Crash Course on UNIX and Systems Tools

Day 1 --- The UNIX Environment and The Shell

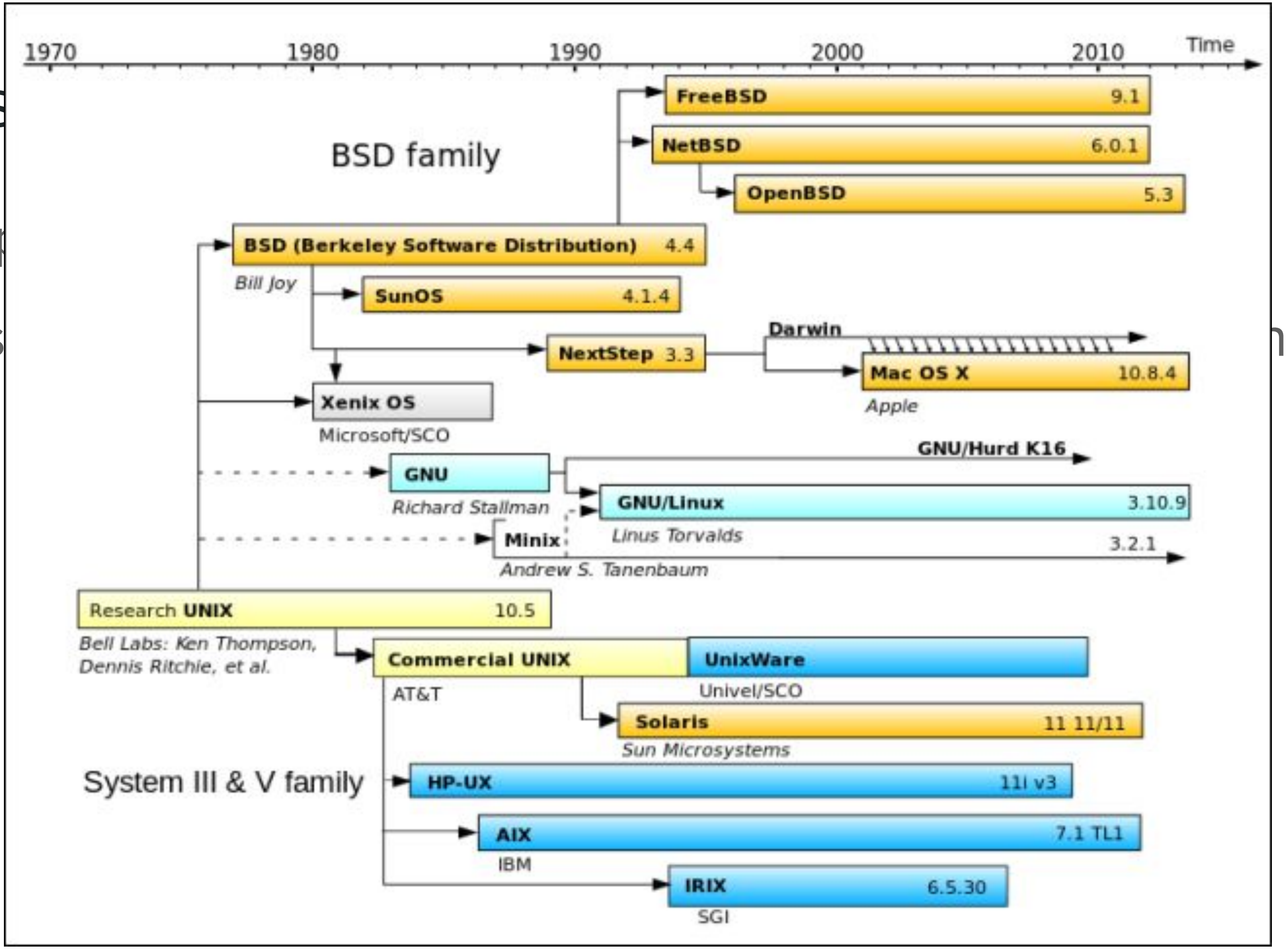
What is UNIX?

- An operating system from the 1960s
- Consists of a kernel, **shell**, and modules/programs to run
- Who cares? UNIX design is *very* influential

What is

- An op
- Cons
- Who

n



What is UNIX?

