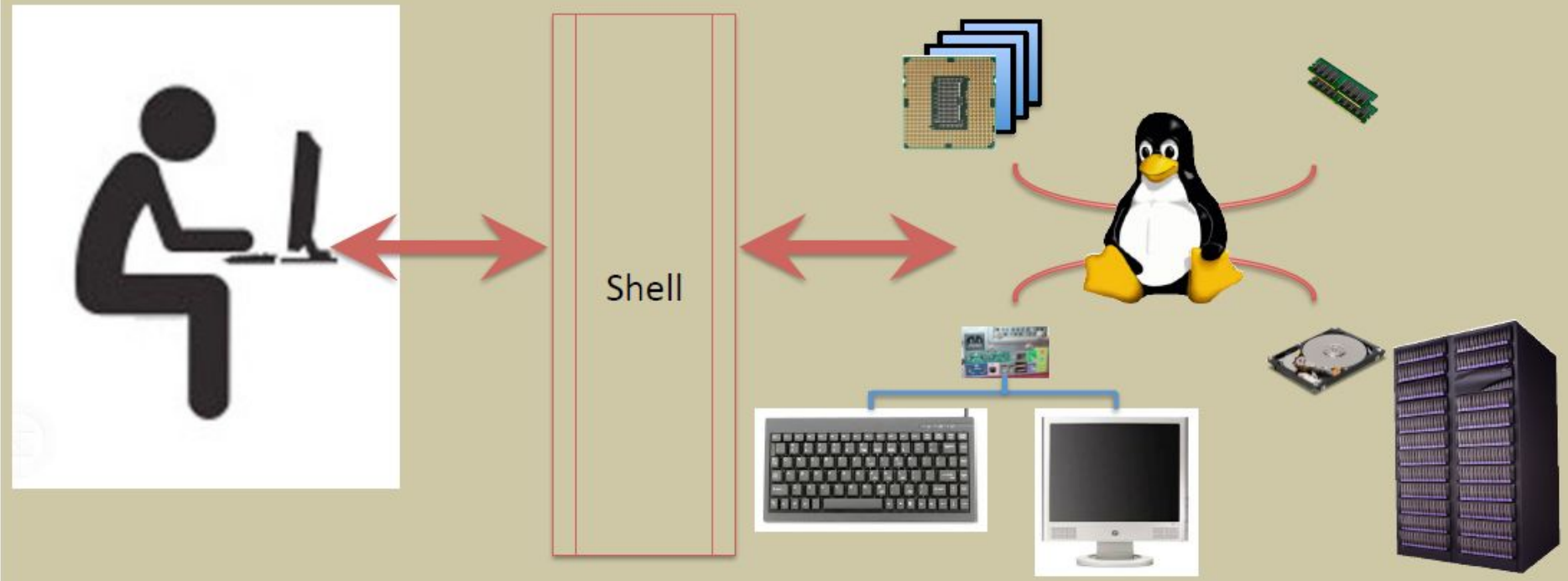


How can a user communicate with the Operating System?



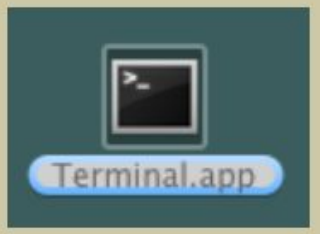
- User talks to the shell
- The shell interprets the commands and talks to the operating system

We can communicate with the Operating System by using the Shell



Using the Terminal Window

- When you select the terminal icon from your desktop, you create an interface with a Shell
- Every keystroke is sent to the shell, every character sent back is displayed in the window



