

# GNU/Linux 101

Casey McLaughlin

Research Computing Center  
Spring Workshop Series 2018



rccworkshop

IC;3df4mu

# bash-2.1~# man workshop

Linux101

RCC Workshop

L101

## OBJECTIVES

- Operating system concepts
- Linux concepts
- Linux CLI
- Basic systems administration
- HPC, Spear, other RCC resources

```
bash-2.1~# whoami
```

```
# cat ~/casey.yml
```

```
-----
```

```
name: Casey McLaughlin
```

```
- 13+ years at FSU
```

```
- Support, SysAdmin, Web Dev
```

```
- Long-time Linux Nerd
```

# Basic Commands

```
$ whoami
```

```
$ hostname
```

```
$ date
```

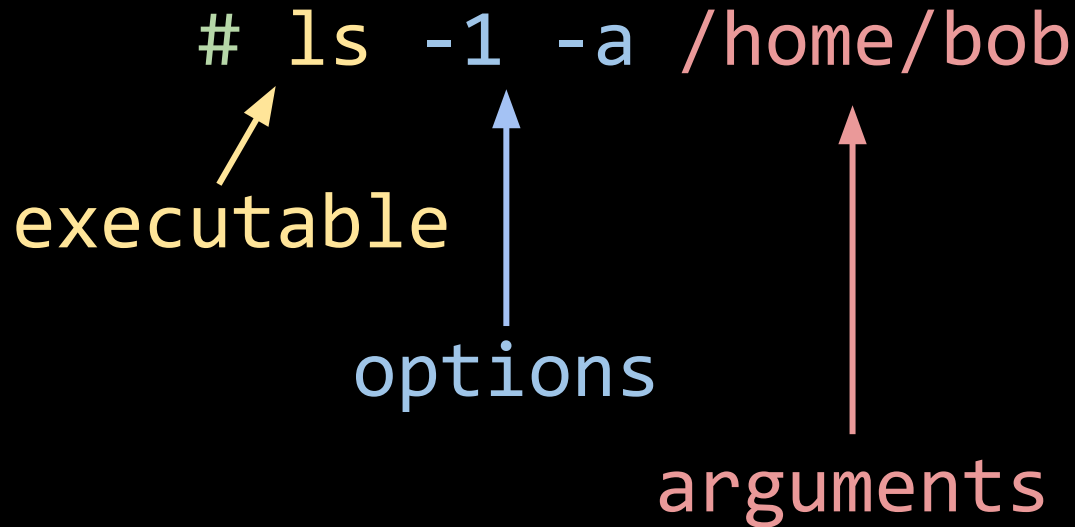
```
$ cal
```

```
$ df
```

```
$ free
```

```
$ exit
```

# Anatomy of a Command



# Example Arguments

```
$ cal -j
```

```
$ date -R
```

```
$ df -h
```

```
$ df -h --total
```

```
$ ls -l /etc
```

```
$ ls -lA /etc
```

# Getting Help With Commands

\$ `whatis`

\$ `man`

\$ `info`

\$ `type`

\$ `which`

# Command Tidbits

```
$ clear
```

*Up Arrow*

*TAB*

```
$ history
```

```
$ history | grep [command]
```

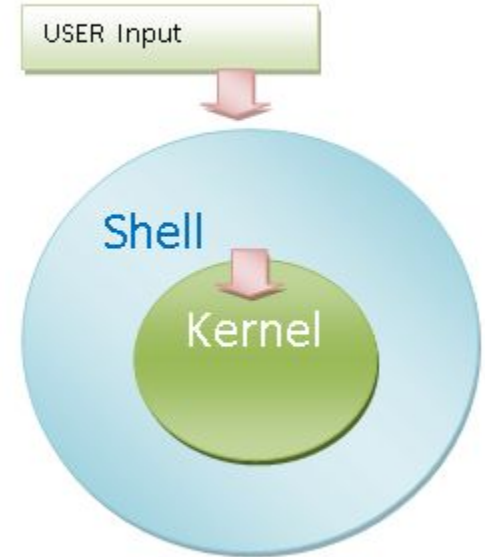
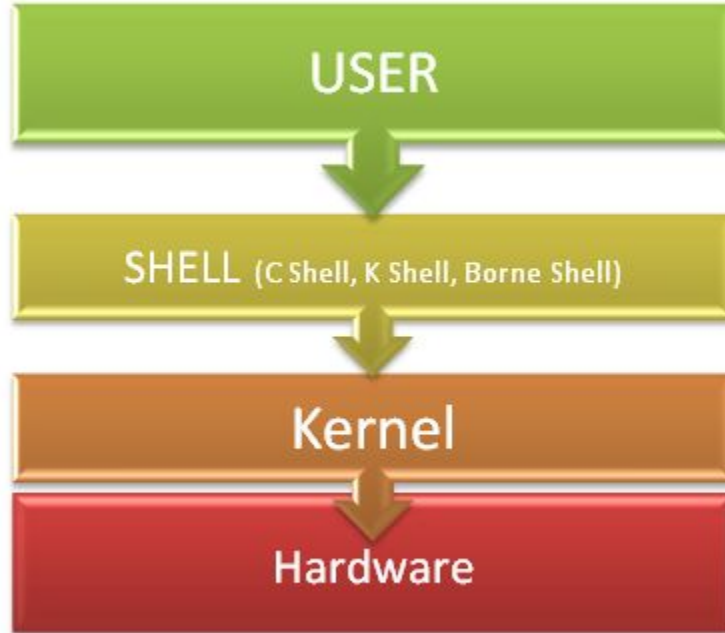
```
$ !!
```

```
$ !number
```

```
$ !?string
```



