Operating System Design

Introduction
System Calls
Trap

Neda Nasiriani
Fall 2018

Class Dynamics

- We meet 3 times per week MWF at 2:00PM
- Office hours: Wednesday 3:00-4:30 or by appointment. Walk-ins are welcome if my office door is open
- Email: nn005@bucknell.edu
- Office: Breakiron 370
- Attendance is expected. In case of legitimate reasons (sickness, interview, …) let me know before the class
- The lectures will be student-oriented and include many different **activities**, **quizzes** (mostly from reading assignment) and **discussions**, so please come to the class prepared and do the **reading before the class**

Class Dynamics (continued)

- Collaboration of any kind (from web, or a friend, …) for individual assignments is not permitted. Although high level questions can be discussed as long as the name of the person is listed on your delivered assignment
- Course requires about 15 hours of outside classroom preparation, reading (learning actively), designing, coding and documenting
- If you have any questions or suggestions to make the learning experience better for you, please do not hesitate to share. Suggestions are appreciated
- In your emails about this course please include the course number (CSCI 315) in the subject.

Class Dynamics (continued)

- There will be two one hour midterm exams and one final exam as scheduled by the registrar. Details of the exams will be announced one week before each exam.
- The grade components
    - Quizzes: 5%
    - Activities: 5%
    - Professionalism: 5%
        - Attending the class
        - Engaging in the activities
        - Showing enthusiasm for learning
        - …
    - Exam 1: 15%
    - Exam 2: 15%
    - Final Exams: 15%
    - Labs (and other assignments): 40%

Class Dynamics (continued)

- Note that you need to get a passing grade for each category to be able to pass the course
- Labs are very important part of this course (about 40% of your grade) and they can not be made up for after the deadline
- There are usually prelab assignments that need to be completed before the lab session (There is one prelab for tomorrow, so don’t forget to complete it today)
- Each lab is due at the beginning of the next lab session. So you will have one week to complete each lab assignment and the next prelab.
- Any questions or concerns, please come and see me.

Good Luck!

What types of computation devices do you know or have used?

- PC
- Laptop
- Cellphone
- ipad
- Car: cruise control, temperature control
- Washing machine controller
- Cloud computing
- …