Format of Commands

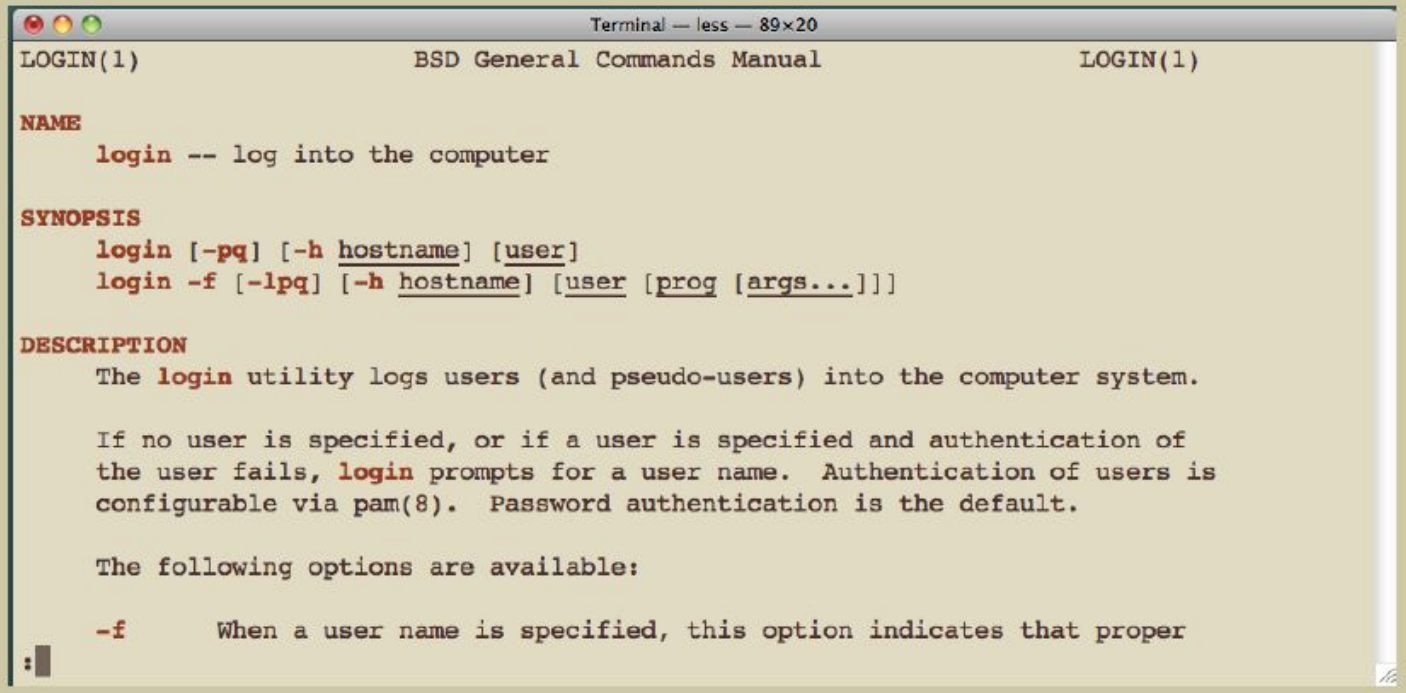
<cmd name> <options> <parameters>

- Most commands are programs
- Options tell the program how we want to process the data
- Parameters are the data the command will process

Getting help with Commands

- Most programs will understand the "--help" option and print information about the programs usage
- The shell can also lookup the usage by using the "man" command. To get help on connecting to a server,