# What is a Shell?



# Linux is Multi-User!

[CTRL+ALT+F2] ← Escape the GUI

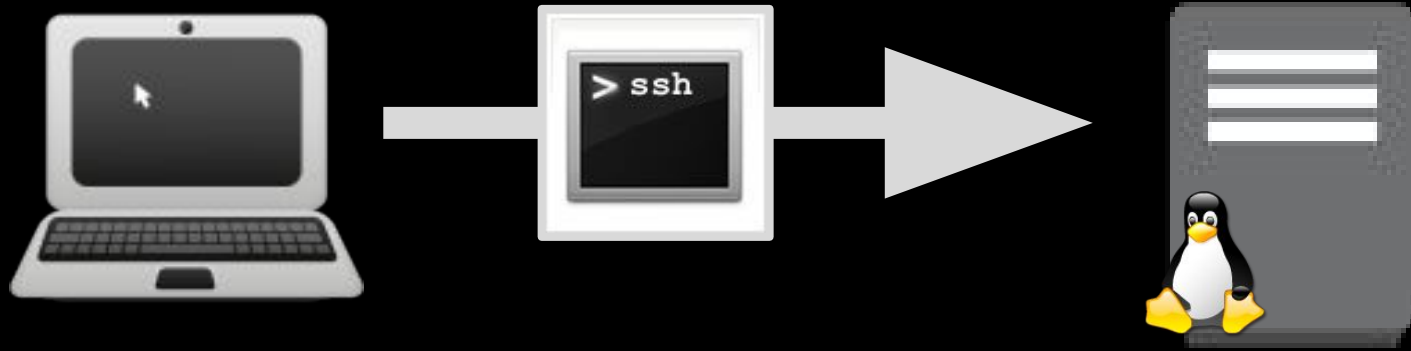
F1 - F6 are TTYs

[CTRL+ALT+F1] ← Return to the GUI

# Connecting Remotely

128.186.1.32

144.174.87.24



```
$ ssh [USER]@hpc-login.rcc.fsu.edu  
PW: linuxRocks#1
```

# Connecting Remotely

```
$ ip addr show
```

```
$ ifconfig
```

```
$ ping
```

```
$ hostname
```

# Linux vs BSD vs tons of others

POSIX: "Portable Operating System Interface"

IEEE spec for maintaining compatibility between OSes

POSIX systems have similar shells.

