SE 234 Basic Development and Operations

#7 Basic Linux

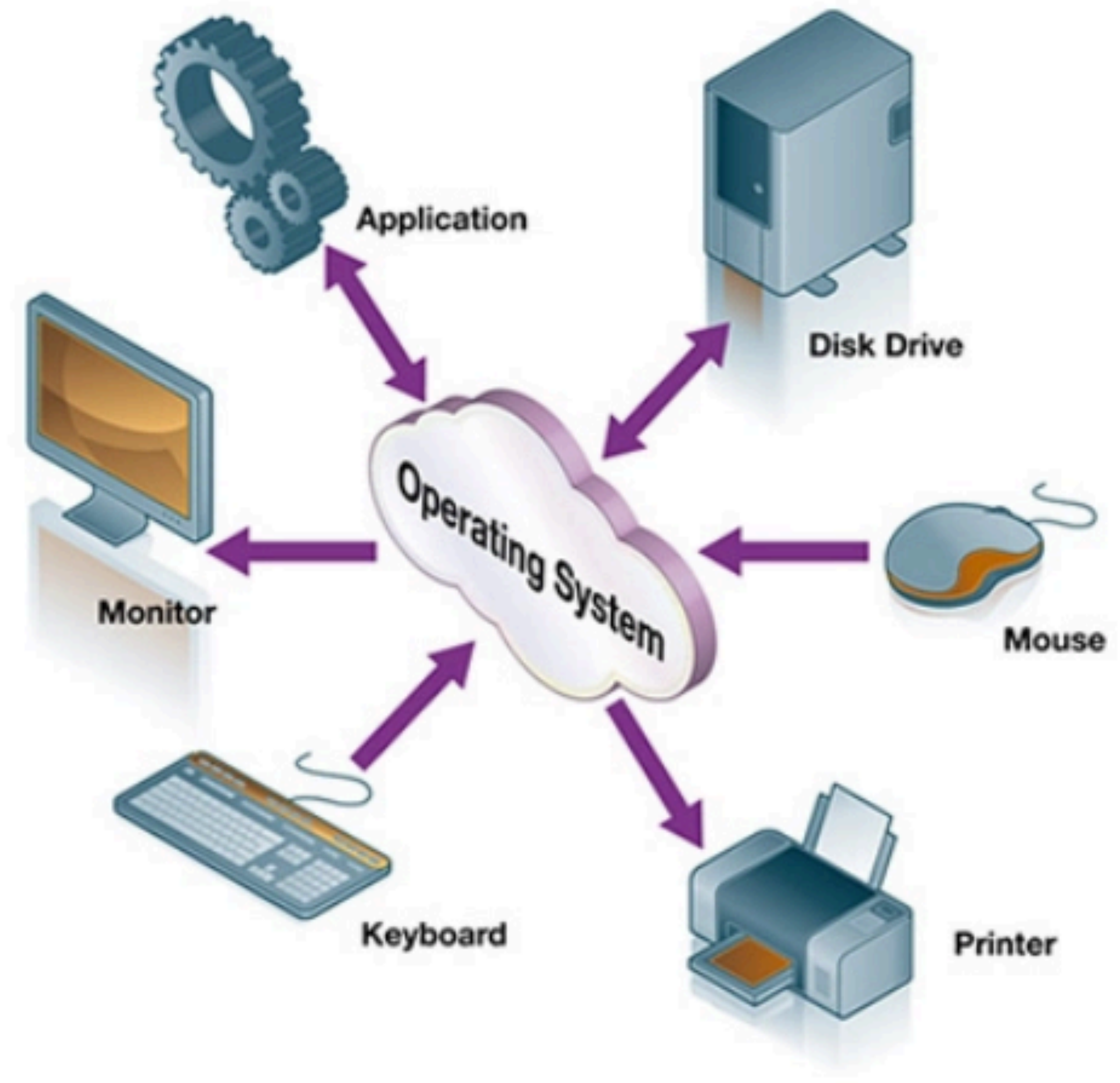
College of Arts, Media and Technology
Chiang Mai University

Lect Passakorn Phannachitta, D.Eng.
passakorn.p@cmu.ac.th
College of Arts, Media and Technology
Chiang Mai University, Chiangmai, Thailand

Agenda

- What is Unix?
- Linux vs Unix
- CLI
- Commonly-used commands

Operating System

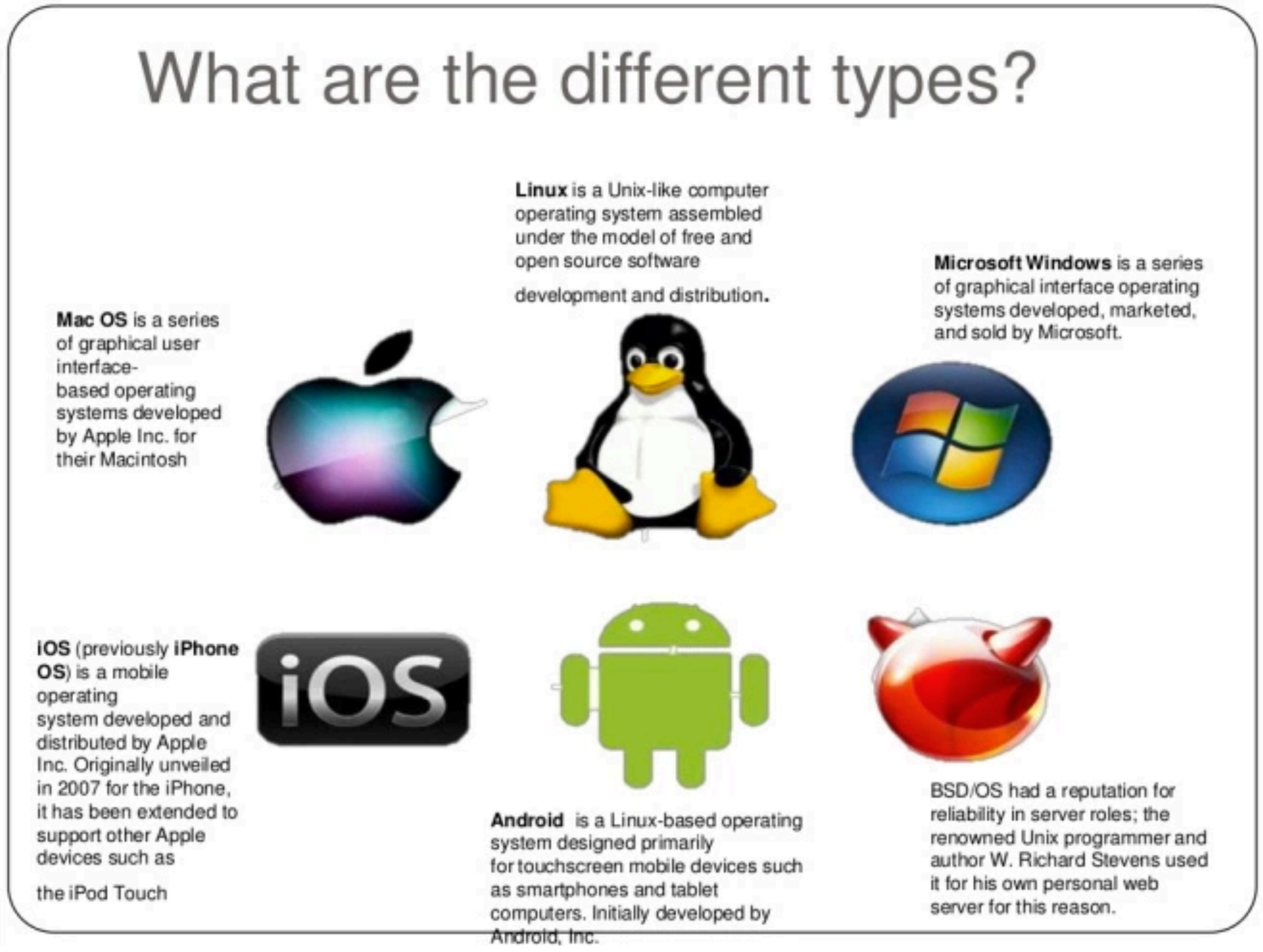


ref: https://linuxinfotech.wordpress.com/ (Accessed 2019)

What is Unix

- A multi-task and multi-user Operating System
- Principle: Do one thing, do it well
- Support most of the platforms available

Why we need to know about Unix



What are the different types?

