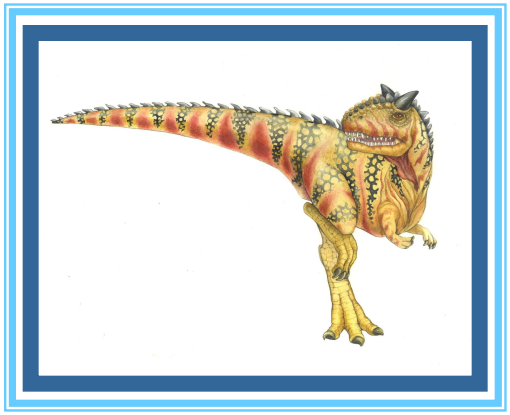
Lecture 1: Introduction

Lecture : 03



Recap

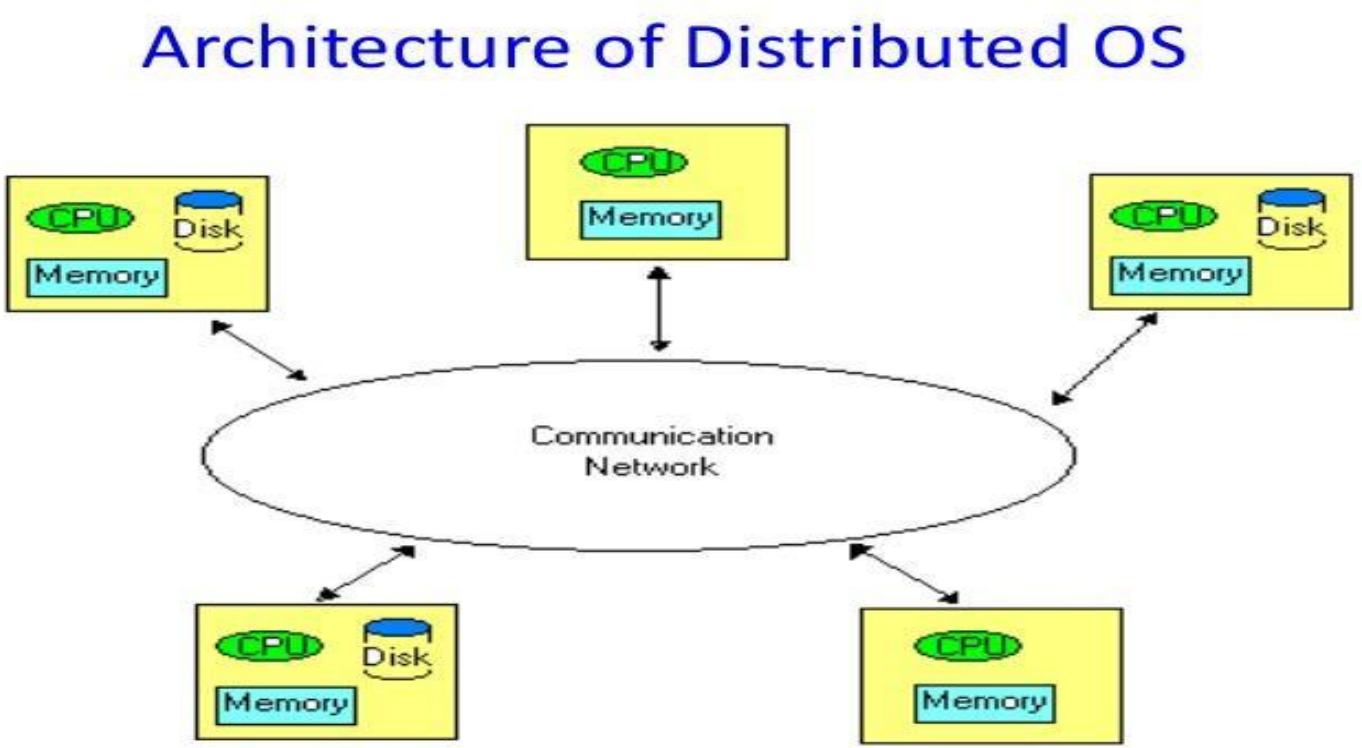
- What is an Operating System
- What Operating Systems Do
- Computer-System Organization
- Operating-System Structure
- Storage Structure
- Operating System Operations
- Operating System architecture

Objectives

- Distributed Systems
- Process Management
- Memory Management
- Storage Management

Distributed Systems

- A distributed system contains multiple nodes that are physically separate but linked together using the network. All the nodes in this system communicate with each other and handle processes in tandem.
- Communication is via a network. These systems are termed loosely-coupled or distributed systems. The processors vary in size and function and are called nodes.