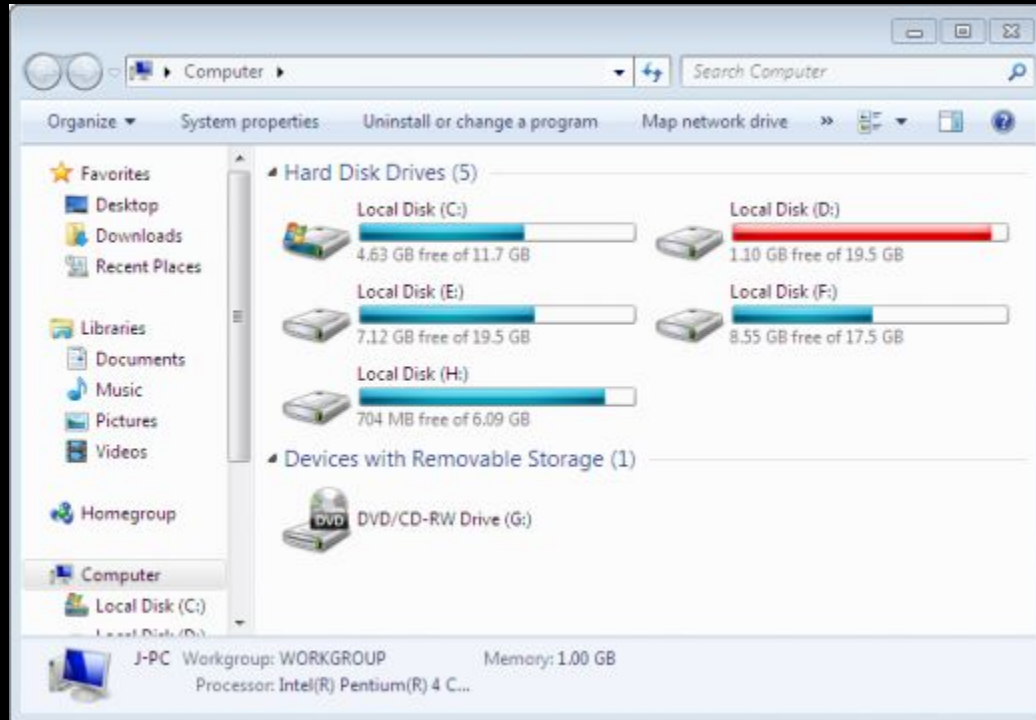
# Linux Distributions

<http://futurist.se/gldt/wp-content/uploads/12.09/gldt1209.png>

# Windows Re-Education




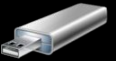



# Windows Disk Drives





# Windows Disk Drives

- C:  First Hard Drive
- D:  Second Hard Drive
- E:  CD-ROM Drive
- G:  USB Drive
- I:  Network Mapped Drive

# Linux File System



/

← Primary mount



/mnt/myserver

← Network mount



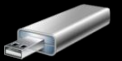
/media/cdrom

← CD-ROM mount



/foobar

← Second HDD



/my/happy/usb

← USB Drive

# Linux Filesystem

```
$ cd
```

```
$ pwd
```

```
$ ls
```

Relative vs absolute paths

‘..’ syntax

# Linux Filesystem Directories

/	Filesys Root	/opt	Miscellaneous
/bin	Binaries	/proc	Internal Stuff
/boot	Kernel	/root	Root user home
/dev	Devices	/sbin	Sys Binaries
/etc	Config	/srv	Services
/home	User Dirs	/tmp	Temp Directory
/lib	Core Libs	/usr	User Programs
/media	CDROM, USB	/var	Variable libs*
/mnt	Other mounts	/lost+found	Trash

# Some Important Things

`~` = My Home Directory

`clear` = Clear the terminal

`echo` = Print something on the screen

`[TAB]` = Auto-complete commands

# The Environment

```
$ printenv
```

```
$ echo $PATH
```

```
$ echo $PWD
```

```
$ sh
```

# `/bin` and `/usr/bin` (and `*`)

```
$ ls /bin
```

```
$ ls /usr/bin
```

```
$ ls /usr/bin/c*
```

```
$ ls /usr/bin/c?9
```

```
$ ls /usr/bin/c?9*
```

# Manipulating Files

```
$ file
```

```
$ touch
```

```
$ echo "... " > file.txt
```

```
$ cat
```

```
$ more
```

```
$ less
```



# Manipulating Files

```
$ mkdir
```

```
$ cp
```

```
$ mv
```

```
$ rmdir
```

```
$ rm (-r)
```

# Searching for Files

```
$ locate [filename]
```

```
$ find [location] --name [filename]
```

*Ex:*

```
$ locate passwd
```

```
$ find /etc --name passwd
```

# Text Editors

```
$ nano somenewfile.txt
```

```
$ vi somenewfile.txt
```

# Getting files from the Internet

```
$ whatis wget
```

```
$ type wget
```

```
$ cd ~      (go home)
```

```
$ wget http://bit.ly/twain4
```

```
$ wget -O tw.txt http://bit.ly/twain4
```

# Chaining Commands

```
$ ls /etc && ls ~ && ls /
```

```
$ whatis grep && type grep
```

# grep

```
$ grep philosopher tw.txt
```

```
$ grep workshop /etc/passwd
```

```
$ grep -n philosopher tw.txt
```

# Piping and Redirection

Default Behavior:

```
$ some command → Terminal Screen
```

Piping (substitute | for →):

```
$ some-cmd → another-cmd → etc → Terminal
```

```
$ some-cmd → another-cmd >> a-file
```

