UNIT-1

INTRODUCTION TO OPERATING SYSTEM

What is an Operating System?

An operating system is system software that acts as an intermediary between a user of a computer and the computer hardware. It is software that manages the computer hardware and allows the user to execute programs in a convenient and efficient manner.

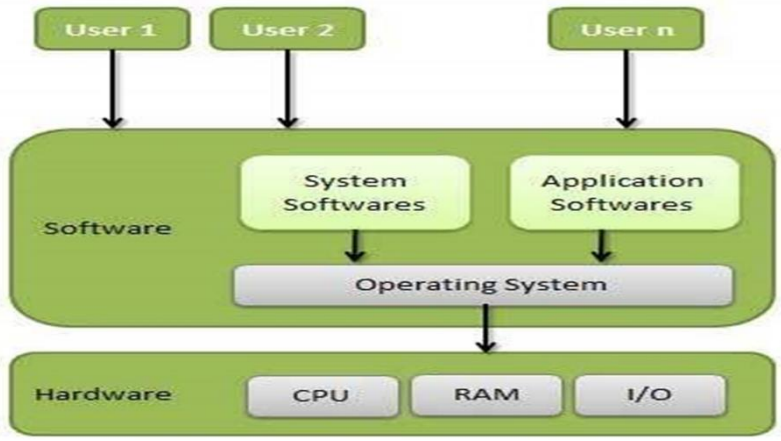
Operating system goals:

• Make the computer system convenient to use. It hides the difficulty in managing the hardware.

• Use the computer hardware in an efficient manner

• Provide and environment in which user can easily interface with computer.

• It is a resource allocator



Characteristics of Operating System

- **Convenience:** An OS makes a computer more convenient to use.
- **Efficiency:** An OS allows the computer system resources to be used efficiently.

- **Ability to Evolve:** An OS should be constructed in such a way as to permit the effective development, testing, and introduction of new system functions at the same time without interfering with service.
- **Throughput:** An OS should be constructed so that It can give maximum **throughput** (Number of tasks per unit time).

Computer System Structure (Components of Computer System)

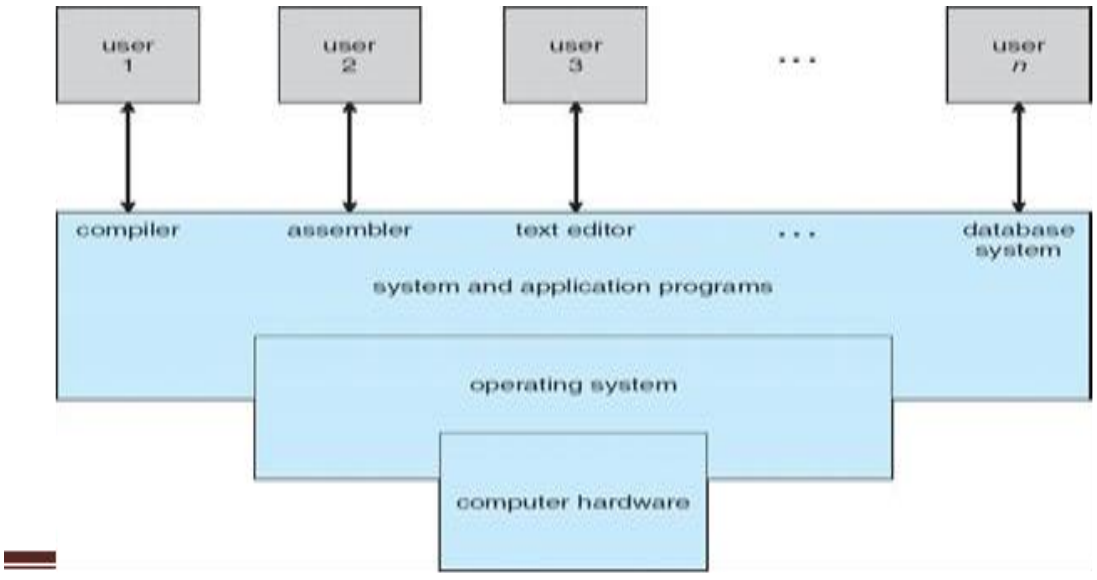
Computer system mainly consists of four components-

• **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 computer



The basic hardware components comprises of CPU, memory, I/O devices. The

application program uses these components. The OS controls and co-ordinates the use of hardware, among various application programs (like compiler, word processor etc.) for various users.

The OS allocates the resources among the programs such that the hardware is efficiently used.

The operating system is the program running at all the times on the computer. It is usually called as the kernel.

Kernel functions are used always in system, so always stored in memory. Non kernel functions are stored in hard disk, and it is retrieved whenever required.

Views of OS

Operating System can be viewed from two viewpoints–

User views & System views

1. User Views:-

The user's view of the operating system depends on the type of user.

i. If the user is using standalone system, then OS is designed for ease of use and high performances. Here resource utilization is not given importance.

ii. If the users are at different terminals connected to a mainframe or minicomputers, by sharing information and resources, then the OS is designed to maximize resource utilization. OS is designed such that the CPU time, memory and i/o are used efficiently and no single user takes more than the resource allotted to them.

iii. If the users are in workstations, connected to networks and servers, then the user have a system unit of their own and shares resources and files with other systems. Here the OS is designed for both ease of use and resource availability (files).

iv. Users of hand held systems, expects the OS to be designed for ease of use and performance per amount of battery life.

v. Other systems like embedded systems used in home devies (like washing m/c) & automobiles do not have any user interaction. There are some LEDs to show the status of its work.

2. System Views:-

Operating system can be viewed as a resource allocator and control program.

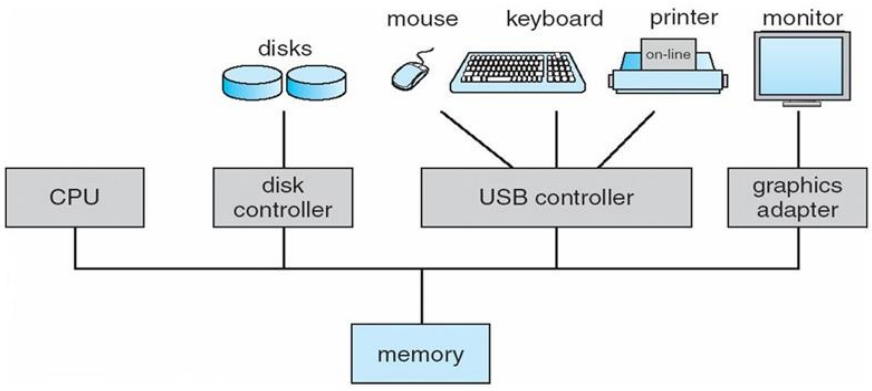
i. Resource allocator - The OS acts as a manager of hardware and software resources. CPU time, memory space, file-storage space, I/O devices, shared files etc. are the different resources required during execution of a program. There can be conflicting request for these resources by different programs running in same system. The OS assigns the resources to the requesting program depending on the priority.

ii. Control Program – The OS is a control program and manage the execution of user program to prevent errors and improper use of the computer.

