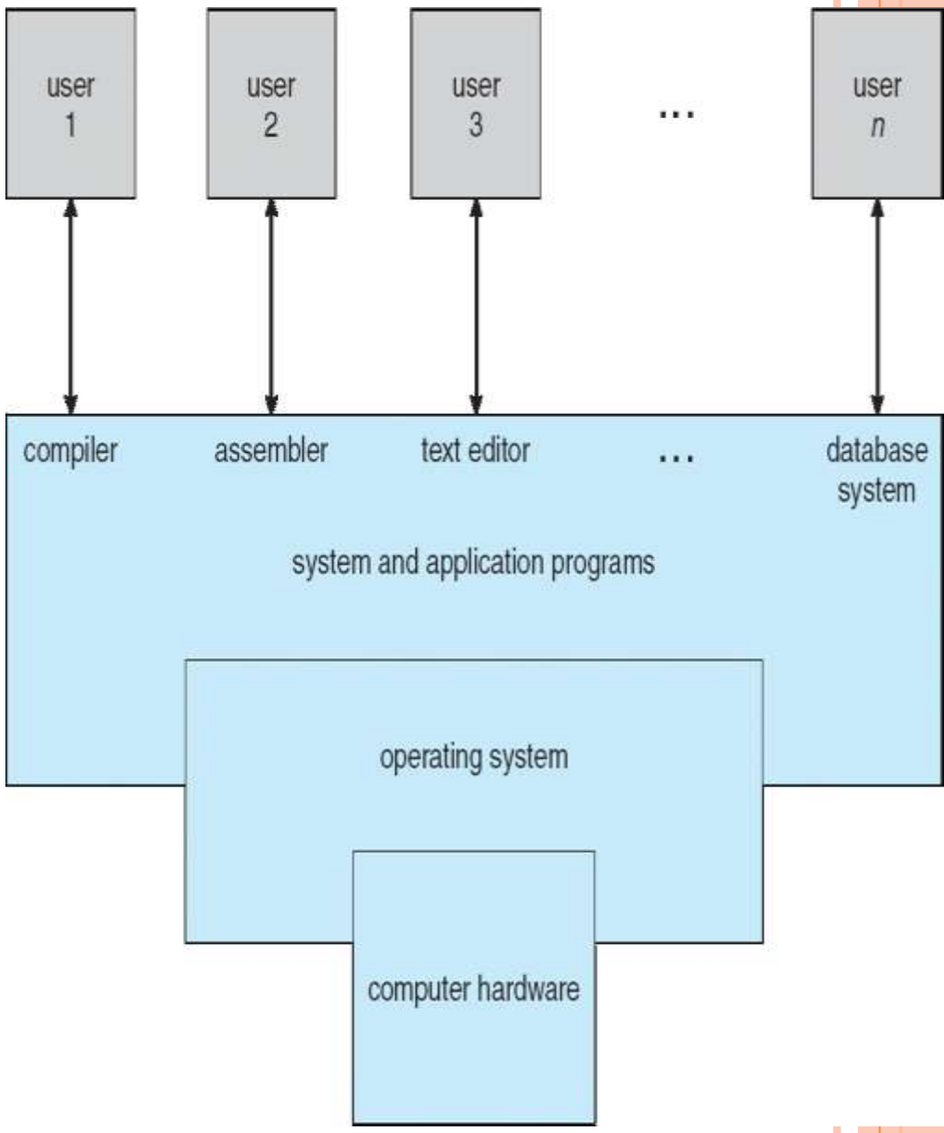
What is an Operating System?

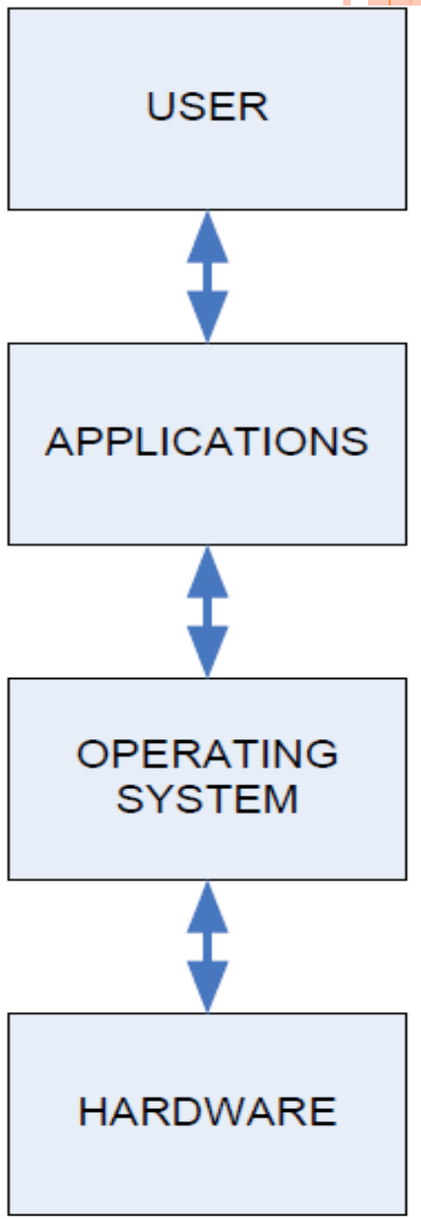
- Computer System = Hardware + Software
- Software = Application Software + System Software(OS)
- An Operating System is a system Software that acts as an intermediary/interface between a user of a computer and the computer hardware.
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

The Structure of Computer Systems

- Accessing computer resources is divided into *layers*.
- Each layer is isolated and only interacts directly with the layer below or above it.
- Hardware – provides basic computing resources
  CPU, memory, I/O devices
- Operating system
  Controls and coordinates use of hardware among various applications and users
- Application programs
  define the ways in which the system resources are used to solve the computing problems of the users.
  Word processors, compilers, web browsers, database systems, video games
- Users
  People, machines, other computers



- If we install a new hardware device
  - ✓ No need to change anything about the user/applications.
  - ✓ However, you do need to make changes to the operating system.
  - ✓ You need to install the device drivers that the operating system will use to control the new device.
- If we install a new software application
  - ✓ No need to make any changes to your hardware.
  - ✓ But we need to make sure the application is supported by the operating system
  - ✓ user will need to learn how to use the new application.
- If we change the operating system
  - ✓ Need to make sure that both applications and hardware will compatible with the new operating system.



Viewpoints of Operating System

The operating system may be observed from the viewpoint of the user or the system. It is known as the user view and the system view. There are mainly two types of views of the operating system. These are as follows:

- **User View**
- **System View**

The user viewpoint focuses on how the user interacts with the operating system through the usage of various application programs

Depends on the point of view

Single user system

Users want convenience, **ease of use**

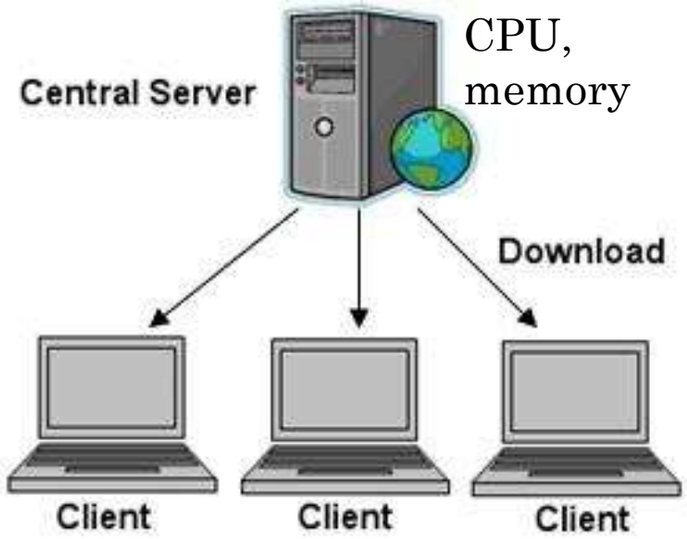
Don't care about **resource utilization**

Optimized for single user experience

Shared computer such as **mainframe** must keep all users happy

Response time minimum

Keep all the users happy



. In contrast, the system viewpoint focuses on how the hardware interacts with the operating system to complete various tasks.

OS is a **resource allocator**

Manages all resources

Decides between conflicting requests for efficient and fair resource use

OS is a **control program**

Controls execution of programs to prevent errors and improper use of the computer

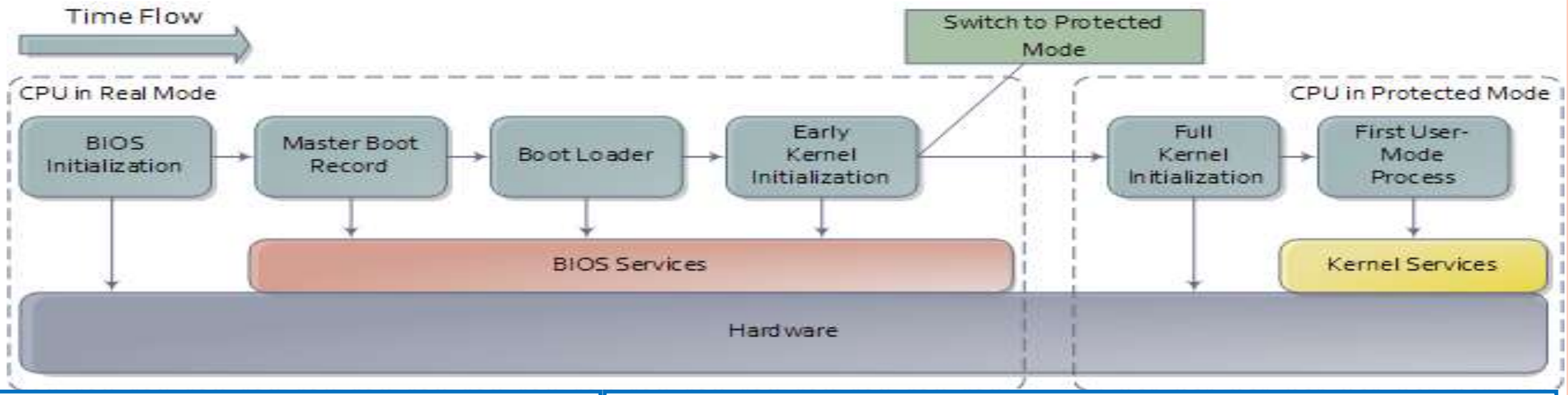
CPU – Central Processing Unit

- This is the brain of your computer.
- It performs all of the calculations.
- In order to do its job, the CPU needs commands to perform, and data to work with.
- The instructions and data travel to and from the CPU on the system bus.
- The operating system provides rules for how that information gets back and forth, and how it will be used by the CPU.

RAM – Random Access Memory

- This is like a desk, or a workspace, where your computer temporarily stores all of the information (data) and instructions (software or program code) that it is currently using.
- Each RAM chip contains millions of address spaces.
- Each address space is the same size, and has its own unique identifying number (address).
- The operating system provides the rules for using these memory spaces, and controls storage and retrieval of information from RAM.
- Device drivers for RAM chips are included with the operating system.

STARTING AN OPERATING SYSTEM(BOOTING)



- ✓ Power On Switch sends electricity to the motherboard on a wire called the *Voltage Good* line.
- ✓ If the power supply is good, then the BIOS (Basic Input/Output System) chip takes over.
- ✓ In Real Mode, CPU is only capable of using approximately 1 MB of memory built into the motherboard.
- ✓ The BIOS will do a Power-On Self Test (POST) to make sure that all hardware are working.

- ✓ BIOS will then look for a small sector at the very beginning of your primary hard disk called MBR.
- ✓ The MBR contains a list, or map, of all of the partitions on your computer‘s hard disk (or disks).
- ✓ After the MBR is found the Bootstrap Loader follows basic instructions for starting up the rest of the computer, including the operating system.
- ✓ In Early Kernel Initialization stage, a smaller core of the Kernel is activated.
- ✓ This core includes the device drivers needed to use computer‘s RAM chips.

BIOS

- BIOS firmware was stored in a ROM/EPROM (Erasable Programmable Read-Only Memory) chip known as **firmware** on the PC motherboard.
- BIOS can be accessed during the initial phases of the boot procedure by pressing del, F2 or F10.
- Finally, the firmware code cycles through all storage devices and looks for a boot-loader. (usually located in first sector of a disk which is 512 bytes)
- If the boot-loader is found, then the firmware hands over control of the computer to it.

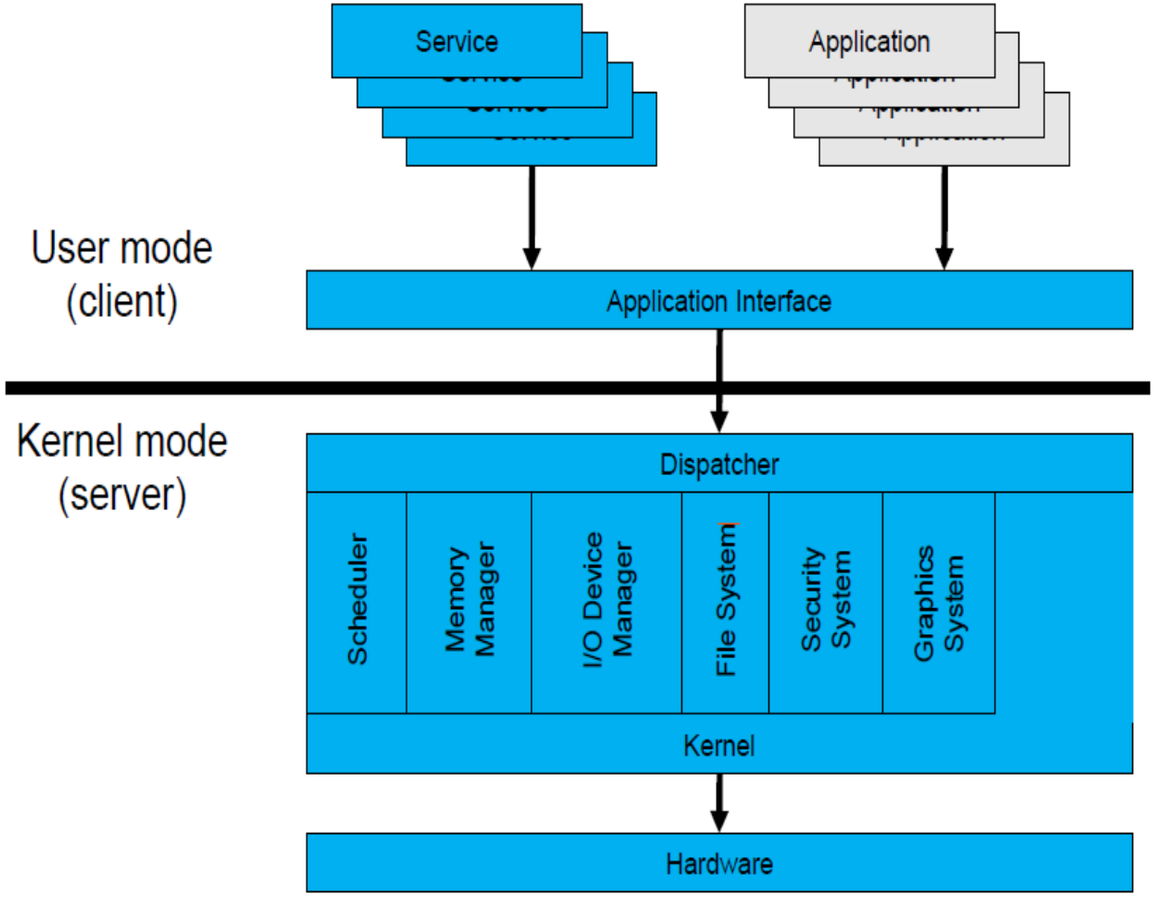
UEFI

- UEFI stands for Unified Extensible Firmware Interface. It does the same job as a BIOS, but with one basic difference: it stores all data about initialization and startup in an .efi file, instead of storing it on the firmware.
- This .efi file is stored on a special partition called EFI System Partition (ESP) on the hard disk. This ESP partition also contains the bootloader.
- UEFI was designed to overcome many limitations of the old BIOS, including:
  - UEFI supports drive sizes upto 9 zettabytes, whereas BIOS only supports 2.2 terabytes.
  - UEFI provides faster boot time.
  - UEFI has discrete driver support, while BIOS has drive support stored in its ROM, so updating BIOS firmware is a bit difficult.
  - UEFI offers security like "Secure Boot", which prevents the computer from booting from unauthorized/unsigned applications. This helps in preventing rootkits.
  - UEFI runs in 32bit or 64bit mode, whereas BIOS runs in 16bit mode. So UEFI is able to provide a GUI (navigation with mouse) as opposed to BIOS which allows navigation only using the keyboard.

Operating System Mode

- The *User Mode* is concerned with the actual interface between the user and the system.
- It controls things like running applications and accessing files.

- The *Kernel Mode* is concerned with everything running in the background.
- It controls things like accessing system resources, controlling hardware functions and processing program instructions.
- System calls are used to change mode from User to Kernel.



KERNEL

- Kernel is a software code that reside in central core of OS. It has complete control over system.
- When operating system boots, kernel is first part of OS to load in main memory.
- Kernel remains in main memory for entire duration of computer session. The kernel code is usually loaded in to protected area of memory.
- Kernel performs it's task like executing processes and handling interrupts in kernel space.
- User performs it's task in user area of memory.
- This memory separation is made in order to prevent user data and kernel data from interfering with each other.
- Kernel does not interact directly with user, but it interacts using SHELL (converts human programs into machine language) and other programs and hardware.

KERNEL CONT…

- Kernel includes:-

  1. Scheduler: It allocates the Kernel's processing time to various processes.

  2. Supervisor: It grants permission to use computer system resources to each process.

  3. Interrupt handler : It handles all requests from the various hardware devices which compete for kernel services.

  4. Memory manager : allocates space in memory for all users of kernel service.

- kernel provides services for process management, file management, I/O management, memory management.
- System calls are used to provide these type of services.

System Call

- **System call** is the programmatic way in which a computer program/user application requests a service from the kernel of the operating system on which it is executed.
- Application program is just a user-process. Due to security reasons , user applications are not given access to privileged resources(the ones controlled by OS).
- When they need to **do any I/O** or have **some more memory** or **spawn a process(**Spawn in computing refers to **a function that loads and executes a new child process**. The current process may wait for the child to terminate or may continue to execute concurrent computing.) or wait for **signal/interrupt,** it requests operating system to facilitate all these. This **request is made through System Call.**
- System calls are also called **software-interrupts.**

What is a System Call?

A system call is a method for a computer program to request a service from the kernel of the operating system on which it is running. A system call is a method of interacting with the operating system via programs. A system call is a request from computer software to an operating system's kernel.

The **Application Program Interface (API)** connects the operating system's functions to user programs. It acts as a link between the operating system and a process, allowing user-level programs to request operating system services. The kernel system can only be accessed using system calls. System calls are required for any programs that use resources.

Why do you need system calls in Operating System?

There are various situations where you must require system calls in the operating system. Following of the situations are as follows:

- It is must require when a file system wants to create or delete a file.
- Network connections require the system calls to sending and receiving data packets.
- If you want to read or write a file, you need to system calls.
- If you want to access hardware devices, including a printer, scanner, you need a system call.
- System calls are used to create and manage new processes.

How System Calls Work

The Applications run in an area of memory known as user space. A system call connects to the operating system's kernel, which executes in kernel space. When an application creates a system call, it must first obtain permission from the kernel. It achieves this using an interrupt request, which pauses the current process and transfers control to the kernel.

If the request is permitted, the kernel performs the requested action, like creating or deleting a file. As input, the application receives the kernel's output. The application resumes the procedure after the input is received. When the operation is finished, the kernel returns the results to the application and then moves data from kernel space to user space in memory.

A simple system call may take few nanoseconds to provide the result, like retrieving the system date and time. A more complicated system call, such as connecting to a network device, may take a few seconds. Most operating systems launch a distinct kernel thread for each system call to avoid bottlenecks. Modern operating systems are multi-threaded, which means they can handle various system calls at the same time.

Types of System Calls

Process Control

Process control is the system call that is used to direct the processes. Some process control examples include creating, load, abort, end, execute, process, terminate the process, etc.

File Management

File management is a system call that is used to handle the files. Some file management examples include creating files, delete files, open, close, read, write, etc.

Device Management

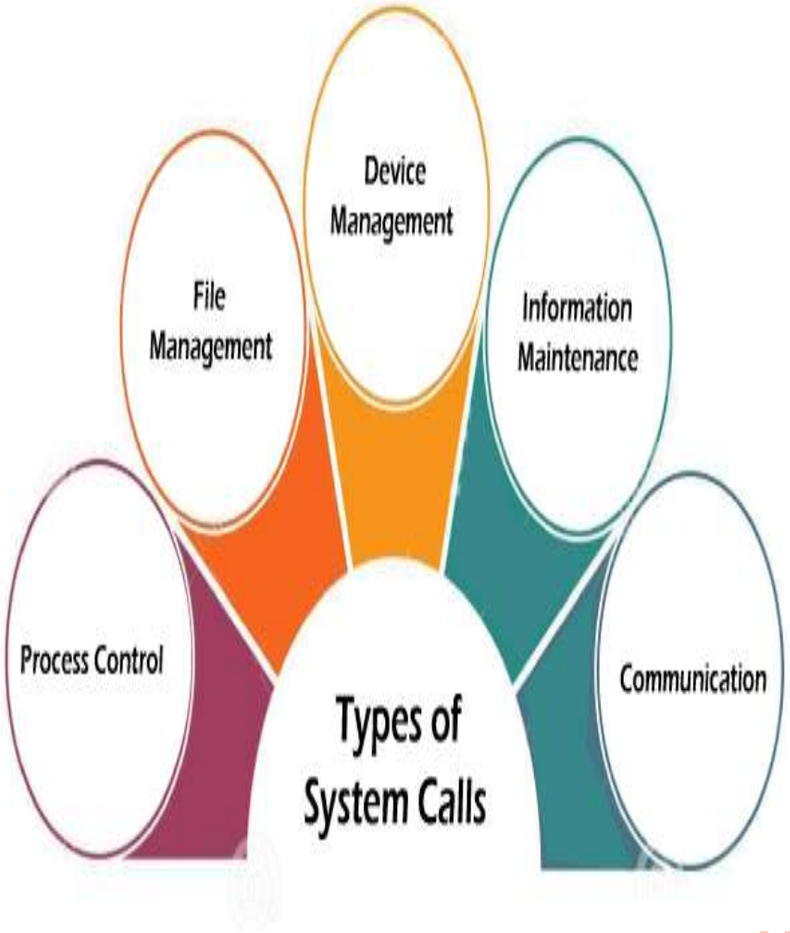
Device management is a system call that is used to deal with devices. Some examples of device management include read, device, write, get device attributes, release device, etc.

Information Maintenance

Information maintenance is a system call that is used to maintain information. There are some examples of information maintenance, including getting system data, set time or date, get time or date, set system data, etc.

Communication

Communication is a system call that is used for communication. There are some examples of communication, including create, delete communication connections, send, receive messages, etc.