# Piping and Redirection

```
$ ls -l | wc -l
```

```
$ cat tw.txt | wc -l
```

```
$ cat /etc/passwd | wc -l
```

```
$ cat /etc/passwd | sort
```

```
$ cat /etc/passwd | sort | less
```



# Piping and Redirection

```
$ wget -O cr.txt http://bit.ly/ctries4
```

```
$ cat cr.txt
```

```
$ cat cr.txt | wc -l
```

```
$ cat cr.txt | sort
```

```
$ cat cr.txt | sort > cr-sorted.txt
```

```
$ less cr-sorted.txt
```

# Piping to *grep* is the shizzle

```
$ cat cr.txt | grep -n "Luxembourg"
```

```
$ history | grep "wget"
```

```
$ history | grep "wget" > ~/wgcmds.txt
```

```
$ man ls | grep recursive
```

```
$ man grep | grep case
```

```
$ ls -l /bin/c* | grep ch | wc -l
```

# Manipulating Data from the Net

```
$ curl http://bit.ly/twain4
```

```
$ man curl | grep location
```

```
$ curl -sL http://bit.ly/twain4
```

```
$ curl -sL http://bit.ly/twain4 | wc -l
```

```
$ curl -sL http://bit.ly/twain4 > twain.txt
```

```
$ curl -sL http://bit.ly/twain4 | grep Literary
```

# Piping Summary

- | Pipe output to another command
- > Pipe output to file (overwrite)
- >> Pipe output to file (append)
- < Pipe file contents to command

```
$ echo -e "\nNarnia" >> cr.txt && \  
sort cr.txt > cr.txt
```

# Users and Permissions

Users can be in multiple groups.

Root user is special.

# Linux File Permissions

```
# ls -l /bin
```

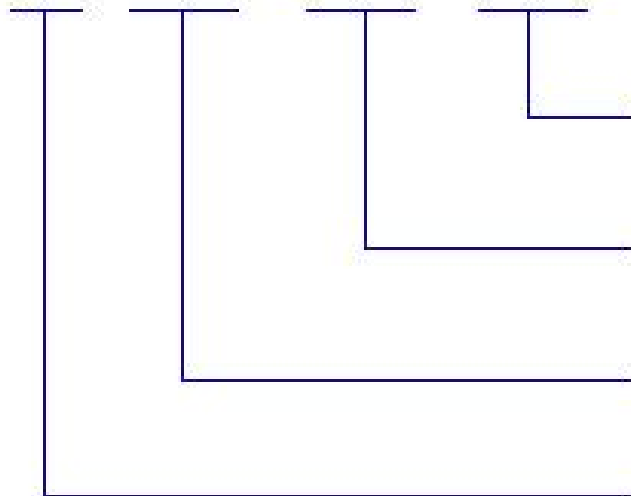
```
-rwxr-xr-x root root 1779 Jan 10 05:05 uname
```

```
# ls -l /etc/host.conf
```

```
-rw-r--r-- root root 1779 Jan 10 05:05 uname
```

# Linux File Permissions

- rwx rw- r--



Read, write and execute permissions for all other users

Read, write and execute permissions for members of the group owning the file

Read, write and execute permissions for the owner of the file

File type: "-" means a file.  
"d" means a directory.

# Octal File Permissions

0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwX



# User File Commands

\$ id *or* \$ whoami

\$ groups

\$ file [somefile]

\$ chmod [perms] [file]

\$ chown [user] [file]