Linux is a Unix-like computer operating system assembled under the model of free and open source software development and distribution.

Microsoft Windows is a series of graphical interface operating systems developed, marketed, and sold by Microsoft.

Mac OS is a series of graphical user interface-based operating systems developed by Apple Inc. for their Macintosh

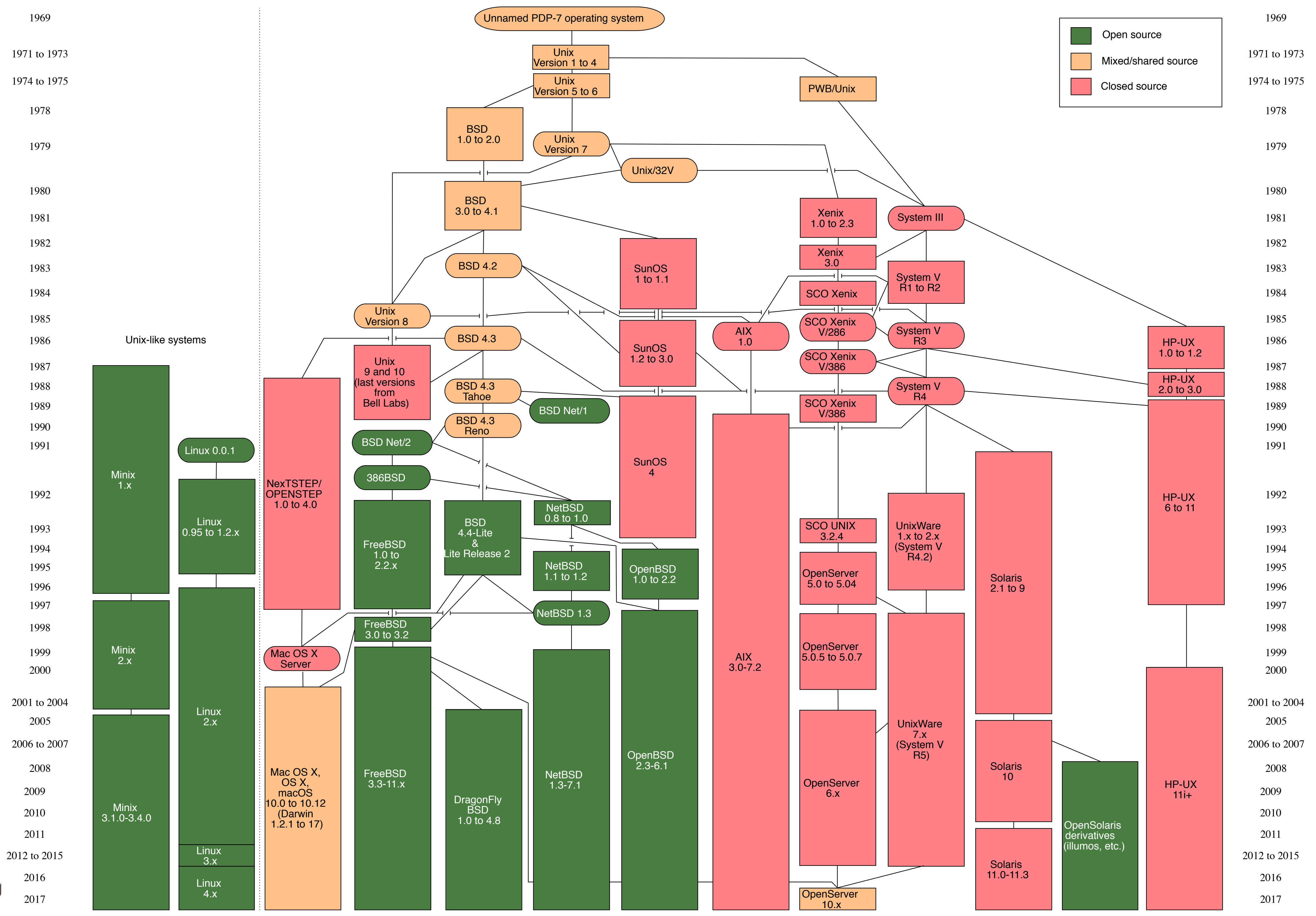
iOS (previously **iPhone OS**) is a mobile operating system developed and distributed by Apple Inc. Originally unveiled in 2007 for the iPhone, it has been extended to support other Apple devices such as the iPod Touch

Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet computers. Initially developed by Android, Inc.

BSD/OS had a reputation for reliability in server roles; the renowned Unix programmer and author W. Richard Stevens used it for his own personal web server for this reason.

ref: https://linuxinfotech.wordpress.com/ (Accessed Jan 2019)

History and Family Tree



College of Ar
Chiang Mai University

Let's start with Unix

- Unix was born in 1969 (Linus Torvalds was also born in this year)
- Before its success in 1969 AT&T has also collaborates with MIT and GE in an attempt to build an OS names MULTICS but failed.
- Ken Thompson, the head, brought the idea back to Bell Labs and created the first version in assembly.

History

- After collaborating Dennis Ritchie, the father of C language, the first kernel was successful around 1973.
- In press in 1974 and has become reconginzed.
- AT&T was sued against the anti-monopoly law, and it was banned from selling software. Thus Unix kernel is forced to be distributed to universities.

History

- In 1979, the most sophisticated distribution built by the University of California at Berkeley has become recognized as BSD Unix — Free version.
- AT&T's version more focus on commercialize and was successfully the os of the choice in many workstations markets, e.g., seen in that of Sun, HP, IBM, and NeXT.
- Free version (BSD) and commercialized version (AT&T) have become fully apart.

History

- AT&T's coined a regulation named Single Unix specification where MacOS was the only OS qulified the specification and still remained.
- In 1984, Richard Stallman was upset about the too commercialized AT&T Unix with numerous license and regulation matters, so he started GNU project
  - GNU (GNU is Not Unix) Software
  - With a purpose of a free UNIX — "Free as in freedom, not free beer"
  - Begin by reimplementing the UNIX utilities, e.g., compiler and libraries

History

- AT&T's coined a regulation named Single Unix specification where MacOS was the only OS qulified the specification and still remained.
- In 1984, Richard Stallman was upset about the too commercialized AT&T Unix with numerous license and regulation matters, so he started GNU project
    - GNU (GNU is Not Unix) Software
    - With a purpose of a free UNIX — "Free as in freedom, not free beer"
    - Begin by reimplementing the UNIX utilities, e.g., compiler and libraries