Computer System Organization

Computer-system operation

One or more CPUs, device controllers connect through common bus providing access to shared memory. Each device controller is in-charge of a specific type of device. To ensure orderly access to the shared memory, a memory controller is provided whose function is to synchronize access to the memory. The CPU and other devices execute concurrently competing for memory cycles. Concurrent execution of CPUs and devices competing for memory cycles



When system is switched on, ‘Bootstrap’ program is executed. It is the initial program to run in the system. This program is stored in read-only memory (ROM) or in electrically erasable programmable read-only memory(EEPROM). It initializes the CPU registers, memory, device controllers and other initial setups. The program also locates and loads, the OS kernel to the memory. Then the OS starts with the first process to be executed (ie. ‘init’ process) and then wait for the interrupt from the user.

Switch on --------→ ‘Bootstrap’ program

- Initializes the registers, memory and I/O devices
- Locates & loads kernel into memory
- Starts with ‘init’ process
- Waits for interrupt from user

I/O Structure

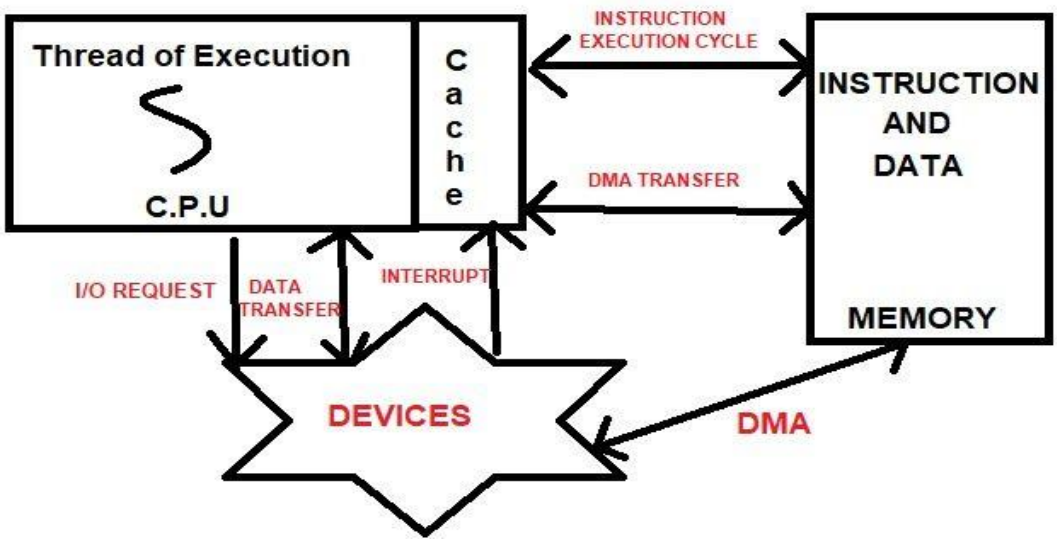
• A large portion of operating system code is dedicated to managing I/O, both because of its importance to the reliability and performance of a system and because of the varying nature of the devices.

• Every device has a device controller, maintains some local buffer and a set of special- purpose registers. The device controller is responsible for moving the data between the peripheral devices. The operating systems have a device driver for each device controller.

• Interrupt-driven I/O is well suited for moving small amounts of data but can produce high

overhead when used for bulk data movement such as disk I/O. To solve this problem, direct memory access (DMA) is used.

• After setting up buffers, pointers, and counters for the I/O device, the device controller transfers an entire block of data directly to or from its own buffer storage to memory, with no intervention by the CPU. Only one interrupt is generated per block, to tell the device driver that the operation has completed.



Storage Structure

The CPU can load instructions only from memory, so any programs to run must be stored there. General-purpose computers run most of their programs from rewriteable memory, called main memory (also called or RAM). Main commonly is implemented in a semiconductor technology called DRAM.

All forms of memory provide an array of words. Each word has its own address. Interaction is achieved through a sequence of load or store instructions to specific memory addresses. The load instruction moves a word from main memory to an internal register within the CPU, whereas the store instruction moves the content of a register to main memory.

Ideally, we want the programs and data to reside in main memory permanently. This arrangement usually is not possible for the following two reasons:

1) Main memory is usually too small to store all needed programs and data permanently.

2) Main memory is a volatile storage device that loses its contents when power is turned off or otherwise lost.

Thus, most computer systems provide secondary storage as an extension of main memory. The main requirement for secondary storage is that it be able to hold large quantities of data permanently. The most common secondary-storage device is a magnetic disk which provides storage for both programs and data.

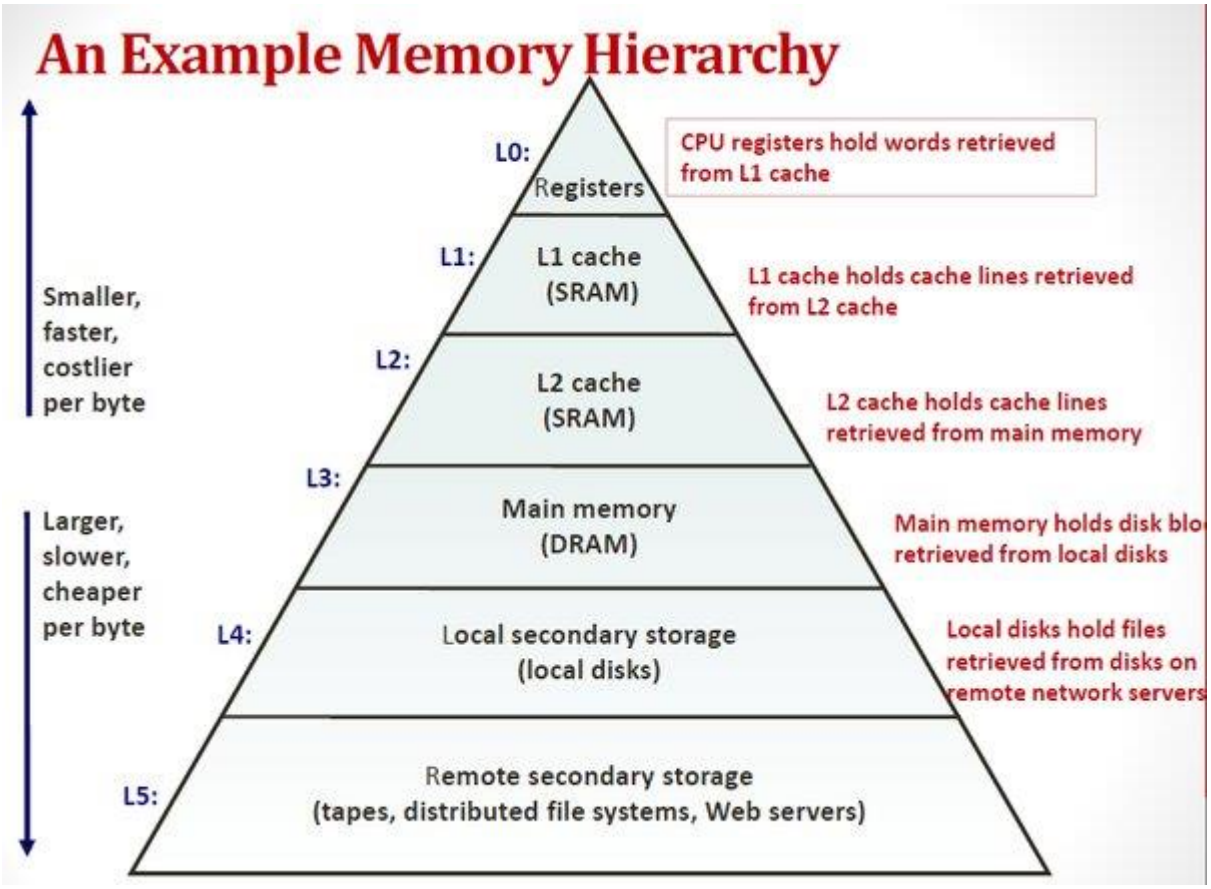
• **Main memory** – only large storage media that the CPU can access directly

