



Unix  
Bash  
C  
GNU  
Systems

# Introduction to Software Systems

## Comp-206 : Introduction to Software Systems Lecture Week 1

Joseph Vybihal  
Computer Science  
McGill University

COMP 206 - Joseph Vybihal  
Software Systems



Unix  
Bash  
C  
GNU  
Systems

# Introduction

## Lecture 1



# Coordinates

- Joseph Vybihal
- McConnell 323
- [jvybihal@cs.mcgill.ca](mailto:jvybihal@cs.mcgill.ca)
- [www.cs.mcgill.ca/~jvybihal](http://www.cs.mcgill.ca/~jvybihal)
- Facebook group
  - Probably the best way to contact me for quick answers
  - Please tag me when you post and I will try to respond quickly
- Office hours:
  - Tuesday and Wednesday 3PM to 4PM
  - Or by appointment



# Research

## Intelligent Systems

- Artificial intelligence
- Group intelligent interactions (simulated & robotic)
- What is intelligence and thought?
- Inserting intelligent algorithms into every day software

## Undergraduate student opportunities

- COMP 400 or COMP 396
- — — — — 396
  - ECSE
  - COG
  - PHYS
  - Biology, neurology, studies in multi-agent systems



# Teaching Method

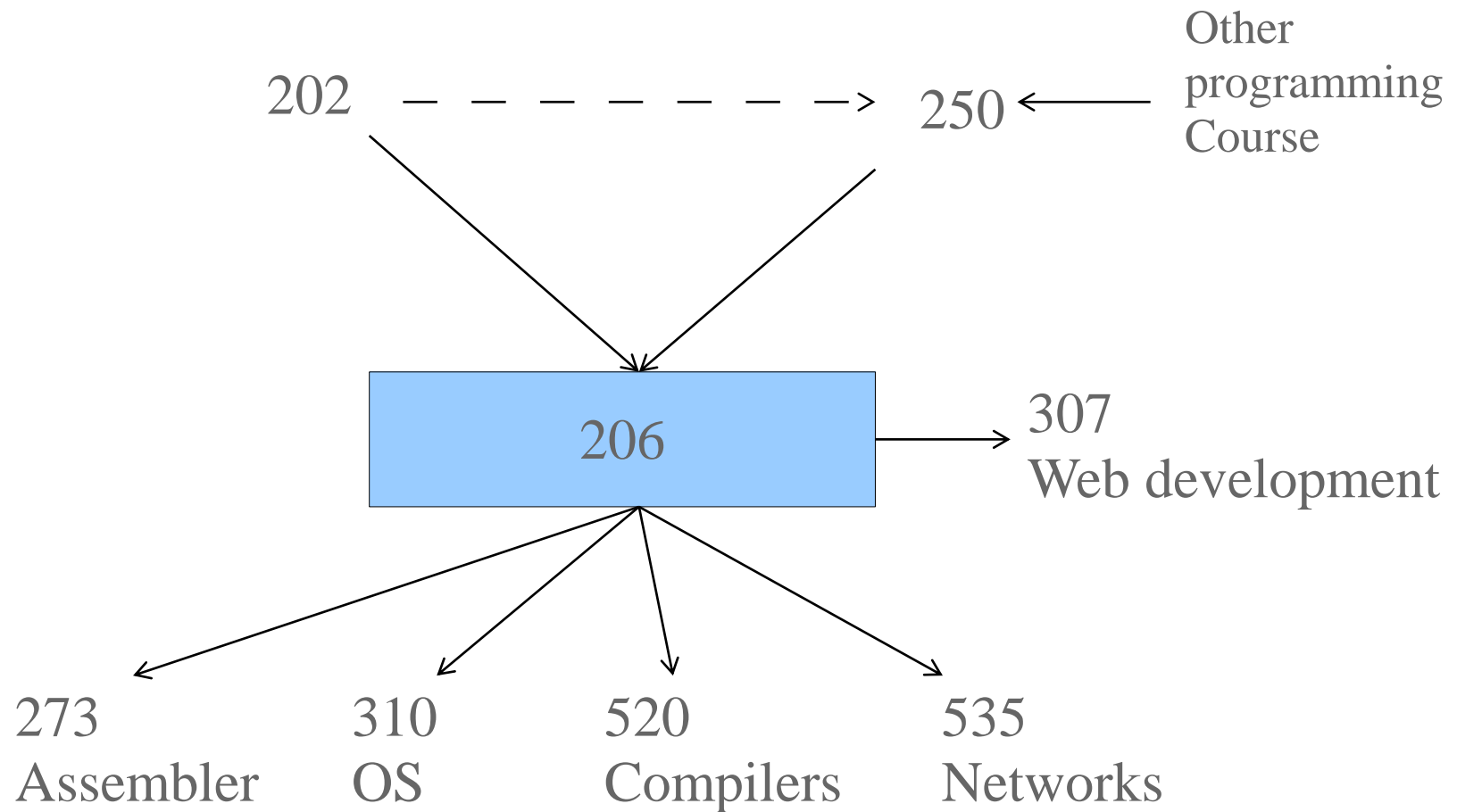
- Laptop
  - You might want to have yours handy
- Slides
  - They serve as both lecture outline, basic notes, and contain links to further readings
- Digital Ink
  - Some notes will be written digitally during class
- Lecture recordings?
  - Yes, if possible, however...
  - Students who attend class do better



Unix  
Bash  
C  
GNU  
Systems

COMP 206 - Joseph Vybihal  
Software Systems

# Prerequisites



It is not required but it is recommended to take this course only after you have taken 2 programming courses



# Course Contents

- Introduction to “under the hood” development
  - Command-line interfacing (next level after GUI)
  - The Unix operating system (we will use Linux)
  - Bash programming within a system’s context
  - Low-level system’s programming using the C language
  - Basic software engineering techniques
    - Git, gprof, gdb, make, gcc
  - Multi-language and systems integration programming



Unix  
Bash  
C  
GNU  
Systems

# Course Usefulness

## Science Students

- Introduction to systems
- Teaches intermediate programming techniques
- Learn algorithms that help in data capture and analysis
- Opens doors to further programming and science courses

## Software Engineers

- Combines programming with operating systems
- Introduction to basic software engineering techniques
- Introduces holistic and integrated thinking of the computer for software development





# Course Evaluation

- 6 assignments . . . 30%
- 2 class tests . . . 20% (10% each)
- 1 final exam. . . 50% (during final exam period)
- Deferred exam possible (no supplemental)
- 70% final exam option possible
  - Replaces class tests only
- Final exam tutorial
- Linux, Bash, C and advanced C tutorials



# Course Reading Material

- **Primary text**
  - Software Systems  
Vybihal & Azar  
Kendall/Hunt  
ISBN 978-0-7575-9514-1 (<https://www.kendallhunt.com/vybihal/>)
- **Other texts:**
  - Free on web
    - GNU Software; Louksides & Oram; O'Reilly; ISBN 1565921127
    - Drive into Python; Mark Pilgrim; Apress; ISBN 1590593561
  - Just Enough Unix; P.K. Anderson; McGraw Hill;  
ISBN 0697131726
  - C Programming Language; Kernighan & Ritchie; Prentice-Hall;  
ISBN 0131101633



# Special Instructions

**Late work:** You will be notified in advance of assignment due dates. All assignments are due on My Courses at the indicated time and date. Late assignments will lose 5% of its grade per day late. Assignments beyond 2 days late will not be accepted. You may not submit assignments via e-mail without permission.

**Additional Work:** Students with grades of D, F or J will not be given the opportunity to complete additional work to upgrade their grade.

**Re-grading:** Mistakes can occur when grading. Not surprisingly, requests for re-grading always involve those mistakes in which the student received fewer points than they deserved, rather than more points than they deserved. With that in mind: if you wish me to re-grade a question on an exam or assignment, I will do so. I reserve the right to re-grade other questions as well.

**Cheating/Collaboration:** Collaboration is encouraged but your discussions should be public in the sense that anyone including the professor should be allowed to listen in. Assignments are original works created by the student alone.

**Grading:** All software solutions must compile with zero errors and must run to be graded. It does not need to run correctly for grading but it must run. The grader will not fix your code or look at the source code to give you partial grades.



# Course Resources

- Detailed course outline
- Course website
- MyCourses
  - Assignments
  - Discussion board
  - Grades
  - TA information
  - Lecture slides



# Labs

- Labs
  - Trottier 3<sup>rd</sup> floor (any unlocked room and hallway)
  - Assistance:
    - [help@cs.mcgill.ca](mailto:help@cs.mcgill.ca)
    - Office: McConnell 209N
    - Phone: 514-398-7087
- Lab accounts
  - **DO NOT** use your McGill account! [First.last@mail.mcgill.ca](mailto:First.last@mail.mcgill.ca)
  - You must use a SOCS account
    - <https://newuser.cs.mcgill.ca>
    - You must be on campus for this to work or on a VPN
    - Forgot your Username and/or Password?
      - Reset it at <https://newpassword.cs.mcgill.ca/>



# What is a software system?

Readings: chapter 1, <http://whatis.techtarget.com/definition/system-software>,  
[https://en.wikipedia.org/wiki/System\\_software](https://en.wikipedia.org/wiki/System_software)



