

Introduction to Linux

Bruce Loftis

bruce.loftis@colorado.edu

Monday 09 January 2017

Agenda

- ❑ Goal – provide a quick overview – then examples ...
- ❑ What is Linux
- ❑ Why use Linux
- ❑ What happens when you log in
- ❑ Shells and environment
- ❑ Commands
- ❑ Filesystem basics
- ❑ Processes
- ❑ More . . .

What Is Linux?

- ❑ Part of the Unix family of operating systems
- ❑ Started in early '90's by Linus Torvalds
- ❑ Open source
- ❑ Several distributions are available – enterprise-quality like RHEL and SUSE – also Ubuntu and others for individual users

What Use Linux?

- ❑ Default operating system on almost all HPC systems
- ❑ Extremely flexible
- ❑ Tries to not get in your way
- ❑ Fast and powerful
- ❑ Has powerful tools for software development
- ❑ You can get started with just a few command lines

How Do You Log in?

- ❑ To a remote system – use Secure Shell (ssh)
- ❑ From Windows – GUI app such as PuTTY
- ❑ From Mac – window from terminal app

What Happens at Login?

- ❑ Login is authenticated (password or other key)
- ❑ Assigned a tty – controlling terminal
- ❑ Shell starts
- ❑ Environment is set up
- ❑ You see a prompt – waiting for a command

Shell

- ❑ The shell reads and interprets commands, runs other executables, returns response
- ❑ Bourne (*sh*) – early
- ❑ Bourne-again (*bash*) – default in Linux with many user-friendly features
- ❑ C (*csh*) – has C-like syntax
- ❑ T (*tcsh*) – extended version of *csh*
- ❑ Korn (*ksh*) – early extension of *sh*

Shell Features

- ❑ Tab completion
- ❑ History
- ❑ Command-line editing
- ❑ Scripting and programming
- ❑ Built-in utilities

Anatomy of a Linux Command

- ❑ *command [flags] [arguments] target(s)*
- ❑ *tar -c -f archive.tar thisdir*
- ❑ Flags have different meanings in different commands
- ❑ Case matters
- ❑ Order of flags may be important

Important Linux Command

❑ *man* – output manual info

- *man* <command>
- *man* -k <keyword>
- *man tar*

File and Directory Commands

- ❑ *pwd* – prints full path to current directory
- ❑ *cd* – change directory – using full or relative path as target
- ❑ *mkdir* – make a new directory
- ❑ *rmdir* – remove an empty directory
- ❑ *cp* – copies a file
- ❑ *mv* – moves (renames) a file
- ❑ *ls* – lists content of directory
- ❑ *ls -l* – gives detailed listing
- ❑ *chmod/chown* – *df* – *du* – ...

Create Our Directory and Files

```
$ cd
```

```
$ tar xf intro-linux.tar
```

```
# set prompt
IT-RC-L-006:~ 1> PS1='$ '
$ export PS1
$
```

```
# where am I ?
$ cd
$ pwd
/Users/bruce
$ cd linux
$ pwd
/Users/bruce/linux
```

```
# list files in current directory
$ ls
action-items.txt      critters/             flowers/
```

\$ *man ls*

LS(1)

BSD General Commands Manual

LS(1)

NAME

ls -- list directory contents

SYNOPSIS

ls [-ABCFGHLOPRSTUW@abcdefghijklmnopqrstuvwxyz1] [file ...]

DESCRIPTION

o o o

The following options are available:

-@ Display extended attribute keys and sizes in long (**-l**) output.

-1 (The numeric digit "one".) Force output to be one entry per line. This is the default when output is not to a terminal.

o o o

Linux Filesystem

- ❑ Files are arranged on disk
- ❑ Directories (folders) can contain files or other directories
- ❑ Levels in full paths are separated by (forward) slashes
`/Users/bruce/linux/action-items.txt`
- ❑ Case-sensitive
- ❑ Spaces in names are discouraged
- ❑ Initial character not a “-”
- ❑ `., .., ~` are useful shorthand