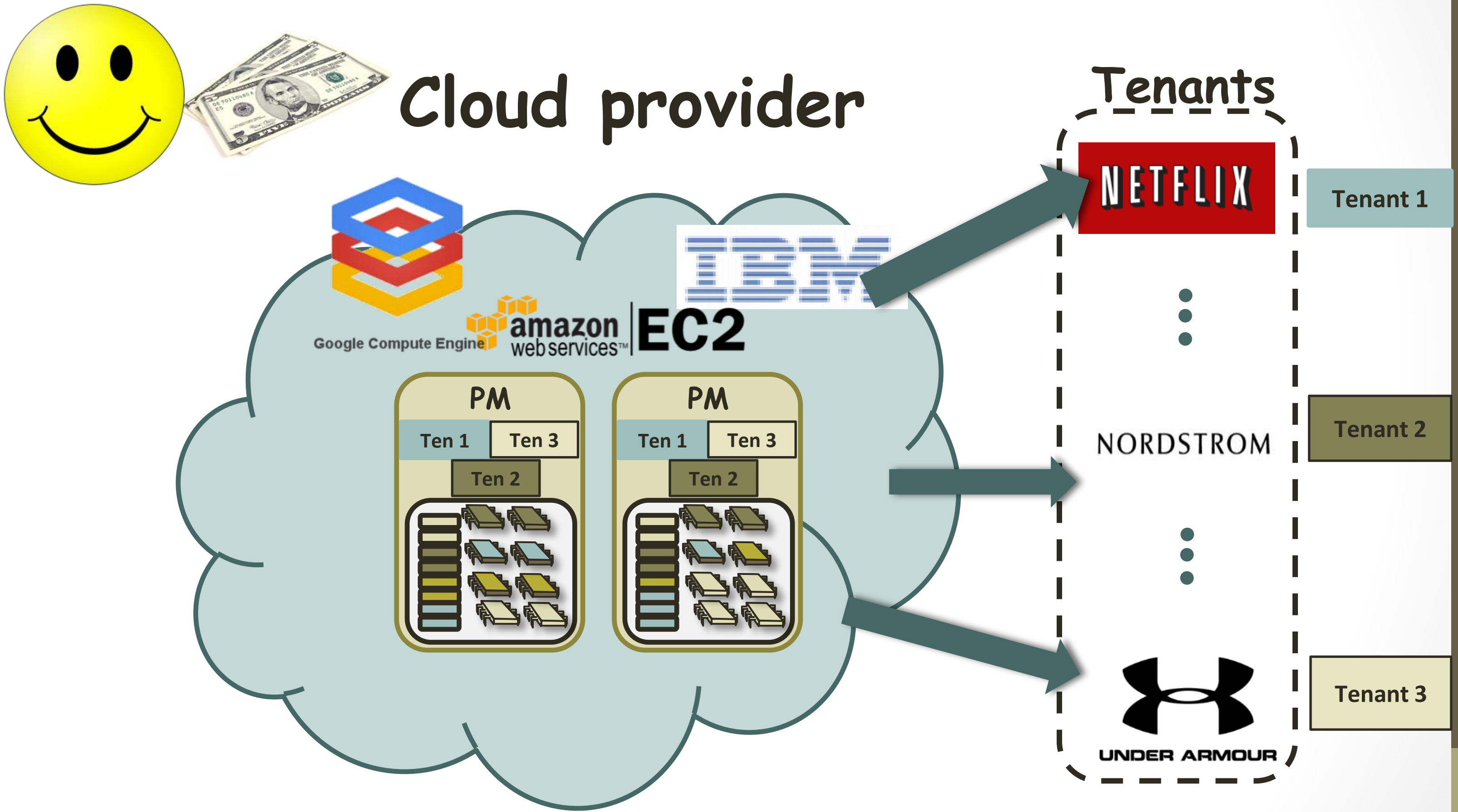
Cloud Computing

- What type of cloud platforms do you know about?

- Which one of these have you used before? Which provider?

- There are usually the physical servers at the cloud's datacenter which hosts all the cloud services
- Sometimes there are tens of different customers running on the same physical machine!!!

Why does the cloud co-locate its customers on the same PM?

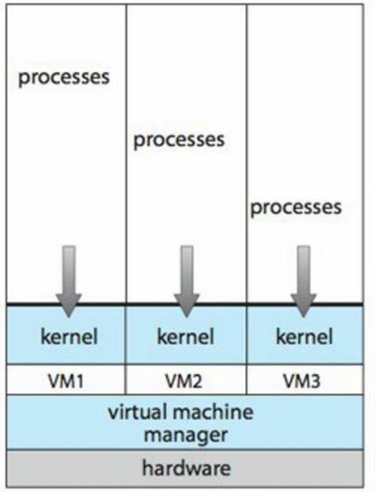


PM: Physical Machine

Cloud Computing: Activity

- Can you draw a schematic of a cloud’s physical server with 4 ×3.2 GHz CPUs and 16 GB RAM and 1Gbps network bandwidth
- Assume this server is hosting 8 different tenants with each having the following specs: 1 ×1.2 GHz vCPU, 2 GB RAM, and 128 Mbps network bandwidth
- How would you adjust your schematic?
- What are the problems and challenges here?
- Is it at all possible?!? It seems like it should be…
- Who is in charge of all this management and

coordination?



Cloud as the future of the computation

- We are not going to solve all these problems today, for sure!!!
- It is important to understand the importance of the cloud and being familiar with its challenges
- At the same time remember that learning all these requires understanding the principles of Operating Systems Design and Resource Virtualization and Management in standalone systems (your PCs)
- Also do not forget that different customer or tenants in the cloud are similar to the different processes running on your machine at the same time!!! So if you think about it again many of the problems and challenges are similar…
- So let’s dive into understanding Operating System Design first, and then come back to the cloud operations

Computer Important Hardware

- CPU
- Memory
- Disk
- Network Adapter
- Caches
- Memory bus
- Keyboard
- Mouse
- Touch screen
- …

