# COSC 3360/6310 Wednesday, January 20

## Welcome to COSC 3360!

Spring 2021 Edition J.-F. Pâris

#### ZITS

#### BY JERRY SCOTT AND JIM BORGMAN

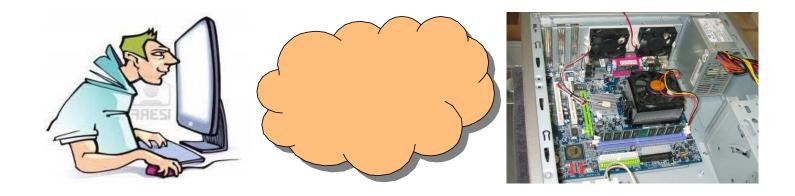


# Why should you study OS?



## What is an operating system?

"What stands between the user and the bare machine"





## What is an operating system?

- The *basic* software required to operate a computer.
- Similar role to that of the conductor of an orchestra



### Functions of an OS

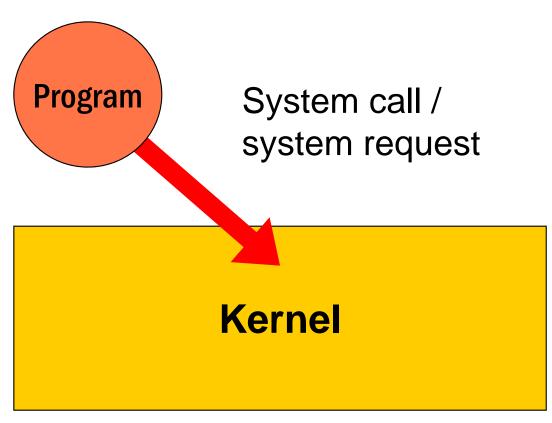
- Four basic functions
  - To provide a better user interface
  - □ To manage the system resources
  - □ To protect users' programs and data
  - To let programs exchange information



### The core of the OS

- Part that remains in main memory
- Controls the execution of all other programs.
- Known as the kernel
  - Also called monitor, supervisor, executive
- Other programs interact with it through system calls







## Understanding the system call interface

- Essential to the advanced programmer
  - □ To create files with the right protection attributes
  - □ To write multithreaded programs
  - □ To synchronize their threads
  - □ To write distributed applications

# One exciting development



#### Main memory

- ☐ Cache and DRAM
- Byte-addressable
- Volatile

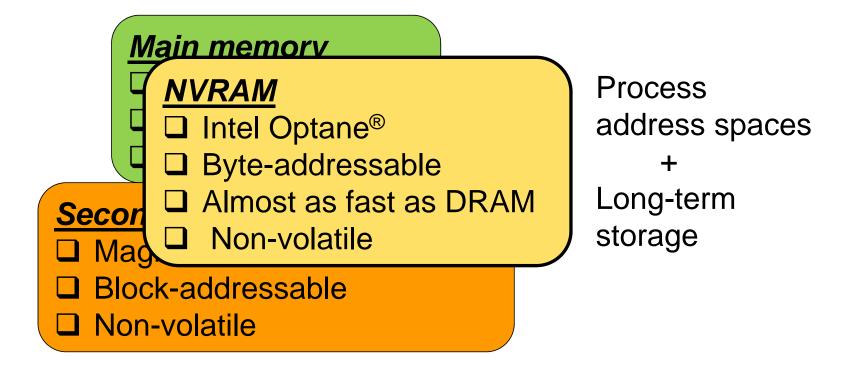
Process address spaces

### Secondary store

- Magnetic disks and now SSD
- Block-addressable
- Non-volatile

Long-term storage







## What to expect

- NVRAM breaks the boundaries between main memory and secondary store
- Challenges the traditional file system interface
  - □ POSIX
  - □ Based on system calls