History

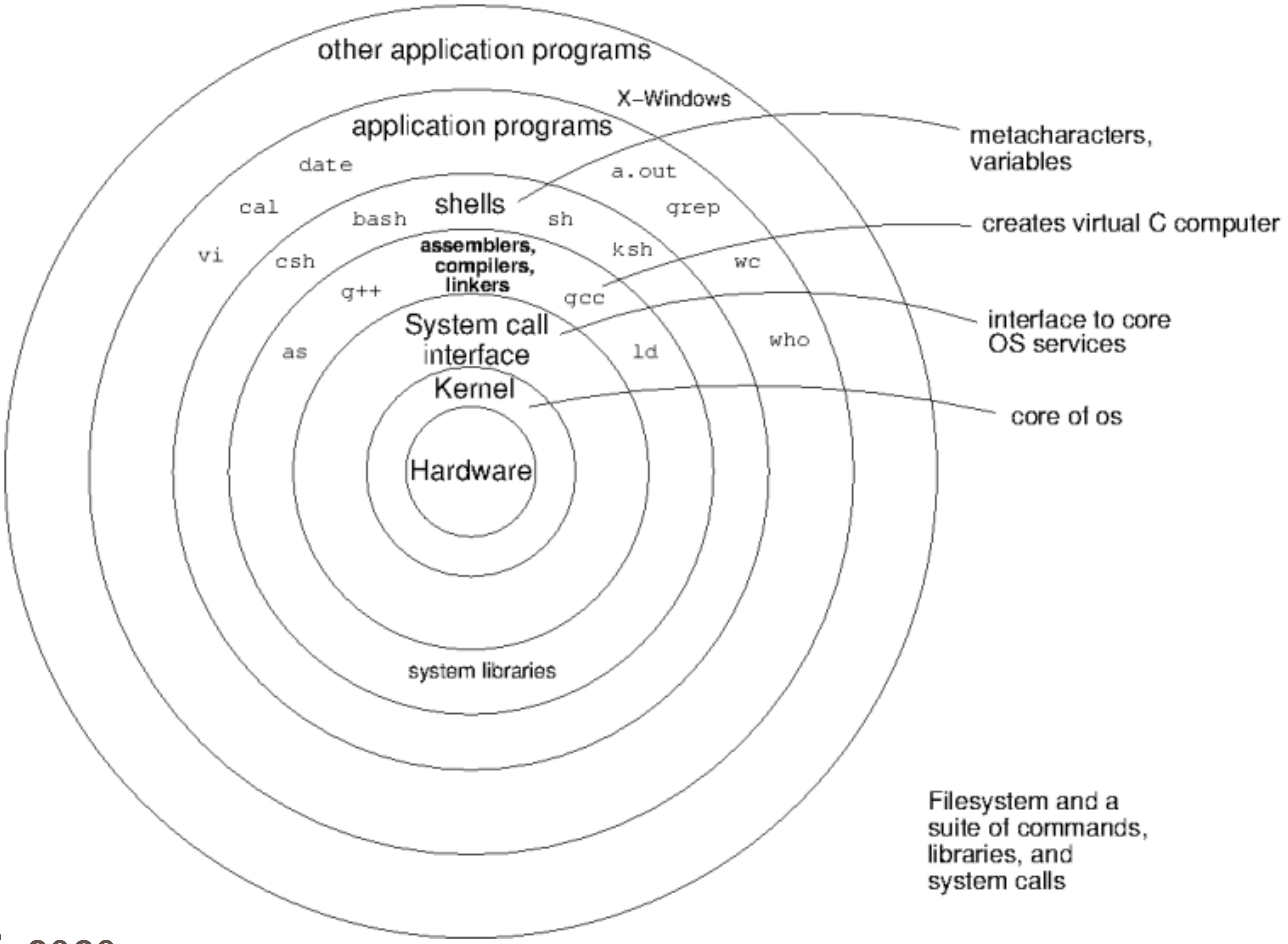
- GNU project was very successful that many developers were willing to join Stallman's great idea.
- Free Software Foundation was founded to strengthen this open-source software idea and soon GPL (GNU General Public License) was widely used.
- In 1991, as a undergrad student, Linus Torvalds tried to develop a full OS from a kernel named Minix and he was successful and named it Linux.

In sum, why Unix/Linux?

- Free
- Fully customizable
- Dramatically Stable

- = Ideal for programmers and scientists

Architecture



Filesystem and a suite of commands, libraries, and system calls

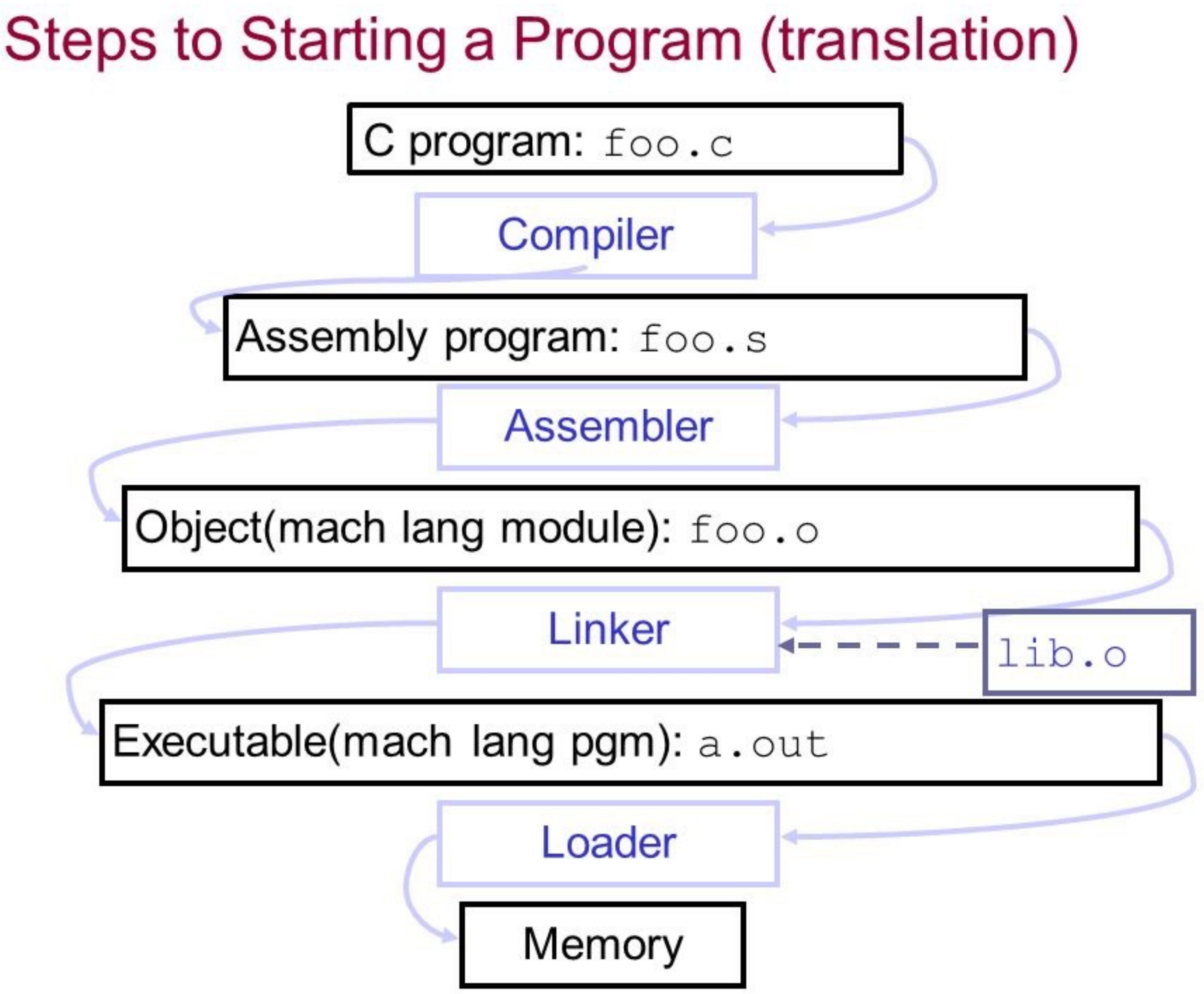
Kernel

- A german word, kern, which means core
- It is the main component of most computer operating systems; it's a bridge between application software to the hardware of a computer

System call

- “The programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on” … wikipedia
- An application program **makes a system call** to get the operating system to perform a service for it, like reading from a file.

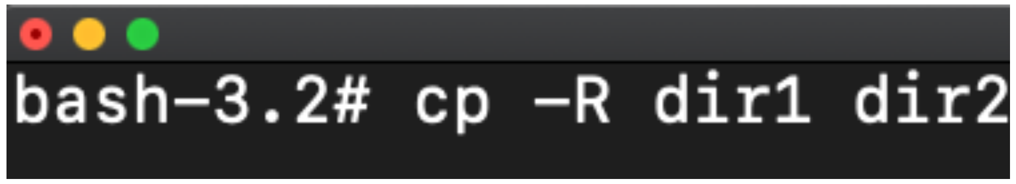
Assemblers, Compilers, Linkers, and Loaders


Steps to Starting a Program (translation)
C program: foo.c
Compiler
Assembly program: foo.s
Assembler
Object(mach lang module): foo.o
Linker
lib.o
Executable(mach lang pgm): a.out
Loader
Memory

Ref: https://slideplayer.com/slide/7068459/ (Accessed Jan 2018)

Shell

- After logging in, Linux/Unix starts another program called the shell
- The shell is a software run through terminal for interpreting commands inputted by the user, then manages their execution, and finally get the result.
- The shell communicates with kernel
- Shell interacts with users via CLI (Command-line interface)



```
bash-3.2# cp -R dir1 dir2
```

Responsibilities of Shell

- Reading input and parsing the command line
- Evaluating special characters
- Setting up **pipes**, **redirection**, and **background** processing
- Handling **signals**
- Setting up programs for execution

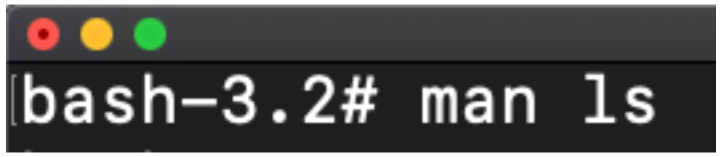
BASH - Bourne Again Shell

- Bash is a command language first released in 1989
- Bash has been distributed widely as the default login shell for most Linux distributions and Apple's macOS.
- Bash is a command processor that typically runs in a text window where the user types commands that cause actions. Bash can also read and execute commands from a file, called a **shell script**.

