CMP325 Operating Systems Lecture 01

Introduction to Operating Systems

Fall 2021

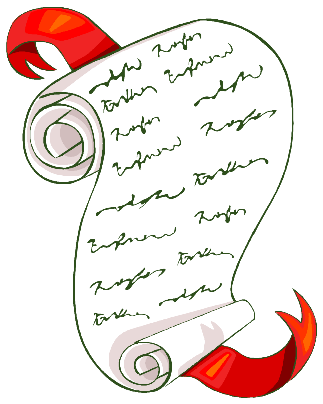
Arif Butt (PUCIT)

Note:
Some slides and/or pictures are adapted from course text book and Lecture slides of
- Dr Syed Mansoor Sarwar
- Dr Kubiatowicz
- Dr P. Bhat
- Dr Hank Levy
- Dr Indranil Gupta

For practical implementation of operating system concepts discussed in these slides, students are advised to watch and practice video lectures on the subject of **OS with Linux** by Arif Butt available on the following link:
http://www.arifbutt.me/category/os-with-linux/

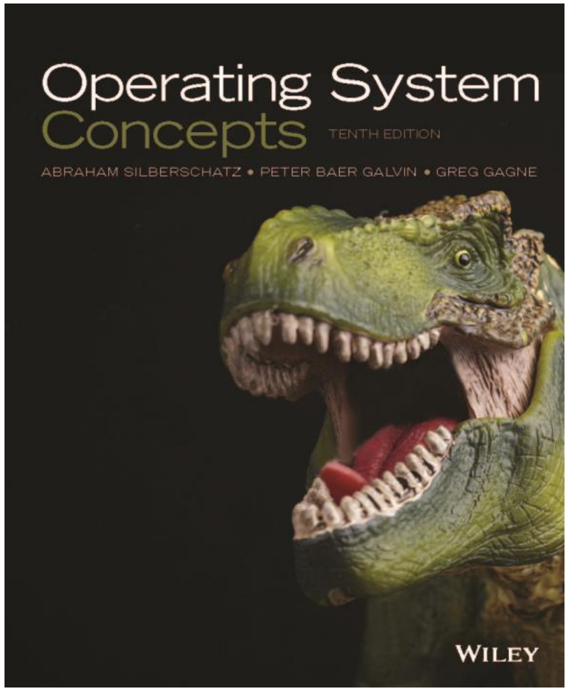
Today's Agenda

- Basic Course information and Class Protocols
- Course Outline
- Operating System Overview
  - OS An Abstraction
  - Interrupt, Trap and Signal
  - Dual mode operation
  - I/O, Memory and CPU Protection
  - Operating System Services
  - Types of operating system
  - Computing environments



Course Information

- **Required Textbook:** Operating System Concepts, 10th Edition Silbershatz, Galvin, Gagne
- **Grades Website:** http://online.pucit.edu.pk
- **Resources Website:** http://arifbutt.me


Operating System Concepts
TENTH EDITION
ABRAHAM SILBERSCHATZ • PETER BAER GALVIN • GREG GAGNE
WILEY

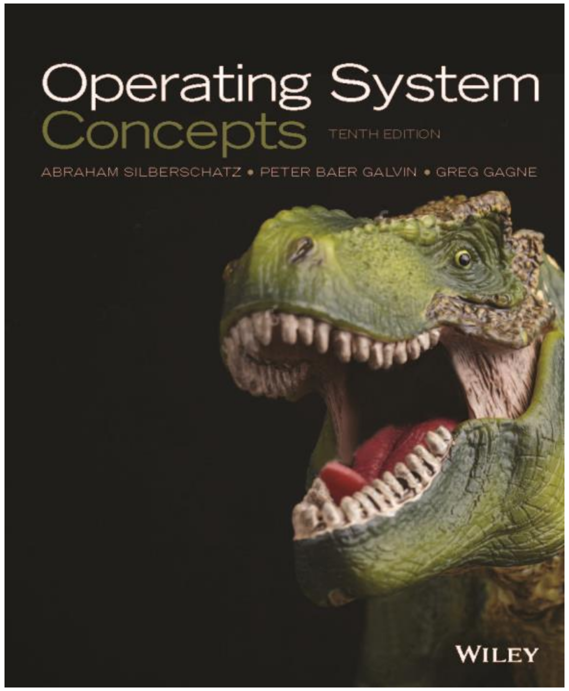
- **Prerequisites :** Nill
- **Corequisites :** CMP223, CMP210
- **Office:** New Faculty Building # G7
- **Students Counseling hours:**
  - Mentioned on http://arifbutt.me
- **Teaching Assistant info:**
  - Mentioned on http://arifbutt.me
- **24 hour turnaround for email:** arif@pucit.edu.pk

Course Information

Visit Course Web Site for following information:

http://www.arifbutt.me

- Lecture Slides
- Video Lectures (OS with Linux)
- Quizzes + Assignments + Labs
- Announcements
- Teaching Assistants
- SOPs and Course related Policies
- Misc Resources


Operating System Concepts
TENTH EDITION
ABRAHAM SILBERSCHATZ • PETER BAER GALVIN • GREG GAGNE
WILEY

Lecture Format

- Please come to class **(in time)**
- I will mark the attendance in the first five minutes. Students with more than eight absents will not be allowed to sit in the class and their names will be struck off the register. **(Be watch full)**
- To Help you understand important and hard OS concepts please read the relevant textbook sections ***before*** lecture
- Exam, quiz, PA and HW questions could be from *anywhere* in the lecture slides, textbook sections covered, reading material provided. **Your best strategy is to play it safe – read all material referred to in lecture**
- Make sure you also
  - Periodically check lecture notes on course web page
  - **Utilize the student counseling hours**

Lab Format

- Please come to Labs **(in time)**
- There are one or two video lectures associated with each lab, uploaded on course web site. Please listen to those lectures twice before coming to lab
- To Help you understand important and hard OS concepts please read the relevant Chapters of UNIX The Textbook by Dr Mansoor Sarwar
- Quizzes might be taken in class or in Lab, so don't miss
- Contents covered in the Lab will come in the Quizzes as well as in the Mid and Final exams

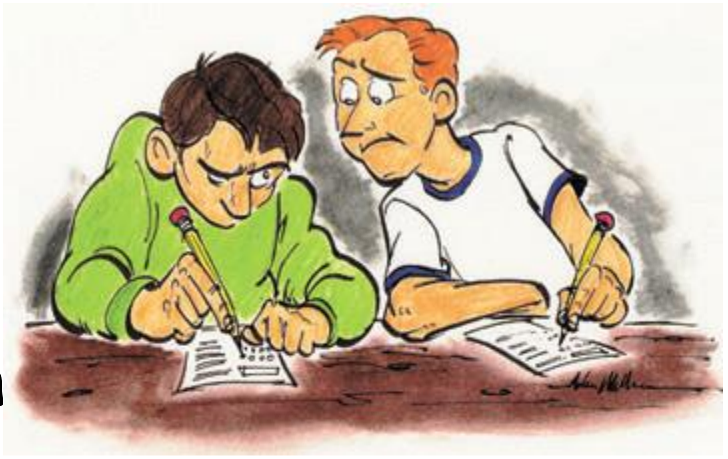
Who cares to get an A?

- Final exam: 40%
- Mid-exam: 35%
- Sessionals: 25%
  - Surprise Quizzes: 10%
  - Assignments (PA/LW/HW): 5%
  - Lab : 10%

Surprise Quizzes

- There will be surprise quizzes, given at the start of a lecture, during any lecture. The total number of quizzes could be anywhere between 4 and 40.
- **NO LATE or MAKEUP SURPRISE QUIZZES**, under any circumstances whatsoever
- Surprise quizzes are completely individual efforts.
- **Your best strategy is to play it safe – attend every lecture and do the reading/programming assignments**

Cheating Policy



- Academic integrity
- Both the cheater and the studen cheater will be held responsible for the cheating
- The instructor may take actions such as:
  - require repetition of the subject work,
  - assign **'zero'** or may be **'negative'** marks for the subject work,
  - for serious offenses, assign an **F** grade for the **course**

Late Policy for HWs and PAs

- Grade book system details can be checked on : http://online.pucit.edu.pk
- Late policy for Assignment, Quizzes, and other deliverables
  - No late Assignment submissions!
  - No late quizzes or exams!
- Sticking to dates is your responsibility!
  - Check announcements on lecture notes regularly
- Your best strategy is to play it safe - submit everything on time