  Type: **man login**



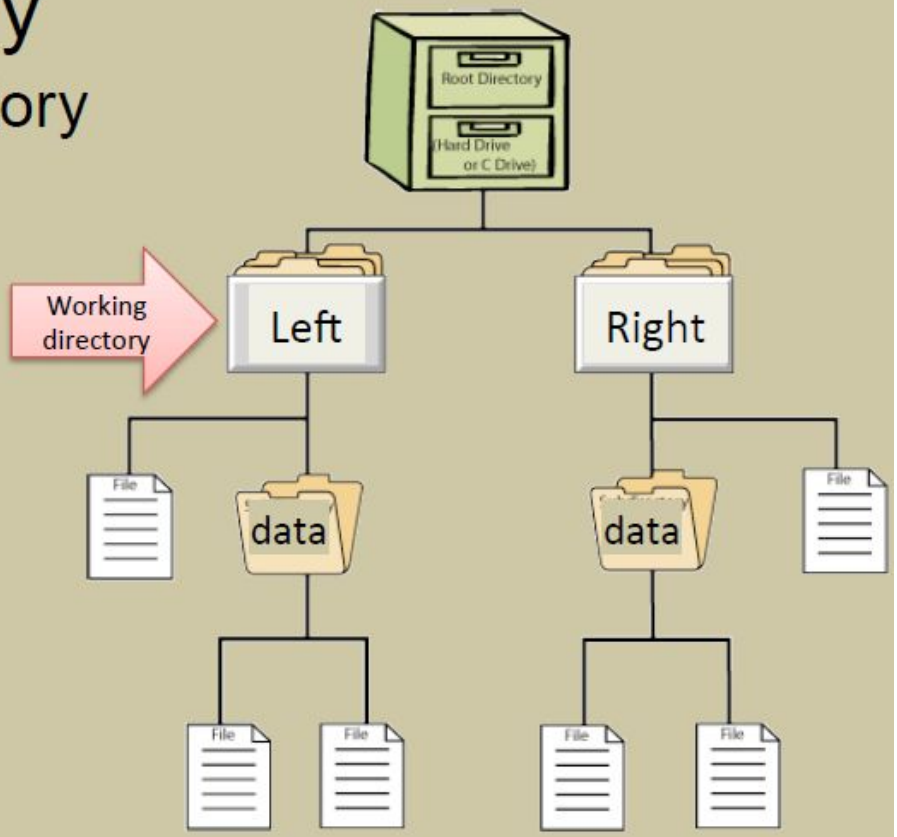
"Everything is a file"

- In Unix, everything looks like a file:
  - documents stored on disk
  - directories
  - inter-process communication
  - network connections
  - devices (printers, graphics cards, interactive terminals, …)
  - programs
  - scripts
- They are accessed in a uniform way:
  - consistent API (e.g., read, write, open, close, …)
  - consistent naming scheme (e.g., /home/debray, /dev/cdrom)