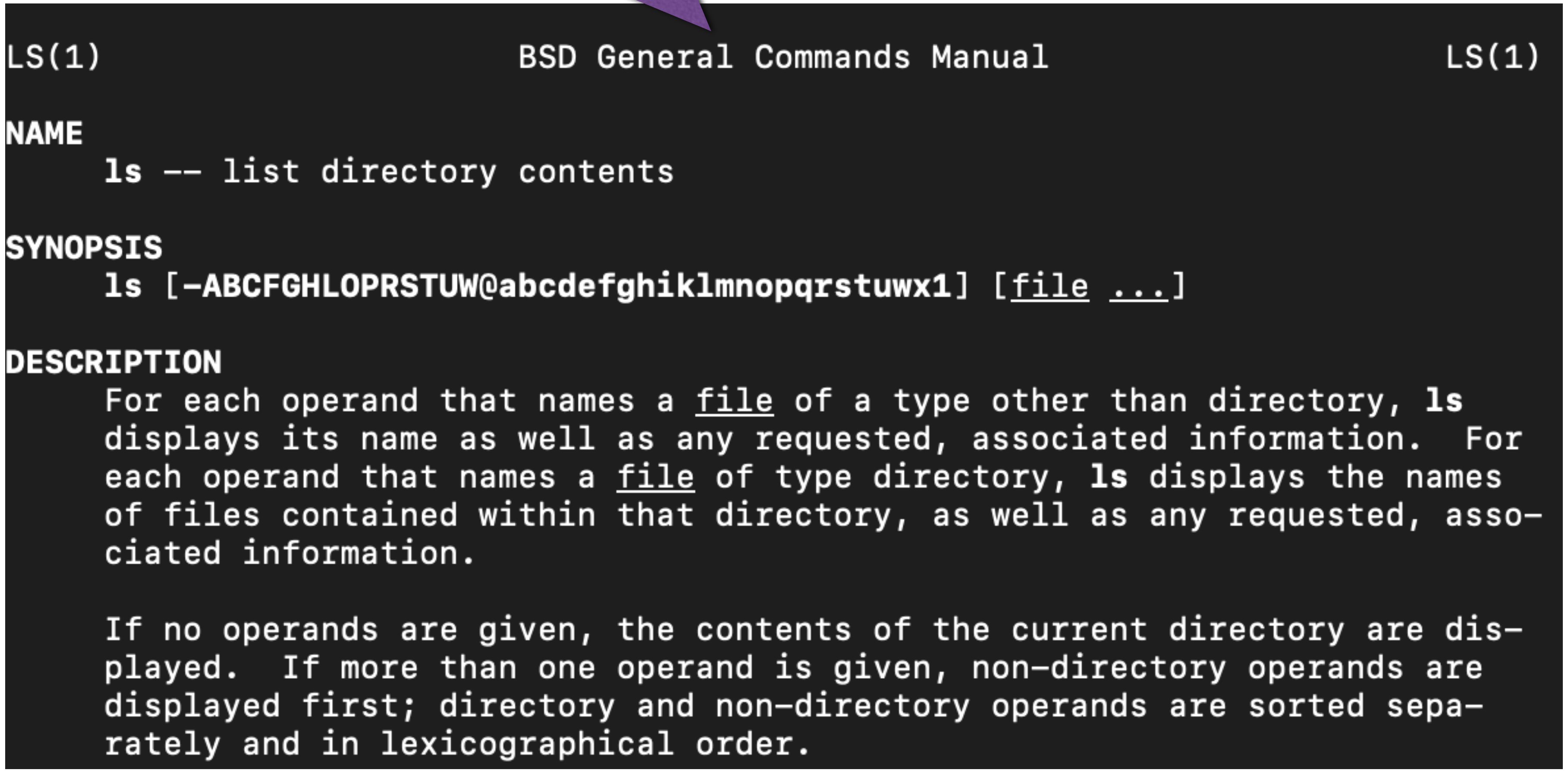
```
bash-3.2# 
```

Whenever you need help with a command

- Type “man”, e.g., it stands for manual, before a command name
- E.g.,



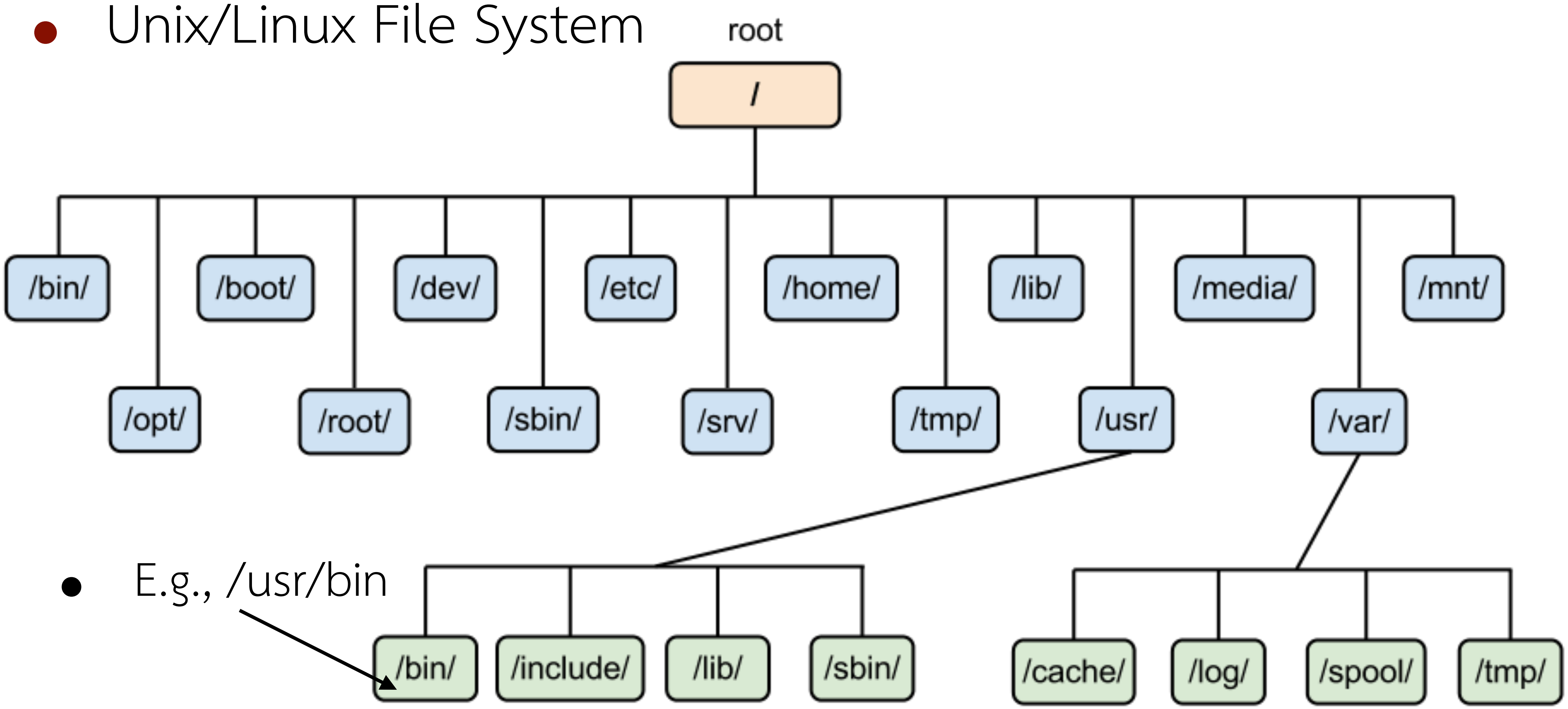
```
bash-3.2# man ls
```



```
LS(1)                     BSD General Commands Manual                    LS(1)

NAME
     ls -- list directory contents

SYNOPSIS
     ls [-ABCFGHLOPRSTUW@abcdefghiklmnopqrstuwx1] [file ...]

DESCRIPTION
     For each operand that names a file of a type other than directory, ls
     displays its name as well as any requested, associated information.  For
     each operand that names a file of type directory, ls displays the names
     of files contained within that directory, as well as any requested, asso-
     ciated information.

     If no operands are given, the contents of the current directory are dis-
     played.  If more than one operand is given, non-directory operands are
     displayed first; directory and non-directory operands are sorted sepa-
     rately and in lexicographical order.
```

