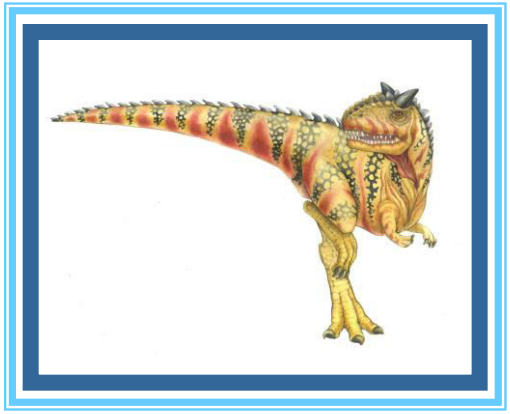
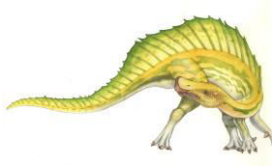
Introduction



Computer System Architecture



Single Processor Systems

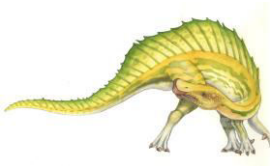
- *Most computer systems use **a single general-purpose processor**.*
- *On a single processor system there is **one main CPU capable** of executing a general purpose instruction set, including instructions from user processes.*
- *Almost all single processor systems have other special-purpose processor as well. They may come in the form of device specific processors such as disk, keyboard etc. All these special purpose processors run a limited instruction set and do not run user processes.*

Multiprocessor Systems

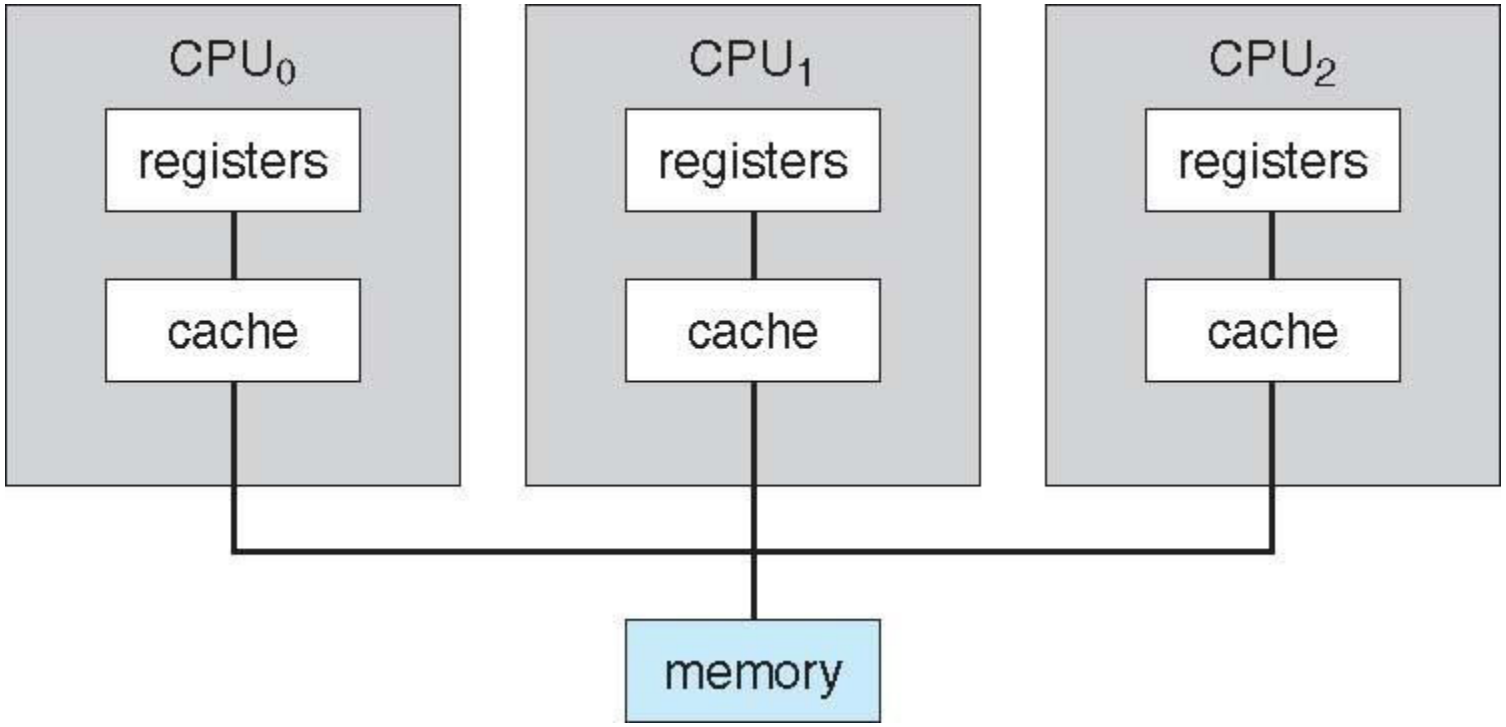
- *Such systems have **two or more processors** sharing the computer bus, memory and peripheral devices. Also known as **parallel systems, multicore systems**.*
- *Advantages include:*
  - *Increased throughput*
  - *Economy of scale*
  - *Increased reliability – graceful degradation or fault tolerance*

Multiprocessor Systems

- ***Graceful degradation:*** *the ability to continue providing service proportional to the level of surviving hardware is called graceful degradation.*
  - *Or*
- ***Graceful degradation*** *is the ability of a computer system to maintain limited functionality.*



Multiprocessor Systems



Multiprocessor Systems

- *Systems designed for graceful degradation are called* ***fault tolerant***

  Fault tolerance *refers to the ability of a system to continue operating without interruption when one or more of its components fail.*

Two types

- ***Asymmetric Multiprocessing*** *– boss-worker relationship (In this type of system, one processor behaves as a master and the other processors behave as slaves.) SUNOS version 4*
- ***Symmetric Multiprocessing*** *– All processors are peers no boss worker relationship exist between processors. EX: IBM's Advanced Interactive eXecutive, or AIX*

Symmetric & Asymmetric

- *In **Symmetric Multiprocessing**, processors shares the same memory.*
- *In **Asymmetric Multiprocessing** there is a one master processor that controls the data structure of the system and each processor has private memory and shares I/O devices through a common bus..*