© 2015 Elizabeth S. Boese

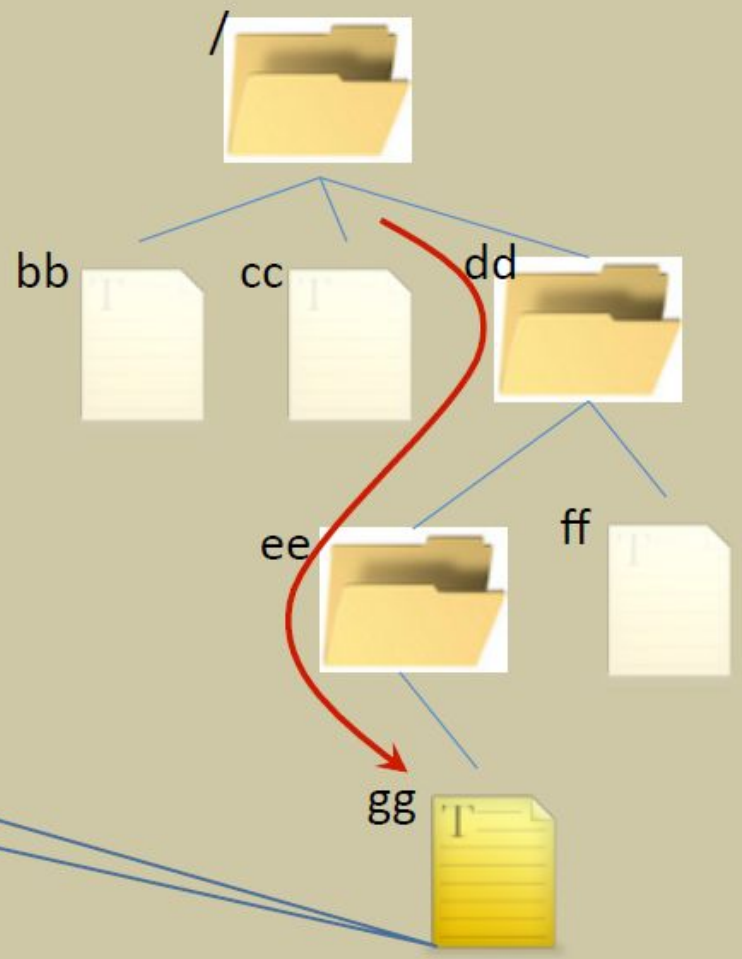
Navigating the File System

- Current working directory
  - pwd – print working directory
  - cd – change directory
  - Relative path vs full path
- File System commands:
  - ls – list directory contents
  - cp – copy files
  - rm – remove files
  - mv – move files
  - mkdir – make directory
  - rmdir – remove directory



Referring to files: Absolute Paths

- *Absolute path*
  - list the directories on the path from the root ("/")
  - separated by "/"



What is the absolute path of gg?

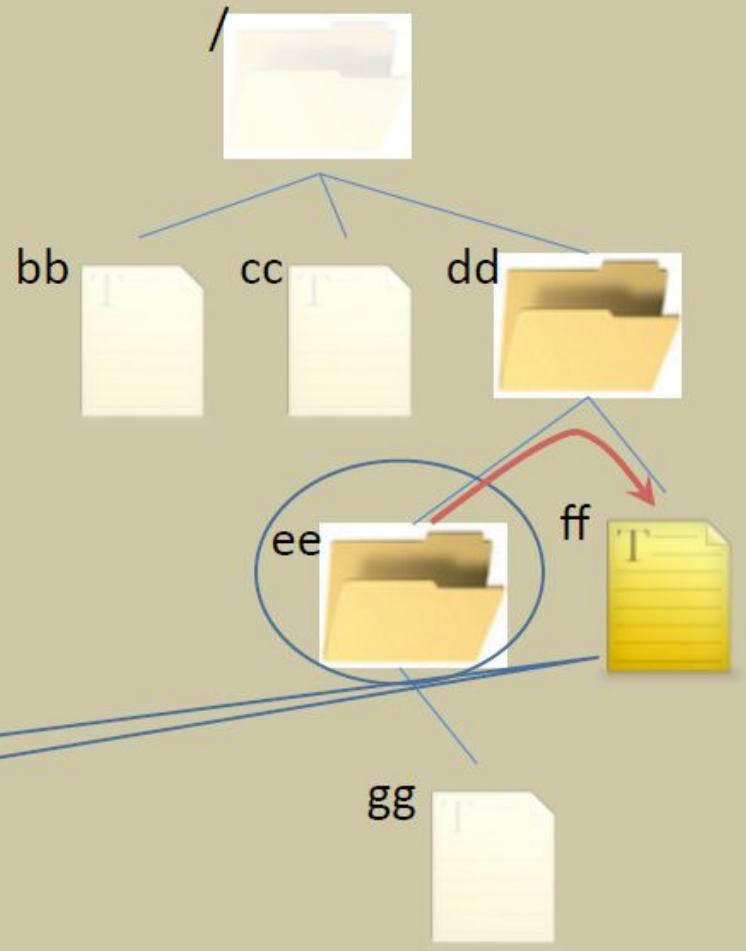
© 2015 Elizabeth S. Boese

Referring to files: Relative Paths

- *Current directory*
- *Relative path*

..