File Editing Programs

- ❑ ***nano*** – simple; easy to get started; not powerful
- ❑ ***vi*** or ***vim*** – universal; powerful but some learning curve required
- ❑ ***emacs*** – keyboard and GUI versions; useful extension for programmers; big learning curve required
- ❑ ...
- ❑ <http://xkcd.com/378/>


```
$ whoami    # username
```

```
bruce
```

```
$
```

```
$ pwd
```

```
/Users/bruce/linux
```

```
$ ls
```

```
action-items.txt          critters/                  flowers/
```

```
$ # rename
```

```
$ mv action-items.txt action
```

```
$ ls
```

```
action                    critters/                  flowers/
```

```
$ cp action action-2017
```

```
$ ls
```

```
action                    action-2017                critters/                  flowers/
```

```
$ # Root Directory
```

```
$ cd /
```

```
$ pwd
```

```
/
```

```
$ ls
```

```
Applications/
```

```
Library/
```

```
Network/
```

```
System/
```

```
Users/
```

```
net/
```

```
cores/
```

```
dev/
```

```
etc/
```

```
home/
```

```
var/
```

```
bin/
```

```
private/
```

```
sbin/
```

```
tmp/
```

```
usr/
```

```
Volumes/
```

```
opt/
```

```
$ # system commands
```

```
$ cd /bin
```

```
$ ls
```

date	expr	ln	pwd	stty	zsh
bash	dd	hostname	ls	rcp	sync
cat	df	kill	mkdir	rm	tcsh
chmod	domainname	ksh	mv	rmdir	test
cp	echo	launchctl	pax	sh	unlink
cs	ed	link	ps	sleep	wait4path

```
$ # more system commands at /sbin /usr/bin /usr/sbin
```

```
$ cd /usr/bin
```

```
$ ls
```

```
...
```

```
$ # locate a command
```

```
$ whereis sleep
```

```
/bin/sleep
```

```
$ which sleep # just locates a command in user's path
```

```
/bin/sleep
```

```
$ # directory shortcuts
```

```
$ cd
```

```
$ pwd
```

```
/Users/bruce
```

```
$ cd linux/critters
```

```
$ pwd
```

```
/Users/bruce/linux/critters
```

```
$ cd .
```

```
$ pwd
```

```
/Users/bruce/linux/critters
```

```
$ cd ../; pwd
```

```
/Users/bruce/linux
```

```
$ cd ~; pwd  
/Users/bruce
```

```
$ cd ~bruce; pwd  
/Users/bruce
```

```
$ # prompt
```

```
$ PS1='$ '; export PS1
```

```
$
```

```
$ PS1='\u> ' # username
```

```
bruce>
```

```
$ PS1='\u-\!> ' # history count
```

```
bruce-16>
```

```
$ PS1='\@> '
```

```
11:03 AM>
```

```
$ PS1='\W> ' # current directory
```

```
linux>
```

```
$ PS1='\w> ' # complete current working directory
```

```
~/linux>
```

```
$ pwd
/Users/bruce/linux
```

```
$ # detailed list of files
```

```
$ ls -l          # letter - l=>long
```

```
total 8
```

```
-rw-r--r--  1 bruce  staff   54 Dec 28 11:20 action-items.txt
drwxr-xr-x  6 bruce  staff  204 Dec 28 10:26 critters/
drwxr-xr-x  3 bruce  staff  102 Dec 28 10:27 flowers/
```

```
$ # list one item per line
```

```
$ ls -l          # numeral one
```

```
action-items.txt
critters/
flowers/
```

```
$ pwd
```

```
/Users/bruce/linux
```

```
$ ls -l # letter
```

```
total 8
```

```
-rw-r--r--  1 bruce  staff   54 Dec 28 11:20 action-items  
drwxr-xr-x  6 bruce  staff  204 Dec 28 10:26 critters  
drwxr-xr-x  3 bruce  staff  102 Dec 28 10:27 flowers
```

```
$ ls critters
```

```
aardvark  armadillo  ewe        zebra
```

```
$ ls -a critters
```

```
.          ..          aardvark  armadillo  ewe        zebra
```

- ❑ Other *dot* files may exist – “-a” flag is needed to list them

❑ Where do we find executables?

```
$ echo $PATH
```

```
/Users/bruce/bin:./usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/opt/vdt/globus:/opt/openssl/bin:/opt/graphics/bin:/Library/PostgreSQL/8.4/bin:/Users/bruce/anaconda/bin:/Applications:/Applications/VisIt.app/Contents/Resources/bin
```

❑ Executables are searched for in the left-to-right order of the PATH variable

- First is my own *bin* directory
- Second is the current directory
- Then system *bin* directories
- Then others
- Can be configured as preferred

