

Introduction to UNIX/Linux

Biochemistry Boot Camp 2023

Session #5

Nicholas Fitzkee

nfitzkee@chemistry.msstate.edu

Some terms

- Operating system (OS)

Command-line interface (CLI)

```
File Edit View Terminal Help
top - 18:18:50 up 3:02, 2 users, load average: 0.79, 0.76, 0.73
Tasks: 141 total, 1 running, 140 sleeping, 0 stopped, 0 zombie
Cpu(s): 13.2%us, 7.9%sy, 0.0%ni, 78.1%id, 0.0%wa, 0.7%hi, 0.0%si, 0.0%st
Mem: 961680k total, 941356k used, 20324k free, 70976k buffers
Swap: 1028088k total, 0k used, 1028088k free, 537960k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU %MEM    TIME+  COMMAND
 809 root        20   0 43872 25m  10m  S 10.6  2.7   5:36.55 Xorg
3356 user01    20   0 38088  11m 8892  S  4.0  1.2   0:00.38 gnome-screensho
1539 user01    20   0 60464  22m 8824  S  2.0  2.4   0:56.74 compiz
1585 user01    20   0 41604  12m 10m   S  1.3  1.4   0:06.18 wnck-applet
3338 user01    20   0 47148  12m 9540  S  1.0  1.3   0:00.32 gnome-terminal
   6 root        20   0   0     0   0   S  0.3  0.0   0:02.56 events/0
 675 messageb  20   0 3148 1500  764  S  0.3  0.2   0:03.29 dbus-daemon
 789 root        20   0   0     0   0   S  0.3  0.0   3:29.56 phy0
1522 user01    20   0 88912 9236 7228  S  0.3  1.0   0:04.67 gnome-settings-
1629 user01    20   0 20368 9912 7964  S  0.3  1.0   0:02.95 gtk-window-deco
3160 user01    20   0 239m  84m  63m  S  0.3  9.0   1:20.94 soffice.bin
3355 user01    20   0 2544 1196  908  R  0.3  0.1   0:00.08 top
   1 root        20   0 2804 1640 1184  S  0.0  0.2   0:00.57 init
   2 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthreadd
   3 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 migration/0
   4 root        20   0   0     0   0   S  0.0  0.0   0:00.00 ksoftirqd/0
   5 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 watchdog/0
```

Graphical user interface (GUI)



Why UNIX?

- **Stability:** Systems can run for months or more
- **Multitasking:** Easy to running many programs at once (used to be very unique)
- **Flexibility:** Graphical environment is optional, can be pared down to bare minimum, optimizing performance

Why UNIX?

- **Science focus:** Lots of scientific software runs exclusively on UNIX
- **Data Storage:** Unix handles lots of files well
- **Historical reasons:** mainframes, early software written on UNIX

Where did Linux come from?

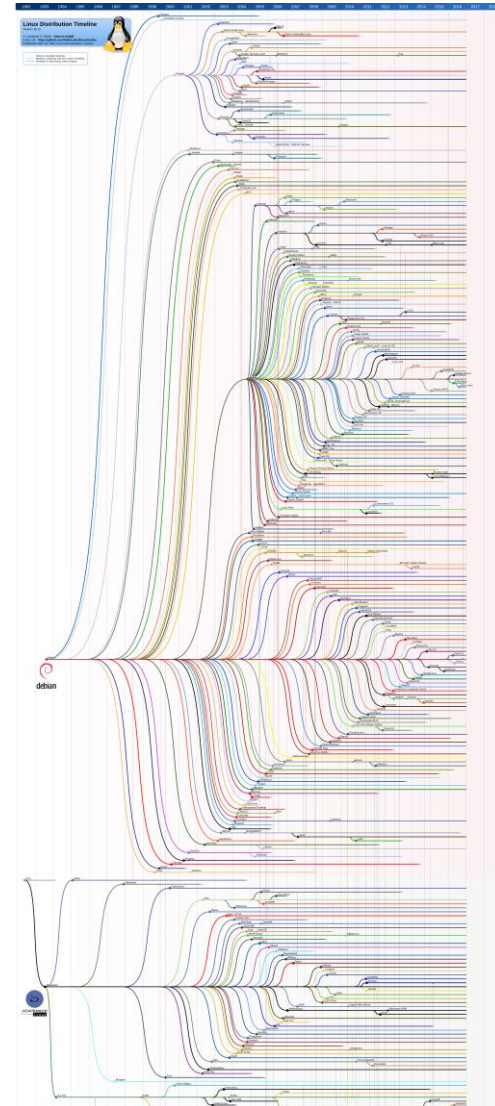
- Linus Torvalds created it
 - with assistance from programmers around the world
 - first posted on Internet in 1991
- Linux 1.0 in 1994; 2.2 in 1999
- Today used on 7-10 million computers
 - with 1000's of programmers working to enhance it