Distributed Systems (contd.)

Some advantages of Distributed Systems are as follows –

- All the nodes in the distributed system are connected to each other. So nodes can easily share data with other nodes.
- More nodes can easily be added to the distributed system i.e. it can be scaled as required.
- Failure of one node does not lead to the failure of the entire distributed system. Other nodes can still communicate with each other.
- Resources like printers can be shared with multiple nodes rather than being restricted to just one.

Pipeline Processing

- Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as **pipeline processing**.
- Pipelining is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end.
- Pipelining increases the overall instruction throughput.
- In pipeline system, each segment consists of an input register followed by a combinational circuit. The register is used to hold data and combinational circuit performs operations on it. The output of combinational circuit is applied to the input register of the next segment.

Pipeline Processing

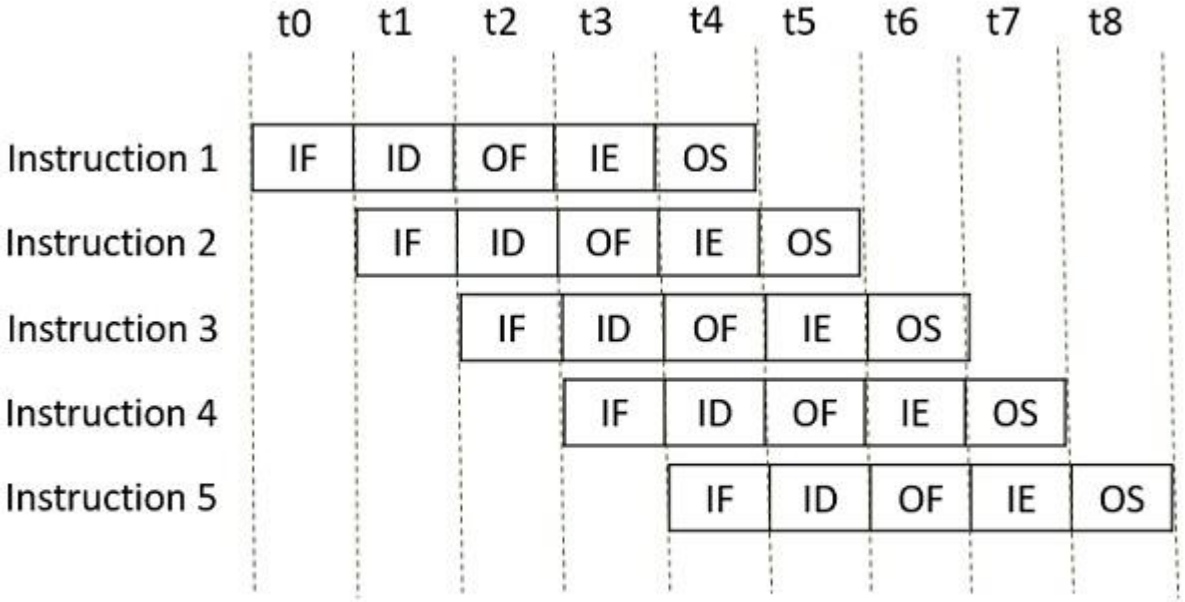
- An instruction in a process is divided into 5 subtasks likely,

Instruction Fetch	Instruction Decode	Operand Fetch	Instruction Execute	Operand Store

- In the first subtask, the instruction is fetched.
- The fetched instruction is decoded in the second stage.
- In the third stage, the operands of the instruction are fetched.
- In the fourth, arithmetic and logical operation are performed on the operands to execute the instruction.
- In the fifth stage, the result is stored in memory.