- An operating system from the 1960s
- Consists of a kernel, **shell**, and modules/programs to run
- Who cares? UNIX design is *very* influential
- *Really* important base for many systems and low-level tools
- 213, 343, many other systems courses

What is the shell?

- The interactive interface between the user and the OS --- you can run programs, run scripts, execute shell commands, write to files/directories, etc.
- Many brands of shells --- we're going to use **bash**

```
$ █
```

Navigation: *Where am I?*

`pwd`

- “Present working directory”
- Outputs your current **absolute path**

```
----------

$ pwd
/root/sandbox
```

Navigation: *Where am I?*

- **Absolute path**: location with respect to the **root directory** "`/`"
- **Relative path**: location with respect to:
  - The **home** directory "**`~/`**"
  - The **current** directory "`./`"
  - The **parent** directory "`../`"
  - Any combination of the above choices, i.e. "`./`**`mydir`**`/../`" etc.

Navigation: *What’s in here?*

`ls`

- “List”
- Outputs all files/directories inside the **current** directory
- Or, specify a path (absolute or relative) to see its contents

```
----------

$ ls
file1  file2
```

Navigation: *What’s in here?*

```
$ ls -a
./   ../   .bash_history   another_file   my_file
```

- Entries that start with “ . ”

Navigation: *What’s in here?*

```
$ ls -ltr
total 1
-rw-r--r-- 1 root root 0 Jan  2 23:07 file1
-rw-r--r-- 1 root root 0 Jan  2 23:07 file2
```

- **`-ltr`** --- Show details of entries in chronological order
  - **`-l`** lists in “**long**” format
  - **`-t`** sort entries by **time**
  - **`-r`** **reverse** the order

Navigation: *What’s in here?*

```
$ ls -ltr
total 1
-rw-r--r-- 1 root root 0 Jan  2 23:07 file1
-rw-r--r-- 1 root root 0 Jan  2 23:07 file2
```

- What does all of this info mean? **man-pages!**

```
$ man ls
```

- Spacebar to move down, “q” to quit

Navigation: *Where can I go?*

`cd`

- "Change directory"
- Changes current directory based on specified path

```
----------

$ cd ../
Desktop/  Documents/  Downloads/  Music/
Pictures/  Public/  Templates/  Videos/ sandbox/
```

Building: *Creating directories*

`mkdir`

- “Make directory”
- Creates new directory based on specified name

```
----------

$ mkdir mydir
$ ls
file1  file2  mydir/
```

Building: *Creating directories*

```
$ mkdir parent/child
mkdir: cannot create directory ‘parent/child’: No
such file or directory
$ mkdir -p parent/child
```

- **`-p`** builds the specified directory structure if it doesn’t already exist (including parent directories)
- Very useful when scripting

Building: *Creating files*

`touch`

- Modifies file timestamps --- creation, access, modification, etc.
- Often used to create files (without input or redirection)

```
----------

$ cd mydir ; touch newfile
$ ls
newfile
```

Building: *Moving and renaming files*

`mv`

- “Move”
- Moves or renames files/directories (**be careful**! --- see manual)

```
----------

$ mv newfile otherfile
$ ls
otherfile
```

Building: *Moving and renaming files*

```
$ cd ../ ; ls
file1  file2  mydir/
$ mv mydir/ otherdir
$ ls
file1  file2  otherdir/
$ mv file1 otherdir/ ; ls
file2  otherdir/
$ ls otherdir/
file1  otherfile
```

Building: *Removing files*

`rm`

- “Remove”
- Removes files (**be careful**! --- see manual)

```
----------

$ rm file2 ; ls
otherdir/
```

Building: *Removing directories*

`rmdir`

- “Remove directory”
- Removes the specified directory if it’s empty

```
----------

$ rmdir otherdir/
rmdir: failed to remove 'otherdir/': Directory
not empty
```

Building: *Removing directories*

```
$ rm -r mydir
```

- Removes a directory *and* its files recursively

```
----------

$ ls otherdir/
file1  otherfile
$ rm -r otherdir/
$ ls
```

Building: *Redirection*

```
“>” (redirect stdout)
“&>” (redirect stdout and stderr)
“>>” (append stdout to ...)
“&>>” (append stdout and stderr to ...)

----------

$ lscpu > lscpu.out
```