- ❑ Create an executable

```
$ cd ~/bin
```

- ❑ Use nano or other editor to create a file name “ll” (letter-L) with contents “ls -l”

```
$ # this will work
```

```
$ echo "ls -l" > ll
```

```
$ # need to update PATH (will be OK for next login)
```

```
$ rehash
```

- ❑ And we need to set execute permissions

❑ Permissions

```
$ pwd
```

```
/Users/bruce/bin
```

```
$ ls -l ll
```

```
-rw-r--r--  1 bruce  staff   6 Dec 28 14:03 ll
```

```
$ chmod +x ll
```

```
$ ls -l ll
```

```
-rwxr-xr-x  1 bruce  staff   6 Dec 28 14:03 ll
```

- ❑ Now we have execute permission set for user, group, other
- ❑ User => rwx => read, write, execute permission
- ❑ Other => r-x => can read and execute but not write

```
$ chmod g-x ll
```

```
$ chmod o-wx ll
```

```
$ pwd
```

```
/Users/bruce/bin
```

```
$ ls -l ll
```

```
-rwxr-xr-x  1 bruce  staff   6 Dec 28 14:03 ll*
```

```
$ # ~ (tilde) refers to home directory
```

```
$ cd ~/linux
```

```
$ ll
```

```
total 8
```

```
-rw-r--r--  1 bruce  staff   54 Dec 28 11:20 action-items.txt
```

```
drwxr-xr-x  6 bruce  staff  204 Dec 28 10:26 critters
```

```
drwxr-xr-x  3 bruce  staff  102 Dec 28 10:27 flowers
```

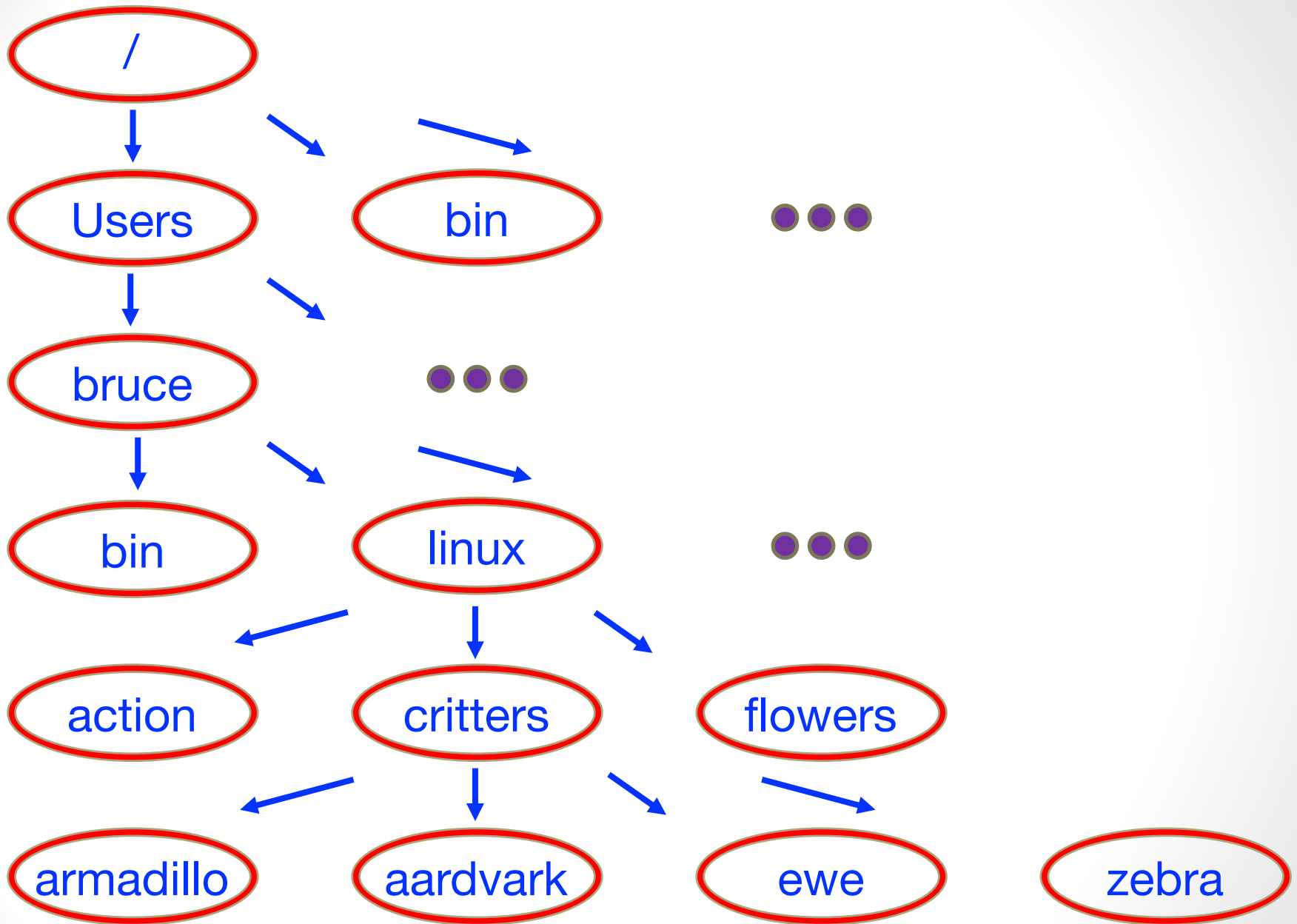
```
$ # cd - (minus) take you back
```

```
$ pwd; cd /usr/bin; pwd; cd -; pwd
```

```
/Users/bruce/linux
```

```
/usr/bin
```

```
/Users/bruce/linux
```