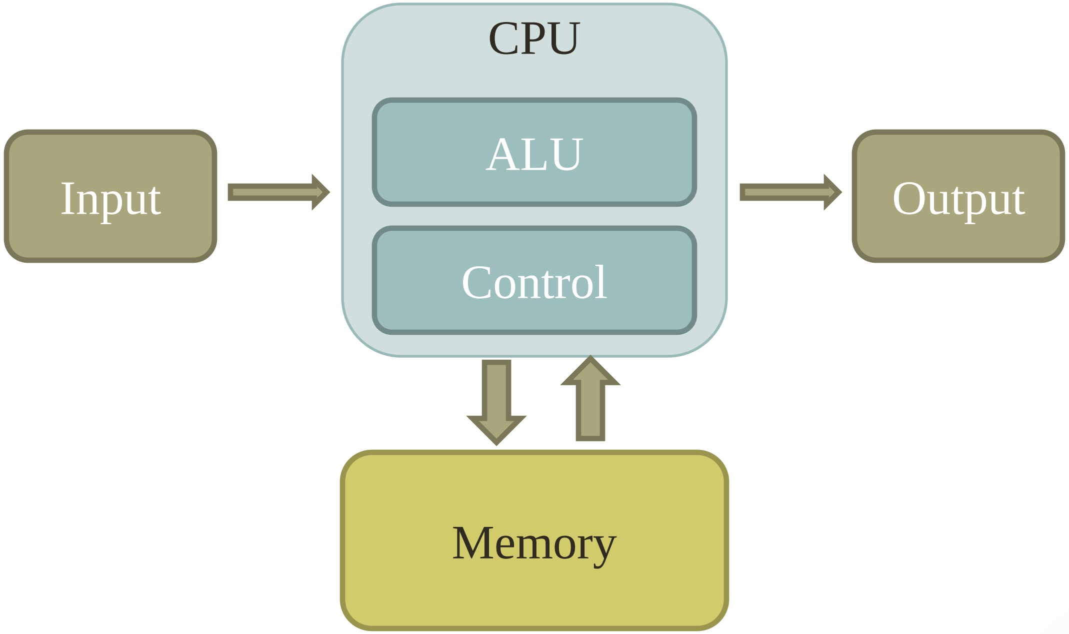
Imagine you have to develop an interface to use the physical hardware (keyboard, printer) to

- Read the key strokes of the keyboard
- Print a page of pdf using a printer

Do you rather just use the API and use cin, scanf, …

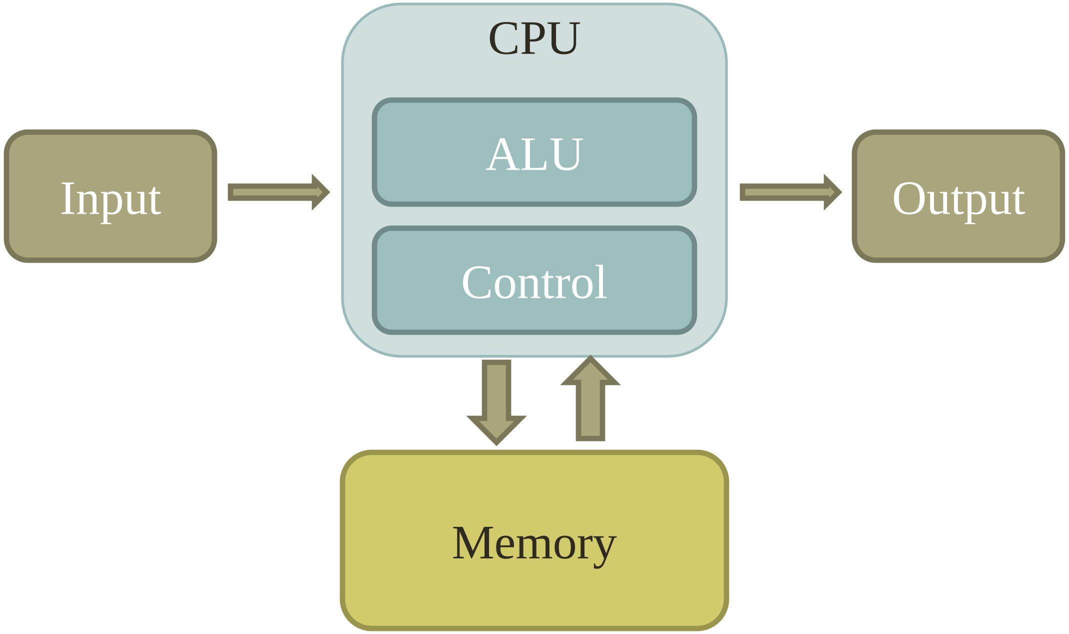
How does a computer work?

- Today computers work based on a Von-Neumann architecture
- What is Von Neumann architecture?
  - A stored computer program model
  - Note: a running program is called a “process”
  - Instructions are fetched from the memory to CPU register (Instruction register) and is executed