The next thing to know

- Unix/Linux File System



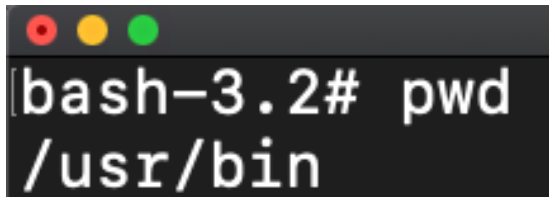
- E.g., /usr/bin

- Unix/Linux file names are case sensitive.

ref: https://freedompenguin.com/articles/how-to/learning-the-linux-file-system/ Accessed Jan 2019

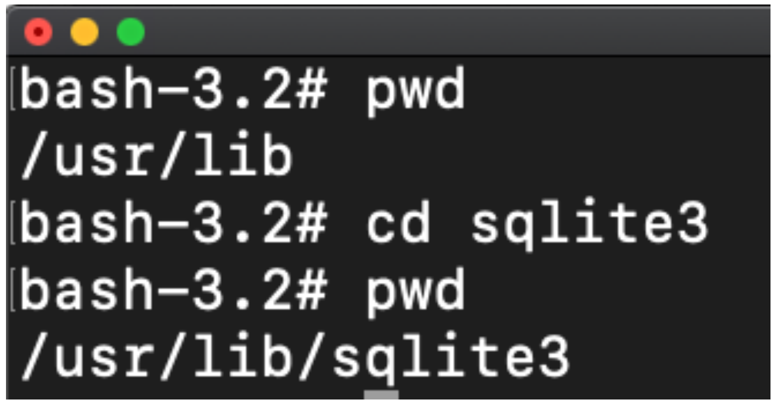
Commands related to filesystem traversal

- `pwd` is used to find your current path.



```
bash-3.2# pwd
/usr/bin
```

- `cd` is used to change to a specific directory.

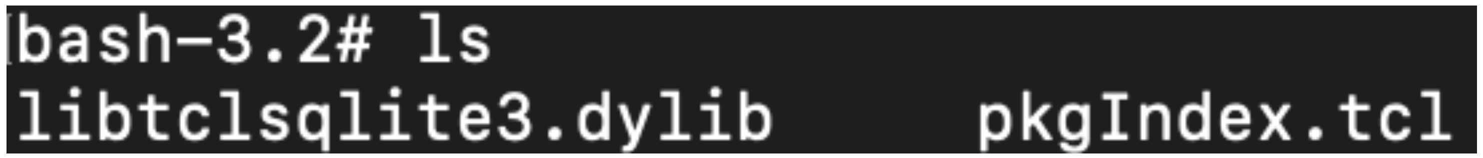


```
bash-3.2# pwd
/usr/lib
bash-3.2# cd sqlite3
bash-3.2# pwd
/usr/lib/sqlite3
```

  - Try `cd ..`, `cd ~`, and `cd` and see the result.

Commands related to filesystem traversal

- `ls` is used to list the files in the current directory
- ls has many options (try `man ls` to see more options)
  - -l long list (displays lots of info)
  - -t sort by modification time
  - -S sort by size
  - -h list file sizes in human readable format
  - -r reverse the order
- Options can be combined: `ls -al`



```
bash-3.2# ls
libtclsqlite3.dylib      pkgIndex.tcl
```

Displaying a File

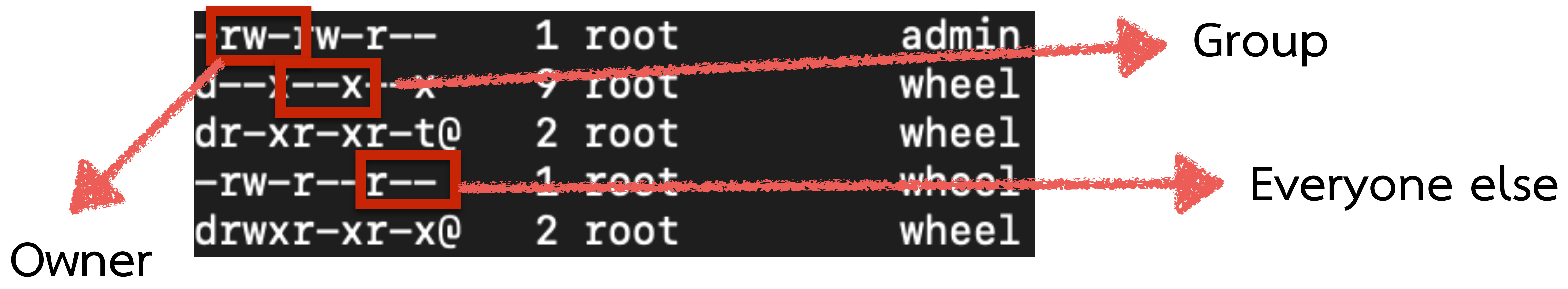
- Common commands are cat, less, head, and tail
  - `cat` dumps the entire file to stdout
  - `less` displays a file. It allows forward/backward movement within it
  - `head` displays the top part, e.g., 10 lines, of the file. A `-nXX` is commonly used to specific the number of lines.
  - `head` displays the bottom part, e.g., 10 lines, of the file. A `-nXX` is commonly used to specific the number of lines.

Manipulating File(s)

- Common commands are cp, mv, and rm.
  - `cp` copies a file
  - `mv` moves or renames a file
  - `rm` removes a file. Adding a `-r` arg will allow rm to remove a directory. (r stands for recursively)

Permission Levels in Linux

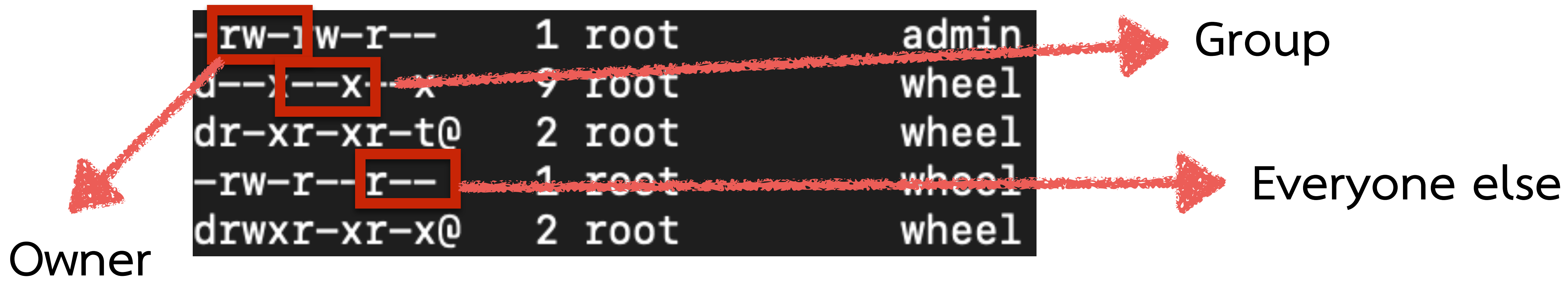
- There are 3 symbols representing permission: r, w, and x.
  - r means read permission, w means write permission, and x means execute permission
- There are 3 administrative spaces
- These information can be shown by `ls -l`



```
-rw-rw-r--     1 root            admin
d--x--x--x     9 root            wheel
dr-xr-xr-t@    2 root            wheel
-rw-r--r--     1 root            wheel
drwxr-xr-x@    2 root            wheel
```

Permission Levels in Linux

- A command \`chmod\` is used to modify these permission
  - Syntax: chmod [user/group/others/all]+[permission] [file(s)]