Building: *Viewing files*

`more` and **`less`**

- Utilities to view or **page** through files
- **`less`** is a modern version of **`more`**, please use **`less`**
- http://www.greenwoodsoftware.com/less/faq.html#history

```
$ less lscpu.out
```

- "Space" to page, can use "u" , "d" , or arrows to move

Building: *Writing to files*

`echo`

- Output a string to the shell (**`stdout`**)
- **`-e`** recognizes escaped characters

```
$ echo "Hello world!"
Hello world!
$ echo -e "\"Hello world\"!"
"Hello world"!
```

Building: *Writing to files*

```
$ echo "Hello world!" > hello.out
$ less hello.out
Hello world!
```

- Echo a string and redirect it to a file
  - “**`>`**” operator b/c **`stdout`**
  - View the file (via **`less`**)

Building: *Writing to files*

`cat`

- Concatenate files, write to **`stdout`**, and even write to files
- If no path is given, **`cat`** begins to read from **`stdin`**

```
$ cat hello.out
Hello world!
```

Building: *Writing to files*

```
$ cat > hello2.out
This is some text
^D
```

- Writing to a file requires redirection (“**`>`**”) to a file (**`hello2.out`**)
- Enter **`ctrl d`** (**`^D`**) to finish writing

Building: *Writing to files*

```
$ cat hello.out hello2.out > hellos.out
$ cat hellos.out
Hello world!
This is some text
```

- Specify however many files --- redirect them to an output file

Building: *Revisiting redirection*

“|” (pipe)

- Redirecting **output** of one program **as input** of another program

```
----------

$ netstat | less
```

Utilities: *Pattern matching*

`grep`

- “**G**lobally search for a **r**egular **e**xpression and **p**rint matching lines”
- Matches a string/pattern input to files (per line)
    - **`-i`** to ignore cases
    - **`-E`** to use (extended) regular expressions (more on this later)
    - **`-R`** to recursively match

Utilities: *Pattern matching*

```
$ grep -i "intel" lscpu.out
Vendor ID:                GenuineIntel
Model name:               Intel(R) Xeon(R) CPU @ 2.20GHz
```

Utilities: *File contents*

`wc`

- "Word count"
- Statistics on bytes, characters, words, lines, etc. of a file

```
----------

$ wc lscpu.out
  24  153 1147 lscpu.out
```

Utilities: *File contents*

`diff`

- Difference between two files
- The output shows the differences as --- **which lines** of **file 1** need to be [added, changed, or deleted (**`a`**, **`c`**, **`d`**)] to match **which other lines** of **file 2**

Utilities: *File contents*

```
$ lscpu -x > lscpux.out
$ diff lscpu.out lscpux.out
5c5
< On-line CPU(s) list: 0-7
---
> On-line CPU(s) mask: ff
15,16c15,16
< CPU MHz:                  2200.188
< BogoMIPS:                 4400.37
```