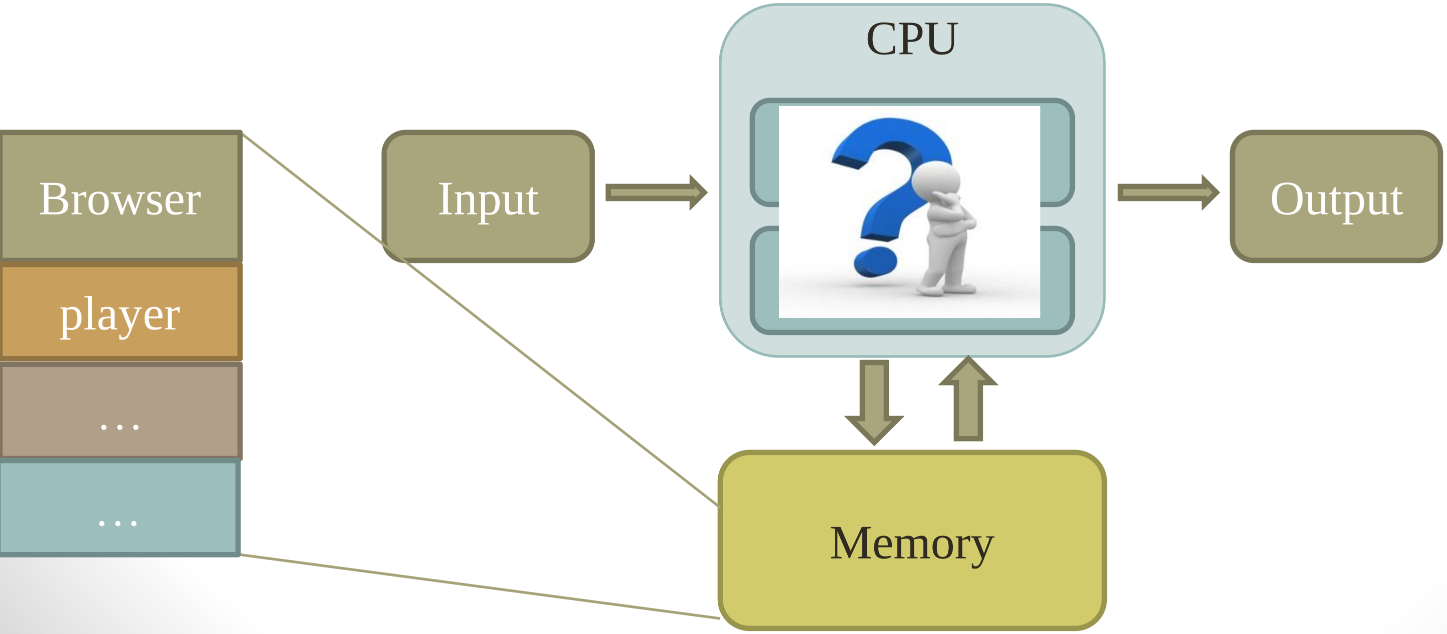
How does a computer work? (continued)

- For a program to execute, the instructions should be loaded into memory and executed one by one by the CPU
- What if there are multiple programs (Multiprogramming system) running simultaneously (multiple processes), e.g., browsing the web, while listening to music
  - What are the possible problems here?!?



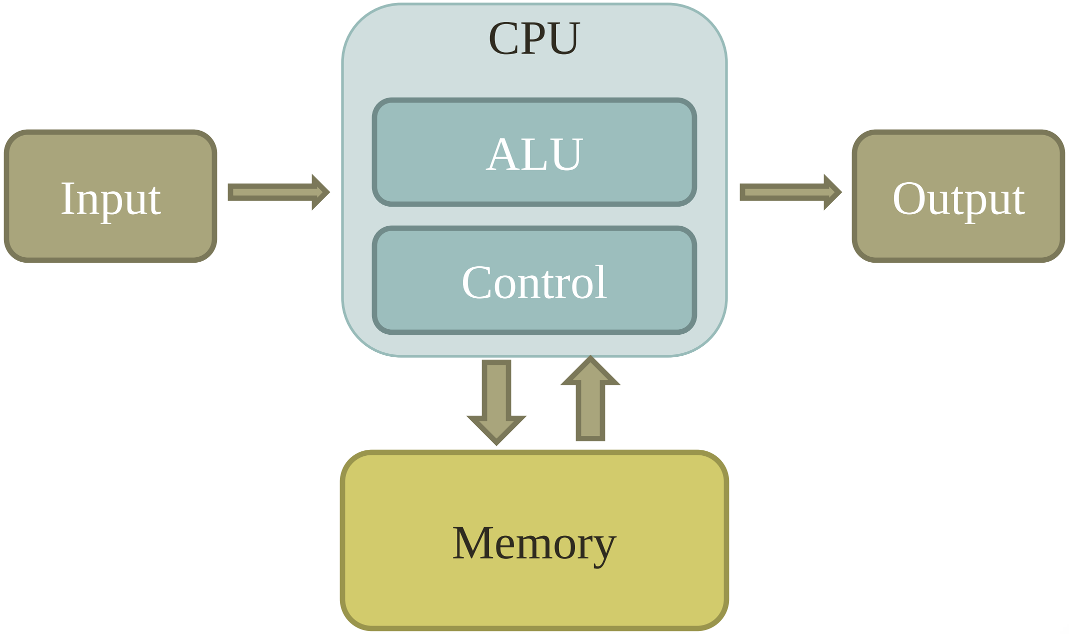
How does a computer work? (continued)

- For a program to execute, the instructions should be loaded into memory and executed one by one by the CPU
- What if there are multiple programs (Multiprogramming system) running simultaneously, e.g., browsing the web, while listening to music
  - The memory does not know which part of it belongs to which process
  - CPU does not know which process to run at each time (should it finish the browsing and then play 5 mins of the song!?!)