Shell and Environment Variables

- ❑ Shell variables – only effective in the current shell
- ❑ Environment variables – also active in future commands and shells
- ❑ Set default values at login in file *.bash_profile*
- ❑ *var_name[=value]* (shell)
- ❑ *export var_name[=value]* (environment)
- ❑ *env* (shows current variables)

Useful Variables

- ❑ ***PATH*** – directories to search for commands
- ❑ ***HOME*** – home directory
- ❑ ***SHELL*** – location of shell being used
- ❑ ***PWD*** – current working directory
- ❑ ***USER*** – username
- ❑ ***PS1***
- ❑ ...

❑ Environment Variables

```
$ env / sort
```

```
GROUP=staff
```

```
HOME=/Users/bruce
```

```
HOST=OIT-RC-L-006
```

```
HOSTTYPE=unknownLOGNAME=bruce
```

```
MACHTYPE=x86_64
```

```
OSTYPE=darwin
```

```
PATH=:::/Users/bruce/bin:~/anaconda/bin: ...
```

```
PS1=$
```

```
PWD=/Users/bruce
```

```
TERM=vt100
```

```
TERM_SESSION_ID=6FAADAA9-9F3A-4D8C-8471-64A3C2349D72
```

```
TMPDIR=/var/folders/l0/q31cnvhn36q4j1b02hxjwm6h0000gn/T/
```

```
USER=bruce
```

```
VENDOR=apple
```

```
_=/usr/bin/env
```

```
today=28dec16
```


❑ Remove files

```
$ cd ~/linux/critters; ls
```

```
aardvark      armadillo     ewe           zebra
```

```
$ # touch creates a file with no content – sometimes useful
```

```
$ touch buffalo; ls
```

```
aardvark      armadillo     buffalo       ewe           zebra
```

```
$ rm buffalo; ls
```

```
aardvark      armadillo     ewe           zebra
```

❑ Files do not move to a trash folder – they are gone

❑ Create and remove directories

```
$ mkdir dinosaur; ls -F
```

```
aardvark      armadillo     dinosaur/     ewe           zebra
```

```
$ rmdir dinosaur; ls -F
```

```
aardvark      armadillo     ewe           zebra
```

```
$ mkdir dinosaur; $ cd dinosaur
```

```
$ touch brontosaurus ichthyosaur coelophysoidea
```

```
$ cd ..; $ rmdir dinosaur
```

```
rmdir: dinosaur/: Directory not empty
```

```
$ # options to remove a not-empty directory
```

```
$ # cd dinosaur; rm * --or-- rm -ri dinosaur
```

❑ Word Count

```
$ cd ~/linux/critters
```

```
$ wc -l *
```

```
12 aardvark
 2 armadillo
 1 ewe
 1 zebra
16 total
```

```
$ wc *
```

```
12      103      641 aardvark
 2       14      103 armadillo
 1        2       13 ewe
 1        3       16 zebra
16      122      773 total
```

```
# try "wc -l" forgetting the filename(s)
```

```
# it waits for file names – hangs
```

```
# type cntl-C to escape
```