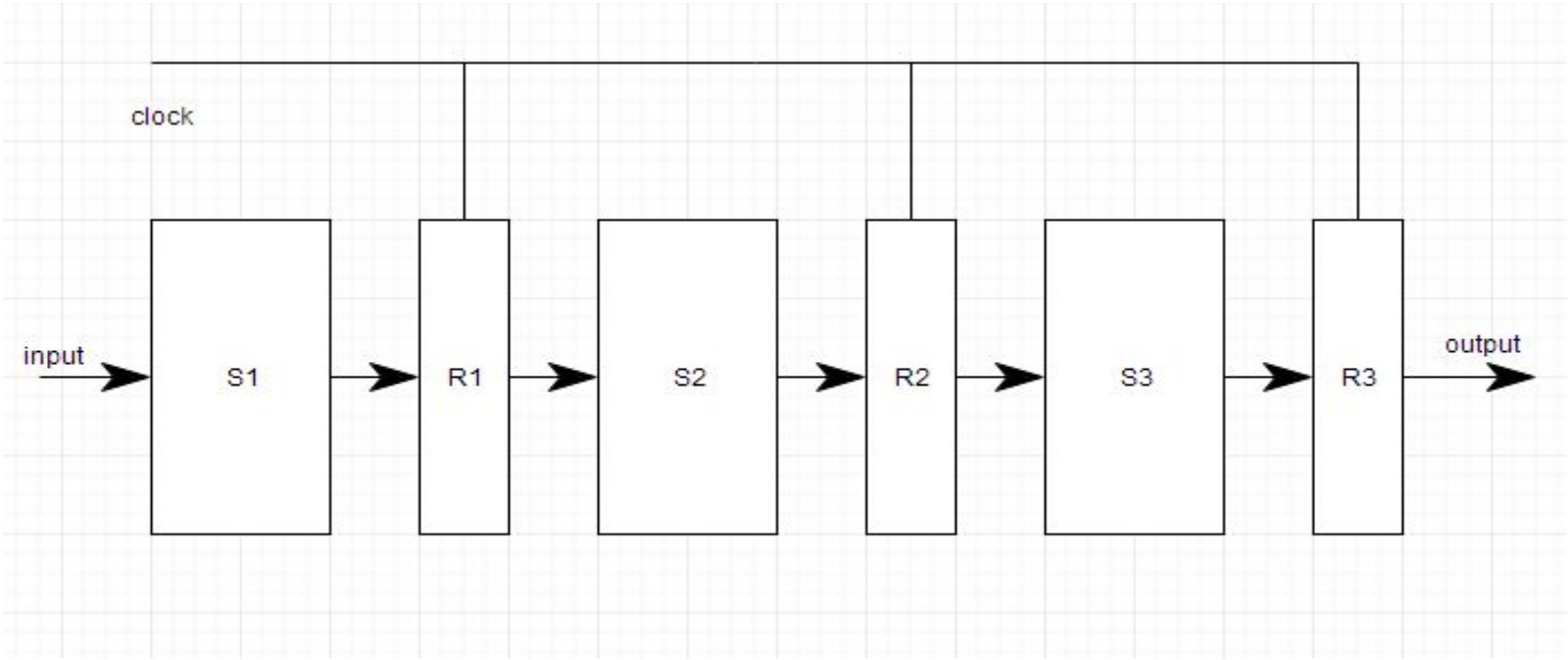
Pipeline Processing

- Now, understanding the division of the instruction into subtasks. Let us understand, how the n number of instructions in a process, are pipelined.
- Look at the figure below the 5 instructions are pipelined. The first instruction gets completed in 5 clock cycle. After the completion of first instruction, in every new clock cycle, a new instruction completes its execution.



Pipelining of 5 Instructions

Pipeline Processing



Pipeline system is like the modern day assembly line setup in factories. For example in a car manufacturing industry, huge assembly lines are setup and at each point, there are robotic arms to perform a certain task, and then the car moves on ahead to the next arm

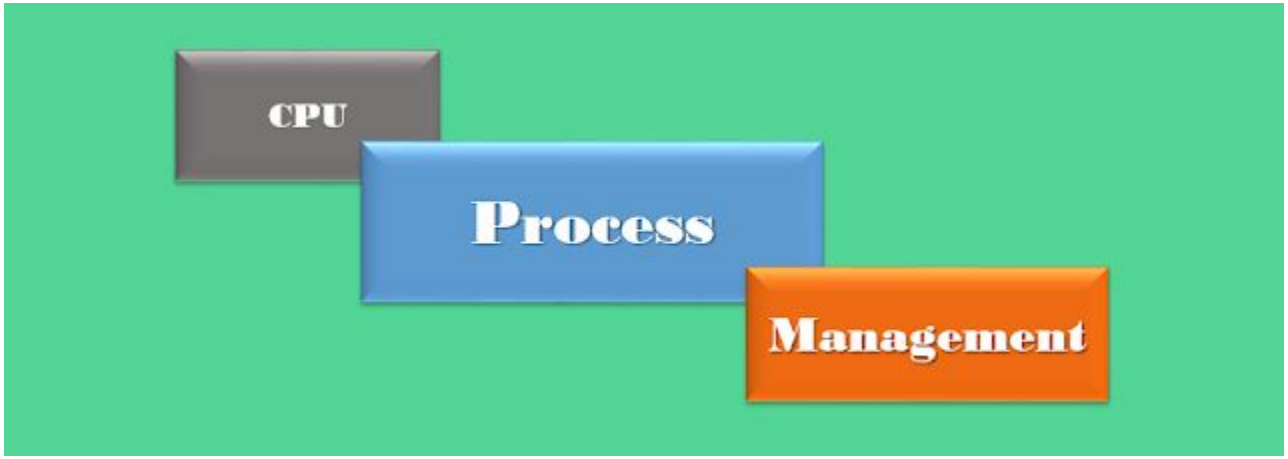
Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a ***passive entity***, process is an ***active entity***.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads

Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling



Memory Management

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

Mass-Storage Management

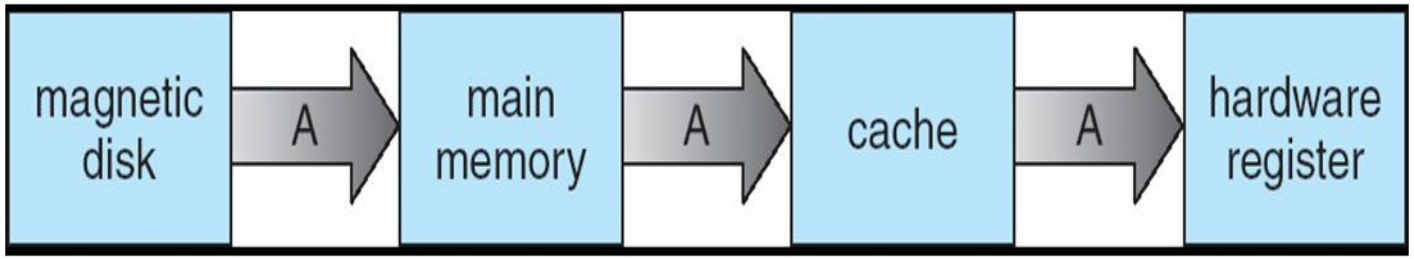
- Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast
  - Such storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications

Performance of Various Levels of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Migration of Integer A from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy



- Multiprocessor environment must provide cache coherency in hardware such that all CPUs have the most recent value in their cache
- Distributed environment situation even more complex
  - Several copies of a datum can exist
  - Various solutions covered in Chapter 17

I/O Subsystem

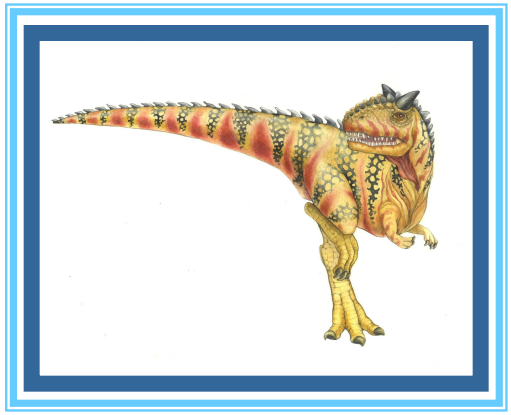
- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), and spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices

Protection and Security

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- The main difference between protection and security is that the protection focuses on internal threats in a computer system while security focuses on external threats to a computer system.
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file

Protection and Security

BASIC	SECURITY	PROTECTION
Basic	Provides the system access to the legitimate users only.	Controls the access to the system resources.
Handles	More complex concerns.	Quite simple queries.
Policy	Describes which person is allowed to use the system.	Specifies what files can be accessed by a particular user.
Type of threat involved	External	Internal