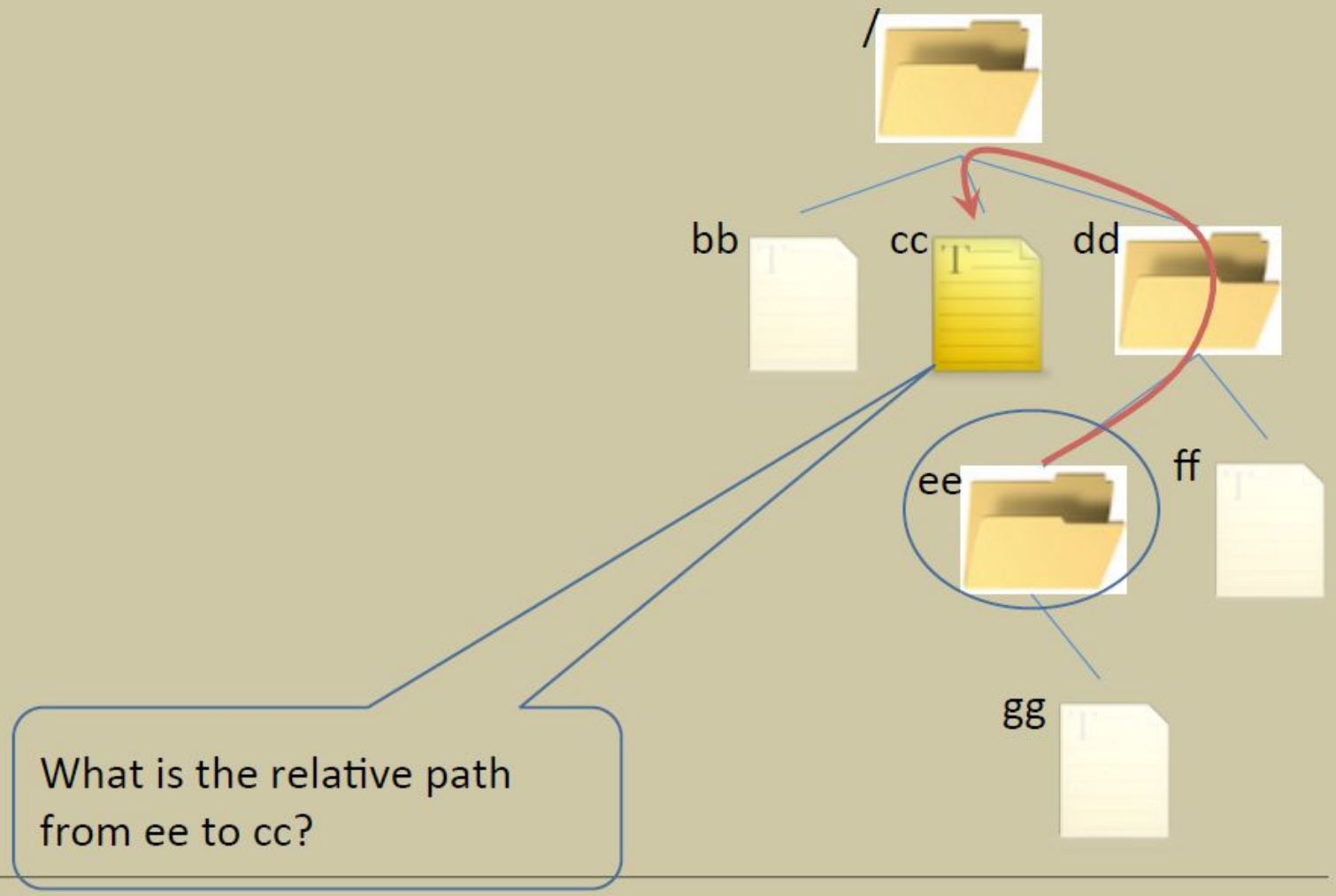
.



If you are in the ee directory, what is the relative path to ff?