Flavors of Linux

- “Linux” only refers to the kernel: the core program that runs the operating system
- Many programs contribute to the OS experience:
 - Window environment (GUI)
 - Command line interface (CLI)
 - Even simple utilities like the list of printers
- Because Linux is open-source, there are lots of combinations of programs that work around the same kernel
 - These are called ***distributions*** or *distros*

Linux Distributions: History

- Distributions made installing software easier
 - Lots of software needed for a working system
 - Oldest and still active: “Slackware”
- Distributions frequently “forked” as new features added or removed
- Some distros also died out



Common Linux Distributions

- **CentOS:** Very stable at the expense of some newer features & hardware support (discontinued but still ubiquitous)



- **Ubuntu:** User friendly, a good mix of “cutting edge” updates and stability



- **Debian:** Cutting edge, but stable software is available for installation too



A Beginners Guide CHOOSING A LINUX DISTRO

Which one is best for you?

1 Ubuntu <i>User friendly</i>	2 CentOS <i>Reliable, Steady, Secure</i>
 <p>EASE OF USE, REGULAR UPDATES, USE LTS RELEASE FOR STABILITY, EASIER FOR NOVICES, USE AS A DESKTOP OR SERVER</p>	 <p>EMPHASIS ON RELIABILITY & SECURITY, YOU WON'T FIND BLEEDING EDGE SOFTWARE HERE, LINUX VETERANS LOVE IT. USE AS A DESKTOP OR SERVER</p>
4 DEBIAN <i>Stable or bleeding edge</i>	 <p>UNIX NOT LINUX, FAST NETWORKING, MINIMALIST, UNIFIED DEVELOPMENT TEAM, STABLE & SECURE. NO PROPRIETARY DRIVERS. USE AS A SERVER</p>
 <p>CONSERVATIVE, FAST, SECURE, STABLE, BUT BLEEDING EDGE SOFTWARE AVAILABLE IF NEEDED, USE AS A DESKTOP OR SERVER</p>	3 FreeBSD <i>Experience is key</i>
 <p>BASED ON REDHAT & CENTOS, BLEEDING EDGE & LATEST FEATURES, USE AS A DESKTOP OR SERVER</p>	 <p>VIRTUALISATION, CLUSTERING, AVAILABILITY, SPECIFIC DESIGN & PURPOSE, UNIQUE & DIFFERENT, USE AS A SERVER</p>
5 Fedora <i>Bleeding Edge</i>	6 COREOS <i>New Kid on the Block</i>

CAUTION- THERE ARE SOME SIMPLISTIC, GENERAL STATEMENTS ON THIS INFOGRAPHIC. THE POWER OF A LINUX OR UNIX OPERATING SYSTEM IS THE ABILITY TO CUSTOMISE IT TO MATCH YOUR EXACT REQUIREMENTS. ITS INTENTION IS TO GIVE BEGINNERS A GOOD PLACE TO START & A GENERAL OVERVIEW

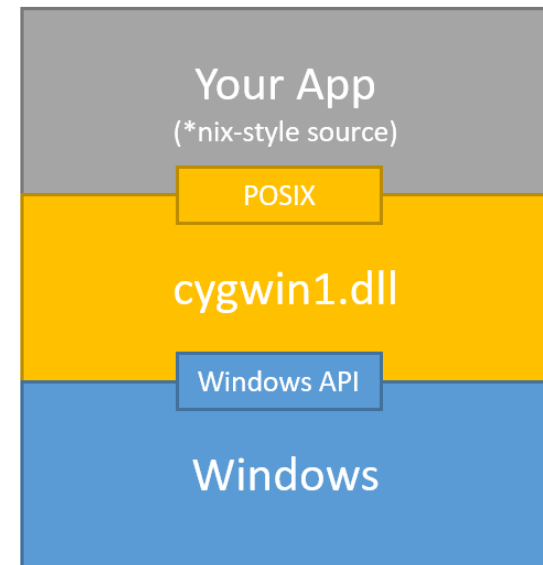
Linux vs. “Linux-like” Environment

- An OS may provide a Linux-like (POSIX) environment without actually running the Linux kernel
- **MacOS:** XNU kernel, built around the “mach” kernel; readily supports Linux-like shell scripting and windowing environment (true POSIX)
 - X-Windows supported via Xquartz (<https://www.xquartz.org/>)
- **Windows 10/11:** Windows kernel, but supports a Linux subsystem to provide Linux support (added in 2016)
 - This session may change in years to come!
<https://arstechnica.com/gadgets/2022/11/windows-subsystem-for-linux-with-gui-apps-launches-for-windows-10/>
 - Alternative to CygWin:
<https://learn.microsoft.com/en-us/windows/wsl/faq>