Hardware Protection

- Dual-Mode operation
- I/O Protection
- Memory Protection
- CPU Protection

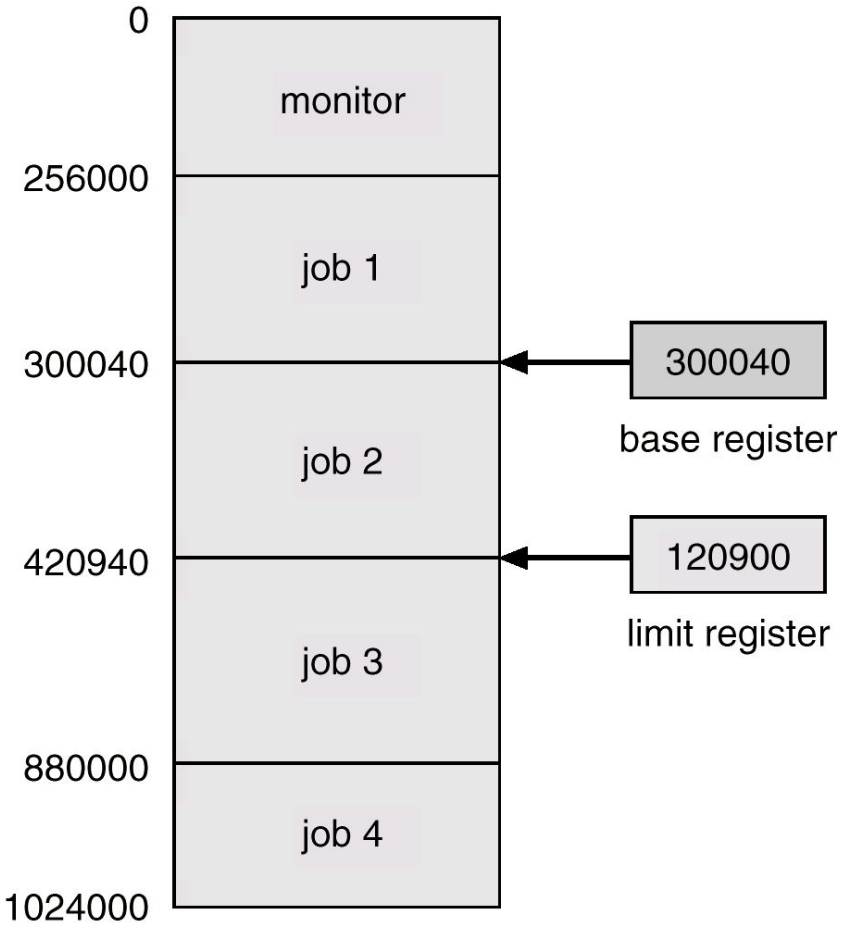
I/O Protection

- All I/O instructions are privileged instructions.
- **A user process might disrupt normal operation of the system by issuing the illegal I/O instructions, by accessing memory locations and addresses within the operating system itself, or by** refusing to surrender CPU. We can use of several mechanisms to ensure that such disruptions should not take place in the system.
- To prevent the users from performing the illegal I/O, we define all the I/O instructions to be as privileged instructions. Therefore users cannot issue the I/O instructions directly; they should do it by making use of the operating system.
- For I/O protection and security to be complete, we should be sure that the user program can never gain control of the computer in the monitor mode. If it could, then the I/O protection could be compromised.