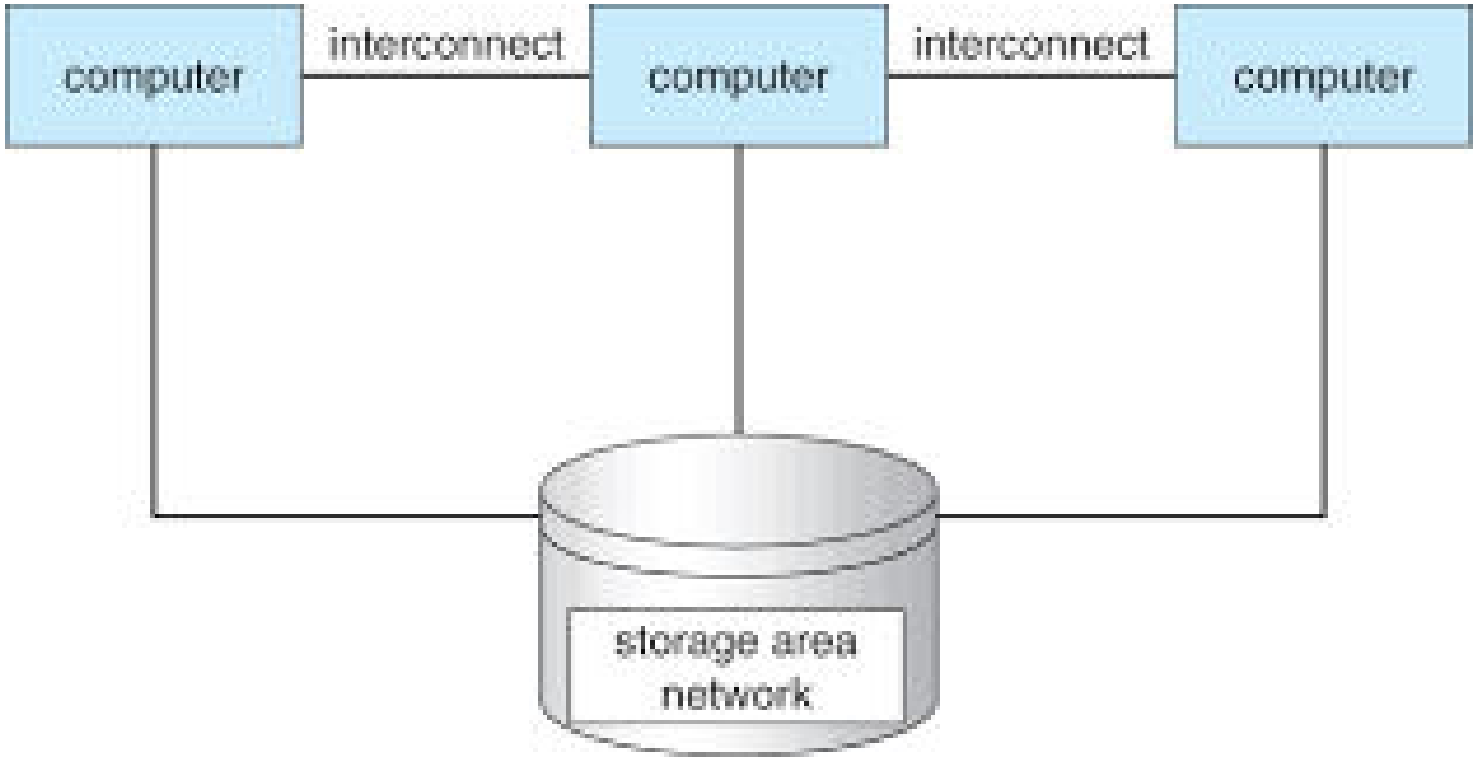
Clustered Systems

- *Another type of **multiple-CPU system** is the clustered system. Like multiprocessor systems, clustered systems gather together multiple CPUs to accomplish computational work. Composed of two or more individual systems or nodes joined together.*
- *The generally accepted definition is that clustered computers share storage and are closely linked via a local-area network (LAN) or a faster interconnect.*
- *Clustering is usually used to **provide high-availability** service; that is, service will continue even if one or more systems in the cluster fail.*
  - ***Asymmetric clustering** has one machine in hot-standby mode.*
  - ***Symmetric clustering** has multiple nodes running applications, monitoring each other*

Clustered Systems

- ***Asymmetric clustering*** *In asymmetric clustering, one machine is in hot-standby mode while the other is running the applications. The hot-standby host machine does nothing but monitor the active server. If that server fails, the hot-standby host becomes the active server.*
- ***Symmetric mode*** *two or more hosts are running applications, and are monitoring each other.*
- *Some clusters are for* ***high-performance computing (HPC)***
- *Applications must be written to use* ***parallelization.***
- *Some have* ***distributed lock manager*** *(****DLM****) to avoid conflicting operations*

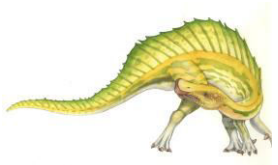
Clustered Systems



Operating System Structure

Multiprogramming *needed for efficiency*

- *Single process cannot keep CPU and I/O devices busy at all times*
- *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*
- *The jobs are kept initially on the disk in the job pool. This job pool consists of all processes residing on disk awaiting allocation of MM.*



Multiprogramming Operating system

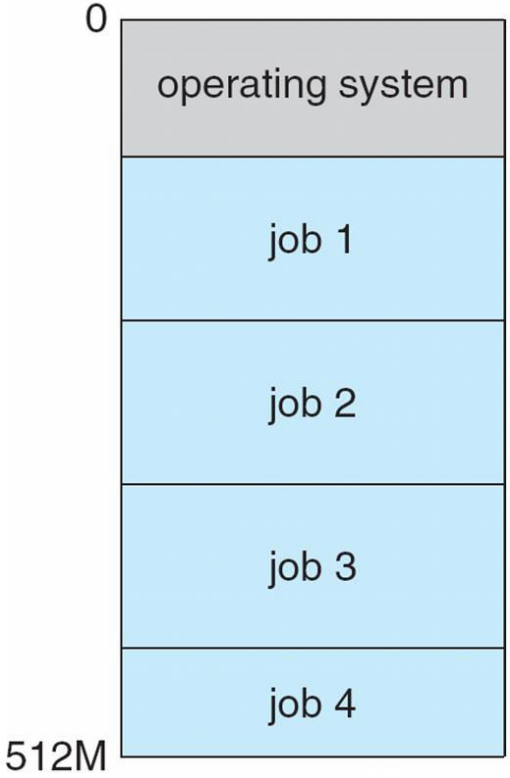
- *Maximize **CPU utilization**, **No IDLENESS*** for CPU.
- ***More than one program resides in main memory*** which are ready to execute.
- Process generally requires ***CPU time*** as well as ***I/O time***. So if running process performs any ***I/O or other task*** which do no not require CPU, the processor does not stay idle, instead it will make ***context switch*** and ***will run another program***.
- ***CPU will never sit in idle*** state unless there is no process ready to execute or during context switch.

Multiprogramming Operating system

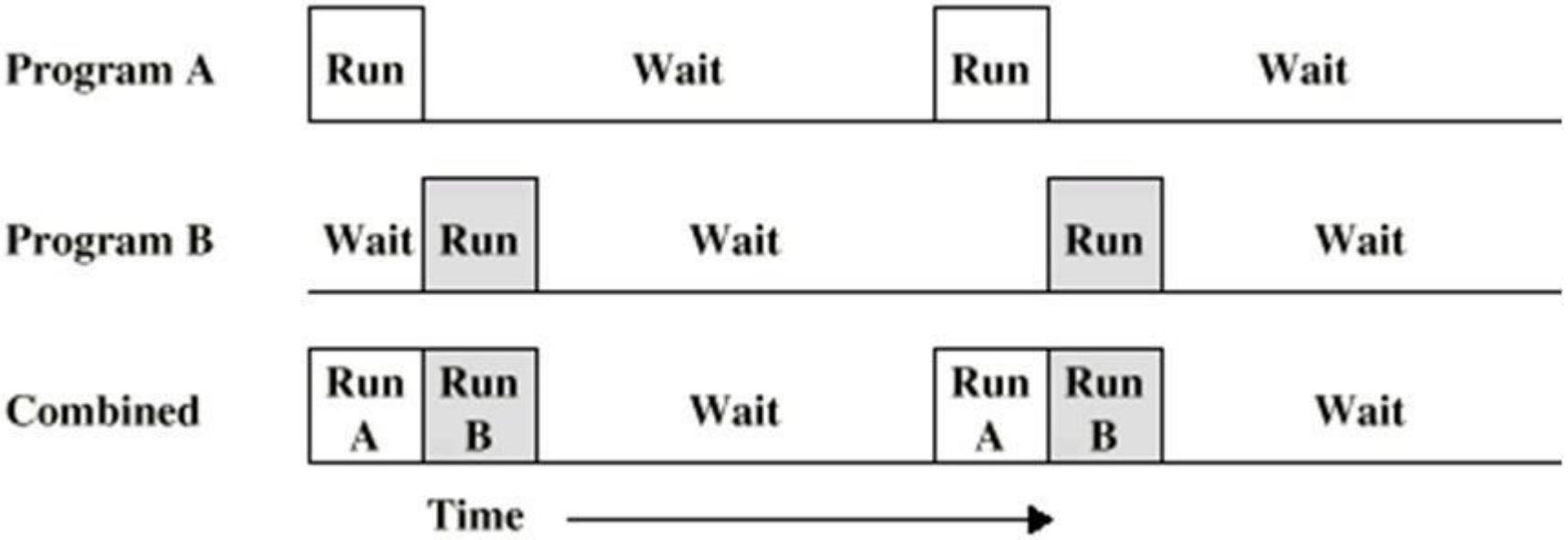
- ***Advantages:***
- *High CPU utilization.*
- *Less waiting/response time.*

- ***Disadvantages:***
- *Difficult scheduling*
- *Main memory management required*
- *Main memory fragmentation*
- *Paging*