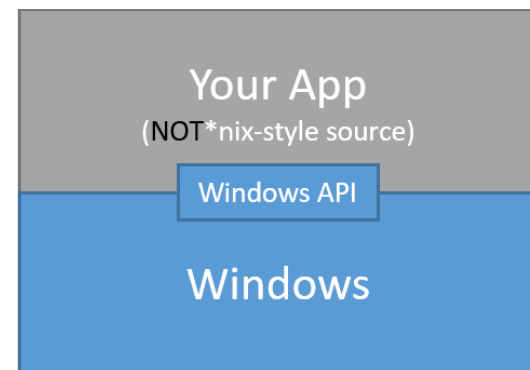
Windows Subsystem for Linux

- I recommend Cygwin for now
- Very little difference to the end user “feel”, and WSL is easier to install
- But Cygwin may be more unfamiliar for scientific software developers

With Cygwin



With MinGW (WSL)

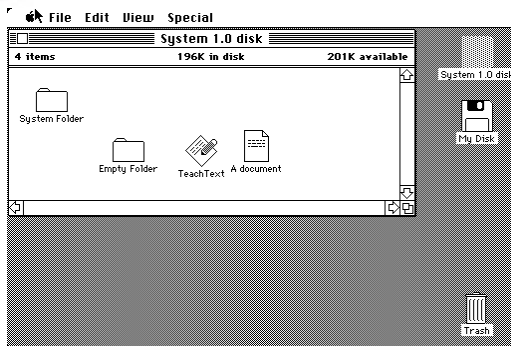


1969



1981
MS-DOS 1.0

1984 (GUI)
Apple System 1.0



1992
Windows 3.1 (GUI)

1991
Linux 0.0.1



2001



2012



2021

Practical Application

This is all very interesting, but...

How do you get to a Linux-like Command Line
and what can you do when you get there?

Mac Users

- Basic command line interface can be found in the “Applications/Utilities” folder – run the program “Terminal”
- We will use XQuartz as well, which can be found at www.xquartz.org
 - Download and install this program while PC owners struggle through tutorial!
- File Transfer Client: **FileZilla (Client)** or **Cyberduck**
<https://filezilla-project.org/download.php?platform=osx>
<https://cyberduck.io/>

PC Users

- Getting to Linux is a bit more difficult
- Follow the instructions in the “Running X11 on Windows” handout
- File Transfer Client: **WinSCP**
<https://winscp.net/eng/download.php>

Connect to a Linux Server via SSH

- Mac Users
 - Make sure XQuartz is running
 - Open a new XTerm window and run SSH
 - **Example:** `ssh -Y jsmith@(hostname with no spaces)`
 - Replace `jsmith` with your username and `host` with the value given in class!
- PC Users
 - Follow directions in X11 Handout using the SSH Secure Shell Program (Quick Connect)

Getting Files To/From the Server

- **PC/Mac:** Open up WinSCP or FileZilla/Cyberduck
- **Old School:** Open up another Xterm, then:
`sftp <username>@(hostname with no spaces)`
 - Need to use the `cd` command to navigate to the right folder, then get to copy the file
 - It will save the files to wherever you started the `sftp` program

Demo: Common Linux Tasks

- Navigating the file system: Where am I? What's here?
- Copying/Manipulating Files and Directories
- Running programs in the background
 - Ampersand (“&”) trick
 - Useful commands: Ctrl-Z, bg, fg
- Running software via X11
 - Text file editing (xemacs)

Try It Yourself: Linux Tutorial

- Very helpful, and covers basic to advanced topics:

<http://www.ee.surrey.ac.uk/Teaching/Unix/>

- Your Linux account will be active for **14 days**, after which it will be disabled
 - Contact Dr. Fitzkee if you need more time

Handy Linux Cheat Sheet:

Common commands you can type from the command line.