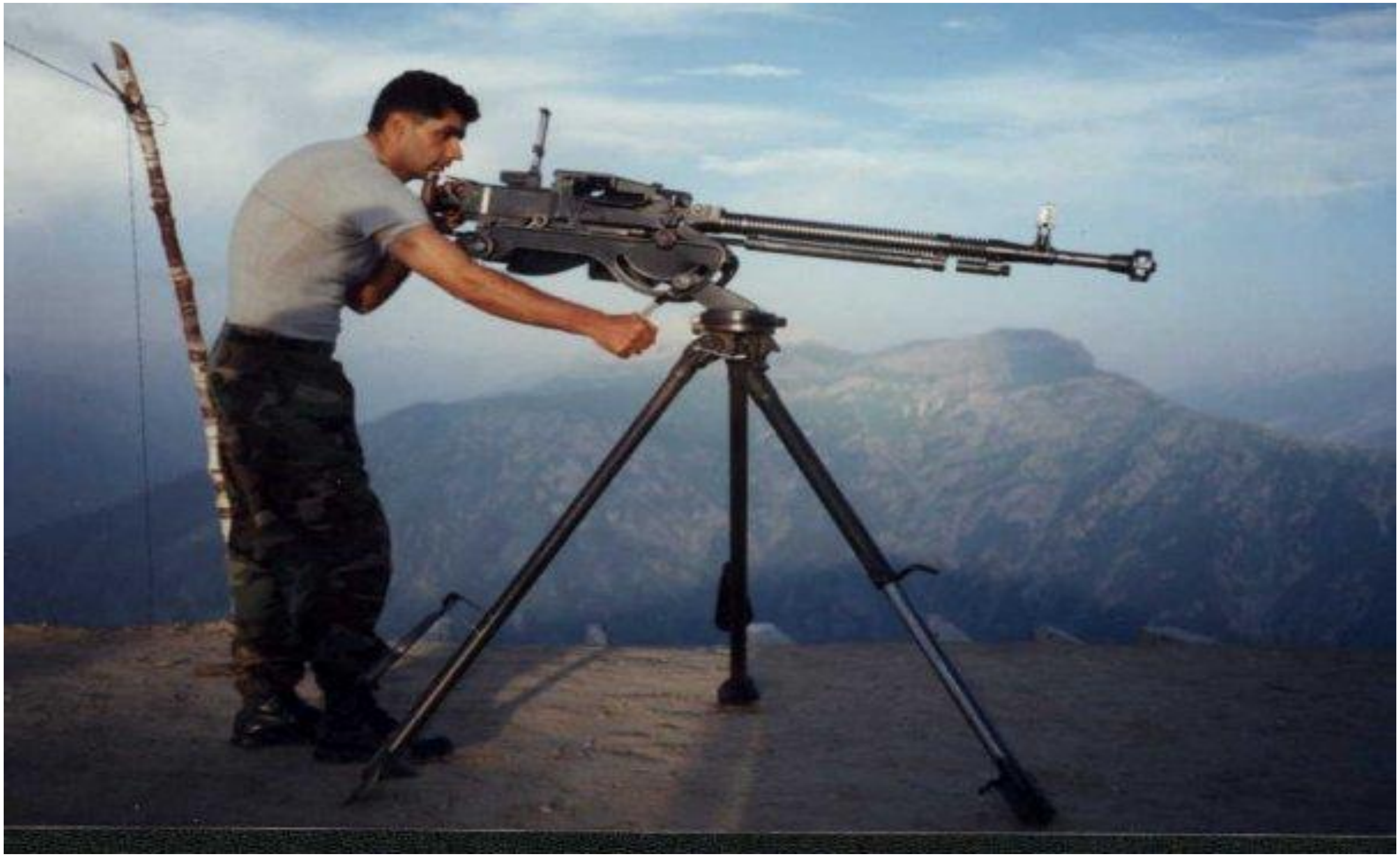
Playing it Safe in CMP325

If you follow these 4 simple rules during the CMP320 class, you'll make sure that you do well in the course:

1. ***Attend every lecture + Lab***
2. ***Study the course material (video lectures, textbook sections assigned + slides + Reading assignments)***
3. ***Submit everything (PAs, HWs, quizzes, exams) on time - don't be late***
4. ***Don't cheat***