Generations of Operating System

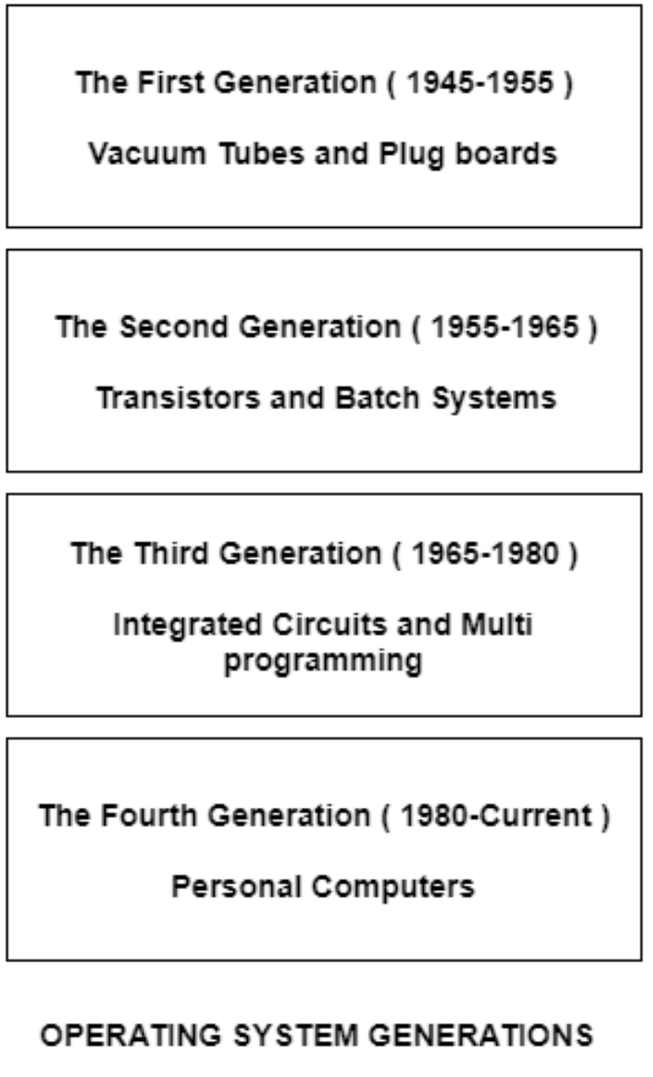
The First Generation (1945 - 1955): Vacuum Tubes and Plug boards

These early computers were designed, built and maintained by a single group of people. Programming languages were unknown and there were no operating systems so all the programming was done in machine language. All the problems were simple numerical calculations.

The Second Generation (1955 - 1965): Transistors and Batch Systems

Transistors led to the development of the computer systems that could be manufactured and sold to paying customers. These machines were known as mainframes and were locked in air-conditioned computer rooms with staff to operate them.

The Batch System was introduced to reduce the wasted time in the computer. A tray full of jobs was collected in the input room and read into the magnetic tape. After that, the tape was rewound and mounted on a tape drive. Then the batch operating system was loaded in which read the first job from the tape and ran it. The output was written on the second tape. After the whole batch was done, the input and output tapes were removed and the output tape was printed.



Generations of Operating System

The Third Generation (1965 - 1980): Integrated Circuits and Multiprogramming

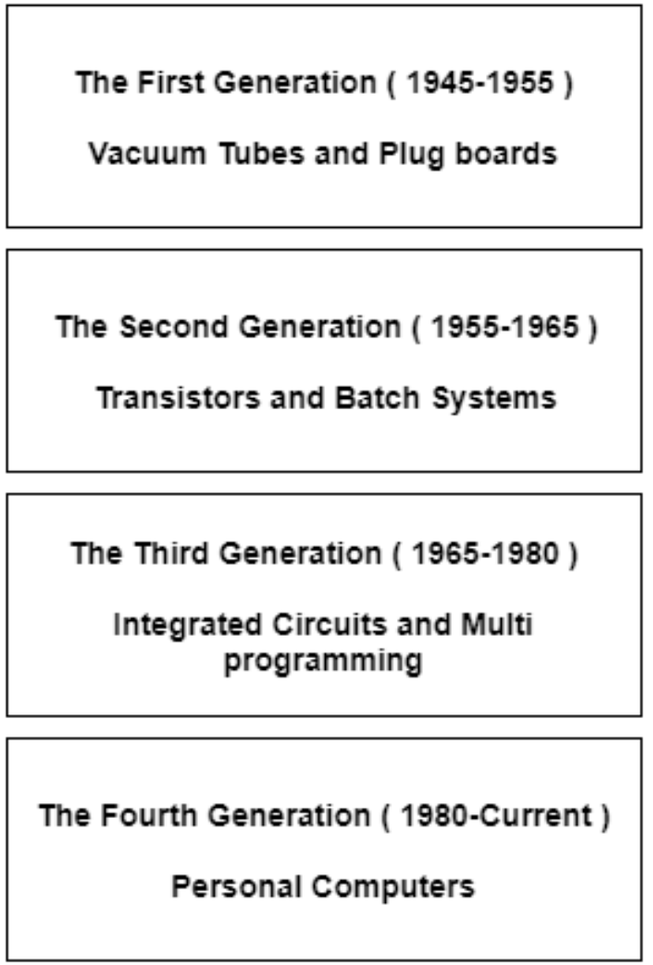
Until the 1960's, there were two types of computer systems i.e. the scientific and the commercial computers. These were combined by IBM in the System/360. This used integrated circuits and provided a major price and performance advantage over the second generation systems.

The third generation operating systems also introduced multiprogramming. This meant that the processor was not idle while a job was completing its I/O operation. Another job was scheduled on the processor so that its time would not be wasted.

The Fourth Generation (1980 - Present): Personal Computers

Personal Computers were easy to create with the development of large-scale integrated circuits. These were chips containing thousands of transistors on a square centimetre of silicon. Because of these, microcomputers were much cheaper than minicomputers and that made it possible for a single individual to own one of them.

The advent of personal computers also led to the growth of networks. This created network operating systems and distributed operating systems. The users were aware of a network while using a network operating system and could log in to remote machines and copy files from one machine to another.

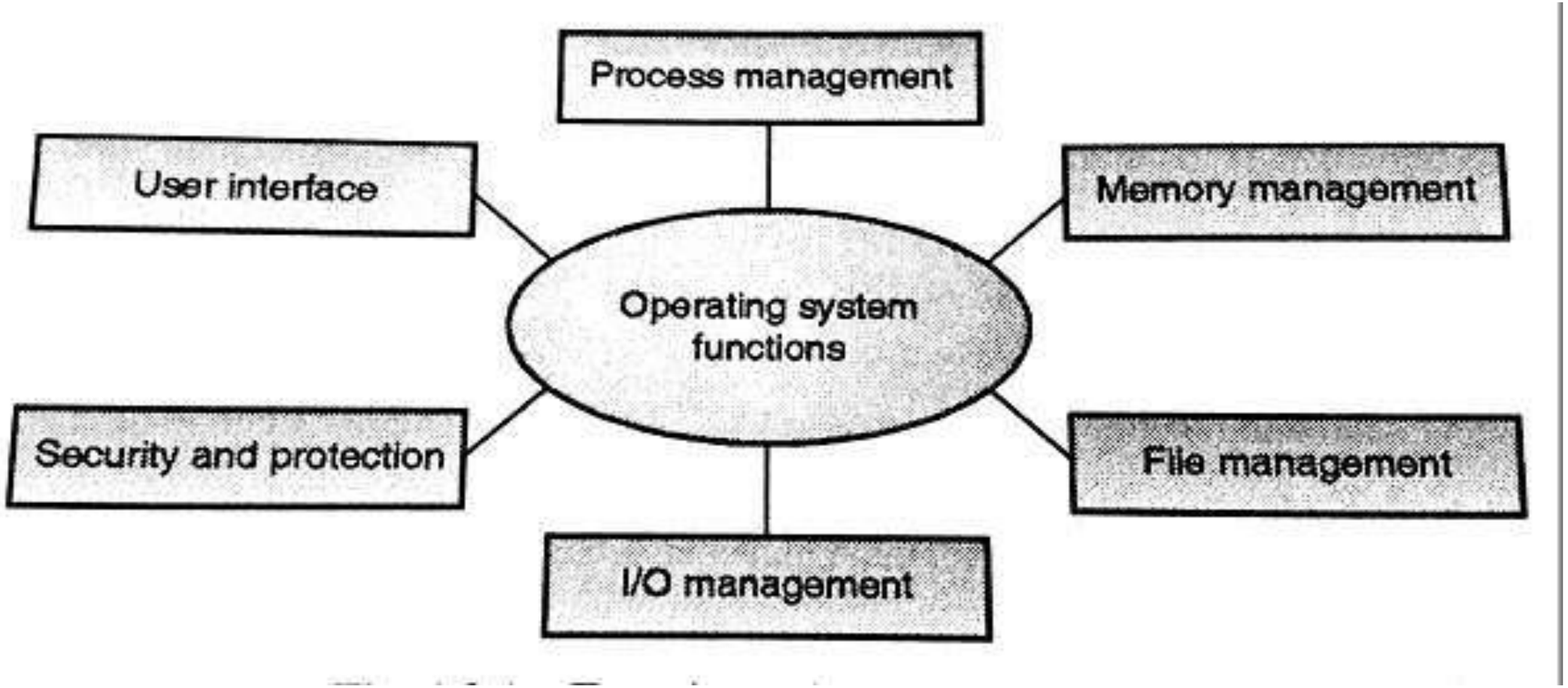


OPERATING SYSTEM GENERATIONS

History of Operating System

- **The First Generation (1940's to early 1950's)**
  - No Operating System
  - All programming was done in absolute machine language, often by wiring up plug-boards to control the machine's basic functions.
- **The Second Generation (1955-1965)**
  - First operating system was introduced in the early 1950's.It was called GMOS
  - Created by General Motors for IBM's machine the 701.
  - Single-stream batch processing systems
- **The Third Generation (1965-1980)**
  - Introduction of multiprogramming
  - Development of Minicomputer
- **The Fourth Generation (1980-Present Day)**
  - Development of PCs
  - Birth of Windows/MaC OS

Functions of Operating System



1. Process Management

- **A *process* is a program in execution.**
- A process needs certain resources, including CPU time, memory, files, and I/O devices to accomplish its task.
- Simultaneous execution leads to multiple processes. Hence creation, execution and termination of a process are the most basic functionality of an OS
- If processes are dependent, than they may try to share same resources. thus task of process synchronization comes to the picture.
- If processes are independent, than a due care needs to be taken to avoid their overlapping in memory area.
- Based on priority, it is important to allow more important processes to execute first than others.

2. Memory management

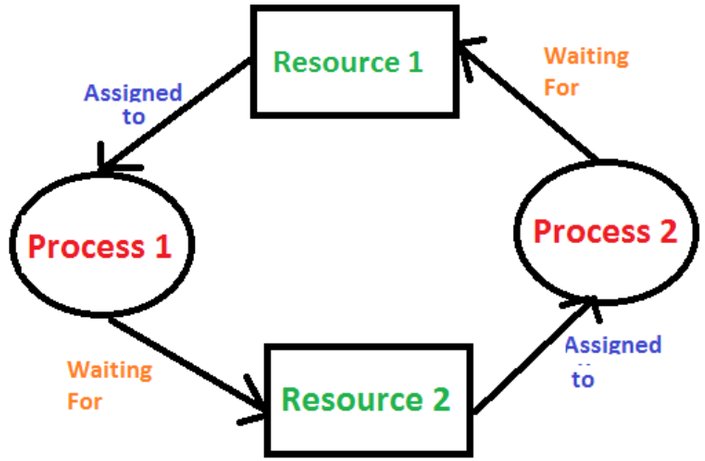
- Memory is a large array of words or bytes, each with its own address.
- It is a repository of quickly accessible data shared by the CPU and I/O devices.
- Main memory is a **volatile** storage device. When the computer made turn off everything stored in RAM will be erased automatically.
- In addition to the physical RAM installed in your computer, most modern operating systems allow your computer to use a *virtual memory system. Virtual memory allows your computer to use part of a permanent storage device (such as a hard disk) as extra memory.*
- The operating system is responsible for the following activities in connections with memory management:
  - Keep track of which parts of memory are currently being used and by whom.
  - Decide which processes to load when memory space becomes available.
  - Allocate and de-allocate memory space as needed.

3. File Management

- A file is a collection of related information defined by its creator.
- *File systems provide the conventions for the encoding, storage and management of data on a storage device such as a hard disk.*
  - FAT12 (floppy disks)
  - FAT16 (DOS and older versions of Windows)
  - FAT32 (older versions of Windows)
  - NTFS (newer versions of Windows)
  - EXT3 (Unix/Linux)
  - HFS+ (Max OS X)
- The operating system is responsible for the following activities in connections with file management:
  - ✦ File creation and deletion.
  - ✦ Directory creation and deletion.
  - ✦ Support of primitives for manipulating files and directories.
  - ✦ Mapping files onto secondary storage.
  - ✦ File backup on stable (nonvolatile) storage media.

4. Device Management or I/O Management

- *Device controllers are components on the motherboard (or on expansion cards) that act as an interface between the CPU and the actual device.*
- *Device drivers, which are the operating system software components that interact with the devices controllers.*
- A special device (inside CPU) called the Interrupt Controller handles the task of receiving interrupt requests and prioritizes them to be forwarded to the processor.
- Deadlocks can occur when two (or more) processes have control of different I/O resources that are needed by the other processes, and they are unwilling to give up control of the device.
- It performs the following activities for device management.
  - Keeps tracks of all devices connected to system.
  - Designates a program responsible for every device known as Input/output controller.
  - Decides which process gets access to a certain device and for how long.
  - Allocates devices in an effective and efficient way.
  - Deallocates devices when they are no longer required.



5. Security & Protection

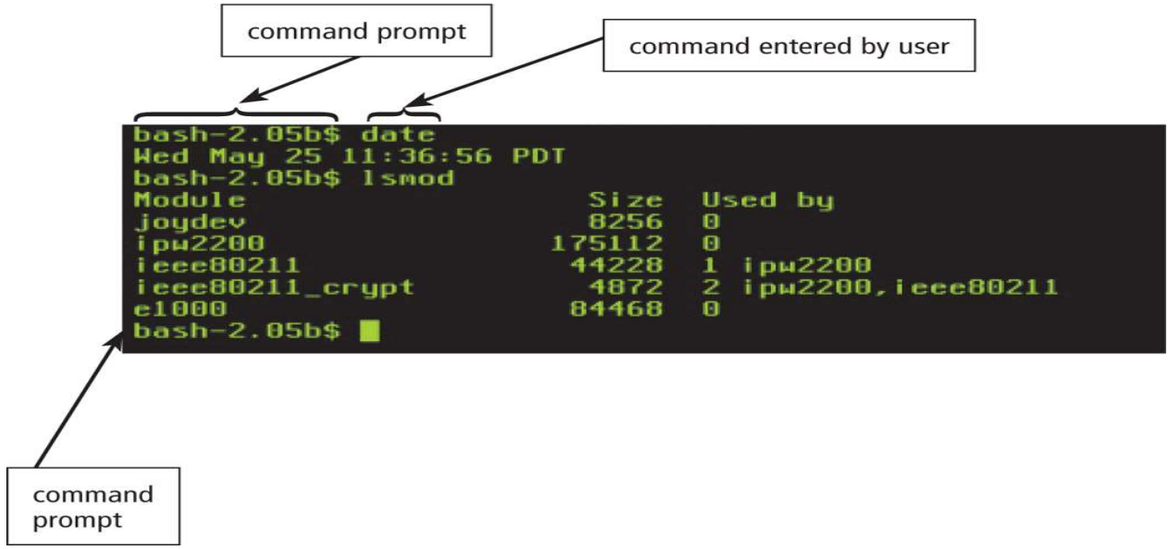
- The operating system uses password protection to protect user data and similar other techniques.
- It also prevents unauthorized access to programs and user data by assigning access right permission to files and directories.
- The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other.

6. User Interface Mechanism

- A **user interface** (**UI**) controls how you enter data and instructions and how information is displayed on the screen
- There are two types of user interfaces
  1. Command Line Interface
  2. Graphical user Interface

1. Command-line interface

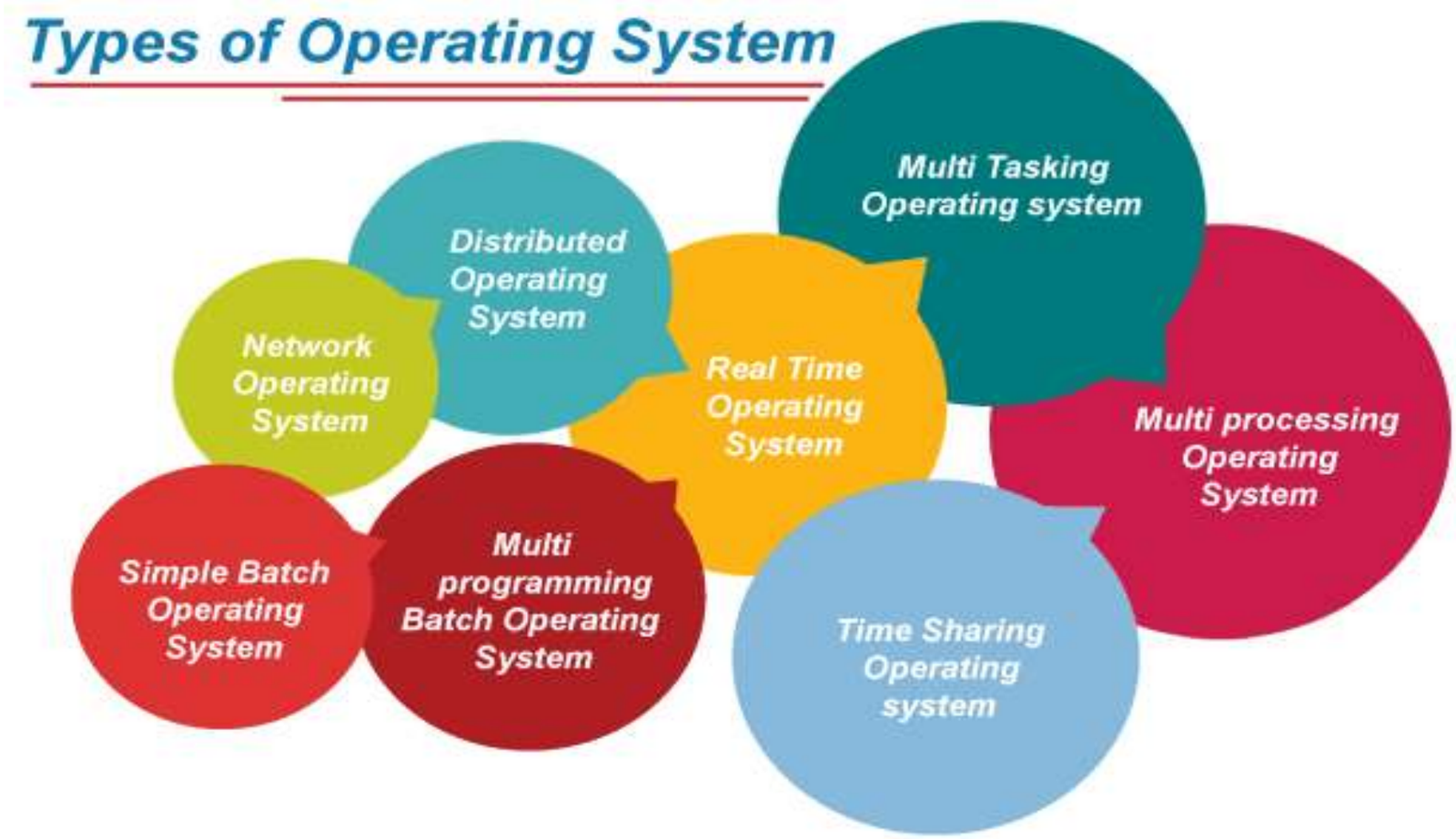
- In a command-line interface, a user types commands represented by short keywords or abbreviations or presses special keys on the keyboard to enter data and instructions



2. Graphical User Interface

- With a graphical user interface (GUI), you interact with menus and visual images

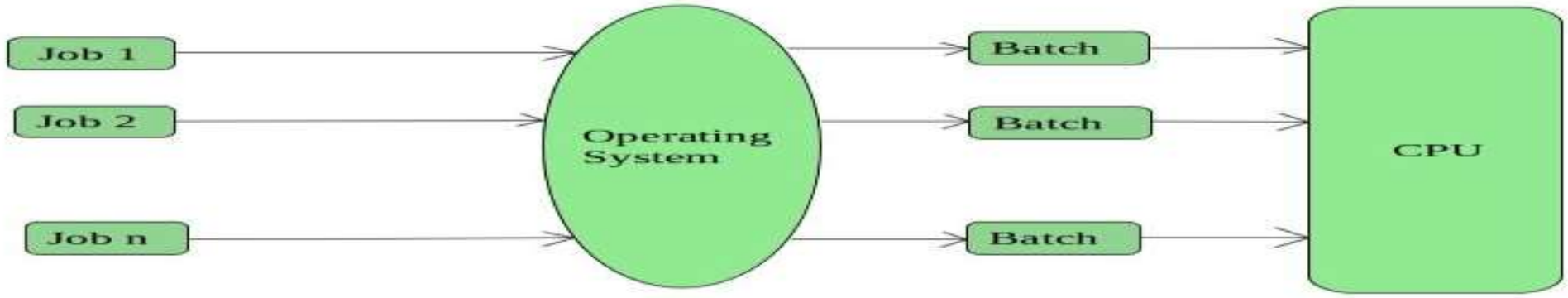
Types of Operating Systems



1. Batch Operating System

In the 1970s, Batch processing was very popular. In this technique, similar types of jobs were batched together and executed in time. People were used to having a single computer which was called a mainframe.
In Batch operating system, access is given to more than one person; they submit their respective jobs to the system for the execution.
The system put all of the jobs in a queue on the basis of first come first serve and then executes the jobs one by one. The users collect their respective output when all the jobs get executed.

- The users of this type of operating system does not interact with the computer directly.
- Each user prepares his job on an off-line device like punch cards and submits it to the computer operator
- There is an operator which takes similar jobs having the same requirement and group them into batches.



The purpose of this operating system was mainly to transfer control from one job to another as soon as the job was completed. It contained a small set of programs called the resident monitor that always resided in one part of the main memory.

Advantages of Batch OS

❖The use of a resident monitor improves computer efficiency as it eliminates CPU time between two jobs.

- ❖ Multiple users can share the batch systems
- ❖ The idle time for the batch system is very less
- ❖ It is easy to manage large work repeatedly in batch systems

Disadvantages of Batch OS

1. Starvation

Batch processing suffers from starvation. There are five jobs J1, J2, J3, J4, and J5, present in the batch. If the execution time of J1 is very high, then the other four jobs will never be executed, or they will have to wait for a very long time. Hence the other processes get starved.

2. Not Interactive

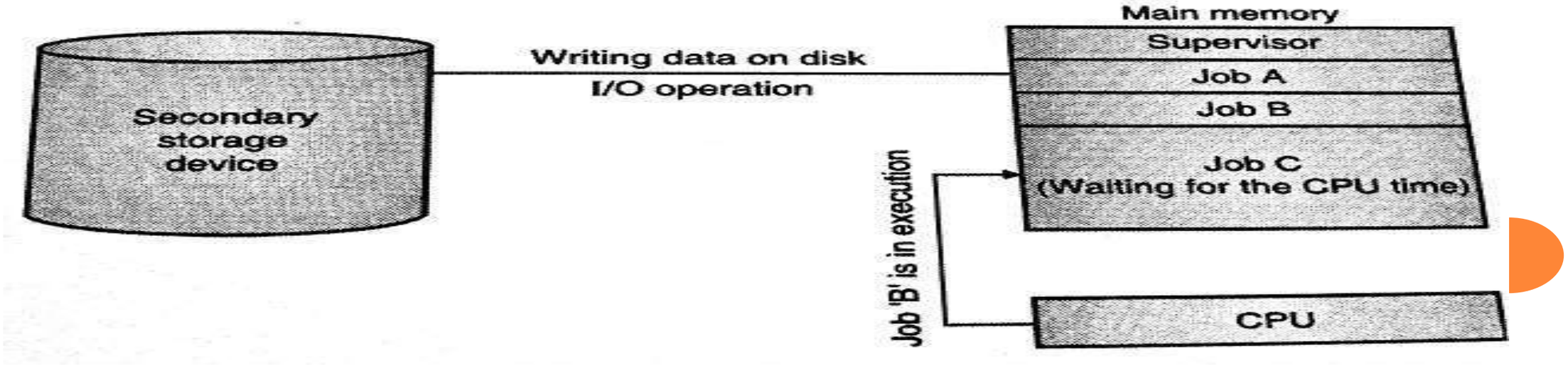
Batch Processing is not suitable for jobs that are dependent on the user's input. If a job requires the input of two numbers from the console, then it will never get it in the batch processing scenario since the user is not present at the time of execution.

3. The computer operators should be well known with batch systems

2. Multiprogramming Operating System:

Multiprogramming is an extension to batch processing where the CPU is always kept busy. Each process needs two types of system time: CPU time and IO time. In a multiprogramming environment, when a process does its I/O, The CPU can start the execution of other processes. Therefore, multiprogramming improves the efficiency of the system.