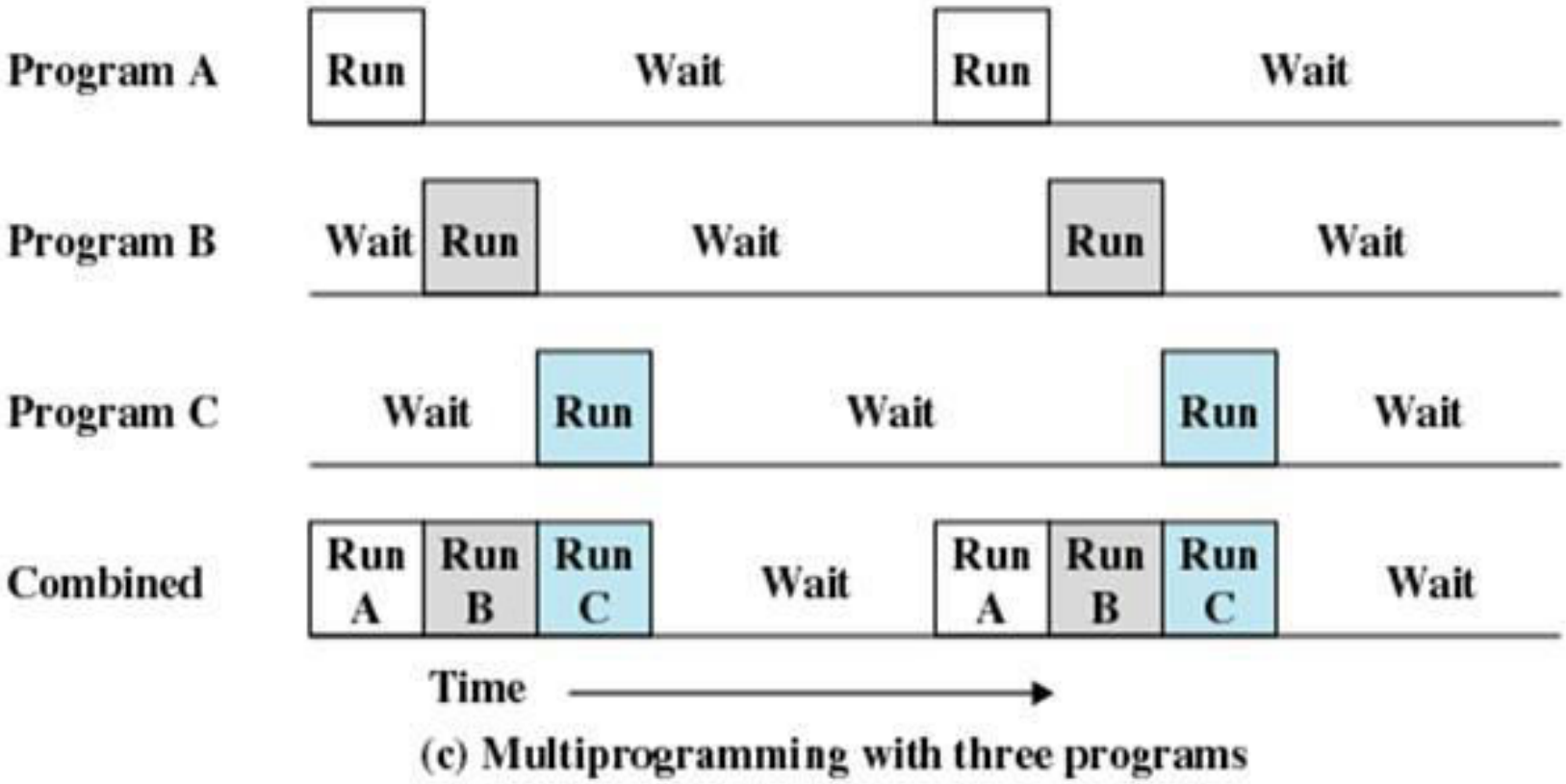
Memory Layout for Multiprogrammed System



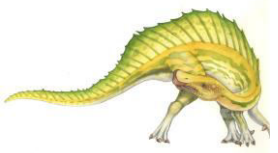
Multiprogramming Operating system



Multiprogramming Operating system

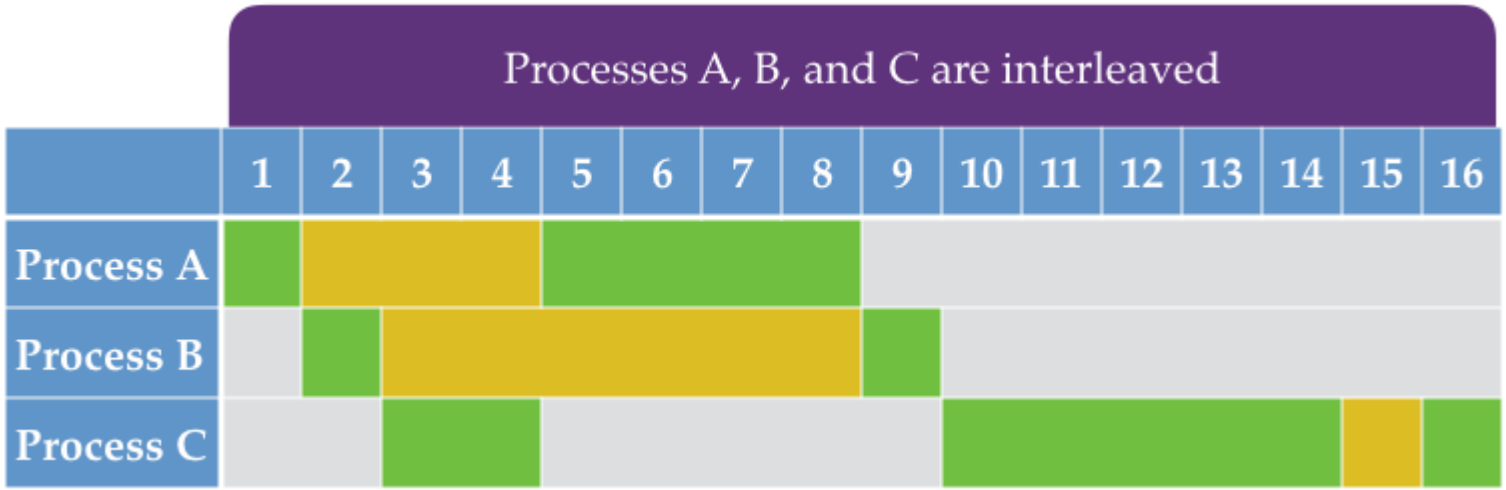


(c) Multiprogramming with three programs



Multiprogramming Operating system

Consider the following timeline illustration for the same three processes from the previous example, but in a multiprogramming environment.



Multiprogramming Operating system

$$CPU\ utilization = \frac{5+2+8}{5+2+8+1} = \frac{15}{16} = 93.75\%$$

Multiprogramming Operating system

Consider two jobs JOB1 and JOB2. JOB1 runs in a loop for 100 iterations that requires 3 seconds of cpu time, followed by 6 seconds of I/O to disk, followed by 3 seconds of cpu time. JOB2 runs in a loop for 100 iterations that requires 3 seconds of cpu time followed by 2 seconds of disk I/O.

- ***How long would it take to run the jobs consecutively in non multiprogramming mode?***
- ***How would this improve if multiprogramming was allowed?***

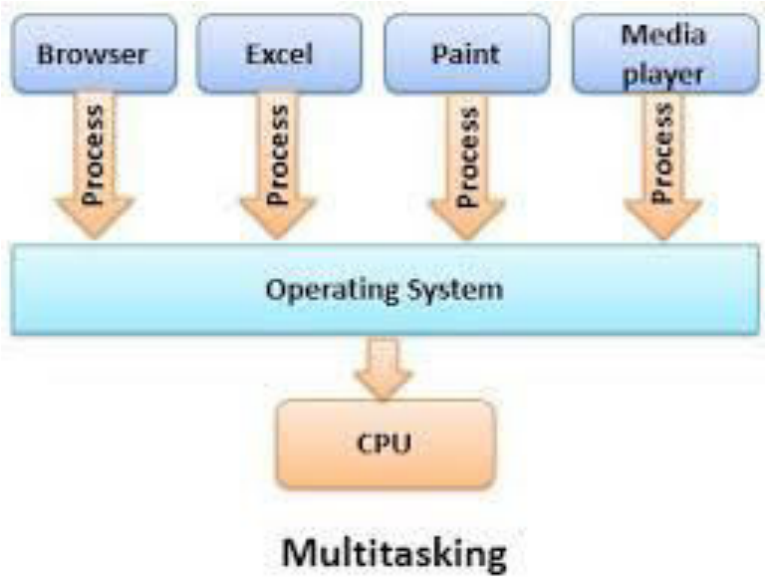
In non multiprogramming mode ***Job 1 takes 3+6+3= 12 seconds***

Job 2 takes 3+2= 5 seconds

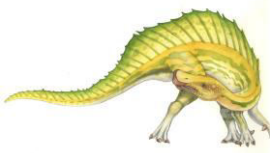
- It would take ***17 seconds*** together for both of them to complete consecutively in batch mode.
- If multiprogramming was allowed, job 2 would finish while job 1 is running and so it would take a total of only ***12 seconds*** to finish both the jobs.

Multitasking/Time sharing/Fare share/Multitasking with round robin OS

- Multitasking is ***multiprogramming with time sharing***.
- Main objective is ***Responsiveness***.
- CPU is one but switches between processes so quickly that it gives illusion that all executing at the same time.
- ***Main idea is*** better response time and executing multiple process together.