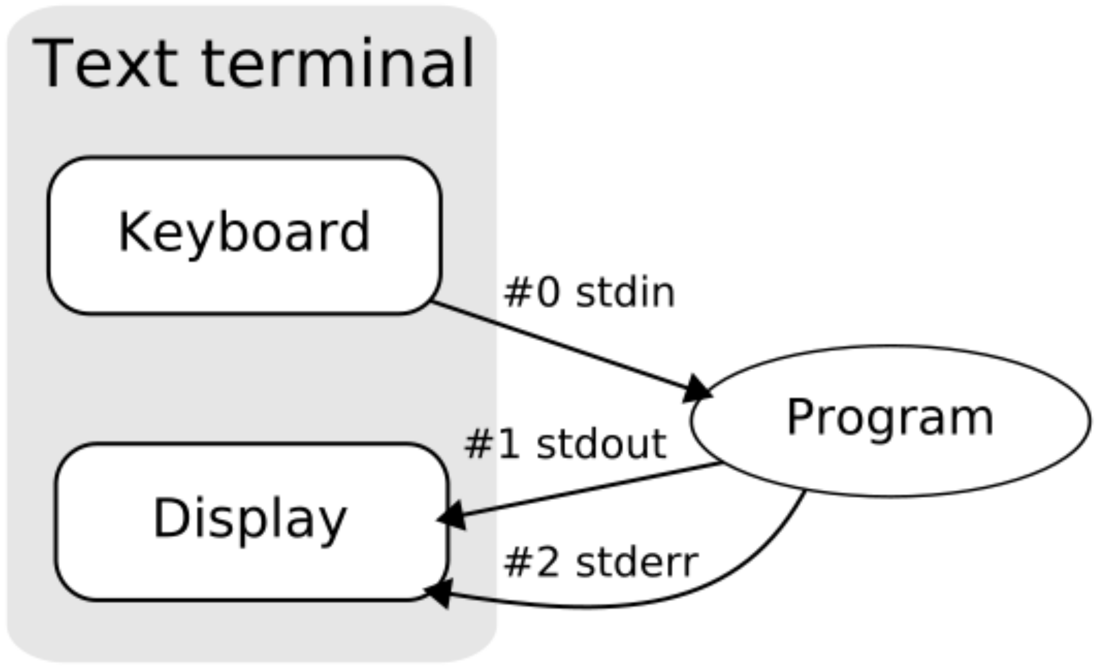


```
-rw-rw-r--     1 root          admin
d--x--x--x     9 root          wheel
dr-xr-xr-t@    2 root          wheel
-rw-r--r--     1 root          wheel
drwxr-xr-x@    2 root          wheel
```

Executing an Executable Program

- Use `./program name` to execute it

Unix/Linux's Standard Streaming

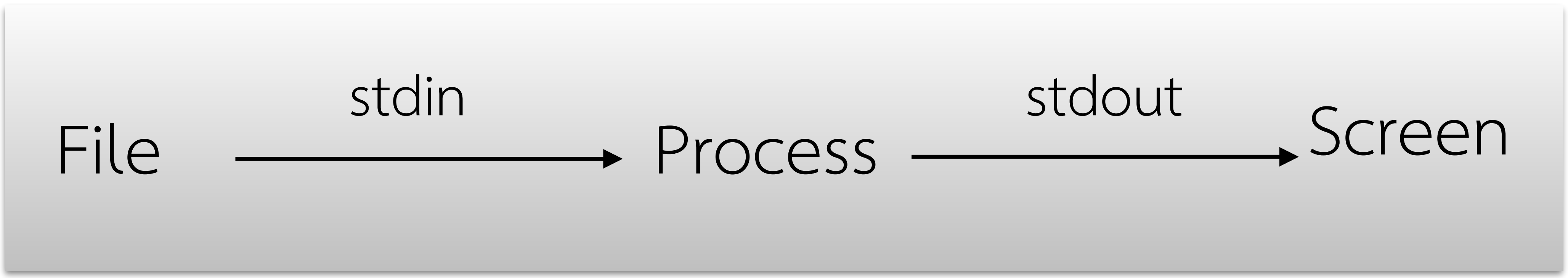


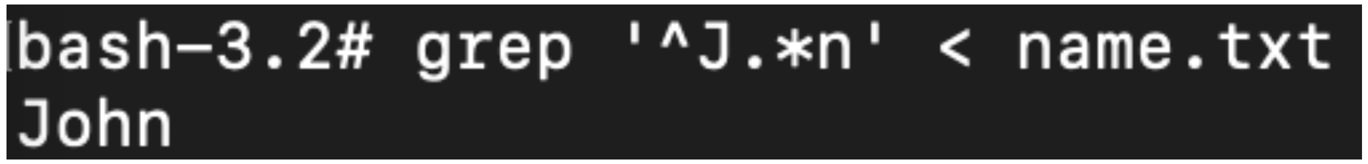
- stdin = standard input
- stdout = standard output
- stderr = standard error

ref: https://www.thaicert.or.th/papers/general/2011/pa2011ge006.html (Accessed Jan 2019)

Redirection

- Input redirection (<)





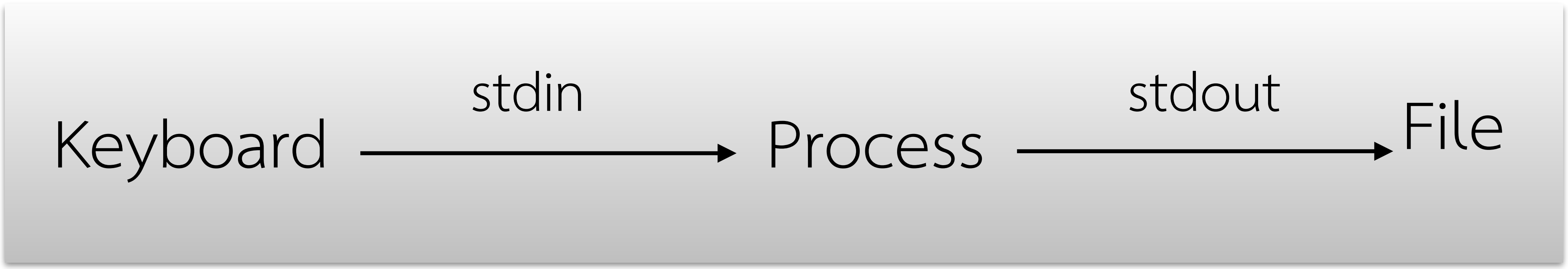
```
bash-3.2# grep '^J.*n' < name.txt
John
```

- E.g., File name "name.csv"

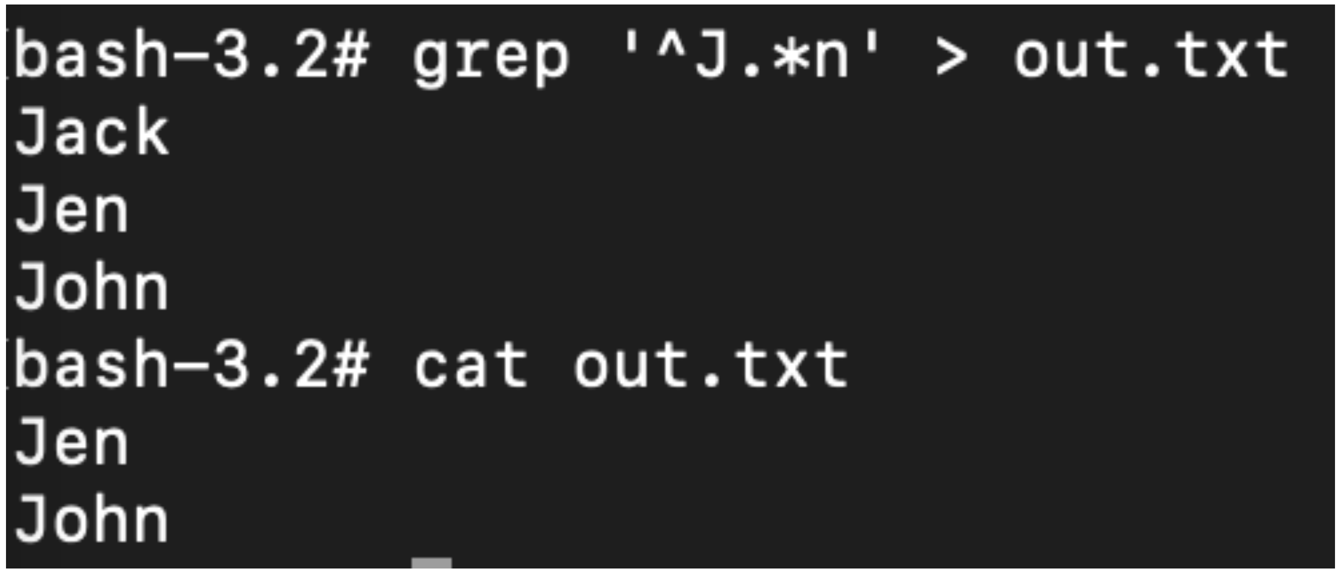
```
Michael
James
John
Robert
David
```

Redirection

- Output redirection (>,>>)