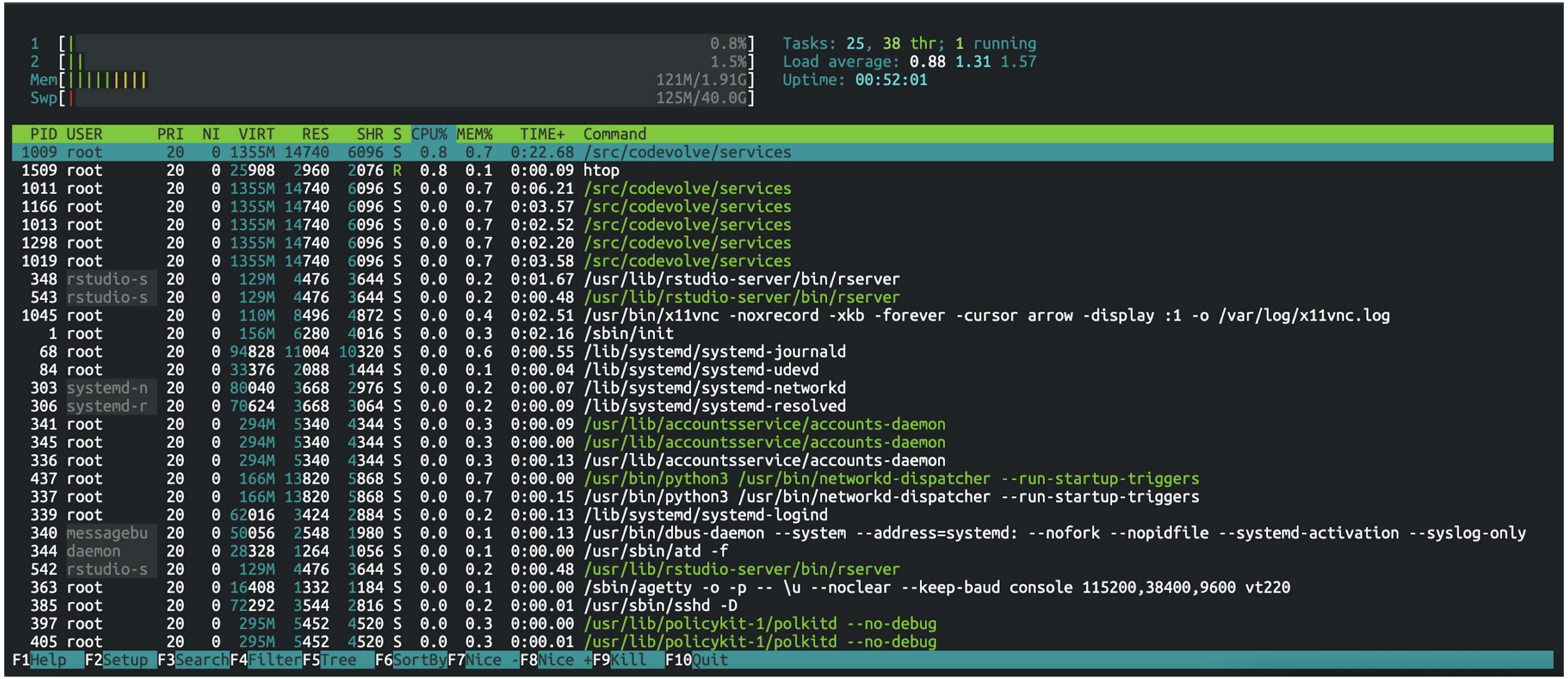
... see next slide

Utilities: *File contents*

```
---
> CPU MHz:
2199.998
> BogoMIPS:
4399.99
23c23
< NUMA node0 CPU(s):   0-7
---
> NUMA node0 CPU(s):   ff
```