❑ Redirecting output

```
$ cd ~/linux/critters
$ wc -l * > lines-per-critter
$ cat lines-per-critter
    13 aardvark
     1 armadillo
     0 ewe
     0 zebra
    14 total

$ wc -l a* > lines
$ echo '###' >> lines
$ wc -l a* >> lines
$ cat lines
    13 aardvark
     1 armadillo
    14 total
'###'
    13 aardvark
     1 armadillo
    14 total
```

❑ Redirecting output

```
$ cd ~/linux/critters
$ wc -l * > lines-per-critter
$ cat lines-per-critter
 12 aardvark
   2 armadillo
   1 ewe
   1 zebra
 16 total
```

```
$ wc -l a* > lines
$ echo "###" >> lines
$ wc -l a* >> lines
$ cat lines
    12 aardvark
     2 armadillo
    14 total

###

    12 aardvark
     2 armadillo
    14 total
```

- ❑ Pipes – output from one command is used as input to the next command

```
$ env | sort | head -4
```

```
EXINIT=set showmatch number ignorecase
```

```
GROUP=staff
```

```
HOME=/Users/bruce
```

```
HOST=OIT-RC-L-006
```

```
$ env | wc -l
```

```
32
```

```
$ cat a* | wc -l
```

```
$ ls a* | wc -l
```

❑ Loops

```
$ echo $PS1 $PS2
```

```
$ >
```

```
$ for f in a*
```

```
> do
```

```
>     head -1 $f
```

```
> done
```

The aardvark is a medium-sized, burrowing, nocturnal mammal

The pink fairy armadillo (*Chlamyphorus truncatus*) or

```
$ # or ...
```

```
$ for f in a* ; do head -1 $f ; done
```

❑ History

```
$ history | tail -3
```

```
53  cd ~/linux
```

```
54  cd critters
```

```
55  ls a*
```

```
$ !55
```

```
ls a*
```

```
aardvark
```

```
armadillo
```


❑ Simple Scripting

❑ Use nano or other editor or ...

```
$ echo wc \"$1 > count
```

```
$ chmod u+x count
```

```
$ cat count
```

```
wc $1
```

```
$ count aardvark
```

```
12 119 726 aardvark
```

```
$ # just in case filename has a space in it
```

```
$ echo wc \"$1\" > count
```

❑ Finding things

```
$ # grep => global-regular-expression-print (huh?)
```

```
$ grep the a*
```

```
aardvark:mammal native to Africa.[2] It is the only living  
aardvark:species of the order Tubulidentata,[3][4] although  
aardvark:other prehistoric species and genera of Tubulidentata  
aardvark:are known. Unlike other insectivores, it has a long  
aardvark:roams over most of the southern two-thirds of the  
aardvark:termites, which it will dig out of their hills using  
armadillo:pichiciego is the smallest species of armadillo
```

```
$ grep -n nocturnal a*
```

```
aardvark:1:The aardvark is a medium-sized, burrowing, nocturnal  
aardvark:9:rocky. A nocturnal feeder, it subsists on ants and
```

```
$ man grep; man egrep
```

❑ More finding

```
$ find ~ | grep vark
```

```
$ find ~bruce | grep vark
```

```
/Users/bruce/linux/critters/aardvark
```

```
$ pwd
```

```
/Users/bruce/linux/critters
```

```
$ find .
```

```
.
```

```
./aardvark
```

```
./armadillo
```

```
./ewe
```

```
./zebra
```

```
$ find / | grep -i paraview
```

```
/Users/bruce/training/ParaView/01-intro-to-paraview.pdf
```

```
$ # avoid "permission denied" files
```

```
$ find / | grep -i paraview > ~find-paraview
```

❑ Processes

```
$ # list processes
```

```
$ ps
```

PID	TTY	TIME	CMD
75402	ttys000	0:00.05	-tcsh
96723	ttys000	0:00.04	bash
44404	ttys001	0:00.04	-tcsh
44487	ttys001	0:00.01	bash

```
$ # list all processes
```

```
$ ps -edalf
```

```
$ # kill a process – stop runaway processes
```

```
$ man kill
```

```
$ kill -9 44487
```



bruce.loftis @ colorado . edu