- Both, > and >> are to redirect to file, but >> is the append mode

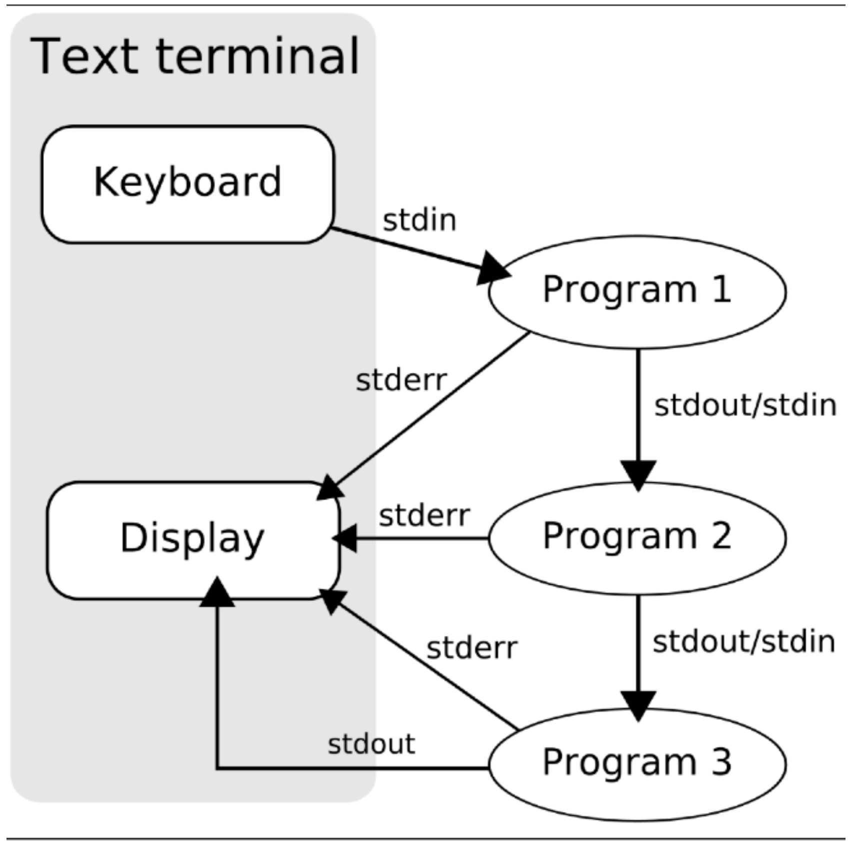


```
bash-3.2# grep '^J.*n' > out.txt
Jack
Jen
John
bash-3.2# cat out.txt
Jen
John
```

CAMT 2020
College of Arts, Media and Technology
Chiang Mai University

Pipeline

- One of the most powerful standard tool for Linux

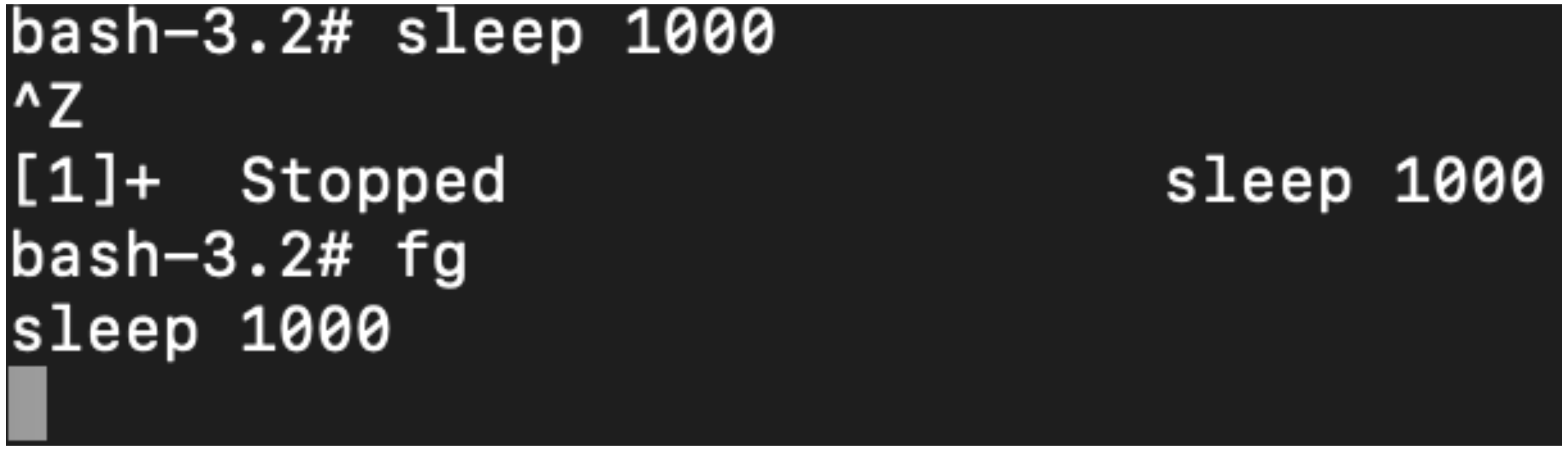


- It allows us to do something like:

```
bash-3.2# $ grep '^J.*n' < in.txt | cut -d, -f2 | sort | uniq > out.txt
```

Process Manipulation

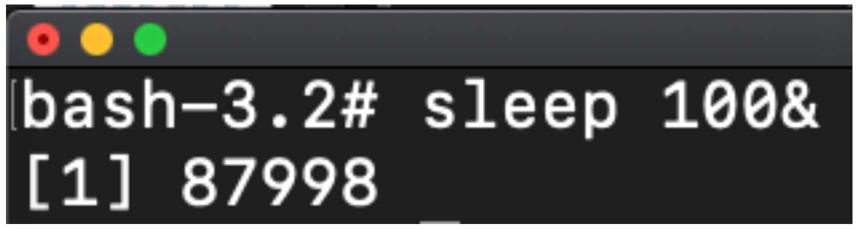
- Ctrl + D = End of the input
- Ctrl + C = Force terminate the process
- Ctrl + Z = Bring the process to background
  - You can use the command `fg` to bring it back to foreground



```
bash-3.2# sleep 1000
^Z
[1]+  Stopped                 sleep 1000
bash-3.2# fg
sleep 1000
```

Process Manipulation

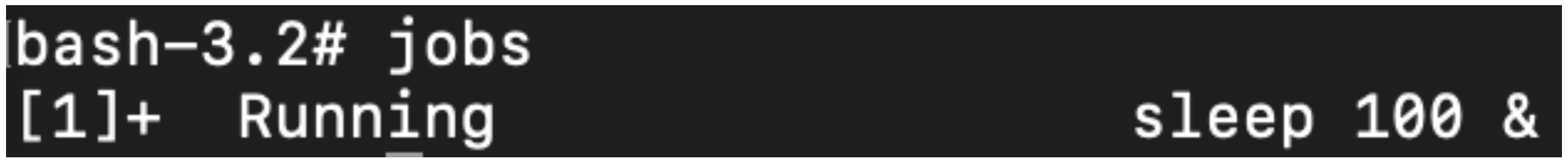
- You can force a process to run on background at the beginning by tailing an '&' to the end of the command before executing it.



```
bash-3.2# sleep 100&
[1] 87998
```