- This type of OS is used to execute more than one jobs simultaneously by a single processor.
- It increases CPU utilization by organizing jobs so that the CPU always has one job to execute.
- Multiprogramming operating systems use the mechanism of job scheduling and CPU scheduling.



Advantages of Multiprogramming OS

•Throughout the system, it increased as the CPU always had one program to execute.

•Response time can also be reduced.

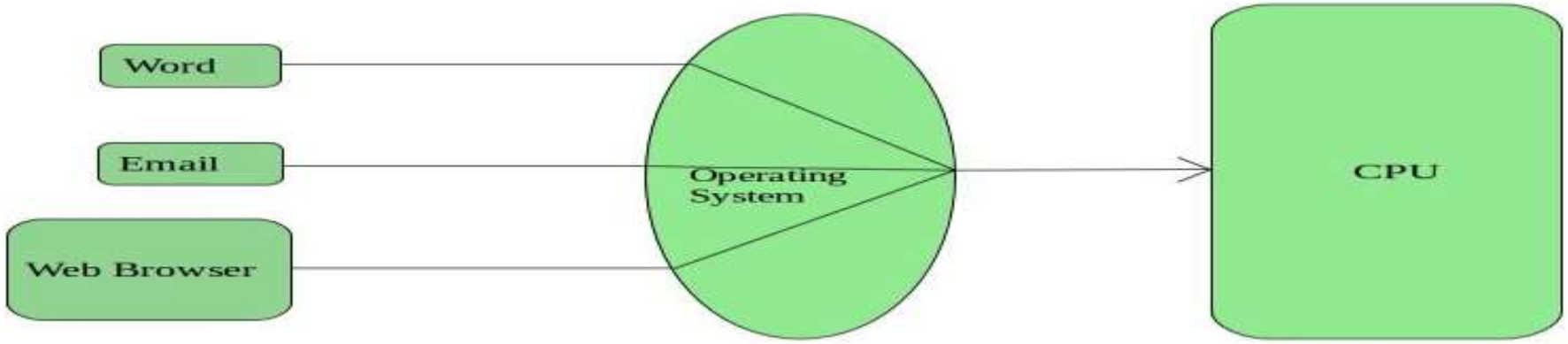
Disadvantages of Multiprogramming OS

Multiprogramming systems provide an environment in which various systems resources are used efficiently, but they do not provide any user interaction with the computer system.

3. Time-Sharing Operating Systems

In the Time Sharing operating system, computer resources are allocated in a time-dependent fashion to several programs simultaneously. Thus it helps to provide a large number of user's direct access to the main computer. It is a logical extension of multiprogramming. In time-sharing, the CPU is switched among multiple programs given by different users on a scheduled basis.

- Each task is given some time to execute so that all the tasks work smoothly.
- These systems are also known as **Multi-tasking Systems.**
- The task can be from a single user or different users also.
- The time that each task gets to execute is called quantum.
- After this time interval is over OS switches over to the next task.



Advantages of Time Sharing Operating System

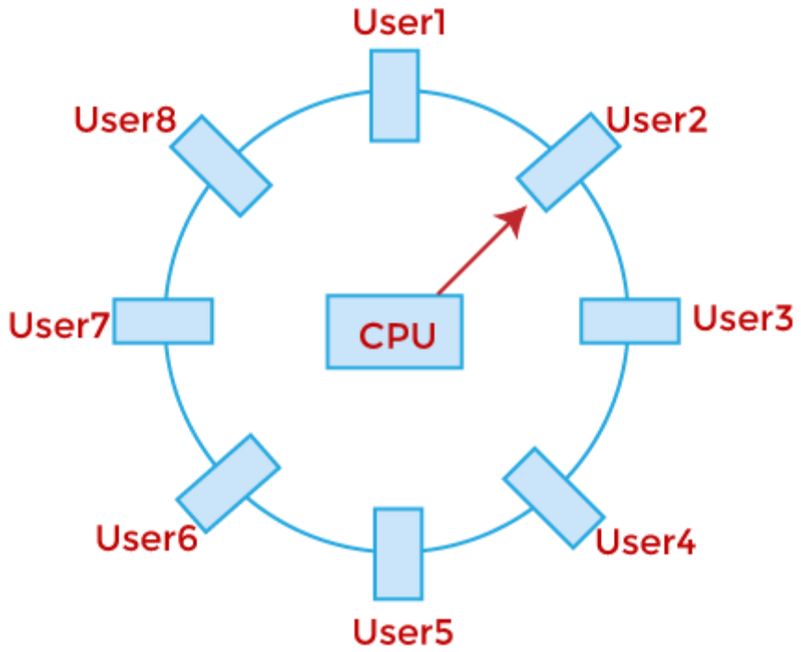
The time-sharing operating system provides effective utilization and sharing of resources.

This system reduces CPU idle and response time.

Disadvantages of Time Sharing Operating System

Data transmission rates are very high in comparison to other methods.

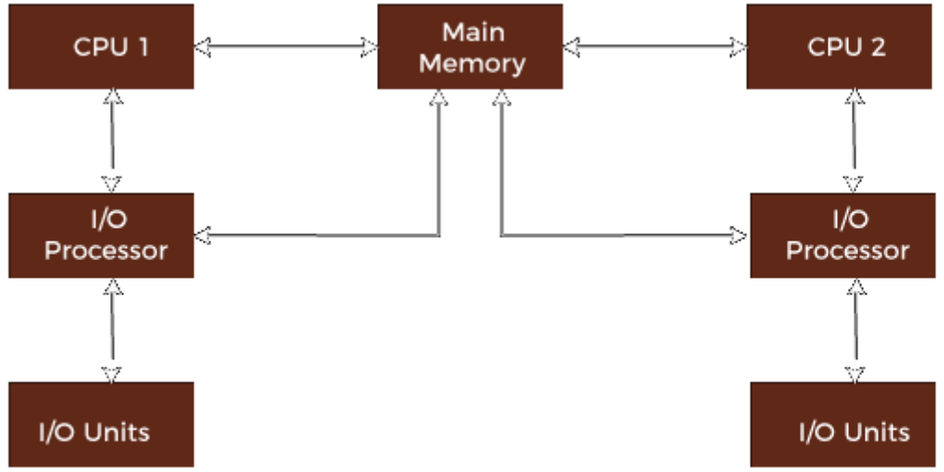
Security and integrity of user programs loaded in memory and data need to be maintained as many users access the system at the same time.



Timesharing in case of 8 users

4. MULTIPROCESSOR OPERATING SYSTEMS

In Multiprocessing, Parallel computing is achieved. There are more than one processors present in the system which can execute more than one process at the same time. This will increase the throughput of the system.



Advantages of Multiprocessing operating system:

•Increased reliability: Due to the multiprocessing system, processing tasks can be distributed among several processors. This increases reliability as if one processor fails, the task can be given to another processor for completion.

•Increased throughout: As several processors increase, more work can be done in less.

Disadvantages of Multiprocessing operating System

•Multiprocessing operating system is more complex and sophisticated as it takes care of multiple CPUs simultaneously.

4. Multiprocessor operating systems

- Multiprocessor operating systems are also known as parallel OS or tightly coupled OS.
- Such operating systems have more than one processor in close communication that sharing the computer bus, the clock and sometimes memory and peripheral devices.
- It executes multiple jobs at the same time and makes the processing faster.
- It supports large physical address space and larger virtual address space.
- If one processor fails then other processor should retrieve the interrupted process state so execution of process can continue.
- Inter-processes communication mechanism is provided and implemented in hardware.

5. Distributed Operating System

The Distributed Operating system is not installed on a single machine, it is divided into parts, and these parts are loaded on different machines. A part of the distributed Operating system is installed on each machine to make their communication possible. Distributed Operating systems are much more complex, large,

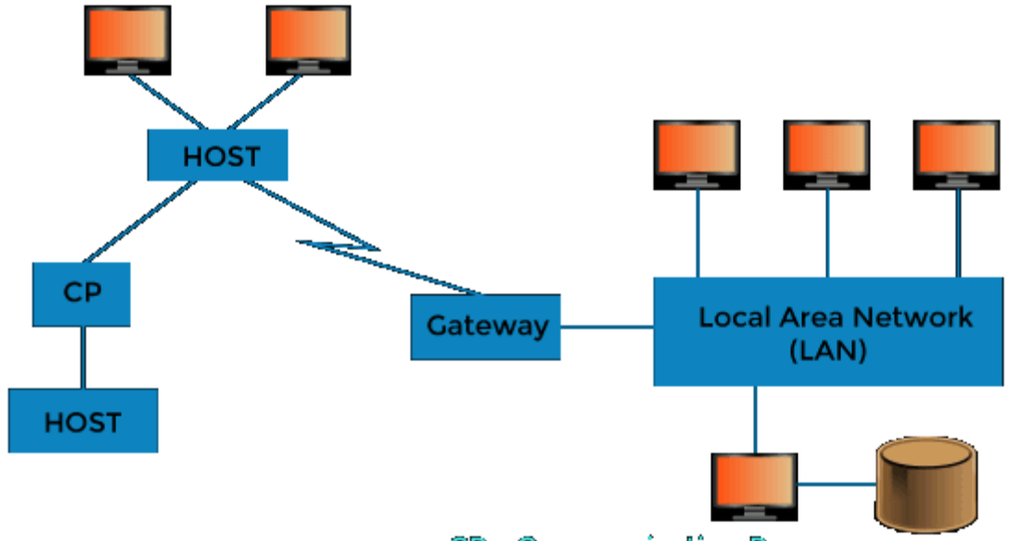
- Various autonomous interconnected computers communicate with each other using a shared communication network.
- Independent systems possess their own memory unit and CPU.
- These are referred to as loosely coupled systems.
- Examples:- Locus, DYSEAC

Advantages of Distributed Operating System

- The distributed operating system provides sharing of resou
- This type of system is fault-tolerant.

Disadvantages of Distributed Operating System

- Protocol overhead can dominate computation cost.



A Typical View of a Distributed System

6. Network Operating System

- These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions.
- These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network.
- The “ other" computers arc called client computers, and each computer that connects to a network server must be running client software designed to request a specific service.
- popularly known as **tightly coupled systems**.

6. Network Operating System

Advantages of Network Operating System:

- Highly stable centralized servers
- Security concerns are handled through servers
- New technologies and hardware up-gradation are easily integrated into the system
- Server access is possible remotely from different locations and types of systems

Disadvantages of Network Operating System:

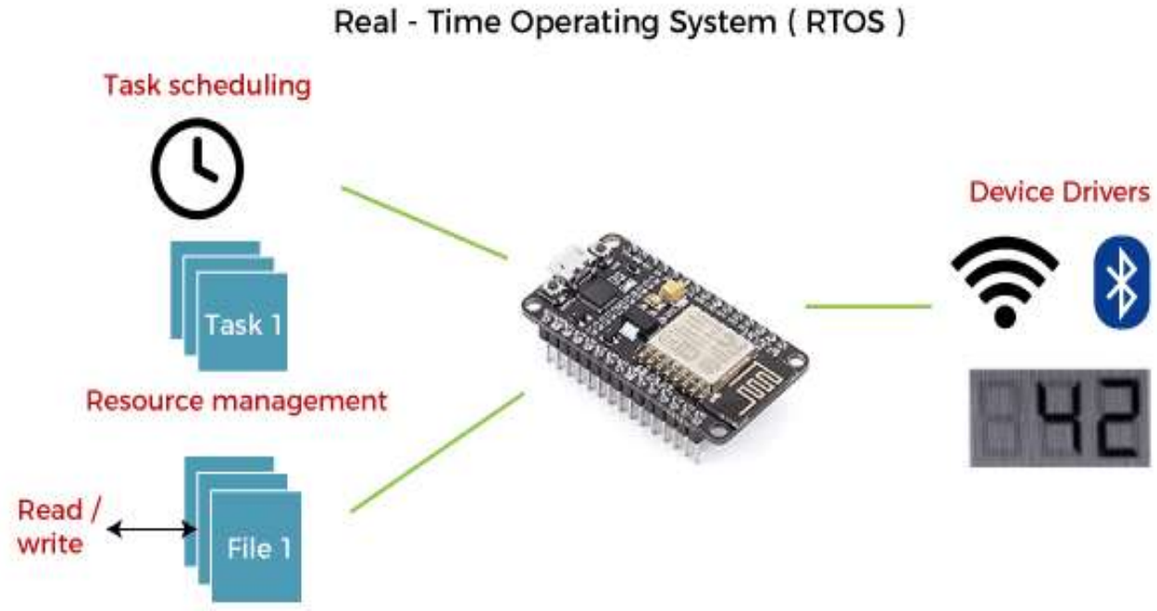
- Servers are costly
- User has to depend on a central location for most operations
- Maintenance and updates are required regularly

Examples of Network Operating System are:

Microsoft Windows Server 2003/2008/2012, UNIX, Linux, Mac OS X, Novell NetWare, and BSD, etc.

7. Real-Time Operating System

- These types of OSs serve real-time systems.
- The time interval required to process and respond to inputs is very small.
- This time interval is called **response time**.
- **Real-time systems** are used when there are time requirements that are very strict like
  - missile systems,
  - air traffic control systems,
  - robots, etc.



8. Embaded Operating System

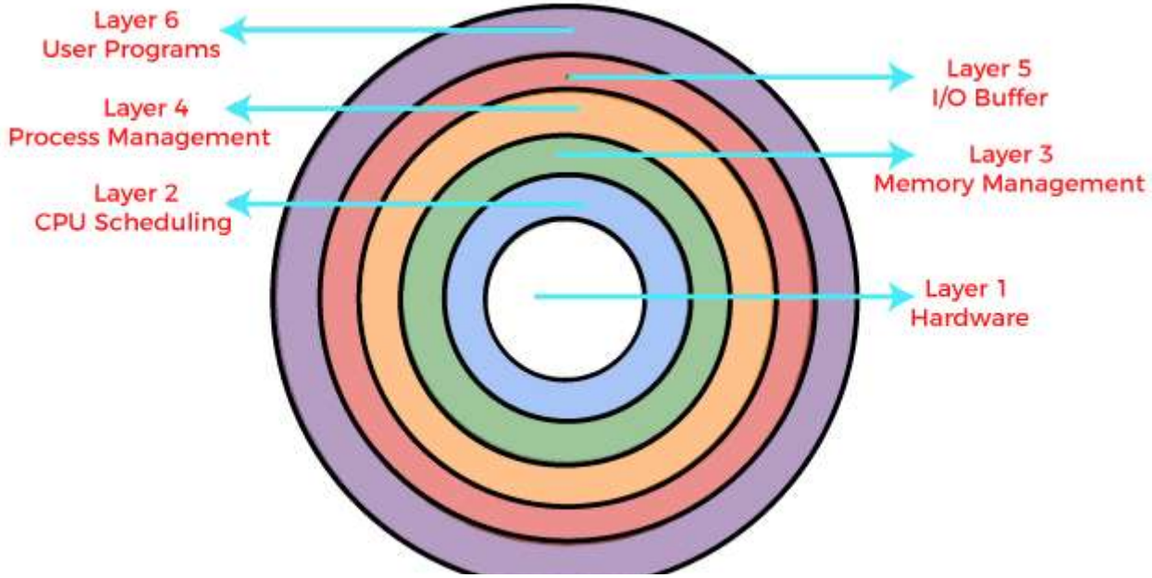
- An embedded operating system is one that is built into the circuitry of an electronic device.
- Embedded operating systems are now found in automobiles, bar-code scanners, cell phones, medical equipment, and personal digital assistants.
- The most popular embedded operating systems for consumer products
  - Windows XP Embedded
  - Windows CE .NET:- it supports wireless communications, multimedia and Web browsing. It also allows for the use of smaller versions of Microsoft Word, Excel, and Outlook.
  - Palm OS:- It is the standard operating system for Palm-brand
  - Symbian:- OS found in " smart" cell phones from Nokia and Sony Ericsson

Architecture/Structure of Operating System

The operating system can be implemented with the help of various structures. The structure of the OS depends mainly on how the various common components of the operating system are interconnected and melded into the kernel.

Layered Architecture

The layered structure approach breaks up the operating system into different layers and retains much more control on the system. 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 only.



Layering provides a distinct advantage in an operating system. All the layers can be defined separately and interact with each other as required. Also, it is easier to create, maintain and update the system if it is done in the form of layers. Change in one layer specification does not affect the rest of the layers.

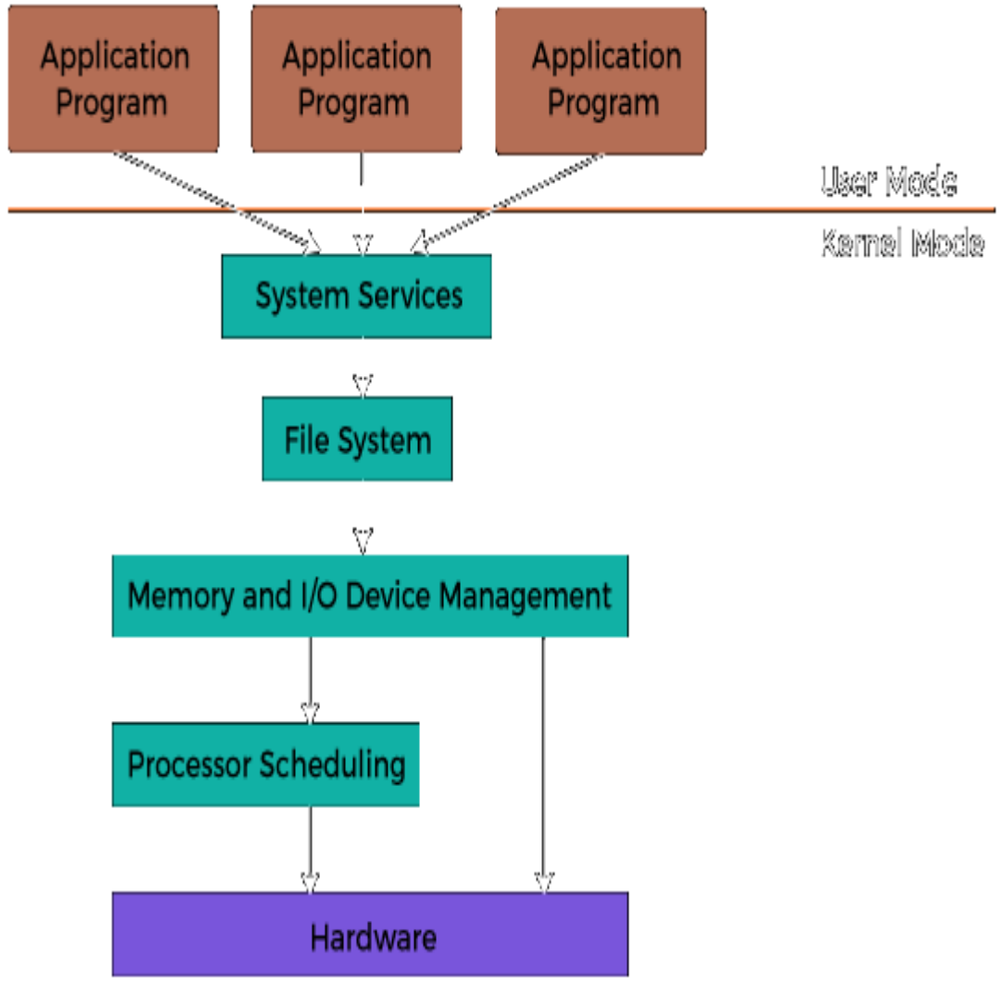
Each of the layers in the operating system can only interact with the above and below layers. The lowest layer handles the hardware, and the uppermost layer deals with the user applications.

Hardware: This layer interacts with the system hardware and coordinates with all the peripheral devices used, such as a printer, mouse, keyboard, scanner, etc. The hardware layer is the lowest and most authoritative layer in the layered operating system architecture. It is attached directly to the core of the system.

CPU Scheduling: This layer deals with scheduling the processes for the CPU. This layer is responsible for managing how many processes will be allocated to the CPU and how many will stay out of the CPU.

Memory Management: Memory management deals with memory and moving processes from disk to primary memory for execution and back again. This is handled by the third layer of the operating system. All memory management is associated with this layer. There are various types of memories in the computer like RAM, ROM.

If you consider RAM, then it is concerned with swapping in and swapping out of memory. When our computer runs, some processes move to the main memory (RAM) for execution, and when programs, such as calculator, exit, it is removed from the main memory.



Process Management: This layer is responsible for managing the processes, i.e., assigning the processor to a process and deciding how many processes will stay in the waiting schedule. The priority of the processes is also managed in this layer. The different algorithms used for process scheduling are FCFS (first come, first served), SJF (shortest job first), priority scheduling etc.

I/O Buffer: I/O devices are very important in computer systems. This layer handles the buffers for I/O devices and makes sure that they work correctly.

Suppose you are typing from the keyboard. There is a keyboard buffer attached with the keyboard, which stores data for a temporary time. Similarly, all input/output devices have some buffer attached to them. This is because the input/output devices have slow processing or storing speed. The computer uses buffers to maintain the good timing speed of the processor and input/output devices.

User Programs: This is the highest layer in the layered operating system. This layer deals with the many user programs and applications that run in an operating system, such as word processors, games, browsers, etc. also called as an application layer because it is concerned with application programs.

Advantages of Layered Structure

There are several advantages of the layered structure of operating system design, such as:

Modularity: This design promotes modularity as each layer performs only the tasks it is scheduled to perform.

Easy debugging: As the layers are discrete so it is very easy to debug. Suppose an error occurs in the CPU scheduling layer. The developer can only search that particular layer to debug, unlike the Monolithic system where all the services are present.

Easy update: A modification made in a particular layer will not affect the other layers.

No direct access to hardware: The hardware layer is the innermost layer present in the design. So a user can use the services of hardware but cannot directly modify or access it, unlike the Simple system in which the user had direct access to the hardware.

Abstraction: Every layer is concerned with its functions. So the functions and implementations of the other layers are abstract to it.

Disadvantages of Layered Structure

Though this system has several advantages over the Monolithic and Simple design, there are also some disadvantages, such as:

Complex and careful implementation: As a layer can access the services of the layers below it, so the arrangement of the layers must be done carefully. For example, the backing storage layer uses the services of the memory management layer. So it must be kept below the memory management layer. Thus with great modularity comes complex implementation.

Slower in execution: If a layer wants to interact with another layer, it requests to travel through all the layers present between the two interacting layers. Thus it increases response time, unlike the Monolithic system, which is faster than this. Thus an increase in the number of layers may lead to a very inefficient design.

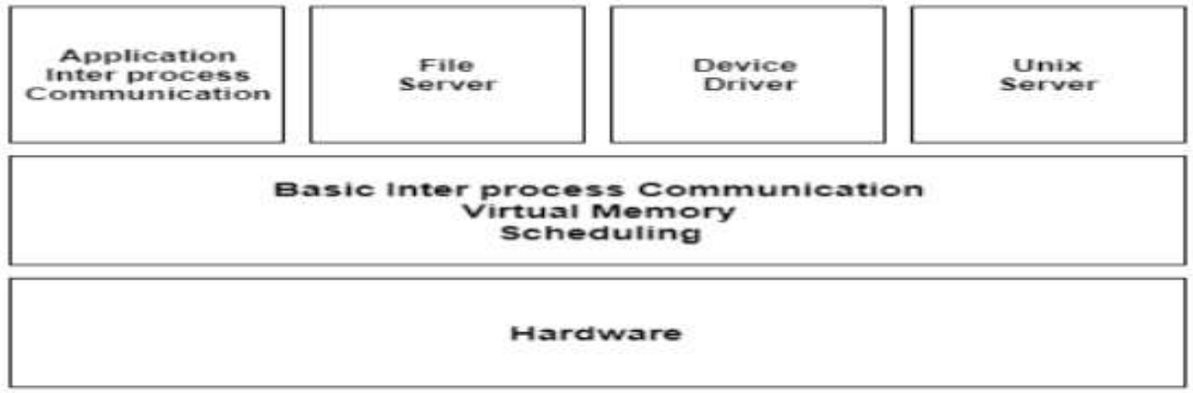
Functionality: It is not always possible to divide the functionalities. Many times, they are interrelated and can't be separated.