Programming Assignments & Linux Shell Commands

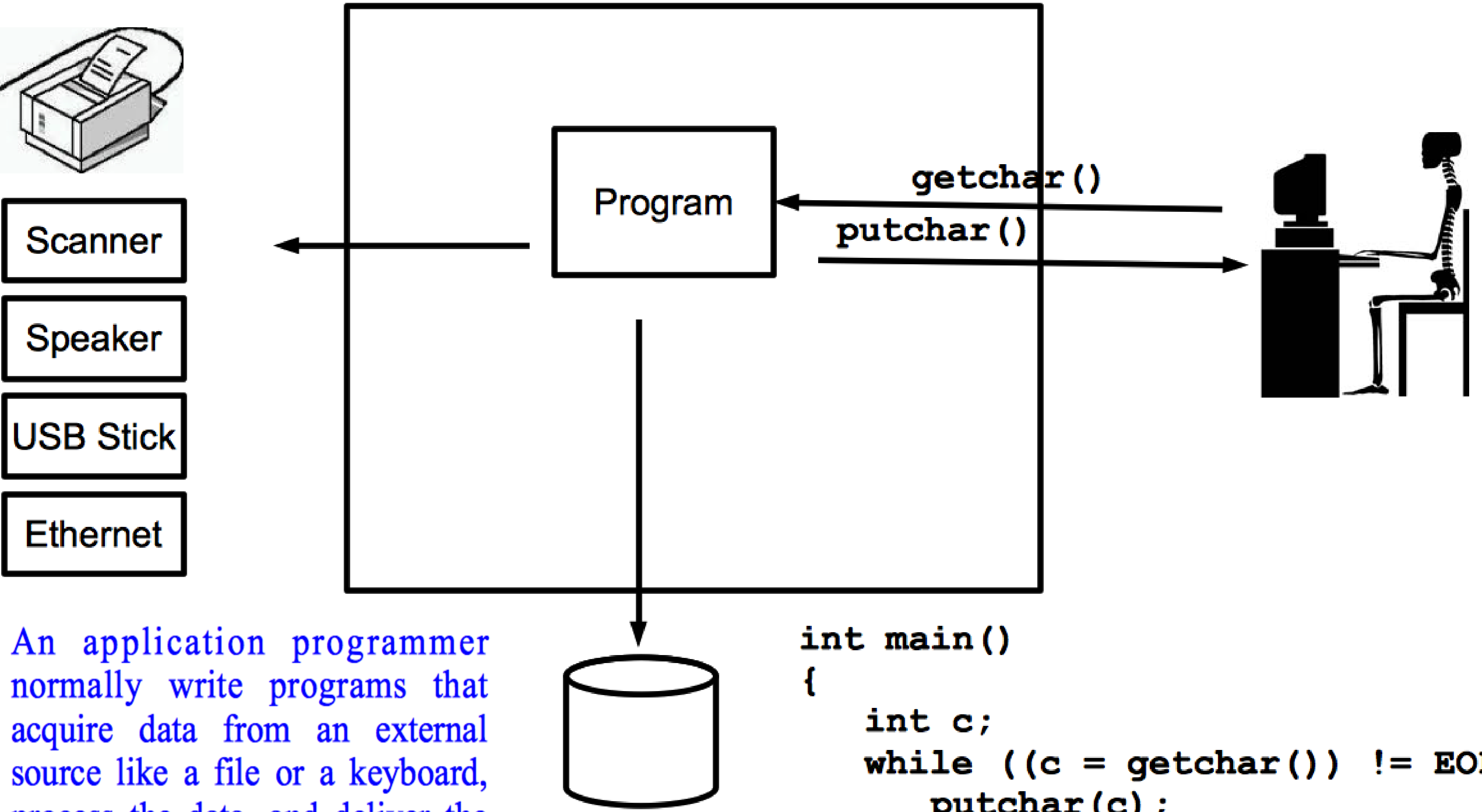
No Pains No Gains



Outline

Operating System Overview

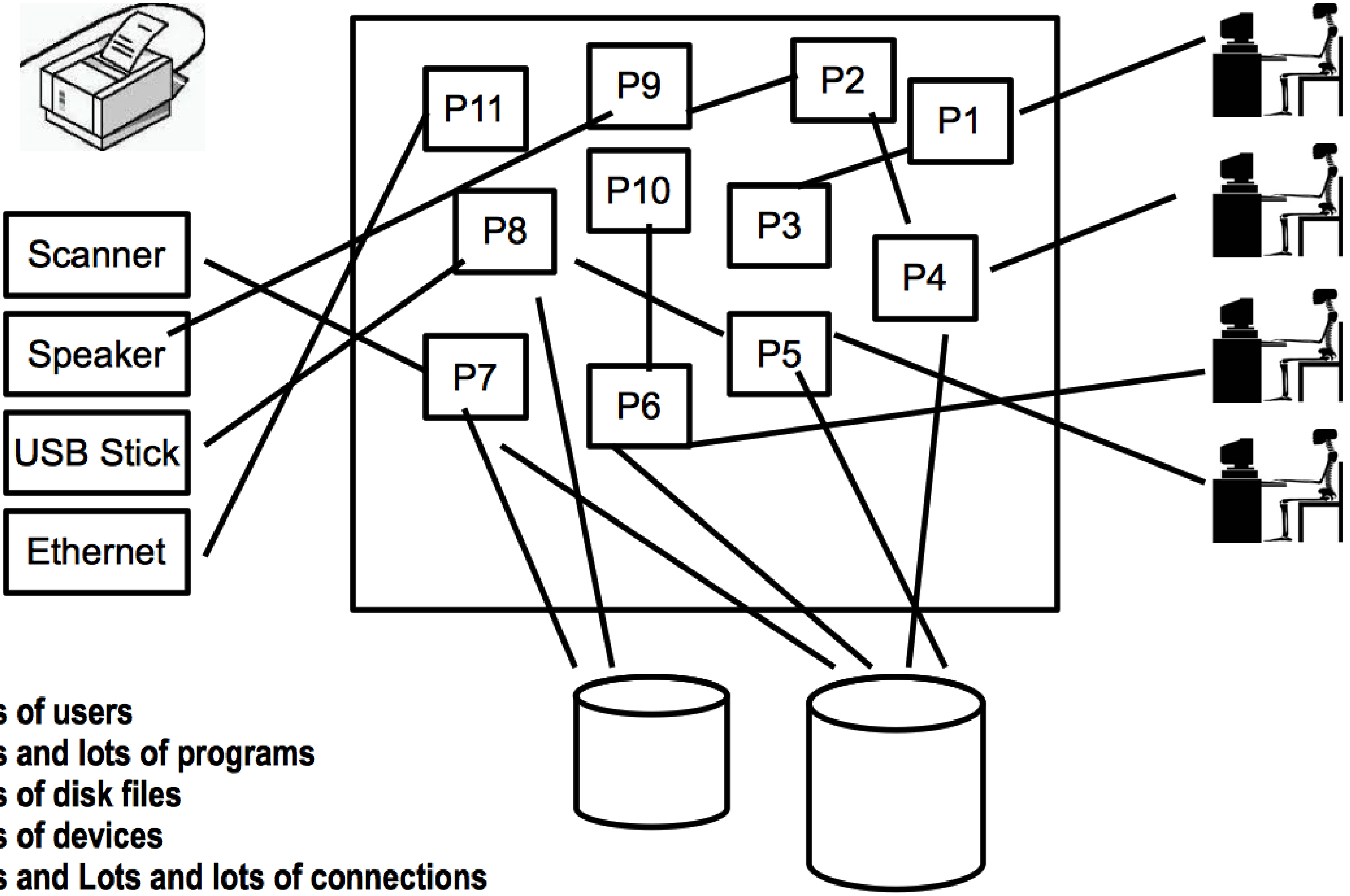
Operating System-An Abstraction



An application programmer normally write programs that acquire data from an external source like a file or a keyboard, process the data, and deliver the output to external sources such as files or the console

```
int main()
{
    int c;
    while ((c = getchar()) != EOF)
        putchar(c);
    return 0;
}
```

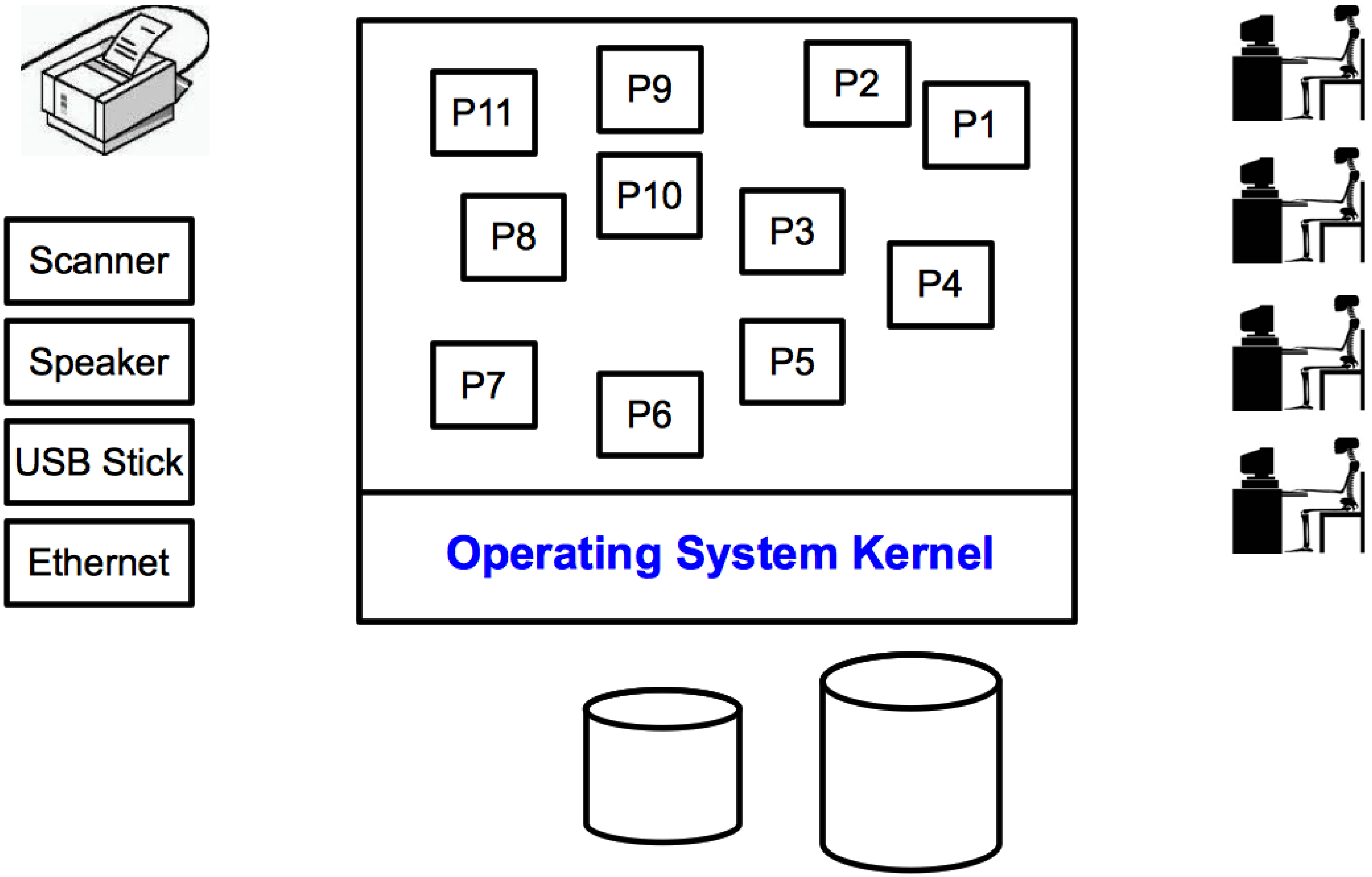
Operating System-An Abstraction



Lots of users
Lots and lots of programs
Lots of disk files
Lots of devices
Lots and Lots and lots of connections