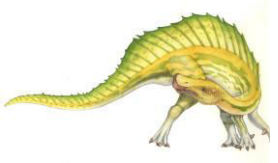


Multitasking



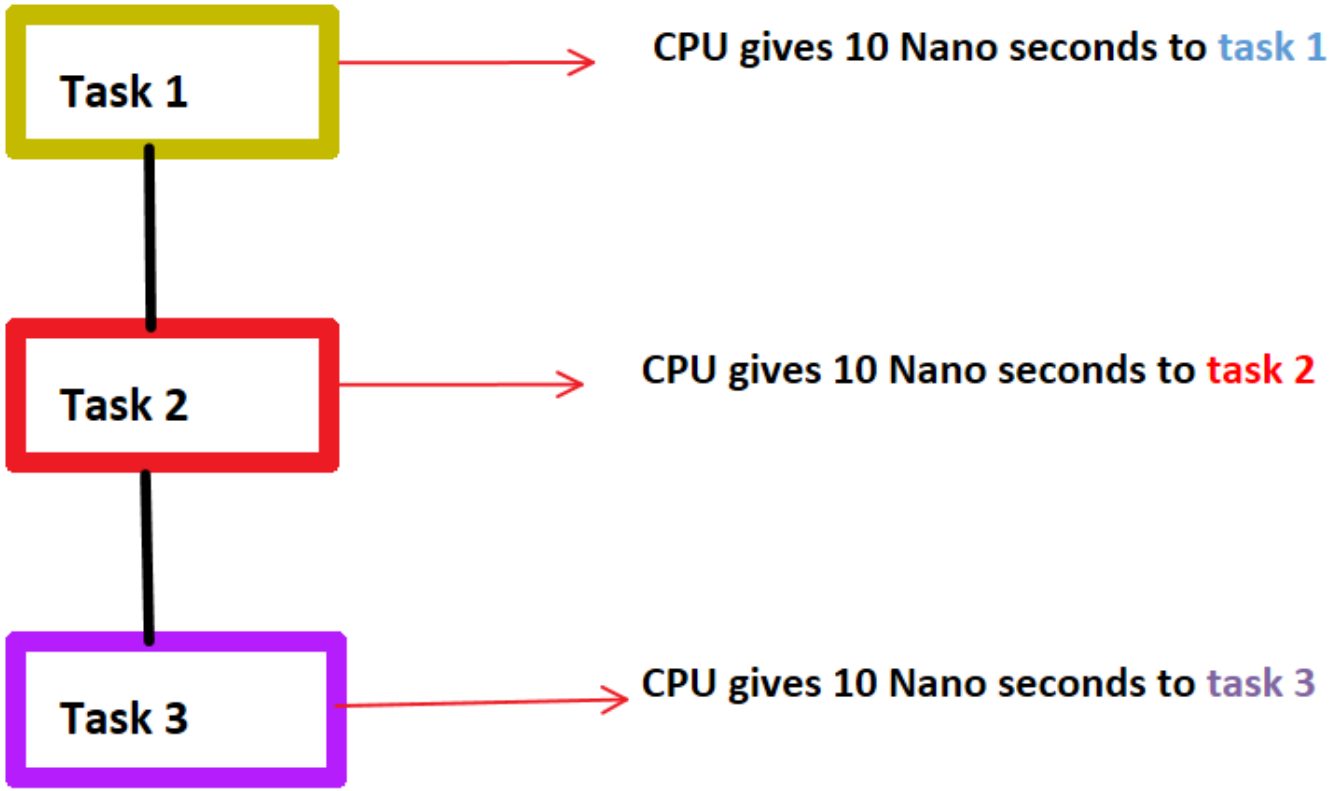
Multitasking/Time sharing/Fare share/Multitasking with round robin OS

- ***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*



Multitasking Operating system

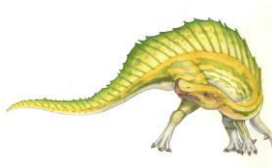
•



Multi-Tasking In Operating System

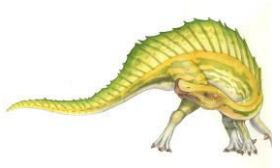
- .

In multiprogramming context switching is used and in multitasking, time-sharing is used.



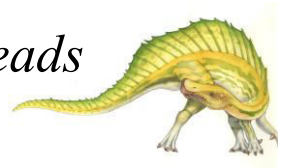
Timer

- *We must prevent a user program from getting **stuck** in an infinite loop and **never returning** control to the operating system.*
- *To accomplish this goal, we can use a **timer**. A timer can be set to interrupt the computer after a specified period.*
- *Timer is set to interrupt the computer after some **time period***
  - *Keep a counter that is **decremented** by the physical clock.*
  - *Operating system set the counter **(privileged instruction)***
  - *When counter zero generate an interrupt*
  - *Set up before scheduling process to regain control or terminate program that exceeds allotted time*



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*
- *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

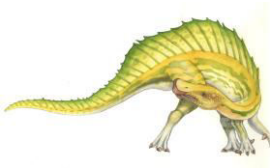
The operating system is responsible for the following activities in connection with process management:

- *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

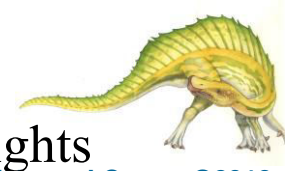
- *To execute a program all (or part) of the instructions must be in memory.*
- *All (or part) of the data that is needed by the program must be in memory.*
- *Memory management determines what is in memory and when it came*

  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*



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 of attacks possible including denial-of-service, viruses, identity theft, theft of service etc.
- 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
- Privilege escalation allows user to change to effective ID with more rights



Mass Storage Management

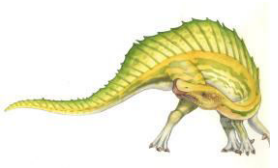
- *OS provides uniform, logical view of information storage: Abstracts physical properties to logical storage unit - **file***
- ***File-System management***
  - *Files usually organized into directories*
  - *Access control on most systems to determine who can access what*
  - *OS activities include*
    - *Creating and deleting files and directories*
    - *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*



Mass Storage Management

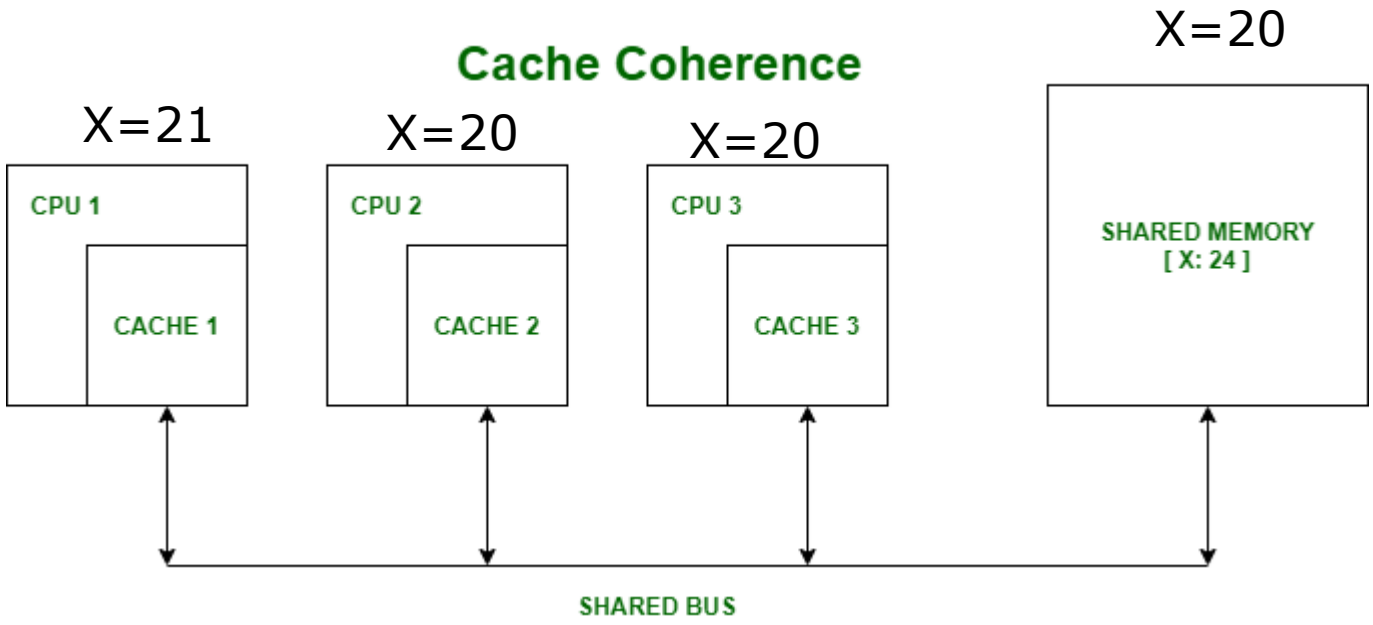
Caching: *Caching is a process that stores multiple copies of data or files in a temporary storage location—or cache—so they can be accessed faster. ... For example, when a user visits a website for the first time, an application or browser retains information to help them access it faster and more efficiently.*

Cache coherence :

In a multiprocessor system, data inconsistency may occur among adjacent levels or within the same level of the memory hierarchy.

In a shared memory multiprocessor with a separate cache memory for each processor, it is possible to have many copies of any one instruction operand: one copy in the main memory and one in each cache memory. When one copy of an operand is changed, the other copies of the operand must be changed also.