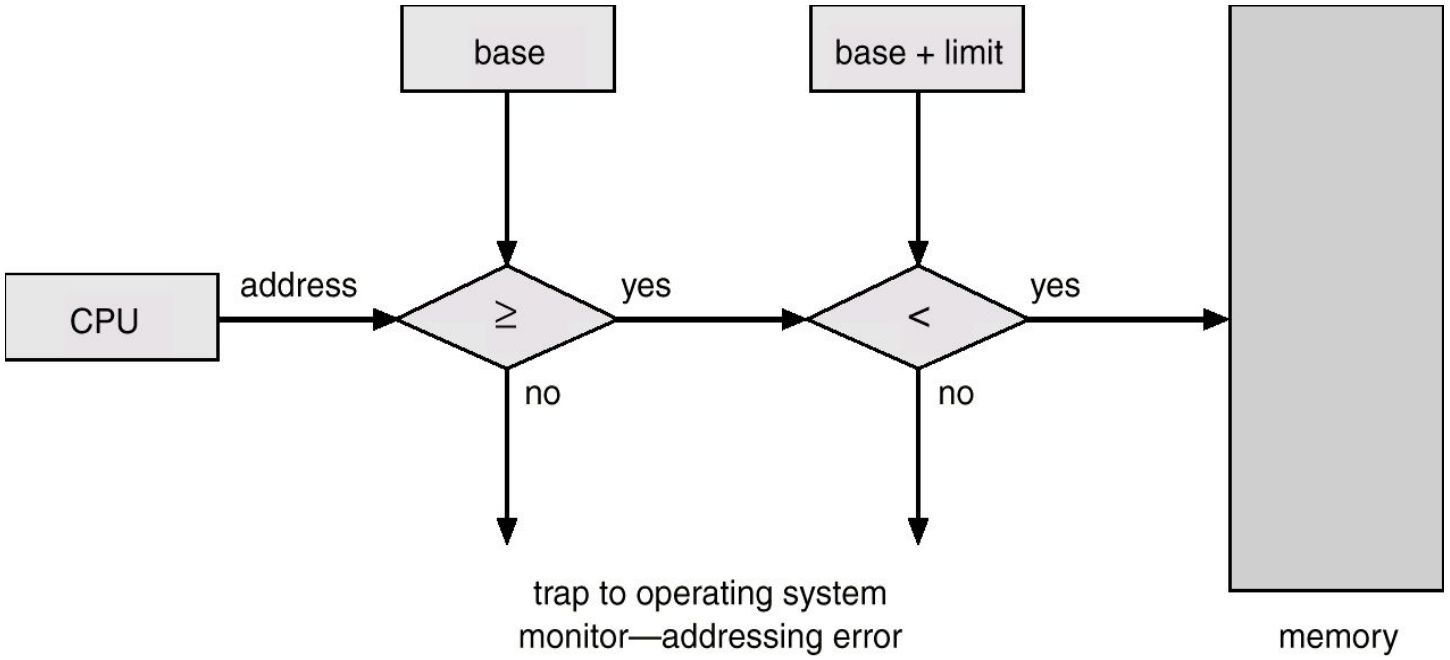
Memory Protection

- Must provide memory protection at least for the interrupt vector and the interrupt service routines.
- Memory protection is a way to control memory access rights on a computer, and is a part of most modern operating systems.
- The main purpose of memory protection is to prevent a process from accessing memory that has not been allocated to it. This prevents a bug within a process from affecting other processes, or the operating system itself
- In order to have memory protection, add two registers that determine the range of legal addresses a program may access:
  - **base register** – holds the smallest legal physical memory address.
  - **Limit register** – contains the size of the range
- Memory outside the defined range is protected.
- **Base and limit registers** are special hardware **registers**. When a process is run, the **base register** is loaded with the physical location where the process begins in memory. The **limit register** is loaded with the length of the process. In other words, they define the logical address space.

A Base And A limit Register Define A Logical Address Space



Protection Hardware



- When executing in monitor mode, the operating system has unrestricted access to both monitor and user's memory.
- The load instructions for the *base* and *limit* registers are privileged instructions.

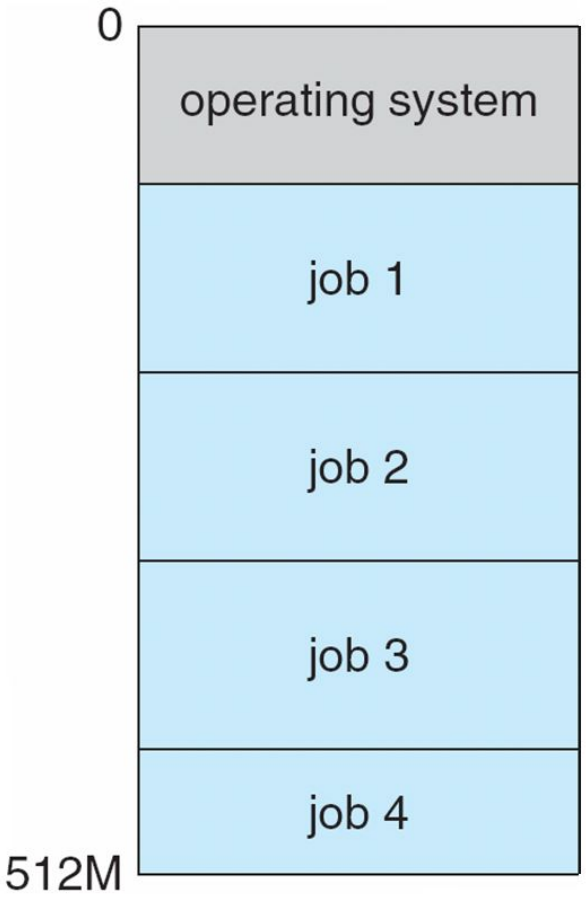
CPU Protection

- *Timer* – interrupts computer after specified period to ensure operating system maintains control.
  - Timer is decremented every clock tick.
  - When timer reaches the value 0, an interrupt occurs.
- CPU protection is needed to prevent a user program from getting stuck in an infinite loop and never returning control to the O/S.
- A timer is used to prevent this. The timer is set to interrupt, say every *N* msecs. The O/S then switches the CPU to another process in a multitasking O/S.
- Loading/setting a timer is a privileged instruction.