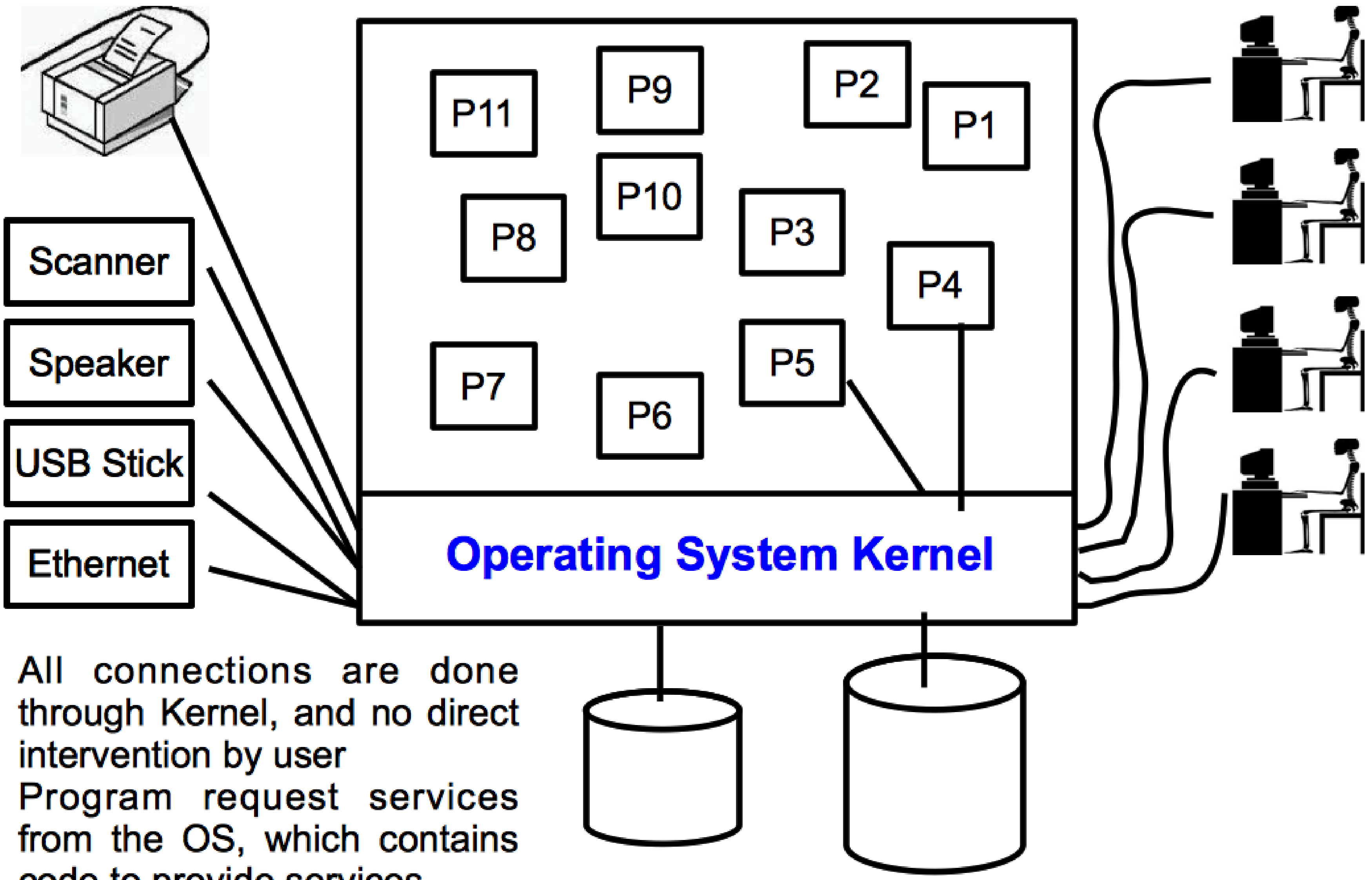
Who is managing all these resources and connecting various devices to correct programs

Operating System-An Abstraction



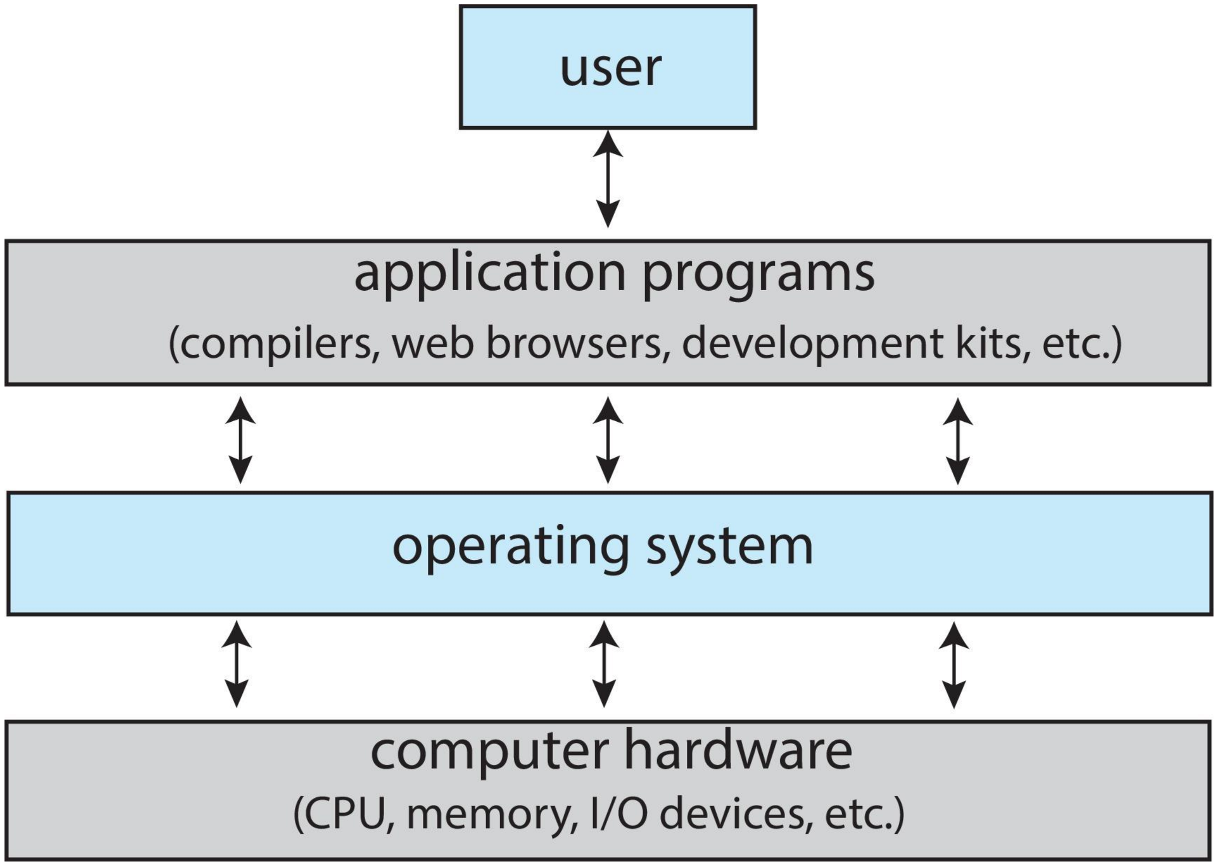
Role of Operating System is to manage all these resources and to connect various devices to the correct programs

Operating System-An Abstraction

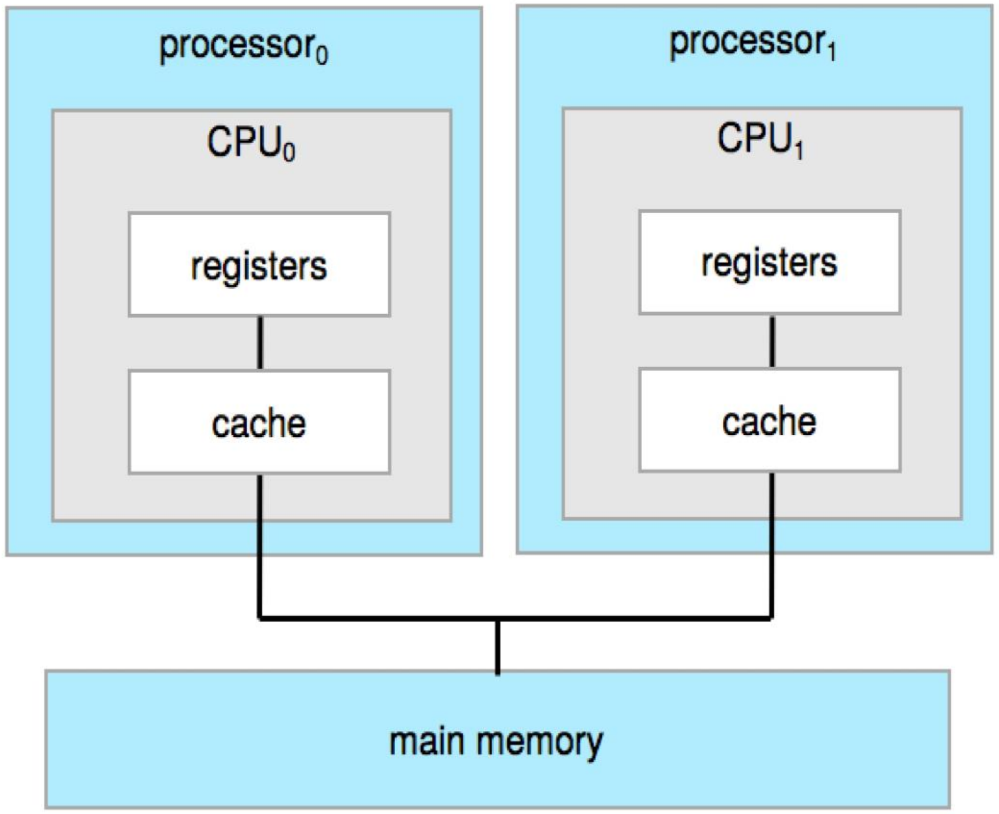


- All connections are done through Kernel, and no direct intervention by user
- Program request services from the OS, which contains code to provide services
- An operating system provides these services via its API