Mass Storage Management



Memory Management

- *Some storage need not be fast*
  - *Tertiary storage includes optical storage, magnetic tape*
  - *Still must be managed – by OS or applications*
  - *Varies between WORM (write-once, read-many-times) and RW (read-write)*

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), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)*
- *General device-driver interface*
- *Drivers for specific hardware devices*

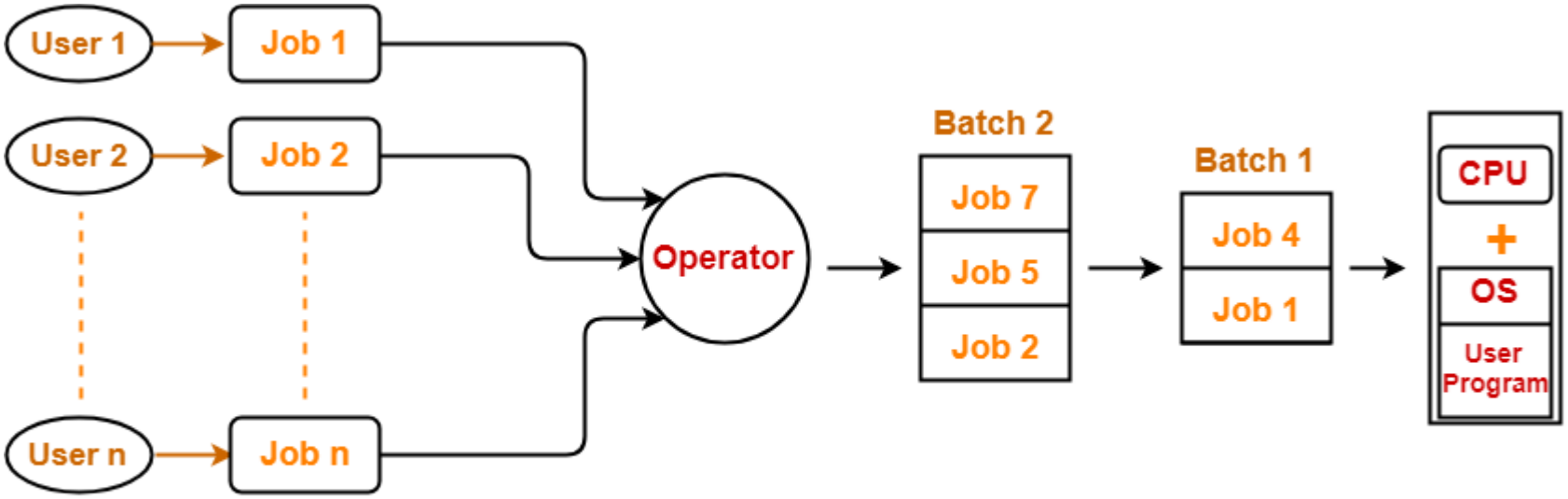
Computing Environments - Traditional

- Stand-alone general purpose machines connected to the network. Portability was achieved by use of laptop computers.
- Just few years ago, fewer remote access and portability options.
- Web technologies and increasing WAN bandwidth are stretching the boundaries of traditional computing.
- Companies establish **Portals,** which provides web access to their internal servers.
- **Network computers** (**thin clients**) are like Web terminals are used in place of traditional workstations where more security and easier maintenance is required.
- Mobile computers interconnect via **wireless networks** and cellular data networks to use the company's web portal.
- Networking becoming ubiquitous – even home systems use **firewalls** to protect home computers from Internet attacks

Computing Environments - Traditional

- Networking becoming ubiquitous – even home systems use **firewalls** to protect home computers from Internet attacks.
- For a period of time, systems were either batch or interactive. Jobs with ***similar needs are batched together*** and executed through the processor with predetermined input from file or other data source.
- Interactive systems waited for input from users.
- To optimize the use of computing resources multiple users shared time on these systems.
- Time sharing systems used a timer and scheduling algorithm to cycle processes rapidly through the CPU, giving each user share of the resources.

Batch Operating System

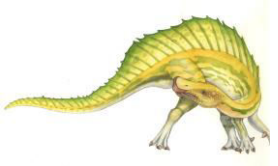


Batch Operating System

Batch Operating System

Advantages-

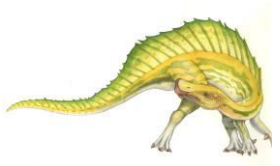
- It saves the time that was being wasted earlier for each individual process in ***context switching from one environment to another environment.***
- No ***manual intervention*** is needed.



Batch Operating System

Disadvantages-

- All the jobs of a batch are ***executed sequentially*** one after the another and the output is obtained only after all the jobs are executed. Thus, ***priority can not be implemented*** if a certain job has to be executed on an urgent basis.
- The jobs of a particular batch might take long time for their execution. This might lead to ***starvation*** to other jobs in other batches.
- If the jobs of a batch require some ***I/O*** operation, then CPU must ***wait*** till the I/O operation gets completed, since I/O devices are very slow, CPU remains idle for a long time.

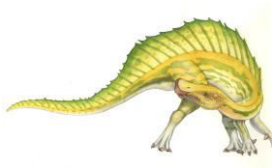


Mobile Computing

- Mobile computing refers to computing on handheld smartphones and tablet computers. These device distinguishing physical features of being portable and lightweight.
- Historically, compared with desktop and laptop computers, mobile systems gave up screen size, memory capacity, and overall functionality in return for handheld mobile access to services such as email and web browsing.
- Over the past few years, however, features of mobile devices have been so rich that the distinction in functionality between, say, a consumer laptop and a tablet computer may be difficult to discern.
- In fact, we might argue that the features of a contemporary mobile device allow it to provide functionality that is easier unavailable or impractical on a desktop or a laptop computer.

Mobile Computing

- Today, mobile systems are used not only for e-mail and web browsing but also for playing music and video, reading digital books, taking photos, and recording high-definition video.
- Two operating systems currently dominate mobile computing: Apple iOS and Google Android. iOS was designed to run on Apple iPhone and iPad mobile devices. Android powers smartphones and tablets computers available from many manufactures.

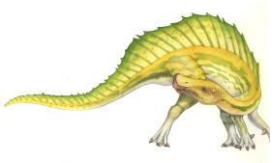


Distributed systems

- *A distributed system is a collection of physically separate, possibly heterogeneous, computer systems that are networked to provide users with access to the various resources that the systems maintain.*
- *Access to a shared resource increases computation speed, functionality, data availability, and reliability.*
- *The different computers communicate closely enough to provide the illusion that only a single operating system controls the network.*
- *Distributed systems distribute computation among several processors.*
- *In contrast to tightly coupled systems (i.e parallel systems), the processor do not share memory or clock. Instead each processor has its own local memory.*
- *These are referred as loosely coupled systems or distributed systems.*

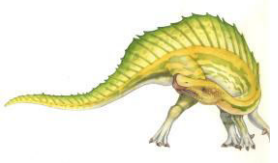
Distributed systems

- These processors are known as sites, nodes, computers and so on.
- ***Advantages:***
- Resource sharing
- Reliability