How does a computer work? (continued)

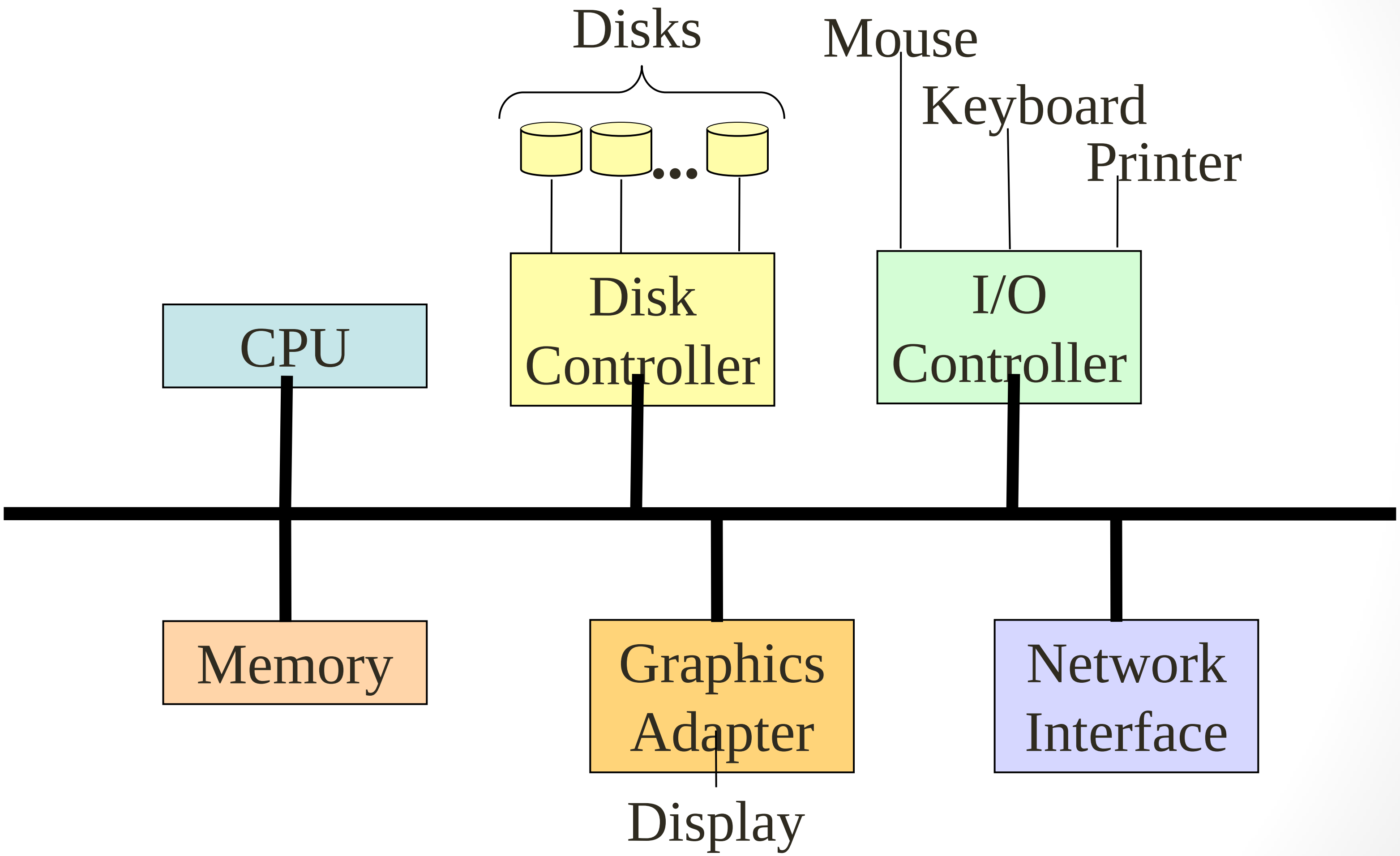
- Another example: I am typing a document and the music player is running. Should the player stops until I finish typing?
- How can the CPU know if a keyboard key was pressed?
  - Should it wait in a while(true) loop for a key to be clicked?!?