Unix  
Bash  
C  
GNU  
Systems

# Systems Programming

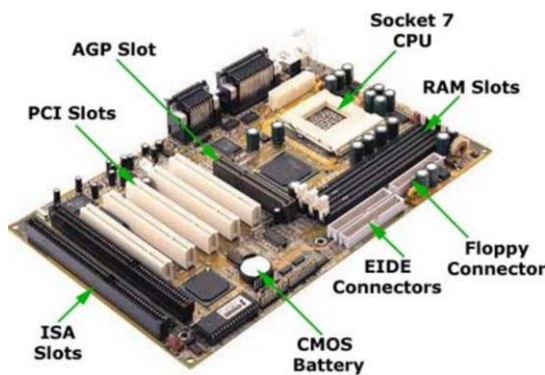


software

Linux™



Operating  
System



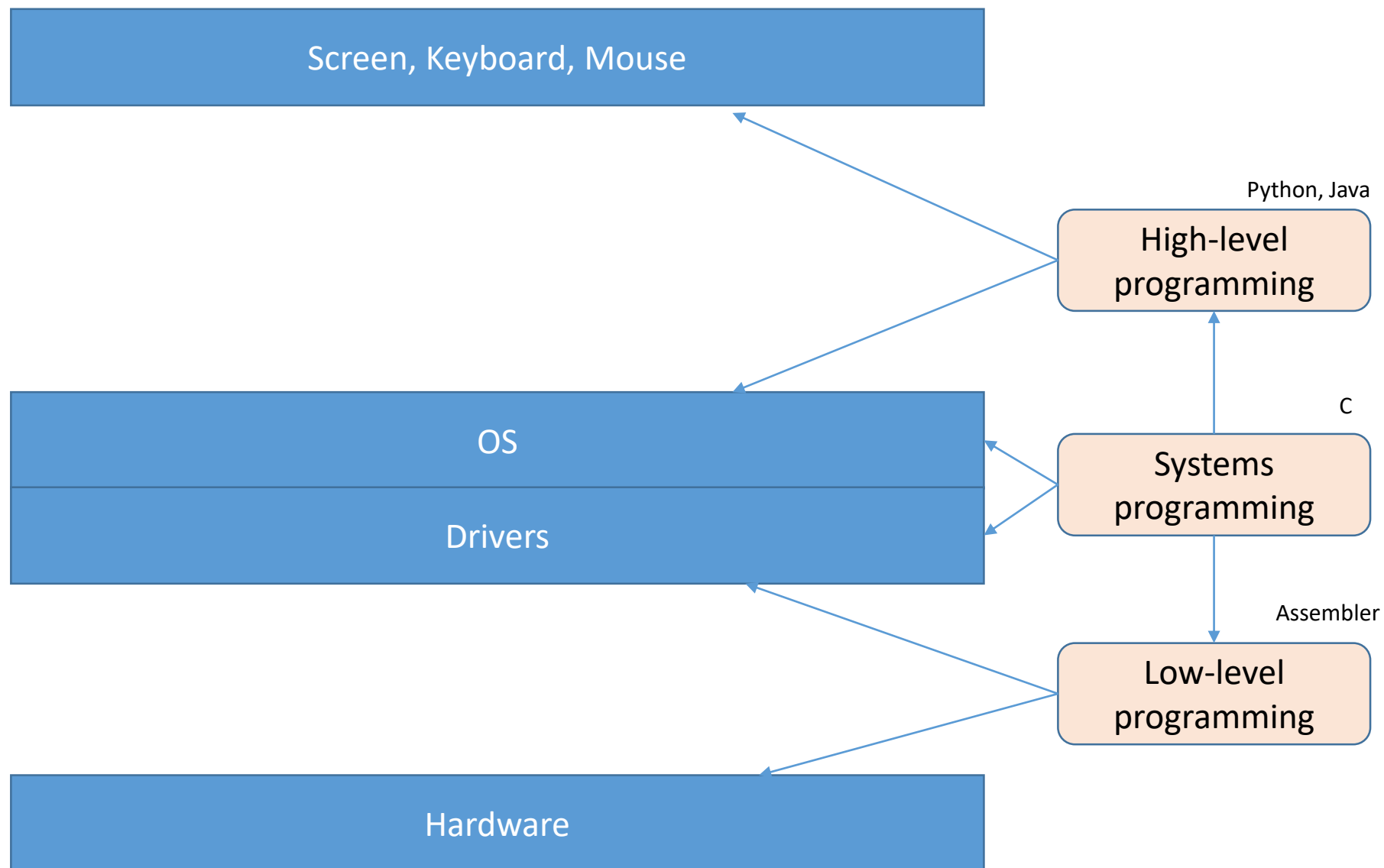
Writing software that  
directly interacts with  
either the operating  
system or the machine.