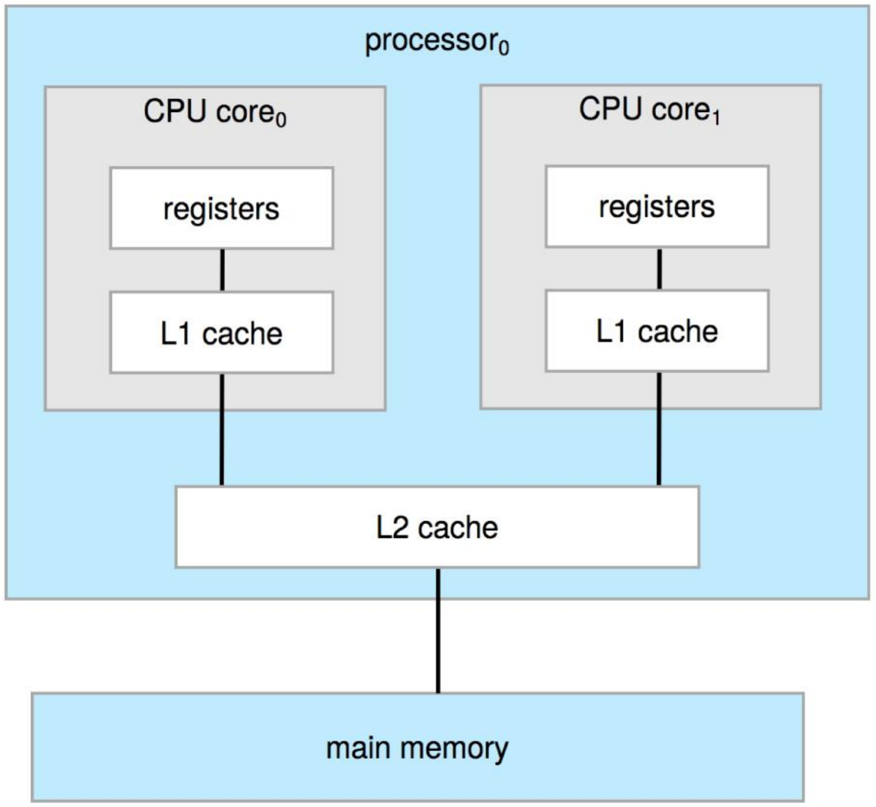
Operating System–An Abstraction


user
application programs
(compilers, web browsers, development kits, etc.)
operating system
computer hardware
(CPU, memory, I/O devices, etc.)

Computer System Organization



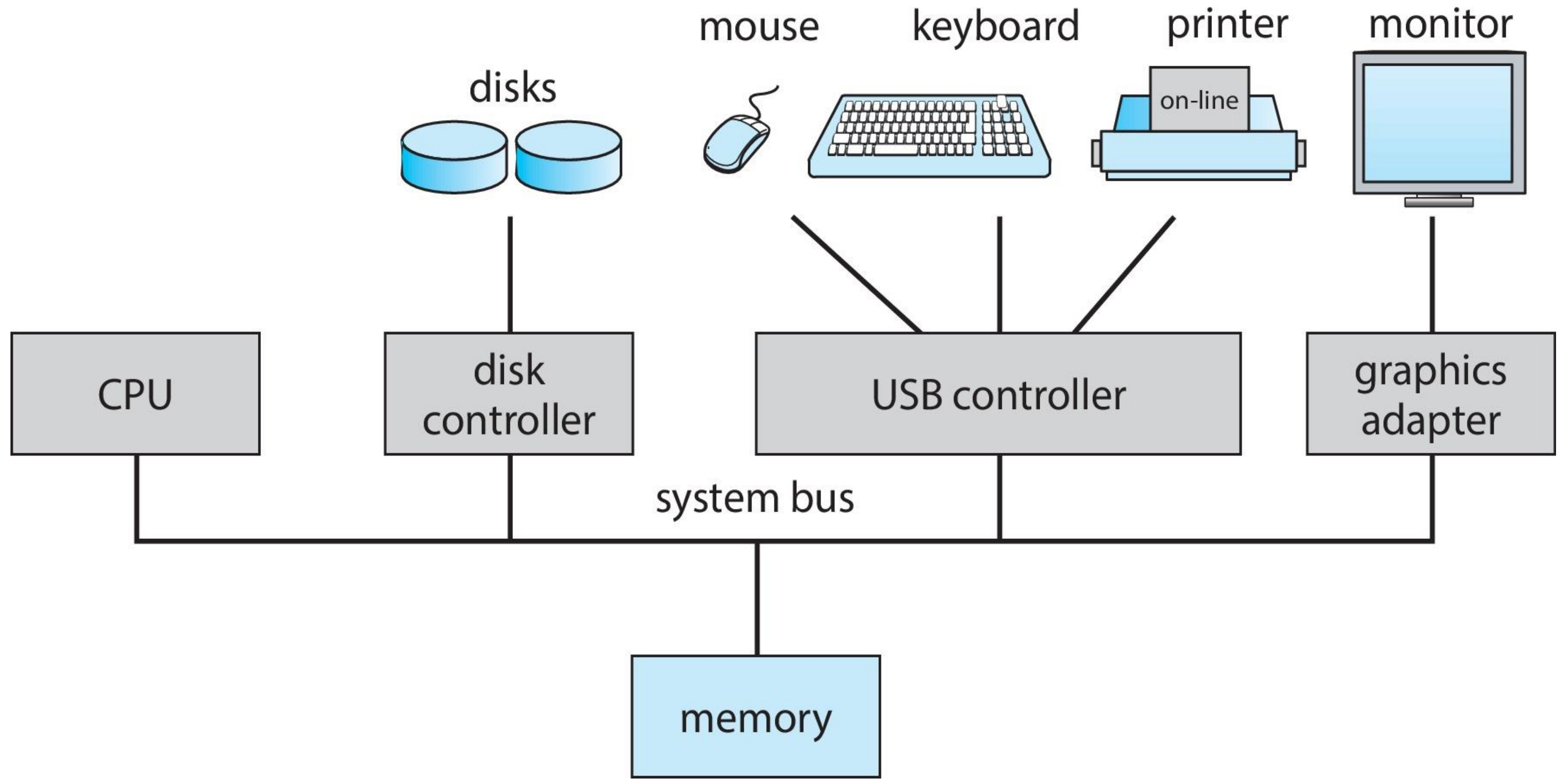
Multi-processor Architecture



Multi-core Architecture

Computer System Organization

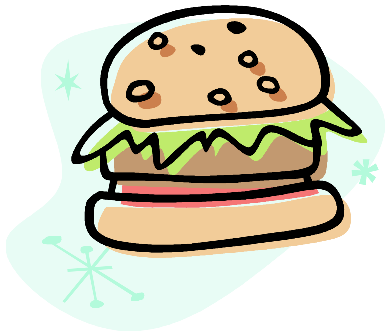
- One or more CPUs, device controllers connect through common bus providing access to shared memory
- Concurrent execution of CPUs and devices competing for memory cycles



Computer System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- Each device controller type has an operating system device driver to manage it
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an interrupt

What Is an OS?



"Code" that:

- Sits between programs & hardware
- Sits between different programs
- Sits betweens different users

But what does it do?

Provides an orderly and controlled allocation of the processor(s), memory(ies) and I/O devices among the various programs competing for them

Loose analogy:

- **Government: creates and enforces laws that govern resources (money, land, houses, vehicles, oil, etc.) but allows citizens to have freedom with using the resources (as long as the citizen obeys laws)**

What Is an OS? (…)

Resources

- Allocation
- Protection
- Reclamation
- Virtualization

Services

- Abstraction
- Simplification
- Convenience
- Standardization

Resource management includes multiplexing (sharing) resources in two ways:

- **Time Multiplexed** (CPU sharing, Printer sharing).
- **Space Multiplexed** (Memory, Hard Disk).

What is an OS? (…)

Objective of an OS is to Tame the h/w so that people can program and can get something useful out of it.

Top Down View: A program that acts as an intermediary between a user of a computer and the computer hardware. Present the user with the equivalent of an **extended machine** or **virtual machine** that is easier to program than the underlying hardware. **(HOW?)**

Bottom Up View: A program that allocates and de-allocates computer system resources (CPUs, memories, timers, disks, mice, network interfaces, printers, …) in an efficient, fair and secure manner. **(HOW?)**

"An OS is a program running at all times on the computer (usually called the kernel), that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware."

"Primary goal of OS is convenience of user and secondary goal is efficient operation of the computer system."