Best Practice:

- Work with someone who is knowledgeable, write down the commands they use
- Use “`man <command>`” to get information about the commands they type

File Commands	System Info
ls – directory listing	date – show the current date and time
ls -la – formatted listing with hidden files	cal – show this month's calendar
cd dir – change directory to <i>dir</i>	uptime – show current uptime
cd – change to home directory (e.g. <code>/home/nfitzkee</code>)	w – display who is online
pwd – show current directory	whoami – who you are logged in as
mkdir dir – create a directory <i>dir</i>	finger user – display information about <i>user</i>
rm file – delete <i>file</i>	uname -a – show kernel information
rm -r dir – delete directory <i>dir</i>	cat /proc/cpuinfo – cpu information
rm -f file – force remove <i>file</i>	cat /proc/meminfo – memory information
rm -rf dir – force remove directory <i>dir</i> * (see warning below!)	man command – show the manual for <i>command</i>
cp file1 file2 – copy <i>file1</i> to <i>file2</i>	df – show disk usage
cp -r dir1 dir2 – copy <i>dir1</i> to <i>dir2</i> ; create <i>dir2</i> if it doesn't exist	du – show directory space usage
mv file1 file2 – rename or move <i>file1</i> to <i>file2</i>	free – show memory and swap usage
if <i>file2</i> is an existing directory, moves <i>file1</i> into directory <i>file2</i>	whereis app – show possible locations of <i>app</i>
ln -s file link – create symbolic link <i>link</i> to <i>file</i>	which app – show which <i>app</i> will be run by default
touch file – create or update <i>file</i>	
cat > file – places standard input into <i>file</i>	Compression
more file – output the contents of <i>file</i> (alternatively: less file)	tar cf file.tar files – create a tar named <i>file.tar</i> containing <i>files</i>
head file – output the first 10 lines of <i>file</i>	tar xf file.tar – extract the files from <i>file.tar</i>
tail file – output the last 10 lines of <i>file</i>	tar czf file.tar.gz files – create a tar with Gzip compression
tail -f file – output the contents of <i>file</i> as it grows, starting with the last 10 lines	tar xzf file.tar.gz – extract a tar using Gzip
	zip -r file.zip files – create a Windows-compatible zip archive
Process Management	unzip file.zip – extract zip archive
ps – display your currently active processes	gzip file – compresses <i>file</i> and renames it to <i>file.gz</i>
top – display all running processes	gzip -d file.gz – decompresses <i>file.gz</i> back to <i>file</i>
kill pid – kill process id <i>pid</i>	
killall proc – kill all processes named <i>proc</i> *	Network
bg – lists stopped or background jobs; resume a stopped job in the background	ping host – ping <i>host</i> and output results
fg – brings the most recent job to foreground	whois domain – get whois information for <i>domain</i>
fg n – brings job <i>n</i> to the foreground	dig domain – get DNS information for <i>domain</i>
	dig -x host – reverse lookup <i>host</i>
File Permissions	wget file – download <i>file</i>
chmod octal file – change the permissions of <i>file</i> to <i>octal</i> , which can be found separately for user, group, and world by adding:	wget -c file – continue a stopped download
<ul style="list-style-type: none"> □ 4 – read (r) □ 2 – write (w) □ 1 – execute (x) 	Lab Utilities
Examples:	NMR Viewers: sparky or nmrDraw or nvj or analysis
chmod 777 – read, write, execute for all	pipe2ucsf file.ft2 file.ucsf – convert NMRPipe spectrum to UCSF format
chmod 755 – rwx for owner, rx for group and world	/home/databases/pdb/ – location of all PDB structures
For more options, see man chmod .	PDB Viewers: pymol or molmol or rasmol or vmd
	Text editors: xemacs or emacs or gedit or vi
SSH	All can be invoked with a file, e.g. xemacs file
ssh user@host – connect to <i>host</i> as <i>user</i>	./script – run an executable file <i>script</i> in the current directory
ssh -p port user@host – connect to <i>host</i> on port <i>port</i> as <i>user</i>	Shortcuts
sftp user@host – connect to <i>host</i> as <i>user</i> for file transfer	Ctrl+C – halts the current command
gftp – graphical file transfer client	Ctrl+Z – stops the current command, resume with fg in the foreground or bg in the background
Searching	Ctrl+D – log out of current session, similar to exit
grep pattern files – search for <i>pattern</i> in <i>files</i>	Ctrl+W – erases one word in the current line
grep -r pattern dir – search recursively for <i>pattern</i> in <i>dir</i>	Ctrl+U – erases the whole line
command grep pattern – search for <i>pattern</i> in the output of <i>command</i>	Ctrl+R – type to bring up a recent command
locate file – find all instances of <i>file</i>	!! – repeats the last command
find . -name "pattern" – search for the file named <i>pattern</i> in the current directory (or below); <i>pattern</i> can contain wildcards (e.g. <code>*.txt</code>)	exit – log out of current session
	* use with extreme caution.