Operating System Structure

- **Multiprogramming** (**Batch system**) needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) 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 program executing in memory □ **process**
  - If several jobs ready to run at the same time □ **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

Memory Layout for Multiprogrammed System



Computing Environments - Mobile

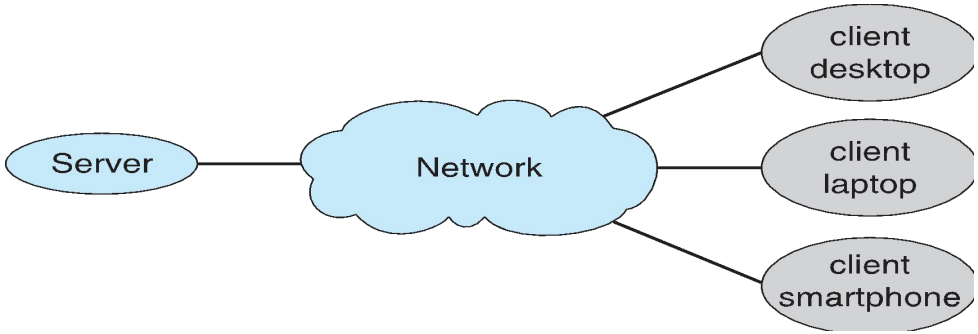
- Handheld smartphones, tablets, etc
- What is the functional difference between them and a “traditional” laptop?
- A technology that is capable of providing an environment which enables users to transmit data from one device to other device without the use of any physical link/cables is known as Mobile Computing.
- It means, data transmission is done wireless-ly with the help of wireless devices such as mobiles, laptops etc.
- Whenever any device is connected to a network without being connected physically over a link or cable, data transmission such as messages, voice recording, videos etc. can be done be done by using the concept of mobile computing.
- Mobile Computing technology helps users to access and transmit data from any remote locations without being present there physically.
- Thus, having such a big coverage diameter, it is one of the fastest and most reliable sectors of computing technology field.

Computing Environments – Distributed

- Distributed computing
  - Collection of separate, possibly heterogeneous, systems networked together
- A distributed computing environment contains multiple nodes that are physically separate but linked together using the network. All the nodes in this system communicate with each other and handle processes in cycle. Each of these nodes contains a small part of the distributed operating system software.
  - **Network** is a communications path, **TCP/IP** most common
    - **Local Area Network** (**LAN**)
    - **Wide Area Network** (**WAN**)
    - **Metropolitan Area Network** (**MAN**)
    - **Personal Area Network** (**PAN**)
  - **Network Operating System** provides features between systems across network
    - Communication scheme allows systems to exchange messages
    - Illusion of a single system

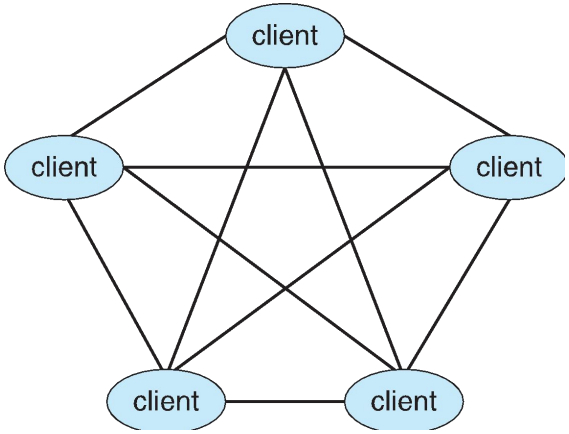
Computing Environments – Client-Server

- Client-Server Computing
- A Computer networking model where one or more powerful computers (servers) provide the different computer network services and all other user'of computer network (clients) access those services to perform user's tasks is known as client/server computer networking model.
- In such networks, there exists a central controller called server. A server is a specialized computer that controls the network resources and provides services to other computers in the network.
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system** provides an interface to client to request services (i.e., database)
    - **File-server system** provides interface for clients to store and retrieve files



Computing Environments - Peer-to-Peer

- 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
  - Examples include Napster and Gnutella, **Voice over IP** (**VoIP**) such as Skype