Unix  
Bash  
C  
GNU  
Systems

COMP 206 - Joseph Vybihal  
Software Systems

# System's Programming



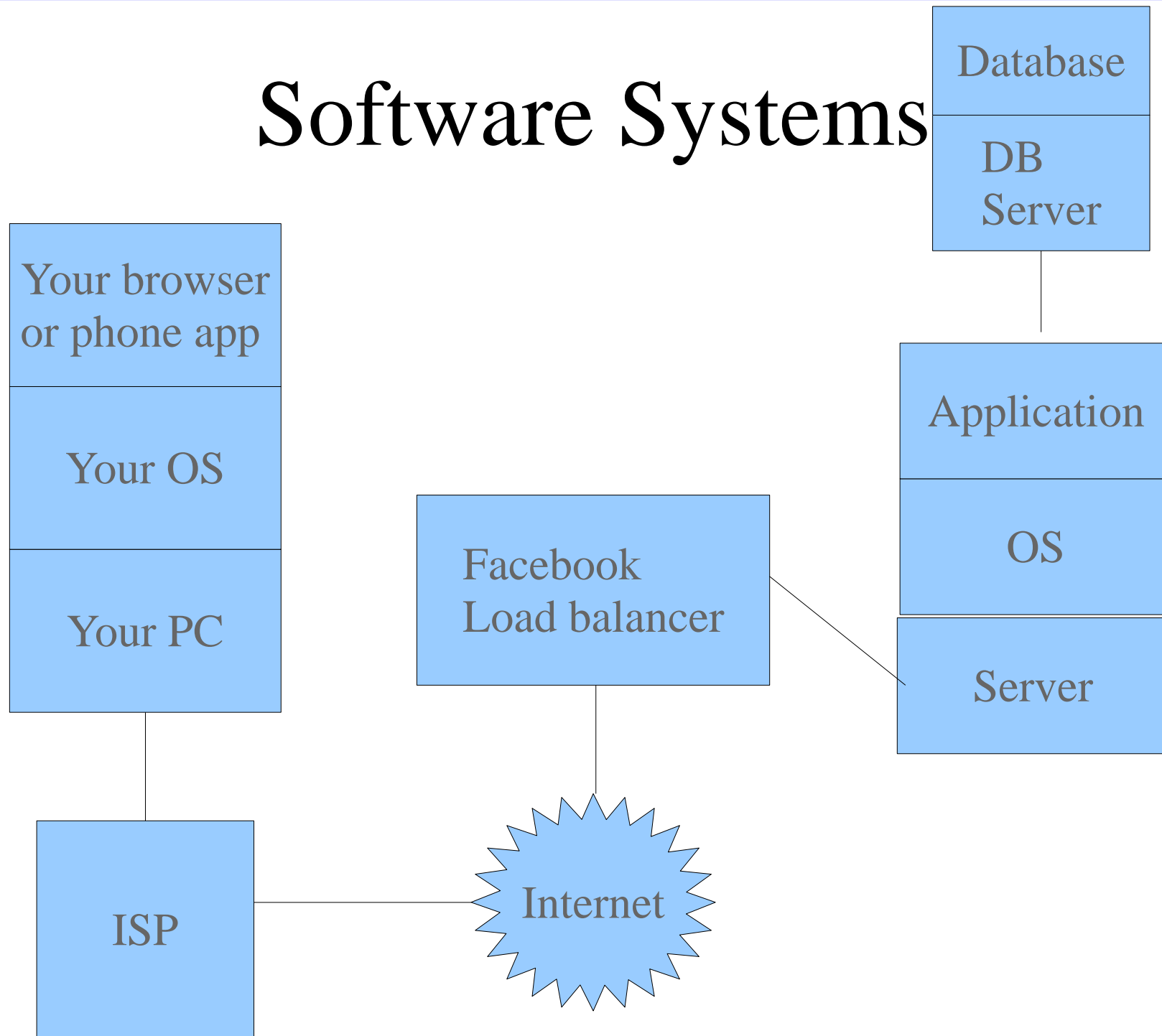




Unix  
Bash  
C  
GNU  
Systems

COMP 206 - Joseph Vybihal  
Software Systems

# Software Systems

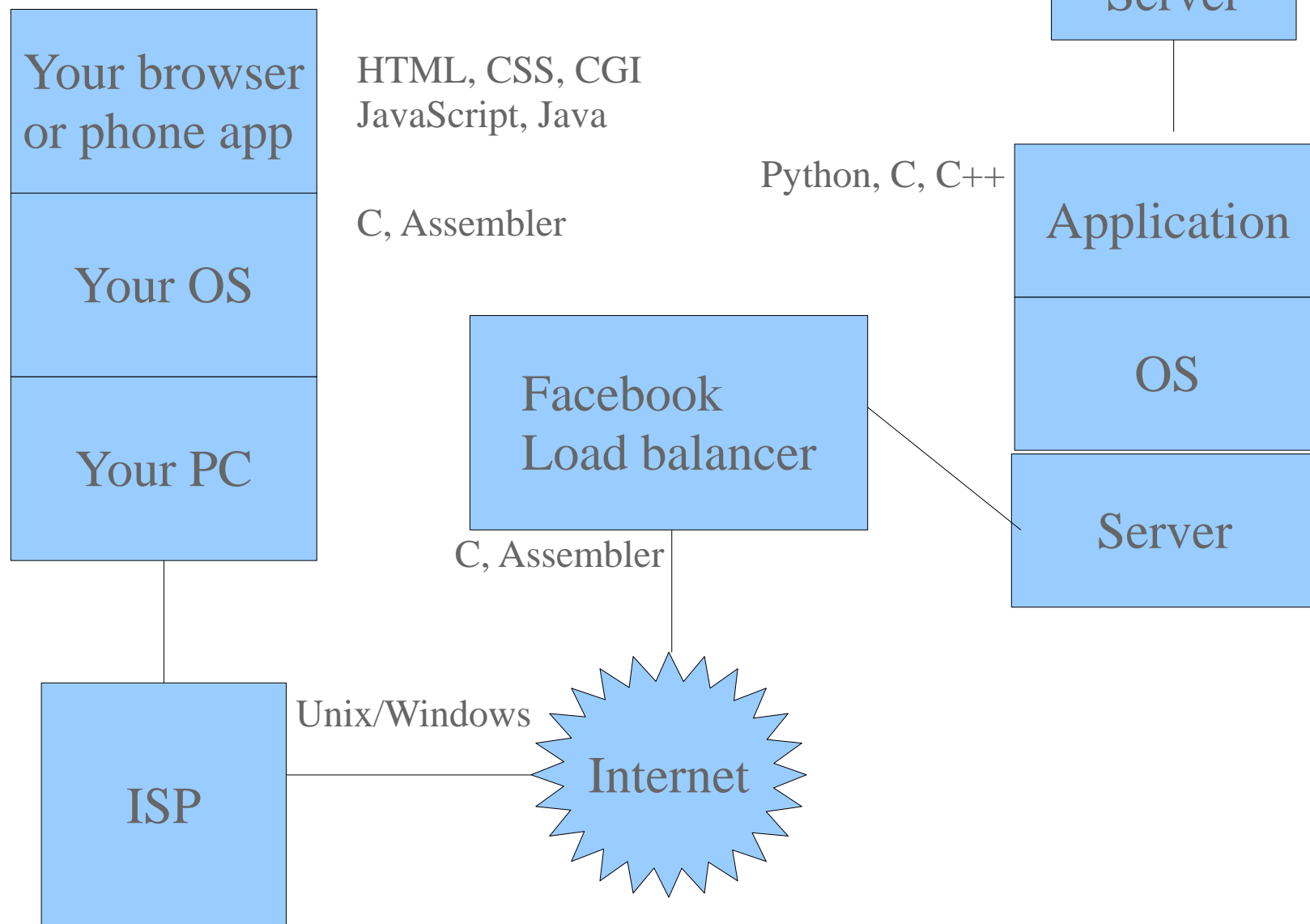




Unix  
Bash  
C  
GNU  
Systems

COMPTON 206 - Joseph Vybihal  
Software Systems

# Software Systems





Unix  
Bash  
C  
GNU  
Systems

# Software Development



Building software is a group effort

GIT, gProf, GDB, Make, GCC



# Command-line Development

```
himanshu@himanshu:~$ ls -l
total 116
-rw--x--x 1 himanshu himanshu  41 Jul  3 23:23 anotherfile
drwxr-xr-x 3 himanshu himanshu 4096 Jul  3 14:26 Desktop
drwxr-xr-x 2 himanshu himanshu 4096 Mar 31 17:54 Documents
drwxr-xr-x 2 himanshu himanshu 4096 Jul  3 21:53 Downloads
-rw-r--r-- 1 himanshu himanshu 8980 Mar 31 17:32 examples.desktop
-rw-r--r-- 1 himanshu himanshu  22 Jul  4 14:42 file1
-rw-r--r-- 1 himanshu himanshu  22 Jul  4 14:42 file2
-rw-r--r-- 1 himanshu himanshu  179 Jul  4 14:38 ls-dump
drwxr-xr-x 3 himanshu himanshu 4096 May 10 20:00 Music
-rw-r--r-- 1 himanshu himanshu   0 Jul  3 21:48 newfil
drwxr-xr-x 2 himanshu himanshu 4096 Jul  4 15:56 Pictures
drwxr-xr-x 3 himanshu himanshu 4096 Jul  4 15:13 practice
-rw-r--r-- 1 himanshu himanshu 20480 Jul  4 15:09 practice.tar
-rw-r--r-- 1 himanshu himanshu  2703 Jul  4 15:19 practice.tar.gz
drwxr-xr-x 2 himanshu himanshu 4096 Mar 31 17:54 Public
-rw--x--x 1 himanshu himanshu   52 Jul  3 23:22 somefile
drwxr-xr-x 2 himanshu himanshu 4096 Mar 31 17:54 Templates
drwxr-xr-x 2 himanshu himanshu 4096 Jul  3 21:31 test
-rw-r--r-- 1 himanshu himanshu  149 Jul  3 21:32 test.tar.gz
-rw-r--r-- 1 himanshu himanshu  310 Jul  3 21:45 test.zip
-rw-r--r-- 1 himanshu himanshu   36 Jul  4 15:56 textfile
drwxrwxr-x 2 himanshu himanshu 4096 Apr 21 12:47 Ubuntu One
-rw-r--r-- 1 himanshu himanshu  101 Jul  4 14:30 uname-output
drwxr-xr-x 2 himanshu himanshu 4096 Mar 31 17:54 Videos
himanshu@himanshu:~$
```

A lot of systems development occurs here. This course focuses on this interface.



# How to do well

- Program everything yourself
- Get used to using all the computer resources
  - Experiment
- Do the readings from the course outline
- Use the Help Desk for technical issues
- Use the TA for class & assignment help
- Use Facebook and Tag me for questions
  - Or the myCourses discussion board



Unix  
Bash  
C  
GNU  
Systems

COMP 206 - Joseph Vybihal  
Software Systems

# Questions?





## Lecture 2

# What is an operating system? Unix

Readings: chapter 2, [https://www.tutorialspoint.com/operating\\_system/os\\_linux.htm](https://www.tutorialspoint.com/operating_system/os_linux.htm)



# Outline

- Why is an operating system important?
- The Unix story
- The Unix architecture
- Remote access to the SOCS servers
- The SOCS server architecture
- Login, file transfer, basic working method
- Be careful of...

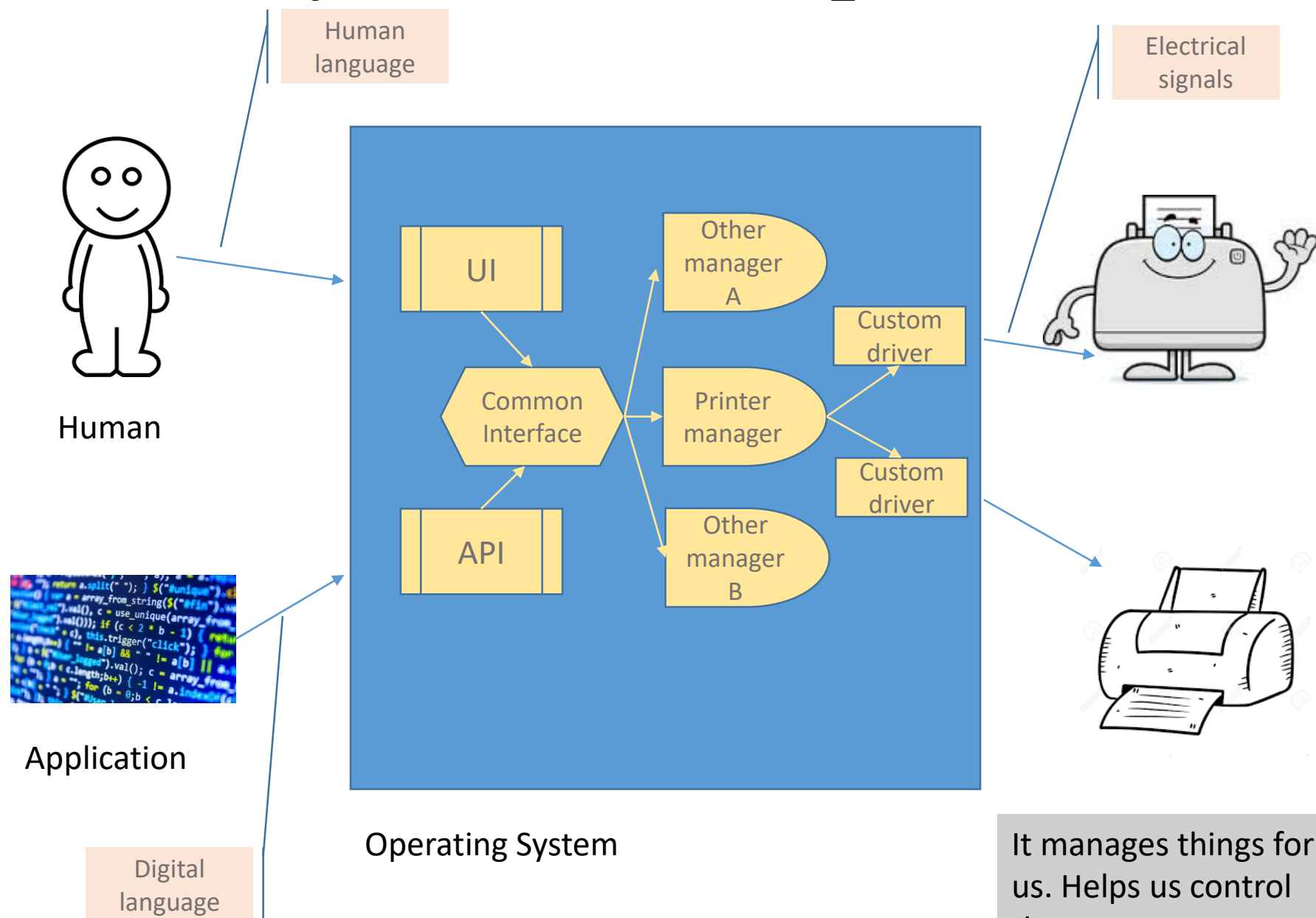




Unix  
Bash  
C  
GNU  
Systems

COMP 206 - Joseph Vybihal  
Software Systems

# Why is an OS important?



It manages things for us. Helps us control the computer.



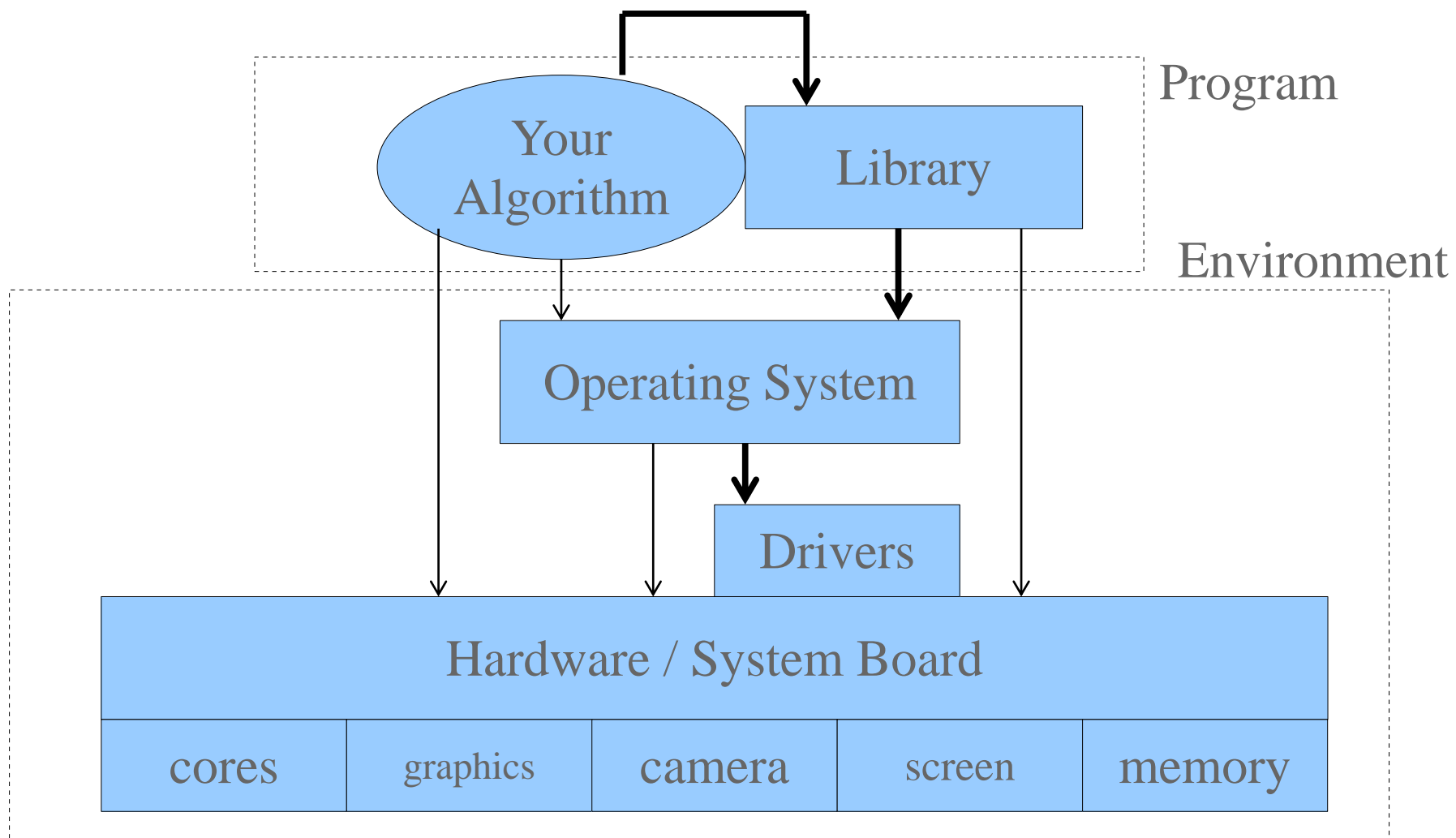
# What is an Operating System?

- An operating system is a piece of software that allows us to **interact** with a computer without having to know the inner working of a computer.
- Its primary function is to **manage** the computer's resources.
- An operating system also provides us with **libraries** to interact with these resources.



Unix  
Bash  
C  
GNU  
Systems

# From the POV of the application





Unix  
Bash  
C  
GNU  
Systems

# Why so many operating systems?

- DOS
- Windows
- Solaris
- Linux
- FreeBSD
- BeOS
- FreeDos
- HP-UX
- AIX
- MacOs X



# A little bit of history...

- The history of Unix begins in a failed operating system by AT&T Bell Laboratories called **Multics** .
- Ken Thompson who was working on this project, wrote a game called *Space Travel* .
- When the project was cancelled, he decided to port the game to the PDP-7 computer.
- He wrote Unix as an operating system to make it easier to port the game.