  - A process id is return

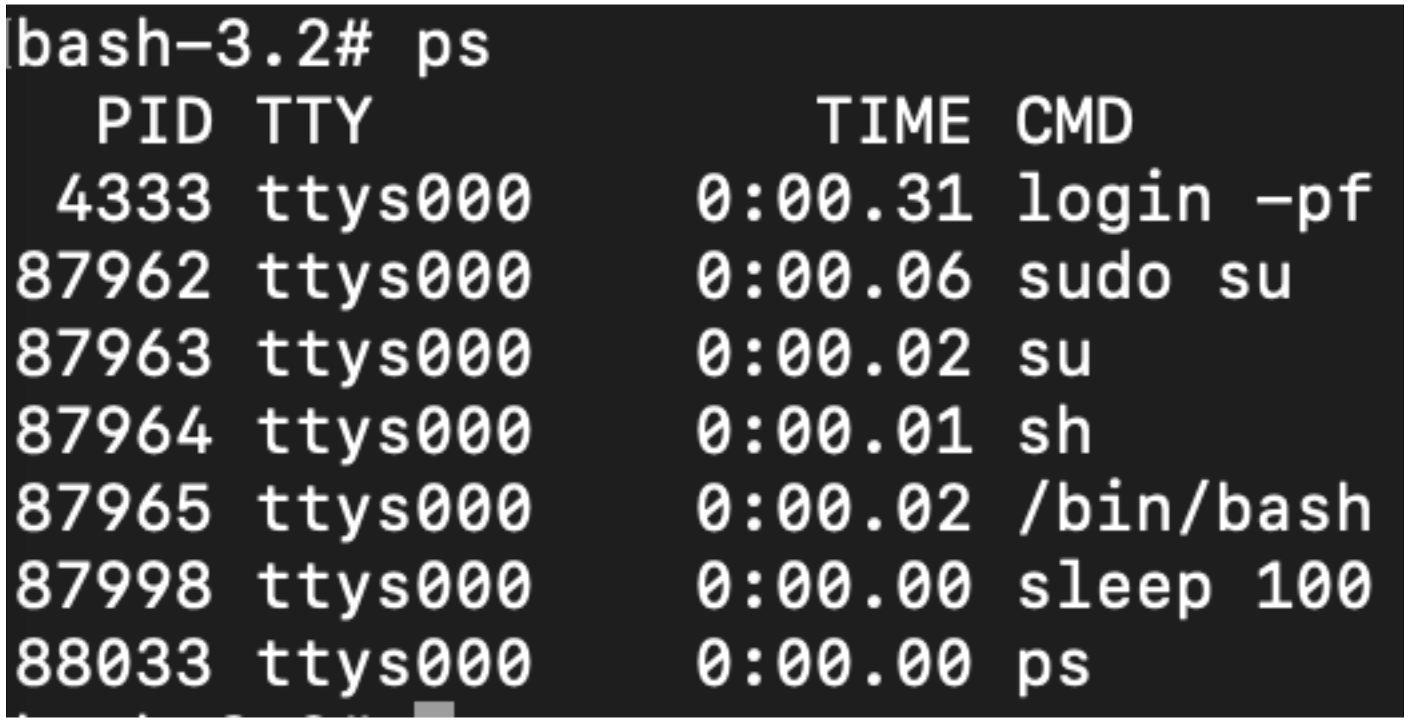
- \`jobs\` command will show the list of the background process



```
bash-3.2# jobs
[1]+  Running                 sleep 100 &
```

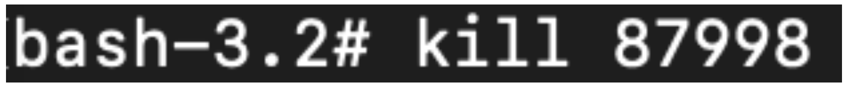
Process Manipulation

- A command `ps` will let you see all the process called in the current terminal. Adding a `-e` arg will show all the processes in the memory



```
bash-3.2# ps
  PID TTY           TIME CMD
 4333 ttys000    0:00.31 login -pf
87962 ttys000    0:00.06 sudo su
87963 ttys000    0:00.02 su
87964 ttys000    0:00.01 sh
87965 ttys000    0:00.02 /bin/bash
87998 ttys000    0:00.00 sleep 100
88033 ttys000    0:00.00 ps
```

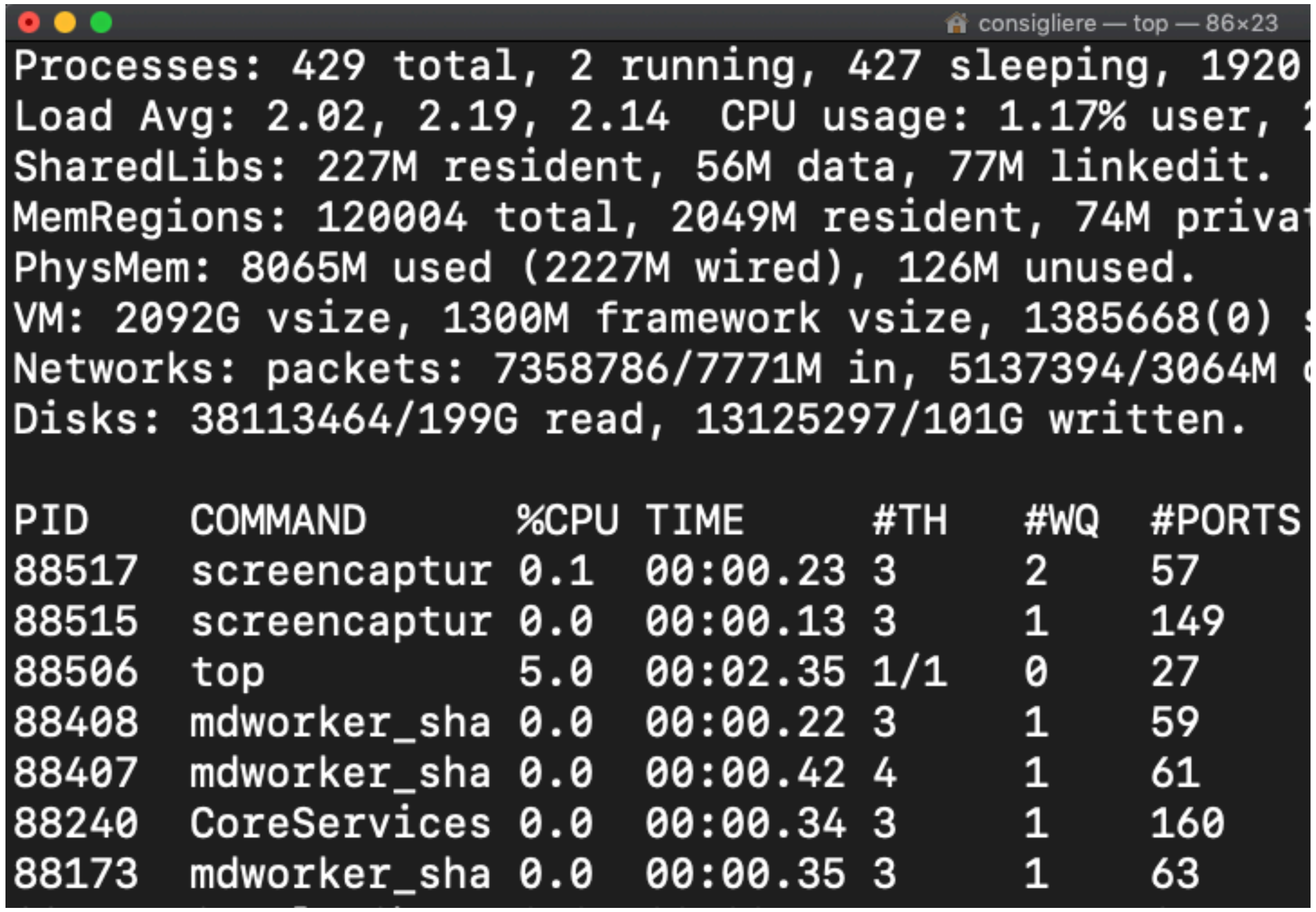
- We can directly kill one or more process using the kill command.



```
bash-3.2# kill 87998
```

The command `top`

- One of the most used command for developer.
- It shows the CPU usage of all processes.


consigliere — top — 86×23

```
Processes: 429 total, 2 running, 427 sleeping, 1920
Load Avg: 2.02, 2.19, 2.14  CPU usage: 1.17% user,
SharedLibs: 227M resident, 56M data, 77M linkedit.
MemRegions: 120004 total, 2049M resident, 74M priva
PhysMem: 8065M used (2227M wired), 126M unused.
VM: 2092G vsize, 1300M framework vsize, 1385668(0)
Networks: packets: 7358786/7771M in, 5137394/3064M
Disks: 38113464/199G read, 13125297/101G written.

PID    COMMAND      %CPU TIME     #TH   #WQ  #PORTS
88517  screencaptur 0.1  00:00.23 3     2    57
88515  screencaptur 0.0  00:00.13 3     1    149
88506  top          5.0  00:02.35 1/1   0    27
88408  mdworker_sha 0.0  00:00.22 3     1    59
88407  mdworker_sha 0.0  00:00.42 4     1    61
88240  CoreServices 0.0  00:00.34 3     1    160
88173  mdworker_sha 0.0  00:00.35 3     1    63
```

To discuss in the next class

- What if we can create a script weaving many commands to get a particular job done for you
    - We can make utilize the control flow, variables, background/foreground process.
- Shell script!

Question Time