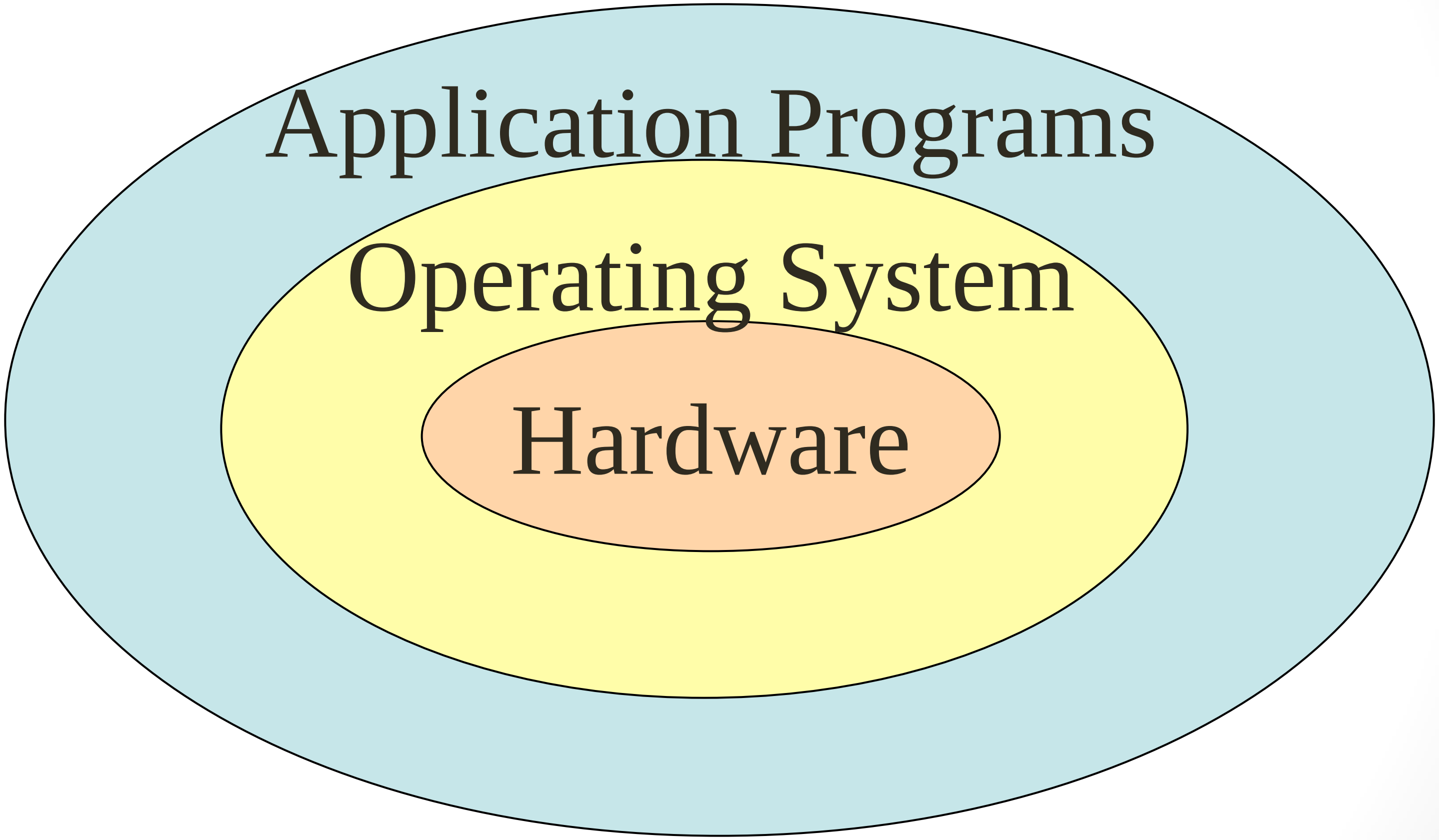
We Need a Manager Here!!!

- The Manager's task is to Facilitate Resource Sharing, e.g., CPU time, storage devices, and I/O devices
- What are some issues when we deal with resource sharing?
  - What if two people ask for the same thing at the same time? (priority)
  - What if one person does not get any of the resources? (starved)
  - What if there is an emergency need for a resource that is being used by someone else? (preemptive)
  - What if A has the sign-in sheet and needs the pen to sign it, but B has the pen and is waiting on the sign in sheet? Can any of them sign in? or should one of them give up its resource to be able to complete its task? (deadlock)

A Modern Computer System



Macroscopic Abstract View of the Computer System



Computer System Components

1. Hardware – provides basic computing resources (CPU, memory, I/O devices).

2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.