Communication: No communication between non-adjacent layers.

What is Microkernel?

A microkernel is one of the classifications of the kernel. Being a kernel it manages all system resources. But in a microkernel, the **user services** and **kernel services** are implemented in different address spaces. The user services are kept in **user address space**, and kernel services are kept under **kernel address space**, thus also reduces the size of kernel and size of an operating system as well.

It provides minimal services of process and memory management. The communication between client program/application and services running in user address space is established through message passing, reducing the speed of execution microkernel. The Operating System **remains unaffected** as user services and kernel services are isolated so if any user service fails it does not affect kernel service. Thus it adds to one of the advantages of a microkernel. It is easily **extendible** i.e. if any new services are to be added they are added to user address space and hence require no modification in kernel space. It is also portable, secure, and reliable.



Microkernel Based Operating System

Microkernel Architecture –
Since the kernel is the core part of the operating system, so it is meant for handling the most important services only. Thus in this architecture, only the most important services are inside the kernel and the rest of the OS services are present inside the system application program. Thus users are able to interact with those not-so-important services within the system application. And the microkernel is solely responsible for the most important services of the operating system they are named as follows:

Inter process-Communication
Memory Management
CPU-Scheduling

The minimum functionalities included in the microkernel are –
Memory management mechanisms like address spaces are included in the microkernel. This also contains memory protection features.
Processor scheduling mechanisms are also necessary in the microkernel. This contains process and thread schedulers.
Inter process communication is important as it is needed to manage the servers that run their own address spaces.

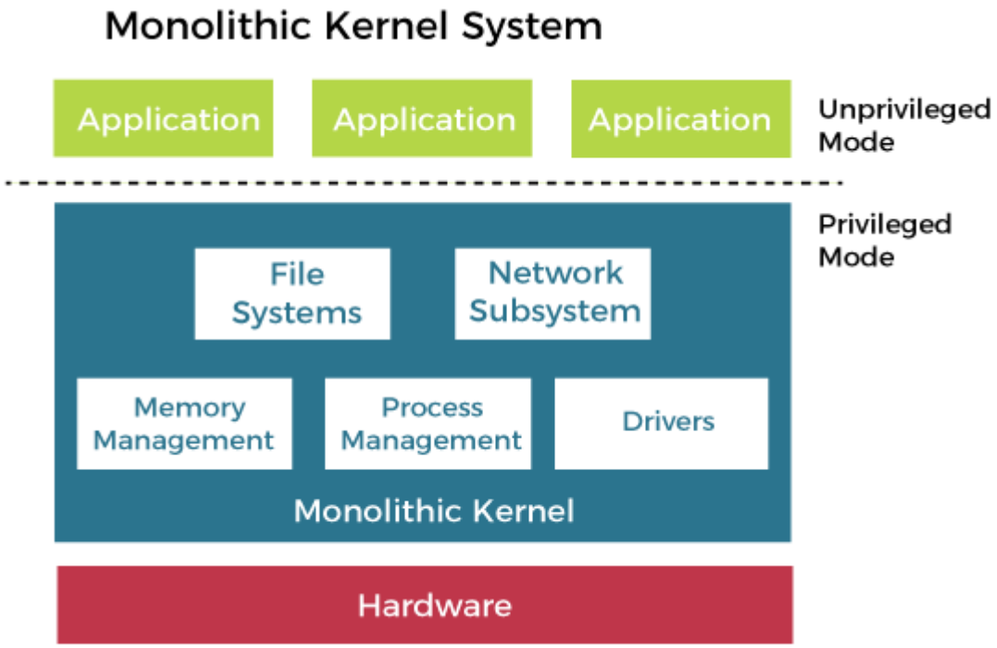
Advantages of Microkernel –

The architecture of this kernel is small and isolated hence it can function better.
Expansion of the system is easier, it is simply added to the system application without disturbing the kernel.
Microkernels are modular and the different modules can be replaced, reloaded, modified, changed etc. as required. This can be done without even touching the kernel.
Microkernels are quite secure as only those components are included that would disrupt the functionality of the system otherwise.
Microkernels contain fewer system crashes as compared to monolithic systems. Also, the crashes that do occur can be handled quite easily due to the modular structure of microkernels.

Monolithic Architecture is like a big container, wherein all the software components of an app are assembled and tightly coupled, i.e., each component fully depends on each other.

The monolithic operating system is a very basic operating system in which file management, memory management, device management, and process management are directly controlled within the kernel. The kernel can access all the resources present in the system,

The monolithic operating system is also known as the monolithic kernel. This is an old operating system used to perform small tasks like batch processing and time-sharing tasks in banks. The monolithic kernel acts as a virtual machine that controls all hardware parts.



Advantages of Monolithic Kernel

Here are the following advantages of a monolithic kernel, such as:

The execution of the monolithic kernel is quite fast as the services such as memory management, file management, process scheduling, etc., are implemented under the same address space.

A process runs completely in single address space in the monolithic kernel.

Disadvantages of Monolithic Architecture:

Here are some disadvantages of monolithic architecture:

The addition of new features or removal of obsolete features is very difficult.

Security issues are always there because there is no isolation among various servers present in the kernel.

virtual machine (VM)

A virtual machine (VM) is an operating system (OS) or application environment that is installed on software, which imitates dedicated hardware. The end user has the same experience on a VM as they would on dedicated hardware.

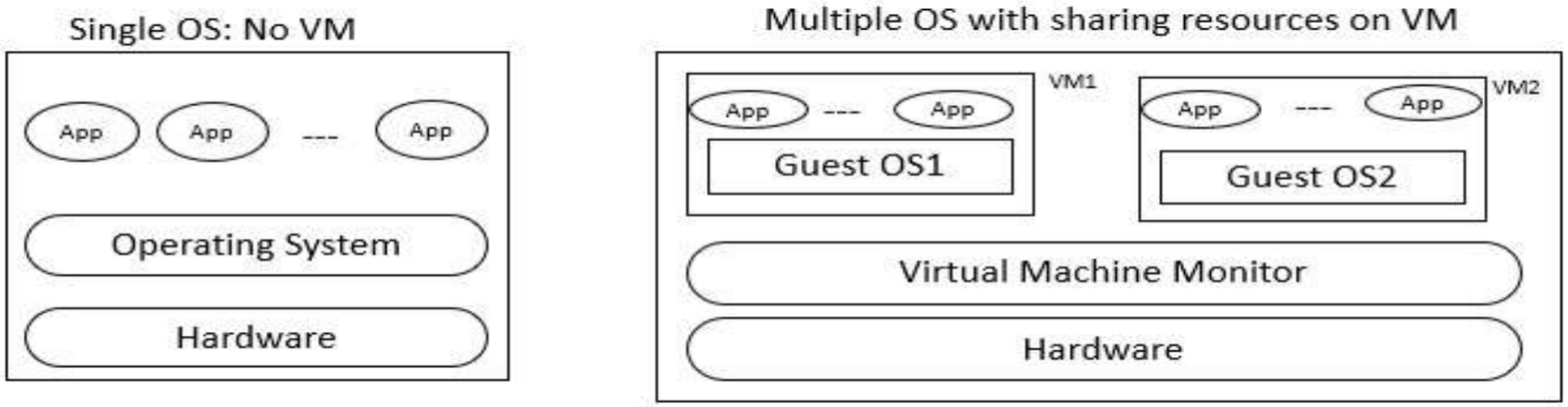
What is a VM and how does it work?

A VM provides an isolated environment for running its own OS and applications independently from the underlying host system or from other VMs on that host. The VM's OS is commonly referred to as the *guest OS*, and it can be the same as or different from the host OS or the other VMs. In this way, a single computer can host multiple VMs, all running different OSes and applications, without affecting or interfering with each other. The VM is still dependent on the host's physical resources, but those resources are virtualized and distributed across the VMs and can be reassigned as necessary, making it possible to run different environments simultaneously, as well as accommodate fluctuating workloads.

A computer that hosts VMs requires specialized software called a ***hypervisor***. The hypervisor emulates the computer's CPU, memory, hard disk, network and other hardware resources, creating a pool of resources that can be allocated to the individual VMs according to their specific requirements. The hypervisor can support multiple virtual hardware platforms that are isolated from each other, enabling VMs to run Linux and Windows Server OSes on the same physical host.

Type 1. Also referred to as a *bare-metal hypervisor*, this type of hypervisor runs directly on the physical host machine and has direct access to its hardware.

Type 2. Sometimes called a *hosted hypervisor*, a Type 2 hypervisor is installed on top of the host machine's OS, which manages calls to the hardware resources.



Why use a VM?

•VMs help organizations consolidate servers and better utilize hardware resources. Because a single server can run multiple VMs simultaneously. organizations can use resources on a single server more efficiently,

•VMs provide isolated environments, making it possible to run different types of OSes and applications on a single server.

•VMs make it easy to scale applications and accommodate fluctuating workloads.

•Organizations also turn to VMs because of the extra layer of security they provide against potential threats. If a VM is compromised, it can be deleted or rolled back to a recent backup or snapshot. Because it's isolated from the host and other VMs, the threat is limited to that VM.

•VMs make it easier to manage multiple environments running different OSes. Because workloads are consolidated on fewer servers, there are also fewer physical systems to deploy and maintain. In addition, most virtualization platforms make it possible to manage the VMs from a single interface, even if those VMs are distributed across multiple hosts.

Advantages of VMs

•Virtualization limits costs by reducing the need for physical hardware systems. This reduces the number of servers that must be deployed and the associated maintenance costs. It also lowers the demand for power and cooling.

•VMs are isolated, self-contained environments that can run different types of applications and OSes on the same server, eliminating potential contention and security issues, as well as the need to deploy multiple physical servers.

•VMs can be easily moved, copied and reassigned between host servers, as well as between on-premises and cloud environments, improving hardware resource utilization, while making it easier to scale applications.

•VMs ease management in multiple ways. Administrators, developers and testers can quickly deploy VMs, and multiple VMs can be easily managed from a centralized interface. Admins can also take advantage of virtual environments to simplify backups, disaster recovery (DR), new deployments and basic system administration tasks.

•Because VMs operate in isolated environments, they can provide an extra level of protection against malicious attacks. They also support such features as snapshots and backups, which make it easy to roll back a VM in the event the current one becomes compromised or corrupted.

•VMs don't require specialized or hypervisor-specific hardware. However, the host computer needs more bandwidth, storage and processing capacity than a traditional server or desktop if the physical hardware is going to host multiple running VMs. Because VMs on a physical host can consume unequal resource quantities -- one might hog the available physical storage, while another stores little -- IT professionals must balance VMs with available resources. Fortunately, virtualization platforms ease the process.

Two types of VMs

VMs are often categorized by the type of hypervisor that manages them or by the type of workloads they support. However, VMs are also categorized by VM type:

Process VMs. A process VM is a temporary, platform-independent programming environment for executing a single process as an application. The environment provides a high-level abstraction that masks the underlying hardware or OS. A process VM is created when the process starts and is destroyed when the process ends. Two common examples of process VMs are Java Virtual Machine, which is part of the Java platform, and Common Language Runtime, which is used for the .NET Framework.

System VMs. A system VM is a fully virtualized environment that's hosted on a physical server and runs its own OS. The VM shares the host's physical resources but provides a complete environment for running applications and services, like a physical machine. Common examples of system VMs include those supported by virtualization platforms such as VMware vSphere and Microsoft Hyper-V.

•What is Windows Operating System?

• □ Windows OS, computer operating system (OS) developed by Microsoft Corporation to run personal computers (PCs). Featuring the first graphical user interface (GUI) for IBMcompatible PCs, the Windows OS soon dominated the PC market. Approximately 90 percent of PCs run some version of Windows.

• □ The first version of Windows, released in 1985, was simply a GUI offered as an extension of Microsoft's existing disk operating system, or MS-DOS.

•□ Subsequent versions introduced greater functionality, including native Windows File Manager, Program Manager, and Print Manager programs, and a more dynamic interface.

• Microsoft also developed specialized Windows packages, including the networkable Windows for Workgroups and the high-powered Windows NT, aimed at businesses. The 1995 consumer release Windows 95 fully integrated Windows and DOS and offered built-in Internet support, including the **World Wide Web browser Internet Explorer**.

•□ With the 2001 release of Windows XP, Microsoft united its various Windows packages under a single banner, offering multiple editions for consumers, businesses, multimedia developers, and others.

•The highly successful XP standard was succeeded in late 2006 by Windows Vista, which experienced a troubled rollout and met with considerable marketplace resistance, quickly acquiring a reputation for being a large, slow, and resource-consuming system. Responding to Vista's disappointing adoption rate, Microsoft developed Windows 7, an OS whose interface was similar to that of Vista but was met with enthusiasm for its noticeable speed improvement and its modest system requirements.

History of Window

➢This is where it all started for Windows. The original Windows 1 was released in November 1985 and was Microsoft's first true attempt at a graphical user interface in 16-bit. Development was spearheaded by Microsoft founder Bill Gates and ran on top of MS-DOS, which relied on command-line input.

➢Two years after the release of Windows 1, Microsoft's Windows 2 replaced it in December 1987. The big innovation for Windows 2 was that windows could overlap each other, and it also introduced the ability to minimise or maximise windows instead of "iconising" or "zooming". The control panel, where various system settings and configuration options were collected together in one place, was introduced in Windows 2 and survives to this day. Microsoft Word and Excel also made their first appearances running on Windows

➢The first Windows that required a hard drive launched in 1990. Windows 3 was the first version to see more widespread success and be considered a challenger to Apple's Macintosh and the Commodore Amiga graphical user interfaces, coming pre-installed on computers from PCcompatible manufacturers including Zenith Data Systems. Windows 3.1 released in 1992.

➢As the name implies, Windows 95 arrived in August 1995 and with it brought the first ever Start button and Start menu.

➢Released in June 1998, Windows 98 built on Windows 95 and brought with it IE 4, Outlook Express, Windows Address Book, Microsoft Chat and NetShow Player, which was replaced by Windows Media Player 6.2 in Windows 98 Second Edition in 1999.

➢Windows ME Considered a low point in the Windows series by many – at least, until they saw Windows Vista – Windows Millennium Edition was the last Windows to be based on MS-DOS, and the last in the Windows 9x line. Released in September 2000, it was the consumer-aimed operating system twined with Windows 2000 aimed at the enterprise market. It introduced some important concepts to consumers, including more automated system recovery tools.

➢The enterprise twin of ME, Windows 2000 was released in February 2000 and was based on Microsoft's business-orientated system Windows NT and later became the basis for Windows XP.

➢Windows XP was released in October 2001 and brought Microsoft's enterprise line and consumer line of operating systems under one roof. It was based on Windows NT like Windows 2000, but brought the consumer-friendly elements from Windows ME. The Start menu and task bar got a visual overhaul, bringing the familiar green Start button, blue task bar and vista wallpaper, along with various shadow and other visual effects.

➢Windows XP stayed the course for close to six years before being replaced by Windows Vista in January 2007. Vista updated the look and feel of Windows with more focus on transparent elements, search and security.

➢, Windows 7 was first released in October 2009. It was intended to fix all the problems and criticism faced by Vista, with slight tweaks to its appearance and a concentration on user-friendly features and less "dialogue box overload".

➢Windows 8 Released in October 2012, Windows 8 was Microsoft's most radical overhaul of the Windows interface, ditching the Start button and Start menu in favour of a more touch-friendly Start screen. A free point release to Windows 8 introduced in October 2013, Windows 8.1

➢On September 30, 2014, Microsoft announced Windows 10 as the successor to Windows 8.1.

➢On June 24, 2021, Windows 11 was announced as the successor to Windows 10 during a livestream. The new operating system was designed to be more user-friendly and understandable. It was released on October 5, 2021.

What is Unix ?

The Unix operating system is a set of programs that act as a link between the computer and the user.

•Unix was originally developed in 1969 by a group of AT&T employees Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna at Bell Labs.

•There are various Unix variants available in the market. Solaris Unix, AIX, HP Unix and BSD are a few examples. Linux is also a flavor of Unix which is freely available.

•Several people can use a Unix computer at the same time; hence Unix is called a multiuser system.

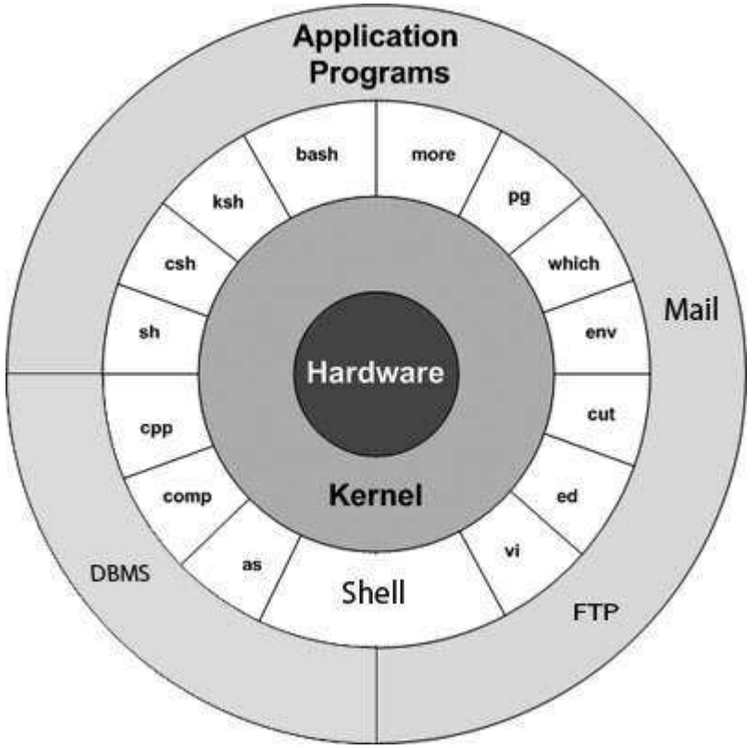
•A user can also run multiple programs at the same time; hence Unix is a multitasking environment.

Unix Architecture

Kernel – The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.

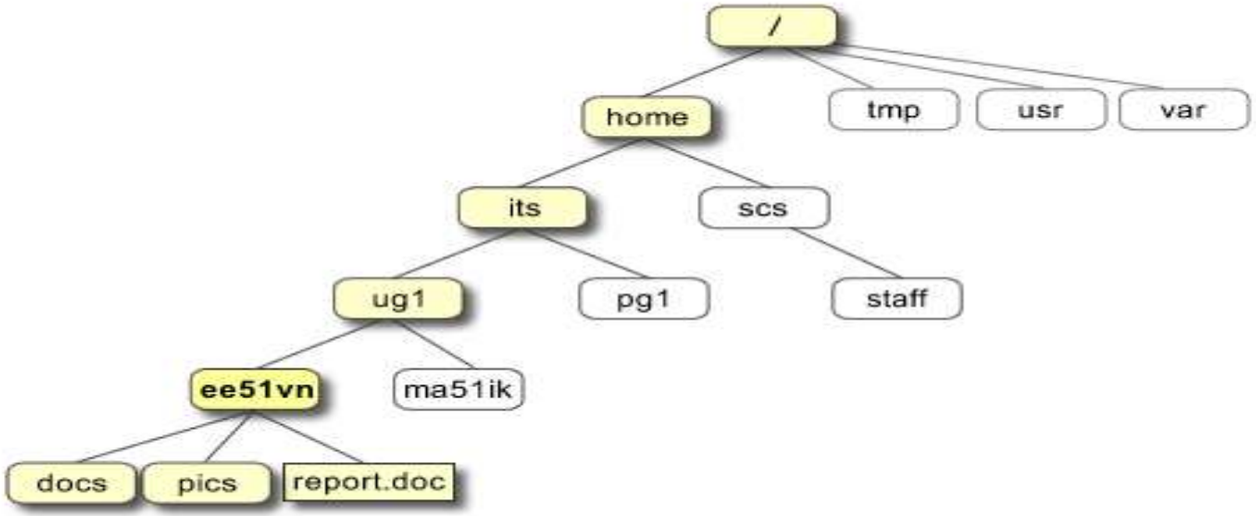
Shell – The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want

Files and Directories – All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the **filesystem**.

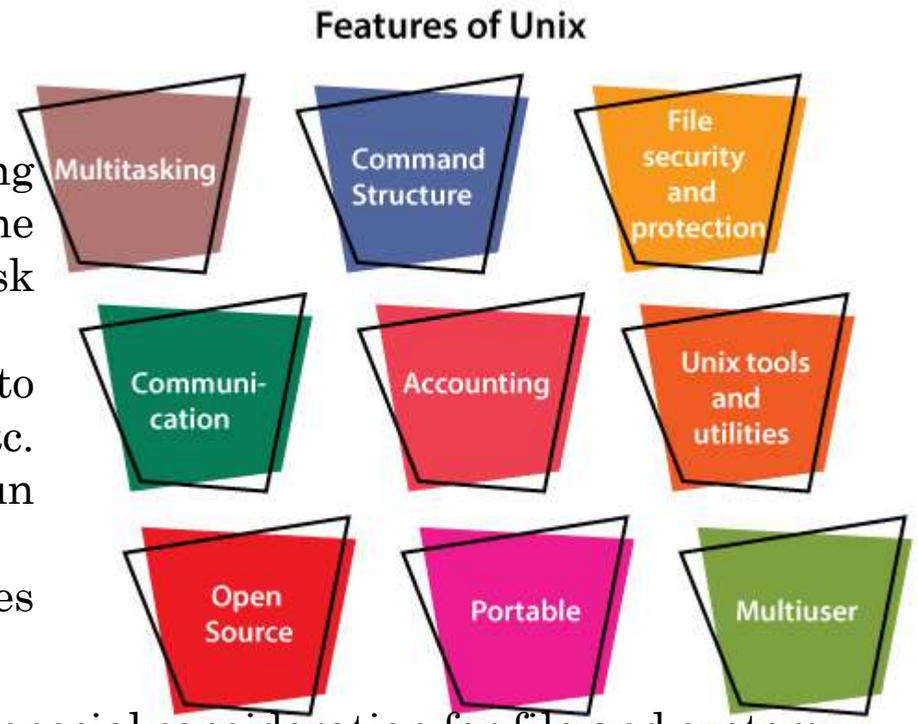


All the files are grouped together in the directory structure. The file-system is arranged in a hierarchical structure, like an inverted tree. The top of the hierarchy is traditionally called **root** (written as a slash /)

home directory **"ee51vn"** contains two sub-directories (**docs**and **pics**) and a file called **report.doc**.



Features of UNIX Operating System:



Multitasking: A UNIX operating system is a multitasking operating system that allows you to initiate more than one task from the same terminal so that one task is performed as a foreground and the other task as a background process.

Multi-user: UNIX operating system supports more than one user to access computer resources like main memory, hard disk, tape drives, etc. Multiple users can log on to the system from different terminals and run different jobs that share the resources of a command terminal.

Portability: This feature makes the UNIX work on different machines and platforms with the ease.

File Security and Protection: Being a multi-user system, UNIX makes special consideration for file and system security. UNIX has different levels of security using assigning username and password to individual users ensuring the authentication, at the level providing file access permission viz. read, write and execute and lastly file encryption to change the file into an unreadable format.

Command Structure: UNIX commands are easy to understand and simple to use.

Communication: In UNIX, communication is an excellent feature that enables the user to communicate worldwide. It supports various communication facilities provided using the write command, mail command, talk command, etc.

Open Source: UNIX operating system is open source it means it is freely available to all and is a community-based development project.

Accounting: UNIX keeps an account of jobs created by the user. This feature enhances the system performance in terms of CPU monitoring and disk space checking.

Features	UNIX Operating System	Windows Operating System
User-Interface	It comes with a Command Line Interface (CLI).	It comes with a Graphical User Interface (GUI).
Licensing	It is a free and open-source operating system.	It is a licensed operating system.
Security	It is more secure because all system updates require explicit user permission.	It is less secure than UNIX operating system.
Processing	It supports multiprocessing.	It doesn't support multiprocessing.
Case-Sensitive	It is fully case-sensitive, and files can be considered separate files.	It has case sensitivity as an option.
Basic	It is a command-based operating system.	It is a menu-based operating system.
Hardware	In a UNIX system, hardware support is limited. Some hardware could not have drivers built-in.	Almost all hardware has drivers available.
File System	It uses the Unix File System (UFS), which includes the STD.ERR and STD.IO file systems.	It makes use of the New Technology File System (NTFS) and the File Allocation System (FAT32).
Reliability	Unix and its distributions are well known for their high level of stability.	Although Windows has become more stable in recent years, it still falls short of the reliability offered by Unix systems.
Data Backup and Recovery	Creating a backup and recovery system in UNIX is time-consuming, but it is becoming easier with the release of new Unix distributions.	It contains a built-in backup and recovery system that makes it more user-friendly.