\$ passwd

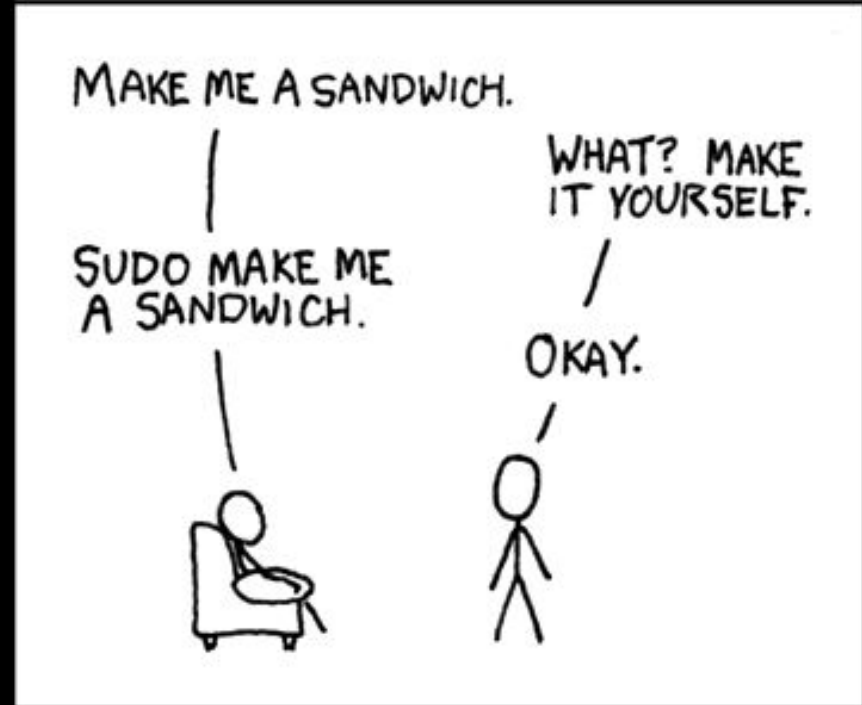
\$ su

# SUDO: Running Commands as root

```
$ sudo [command]
```

```
$ sudo su
```

```
# whoami
```



# Copying files across a network

```
# scp user@[server]:/path/to/file /localpath
```

# BASH Scripts

```
#!/bin/bash
```

```
HNAME=`hostname`
```

```
UNAME=`whoami`
```

```
echo -e "Hello World\nI am running on $HNAME"
```

```
echo -e "\nMy name is $UNAME\n\n"
```

# Compiling Software

```
# whatis gcc
```

```
# type gcc
```

# Compiling Software

```
# wget http://bit.ly/primescprog
```

```
# gcc -o gimmenums primescprog
```

```
# mv primescprog primes.c
```

```
# gcc -o gimmenums primes.c
```

# Compiling Software Part Duex

```
# wget -O d.tgz http://bit.ly/gnudiction
# tar -zvf d.tgz
# ls .
# cd diction-1.11
# ./configure
# make
```

# Submitting Jobs to MOAB

Test Connection and then Move Code to HPC

```
# ssh [user]@hpc-login.rcc.fsu.edu  
# exit  
# scp ~/primes.c [user]@submit.hpc.fsu.edu
```

Login to HPC and compile it

```
# ssh [user]@hpc-login.rcc.fsu.edu  
# gcc -o primes.sh primes.c
```



# Submit Job to HPC Queue

```
$ nano ~/submit.sh
#!/bin/bash
#SBATCH --job-name="myjob"
#SBATCH -n 1
#SBATCH -p quicktest
#SBATCH -t 00:00:05
primes.sh
$ sbatch submit.sh
```

# Hidden Files

Anything beginning in a dot (“.”) in Linux\*\* is hidden.

Why hide files?

```
# ls -a
```

# Aliases

```
# alias lm="ls -l | less"
```

```
# alias countfi="ls -l|wc -l"
```

```
# unalias lm
```

# `.bash_aliases`

```
# nano ~/.bash_aliases
```

```
lm="ls -l | less"
```

```
countfi="ls -l | wc -l"
```

```
# source ~/.bash_aliases
```

# CRON Jobs

```
# * * * * * command to execute
```

```
# T T T T T
```

```
# | | | | |
```

```
# | | | | |
```

```
# | | | | _____ day of week (0 - 6) (0 to 6 are Sun to Sat, or use names;
```

```
# | | | | 7 is Sunday, the same as 0)
```

```
# | | | _____ month (1 - 12)
```

```
# | | _____ day of month (1 - 31)
```

```
# | _____ hour (0 - 23)
```

```
# _____ min (0 - 59)
```

# CRON Jobs

```
# crontab -e
```

```
# Runs every minute
```

```
01 * * * * echo "Hello from Cron"
```

```
# Runs every day at 1:01am
```

```
01 01 * * * /usr/bin/runbackup
```

```
# Runs on May 4 every year @ 4:30pm
```

```
30 16 04 05 * echo "Happy Birthday, Casey"
```

# System Administration

- Process Management (what is running?)
- Installing Software (package management)
- User Management
- Configuring Services

<<< END;

These Slides: <http://bit.ly/1j27VUu>

Book: <http://linuxcommand.org/tlcl.php>



@fsurcc @caseyamcl



FSU Research Computing Center



<https://rcc.fsu.edu>