© 2015 Elizabeth S. Boese

Referring to files: Relative Paths



© 2015 Elizabeth S. Boese

Commands for Processing Files

- cat – copy the contents to the screen
- more (less) – display file contents in user friendly manner
- head – copy the first lines of a file to the screen
- tail – copy the last lines of a file to the screen
- wc – count the number of lines, words, characters in a file
- grep - globally search a regular expression and print

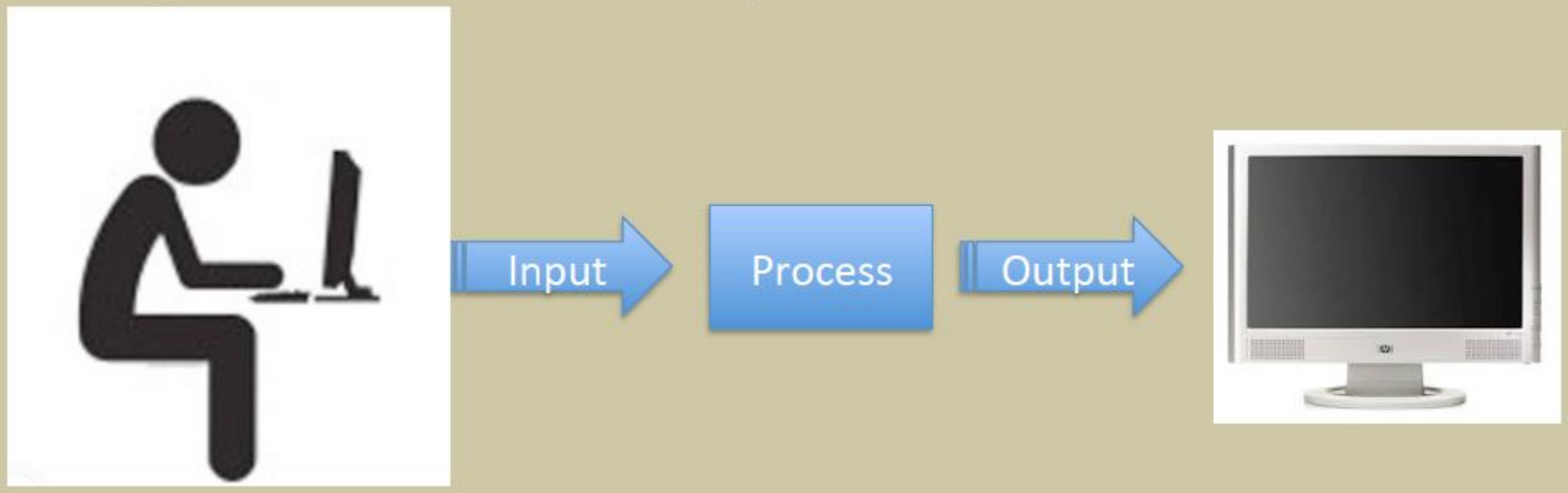
Commands for checking the System Status

- who – list of current users
- whoami – what is my user name
- top – list the processes using the most resources
- ps – process status
- uptime – how long has the system been running
- date – current date and time

How can a user communicate with the Operating System?

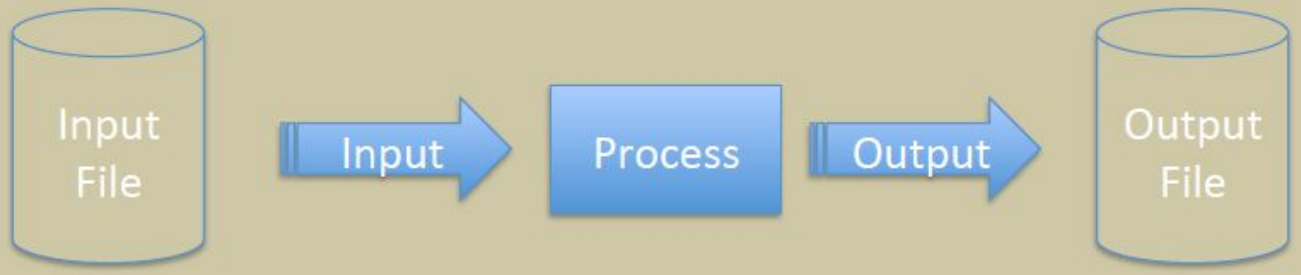
- Shell maintains information
  - PATH : used for finding programs/scripts to be executed
  - HOME : full path to users home directory
  - …