Process

A process is basically a program in execution. A process is defined as an entity which represents the basic unit of work to be implemented in the system. In order to accomplish its task, process needs the computer resources. There may exist more than one process in the system which may require the same resource at the same time. Therefore, the operating system has to manage all the processes and the resources in a convenient and efficient way.

When a program is loaded into the memory and it becomes a process, it can be divided into four sections ─ stack, heap, text and data. **Process memory** is divided into four sections for efficient working .

1	**Stack** The process Stack contains the temporary data such as method/function parameters, return address and local variables.
2	**Heap** This is dynamically allocated memory to a process during its run time.
3	**Text** This includes the current activity represented by the value of Program Counter and the contents of the processor's registers.
4	**Data** This section contains the global and static variables.

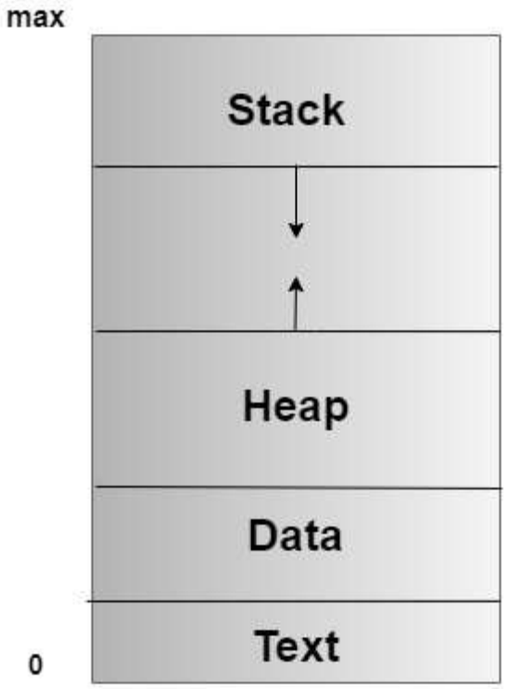
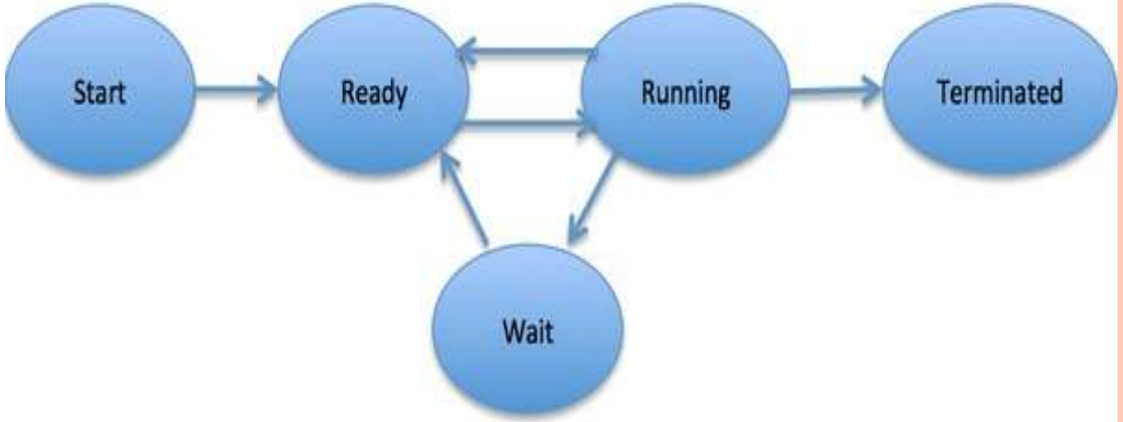


Figure: Process in the Memory

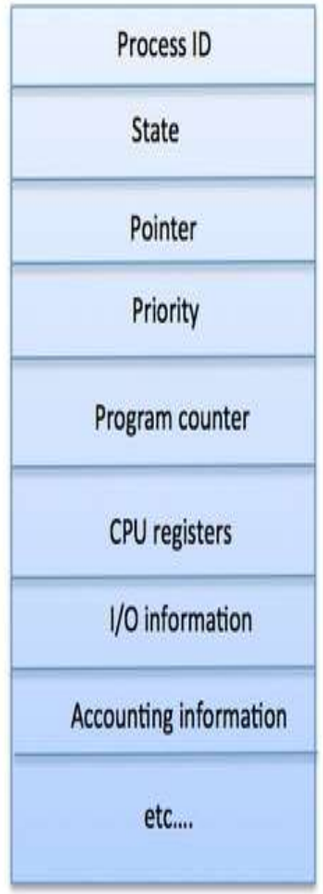
1	**Start** This is the initial state when a process is first started/created.
2	**Ready** The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run. Process may come into this state after **Start** state or while running it by but interrupted by the scheduler to assign CPU to some other process.
3	**Running** Once the process has been assigned to a processor by the OS scheduler, the process state is set to running and the processor executes its instructions.
4	**Waiting** Process moves in to the waiting state if it needs to wait for a resource, such as waiting for user input, or waiting for a file to become available.
5	**Terminated or Exit** Once the process finishes its execution, or it is terminated by the operating system, it is moved to the terminated state where it waits to be removed from main memory.



Process Control Block (PCB)

A Process Control Block is a data structure maintained by the Operating System for every process. The PCB is identified by an integer process ID (PID). A PCB keeps all the information needed to keep track of a process as listed below in the table. The PCB is maintained for a process throughout its lifetime, and is deleted once the process terminates.

Process State The current state of the process i.e., whether it is ready, running, waiting, or whatever.
Process privileges This is required to allow/disallow access to system resources.
Process ID Unique identification for each of the process in the operating system.
Pointer A pointer to parent process.
Program Counter Program Counter is a pointer to the address of the next instruction to be executed for this process.
CPU registers Various CPU registers where process need to be stored for execution for running state.
CPU Scheduling Information Process priority and other scheduling information which is required to schedule the process.
Memory management information This includes the information of page table, memory limits, Segment table depending on memory used by the operating system.
Accounting information This includes the amount of CPU used for process execution, time limits, execution ID etc.
IO status information This includes a list of I/O devices allocated to the process.



Module 1

Introduction Concept of Operating Systems,
Generations of Operating systems,
Types of Operating Systems,
OS Services

Operating System

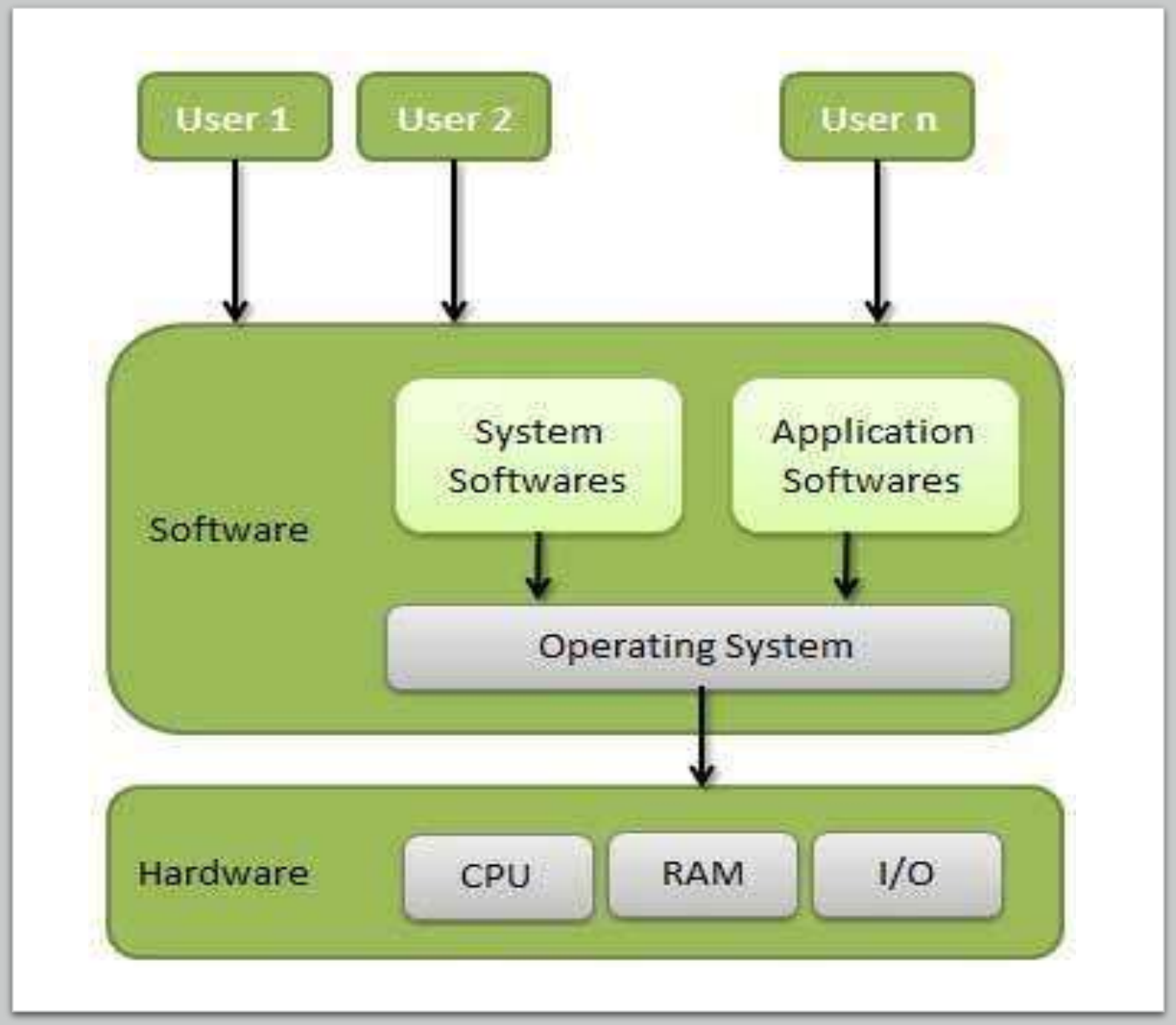
- An operating system (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. The operating system is a vital component of the system software in a computer system.

Why to Learn Operating System?

- An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
- Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.

Definition of OS

- An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.



Important functions of an operating System

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

Applications of Operating System

- **Security** − By means of password and similar other techniques, it prevents unauthorized access to programs and data.
- **Control over system performance** − Recording delays between request for a service and response from the system.
- **Job accounting** − Keeping track of time and resources used by various jobs and users.
- **Error detecting aids** − Production of dumps, traces, error messages, and other debugging and error detecting aids.
- **Coordination between other software's and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Memory Management

- Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.
- Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management −
- Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
- In multiprogramming, the OS decides which process will get memory when and how much.
- Allocates the memory when a process requests it to do so.
- De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

- In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**. An Operating System does the following activities for processor management −
- Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
- Allocates the processor (CPU) to a process.
- De-allocates processor when a process is no longer required.

Device Management

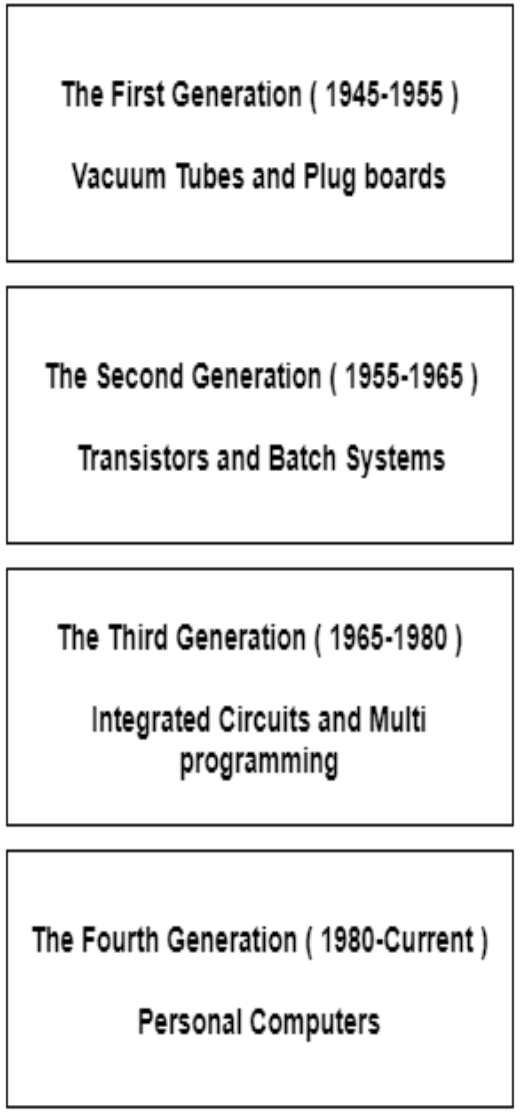
- An Operating System manages device communication via their respective drivers. It does the following activities for device management −
- Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management

- A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.
- An Operating System does the following activities for file management −
- Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

Operating System Generations

- Operating Systems have evolved over the years. So, their evolution through the years can be mapped using generations of operating systems. There are four generations of operating systems.



The First Generation (1945 - 1955): Vacuum Tubes and Plugboards

- Digital computers were not constructed until the second world war. Calculating engines with mechanical relays were built at that time. However, the mechanical relays were very slow and were later replaced with vacuum tubes. These machines were enormous but were still very slow.
- These early computers were designed, built and maintained by a single group of people. Programming languages were unknown and there were no operating systems so all the programming was done in machine language. All the problems were simple numerical calculations.
- By the 1950’s punch cards were introduced and this improved the computer system. Instead of using plugboards, programs were written on cards and read into the system.

The Second Generation (1955 - 1965): Transistors and Batch Systems

- Transistors led to the development of the computer systems that could be manufactured and sold to paying customers. These machines were known as mainframes and were locked in air-conditioned computer rooms with staff to operate them.
- The Batch System was introduced to reduce the wasted time in the computer. A tray full of jobs was collected in the input room and read into the magnetic tape. After that, the tape was rewound and mounted on a tape drive. Then the batch operating system was loaded in which read the first job from the tape and ran it. The output was written on the second tape. After the whole batch was done, the input and output tapes were removed and the output tape was printed.

The Third Generation (1965 - 1980): Integrated Circuits and Multiprogramming

- Until the 1960’s, there were two types of computer systems i.e the scientific and the commercial computers. These were combined by IBM in the System/360. This used integrated circuits and provided a major price and performance advantage over the second generation systems.
- The third generation operating systems also introduced multiprogramming. This meant that the processor was not idle while a job was completing its I/O operation. Another job was scheduled on the processor so that its time would not be wasted.

The Fourth Generation (1980 - Present): Personal Computers

- Personal Computers were easy to create with the development of large-scale integrated circuits. These were chips containing thousands of transistors on a square centimeter of silicon. Because of these, microcomputers were much cheaper than minicomputers and that made it possible for a single individual to own one of them.
- The advent of personal computers also led to the growth of networks. This created network operating systems and distributed operating systems. The users were aware of a network while using a network operating system and could log in to remote machines and copy files from one machine to another.

Types of Operating System

- Batch operating system
- Time-sharing operating systems
- Distributed operating System
- Network operating System
- Real Time operating System

Batch operating system

The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

The problems with Batch Systems are as follows −

- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
- Difficult to provide the desired priority.

Time-sharing operating systems

- Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing.
- The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of Multiprogrammed batch systems, the objective is to maximize processor use, whereas in Time-Sharing Systems, the objective is to minimize response time.

Time-sharing operating systems

- Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response. For example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation. That is, if **n** users are present, then each user can get a time quantum. When the user submits the command, the response time is in few seconds at most.
- The operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

Time-sharing operating systems

Advantages of Timesharing operating systems:

- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages of Time-sharing operating systems :

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

Distributed operating System

- Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly.
- The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as **loosely coupled systems** or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on.

Distributed operating System

The advantages of distributed systems :

- With resource sharing facility, a user at one site may be able to use the resources available at another.
- Speedup the exchange of data with one another via electronic mail.
- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

Network operating System

- A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.
- Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

Network operating System

The advantages of network operating systems:

- Centralized servers are highly stable.
- Security is server managed.
- Upgrades to new technologies and hardware can be easily integrated into the system.
- Remote access to servers is possible from different locations and types of systems.

The disadvantages of network operating systems:

- High cost of buying and running a server.
- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

Real Time operating System

- A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. The time taken by the system to respond to an input and display of required updated information is termed as the **response time**. So in this method, the response time is very less as compared to online processing.
- Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application. A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail. For example, Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

Real Time operating System

There are two types of real-time operating systems.

Hard real-time systems

- Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM. In these systems, virtual memory is almost never found.

Soft real-time systems

- Soft real-time systems are less restrictive. A critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc.

Operating System - Services

An Operating System provides services to both the users and to the programs.

- It provides programs an environment to execute.
- It provides users the services to execute the programs in a convenient manner.

Following are a few common services provided by an operating system −

- **Program execution**
- **I/O operations**
- **File System manipulation**
- **Communication**
- **Error Detection**
- **Resource Allocation**
- **Protection**

Program execution

Operating systems handle many kinds of activities from user programs to system programs like printer spooler, name servers, file server, etc. Each of these activities is encapsulated as a process.

A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use). Following are the major activities of an operating system with respect to program management −

- Loads a program into memory.
- Executes the program.
- Handles program's execution.
- Provides a mechanism for process synchronization.
- Provides a mechanism for process communication.
- Provides a mechanism for deadlock handling.

I/O Operation

An I/O subsystem comprises of I/O devices and their corresponding driver software. Drivers hide the peculiarities of specific hardware devices from the users.

An Operating System manages the communication between user and device drivers.

- I/O operation means read or write operation with any file or any specific I/O device.
- Operating system provides the access to the required I/O device when required.

File system manipulation

- A file represents a collection of related information. Computers can store files on the disk (secondary storage), for long-term storage purpose. Examples of storage media include magnetic tape, magnetic disk and optical disk drives like CD, DVD. Each of these media has its own properties like speed, capacity, data transfer rate and data access methods.
- A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

File system manipulation

Following are the major activities of an operating system with respect to file management:

- Program needs to read a file or write a file.
- The operating system gives the permission to the program for operation on file.
- Permission varies from read-only, read-write, denied and so on.
- Operating System provides an interface to the user to create/delete files.
- Operating System provides an interface to the user to create/delete directories.
- Operating System provides an interface to create the backup of file system.

Communication

- In case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, the operating system manages communications between all the processes. Multiple processes communicate with one another through communication lines in the network.
- The OS handles routing and connection strategies, and the problems of contention and security.

Communication

Following are the major activities of an operating system with respect to communication −

- Two processes often require data to be transferred between them
- Both the processes can be on one computer or on different computers, but are connected through a computer network.
- Communication may be implemented by two methods, either by Shared Memory or by Message Passing.

Error handling

Errors can occur anytime and anywhere. An error may occur in CPU, in I/O devices or in the memory hardware. Following are the major activities of an operating system with respect to error handling −

- The OS constantly checks for possible errors.
- The OS takes an appropriate action to ensure correct and consistent computing.

Resource Management

- In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job. Following are the major activities of an operating system with respect to resource management −
- The OS manages all kinds of resources using schedulers.
- CPU scheduling algorithms are used for better utilization of CPU.

Protection

- Considering a computer system having multiple users and concurrent execution of multiple processes, the various processes must be protected from each other's activities.
- Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an operating system with respect to protection −
- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

Thank You

Module 1

System Calls, Structure of OS

System Call

- In computing, a **system call** is the programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on. A system call is a way for programs to **interact with the operating system**.
- A computer program makes a system call when it makes a request to the operating system's kernel. System call **provides** the services of the operating system to the user programs via Application Program Interface(API).

System Call

- It provides an interface between a process and operating system to allow user-level processes to request services of the operating system.
- System calls are the only entry points into the kernel system.
- All programs needing resources must use system calls.

Services Provided by System Calls

1. Process creation and management
2. Main memory management
3. File Access, Directory and File system management
4. Device handling(I/O)
5. Protection
6. Networking

Types of System Calls

1. **Process control:** end, abort, create, terminate, allocate and free memory.
2. **File management:** create, open, close, delete, read file etc.
3. Device management
4. Information maintenance
5. Communication

Process control

- end, abort
- load, execute
- create process, terminate process
- get process attributes, set process attributes
- wait for time
- wait event, signal event
- allocate and free memory

File management

- create file, delete file
- open, close
- read, write, reposition
- get file attributes, set file attributes

Device management

- request device, release device
- read, write, reposition
- get device attributes, set device attributes
- logically attach or detach devices

Information maintenance

- get time or date, set time or date
- get system data, set system data
- get process, file, or device attributes
- set process, file, or device attributes

Communications

- create, delete communication connection
- send, receive messages
- transfer status information
- attach or detach remote devices

Examples of Windows and Unix System Calls

	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitlializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

System Structure

- An OS Contains a kernel, command processor or shell & graphical user interface.
- Kernel
- Shell
- Command Processor
- Graphical User Interface

Kernel

- Kernel is central component of an operating system that manages operations of computer and hardware. It basically manages operations of memory and CPU time. It is core component of an operating system. Kernel acts as a bridge between applications and data processing performed at hardware level using inter-process communication and system calls.
- Kernel loads first into memory when an operating system is loaded and remains into memory until operating system is shut down again. It is responsible for various tasks such as disk management, task management, and memory management.

Kernel

- It decides which process should be allocated to processor to execute and which process should be kept in main memory to execute. It basically acts as an interface between user applications and hardware. The major aim of kernel is to manage communication between software i.e. user-level applications and hardware i.e., CPU and disk memory.

Objectives of Kernel

- To establish communication between user level application and hardware.
- To decide state of incoming processes.
- To control disk management.
- To control memory management.
- To control task management.

Monolithic Kernel

- It is one of types of kernel where all operating system services operate in kernel space. It has dependencies between systems components. It has huge lines of code which is complex.

Example :

- Unix, Linux, Open Vms, XTS-400
- **Advantage :**
  It has good performance.
- **Disadvantage :**
  It has dependencies between system component and lines of code in millions.

Micro Kernel

- It is kernel types which has minimalist approach. It has virtual memory and thread scheduling. It is more stable with less services in kernel space. It puts rest in user space.

Example :

- Mach, L4, AmigaOS, Minix, K42 etc
- **Advantage :**
  It is more stable.
- **Disadvantage :**
  There are lots of system calls and context switches.

Hybrid Kernel

- It is the combination of both monolithic kernel and mircrokernel. It has speed and design of monolithic kernel and modularity and stability of microkernel.

Example :

- Window NT, Netware, BeOS
- **Advantage :**
  It combines both monolithic kernel and microkernel.
- **Disadvantage :**
  It is still similar to monolithic kernel.

Exo Kernel

- It is the type of kernel which follows end-to-end principle. It has fewest hardware abstractions as possible. It allocates physical resources to applications.
- Eg: Nemesis, ExOS etc
- **Advantage :**
  It has fewest hardware abstractions.
- **Disadvantage :**
  There is more work for application developers.

Nano Kernel

- It is the type of kernel that offers hardware abstraction but without system services. Micro Kernel also does not have system services therefore the Micro Kernel and Nano Kernel have become analogous.
- **Advantage :**
  It offers hardware abstractions without system services.
- **Disadvantage :**
  It is quite same as Micro kernel hence it is less used.

Shell

- A **Shell** provides you with an interface to the Unix system. It gathers input from you and executes programs based on that input. When a program finishes executing, it displays that program's output.
- Shell is an environment in which we can run our commands, programs, and shell scripts. There are different flavors of a shell, just as there are different flavors of operating systems. Each flavor of shell has its own set of recognized commands and functions.

Shell Prompt

- The prompt, **$**, which is called the **command prompt**, is issued by the shell. While the prompt is displayed, you can type a command.
- Shell reads your input after you press **Enter**. It determines the command you want executed by looking at the first word of your input. A word is an unbroken set of characters. Spaces and tabs separate words.