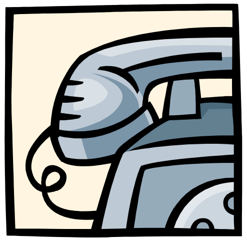
Name some OSs

- UNIX
- Linux
- Sun Solaris
- Macintosh
- PC BSD
- MS Windows

- Android from Google
- BlackBerry from RIM
- iOS from Apple
- Symbian from Nokia
- Windows phone from MS

Interrupt

- Virtually all computers provide a mechanism by which other modules (I/O, memory) may interrupt the normal sequencing of the processor.
- Provided to improve processor utilization
- Most I/O devices are slower than the processor and Processor must pause to wait for device



Answer The Phone
.
.
.
.
.
.
Resume

Sub Routine Call vs Interrupt

Sub routine Call

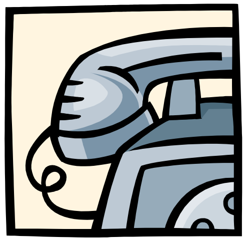
- Occur due to execution of an instruction
- Address of subroutine is determined by the address part of an instruction
- Stores only the contents of PC (return address)

Interrupt

- Occur due to an internal event (setting of flag) or an external event (key press)
- Address of ISR is determined by h/w (vectored / non vectored interrupt)
- Other than contents of PC, store the state of processor as well (all processor registers)

Interrupts, Traps and Signals

- **Interrupt**. An event generated by an I/O device to get the attention of CPU and control goes to the OS. State of the CPU is saved, ISR is executed and then state of CPU is restored.
- **Trap**. An event generated by CPU and control goes to the OS. Normally when an instruction is executed that may cause a division by zero or protection error. State of CPU is not saved and TSR is executed.
- **Signal**. A notification given to a process by the OS because the process did something, the user did something, one process wants to tell another process something.

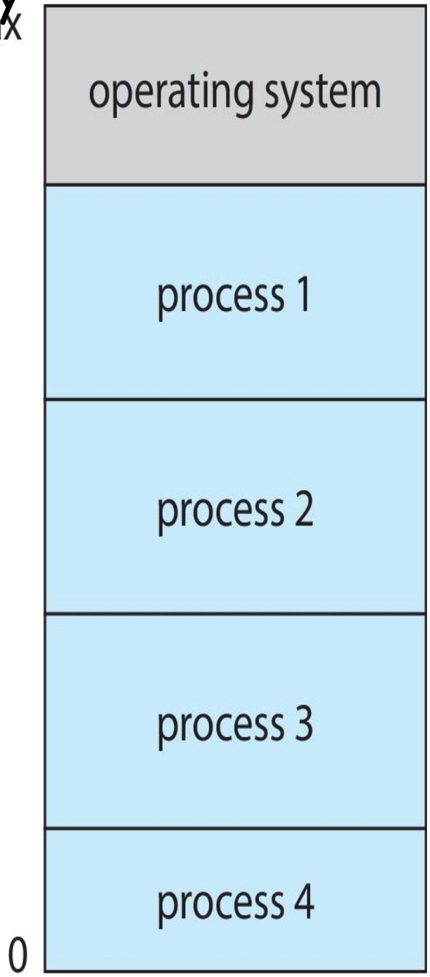


Answer
The
Phone
.
.
.
.
.
.
Resume

Multiprogramming vs Multitasking

Multiprogramming (Batch system) needed for efficiency

- Single user cannot keep CPU and I/O devices busy at all time
- Multiprogramming organizes jobs so CPU always has one to execute
- A subset of total jobs in system is kept in memory
- One job selected and run via job scheduling
- When it has to wait (for I/O for example), OS switches to another job



Timesharing (multitasking) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing

- Response time should be < 1 second
- Each user has at least one process executing in memory
- If several processes are ready to run at the same time, then OS performs CPU scheduling
- If processes don't fit in memory, swapping moves them in and out to run
- Virtual memory allows execution of processes not completely in memory

PROTECTION

In Time Sharing System there are multiple processes using various resources of the computer. Sounds great, but like no free lunch, it has disadvantage as well, i.e. the issue of protection

- Keep user programs from crashing the OS
- Keep user programs from crashing each other
- Keep parts of OS from crashing other parts

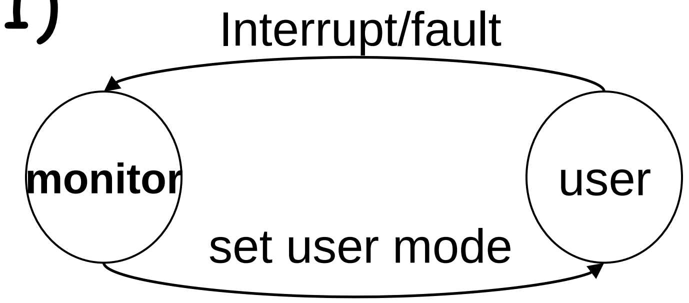
Protection is implemented by keeping two modes

Dual Mode Operation

- To protect the OS and all other programs and their data from any malfunctioning program, protection is needed for any shared resource. It is done by keeping two modes:
  - **User Mode.** Execution done on behalf of a user program
  - **Monitor/Kernel/System/Supervisor Mode.** Execution done on behalf of OS
- Mode bit added to computer h/w to indicate the current mode: Kernel(0), User (1)

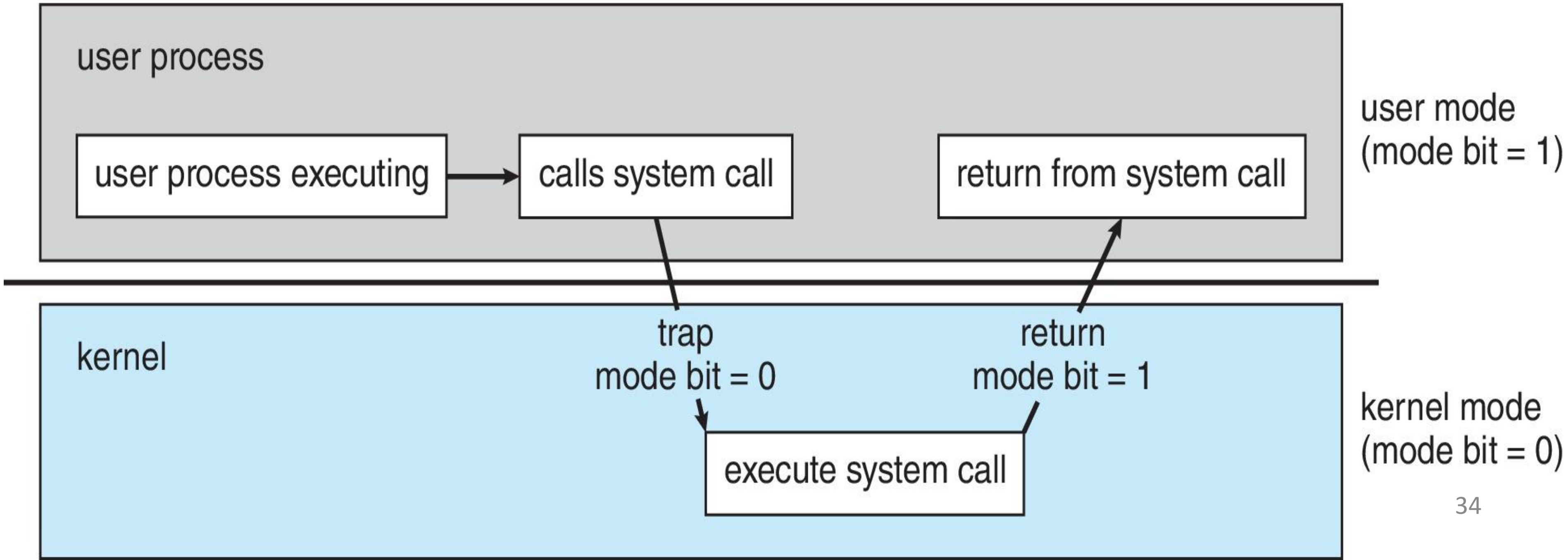
When an interrupt or fault occurs h/w switches to monitor mode.

Privileged instructions can only be executed in monitor mode.



Transition from user to Kernel Mode

- A user process must not be given an open access to kernel code
- Any user/application request that involves access to any system resource must be handled by the kernel code
- The mechanism used by an application program to request a service from Operating System is via a system call (details in next lecture)
- A **System call** is usually a request to the OS kernel to do a h/w, system specific or privileged operation.



How to protect Processes from one another?