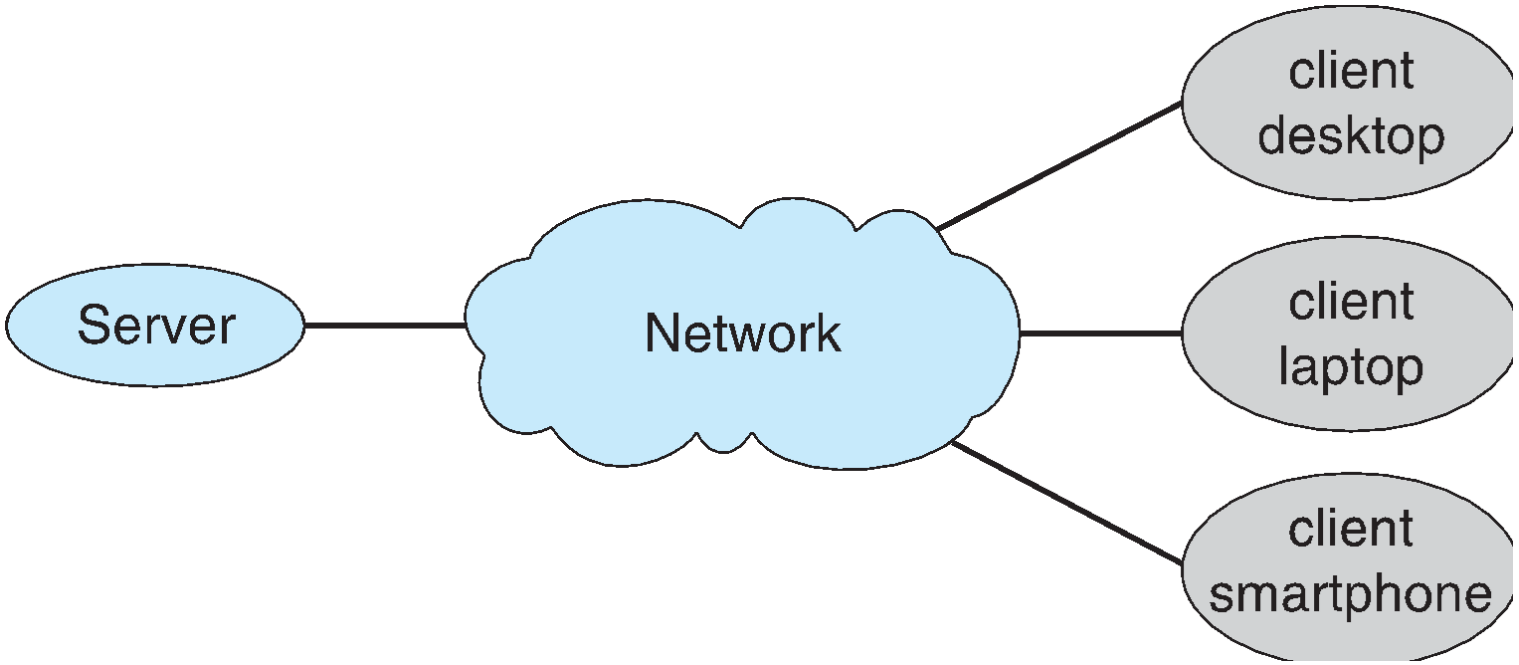
Client Server Computing

- *Server systems satisfy requests generated by client system.*
- *Server systems can be broadly categorized as **compute servers** and **file servers**:*
- ***The computer-server** provides an interface to which a client can send a request to perform an action. In response, the server executes the action and sends the results to the client. Ex: A server running a database that responds to client requests for data.*
- ***The file-server** provides a file-system interface where clients can create, update, read, and delete files. File servers therefore offer users a central storage place for files, which is accessible to all authorized clients.*



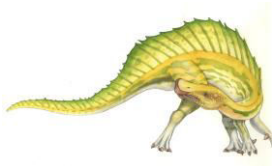
Client Server Computing

A computer or device that is responsible for the central storage and management of data files so that other computers on the same network can access them

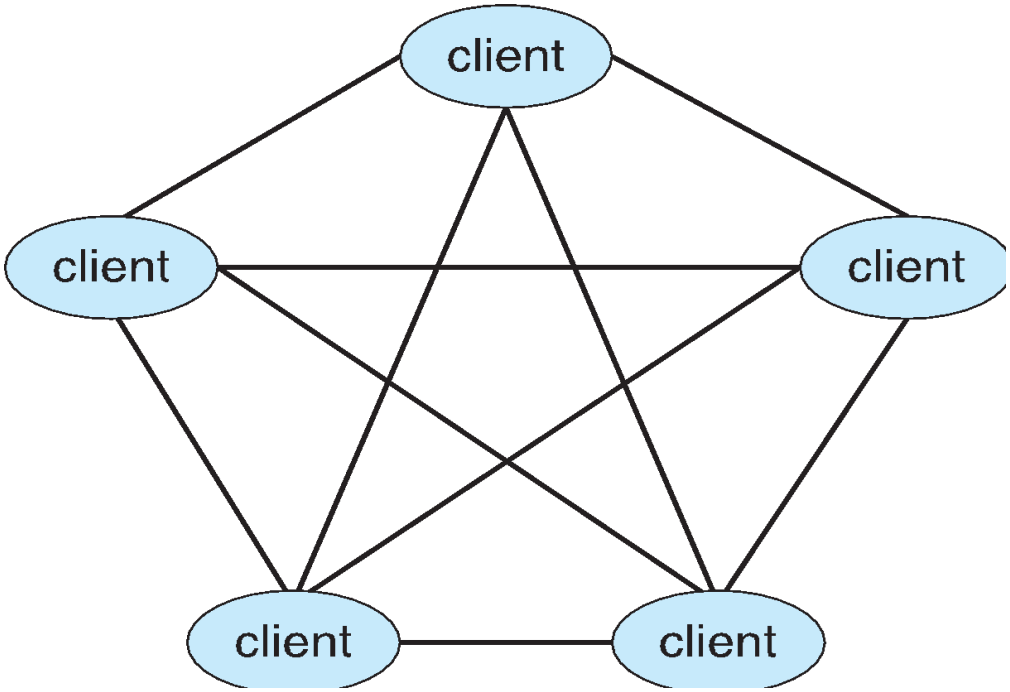


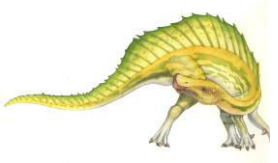
Peer to Peer

- *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.*
- *Services can be provided by **several nodes distributed** throughout the network.*
- *To participate in a peer-to-peer system, a node must first join the network of peers. Once a node has joined the network, it can begin providing services to – and requesting services from – other nodes in the network.*
- *Determining what services are available is accomplished by one of two general ways:*
  - *On joining, it registers its service with a centralized lookup service on the network.*
  - *Broadcast request for service and respond to requests for service via **discovery protocol***
- *Examples include Napster and Gnutella, **Voice over IP (VoIP)** such as Skype*



Peer to Peer



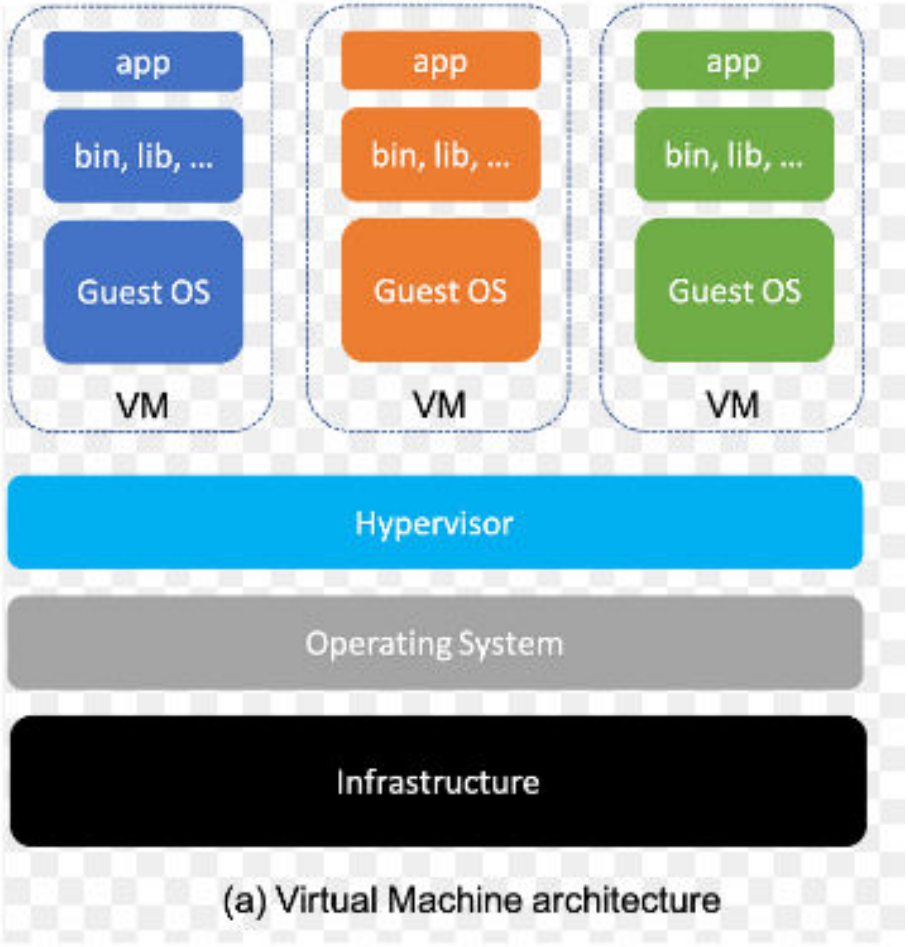


Virtualization

- *Virtualization is a technology that **allows operating systems to run as applications** within other operating systems.*

- *The fundamental idea behind a virtual machine is to abstract the hardware of single computer (the cpu, memory, NIC, disk drive and so forth) into several execution environments thereby creating an illusion that each separate environment is running its own private computer.*

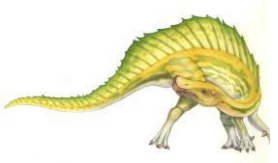
Virtualization



(a) Virtual Machine architecture

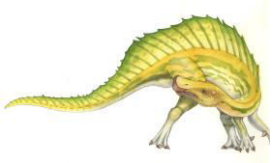
Virtualization

- *Virtualization use Emulation, Emulation is used when source type different from target type.*
- *VMM (virtual machine Monitor) provides virtualization services.*
- *A hypervisor, also known as a virtual machine monitor or VMM, is software that creates and runs virtual machines (VMs). A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing.*