# Types of Unix

- System V UNIX : Operating Systems based on the original AT&T UNIX code fit in this category. These include most commercial UNIX distribution.
  - AIX, Iris, Solaris, UnixWare, etc.
- BSD UNIX : These Operating Systems are based on the Berkeley Software Distribution (BSD) version of UNIX.
  - FreeBSD, OpenBSD, NetBSD and MacOS X.
- UNIX-like systems : Several Operating Systems behave like UNIX, but are not based on the original AT&T code.
  - Linux, Hurd, Minix

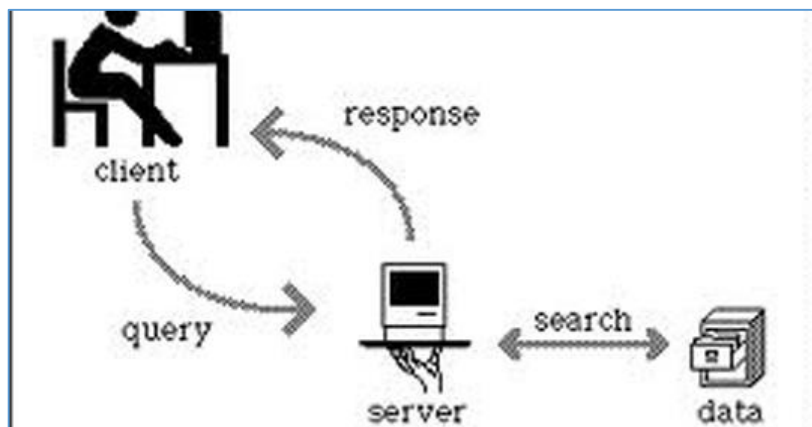


# Unix is...

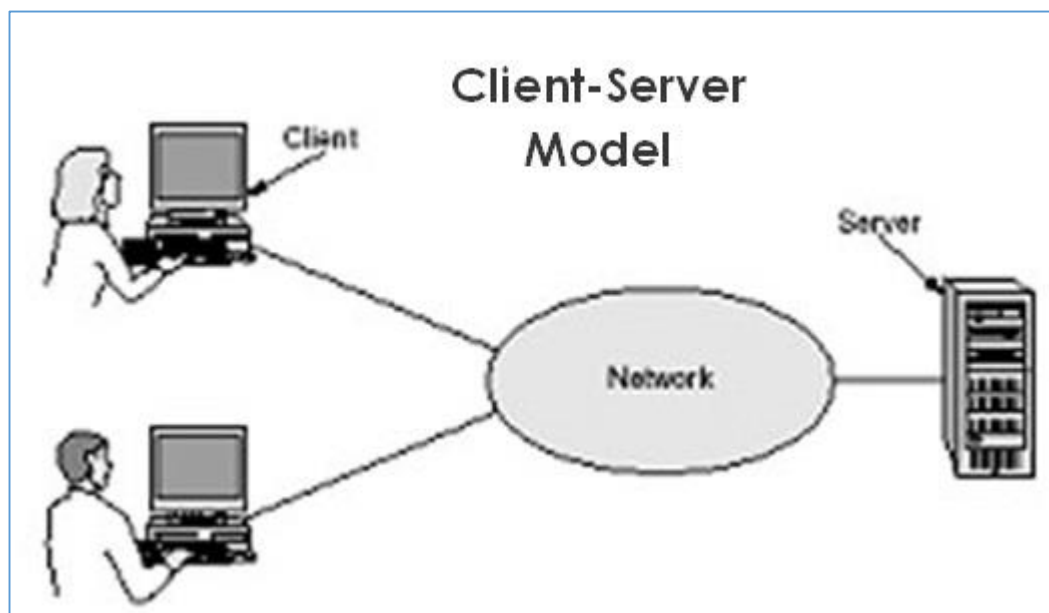
- Optimized and simple
- Password-based security
- Command-line driven
- Network capable
- Client-server architecture



# Client-Server Architecture



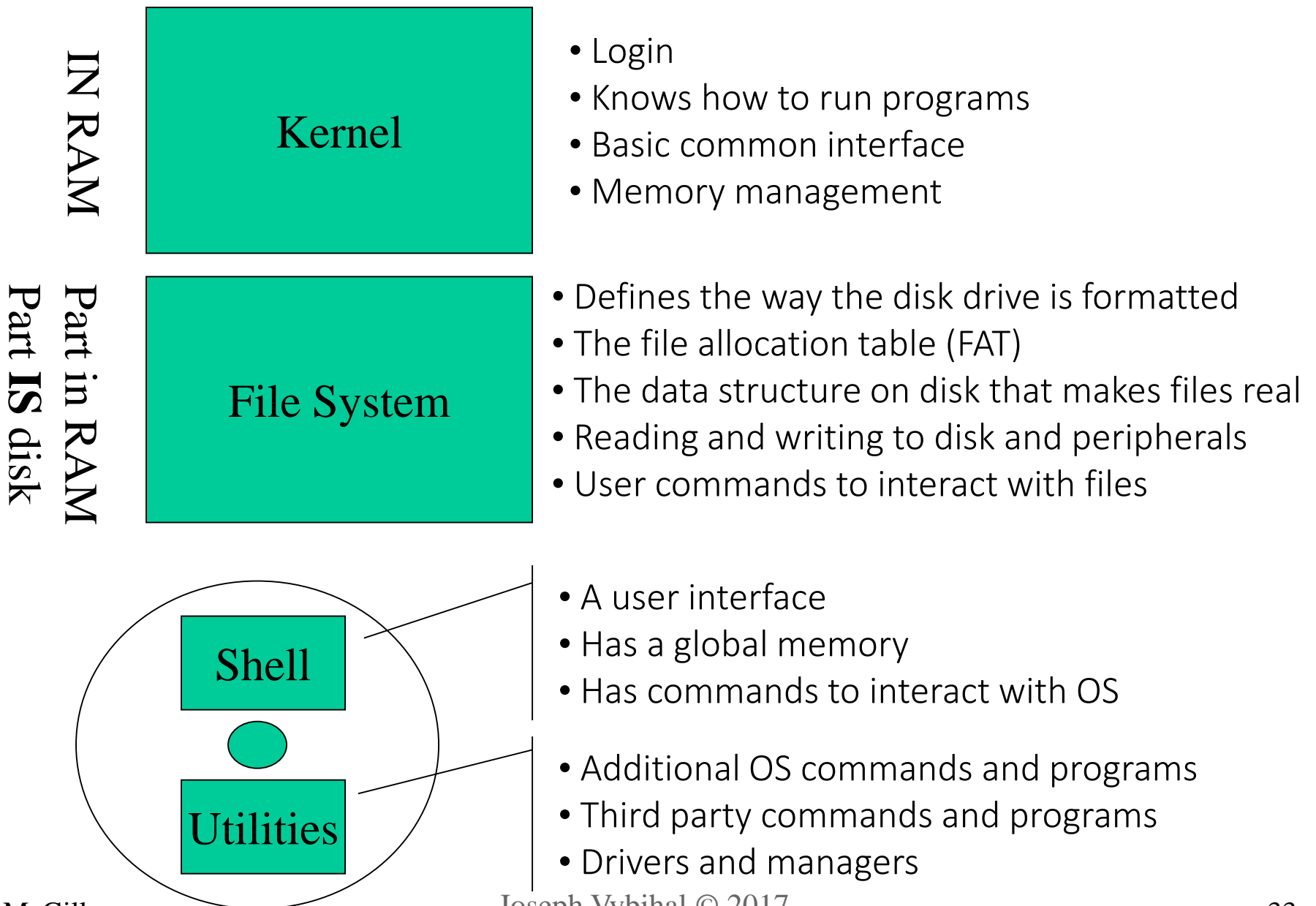
The model assumes that there is data or software on a server that needs to be shared with a remote computer (client).







# Unix OS Components





# SOCS Unix Server Access

- Labs

- Trottier 3<sup>rd</sup> floor (any unlocked room and hallway)
- Assistance:
  - [help@cs.mcgill.ca](mailto:help@cs.mcgill.ca)
  - Office: McConnell 209N
  - Phone: 514-398-7087

- Lab accounts

- **DO NOT** use your McGill account! [First.last@mail.mcgill.ca](mailto:First.last@mail.mcgill.ca)
- You must use a SOCS account
  - <https://newuser.cs.mcgill.ca>
  - You must be on campus for this to work or on a VPN
  - Forgot your Username and/or Password?
    - Reset it at <https://newpassword.cs.mcgill.ca/>

Get your  
account



Unix  
Bash  
C  
GNU  
Systems

COMP 206 - Joseph Vybihal  
Software Systems

# SOCS Server Architecture

Your assignments must  
run on linux.cs.mcgill.ca

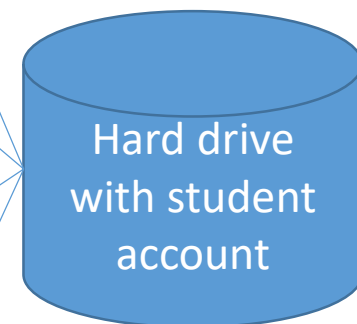


mimi.cs.mcgill.ca  
(server)

ubuntu.cs.mcgill.ca  
(server)

freebsd.cs.mcgill.ca  
(server)

linux.cs.mcgill.ca  
(server)

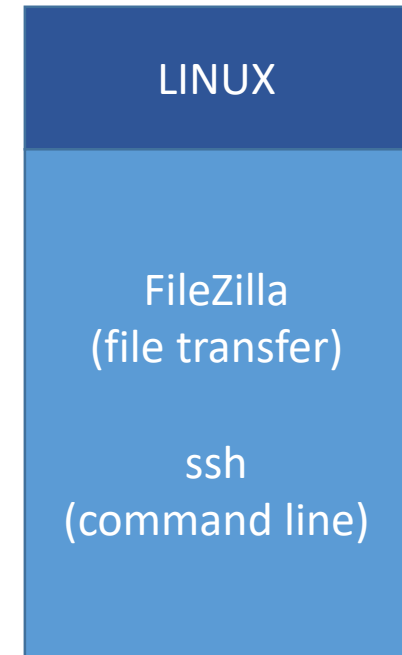
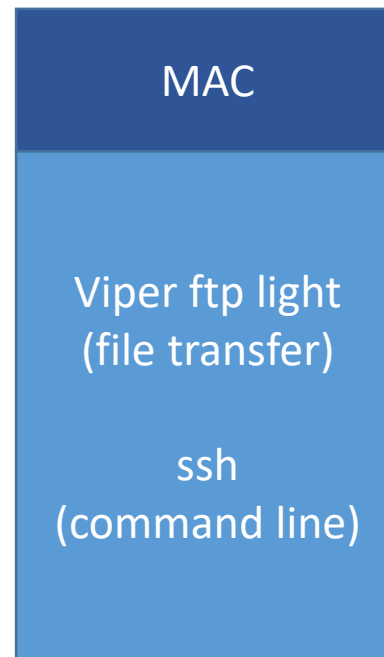
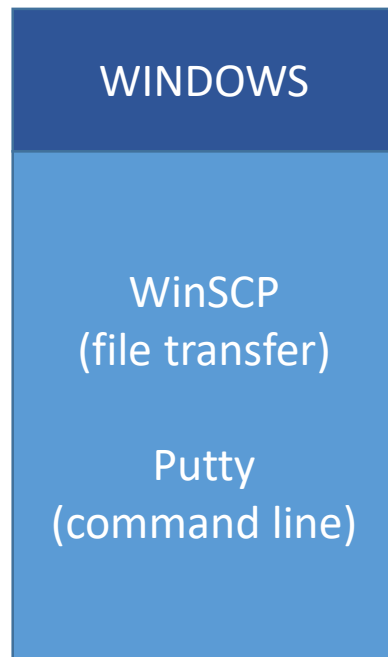


There are many SOCS  
web servers that  
connect to your data.  
IMPORTANT: each  
server is configured  
differently.