Need three important things:

- **Protection of I/O devices**
  - Every process should not have access to every device
- **Protection of memory**
  - A process should not access another process address space
- **Preemptive switching from task to task**
  - Use of timer
  - Must not be able to disable timer from user code

I/O PROTECTION

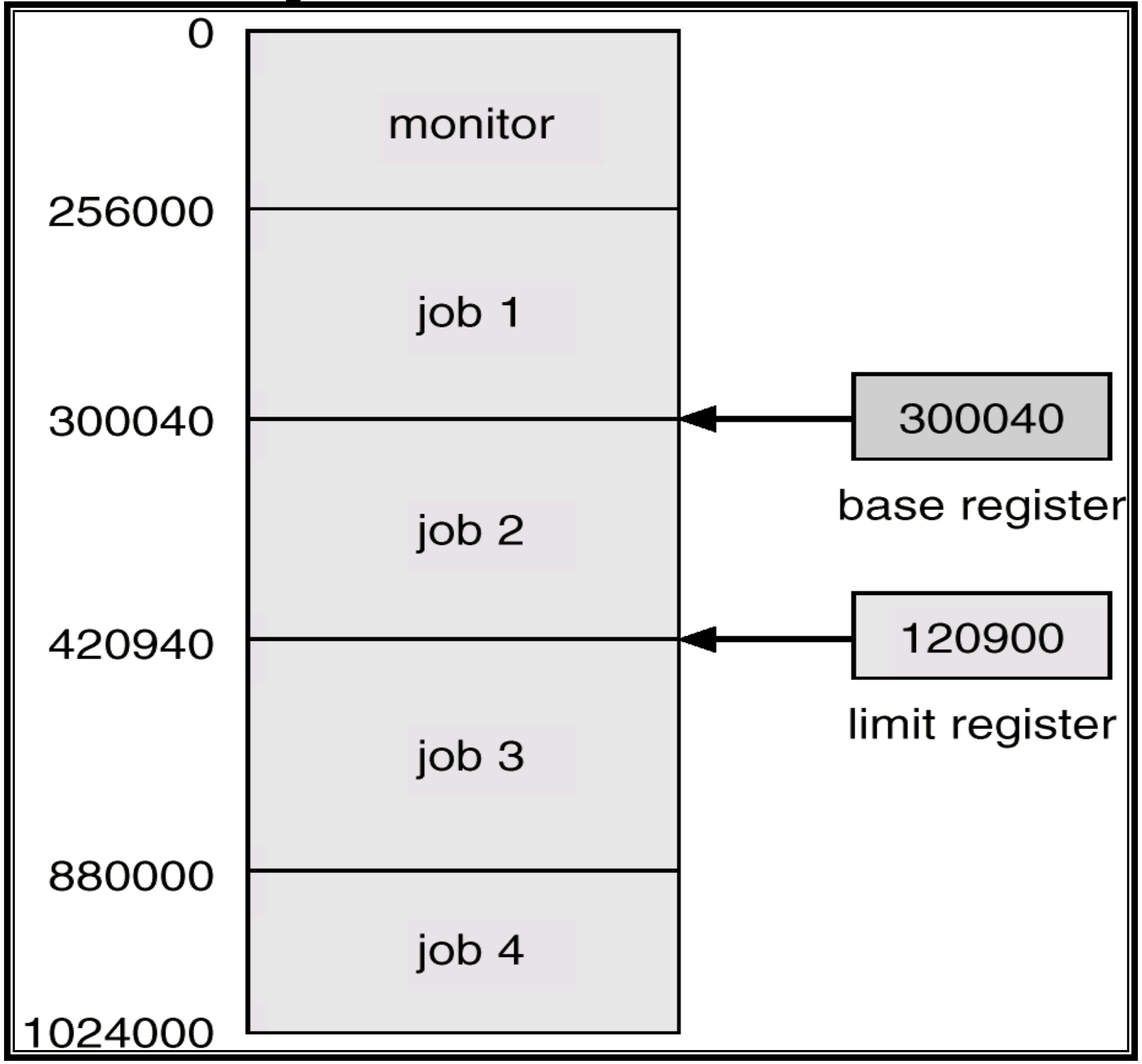
- All I/O instructions are privileged instruction, no user process can execute an I/O instruction directly. If he/she is permitted he/she can always overwrite/delete some ones else data
- I/O is done using system call. A way that an OS allows a user process to invoke an operation such as an I/O operation e.g. opening, reading and writing to a disk file

```
MOV  RAX, 1      ;Place system call number in RAX
MOV RDI, 1       ;Place first argument in RDI
MOV RSI, msg     ;Place second argument in RSI
MOV RDX, 26      ;Place third argument in RDX
syscall          ;write(1, msg, sizeof(msg))
```

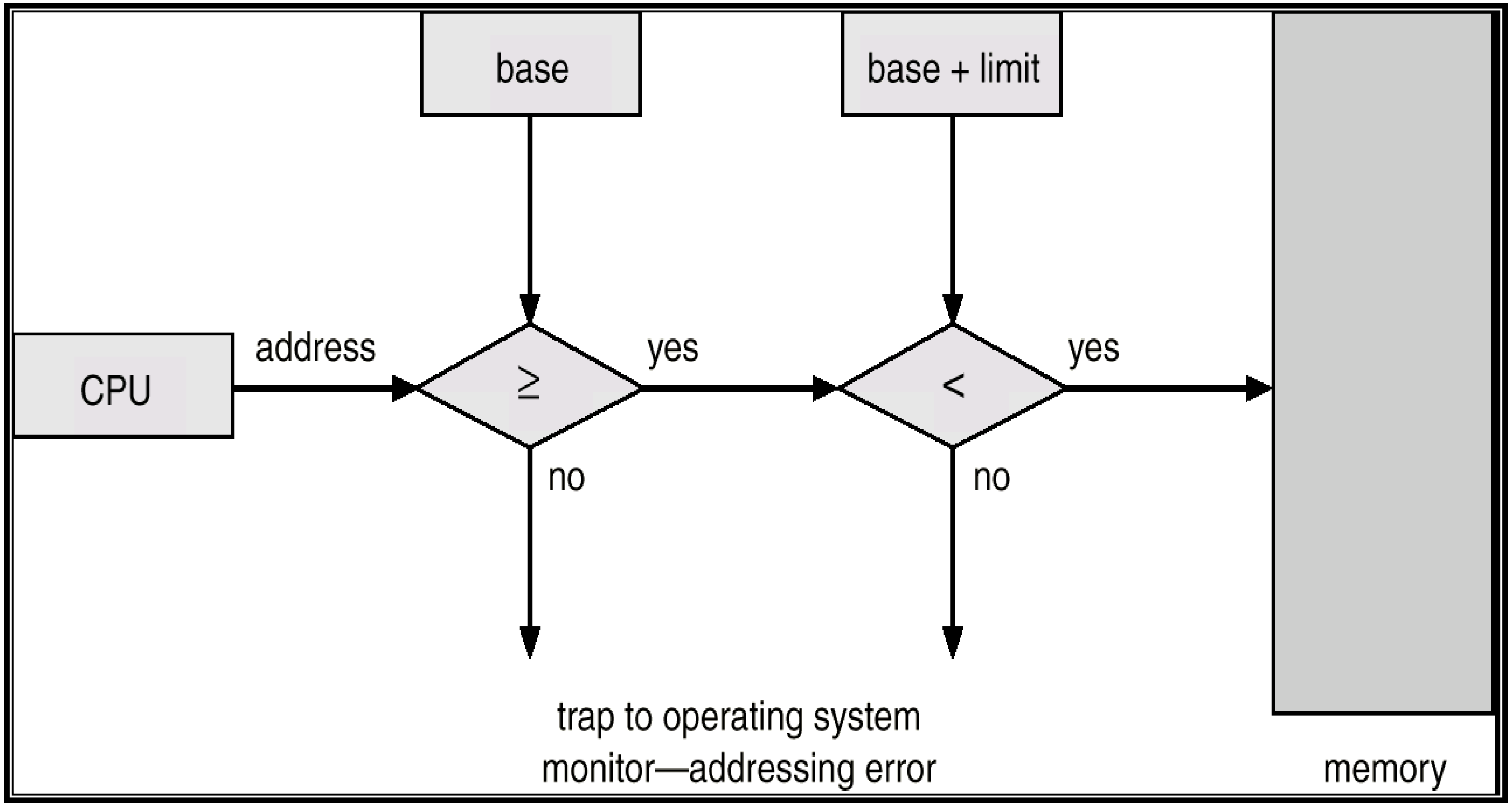
Memory Protection

- **Process Address Space.** Region in the main memory that a process can legally access is known as its address space. A process must not be allowed to go outside its address space
- **Implementation.** CPU uses two registers to determine the range of logical addresses a program may access
  - **Base Register** holds the smallest/starting address of the Address Space assigned to a process
  - **Limit Register** contains the size of the address space allocated to the process

Memory Protection (cont…)



Memory Protection (cont…)