Shell Types

- In Unix, there are two major types of shells −
- **Bourne shell** − If you are using a Bourne-type shell, the **$** character is the default prompt.
- **C shell** − If you are using a C-type shell, the % character is the default prompt.
- The Bourne Shell has the following subcategories −
- Bourne shell (sh)
- Korn shell (ksh)
- Bourne Again shell (bash)
- POSIX shell (sh)
- The different C-type shells follow −
- C shell (csh)
- TENEX/TOPS C shell (tcsh)

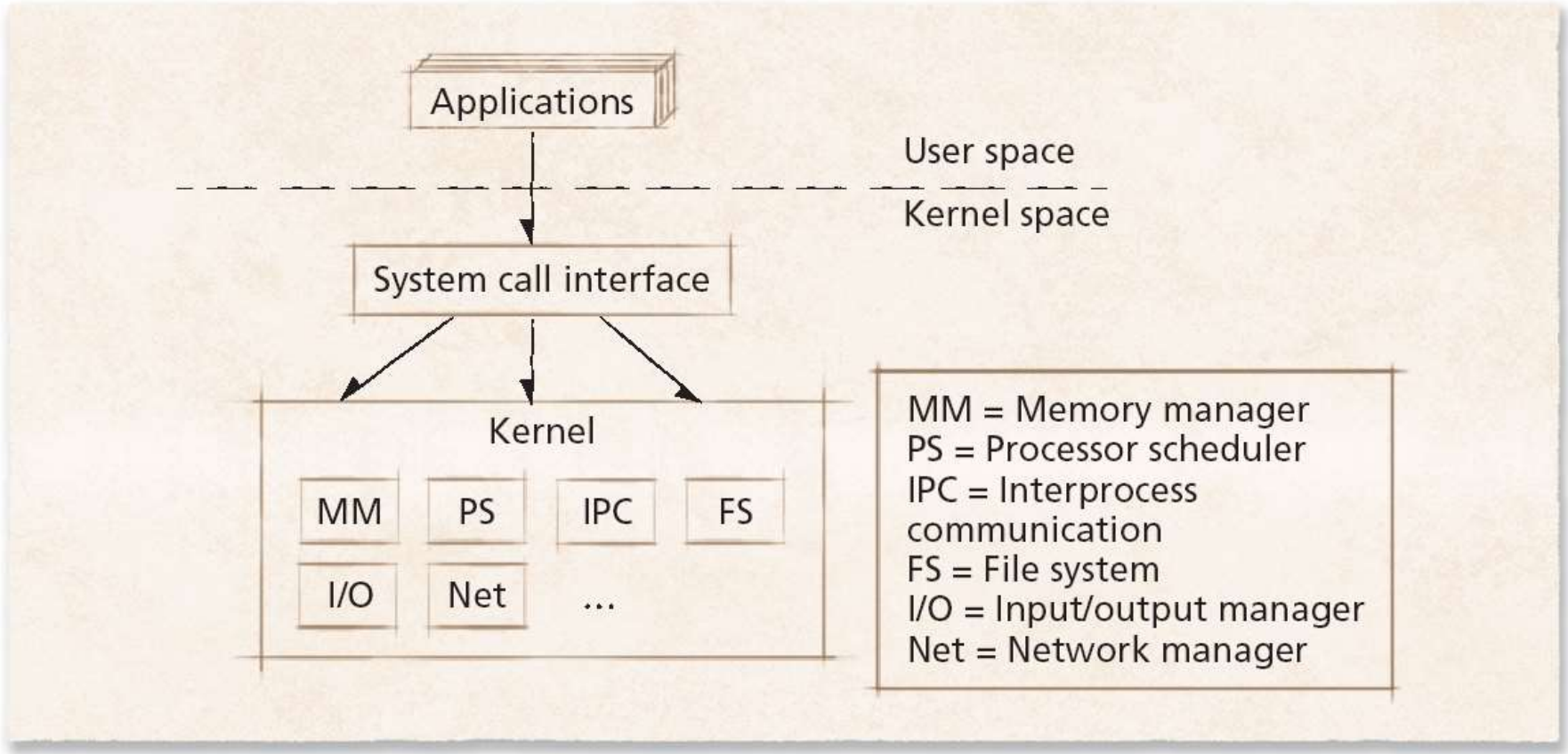
Operating System Architectures

- Today's operating systems tend to be complex
  - Provide many services
  - Support variety of hardware and software
  - Operating system architectures help manage this complexity
    - Organize operating system components
    - Specify privilege with which each component executes

Monolithic Architecture

- Monolithic operating system
  - Every component contained in kernel
    - Any component can directly communicate with any other
  - Tend to be highly efficient
  - Disadvantage is difficulty determining source of subtle errors

Monolithic Architecture



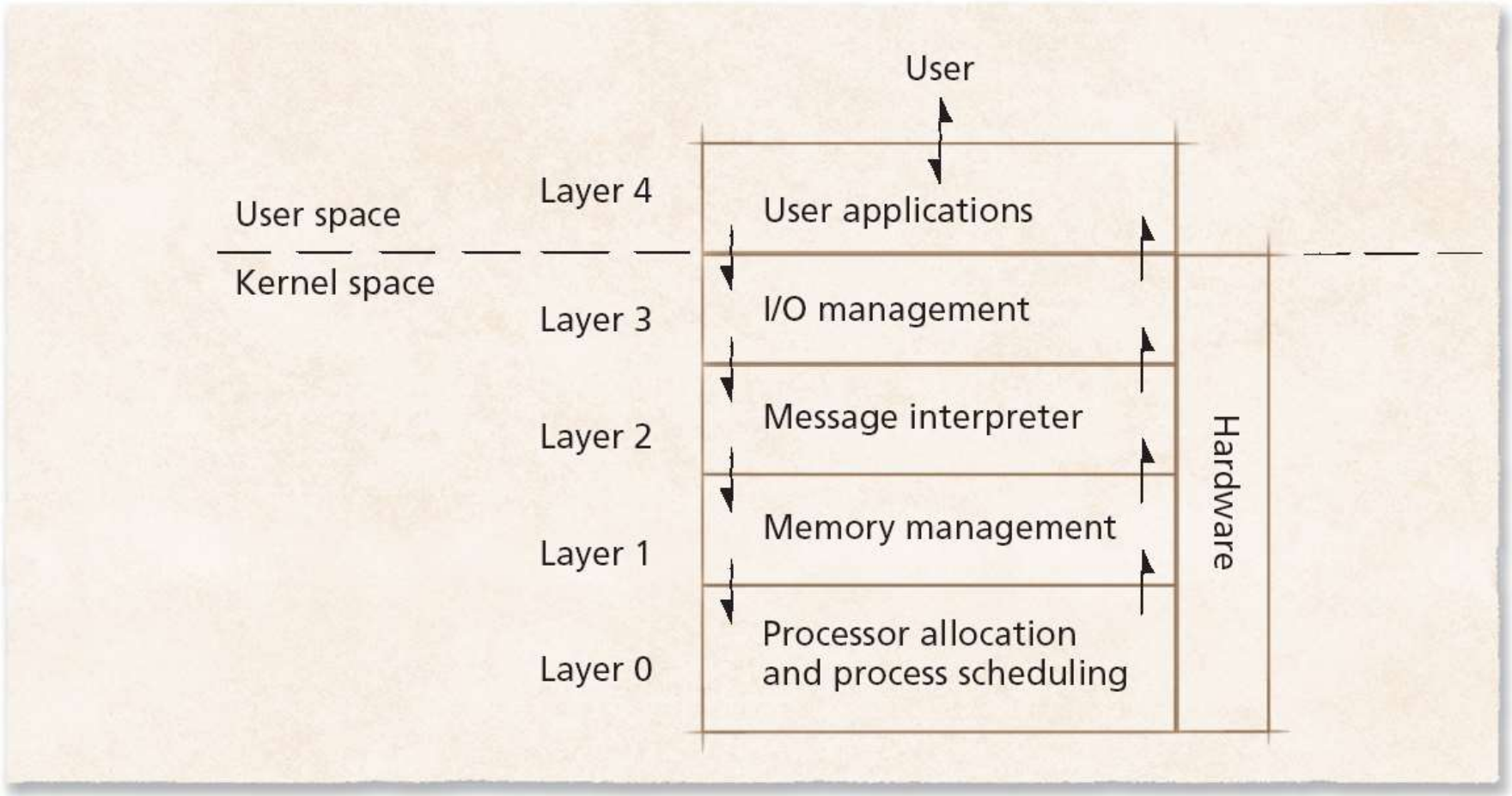
Monolithic Architecture

- Unorganized Modules
- Single Layer between Kernel & User
- No Information Hiding Concept
- Less Portable

Limitations:

- Difficult modify
- Failure of Single program crashes the whole system.
- Modules are not organized.
- Eg: MS DOS & Novell

Layered Architecture



Layered Architecture

- Layered approach to operating systems
  - Tries to improve on monolithic kernel designs
    - Groups components that perform similar functions into layers
  - Each layer communicates only with layers immediately above and below it
  - Processes' requests might pass through many layers before completion
  - System throughput can be less than monolithic kernels
    - Additional methods must be invoked to pass data and control

Layered Architecture

- Well organized in layers
- Each Layer has clearly defined functionality.
- Layer can access only above & Below layer
- Modularity
- Information Hiding Implemented

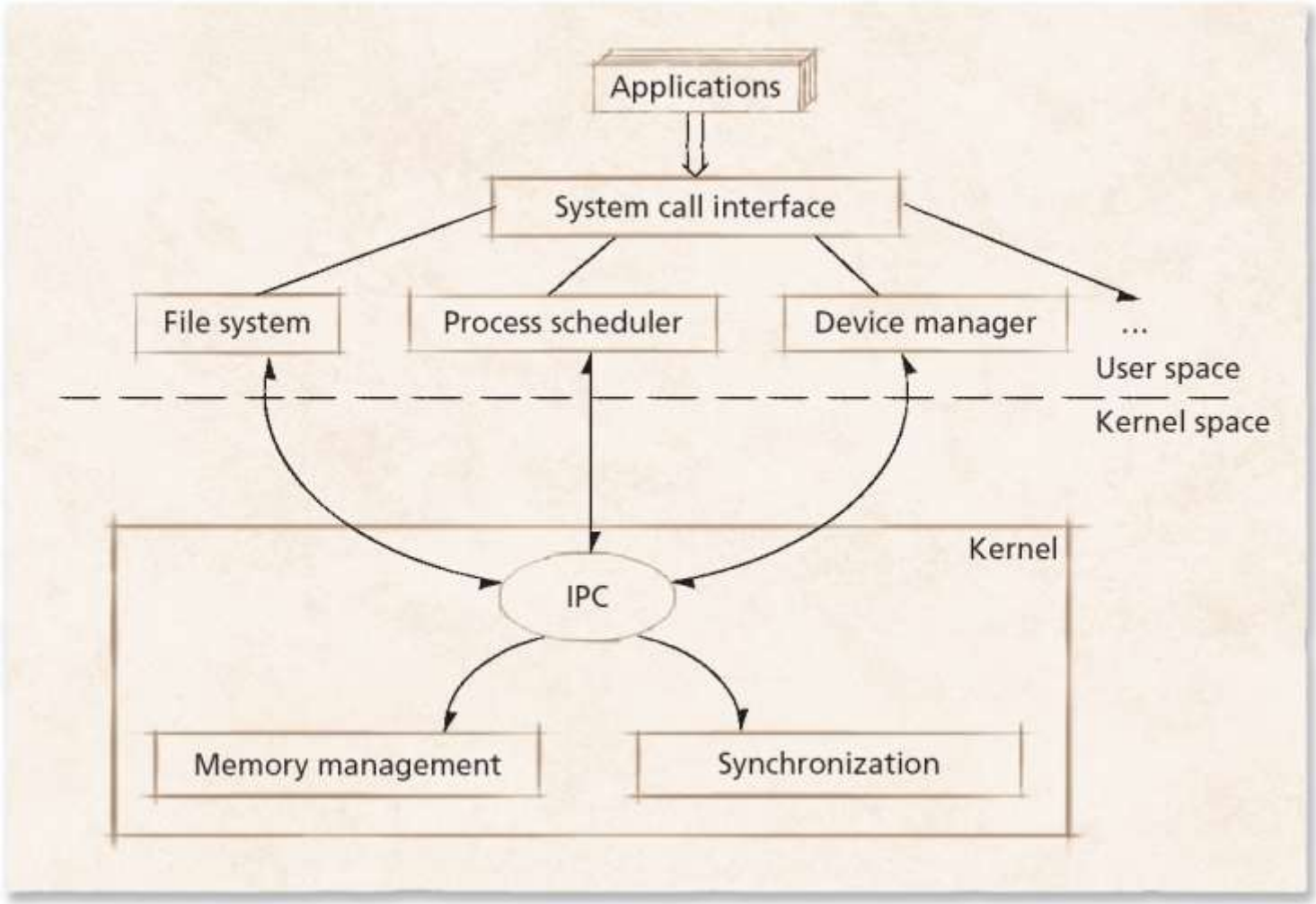
Limitations:

- Processing Time high
- Less Efficient
- Designing will be difficult
- Eg: Unix & VAX/VMS

Microkernel Architecture

- Microkernel operating system architecture
  - Provides only small number of services
    - Attempt to keep kernel small and scalable
  - High degree of modularity
    - Extensible, portable and scalable
  - Increased level of intermodule communication
    - Can degrade system performance

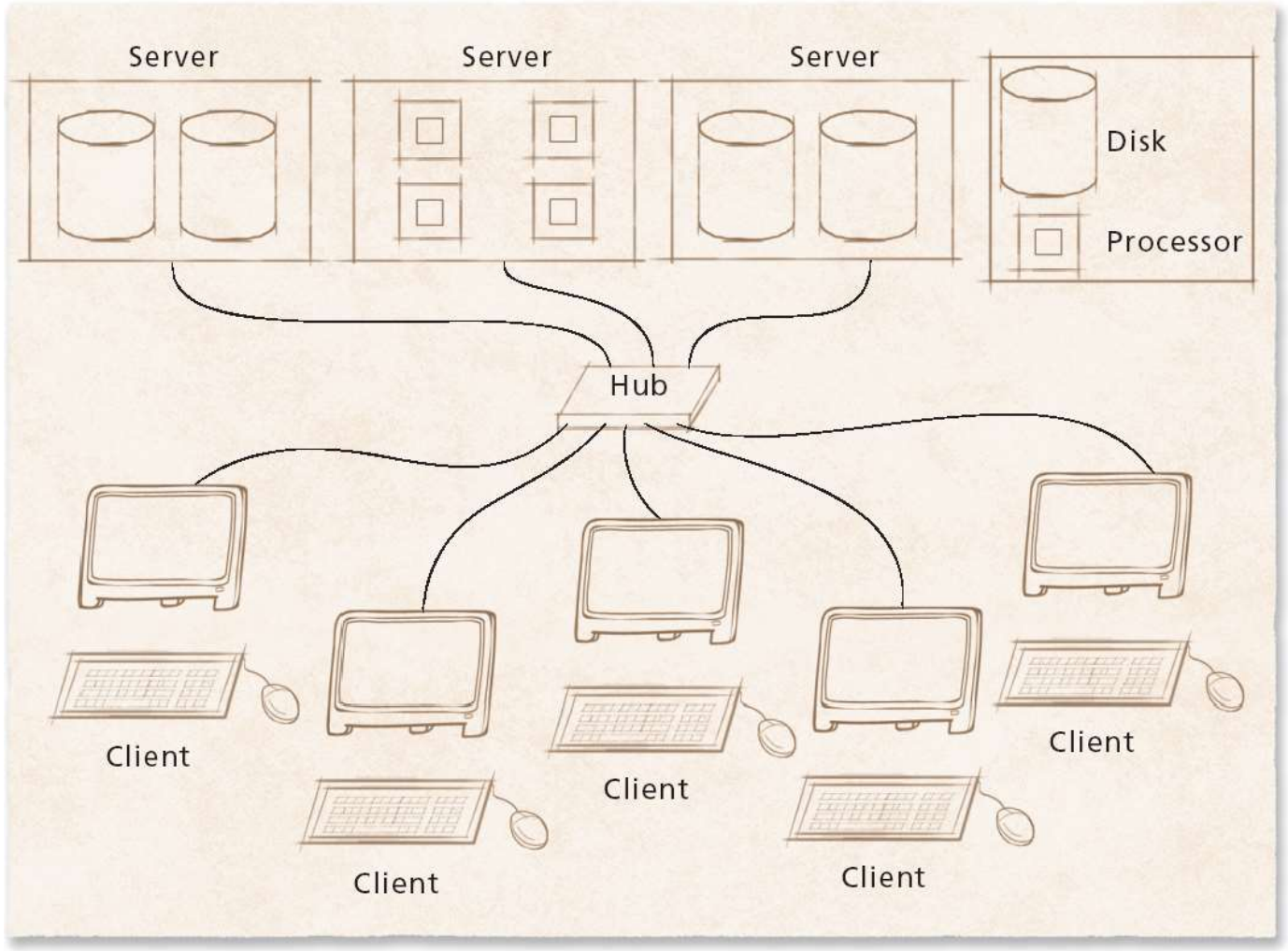
Microkernel Architecture



Networked and Distributed Operating Systems

- Network operating system
  - Runs on one computer
  - Allows its processes to access resources on remote computers
- Distributed operating system
  - Single operating system
  - Manages resources on more than one computer system
  - Goals include:
    - Transparent performance
    - Scalability
    - Fault tolerance
    - Consistency

Networked and Distributed Operating Systems



Virtual Machines in Operating System

- **Virtual Machine** abstracts the hardware of our personal computer such as CPU, disk drives, memory, NIC (Network Interface Card) etc, into many different execution environments as per our requirements, hence giving us a feel that each execution environment is a single computer. For example, VirtualBox.
- When we run different processes on an operating system, it creates an illusion that each process is running on a different processor having its own virtual memory, with the help of CPU scheduling and virtual-memory techniques. There are additional features of a process that cannot be provided by the hardware alone like system calls and a file system. The virtual machine approach does not provide these additional functionalities but it only provides an interface that is same as basic hardware. Each process is provided with a virtual copy of the underlying computer system.

Virtual Machines in Operating System

- We can create a virtual machine for several reasons, all of which are fundamentally related to the ability to share the same basic hardware yet can also support different execution environments, i.e., different operating systems simultaneously.
- The main drawback with the virtual-machine approach involves disk systems. Let us suppose that the physical machine has only three disk drives but wants to support seven virtual machines. Obviously, it cannot allocate a disk drive to each virtual machine, because virtual-machine software itself will need substantial disk space to provide virtual memory and spooling. The solution is to provide virtual disks.

Virtual Machines in Operating System

- Users are thus given their own virtual machines. After which they can run any of the operating systems or software packages that are available on the underlying machine. The virtual-machine software is concerned with multi-programming multiple virtual machines onto a physical machine, but it does not need to consider any user-support software. This arrangement can provide a useful way to divide the problem of designing a multi-user interactive system, into two smaller pieces.

- **Advantages:**

1. There are no protection problems because each virtual machine is completely isolated from all other virtual machines.
2. Virtual machine can provide an instruction set architecture that differs from real computers.
3. Easy maintenance, availability and convenient recovery.

- **Disadvantages:**

1. When multiple virtual machines are simultaneously running on a host computer, one virtual machine can be affected by other running virtual machines, depending on the workload.
2. Virtual machines are not as efficient as a real one when accessing the hardware.

Thank You

Introduction to Operating System (OS)

Course Content:

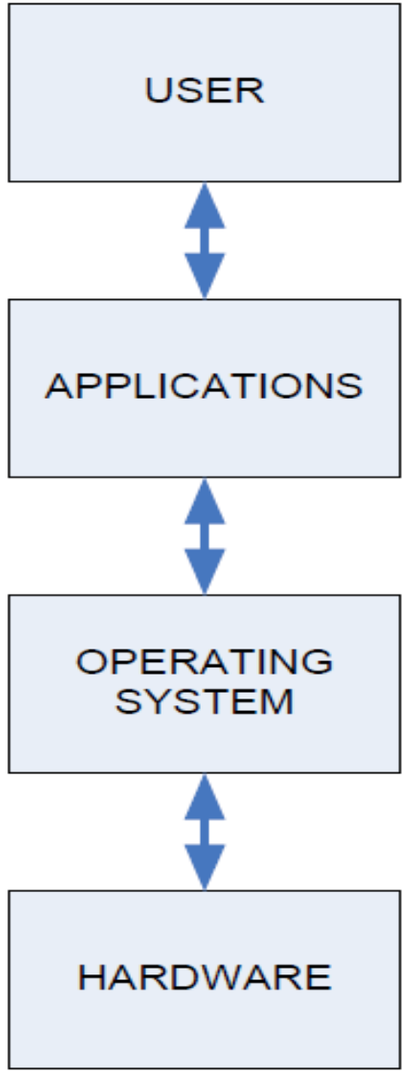
- What is an OS.
- What are its key functions.
- The evaluation of OS.
- What are the popular types of OS.
- Basics of UNIX and Windows.
- Advantages of open source OS like Linux.
- Networks OS.

What is an Operating System?

- Computer System = Hardware + Software
- Software = Application Software + System Software(OS)
- An Operating System is a system Software that acts as an intermediary/interface between a user of a computer and the computer hardware.
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

The Structure of Computer Systems

- Accessing computer resources is divided into *layers*.
- Each layer is isolated and only interacts directly with the layer below or above it.
- If we install a new hardware device
  - ✓ No need to change anything about the user/applications.
  - ✓ However, you do need to make changes to the operating system.
  - ✓ You need to install the device drivers that the operating system will use to control the new device.
- If we install a new software application
  - ✓ No need to make any changes to your hardware.
  - ✓ But we need to make sure the application is supported by the operating system
  - ✓ user will need to learn how to use the new application.
- If we change the operating system
  - ✓ Need to make sure that both applications and hardware will compatible with the new operating system.



Computer Architecture

CPU

Front Side Bus

AGP

Northern Bridge

RAM

Southern Bridge

Keyboard
PS/2

I/O Busses

ISA

COM 1 and 2
LPT

PCI

EIDE
Network adapt
Other adapters

USB

Special busses (roads) connecting all input/output devices to motherboard.

CPU – Central Processing Unit

- This is the brain of your computer.
- It performs all of the calculations.
- In order to do its job, the CPU needs commands to perform, and data to work with.
- The instructions and data travel to and from the CPU on the system bus.
- The operating system provides rules for how that information gets back and forth, and how it will be used by the CPU.