There is only  
one hard drive  
containing all  
your account  
information



# Remote Access



<https://filezilla-project.org/index.php>

Windows, Mac, and Linux

<https://winscp.net/eng/download.php>

Windows

<http://www.putty.org/>

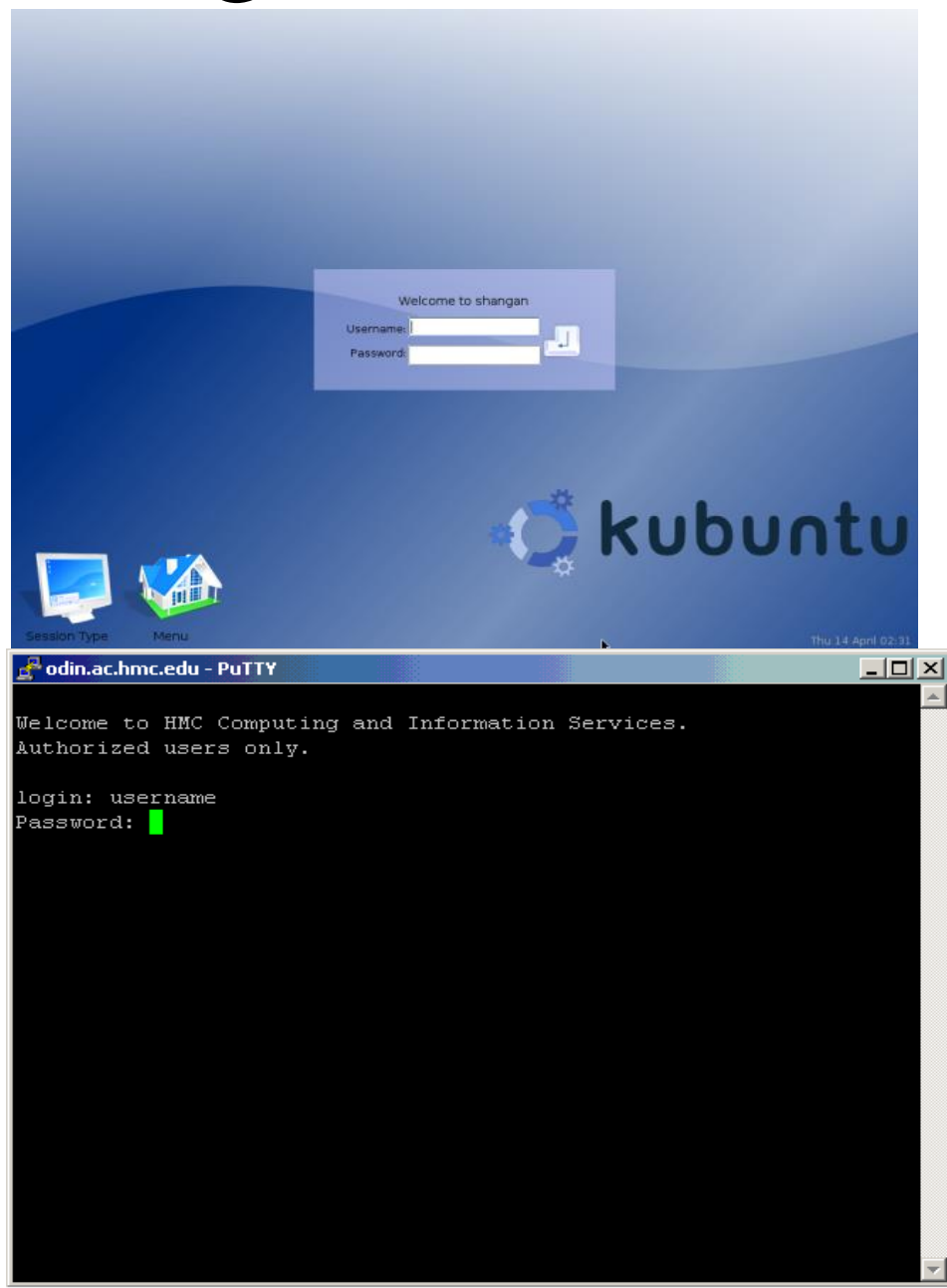
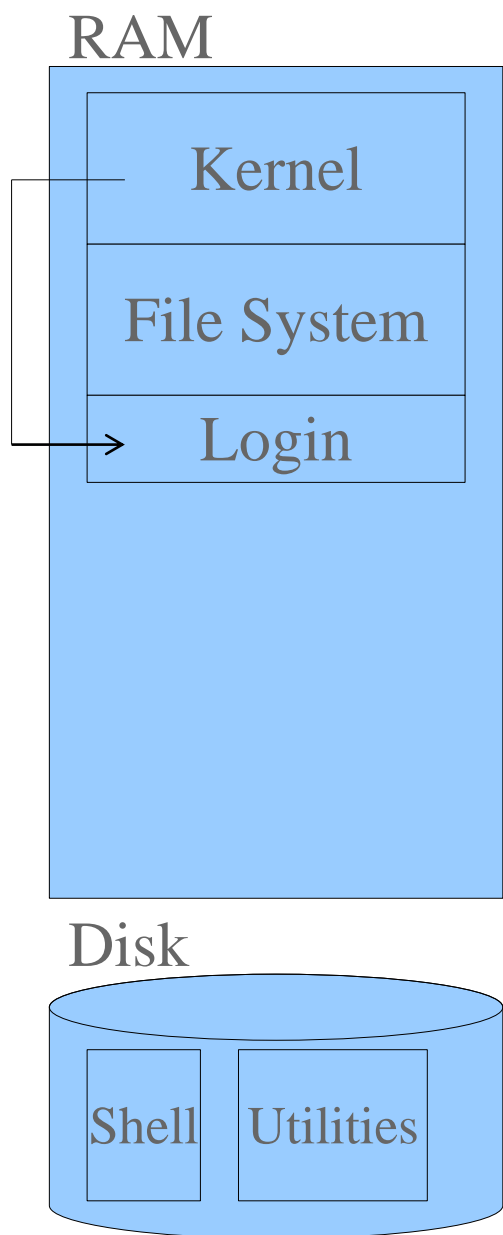
Windows



Unix  
Bash  
C  
GNU  
Systems

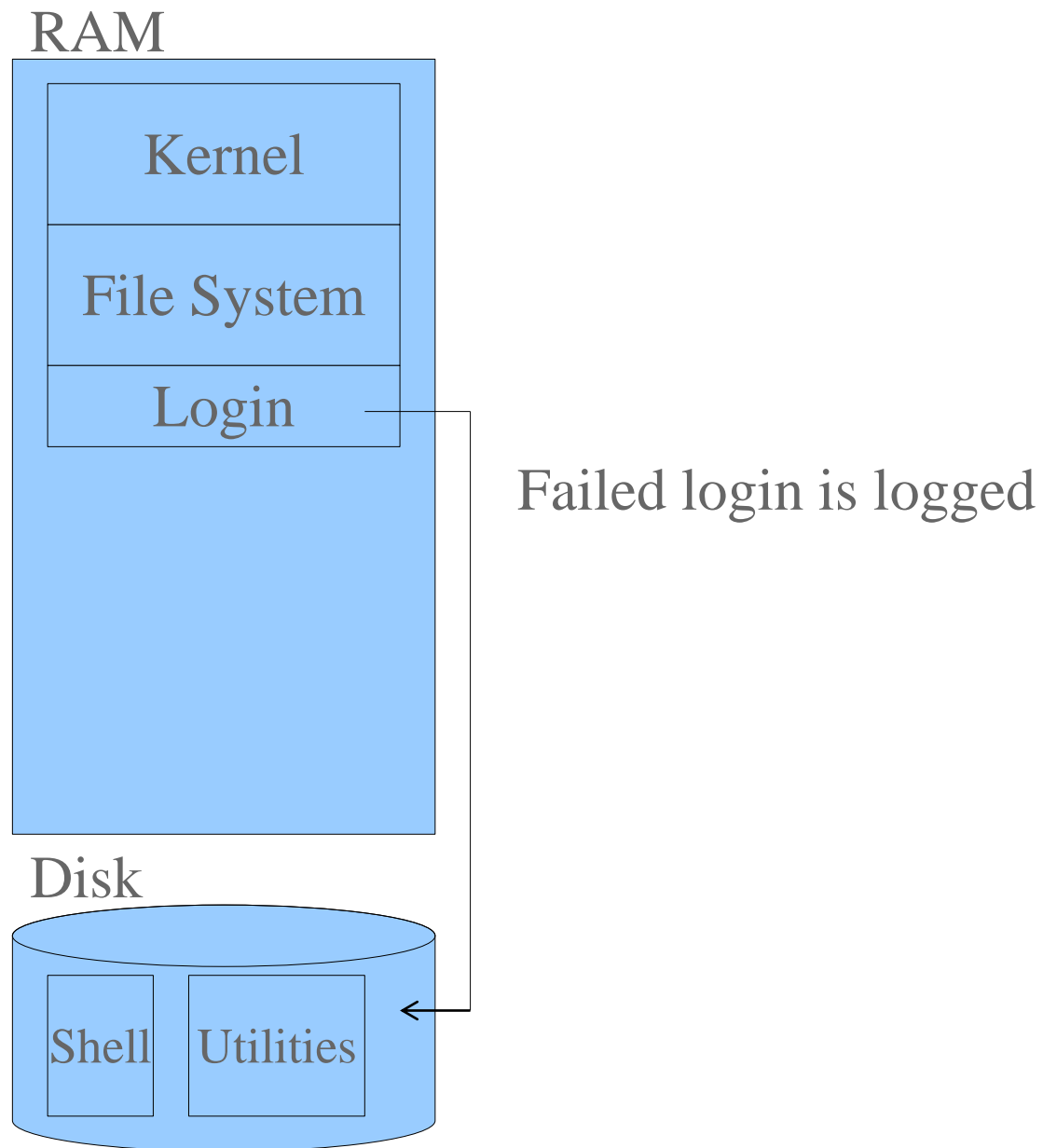
COMP 206 - Joseph Vybital  
Software Systems