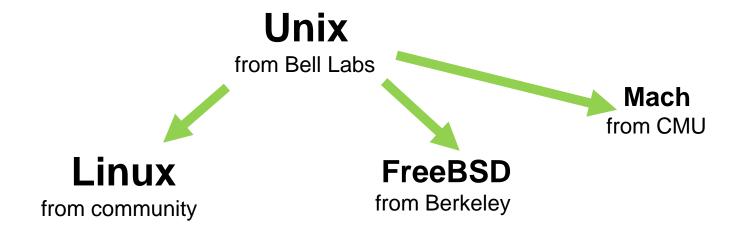
# A two-pronged approach



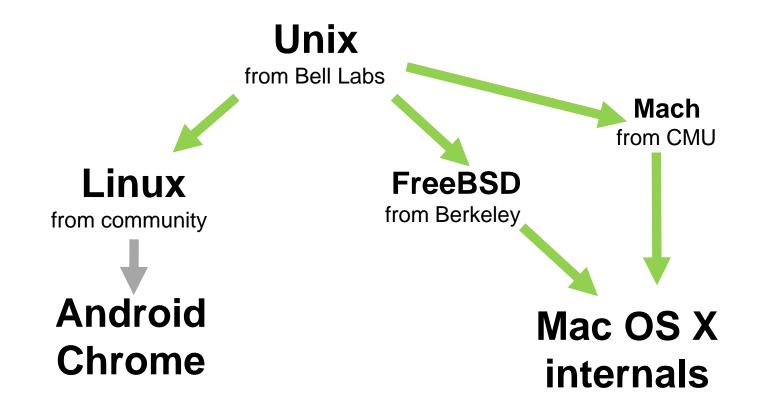
- OS overall organization
- □ How they manage processes/lightweight processes
- □ How they share cores among processes
- □ How they let processes communicate with each other
- □ How they can synchronize cooperating processes
- □ Why we have virtual memory
- □ How the hardware OS manages the virtual memory
- All about file systems

## Introducing new friends

The Unix family of OSes



## And their offspring





## The practice

- Three programming assignments
  - □ To be written in **C/C++**
  - □ In a Linux/Free BSD environment
- First assignment about process management
  - No special system calls
  - □ Will test your knowledge of linked lists
- Second assignment about inter-process communication
- Third assignment about thread synchronization



### For C/C++ newbies

- Several of you have taken COSC 2430 in Java
  - □ Do not panic
  - □ Let me know so I will let you use Java for the first assignment
  - □ You will still have to learn C/C++ for the last two assignments.
    - System programming assignments with minimal coding
- Some of you have taken COSC 2430 in Python
  - □ Ouch!

# Your teaching team



### The instructor

- Jehan-François Paris
  - □ Jehan-François is French for Juan-Francisco
- jfparis AT uh DOT edu
- MW 5:30-6:30 pm and F 10:00-10:30 am on MS Teams



### The TAs

- Christos Smailis
  - csmailis AT uh DOT edu
  - ☐ MW 2:00-4:00 PM on MS Teams
- Pavan Kumar Paluri
  - pvpaluri AT uh DOT edu
  - □ TuTh 11:00 AM-12:00 PM on MS Teams
- Aftab Husain
  - ahussain27 AT uh DOT edu
  - □ Th 5:00-7:30 PM on MS Teams



## Getting in touch

- Primarily through Prulu
  - □ Free online course discussion tool
  - FERPA compliant
- Let me know if you did not receive an invitation
- Better than email
- Will host some course materials
  - □ Lecture notes replicated on "old" UH web site







#### Blackboard

- □ Online quizzes
- Assignment submission

#### Microsoft Teams

- □ Lectures (recordings on MS Streams)
- □ Office hours
- □ Lecture notes and other postings

#### Prulu

- □ Q& A
- □ Backup for lecture notes and other postings

Check your UH account + your spam folder



## Class attendance policy

- Recommended but *not* required
  - □ All video lectures will be recorded
- Required for the five quizzes
  - □ Unless excused absence or other extenuating circumstances
- If you decide to attend the lectures, please consider
  - □ Turning on your camera
  - □ Adding your picture to your MS Teams account



## Resources



## Study materials

- Course slides
  - □ What you should study
  - On MS Teams, some on Prulu
- Textbook
  - □ Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, Operating Systems: Three Easy Pieces
    - http://pages.cs.wisc.edu/~remzi/OSTEP
  - □ Another presentation of most of what we cover in class



## Computing resources

- Your laptop
  - □ For Mac users
    - Bash shell is all you need
  - □ For Windows users
    - If you have 64-bit version of Windows 10
      - Should install Windows Subsystem for Linux
    - Otherwise
      - □ Install *Cygwin*
      - □ Use the class account for second and third assignments



## For PC users: Installing WSL

- Excellent Microsoft tutorial
  - □ <a href="https://docs.microsoft.com/en-us/windows/wsl/install-win10">https://docs.microsoft.com/en-us/windows/wsl/install-win10</a>
- Two step process
  - Open PowerShell as Administrator and run:

Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

- □ Go to Microsoft App store
  - Search For "Linux"
  - Pick your version of Linux, such as Ubuntu 18.04 LTS



### For Mac users

- Few subtle differences between the ways Linux and Mac OS X handles some system calls
  - □ Rare
  - □ Will not affect your grades

# Evaluating your performance



## The quizzes

- Five quizzes
  - □ 60 percent of course average
  - On Blackboard
    - Require Respondus browser and camera
  - □ Non-comprehensive
    - Covers all materials discussed in class since previous quiz
  - Open book but time-limited



## The programming assignments

- Three programming assignments
  - □ 40 percent of semester average
  - ☐ Must be done in C/C++ in a Linux/Free BSD environment
  - □ Must be submitted through your UH *Blackboard* account
- People failing the assignments <u>or</u> the examinations will fail the course.