• **Secondary storage** – extension of main memory that provides large nonvolatile storage capacity

• **Magnetic disks** – rigid metal or glass platters covered with magnetic recording material

Storage Hierarchy

- Storage systems organized in hierarchy
- Speed
- Cost
- Volatility



Caching – copying information into faster storage system; main memory can be viewed as a last cache for secondary storage

Computer-System Architecture

• Most systems use a single general-purpose processor (PDAs through mainframes)

• Most systems have special-purpose processors as well

• Multiprocessors systems growing in use and importance

• Also known as parallel systems, tightly-coupled systems

Advantages include

1.Increased throughput

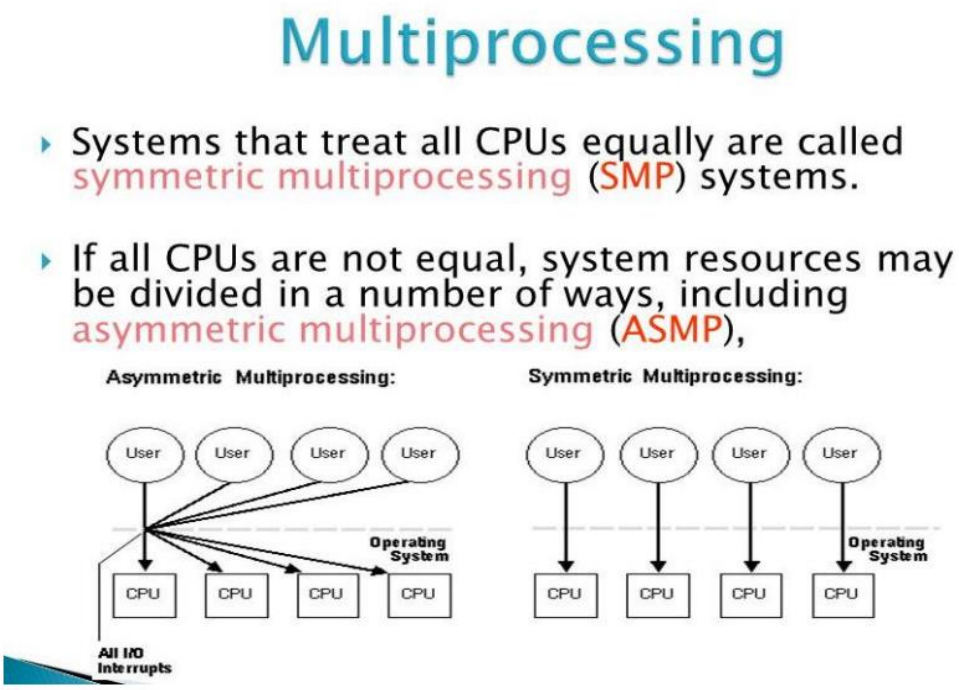
2.Economy of scale

3.Increased reliability – graceful degradation or fault tolerance

Two types

1.Asymmetric Multiprocessing

2.Symmetric Multiprocessing



Evolution of Operating Systems

Operating systems have progressed from slow and expensive systems to today's technology, which has exponentially increased computing power at comparatively modest costs. So let's have a detailed look at the evolution of operating systems.

The operating system can be classified into four generations, as follows:

- First Generation
- Second Generation
- Third Generation
- Fourth Generation

First Generation (1945-1955)

Serial Processing

The evolution of operating systems began with serial processing. It marks the start of the development of electronic computing systems as alternatives to mechanical computers. Because of the flaws in mechanical computing devices, humans' calculation speed is limited, and they are prone to making mistakes. Because there is no operating system in this generation, the computer system is given instructions that must be carried out immediately.

Programmers were incorporated into hardware components without using an operating system by the 1940s and 1950s. The challenges here are scheduling and setup time. The user logs in for machine time by wasting computational time. Setup time is required when loading the compiler, saving the compiled program, the source program, linking, and buffering. The process is restarted if an intermediate error occurs.

Example: Windows 95 and 98 are examples of serial processing operating systems.

Second Generation (1955-1965)

Batch System

The batched systems marked the second generation in the evolution of operating systems. In the second generation, the batch processing system was implemented, which allows a job or task to be done in a series and then completed sequentially. The computer system in this generation does not have an operating system, although various operating system functionalities are available, such as FMS and IBSYS. It is used to improve computer utilization and application. On cards and tapes, jobs were scheduled and submitted. Then, using Job Control Language, they were successively executed on the monitors. The first computers employed in the batch operation method created a computer batch of jobs that never paused or stopped. The software is written on punch cards and then transferred to the tape's processing unit. When the computer finishes one job, it immediately moves on to the next item on the tape. Despite the fact that it is inconvenient for the users, it is designed to keep the pricey computer as busy as possible by running a leveraged stream of operations. Memory protection prevents the memory space that makes up the monitor from being changed, and the timer prevents the job from monopolizing the system. When the input and output devices are in use, the processor remains idle due to poor CPU utilization.

Example: MVS Operating System of IBM is an example of a batch processing operating system.

Third Generation (1965-1980)

Multi-Programmed Batched System

The evolution of operating systems embarks the third generation with multi-programmed batched systems. In the third generation, the operating system was designed to serve numerous users simultaneously. Interactive users can communicate with a computer via an online terminal, making the operating system multi-user and multiprogramming. It is used to execute several jobs that should be kept in the main memory. The processor determines which program to run through job scheduling algorithms.

Example: Windows and IOS are examples of multi-programmed batched operating systems.

Fourth Generation (1980-Now)

The operating system is employed in this age for computer networks where users are aware of the existence of computers connected to one another.

The era of networked computing has already begun, and users are comforted by a Graphical User Interface (GUI), which is an incredibly comfortable graphical computer interface. In the fourth generation, the time-sharing operating system and the Macintosh operating system came into existence.