File Commands

ls – directory listing

ls -la – formatted listing with hidden files

cd dir - change directory to *dir*

cd – change to home directory (e.g. */home/nfitzkee*)

pwd – show current directory

mkdir dir – create a directory *dir*

rm file – delete *file*

rm -r dir – delete directory *dir*

rm -f file – force remove *file*

rm -rf dir – force remove directory *dir* * (see warning below!)

cp file1 file2 – copy *file1* to *file2*

cp -r dir1 dir2 – copy *dir1* to *dir2*; create *dir2* if it doesn't exist

mv file1 file2 – rename or move *file1* to *file2*

if *file2* is an existing directory, moves *file1* into directory *file2*

ln -s file link – create symbolic link *link* to *file*

touch file – create or update *file*

cat > file – places standard input into *file*

more file – output the contents of *file* (alternatively: **less file**)

head file – output the first 10 lines of *file*

tail file – output the last 10 lines of *file*

tail -f file – output the contents of *file* as it grows, starting with the last 10 lines

Process Management

ps – display your currently active processes

top – display all running processes

kill pid – kill process id *pid*

killall proc – kill all processes named *proc* *

bg – lists stopped or background jobs; resume a stopped job in the background

fg – brings the most recent job to foreground

fg n – brings job *n* to the foreground

File Permissions

chmod octal file – change the permissions of *file* to *octal*, which can be found separately for user, group, and world by adding:

- 4 – read (r)
- 2 – write (w)
- 1 – execute (x)

Examples:

chmod 777 – read, write, execute for all

chmod 755 – rwx for owner, rx for group and world

For more options, see **man chmod**.

SSH

ssh user@host – connect to *host* as *user*

ssh -p port user@host – connect to *host* on port *port* as *user*

sftp user@host – connect to *host* as *user* for file transfer

gftp – graphical file transfer client

Searching

grep pattern files – search for *pattern* in *files*

grep -r pattern dir – search recursively for *pattern* in *dir*

command | grep pattern – search for *pattern* in the output of *command*

locate file – find all instances of *file*

find . -name "pattern" – search for the file named *pattern* in the current directory (or below); *pattern* can contain wildcards (e.g. `***)`)

System Info

date – show the current date and time

cal – show this month's calendar

uptime – show current uptime

w – display who is online

whoami – who you are logged in as

finger user – display information about *user*

uname -a – show kernel information

cat /proc/cpuinfo – cpu information

cat /proc/meminfo – memory information

man command – show the manual for *command*

df – show disk usage

du – show directory space usage

free – show memory and swap usage

whereis app – show possible locations of *app*

which app – show which *app* will be run by default

Compression

tar cf file.tar files – create a tar named *file.tar* containing *files*

tar xf file.tar – extract the files from *file.tar*

tar czf file.tar.gz files – create a tar with Gzip compression

tar xzf file.tar.gz – extract a tar using Gzip

zip -r file.zip files – create a Windows-compatible zip archive

unzip file.zip – extract zip archive

gzip file – compresses *file* and renames it to *file.gz*

gzip -d file.gz – decompresses *file.gz* back to *file*

Network

ping host – ping *host* and output results

whois domain – get whois information for *domain*

dig domain – get DNS information for *domain*

dig -x host – reverse lookup *host*

wget file – download *file*

wget -c file – continue a stopped download

Lab Utilities

NMR Viewers: **sparky** or **nmrDraw** or **nvj** or **analysis**

pipe2ucsf file.ft2 file.ucsf – convert NMRPipe spectrum to UCSF format

/home/databases/pdb/ – location of all PDB structures

PDB Viewers: **pymol** or **molmol** or **rasmol** or **vmd**

Text editors: **xemacs** or **emacs** or **gedit** or **vi**

All can be invoked with a file, e.g. **xemacs file**

./script – run an executable file *script* in the current directory

Shortcuts

Ctrl+C – halts the current command

Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background

Ctrl+D – log out of current session, similar to **exit**

Ctrl+W – erases one word in the current line

Ctrl+U – erases the whole line

Ctrl+R – type to bring up a recent command

!! - repeats the last command

exit – log out of current session

* use with extreme caution.