3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).

4. Users (people, machines, other computers).

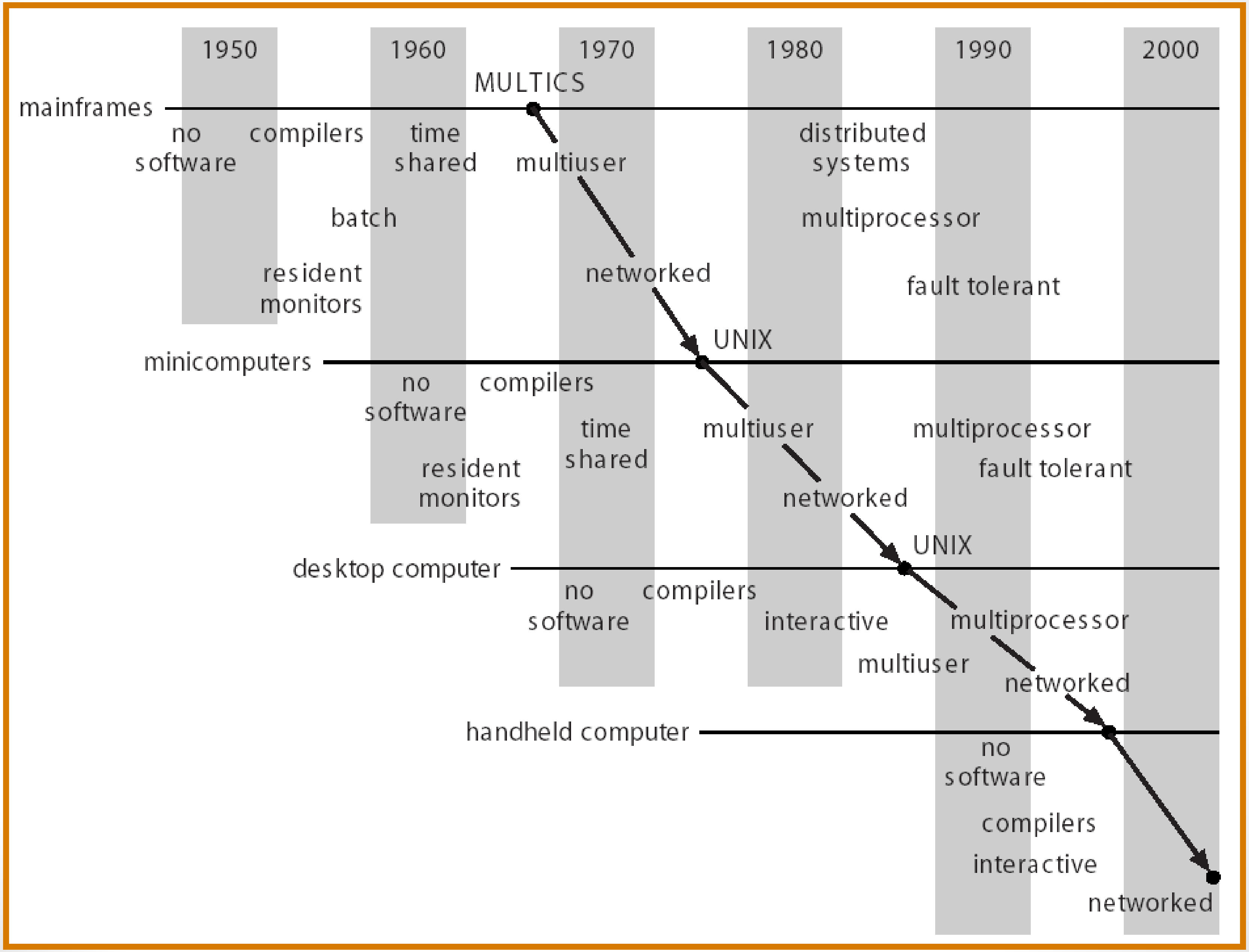
What is an Operating System?

A "program" that acts as an intermediary between a user of a computer and the computer hardware.

- The OS manages resources in the computer system.
- The OS controls the execution of programs.