## The grace days

- You have three grace days
  - □ Late penalty waivers
- Assume you turn one assignment five days late
  - ☐ Still had your three grace days
  - $\square$  Late penalty will be (5-3)=2 late days instead of 5
  - □ Lose your grace days
- Can be used at your own discretion
  - □ No need to ask for my permission



### Warnings

- Start early
  - □ ...
- Do not use MS Visual C++ IDE
  - □ Incompatible with Linux/Free BSD gnu compilers
- Ask for help before it is too late
  - □ Programming problems, personal problems, ...
- Do not cheat
  - □ What you turn in must be your own code



#### Important dates

Monday, February 8 First quiz

Wednesday, February 24 First assignment due

Monday, March 1 Second quiz

Late March
Second assignment due

Monday, March 29
Third quiz

Monday, April 19
Fourth quiz

Monday, May 3
Last lecture. Third assignment due

Friday, May 7
Fifth quiz at 5 pm



#### The fine print

- Details of all course policies are to be found in the course syllabus
  - □ Posted online
- They may change
  - □ Changes will be announced in class, on MS Teams and on Prulu ahead of time

## Listen to the cow

# Chapter I Introduction

Jehan-François Pâris jfparis@uh.edu



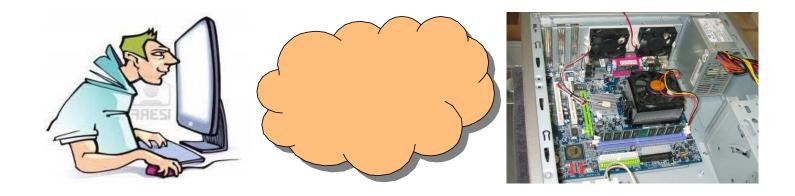
### **Chapter Overview**

- Defining operating systems
- Major functions of an OS
- Types of operating systems
- UNIX
- Kernel organization



### What is an operating system?

"What stands between the user and the bare machine"





### What is an operating system?

- The *basic* software required to operate a computer.
- Similar role to that of the conductor of an orchestra



#### Do not belong to OS

- All user programs
- Compilers, spreadsheets, word processors, and so forth
- Most utility programs
  - mkdir is a user program calling mkdir()
- The command language interpreter
  - Anyone can write his/her UNIX shell



#### The UNIX shells

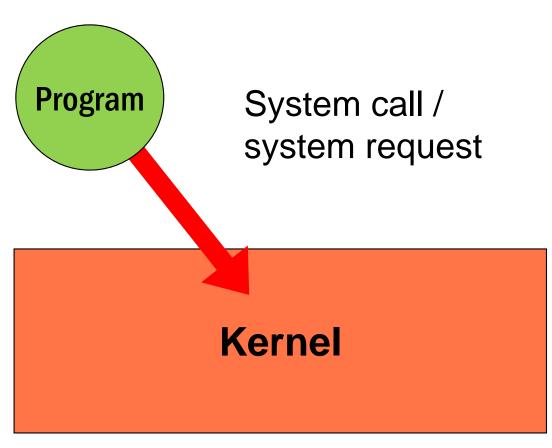
- UNIX has several shells
  - □ *sh* (the Bourne shell) is the original UNIX shell
  - □ *csh* was developed at Berkeley by Bill Joy
  - □ ksh (the Korn shell) was developed by David Korn at AT&T Bell Laboratories
  - □ **bash** (the GNU Bourne-Again shell ) and the list is far from complete



#### The core of the OS

- Part that remains in main memory
- Controls the execution of all other programs.
- Known as the kernel
  - ☐ Also called *monitor*, *supervisor*, *executive*
- Other programs interact with it through system calls

### System calls





### A question

Who among you has already used system calls?



#### The answer

- All of you
  - □ All I/O operations are performed through system calls

## The four missions



#### Functions of an OS

- Four basic functions
  - To provide a better user interface
  - To manage the system resources
  - To protect users' programs and data
  - To let programs exchange information



#### A better user interface

- Accessing directly the hardware would be very cumbersome
- Must enter manually the code required to read into main memory each program
  - □ boot strapping

### How it was done (I)



#### <u>PDP 8</u>

- Early 70's
- 12-bit machine
  - □ 4K RAM!