# Login





# Login

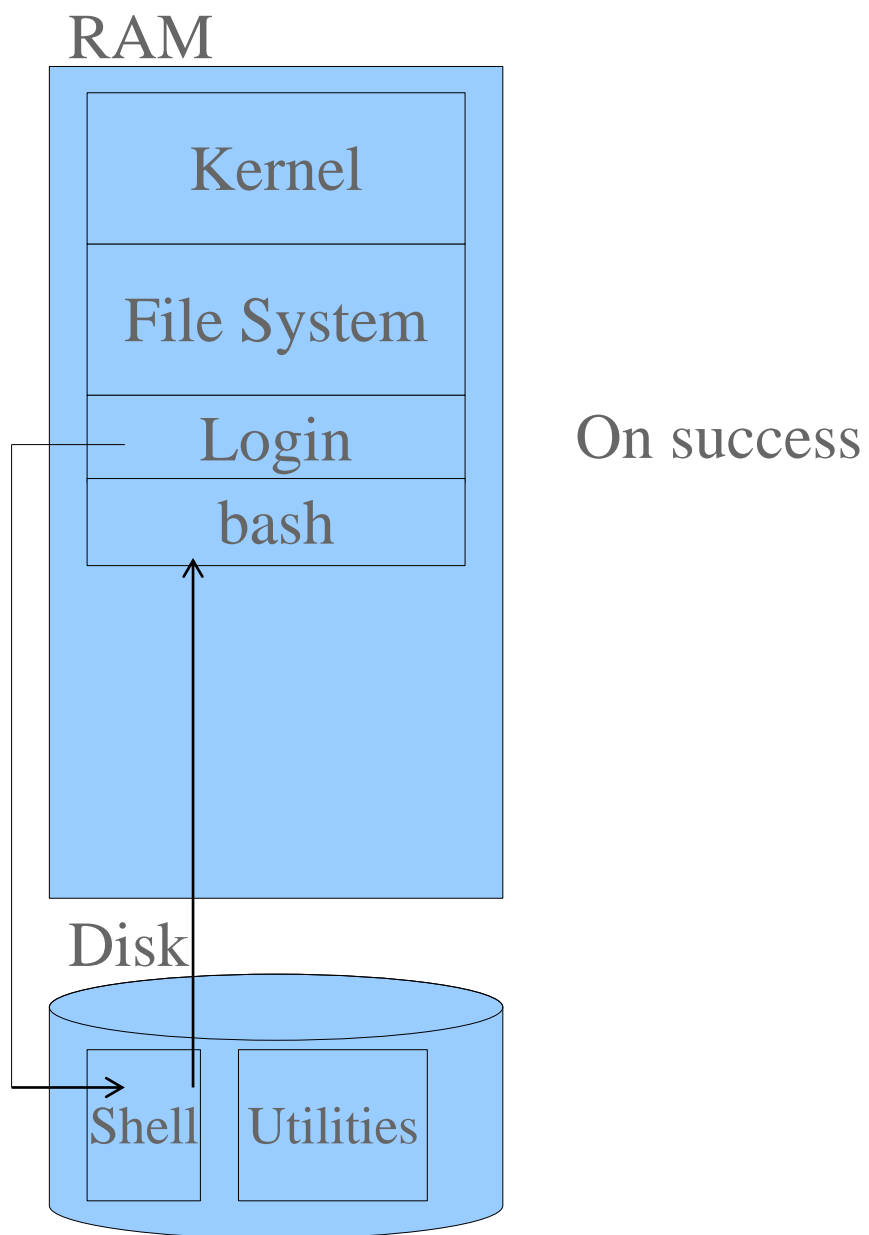




Unix  
Bash  
C  
GNU  
Systems

COMP 206 - Joseph Vybihal  
Software Systems

# Login

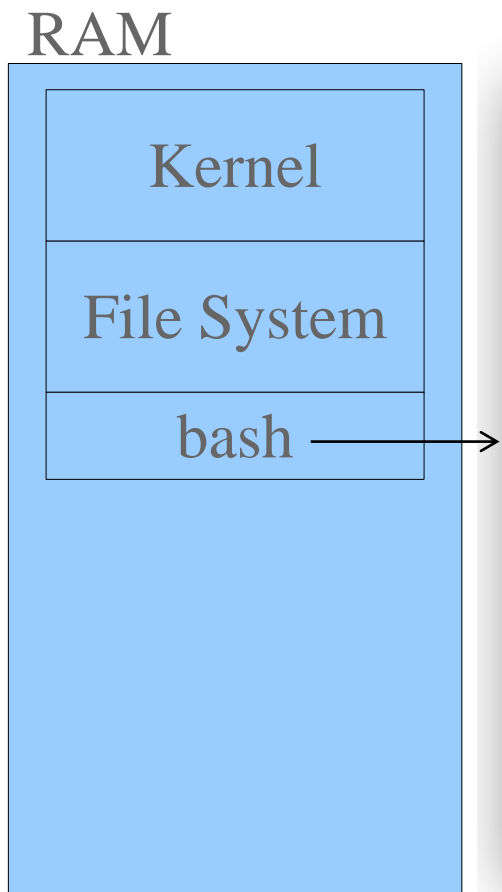




Unix  
Bash  
C  
GNU  
Systems

COMP 206 - Joseph Vybihal  
Software Systems

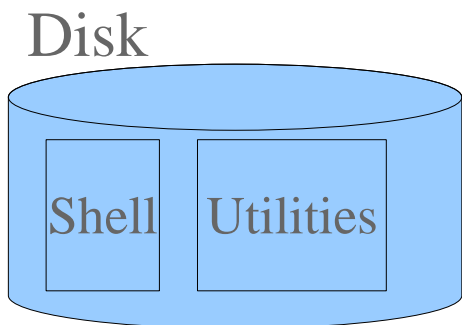
# Command Line Shell



Terminal — bash — 80x24

```
prana@[/Library/Application Support/Apple]
$ ls
total 0
drwxrwxr-x  9 root  admin  306 23 Jan 23:48 ./
drwxrwxr-x 16 root  admin  544 27 Jan 01:05 ../
drwxrwxr-x  3 root  admin  102 24 Sep 09:44 Automator/
drwxrwxr-x  3 root  admin  102 27 Sep 09:31 Grapher/
drwxrwxr-x  3 root  admin  102 29 Sep 12:51 Installer/
drwxrwxr-x  3 root  admin  102 10 Oct 13:14 Mail/
drwxrwxr-x  5 root  admin  170 10 Oct 13:09 ParentalControls/
drwxrwxr-x  3 root  admin  102 23 Jan 23:39 System Image Utility/
drwxrwxr-x 11 root  admin  374 23 Jan 23:59 iChat Icons/

prana@[/Library/Application Support/Apple]
$
```







# Demo & Procedures

- Logging in
  - `mimi.cs.mcgill.ca` or `linux.cs.mcgill.ca`
- File transfer
  - WinSCP file transfer example
  - Putty command-line example
- Basic working method
  - Do not use your computer to complete the assignments because they will not run the same way on the SOCS servers
  - Login to the command-line
  - Do your work
  - If you want to make a backup of your work then use the File Transfer software to copy your work to your laptop
  - Logout



# Be careful of...

- Each web server is installed with different libraries. This means that if you compile or create on one server it might not run the same on another server.
- Text file formats are not exactly the same between Windows, OS X, and Linux. If you develop on your laptop then transfer the file to the server, the file will probably not work correctly.



# Lecture 3

## The Shell

Readings: chapter 2, <https://www.tjhsst.edu/~dhyatt/superap/unixcmd.html>



# Outline

- The shell environment
  - Login
  - Home vs. Root directory
  - Paths
- The command line
  - Basic Commands



Unix  
Bash  
C  
GNU  
Systems

# Readings

- Chapter 1

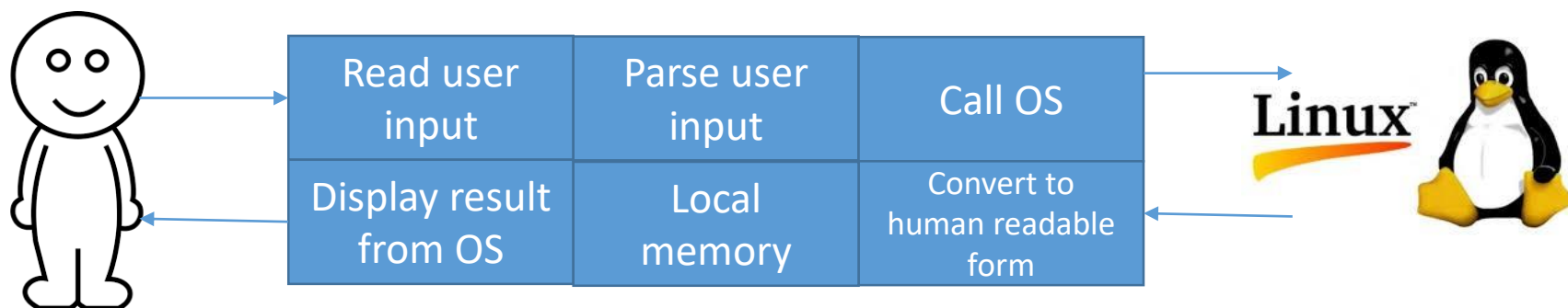


# What is a shell?

```
himanshu@himanshu:~$ ls -l
total 116
-rw-r--r-- 1 himanshu himanshu 41 Jul 3 23:23 anotherfile
drwxr-xr-x 3 himanshu himanshu 4096 Jul 3 14:26 Desktop
drwxr-xr-x 2 himanshu himanshu 4096 Mar 31 17:54 Documents
drwxr-xr-x 2 himanshu himanshu 4096 Jul 3 21:53 Downloads
-rw-r--r-- 1 himanshu himanshu 8980 Mar 31 17:32 examples.desktop
-rw-r--r-- 1 himanshu himanshu 22 Jul 4 14:42 file1
-rw-r--r-- 1 himanshu himanshu 22 Jul 4 14:42 file2
-rw-r--r-- 1 himanshu himanshu 179 Jul 4 14:38 ls-dump
drwxr-xr-x 3 himanshu himanshu 4096 May 10 20:00 Music
-rw-r--r-- 1 himanshu himanshu 0 Jul 3 21:48 newfil
drwxr-xr-x 2 himanshu himanshu 4096 Jul 4 15:56 Pictures
drwxr-xr-x 3 himanshu himanshu 4096 Jul 4 15:13 practice
-rw-r--r-- 1 himanshu himanshu 20480 Jul 4 15:09 practice.tar
-rw-r--r-- 1 himanshu himanshu 2703 Jul 4 15:19 practice.tar.gz
drwxr-xr-x 2 himanshu himanshu 4096 Mar 31 17:54 Public
-rw-r--r-- 1 himanshu himanshu 52 Jul 3 23:22 somefile
drwxr-xr-x 2 himanshu himanshu 4096 Mar 31 17:54 Templates
drwxr-xr-x 2 himanshu himanshu 4096 Jul 3 21:31 test
-rw-r--r-- 1 himanshu himanshu 149 Jul 3 21:32 test.tar.gz
-rw-r--r-- 1 himanshu himanshu 310 Jul 3 21:45 test.zip
-rw-r--r-- 1 himanshu himanshu 36 Jul 4 15:56 textfile
drwxr-xr-x 2 himanshu himanshu 4096 Apr 21 12:47 Ubuntu One
-rw-r--r-- 1 himanshu himanshu 101 Jul 4 14:30 uname-output
drwxr-xr-x 2 himanshu himanshu 4096 Mar 31 17:54 Videos
himanshu@himanshu:~$
```



A program that has three basics tasks: (a) get user input, (b) display OS information, (c) store session information.