Computing Environments - Virtualization

- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
- **Virtualization** is the process of running a virtual instance of a computer system in a layer abstracted from the actual hardware. It refers to running multiple operating systems on a computer system simultaneously. Apple laptop running Mac OS X host, Windows as a guest
- Virtual Memory is a space where large programs can store themselves in form of pages while their execution and only the required pages or portions of processes are loaded into the main memory. This technique is useful as large virtual memory is provided for user programs when a very small physical memory is there.
  - Developing apps for multiple OSes without having multiple systems
  - QA testing applications without having multiple systems
  - Executing and managing compute environments within data centers

Computing Environments – Cloud Computing

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for it functionality.
  - Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage
- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company's own use
  - **Hybrid cloud** – includes both public and private cloud components
  - Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e., word processor), software that's available via a third-party over the internet.
  - Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e., a database server), hardware and software tools available over the internet.
  - Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e., storage available for backup use). cloud-based services, pay-as-you-go for services such as storage, networking, and virtualization.

Computing Environments – Real-Time Embedded Systems

- Real-time embedded systems most prevalent form of computers
- An Embedded System is more of an application oriented system i.e. it is dedicated to perform a single task (or a limited number of tasks, but all working for a single main aim).
  - Vary considerable, special purpose, limited purpose OS, **real-time OS**
- A Real Time Embedded System is a type of computer system with timing constraints i.e. a system which responds to external events or input stimuli in a timely fashion (within finite and specified time).
- Real-time OS has well-defined fixed time constraints
  - Processing ***must*** be done within constraint
  - Correct operation only if constraints met

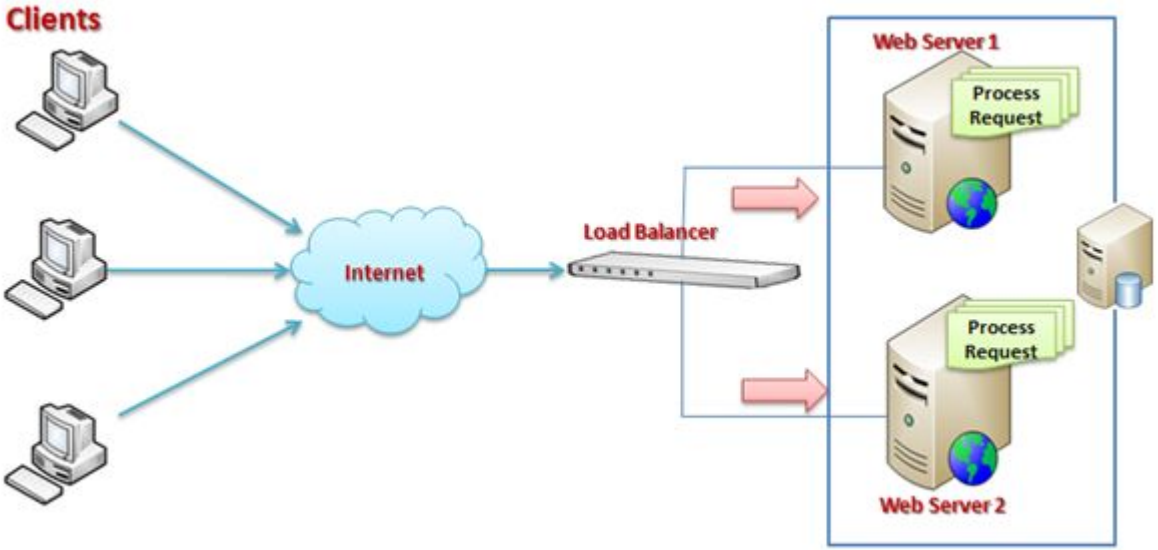
Open-Source Operating Systems

- Open Source operating systems are released under a license where the copyright holder allows others to study, change as well as distribute the software to other people. This can be done for any reason.
- Started by **Free Software Foundation (FSF)**, which has "copyleft" **GNU Public License (GPL)**
- Examples include **GNU/Linux** and **BSD UNIX** (including core of **Mac OS X**), and many more

Web-Based Computing

- Web has become ubiquitous
- PCs used to be more prevalent devices but now mobile devices (e.g. smart phones and tablets) are more prevalent modes of access
- Now **load balancers** are used to manage web traffic among similar servers
- Use of operating systems like Windows 95, client-side, have evolved into Linux and Windows XP, which can be clients and servers

Web-Based Computing



Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary closed-source
- Counter to the copy protection and Digital Rights Management (DRM) movement
- Started by Free Software Foundation (FSF), which has “copyleft” GNU Public License (GPL)
- Examples include GNU/Linux, BSD UNIX (including core of Mac OS X), and Sun Solaris

End of Chapter 1