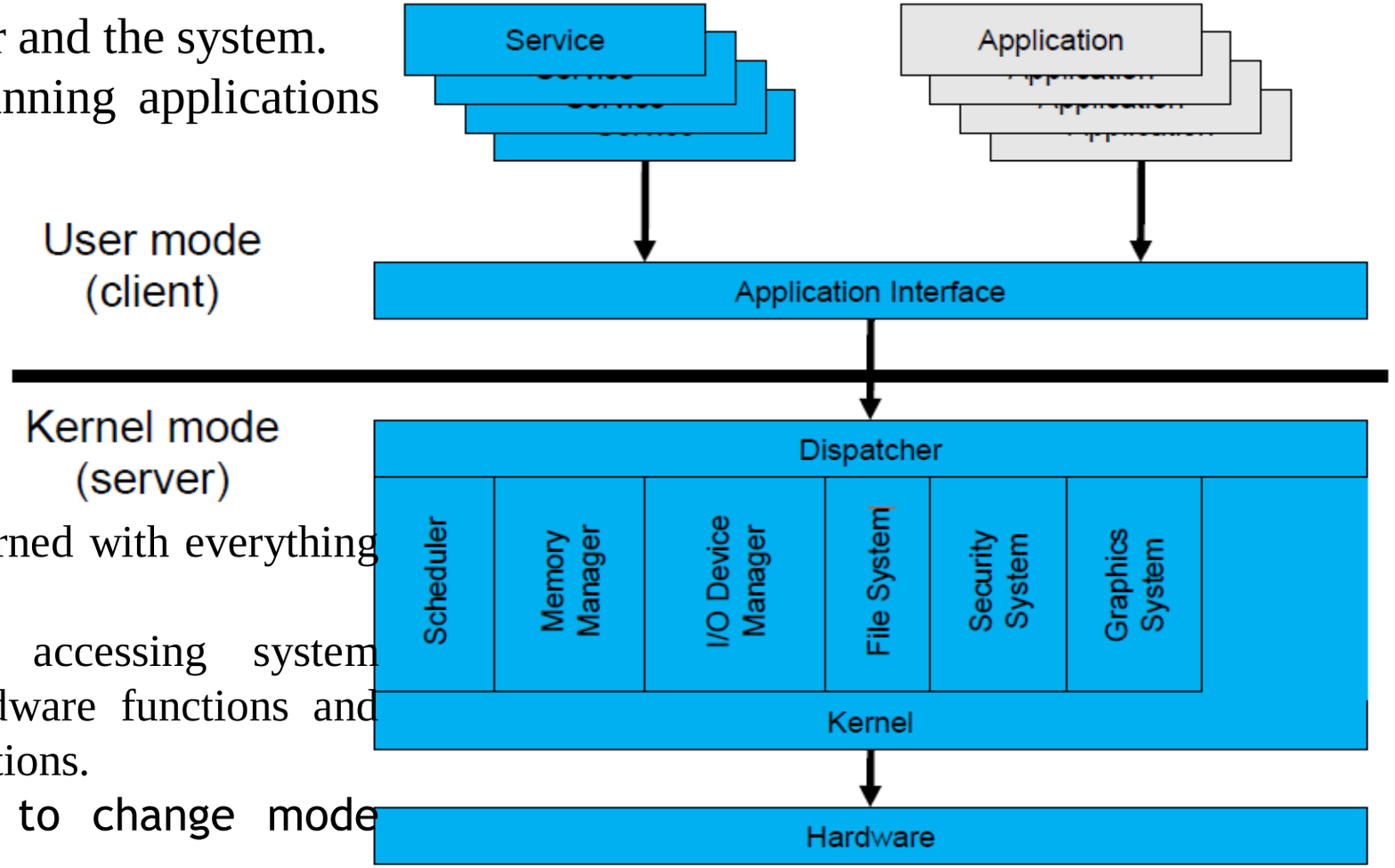
RAM – Random Access Memory

- This is like a desk, or a workspace, where your computer temporarily stores all of the information (data) and instructions (software or program code) that it is currently using.
- Each RAM chip contains millions of address spaces.
- Each address space is the same size, and has its own unique identifying number (address).
- The operating system provides the rules for using these memory spaces, and controls storage and retrieval of information from RAM.
- Device drivers for RAM chips are included with the operating system.

Problem: *If RAM needs an operating system to work, and an operating system needs RAM in order to work, how does your computer activate its RAM to load the operating system?*

Operating System Mode

- The *User Mode* is concerned with the actual interface between the user and the system.
- It controls things like running applications and accessing files.



- The *Kernel Mode* is concerned with everything running in the background.
- It controls things like accessing system resources, controlling hardware functions and processing program instructions.
- System calls are used to change mode from User to Kernel.

Kernel

- Kernel is a software code that reside in central core of OS. It has complete control over system.
- When operation system boots, kernel is first part of OS to load in main memory.
- Kernel remains in main memory for entire duration of computer session. The kernel code is usually loaded in to protected area of memory.
- Kernel performs it's task like executing processes and handling interrupts in kernel space.
- User performs it's task in user area of memory.
- This memory separation is made in order to prevent user data and kernel data from interfering with each other.
- Kernel does not interact directly with user, but it interacts using SHELL and other programs and hardware.

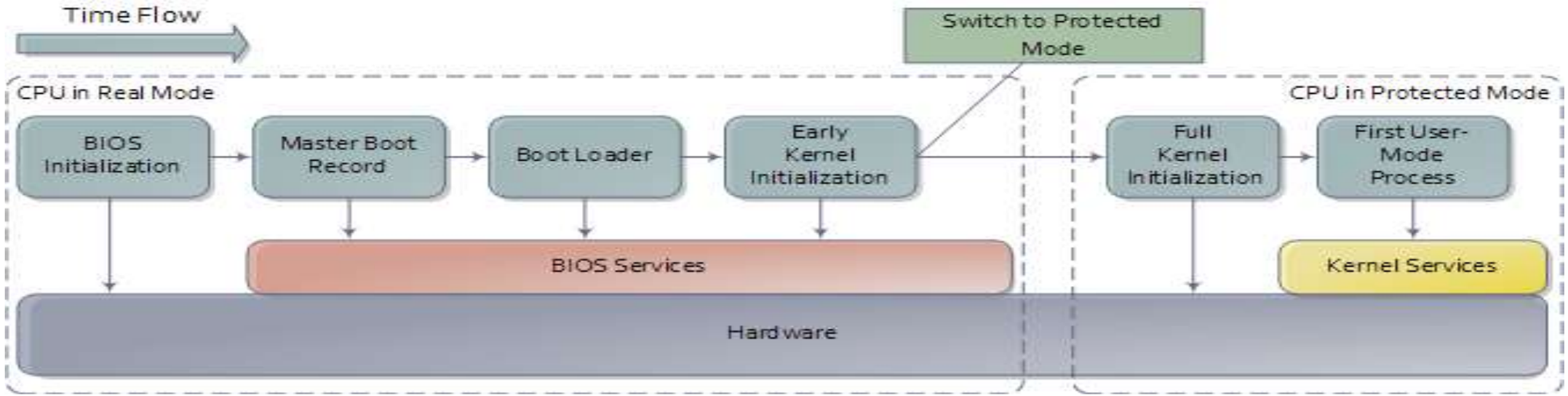
Kernel cont…

- Kernel includes:-
  1. Scheduler: It allocates the Kernel's processing time to various processes.
  2. Supervisor: It grants permission to use computer system resources to each process.
  3. Interrupt handler : It handles all requests from the various hardware devices which compete for kernel services.
  4. Memory manager : allocates space in memory for all users of kernel service.
- kernel provides services for process management, file management, I/O management, memory management.
- System calls are used to provide these type of services.

System Call

- **System call** is the programmatic way in which a computer program/user application requests a service from the kernel of the operating system on which it is executed.
- Application program is just a user-process. Due to security reasons , user applications are not given access to privileged resources(the ones controlled by OS).
- When they need to **do any I/O** or have **some more memory** or **spawn a process** or wait for **signal/interrupt,** it requests operating system to facilitate all these. This **request is made through System Call.**
- System calls are also called **software-interrupts.**

Starting an Operating System(Booting)



- ✓ Power On Switch sends electricity to the motherboard on a wire called the *Voltage Good* line.
- ✓ If the power supply is good, then the BIOS (Basic Input/Output System) chip takes over.
- ✓ In Real Mode, CPU is only capable of using approximately 1 MB of memory built into the motherboard.
- ✓ The BIOS will do a Power-On Self Test (POST) to make sure that all hardware are working.

- ✓ BIOS will then look for a small sector at the very beginning of your primary hard disk called MBR. (Master Boot Record).
- ✓ The MBR contains a list, or map, of all of the partitions on your computer‘s hard disk (or disks).
- ✓ After the MBR is found the Bootstrap Loader follows basic instructions for starting up the rest of the computer, including the operating system.
- ✓ In Early Kernel Initialization stage, a smaller core of the Kernel is activated.
- ✓ This core includes the device drivers needed to use computer‘s RAM chips.

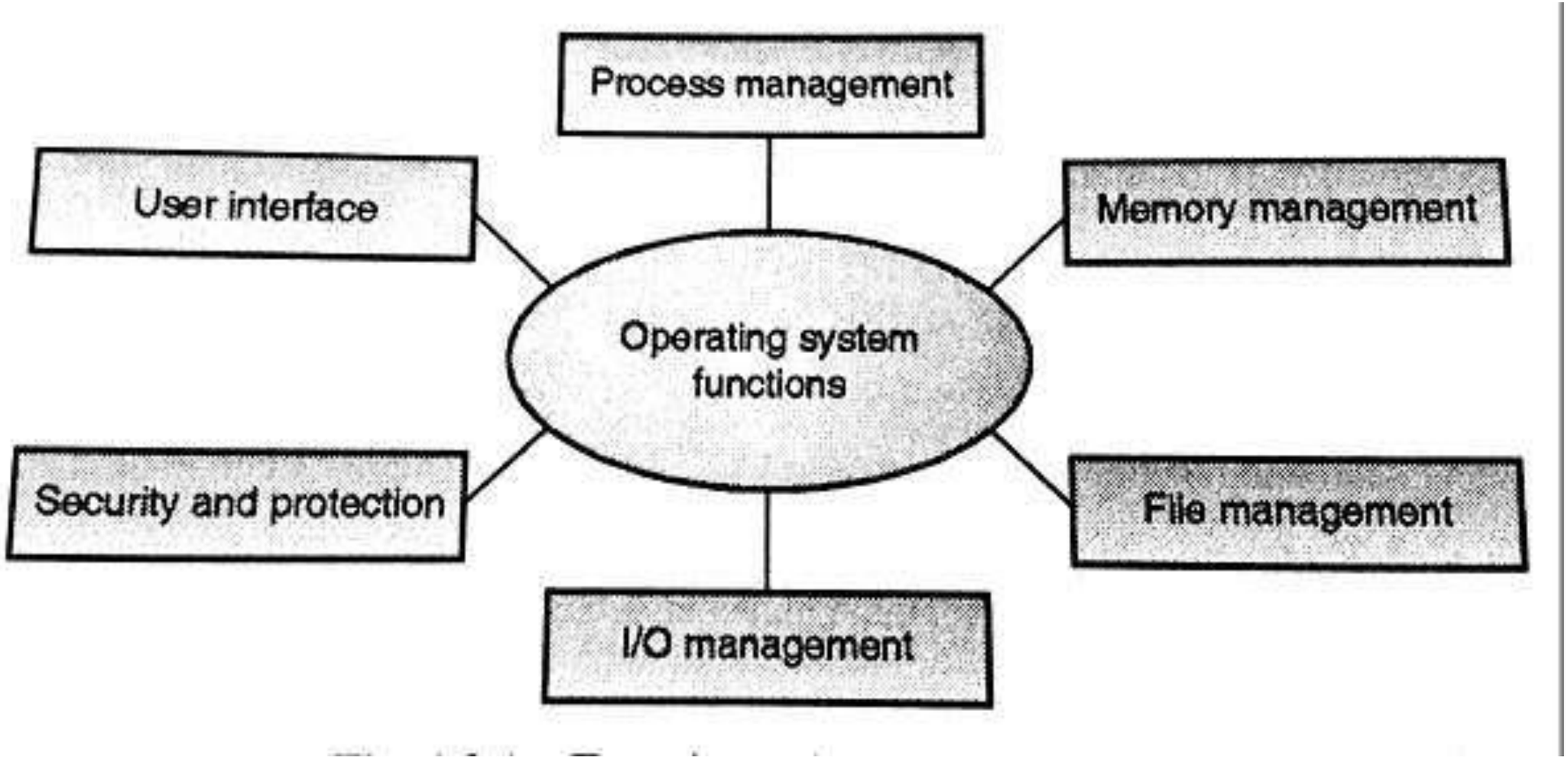
BIOS

- BIOS firmware was stored in a ROM/EPROM (Erasable Programmable Read-Only Memory) chip known as **firmware** on the PC motherboard.
- BIOS can be accessed during the initial phases of the boot procedure by pressing del, F2 or F10.
- Finally, the firmware code cycles through all storage devices and looks for a boot-loader. (usually located in first sector of a disk which is 512 bytes)
- If the boot-loader is found, then the firmware hands over control of the computer to it.

UEFI

- UEFI stands for Unified Extensible Firmware Interface. It does the same job as a BIOS, but with one basic difference: it stores all data about initialization and startup in an .efi file, instead of storing it on the firmware.
- This .efi file is stored on a special partition called EFI System Partition (ESP) on the hard disk. This ESP partition also contains the bootloader.
- UEFI was designed to overcome many limitations of the old BIOS, including:
  - UEFI supports drive sizes upto 9 zettabytes, whereas BIOS only supports 2.2 terabytes.
  - UEFI provides faster boot time.
  - UEFI has discrete driver support, while BIOS has drive support stored in its ROM, so updating BIOS firmware is a bit difficult.
  - UEFI offers security like "Secure Boot", which prevents the computer from booting from unauthorized/unsigned applications. This helps in preventing rootkits.
  - UEFI runs in 32bit or 64bit mode, whereas BIOS runs in 16bit mode. So UEFI is able to provide a GUI (navigation with mouse) as opposed to BIOS which allows navigation only using the keyboard.

Functions of Operating System



1. Process Management

- **A *process* is a program in execution.**
- A process needs certain resources, including CPU time, memory, files, and I/O devices to accomplish its task.
- Simultaneous execution leads to multiple processes. Hence creation, execution and termination of a process are the most basic functionality of an OS
- If processes are dependent, than they may try to share same resources. thus task of process synchronization comes to the picture.
- If processes are independent, than a due care needs to be taken to avoid their overlapping in memory area.
- Based on priority, it is important to allow more important processes to execute first than others.

2. Memory management

- Memory is a large array of words or bytes, each with its own address.
- It is a repository of quickly accessible data shared by the CPU and I/O devices.
- Main memory is a **volatile** storage device. When the computer made turn off everything stored in RAM will be erased automatically.
- In addition to the physical RAM installed in your computer, most modern operating systems allow your computer to use a *virtual memory system. Virtual memory allows your computer to use part of a permanent storage device (such as a hard disk) as extra memory.*
- The operating system is responsible for the following activities in connections with memory management:
  - Keep track of which parts of memory are currently being used and by whom.
  - Decide which processes to load when memory space becomes available.
  - Allocate and de-allocate memory space as needed.

3. File Management

- A file is a collection of related information defined by its creator.
- *File systems provide the conventions for the encoding, storage and management of data on a storage device such as a hard disk.*
  - FAT12 (floppy disks)
  - FAT16 (DOS and older versions of Windows)
  - FAT32 (older versions of Windows)
  - NTFS (newer versions of Windows)
  - EXT3 (Unix/Linux)
  - HFS+ (Max OS X)
- The operating system is responsible for the following activities in connections with file management:
  - ✦ File creation and deletion.
  - ✦ Directory creation and deletion.
  - ✦ Support of primitives for manipulating files and directories.
  - ✦ Mapping files onto secondary storage.
  - ✦ File backup on stable (nonvolatile) storage media.

4. Device Management or I/O Management

- *Device controllers are components on the motherboard (or on expansion cards) that act as an interface between the CPU and the actual device.*
- *Device drivers, which are the operating system software components that interact with the devices controllers.*
- A special device (inside CPU) called the Interrupt Controller handles the task of receiving interrupt requests and prioritizes them to be forwarded to the processor.
- Deadlocks can occur when two (or more) processes have control of different I/O resources that are needed by the other processes, and they are unwilling to give up control of the device.
- It performs the following activities for device management.
  - Keeps tracks of all devices connected to system.
  - Designates a program responsible for every device known as Input/output controller.
  - Decides which process gets access to a certain device and for how long.
  - Allocates devices in an effective and efficient way.
  - Deallocates devices when they are no longer required.

5. Security & Protection

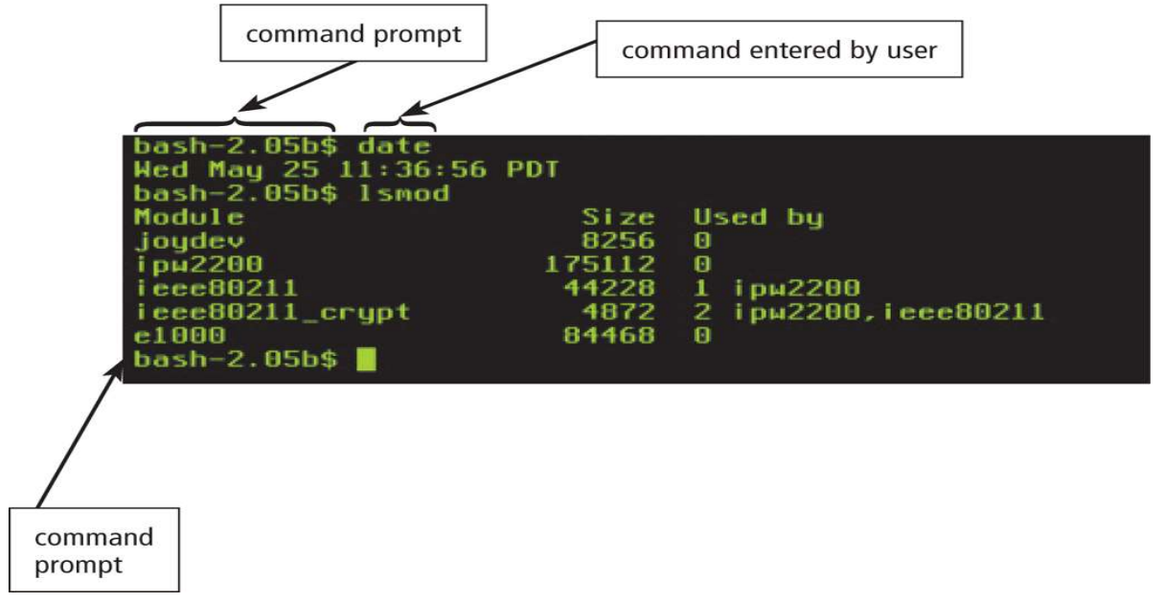
- The operating system uses password protection to protect user data and similar other techniques.
- It also prevents unauthorized access to programs and user data by assigning access right permission to files and directories.
- The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other.

6. User Interface Mechanism

- A **user interface** (**UI**) controls how you enter data and instructions and how information is displayed on the screen
- There are two types of user interfaces
  1. Command Line Interface
  2. Graphical user Interface

1. Command-line interface

- In a command-line interface, a user types commands represented by short keywords or abbreviations or presses special keys on the keyboard to enter data and instructions



2. Graphical User Interface

- With a graphical user interface (GUI), you interact with menus and visual images

History of Operating System

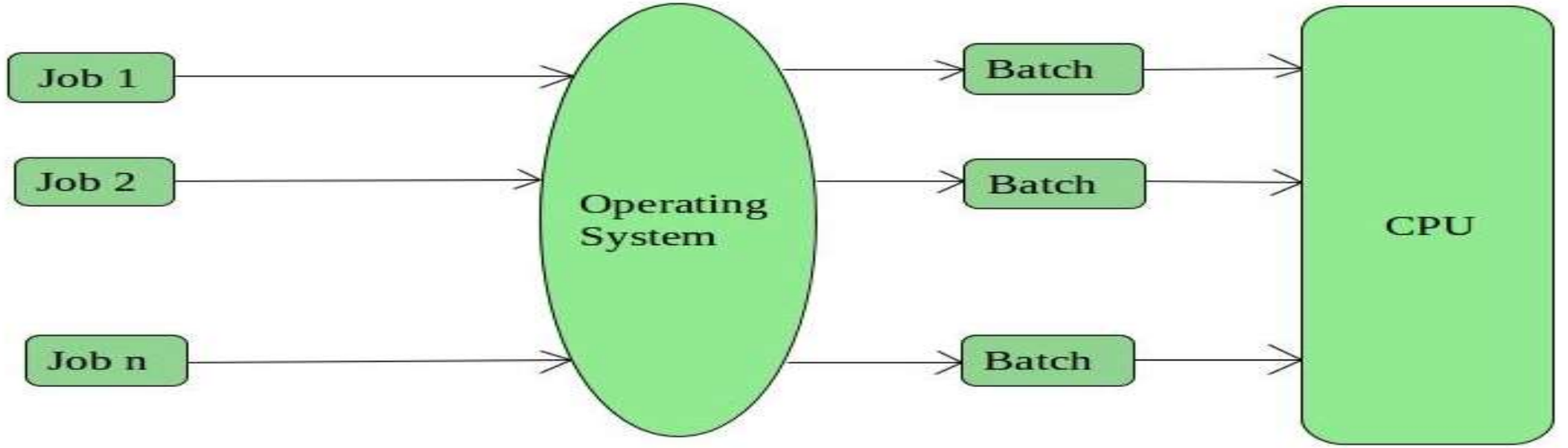
- **The First Generation (1940's to early 1950's)**
  - No Operating System
  - All programming was done in absolute machine language, often by wiring up plug-boards to control the machine’s basic functions.
- **The Second Generation (1955-1965)**
  - First operating system was introduced in the early 1950's.It was called GMOS
  - Created by General Motors for IBM's machine the 701.
  - Single-stream batch processing systems
- **The Third Generation (1965-1980)**
  - Introduction of multiprogramming
  - Development of Minicomputer
- **The Fourth Generation (1980-Present Day)**
  - Development of PCs
  - Birth of Windows/MaC OS

Types of Operating Systems

1. Batch Operating System
2. Multiprogramming Operating System
3. Time-Sharing OS
4. Multiprocessing OS
5. Distributed OS
6. Network OS
7. Real Time OS
8. Embedded OS

1. Batch Operating System

- The users of this type of operating system does not interact with the computer directly.
- Each user prepares his job on an off-line device like punch cards and submits it to the computer operator
- There is an operator which takes similar jobs having the same requirement and group them into batches



1. Batch Operating System cont..

Advantages of Batch Operating System:

- Processors of the batch systems know how long the job would be when it is in queue
- Multiple users can share the batch systems
- The idle time for the batch system is very less
- It is easy to manage large work repeatedly in batch systems

Disadvantages of Batch Operating System:

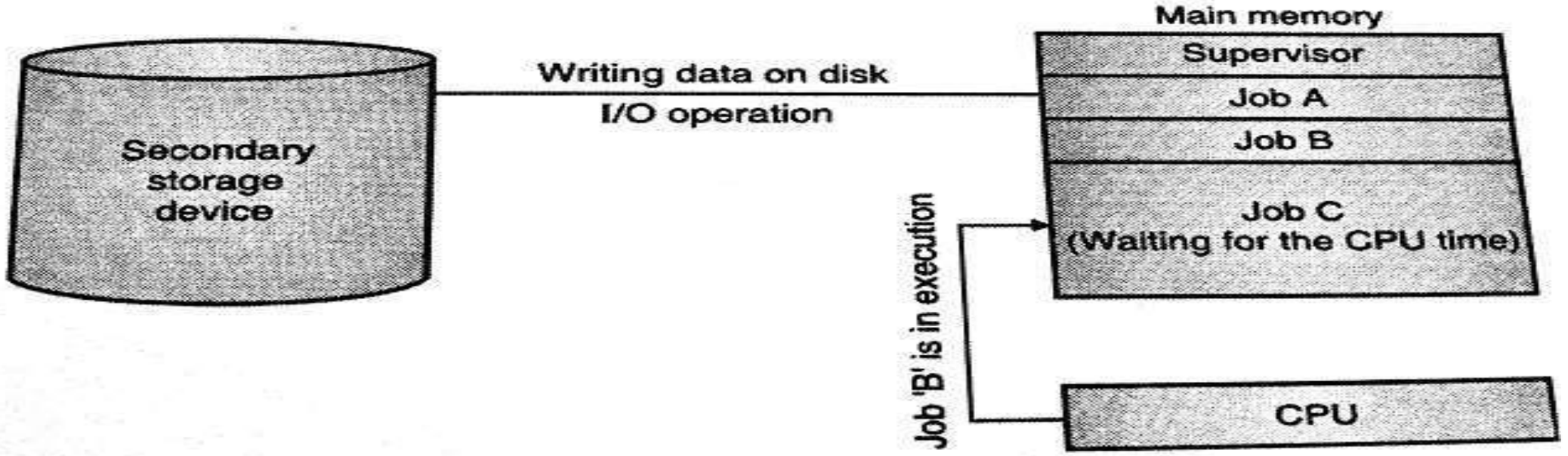
- The computer operators should be well known with batch systems
- Batch systems are hard to debug
- It is sometimes costly
- The other jobs will have to wait for an unknown time if any job fails

Examples of Batch based Operating System:

IBM's MVS

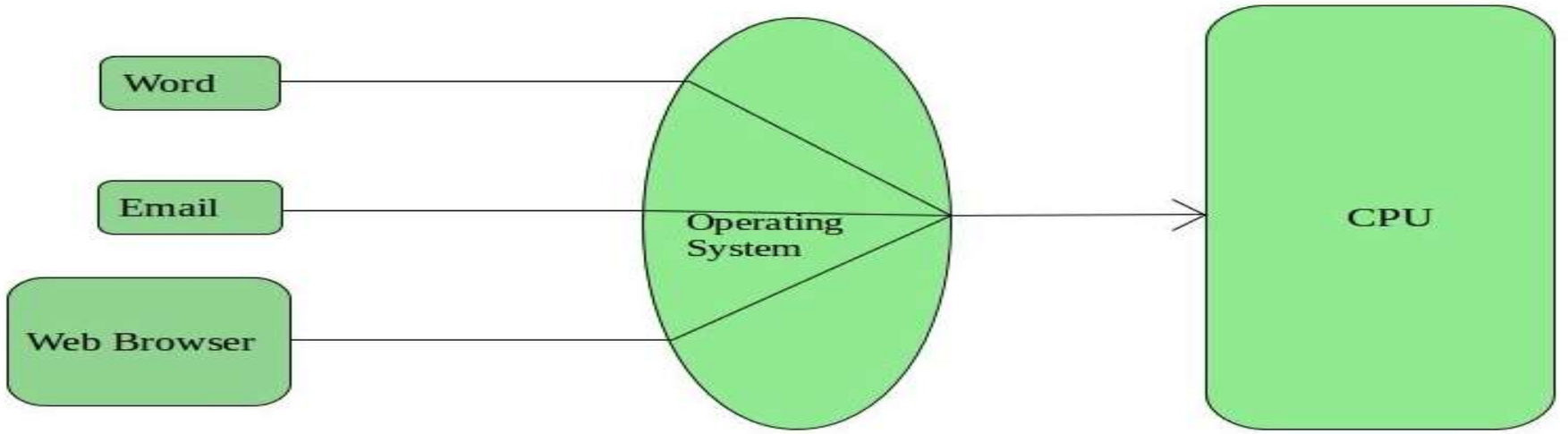
2. Multiprogramming Operating System:

- This type of OS is used to execute more than one jobs simultaneously by a single processor.
- It increases CPU utilization by organizing jobs so that the CPU always has one job to execute.
- Multiprogramming operating systems use the mechanism of job scheduling and CPU scheduling.