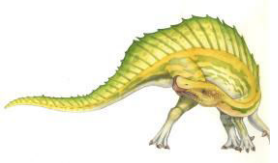
Cloud Computing

- ***Cloud computing*** *is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user.*
- *It is a type of computing that delivers computing storage and even applications as a service across a network.*
- *There are actually many types of cloud computing, including the following:*
- ***Public Cloud*** *– a cloud available via the internet to anyone willing to pay for the services.*
- ***Private Cloud*** *– a cloud run by a company for that company's own use.*
- ***Hybrid Cloud*** *– a cloud that includes both public and private cloud components*



Cloud Computing

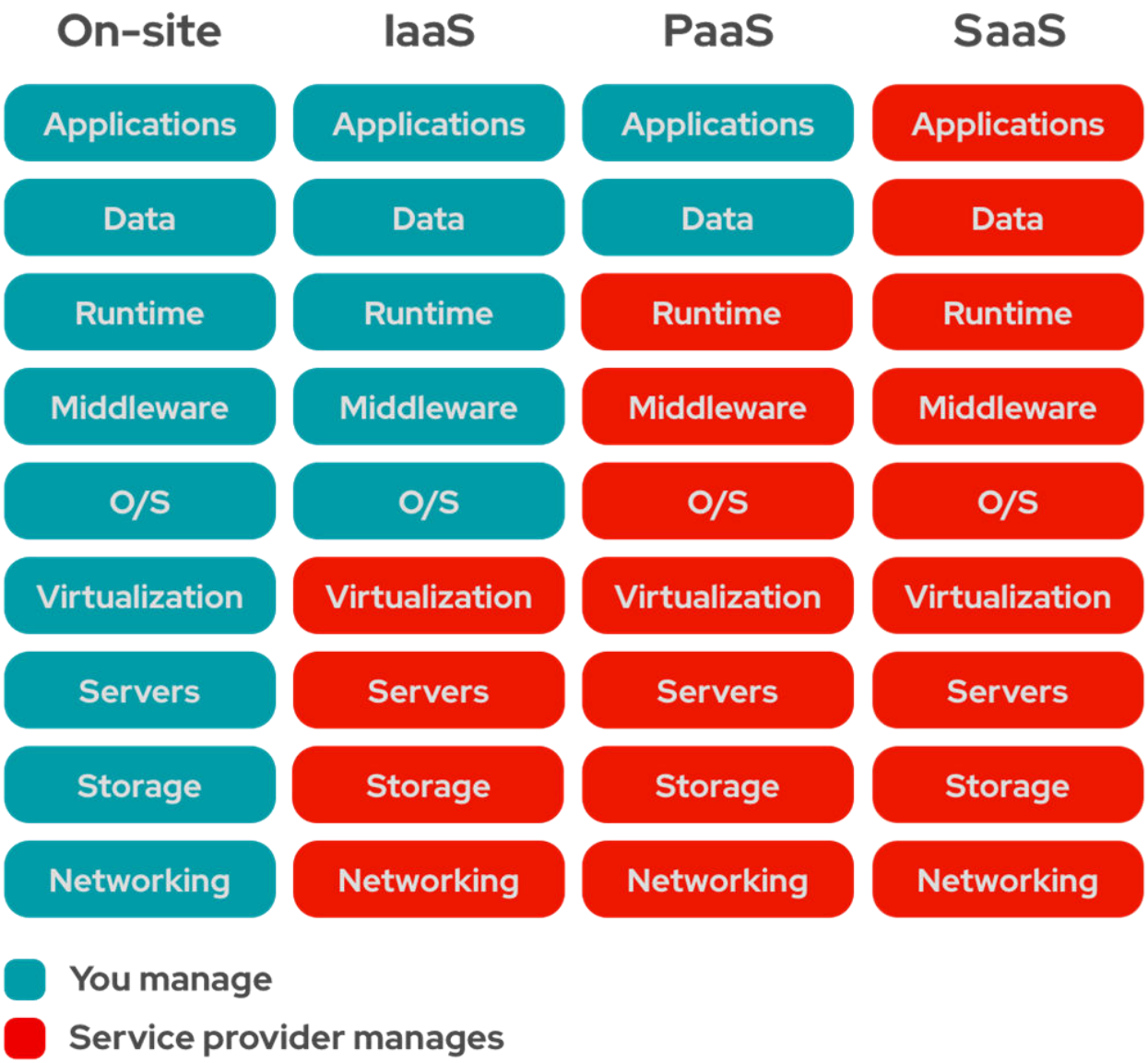
- ***Software as a service (SaaS)*** *– Software as a Service (SaaS) is the most prevalent type of cloud service and provides software like email, word processing, collaboration software, design software and a whole host of other applications. SaaS applications are usually accessible directly through a web browser, removing the need to install applications on individual workstations.*
- ***Platform as a service (PaaS)*** *PaaS, or Platform as a Service, refers to cloud services that provide a framework that companies and developers can use to quickly and easily build (and customise) applications. This model allows developers to focus on the application software without having to manage operating systems, software updates, and other infrastructure matters.to build application and services over the internet and deployment*



Cloud Computing

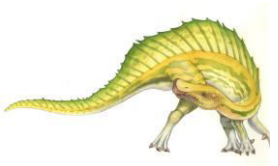
- ***Infrastructure as a service (IaaS)*** – *IaaS, or Infrastructure as a Service, works in a similar manner to traditional computer hardware (i.e. servers, networks, operating systems) but operates in a virtual capacity. Instead of buying the physical hardware, IT managers can purchase the infrastructure as a virtual service through an IaaS provider.*

Cloud Computing



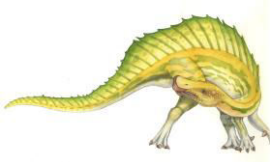
Real-Time Embedded Systems

- *Embedded computers are the most prevalent form of computers in existence.*
- *These devices are found everywhere, from car engines and manufacturing robots DVDs and microwave ovens.*
- *They tend to have very specific tasks.*
- *The operating systems provide limited features.*
- *Usually, they have little or no user interface, preferring to spend their time monitoring and managing hardware devices, such as automobile engines and robotic arms.*



Real-Time Embedded Systems:

- *The use of embedded systems continues to expand.*
- *It almost runs a real-time operating system.*
- *A real-time system is used when* ***rigid time requirements*** *have been placed on the operation of the processor or the flow of data, it is often used as a control device in a dedicated application.*
- *A real-time system functions correctly only if it returns the correct result within its time constraints.*



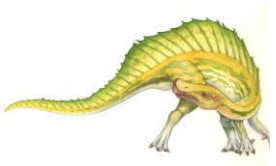
Operating System Structure

Operating system structure can be thought of as the strategy for connecting and incorporating various operating system components within the kernel. Many sorts of structures implement operating systems.

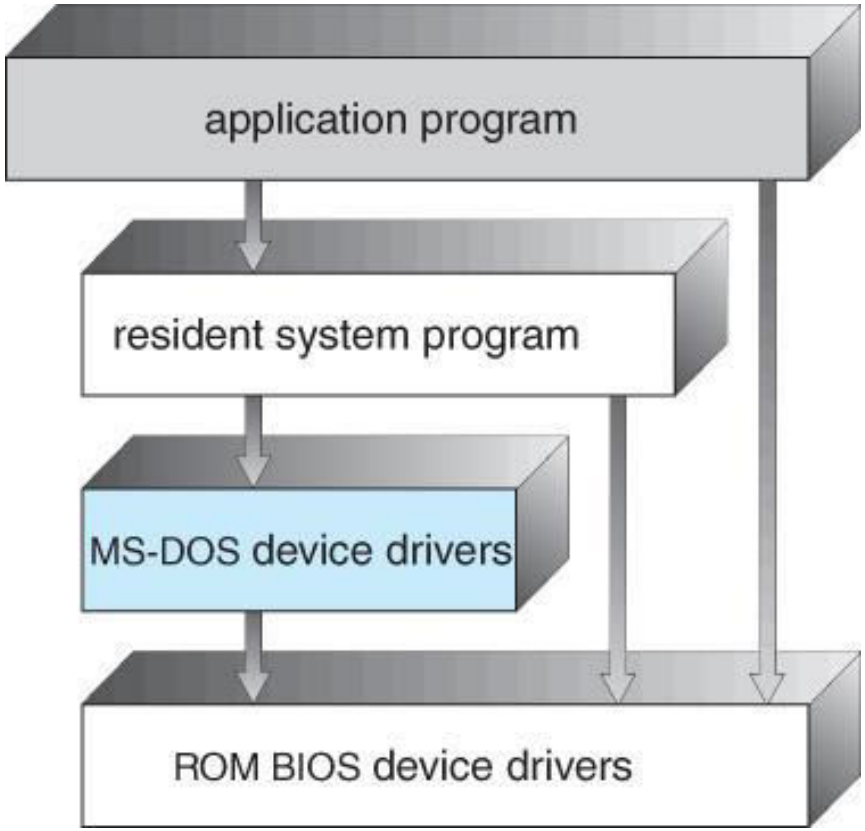
- *Simple Structure*
- *Monolithic Structure*
- *Layered Approach Structure*
- *Micro-Kernel Structure*
- *Exo-Kernel Structure*