Utilities: *Process management*

`htop` or `top`



```
  1  [|                                                                         0.8%]   Tasks: 25, 38 thr; 1 running
  2  [||                                                                        1.5%]   Load average: 0.88 1.31 1.57
  Mem[|||||||||                                                           121M/1.91G]   Uptime: 00:52:01
  Swp[|                                                                   125M/40.0G]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 1009 root       20   0 1355M 14740  6096 S  0.8  0.7  0:22.68 /src/codevolve/services
 1509 root       20   0 25908  2960  2076 R  0.8  0.1  0:00.09 htop
 1011 root       20   0 1355M 14740  6096 S  0.0  0.7  0:06.21 /src/codevolve/services
 1166 root       20   0 1355M 14740  6096 S  0.0  0.7  0:03.57 /src/codevolve/services
 1013 root       20   0 1355M 14740  6096 S  0.0  0.7  0:02.52 /src/codevolve/services
 1298 root       20   0 1355M 14740  6096 S  0.0  0.7  0:02.20 /src/codevolve/services
 1019 root       20   0 1355M 14740  6096 S  0.0  0.7  0:03.58 /src/codevolve/services
  348 rstudio-s  20   0  129M  4476  3644 S  0.0  0.2  0:01.67 /usr/lib/rstudio-server/bin/rserver
  543 rstudio-s  20   0  129M  4476  3644 S  0.0  0.2  0:00.48 /usr/lib/rstudio-server/bin/rserver
 1045 root       20   0  110M  8496  4872 S  0.0  0.4  0:02.51 /usr/bin/x11vnc -noxrecord -xkb -forever -cursor arrow -display :1 -o /var/log/x11vnc.log
    1 root       20   0  156M  6280  4016 S  0.0  0.3  0:02.16 /sbin/init
   68 root       20   0 94828 11004 10320 S  0.0  0.6  0:00.55 /lib/systemd/systemd-journald
   84 root       20   0 33376  2088  1444 S  0.0  0.1  0:00.04 /lib/systemd/systemd-udevd
  303 systemd-n  20   0 80040  3668  2976 S  0.0  0.2  0:00.07 /lib/systemd/systemd-networkd
  306 systemd-r  20   0 70624  3668  3064 S  0.0  0.2  0:00.09 /lib/systemd/systemd-resolved
  341 root       20   0  294M  5340  4344 S  0.0  0.3  0:00.09 /usr/lib/accountsservice/accounts-daemon
  345 root       20   0  294M  5340  4344 S  0.0  0.3  0:00.00 /usr/lib/accountsservice/accounts-daemon
  336 root       20   0  294M  5340  4344 S  0.0  0.3  0:00.13 /usr/lib/accountsservice/accounts-daemon
  437 root       20   0  166M 13820  5868 S  0.0  0.7  0:00.00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
  337 root       20   0  166M 13820  5868 S  0.0  0.7  0:00.15 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
  339 root       20   0 62016  3424  2884 S  0.0  0.2  0:00.13 /lib/systemd/systemd-logind
  340 messagebu  20   0 50056  2548  1980 S  0.0  0.1  0:00.13 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
  344 daemon     20   0 28328  1264  1056 S  0.0  0.1  0:00.00 /usr/sbin/atd -f
  542 rstudio-s  20   0  129M  4476  3644 S  0.0  0.2  0:00.48 /usr/lib/rstudio-server/bin/rserver
  363 root       20   0 16408  1332  1184 S  0.0  0.1  0:00.00 /sbin/agetty -o -p -- \u --noclear --keep-baud console 115200,38400,9600 vt220
  385 root       20   0 72292  3544  2816 S  0.0  0.2  0:00.01 /usr/sbin/sshd -D
  397 root       20   0  295M  5452  4520 S  0.0  0.3  0:00.00 /usr/lib/policykit-1/polkitd --no-debug
  405 root       20   0  295M  5452  4520 S  0.0  0.3  0:00.01 /usr/lib/policykit-1/polkitd --no-debug
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Utilities: *Other commands*

We’ll look at a few more shell commands later in the week including:

- **`find`**
- **`sed`**
- **`make`**
- **`sort`**

… and several others

Usability: *Navigating your own commands*

- **Up/down arrows** allow you to navigate your prior commands
- **`ctrl r`** allows **autocompletion** based on command history
- The **`history`** command is also useful to jog your memory
- Tab completion --- will try to match a file in a specified directory

Usability: *Navigating your own commands*

- **`ctrl c`** interrupts and **kills** a process
- **`ctrl a`** to the **beginning** of the line, **`ctrl e`** to the **back**
- Stringing commands together using boolean operators (**`&&`**) or separators (`;`)
- **`clear`** clears the terminal screen

Usability: *Filesystem navigation tips*

- **`pushd`** and **`popd`** --- create a stack of directories
- Setting **variables** to a particular path
  - Backticks (` ) to **evaluate** a command and use its output (often with **`pwd`**)

```
$ echo `pwd`
/root/sandbox
$ s=`pwd` ; echo $s
/root/sandbox
```

Examples: *Questions*

1. Does your sandboxed machine support “**`avx`**” instructions (according to **`lscpu`**)?
2. How many active sockets (according to **`netstat`**) are of type “**`DGRAM`**”?
3. Can you check how many sockets (according to **`netstat`**) are of type **`DGRAM`** *every 5 seconds*? ***

Exercises: *#1*

- **`lscpu`** has some **output** that we could as **input** for a **pattern match**

```
$ lscpu | grep "avx"
```

Examples: *#2*

- **`netstat`**’s output shows each active socket (and type) **per line**
- We could use the **output** as **input** for a **pattern match**
- We could then **count** the **number of lines** that matched

----

```
$ netstat | grep "DGRAM" | wc -l
```

(What’s **`-l`** ? See the manual!)

Examples: *#3*

Would be nice to check in **a loop,** maybe use a **script**, and **view** the output **incrementally** …

We’ll start with this tomorrow.