Time-Sharing Operating System

The Time-sharing of operating systems had a great impact on the evolution of operating systems. Multiple users can access the system via terminals at the same time, and the processor's time is divided among them. Printing ports were required for programs having a command-line user interface, which required written responses to prompts or written commands. The interaction is scrolled down like a roll of paper. It was previously used to develop batch replacement systems. The user interfaces directly with the computer via printing ports, much like an electric teletype. Few users shared the computer immediately, and each activity was completed in a fraction of a second before moving on to the next. By establishing iterations when they are receiving full attention, the fast server may act on a large number of users' processes at once. Multiple programs use time-sharing systems to apply to the computer system by sharing the system interactively.

Example: Unix Operating System is an example of a time-sharing OS.

Functions of Operating System

1. Memory Management

It is the management of the main or primary memory. Whatever program is executed, it has to be present in the main memory. Main memory is a quick storage area that may be accessed directly by the CPU. When the program is completed, the memory region is released and can be used by other programs.

Therefore, there can be more than one program present at a time. Hence, it is required to manage the memory.

The operating system:

- Allocates and deallocates the memory.
- Keeps a record of which part of primary memory is used by whom and how much.
- Distributes the memory while multiprocessing.
- In multiprogramming, the operating system selects which processes acquire memory when and how much memory they get.

2. Processor Management/Scheduling

Every software that runs on a computer, whether in the background or in the frontend, is a process. Processor management is an execution unit in which a program operates. The operating system determines the status of the processor and processes, selects a job and its processor, allocates the processor to the process, and de-allocates the processor after the process is completed.

When more than one process runs on the system the OS decides how and when a process will use the CPU. Hence, the name is also CPU Scheduling. The OS:

- Allocates and deallocates processor to the processes.
- Keeps record of CPU status.

Certain algorithms used for CPU scheduling are as follows:

- First Come First Serve (FCFS)
- Shortest Job First (SJF)
- Round-Robin Scheduling
- Priority-based scheduling etc.

Purpose of CPU scheduling

The purpose of CPU scheduling is as follows:

- Proper utilization of CPU. Since the proper utilization of the CPU is necessary. Therefore, the OS makes sure that the CPU should be as busy as possible.
- Since every device should get a chance to use the processor. Hence, the OS makes sure that the devices get fair processor time.
- Increasing the efficiency of the system.

3. Device Management

An operating system regulates device connection using drivers. The processes may require devices for their use. This management is done by the OS.

The OS:

- Allocates and deallocates devices to different processes.
- Keeps records of the devices.
- Decides which process can use which device for how much time.

4. File Management

The operating system manages resource allocation and de-allocation. It specifies which process receives the file and for how long. It also keeps track of information, location, uses, status, and so on. These groupings of resources are referred to as file systems. The files on a system are stored in different directories. The OS:

- Keeps records of the status and locations of files.
- Allocates and deallocates resources.
- Decides who gets the resources.

5. Storage Management

Storage management is a procedure that allows users to maximize the utilization of storage devices while also protecting data integrity on whatever media on which it lives. Network virtualization, replication, mirroring, security, compression, deduplication, traffic analysis, process automation, storage provisioning, and memory management are some of the features that may be included. The operating system is in charge of storing and accessing files. The creation of files, the creation of directories, the reading and writing of data from files and directories, as well as the copying of the contents of files and directories from one location to another are all included in storage management.

The OS uses storage management for:

- Improving the performance of the data storage resources.
- It optimizes the use of various storage devices.
- Assists businesses in storing more data on existing hardware, speeding up the data retrieval process, preventing data loss, meeting data retention regulations, and lowering IT costs.

Security – For security, modern operating systems employ a firewall. A firewall is a type of security system that monitors all computer activity and blocks it if it detects a threat.

Job Accounting – As the operating system keeps track of all the functions of a computer system. Hence, it makes a record of all the activities taking place on the system. It has an account of all the information about the memory, resources, errors, etc. Therefore, this information can be used as and when required.

Control over system performance – The operating system will collect consumption statistics for various resources and monitor performance indicators such as reaction time, which is the time between requesting a service and receiving a response from the system.

Error detecting aids – While a computer system is running, a variety of errors might occur. Error detection guarantees that data is delivered reliably across susceptible networks. The operating system continuously monitors the system to locate or recognize problems and protects the system from them.

Coordination between other software and users – The operating system (OS) allows hardware components to be coordinated and directs and allocates assemblers, interpreters, compilers, and other software to different users of the computer system.

Booting process – The process of starting or restarting a computer is referred to as Booting. Cold booting occurs when a computer is totally turned off and then turned back on. Warm booting occurs when the computer is restarted. The operating system (OS) is in charge of booting the computer.

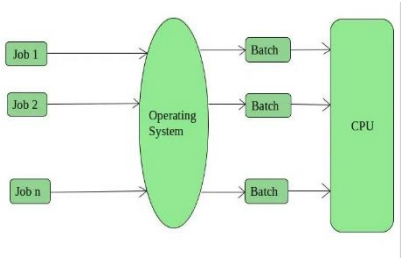
Types of Operating System

- Batch Operating System
- Multi-Programming System
- Multi-Processing System
- Multi-Tasking Operating System

- Time-Sharing Operating System
- Distributed Operating System
- Network Operating System
- Real-Time Operating System

1. Batch Operating System

This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirements and groups them into batches. It is the responsibility of the operator to sort jobs with similar needs.



Batch Operating System

Advantages of Batch Operating System

- It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in the 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.

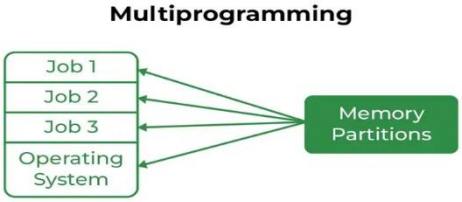
Disadvantages of Batch Operating System

- 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 Operating Systems: Payroll Systems, Bank Statements, etc.

2. Multi-Programming Operating System

Multiprogramming Operating Systems can be simply illustrated as more than one program is present in the main memory and any one of them can be kept in execution. This is basically used for better execution of resources.



MultiProgramming

Advantages of Multi-Programming Operating System

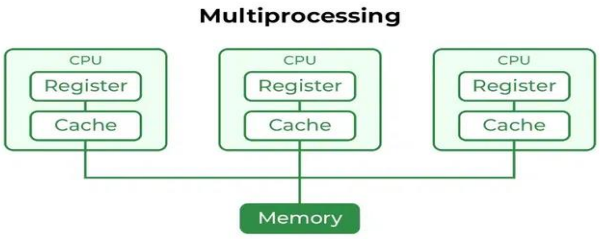
- Multi Programming increases the Throughput of the System.
- It helps in reducing the response time.

Disadvantages of Multi-Programming Operating System

- There is not any facility for user interaction of system resources with the system.

3. Multi-Processing Operating System

Multi-Processing Operating System is a type of Operating System in which more than one CPU is used for the execution of resources. It betters the throughput of the System.