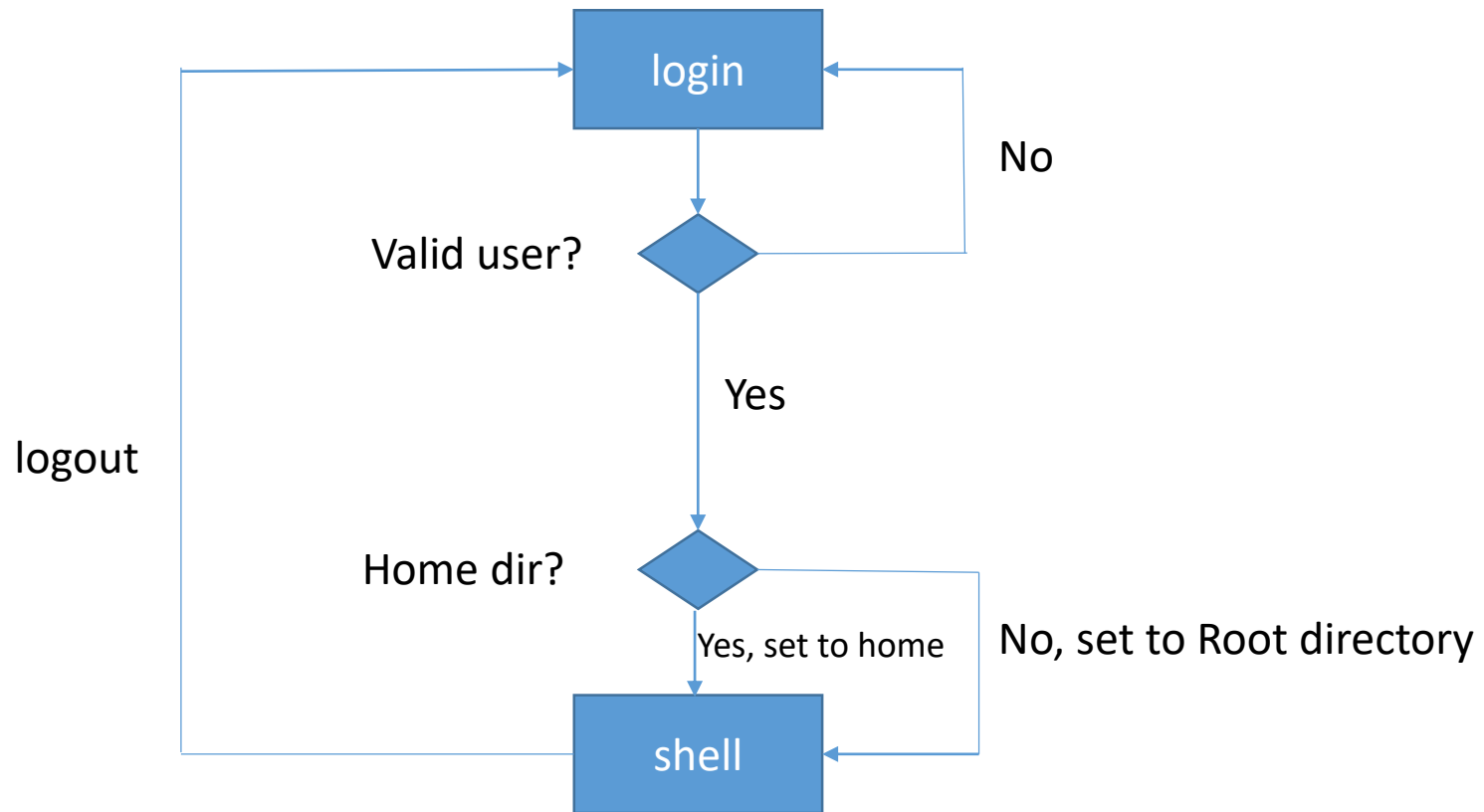


# Why so many shells?

- x Window
- Bash
- sh
- tsh
- csh
- tcsh
- cmd



# The shell environment

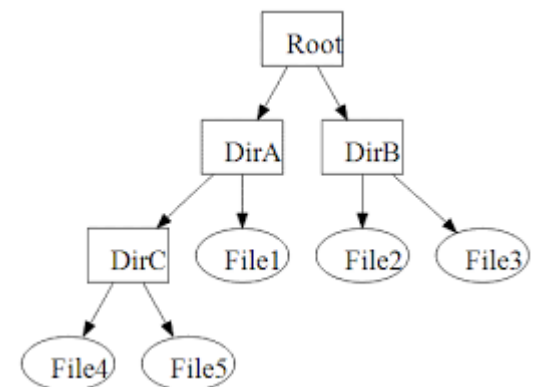


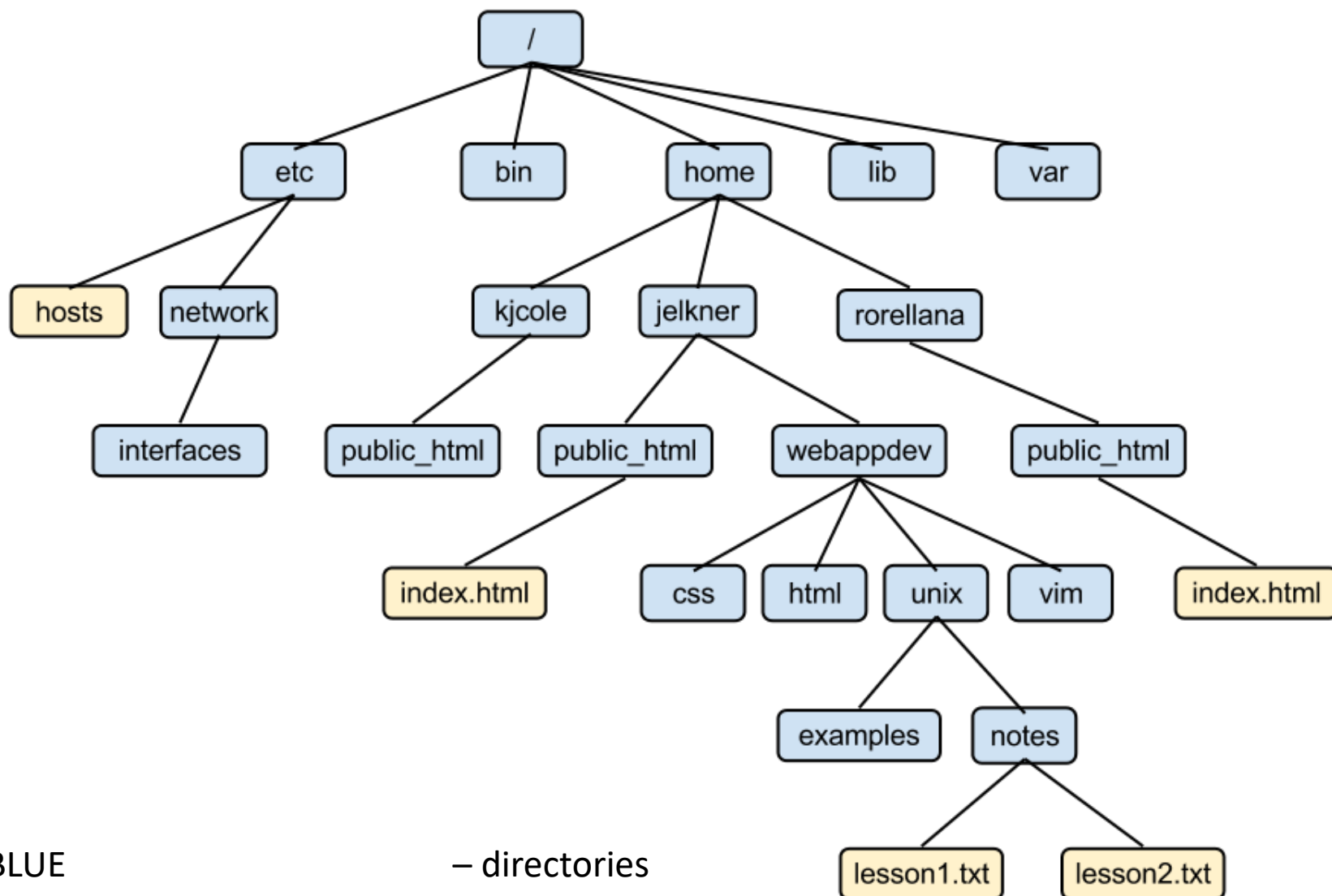




# Directory

- Linux “directory” is a folder
  - Synonyms: Dir, Folder, Directory
  - Def: An OS structure that contains files. This structure can be assigned a name.
- 4 special directories
  - Home
    - This is the top folder in the user’s directory tree
    - Special symbol: ~
  - Root
    - This is the top folder of the OS
    - Special symbol: /
  - Current
    - The directory you are currently within
    - Special symbol: .
  - Parent
    - The directory “above” the current directory
    - Special symbol: ..





BLUE

YELLOW

/

Kjcole, jelkner, rorellana

– directories

– files

- is the root

– examples of home directories



# Path

Def: A path is a string that describes the location of a file or directory within an OS.

Two ways of expressing a path:

## 1. Absolute path

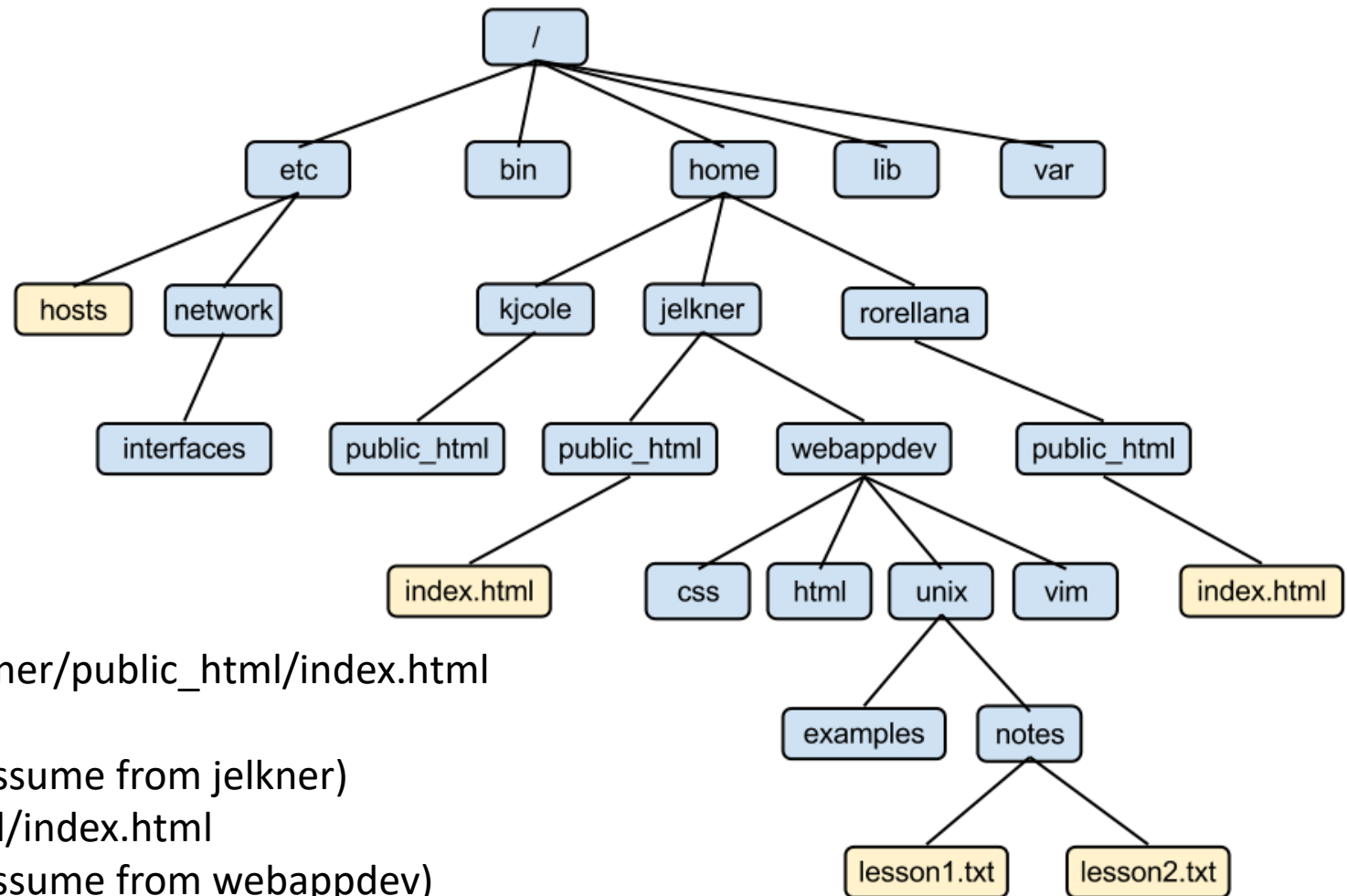
- Def: a string that begins at the Root
- Syntax: `"/dir/dir/dir/file"` or `"/dir/dir/dir"`