CPU Protection

- In multiprogramming or time sharing OS, multiple processes are running at a time and we want to make sure that the CPU must not stay with a process for an infinite amount of time
- **Implementation.**
  - Timer interrupts computer after a specified period to ensure OS maintains control
  - Timer (counter register) is loaded with a predetermined value and decremented every clock tick
  - When the timer reaches the value 0, an interrupt occurs and an ISR is executed to switch CPU to another process
  - **How Timer is loaded?** This is also a privileged instr, because no one other than the admin should be allowed to assign a time slice to various processes

Operating System Services

Some important tasks a kernel performs are:

- Process management and IPC
- Signal handling
- Synchronization and Deadlock management
- Memory management
- Mass storage management and File management
- Information management
- User management and security
- Networking services (wired + wireless)

Two methods by which a program can make requests for services from the Kernel:

- By making a system call
- By calling a library routine that makes use of this system call

Types of Operating Systems

Network Operating Systems

- In NOS, the users are aware of the existence of multiple computers and can log in to remote machines and copy files from one machine to another
- Each machine runs its own local OS and has its own local user(s)
- A NOS is a collection of s/w and associated protocols that allow a set of autonomous computers which are inter-connected by a computer NW to be used together in a convenient and cost effective manner
- **Remote Login.** Each user normally works on his/her own system; using a different system requires some kind of remote login, instead of having the OS dynamically allocate processes to CPU

  telnet *pucit.edu.pk*
- **Remote File Transfer.** Users are aware of where their files are kept and must move file from one system to another with explicit file transfer commands instead of having file placement managed by OS

  ftp *pucit.edu.pk*

Distributed OS

- A Distributed System is a collection of independent computers that appears to its users as a single coherent system
- The users should not be aware of where their programs are being run or where their files are located; that should all be handled automatically and efficiently by the Operating System

"A distributed system is one where I can't do work because some machine (in some other part of the world) I have never heard of isn't working"

Lamport

NOS vs DOS

- **File System.** In NOS, control over file placement must be done manually by the user, where as in a DOS it is done automatically by the system itself
- **Protection.** In NOS, there are various machines, each with its own user to UID, but in DOS there is a single system wide mapping that is valid every where
- **Program Execution.** In the most distributed case the system chooses a CPU by looking at the processing load of the machine, location of file to be used etc. In the least distributed case, the system always run the process on one specific machine. (usually the machine on which the user is logged in)

Real Time Systems

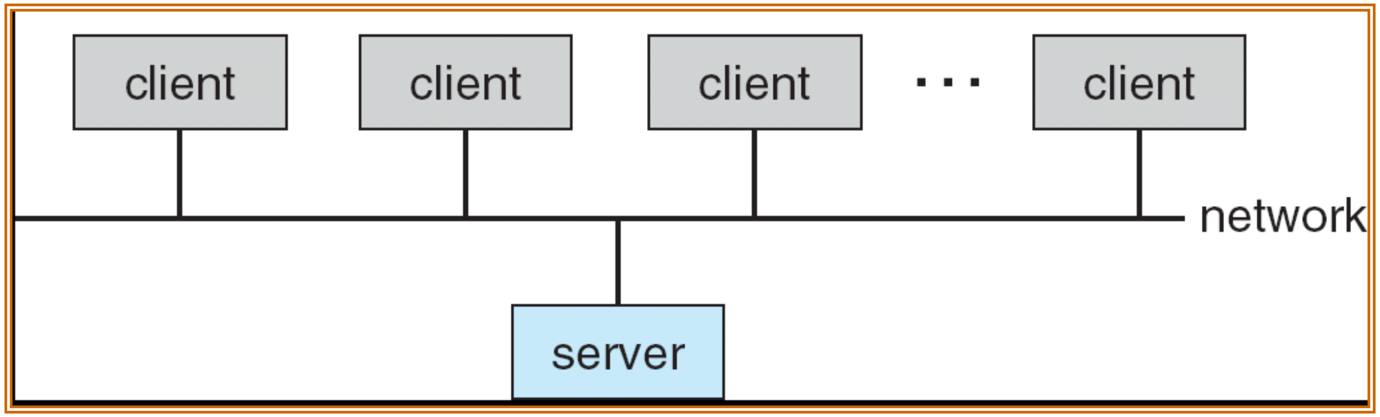
- In Real Time systems, the correctness of the system depends not only on the logical result of computation but also on the time at which the results are produced
- Examples are Process Control Plants, Robotics, Air Traffic Control, Telecommunications, Missile Control System
- **Hard Real Time System.** Output should be produced within the given time constraints, otherwise, the result is life threatening; e.g. Plane landing systems, process control in nuclear power plants, missile control system. Secondary storage is limited or absent, data stored in short term memory or ROM. No virtual memory
- **Soft Real Time System.** Output should be produced within the given time constraints, but if it is not, the result is not life threatening; e.g. applications of multimedia and virtual reality

Computing Environment

- Client Server computing
- Peer to Peer computing
- Web based computing
- Multi-processor computing
  - SMP architecture
  - Asymmetric computing
- Clustered computing
- Embedded computing

Computing Environment (…)

- **Client-Server Computing**
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server** provides an interface to client to request services (i.e. database)
    - **File-server** provides interface for clients to store and retrieve files



Computing Environment (…)

- **Peer to Peer Computing**
- Another model of distributed system
- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - Registers its service with central lookup service on network, or
    - Broadcast request for service and respond to requests for service via *discovery protocol*
  - Examples include ***Napster*** , ***KaZaA***, and ***Gnutella***

Computing Environment (…)

Web Based Computing

- Web has become ubiquitous
- PCs most prevalent devices
- More devices becoming networked to allow web access
- New category of devices to manage web traffic among similar servers: **load balancers**
- Use of operating systems like Windows 95, client-side, have evolved into Linux and Windows 7, which can be clients and servers

Computing Environment (…)

Multiprocessor Systems

- Also called parallel systems or tightly coupled systems
- Multiple CPUs in a single entity, comm using system bus
- **Advantages**
  - Increased throughput
  - Increased reliability (Graceful degradation, fault tolerance)
  - Saves money by not duplicating power supplies, housings, and peripherals
- **Disadvantage**
  - More complex in both hardware and software than uni-processor systems

Computing Environment (…)

Types of Multiprocessor Systems

- **Symmetric multiprocessors** Two or more identical processors connected to a single shared memory, having full access to all I/O devices, controlled by a single operating system instance that treats all processors equally (reserving none for special purpose) is called SMP architecture. SMP architecture treats multiple cores as separate processors

- **Asymmetric multiprocessors** Master slave scenario. Master distributes tasks among the slaves, and I/O is usually done by master only

Computing Environment (…)

Clustered Systems

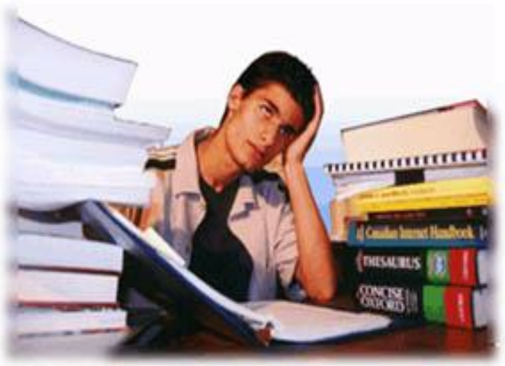
- Constructed by combining multiple computers into a single system to perform a computational task distributed across the cluster
- They share storage and are linked via LAN
- Clustered systems communicate using **messages**, while processors in a multiprocessor system could communicate using **shared memory**
- Used to provide high available service. Can be:
  - **Asymmetric Cluster** One machine is in hot standby mode, doing nothing, while other is running. Hot standby mode machine monitors the server. If server fails hot stand by host become active server
  - **Symmetric / Parallel Cluster** Two or more hosts are running an application and monitoring each other. More efficient as it utilizes all of the available hardware

Embedded OS

- **Pervasive Computing**
  - Cheap processors embedded every where
  - How many are on your body now? In your car?
  - Cell phones, PDAs, iPads, iPod…
- **Typically very constrained h/w resources**
  - Slow processors
  - Small amount of memory
  - No disk or tiny disk
  - Typically only one dedicated application
  - Limited power
- **But technology changes fast**
  - Embedded CPUs are getting faster
  - Storage is growing rapidly

SUMMARY

We're done for now, but Todo's for you after this lecture…



- Go through the slides and Book Sections: 1.1-1.12
- Go through Unix The Text Book Chapters 0 - 3
- Get a Secure Shell account in the Lab. Log in using your SSH account in lab and try executing some basic shell commands.
- Copy following resources from course web site
  - Video Lectures Series on Linux by Arif Butt
  - UNIX The Text Book
  - ITC Lecture Slides (Basic Linux Commands)
  - Basic Linux Tutorial
- If you are using MS Windows at your home PC, please dual boot it with some Linux distribution or install Linux in virtual environment using Virtual Box

If you have problems visit me in student counseling hours. . . .