Multiprocessing

Advantages of Multi-Processing Operating System

- It increases the throughput of the system.
- As it has several processors, so, if one processor fails, we can proceed with another processor.

Disadvantages of Multi-Processing Operating System

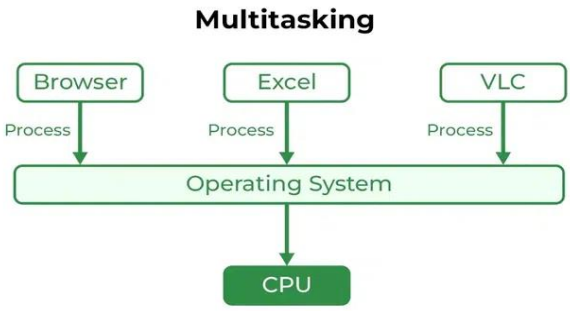
- Due to the multiple CPU, it can be more complex and somehow difficult to understand.

4. Multi-Tasking Operating System

Multitasking Operating System is simply a multiprogramming Operating System with having facility of a Round-Robin Scheduling Algorithm. It can run multiple programs simultaneously.

There are two types of Multi-Tasking Systems which are listed below.

- Preemptive Multi-Tasking
- Cooperative Multi-Tasking



Multitasking

Advantages of Multi-Tasking Operating System

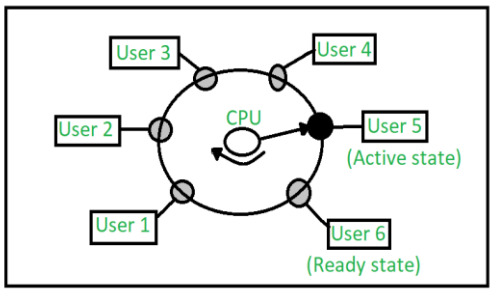
- Multiple Programs can be executed simultaneously in Multi-Tasking Operating System.
- It comes with proper memory management.

Disadvantages of Multi-Tasking Operating System

- The system gets heated in case of heavy programs multiple times.

5. Time-Sharing Operating Systems

Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of the CPU as they use a single system. These systems are also known as Multitasking 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.



Time-Sharing OS

Advantages of Time-Sharing OS

- Each task gets an equal opportunity.
- Fewer chances of duplication of software.
- CPU idle time can be reduced.
- Resource Sharing: Time-sharing systems allow multiple users to share hardware resources such as the CPU, memory, and peripherals, reducing the cost of hardware and increasing efficiency.
- Improved Productivity: Time-sharing allows users to work concurrently, thereby reducing the waiting time for their turn to use the computer. This increased productivity translates to more work getting done in less time.
- Improved User Experience: Time-sharing provides an interactive environment that allows users to communicate with the computer in real time, providing a better user experience than batch processing.

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.

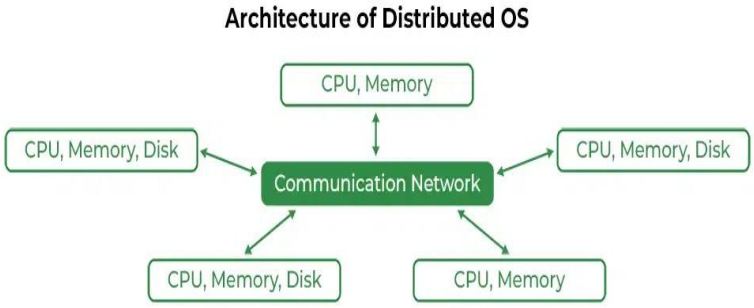
- High Overhead: Time-sharing systems have a higher overhead than other operating systems due to the need for scheduling, context switching, and other overheads that come with supporting multiple users.
- Complexity: Time-sharing systems are complex and require advanced software to manage multiple users simultaneously. This complexity increases the chance of bugs and errors.
- Security Risks: With multiple users sharing resources, the risk of security breaches increases. Time-sharing systems require careful management of user access, authentication, and authorization to ensure the security of data and software.

Examples of Time-Sharing OS with explanation

- **IBM VM/CMS**: IBM VM/CMS is a time-sharing operating system that was first introduced in 1972. It is still in use today, providing a virtual machine environment that allows multiple users to run their own instances of operating systems and applications.
- **TSO (Time Sharing Option)**: TSO is a time-sharing operating system that was first introduced in the 1960s by IBM for the IBM System/360 mainframe computer. It allowed multiple users to access the same computer simultaneously, running their own applications.
- **Windows Terminal Services**: Windows Terminal Services is a time-sharing operating system that allows multiple users to access a Windows server remotely. Users can run their own applications and access shared resources, such as printers and network storage, in real-time.

6. Distributed Operating System

These types of operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, at a great pace. 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 or distributed systems. These systems' processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.



Distributed OS

Advantages of Distributed Operating System

- Failure of one will not affect the other network communication, as all systems are independent of each other.
- Electronic mail increases the data exchange speed.
- Since resources are being shared, computation is highly fast and durable.
- Load on host computer reduces.
- These systems are easily scalable as many systems can be easily added to the network.
- Delay in data processing reduces.

Disadvantages of Distributed Operating System

- Failure of the main network will stop the entire communication.
- To establish distributed systems the language is used not well-defined yet.
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet.

Examples of Distributed Operating Systems are LOCUS, etc.

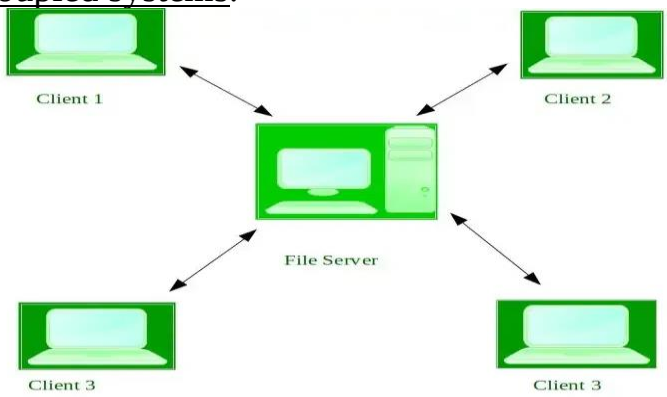
The distributed os must tackle the following issues:

- Networking causes delays in the transfer of data between nodes of a distributed system. Such delays may lead to an inconsistent view of data located in different nodes, and make it difficult to know the chronological order in which events occurred in the system.
- Control functions like scheduling, resource allocation, and deadlock detection have to be performed in several nodes to achieve computation speedup and provide reliable operation when computers or networking components fail.
- Messages exchanged by processes present in different nodes may travel over public networks and pass through computer systems that are not controlled by the distributed operating system. An intruder may exploit this feature to tamper with messages, or create fake messages

to fool the authentication procedure and masquerade as a user of the system.

7. 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 to files, printers, security, applications, and other networking functions over a small private network. One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections, etc. and that's why these computers are popularly known as tightly coupled systems.



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 Systems are Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, BSD, etc.