Simple Structure

- *Such operating systems do not have well-defined structures and are small, simple, and limited.*
- *The interfaces and levels of functionality are not well separated.*
- *MS-DOS is an example of such an operating system.*
- *In MS-DOS, application programs are able to access the basic I/O routines.*
- *These types of operating systems cause the entire system to crash if one of the user programs fails.*



Simple Structure

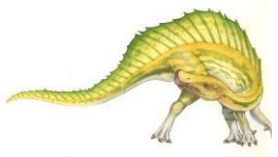


Advantages of Simple Structure:

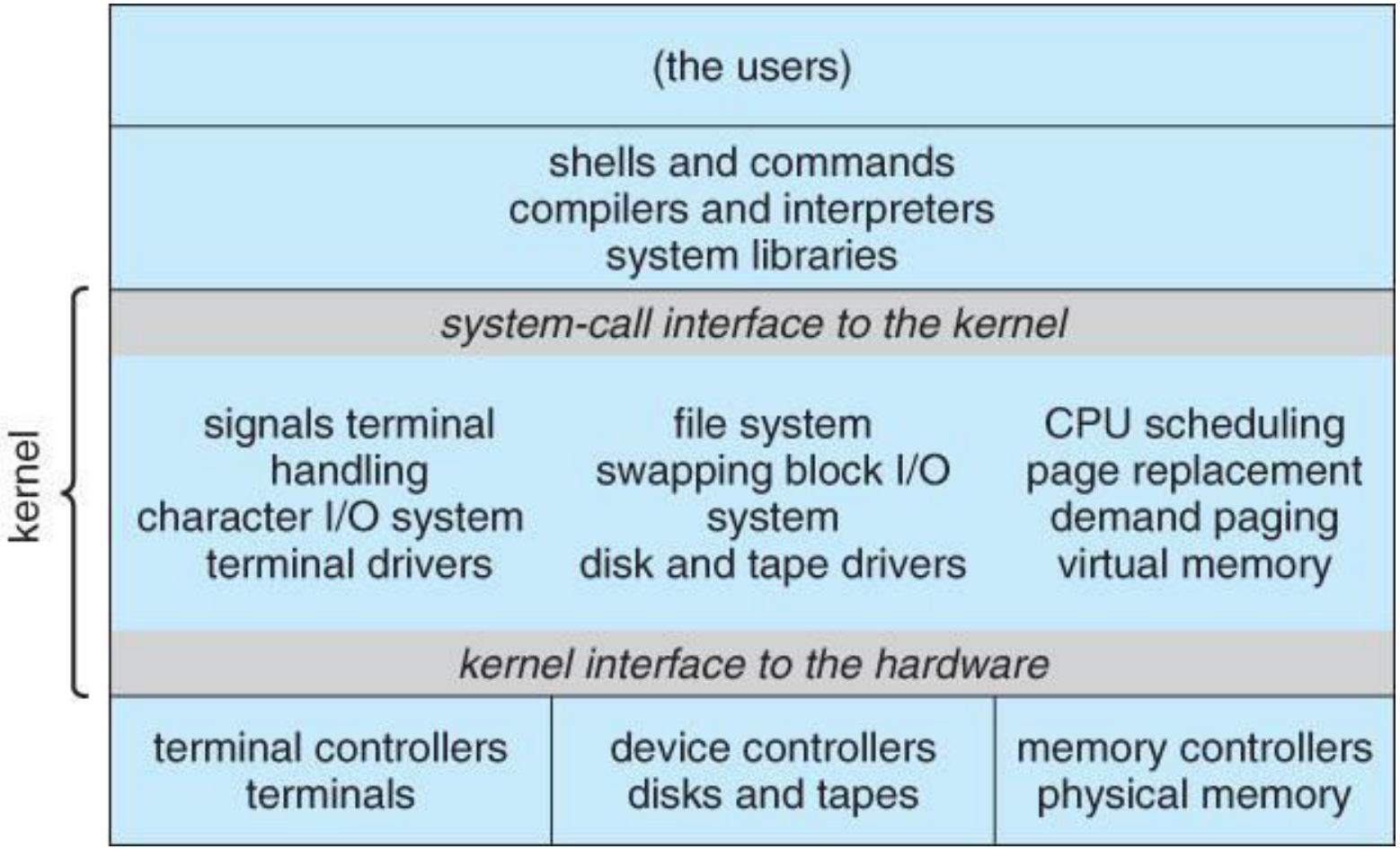
- Because there are only a few interfaces and levels, it is simple to develop.
- Because there are fewer layers between the hardware and the applications, it offers superior performance.

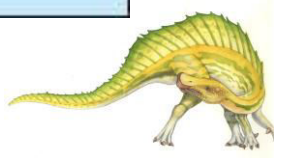
Disadvantages of Simple Structure:

- The entire operating system breaks if just one user program malfunctions.
- Since the layers are interconnected, and in communication with one another, there is no abstraction or data hiding.
- The structure is very complicated, as no clear boundaries exist between modules.



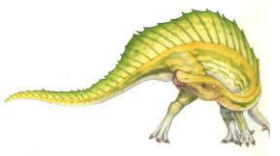
Monolithic


(the users)
shells and commands
compilers and interpreters
system libraries
system-call interface to the kernel
signals terminal handling
character I/O system
terminal drivers
file system
swapping block I/O system
disk and tape drivers
CPU scheduling
page replacement
demand paging
virtual memory
kernel interface to the hardware
terminal controllers
terminals
device controllers
disks and tapes
memory controllers
physical memory
kernel

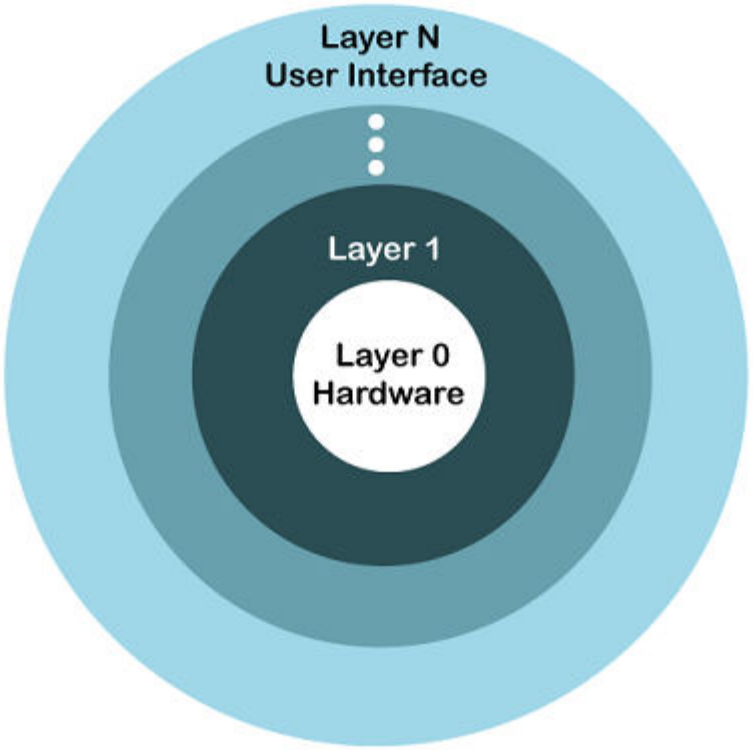


Layered Approach

- *An OS can be broken into pieces and retain much more control over the system.*
- *In this structure, the OS is broken into a number of layers (levels). The bottom layer (layer 0) is the hardware, and the topmost layer (layer N) is the user interface.*
- *These layers are so designed that each layer uses the functions of the lower-level layers. This simplifies the debugging process, if lower-level layers are debugged and an error occurs during debugging, then the error must be on that layer only, as the lower-level layers have already been debugged.*



Layered Approach

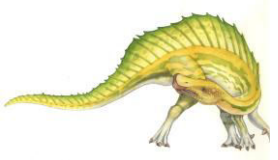


Advantages of Layered structure

- *Layering makes it easier to enhance the operating system, as the implementation of a layer can be changed easily without affecting the other layers.*
- *It is very easy to perform debugging and system verification.*

Disadvantages of Layered structure

- *In this structure, the application's performance is degraded as compared to simple structure.*
- *It requires careful planning for designing the layers, as the higher layers use the functionalities of only the lower layers.*



Micro Kernel