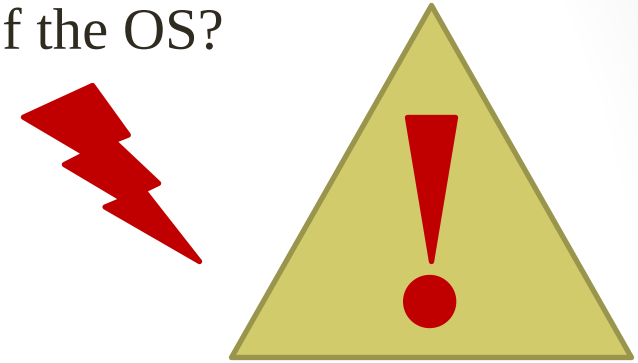
Operating System Definitions

- Resource allocator – manages and allocates resources.
- Control program – controls the execution of user programs and operations of I/O devices.
- Kernel – the one program “running” at all times (all else being application programs).


1950
1960
1970
1980
1990
2000
MULTICS
mainframes
no software
compilers
time shared
multiuser
distributed systems
batch
multiprocessor
resident monitors
networked
fault tolerant
UNIX
minicomputers
no software
compilers
time shared
multiuser
multiprocessor
fault tolerant
resident monitors
networked
UNIX
desktop computer
no software
compilers
interactive
multiprocessor
multiuser
networked
handheld computer
no software
compilers
interactive
networked

System Call

- What are some of the functionalities of the OS?
  - Allocating CPU to a process
  - De-allocating CPU from a process
  - Stopping a process from running
  - File handling
  - Starting a new process, stopping a process
  - Devices handling
  - …
- Some of these are needed by the user processes, e.g.,
  - Accessing files
  - Accessing devices (printer, keyboard,…)
- The OS has an interface called System Calls, that can be used by user programs



Protecting the Processes and the OS

- What if any user is able to execute any system call?
  - Stop another process…
  - Allocate CPU to its own process for as long as it wants
  - Take control of all the devices
  - ….
- What should we do to protect the processes from damaging each other and the OS?
  - We need a layer of protection, but how?
  - Remember that the CPU just pull the instructions and execute them.
  - How can we label instructions which are privileged form the ones that are regular ones?

Privileged Mode

- There are two modes defined for each instructions which is specified by a mode bit: kernel or privileged mode (0) and user mode (1)
- Note that this is embedded in the hardware, hence can not be tampered with by a user process.

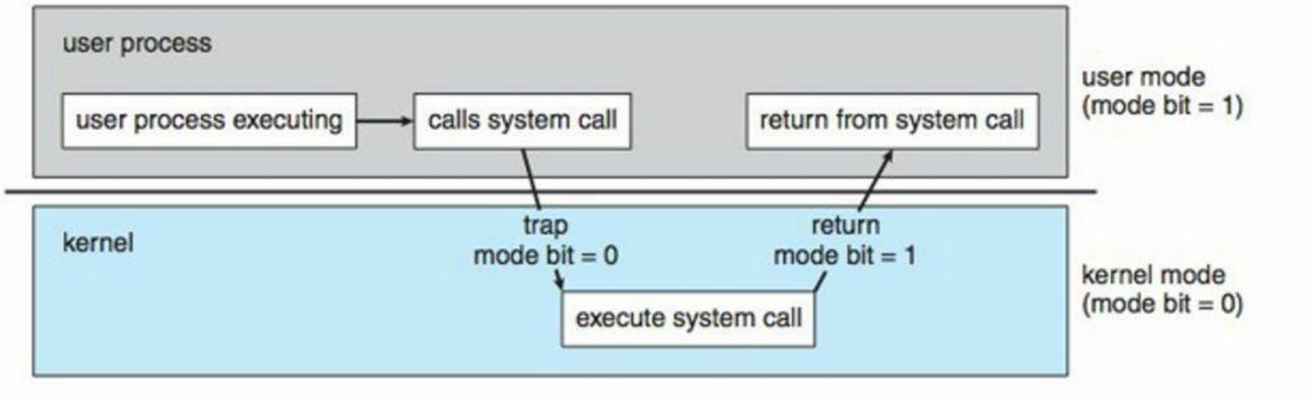


Figure 1.10 Transition from user to kernel mode.

How does this mode change happen?

- Trap is a mechanism that notifies the OS kernel about the intentions of a user process to runs a privileged instructions.
- Trap is a software signal sent to the OS

1. One application of traps is exception handling when running a user program. E.g., (i) division by zero, (ii) accessing memory that is not allocated to the process
   - some error handling is required by the kernel. Hence trap is the name of the signal that is sent to the OS to ask to handle the situation and send back the appropriate error, or take necessary actions
2. The other application of traps it executing system calls
   - When a user process calls a system call, a trap is initiated that returns the control to the OS to execute the system call in the kernel mode. When the execution is complete, the control is transferred back to the user mode.

Next Session

- Polling vs. Interrupts
- Different OS types: Read on your own in quiz next session
- Direct Memory Access (DMA)