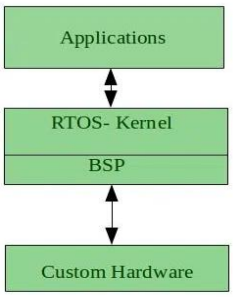
8. Real-Time Operating System

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.

Types of Real-Time Operating Systems

- **Hard Real-Time Systems:** Hard Real-Time OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of an accident. Virtual memory is rarely found in these systems.
- **Soft Real-Time Systems:** These OSs are for applications where time-constraint is less strict.



Real-Time Operating System

Advantages of RTOS

- **Maximum Consumption:** Maximum utilization of devices and systems, thus more output from all the resources.
- **Task Shifting:** The time assigned for shifting tasks in these systems is very less. For example, in older systems, it takes about 10 microseconds in shifting from one task to another, and in the latest systems, it takes 3 microseconds.
- **Focus on Application:** Focus on running applications and less importance on applications that are in the queue.
- Real-time **operating system in** the **embedded system:** Since the size of programs is small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error-free.
- **Memory Allocation:** Memory allocation is best managed in these types of systems.

Disadvantages of RTOS

- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on a few applications to avoid errors.
- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.

- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupts signal to respond earliest to interrupts.
- **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

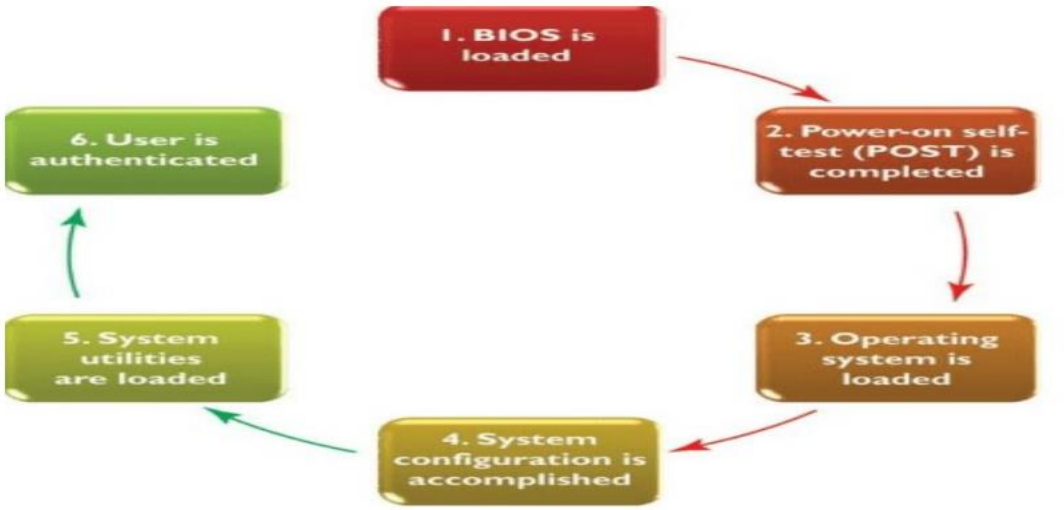
Examples of Real-Time Operating Systems are Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

Booting Process:-

Booting is a process of switching on the computer and starting the operating

system. Six steps of the booting process are BIOS and Setup Program, The PowerOn-Self-Test (POST), The Operating system Loads, System Configuration, System

Utility Loads and Users Authentication.



Steps in the Booting Process: -

1: BIOS and Setup Program

2: The Power-On-Self-Test (POST)

3: The Operating System (OS) Loads

4: System Configuration

5: System Utility Loads

6: Users Authentication

Step 1: BIOS and Setup Program

- ROM (read-only memory): it is a permanent and unchanging memory also
- BIOS (basic input/output system): the part of the system software that includes the instructions that the computer uses to accept input and output
- Load: to transfer from a storage device to memory. The ROM loads BIOS into the computer's memory
- Setup program: a special program containing settings to control hardware. Furthermore, the program can only be accessed while the BIOS information is visible.

Step 2: The Power-On-Self-Test (POST)

- POST (Power-On Self-Test): a series of tests conducted on the computer's

main memory, input/output devices, disk drives, and the hard disk.

- BIOS conducts Power-On-Self-Test to check the input/ output system for

operability.

- The computer will produce a beeping sound if any problem occurs. An error message will also appear on the monitor

Step 3: The Operating System (OS) Loads

- BIOS searches for the operating system.
- Setting in CMOS: complementary metal oxide semiconductor determines where to look for the operating system.
- In this step, the operating system's kernel is also loaded into the computer's memory.
- The operating system takes control of the computer and begins loading system configuration information.

Step 4: System Configuration

- Registry: a database to store information about peripherals and software

- Peripheral: a device connected to a computer
- Drive: a utility program that makes peripheral devices function properly
- The operating system's registry configures the system.
- In this step, drivers are also loaded into memory.

Step 5: System Utility Loads

- System utilities are loaded into memory.
- Volume control
- Antivirus software
- PC card unplugging utility

Step 6: Users Authentication

- Authentication or user login occurs
- Username
- Password
- After all this process, the user interface starts, enabling user interaction with the computer and its programs also.

Types of Booting

There are two types of booting:

Cold Booting

A cold boot is also called a hard boot. It is the process when we first start the computer. In other words, when the computer is started from its initial state by pressing the power button it is called cold boot.

Warm Booting

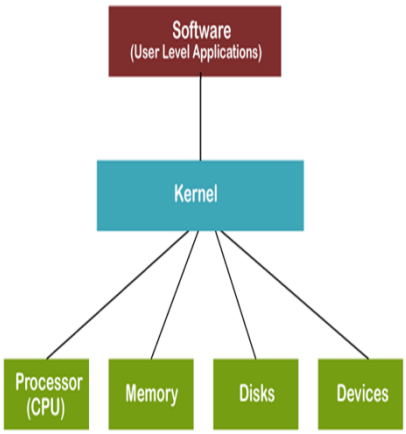
Warm Boot is also called soft boot. It refers to when we restart the computer.

Kernels

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.

It is the main layer between the OS and hardware, and it helps with process and memory management, file systems, device control and networking.