Input and Output for a Process



- Most programs can accept input from the keyboard
- Most programs produce output
- Normally the output is given to the shell to be displayed on the screen

Redirecting Input and Output



- We can tell the shell to pass the characters from a file to a process as input
- We can ask the shell to place the output into a file instead of displaying it on the screen

Redirecting Output

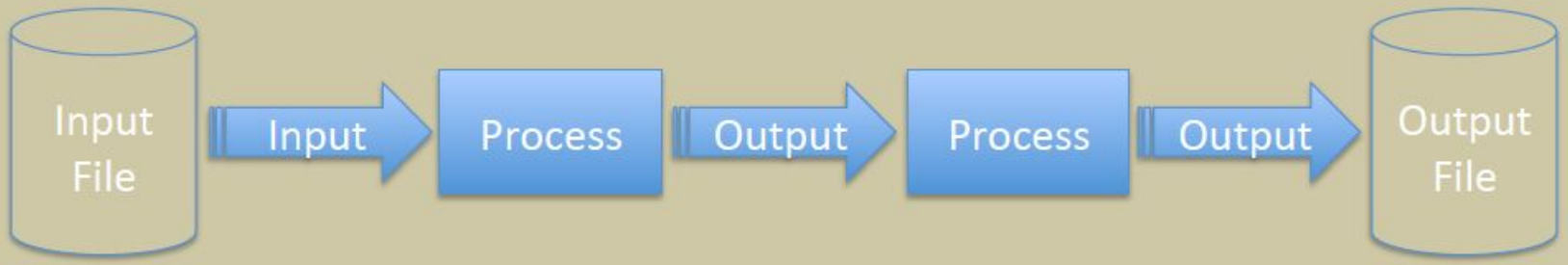
- Shell can redirect the output from any command into a file

history > my_history.out

- Name assigned for default input: stdin
- Name assigned to default output: stdout
- Some programs write output to one file and error information to another file: stderr

history >1 my_history.out >2my_history.err

Piping output of one process as input to another process



- Piping can connect multiple processes into a single command
- "|" is used to indicate the connection

```
ps -ax | wc
```

This command will print the process information to *stdout*. The shell will then take that output and use it as *stdin* to the *wc* command. The output from *wc* is not redirected or piped to another process, therefore it will appear on the screen.

Useful Commands in Piping

- wc - word, line, character, and byte count
- grep - output lines matching a pattern
- sort - sort lines of text files
- uniq - filter out repeated lines in a file
- cut - cut out selected portions of each line of a file
- tee – sends the data to both to a file and *stdout*

File completion

From command line,

- Type part of name of a command or filename, press <tab> and will auto-complete the name for you.

- Arrow keys to go back through history of commands you typed in.

© 2015 Elizabeth S. Boese

history

Display a history of recently used commands

- `history`
- `history 10`
- `history -r 10`
- `!!`
- `!n`
  - repeat command **n** in the history where n is a #
- `!-1`
- `!-2`
- `!ca`

© 2015 Elizabeth S. Boese

Some other useful commands

- **wc**

```
wc file
wc –l file
wc –w file
```

- **grep** *pattern* [*file*]

```
grep public *java
grep include controller.cpp
grep TODO src/*
ls | grep –i main
```


© 2015 Elizabeth S. Boese

diff

Compare two files

- `diff file1 file2`
- `sdiff file1 file2`

© 2015 Elizabeth S. Boese

sort

- sort data.txt
- sort –n data.txt *(on numerical data file)*
  Why is the –n parameter necessary?
- Sort by column
  sort –k2 data.txt
  sort –k2,3 data.txt
- sort –u data.txt

© 2015 Elizabeth S. Boese

find

- Find a file in a directory tree.
  (period means to start in the current directory)
- `find . -name filename -print`

© 2015 Elizabeth S. Boese