## 2. Relative path

- Def: a string that begins at the current location
- Syntax: `"dir/dir/file"` or `"../dir/dir/file"` or `"file"`



# Path examples



Absolute:

/home/jelkner/public\_html/index.html

Relative: (assume from jelkner)

public\_html/index.html

Relative: (assume from webappdev)

../public\_html/index.html

Relative: (assume from vim)

../../public\_html/index.html



# The command line

prompt →

```
Terminal — bash — 80x24
bash
prana@[Library/Application Support/Apple]
$ ls
total 0
drwxrwxr-x  9 root  admin  306 23 Jan 23:48 ./
drwxrwxr-x 16 root  admin  544 27 Jan 01:05 ../
drwxrwxr-x  3 root  admin  102 24 Sep 09:44 Automator/
drwxrwxr-x  3 root  admin  102 27 Sep 09:31 Grapher/
drwxrwxr-x  3 root  admin  102 29 Sep 12:51 Installer/
drwxrwxr-x  3 root  admin  102 10 Oct 13:14 Mail/
drwxrwxr-x  5 root  admin  170 10 Oct 13:09 ParentalControls/
drwxrwxr-x  3 root  admin  102 23 Jan 23:39 System Image Utility/
drwxrwxr-x 11 root  admin  374 23 Jan 23:59 iChat Icons/

prana@[Library/Application Support/Apple]
$ █
```

## Command Format:

Program switches arguments

## Example Syntax:

\$ ls -l ass1.pdf

## Where:

- Program - the command
- Switches - modifies behavior of command
- Arguments - input passed to the command



# The directory listing

```
[mimi] [~] ls
SummerCampEng.flv  public_html  public_html.2005  summ.tgz
[mimi] [~]
```

```
$ ls -l /bin/ar
-r-xr-xr-x  1 bin      bin      21428 Sep 24  1983 /bin/ar
$
```

```
[mimi] [~] ls -l
total 3092
-rw-----  1 summcamp 16618 1376256 Jun  5 10:44 SummerCampEng.flv
drwxr-xr-x 11 summcamp 16618   4096 Jun  7 11:41 public_html
drwxr-xr-x  3 summcamp 16618   4096 Jan  4 2006 public_html.2005
-rwx-----  1 summcamp 16618 1780739 Jul  5 2005 summ.tgz
[mimi] [~]
```

Annotations for the `ls -l` output:

- permissions**: points to the first column (e.g., `-rw-----`)
- links**: points to the second column (e.g., `1`)
- owner**: points to the third column (e.g., `summcamp`)
- group**: points to the fourth column (e.g., `16618`)
- size**: points to the fifth column (e.g., `1376256`)
- Modification date**: points to the sixth and seventh columns (e.g., `Jun 5 10:44`)

What does everything mean?



# Directory Content

- See the contents of a directory (folder)

`ls`

- See contents in long format

`ls -l`

- See a particular file in long format

`ls -l letter.doc`

- Display all the files and hidden files

`ls -a`



# Current Directory

- `pwd` (print working directory)

displays the current directory you are in (considered default)

Syntax : `pwd`

Usage : `pwd`

Result : Displays to the screen `/home/jack` (for example)

- `cd` (change directory)

allows a user to change their current directory.

Syntax : `cd PATH`

Usage : `cd ../home`

Result : current working directory is now `../home`





```
$ whoami
Jvybihal
$ pwd
/jack
$ ls
Stuff.txt  source
$ cd source
$ ls
F1.java f2.java f3.java
$ cd ..
$ ls
Stuff.txt  source
$ cd /
I am at the root now
$ cd/jack/source
$ mkdir docs
$ ls
F1.java f2.java f3.java docs
```



# Directory Manipulation

- `cd [directory]`
  - change directory
- `ls [options] [directory or file list]`
  - directory contents or file permissions
- `mkdir [options] directory`
  - make a directory
- `pwd`
  - print working (current) directory
- `rmdir [options] directory`
  - remove a directory

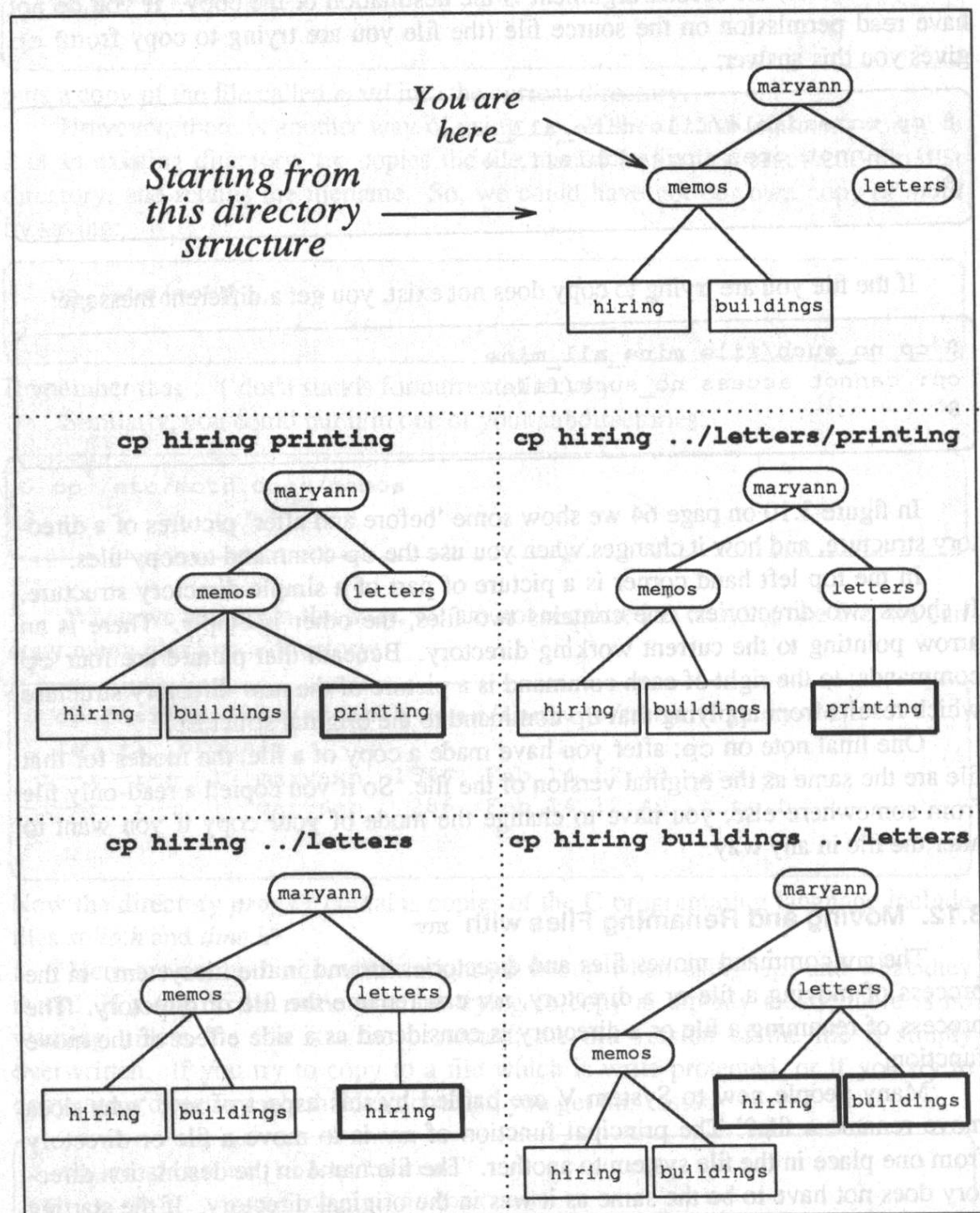


# File Manipulation

- `cp [options] file1 file2`
  - copy file1 into file2. This command creates or overwrites file2.
- `mv [options] file1 file2`
  - move file1 into file2
- `rm [options] file`
  - remove (delete) a file or directory



# Copy Command



\$ cp from to

From = filename

From = path/filename

To = From syntax

To = path

\$ cp from from to



# Options for cp, mv and rm

```
$ cp -r /jack /mary/source
```

- -i : interactive (cp, mv and rm)
  - prompt and wait for confirmation before proceeding
- -r or -R: recursive (cp, rm)
  - recursively visits a directory, first visiting the files and subdirectories beneath it
- -f : force : (mv, rm)
  - don't prompt for confirmation (overrides -i)



# Cat and More

- `cat [options] files`
  - file concatenate and display the concatenated result
- `more [options] file`
  - page through a text file



# Logout vs. Exit

- The *logout* command closes the shell and logs you out
- The *exit* command closes the shell (however, if there is no other shell to send you to then it will automatically call *logout*).



# Linux Manual

- The `man` command allows you to access the on-line manual pages of the various commands available on the shell.
  - These pages are often referred to as “man pages”.
- The man pages are your first source of information when working in the shell.
- To access a man page, simply type `man` and the name of the command at the prompt.

```
man ls
```





## \$ man date

date(1)

date(1)

Version 1.1

### NAME

date - print and set the date

### SYNOPSIS

date [yyymmddhhmm[.ss]]

### DESCRIPTION

If no argument is given, the current date and time are printed.

If an argument is given, the current date is set. `yy` is the last two digits of the year; the first `mm` is the month number; `dd` is the day number in the month; `hh` is the hour number (24 hour system); the second `mm` is the minute number; `ss` is optional and is the seconds. For example:

```
date 10080045
```

sets the date to Oct 8, 12:45 AM. The year, month and day may be omitted, the current values being the defaults. The system operates in GMT. `Date` takes care of the conversion to and from local standard and daylight time.

### FILES

/usr/adm/wtmp to record time setting

### SEE ALSO

utmp(5)

### DIAGNOSTICS

'No permission' if you aren't the super-user and you try to change the date; 'bad conversion' if the date set is syntactically incorrect.

.....

\$

\$ man date