### How it was done (II)



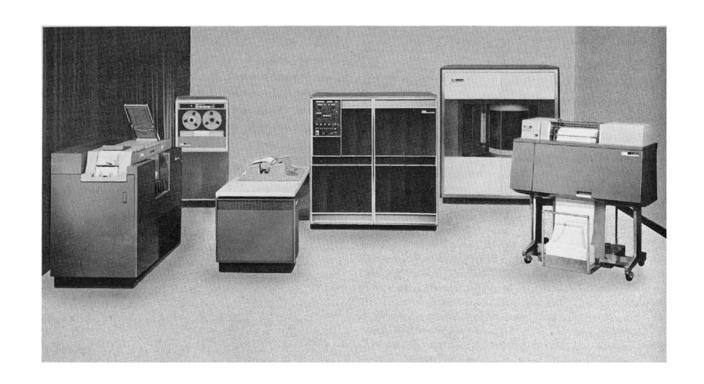
Toggle switches in front panel were used to enter the bootstrap code



#### Batch systems

- Allow users to submit a batches of requests to be processed in sequence
- Include a command language specifying what to do with the inputs
  - Compile
  - □ Link edit
  - □ Execute and so forth

#### An IBM 1401





#### Interactive systems

- Came later
- Allow users to interact with the OS through their terminals:
- Include an *interactive* command language
  - UNIX shells, Windows PowerShell
  - □ Can also be used to write scripts



#### Time sharing

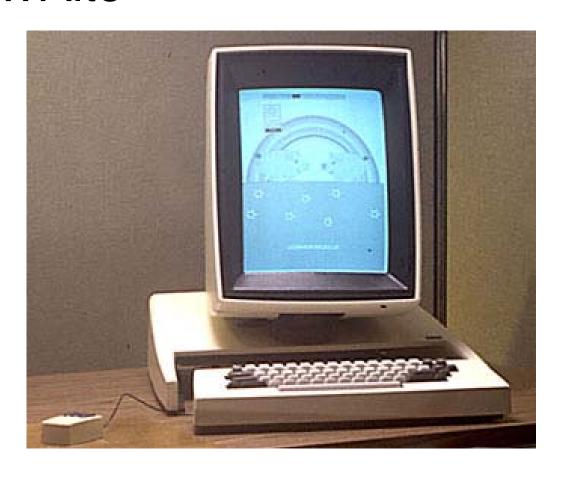
- Lets several interactive users to access a single computer at the same time
- Standard solution when computers were expensive



#### Graphical user interfaces

- Called GUIs (pronounced goo-eys):
   Macintosh, Windows, X-Windows, Linux
  - □ Require a dedicated computer for each user
  - □ Pioneered at Xerox Palo Alto Research Center (Xerox PARC)
  - Popularized by the Macintosh
  - □ Dominated the market with MS Windows

#### The Xerox Alto





#### Xerox PARC (I)

- Founded by XEROX in 1970
- Invented
  - Laser printing
  - Ethernet
  - □ The GUI paradigm
  - Object-oriented programming (Smalltalk)



### Xerox PARC (II)

- All their inventions were brought to market by other concerns
- Popular belief is that Xerox management blew it
- In reality
  - □ Alto workstations were very expensive
  - □ Smalltalk was very slow
  - ☐ Group was too small to deliver a full system



#### Smart phones and tablets

- Convergence of four trends
  - □ Cheaper LCD displays
  - □ Solid-State Storage (SSD)
  - ☐ Faster wireless communications
  - Ubiquitous wireless



#### History repeats itself

- First successful devices introduced by Apple
  - □iPod, iPhone, iPad, ...
    - First iPad was underpowered
- Competition soon grows
  - □ Cheaper Android devices



#### With a difference!

- Apple did not "steal" the concept from anyone
- iPods, iPhones, iPads were an instant success
  - Reasonably priced



#### Two models

- Apple:
  - □ Closed ecosystem (*walled garden*)
  - □ Strict controls on app market
  - Missing features
    - No file system

- Android:
  - □ Just the opposite
  - □ Lax controls on app market
  - □Can access the Linux/Android shell

# Is this paradise?





### Summary

- Six major steps
  - ☐ Bare bone machine
  - ☐ Batch systems
  - □Timesharing
  - □ Personal computer
  - □ Personal computer with GUI
  - □ Smart phone/tablet



#### File systems

- Let users create and delete files without having to worry about disk allocation
  - □ Users lose the ability to specify how their files are stored on the disk
  - □ Database designers prefer to bypass the file system
- Some file systems tolerate disk failures (RAID)