3. Time-Sharing Operating Systems

- Each task is given some time to execute so that all the tasks work smoothly.
- These systems are also known as **Multi-tasking Systems.**
- The task can be from a single user or different users also.
- The time that each task gets to execute is called quantum.
- After this time interval is over OS switches over to the next task.



3. Time-Sharing Operating Systems cont..

- **Advantages of Time-Sharing OS:**
  - Each task gets an equal opportunity
  - Fewer chances of duplication of software
  - CPU idle time can be reduced
- **Disadvantages of Time-Sharing OS:**
  - Reliability problem
  - One must have to take care of the security and integrity of user programs and data
  - Data communication problem
- **Examples of Time-Sharing Oss**

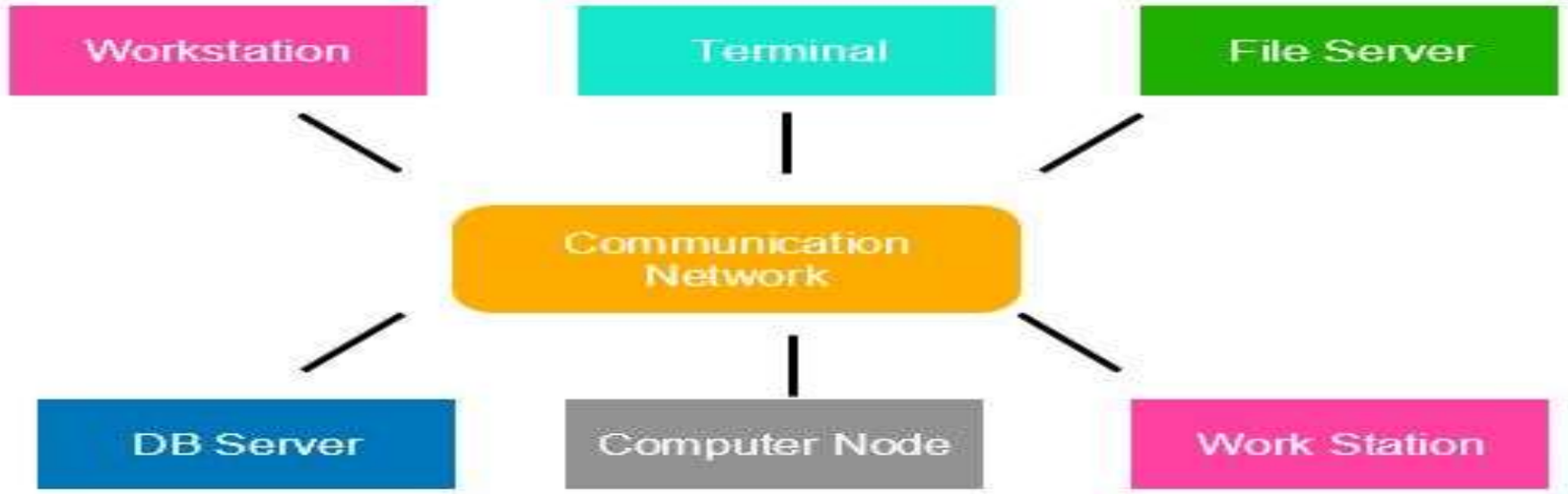
  Multics, Unix, etc.

4. Multiprocessor operating systems

- Multiprocessor operating systems are also known as parallel OS or tightly coupled OS.
- Such operating systems have more than one processor in close communication that sharing the computer bus, the clock and sometimes memory and peripheral devices.
- It executes multiple jobs at the same time and makes the processing faster.
- It supports large physical address space and larger virtual address space.
- If one processor fails then other processor should retrieve the interrupted process state so execution of process can continue.
- Inter-processes communication mechanism is provided and implemented in hardware.

5. Distributed Operating System

- Various autonomous interconnected computers communicate with each other using a shared communication network.
- Independent systems possess their own memory unit and CPU.
- These are referred to as **loosely coupled systems**.
- Examples:- Locus, DYSEAC



6. Network Operating System

- These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions.
- These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network.
- The “ other" computers arc called client computers, and each computer that connects to a network server must be running client software designed to request a specific service.
- popularly known as **tightly coupled systems**.

6. Network Operating System

Advantages of Network Operating System:

- Highly stable centralized servers
- Security concerns are handled through servers
- New technologies and hardware up-gradation are easily integrated into the system
- Server access is possible remotely from different locations and types of systems

Disadvantages of Network Operating System:

- Servers are costly
- User has to depend on a central location for most operations
- Maintenance and updates are required regularly

Examples of Network Operating System are:

Microsoft Windows Server 2003/2008/2012, UNIX, Linux, Mac OS X, Novell NetWare, and BSD, etc.

7. Real-Time Operating System

- These types of OSs serve real-time systems.
- The time interval required to process and respond to inputs is very small.
- This time interval is called **response time**.
- **Real-time systems** are used when there are time requirements that are very strict like
  - missile systems,
  - air traffic control systems,
  - robots, etc.

8. Embaded Operating System

- An embedded operating system is one that is built into the circuitry of an electronic device.
- Embedded operating systems are now found in automobiles, bar-code scanners, cell phones, medical equipment, and personal digital assistants.
- The most popular embedded operating systems for consumer products, such as PDAs, include the following:
  - Windows XP Embedded
  - Windows CE .NET:- it supports wireless communications, multimedia and Web browsing. It also allows for the use of smaller versions of Microsoft Word, Excel, and Outlook.
  - Palm OS:- It is the standard operating system for Palm-brand PDAs as well as other proprietary handheld devices.
  - Symbian:- OS found in “ smart” cell phones from Nokia and Sony Ericsson

Popular types of OS

- Desktop Class
  - ❖ Windows
  - ❖ OS X
  - ❖ Unix/Linux
  - ❖ Chrome OS
- Server Class
  - ❖ Windows Server
  - ❖ Mac OS X Server
  - ❖ Unix/Linux
- Mobile Class
  - ❖ Android
  - ❖ iOS
  - ❖ Windows Phone

Desktop Class Operating Systems:-

- **Platform:** the hardware required to run a particular operating system
  - Intel platform (IBM-compatible)
    - Windows
    - DOS
    - UNIX
    - Linux
  - Macintosh platform
    - Mac OS
  - iPad and iPhone platform

Ms-DOS

- Single User Single Tasking OS.
- It had no built-in support for networking, and users had to manually install drivers any time they added a new hardware component to their PC.
- DOS supports only 16-bit programs.
- Command line user interface.
- So, why is DOS still in use? Two reasons are its size and simplicity. It does not require much memory or storage space for the system, and it docs not require a powerful computer.

Microsoft Windows

- The graphical Microsoft operating system designed for Intel-platform desktop and notebook computers.
- Best known, greatest selection of applications available.
- Current editions include Windows 7, 8, 8.1 and 10.

Mac OS

- User-friendly, runs on Mac hardware. Many applications available.
- Current editions include: Sierra, High Sierra, Mojave, Catalina & Big Sur—Version XI(Released in Nov 2020)

Linux

- **Linux:** An open-source, cross-platform operating system that runs on desktops, notebooks, tablets, and smartphones.
  - The name *Linux* is a combination *Linus* (the first name of the first developer) and *UNIX (another operating system*.
- Users are free to modify the code, improve it, and redistribute it,
- Developers are not allowed to charge money for the Linux kernel itself (the main part of the operating system), but they can charge money for **distributions** (**distros** for short).

Google Chrome OS

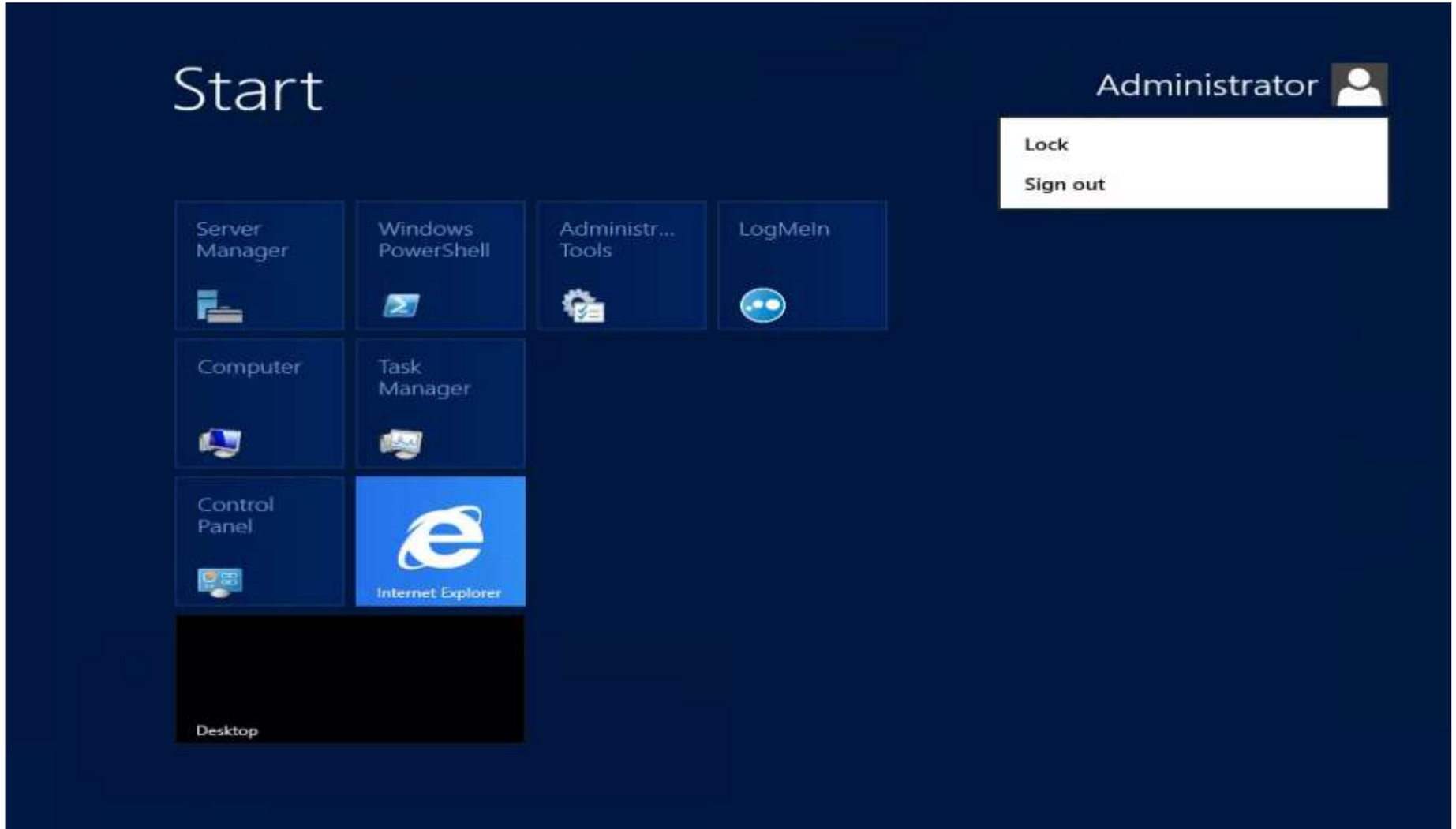
- **Chrome OS**. Is a popular thin client operating system.
- **Thin client** A computer with minimal hardware, designed for a specific task. For example, a thin web client is designed for using the Internet.

chromebook

Server Operating Systems

- Windows Server
  - Familiar GUI interface for those experienced with Windows
- UNIX
  - Very mature server capabilities, time-tested, large user community, stable
- Linux
  - Free, customizable, many free services and utilities available

Windows Server



UNIX

```
mars@marsmain /usr/portage/app-shells/bash $ sudo /etc/init.d/bluetooth status
Password:
* status: started
mars@marsmain /usr/portage/app-shells/bash $ ping -q -c1 en.wikipedia.org
PING rr.esams.wikimedia.org (91.198.174.2) 56(84) bytes of data.

--- rr.esams.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 49.820/49.820/49.820/0.000 ms
mars@marsmain /usr/portage/app-shells/bash $ grep -i /dev/sda /etc/fstab | cut --fields=-3
/dev/sda1               /boot
/dev/sda2               none
/dev/sda3               /
mars@marsmain /usr/portage/app-shells/bash $ date
Sat Aug  8 02:42:24 MSD 2009
mars@marsmain /usr/portage/app-shells/bash $ lsmod
Module                  Size  Used by
rndis_wlan             23424  0
rndis_host              8696  1 rndis_wlan
cdc_ether               5672  1 rndis_host
usbnet                 18688  3 rndis_wlan,rndis_host,cdc_ether
parport_pc             38424  0
fglrx                2388128  20
parport                39648  1 parport_pc
iTCO_wdt               12272  0
i2c_i801                9380  0
mars@marsmain /usr/portage/app-shells/bash $
```

Tablet and Phone Operating Systems

- **System-on-chip (SoC):** An operating system that comes preinstalled on a chip on a portable device such as a smartphone.
- Popular SoC operating systems:
  - iOS: for iPad, iPhone
  - Android: for a variety of tablets and phones
- Downloadable applications (apps) from an App store, for example:
  - Apple App Store
  - Google Play Store

iOS on the iPhone and iPad

- The Apple-created operating system for Apple tablets and phones.
- The current stable version, iOS 14, was released to the public on September 16, 2020.

Android

- Android, a popular OS for smartphones and tablets, is based on Linux Kernel.
  - Developed by Google
- Current versions include:
  - Android 8 Oreo
  - Android 9 Pie
  - Android 10
  - Android 11 (released on Sep, 2020)

Advantage of Linux Operating System

1. Open Source

As it is open-source, its source code is easily available.

Anyone having programming knowledge can customize the operating system.

One can contribute, modify, distribute, and enhance the code for any purpose.

2. Security

The Linux security feature is the main reason that it is the most favourable option for developers.

It is not completely safe, but it is less vulnerable than others.

Each application needs to authorize by the admin user.

Linux systems do not require any antivirus program.

3. Free

Certainly, the biggest advantage of the Linux system is that it is free to use.

We can easily download it, and there is no need to buy the license for it.

It is distributed under GPL (General Public License).

Comparatively, we have to pay a huge amount for the license of the other OS

Advantage of Linux Operating System

4. Lightweight

The requirements for running Linux are much less than other operating system

In Linux, the memory footprint and disk space are also lower.

Generally, most of the Linux distributions required as little as 128MB of RAM around the same amount for disk space.

5. Stability

Linux is more stable than other operating systems.

Linux does not require to reboot the system to maintain performance levels.

It rarely hangs up or slow down. It has big up-times.

Advantage of Linux Operating System

6. Performance

Linux system provides high performance over different networks.

It is capable of handling a large number of users simultaneously.

7. Flexibility

Linux operating system is very flexible.

It can be used for desktop applications, embedded systems, and server applications too.

It also provides various restriction options for specific computers.

We can install only necessary components for a system.

8. Software Updates

In Linux, the software updates are in user control.

We can select the required updates.

There a large number of system updates are available.

These updates are much faster than other operating systems.

So, the system updates can be installed easily without facing any issue.

Advantage of Linux Operating System

9. Distributions/ Distros

There are many Linux distributions available in the market.

It provides various options and flavors of Linux to the users.

We can choose any distros according to our needs.

Some popular distros are **Ubuntu, Fedora, Debian, Linux Mint, Arch Linux,**

For the beginners, Ubuntu and Linux Mint would be useful.

Debian and Fedora would be good choices for proficient programmers.

10. Live CD/USB

Almost all Linux distributions have **a Live CD/USB** option.

It allows us to try or run the Linux operating system without installing it.

11. Graphical User Interface

Linux is a command-line based OS but it provides an interactive user interface like Windows.

Advantage of Linux Operating System

12. Suitable for programmers

It supports almost all of the most used programming languages such as C/C++, Java, Python, Ruby, and more.

Further, it offers a vast range of useful applications for development.

The programmers prefer the Linux terminal over the Windows command line.

The package manager on Linux system helps programmers to understand how things are done.

Bash scripting is also a functional feature for the programmers.

It also provides support for SSH, which helps in managing the servers quickly.

13. Community Support

Linux provides large community support.

We can find support from various sources.

There are many forums available on the web to assist users.

Further, developers from the various open source communities are ready to help us.

Advantage of Linux Operating System

14. Privacy

Linux always takes care of user privacy as it never takes much private data from the user. Comparatively, other operating systems ask for the user's private data.

15. Networking

Linux facilitates with powerful support for networking. The client-server systems can be easily set to a Linux system. It provides various command-line tools such as ssh, ip, mail, telnet, and more for connectivity with the other systems and servers. Tasks such as network backup are much faster than others.

16. Compatibility

Linux is compatible with a large number of file formats as it supports almost all file formats.

17. Installation

Linux installation process takes less time than other operating systems such as Windows. Further, its installation process is much easy as it requires less user input. It does not require much more system configuration even it can be easily installed on old machines having less configuration.

Advantage of Linux Operating System

18. Multiple Desktop Support

Linux system provides multiple desktop environment support for its enhanced use. The desktop environment option can be selected during installation. We can select any desktop environment such as **GNOME (GNU Network Object Model Environment)** or **KDE (K Desktop Environment)** as both have their specific environment.

19. Multitasking

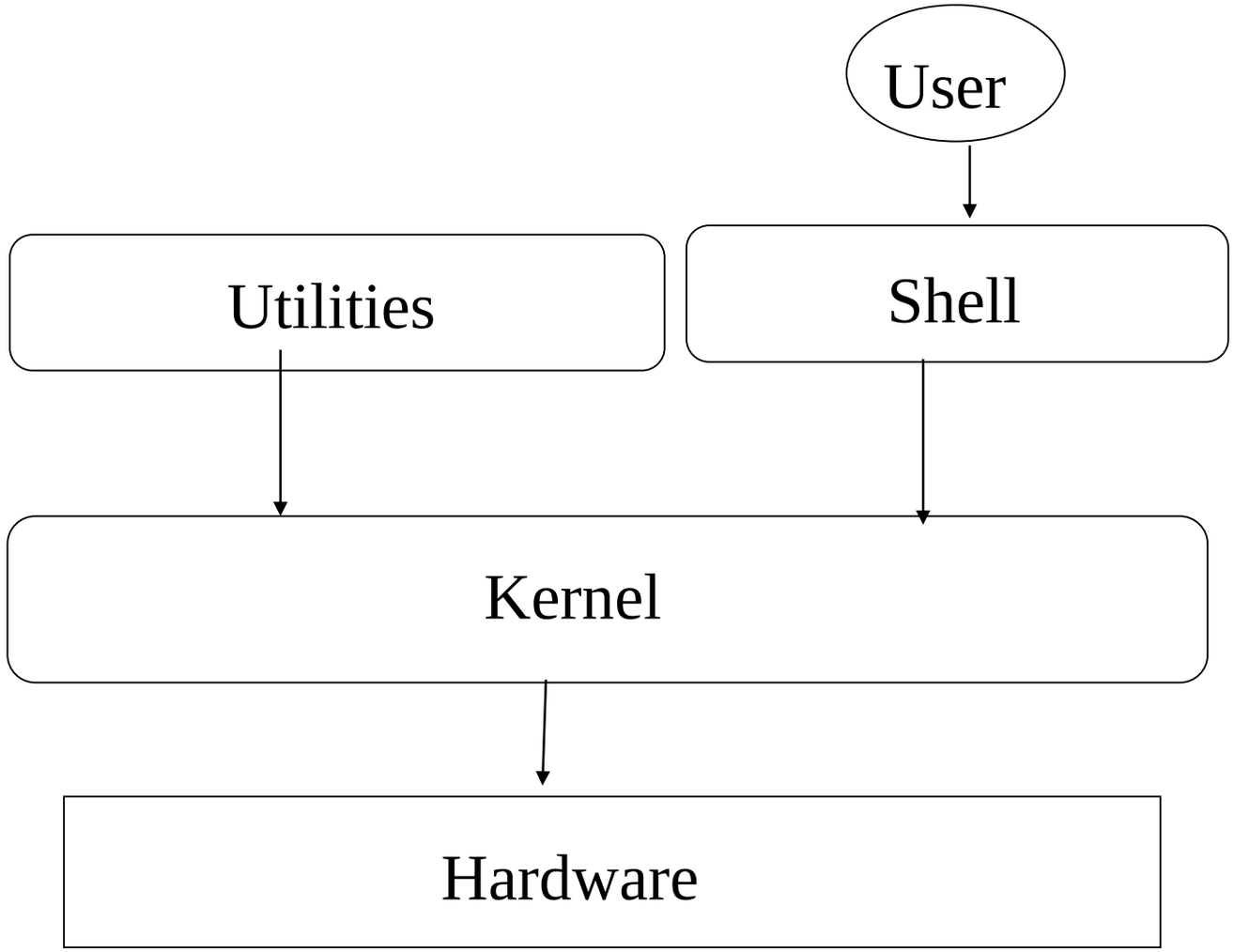
It is a multitasking operating system as it can run multiple tasks simultaneously without affecting the system speed.

20. Heavily Documented for beginners

There are many command-line options that provide documentation on commands, libraries, standards such as manual pages and info pages. Also, there are plenty of documents available on the internet in different formats, such as Linux tutorials, Linux documentation project, Serverfault, and more. To help the beginners, several communities are available such as **Ask Ubuntu**, Reddit, and **StackOverflow.**

UNIX Shell and Utilities

- The shell used to be in the kernel but now is a utility outside of it.
- Easy to change/debug.
- Many of them (sh, bsh, csh, ksh, tcsh, wsh, bash)
- Possible to switch between them (chsh)



A very simplified Shell

```
#define TRUE 1

while (TRUE) {                                  /* repeat forever */
        type_prompt( );                         /* display prompt on the screen */
        read_command(command, parameters);      /* read input from terminal */

        if (fork( ) != 0) {                     /* fork off child process */
             /* Parent code. */
             waitpid(–1, &status, 0);           /* wait for child to exit */
        } else {
             /* Child code. */
             execve(command, parameters, 0);    /* execute command */
        }
}
```


Thank you!