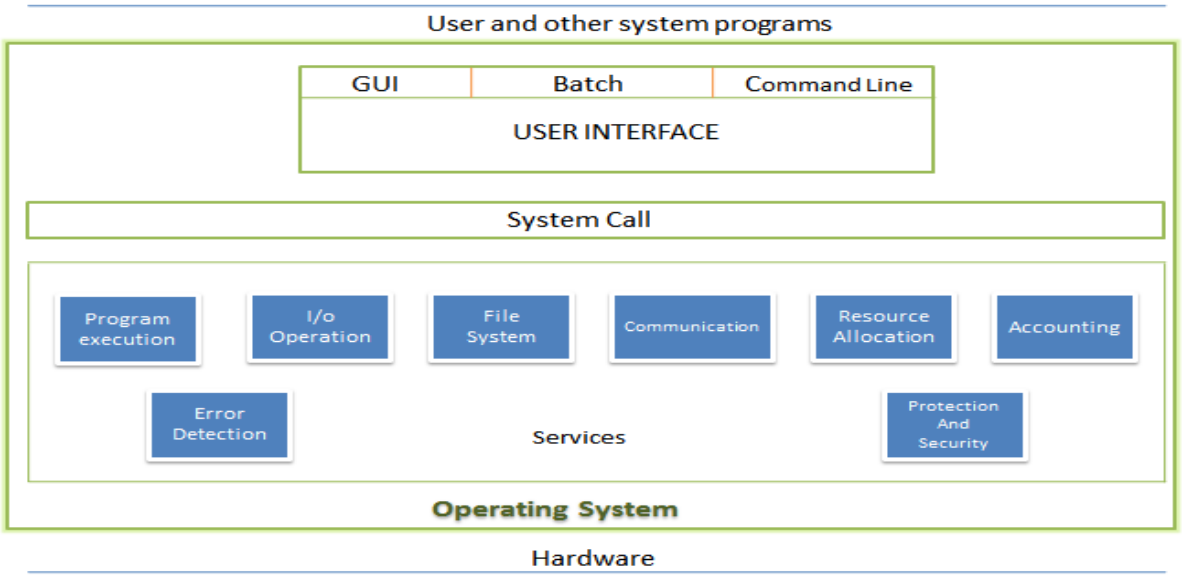
Kernel loads first into memory when an operating system is loaded and remains into memory until operating system is shut down.



Operating-System Services

Q) List and explain the services provided by OS for the user and efficient operation of system.

An operating system provides an environment for the execution of programs. It provides certain services to programs and to the users of those programs.



OS provide services for the users of the system, including:

• **User Interfaces -** Means by which users can issue commands to the system. Depending on the operating system these may be a command-line interface (e.g.

sh, csh, ksh, tcsh, etc.), a Graphical User Interface (e.g. Windows, X-Windows, KDE, Gnome, etc.), or a batch command systems.

In Command Line Interface (CLI)- commands are given to the system.

In Batch interface – commands and directives to control these commands are put in a file and then the file is executed.

In GUI systems- windows with pointing device to get inputs and keyboard to enter the text.

• **Program Execution** - The OS must be able to load a program into RAM, run the program, and terminate the program, either normally or abnormally.

• **I/O Operations** - The OS is responsible for transferring data to and from I/O devices, including keyboards, terminals, printers, and files. For specific devices, special functions are provided (device drivers) by OS.

• **File-System Manipulation** – Programs need to read and write files or directories. The services required to create or delete files, search for a file, list the contents of a file and change the file permissions are provided by OS.

Communications - Inter-process communications, IPC, either between processes running on the same processor, or between processes running on separate processors or separate machines. May be implemented by using the service of OS- like shared memory or message passing.

• **Error Detection** - Both hardware and software errors must be detected and handled appropriately by the OS. Errors may occur in the CPU and memory hardware (such as power failure and memory error), in I/O devices (such as a parity error on tape, a connection failure on a network, or lack of paper in the printer), and in the user program (such as an arithmetic overflow, an attempt to access an illegal memory location).

OS provide services for the efficient operation of the system, including:

• **Resource Allocation** – Resources like CPU cycles, main memory, storage space, and I/O devices must be allocated to multiple users and multiple jobs at the same time.

• **Accounting –** There are services in OS to keep track of system activity and resource usage, either for billing purposes or for statistical record keeping that can be used to optimize future performance.

• **Protection and Security** – The owners of information (file) in multiuser or networked computer system may want to control the use of that information.

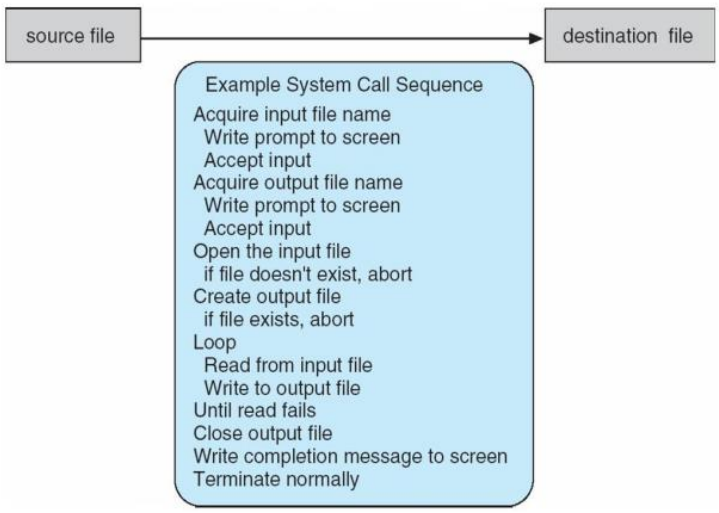
When several separate processes execute concurrently, one process should not interfere with other or with OS. Protection involves ensuring that all access to system resources is controlled. Security of the system from outsiders must also be done, by means of a password.

System Calls

Q) What are system calls? Briefly point out its types.

• System calls provides an interface to the services of the operating system. These are generally written in C or C++, although some are written in assembly for optimal performance.

• The below figure illustrates the sequence of system calls required to copy a file content from one file (input file) to another file (output file).



An example to illustrate how system calls are used: writing a simple program to read data from one file and copy them to another file

• There are number of system calls used to finish this task. The first system call is to write a message on the screen (monitor). Then to accept the input filename. Then another system call to write message on the screen, then to accept the output filename.

When the program tries to open the input file, it may find that there is no file of that name or that the file is protected against access. In these cases, the program should print a message on the console (another system call) and then terminate abnormally (another system call) and create a new one (another system call).

• Now that both the files are opened, we enter a loop that reads from the input file (another system call) and writes to output file (another system call).

• Finally, after the entire file is copied, the program may close both files (another system call), write a message to the console or window (system call), and finally terminate normally (final system call).

• Most programmers do not use the low-level system calls directly, but instead use an "Application Programming Interface", API.

• Instead of direct system calls provides for greater program portability between different systems. The API then makes the appropriate system calls through the system call interface, using a system call table to access specific numbered system calls.

• Each system call has a specific numbered system call. The system call table (consisting of system call number and address of the particular service) invokes a particular service

routine for a specific system call.

• The caller need know nothing about how the system call is implemented or what it does during execution.

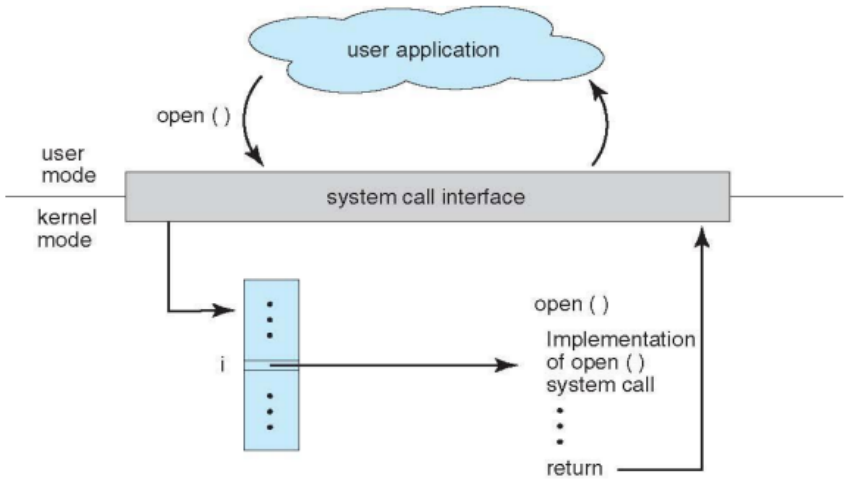


Figure: The handling of a user application invoking the open() system call.

Types of System Calls

The system calls can be categorized into six major categories:

1. Process Control

2. File management

3. Device management

4. Information management

5. Communications

6. Protection

EXAMPLES OF WINDOWS AND UNIX SYSTEM CALLS

The following illustrates various equivalent system calls for Windows and UNIX operating systems.

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

1. Process Control

• Process control system calls include end, abort, load, execute, create process, terminate process, get/set process attributes, wait for time or event, signal event, and allocate and free memory.

• Processes must be created, launched, monitored, paused, resumed, and eventually stopped.

• When one process pauses or stops, then another must be launched or resumed

• Process attributes like process priority, max. allowable execution time etc. are set and retrieved by OS.

• After creating the new process, the parent process may have to wait (wait time), or wait for an event to occur (wait event). The process sends back a signal when the event has occurred (signal event)

2. File Management

The file management functions of OS are –

• File management system calls include create file, delete file, open, close, read, write, reposition, get file attributes, and set file attributes.

• After creating a file, the file is opened. Data is read or written to a file.

• The file pointer may need to be repositioned to a point.

• The file attributes like filename, file type, permissions, etc. are set and retrieved using system calls.

• These operations may also be supported for directories as well as ordinary files.

3. Device Management

• Device management system calls include request device, release device, read, write, reposition, get/set device attributes, and logically attach or detach devices.

• When a process needs a resource, a request for resource is done. Then the control is granted to the process. If requested resource is already attached to some other process, the requesting process has to wait.

• In multiprogramming systems, after a process uses the device, it has to be returned to OS, so that another process can use the device.

• Devices may be physical (e.g. disk drives), or virtual / abstract (e.g. files, partitions, and RAM disks).

4. Information Maintenance

• Information maintenance system calls include calls to get/set the time, date, system data, and process, file, or device attributes.

• These system calls care used to transfer the information between user and the OS.

Information like current time & date, no. of current users, version no. of OS, amount of free memory, disk space etc. are passed from OS to the user.

5. Communication

• Communication system calls create/delete communication connection, send/receive messages, transfer status information, and attach/detach remote devices.

• The message passing model must support calls to:

o Identify a remote process and/or host with which to communicate.

o Establish a connection between the two processes.

o Open and close the connection as needed.

o Transmit messages along the connection.

o Wait for incoming messages, in either a blocking or non-blocking state.

o Delete the connection when no longer needed.

• The shared memory model must support calls to:

o Create and access memory that is shared amongst processes (and threads.)

o Free up shared memory and/or dynamically allocate it as needed.

• Message passing is simpler and easier, (particularly for inter-computer communications), and is generally appropriate for small amounts of data. It is easy to implement, but there are system calls for each read and write process.

• Shared memory is faster, and is generally the better approach where large amounts of data are to be shared. This model is difficult to implement, and it consists of only few system calls.

6. Protection

• Protection provides mechanisms for controlling which users / processes have access to which system resources.

• System calls allow the access mechanisms to be adjusted as needed, and for non- privileged users to be granted elevated access permissions under carefully controlled temporary circumstances.

System Programs

Q) List and explain the different categories of system program?

A collection of programs that provide a convenient environment for program development and execution (other than OS) are called system programs or system utilities.

System programs may be divided into five categories:

1. File management - programs to create, delete, copy, rename, print, list, and generally manipulate files and directories.

2. Status information - Utilities to check on the date, time, number of users, processes running, data logging, etc. System registries are used to store and recall configuration information for particular applications.

3. File modification - e.g. text editors and other tools which can change file contents.

4. Programming-language support - E.g. Compilers, linkers, debuggers, profilers, assemblers, library archive management, interpreters for common languages, and support for make.

5. Program loading and execution - loaders, dynamic loaders, overlay loaders, etc., as well as interactive debuggers.

6. Communications - Programs for providing connectivity between processes and users, including mail, web browsers, remote logins, file transfers, and remote command execution.

Components of Operating System

There are various components of an Operating System to perform well defined tasks. Though most of the Operating Systems differ in structure but logically they have similar components. Each component must be a well-defined portion of a system that appropriately describes the functions, inputs, and outputs.

There are following 8-components of an Operating System:

- Process Management
- I/O Device Management
- File Management
- Network Management
- Main Memory Management
- Secondary Storage Management
- Security Management
- Command Interpreter System

Following section explains all the above components in more detail:

Process Management

A process is program or a fraction of a program that is loaded in main memory. A process needs certain resources including CPU time, Memory, Files, and I/O devices to accomplish its task. The process management component manages the multiple processes running simultaneously on the Operating System.

A program in running state is called a process.

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

- Create, load, execute, suspend, resume, and terminate processes.
- Switch system among multiple processes in main memory.
- Provides communication mechanisms so that processes can communicate with each others
- Provides synchronization mechanisms to control concurrent access to shared data to keep shared data consistent.
- Allocate/de-allocate resources properly to prevent or avoid deadlock situation.

I/O Device Management

One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from the user. I/O Device Management provides an abstract level of H/W devices and keep the details from applications to ensure proper use of devices, to prevent errors, and to provide users with convenient and efficient programming environment.

Following are the tasks of I/O Device Management component:

- Hide the details of H/W devices
- Manage main memory for the devices using cache, buffer, and spooling
- Maintain and provide custom drivers for each device.

File Management

File management is one of the most visible services of an operating system. Computers can store information in several different physical forms; magnetic tape, disk, and drum are the most common forms.

A file is defined as a set of correlated information and it is defined by the creator of the file. Mostly files represent data, source and object forms, and programs. Data files can be of any type like alphabetic, numeric, and alphanumeric.

A files is a sequence of bits, bytes, lines or records whose meaning is defined by its creator and user.

The operating system implements the abstract concept of the file by managing mass storage device, such as types and disks. Also files are normally organized into directories to ease their use. These directories may contain files and other directories and so on.

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

- File creation and deletion
- Directory creation and deletion
- The support of primitives for manipulating files and directories
- Mapping files onto secondary storage
- File backup on stable (nonvolatile) storage media

Network Management

The definition of network management is often broad, as network management involves several different components. Network management is the process of managing and administering a computer network. A computer network is a collection of various types of computers connected with each other.

Network management comprises fault analysis, maintaining the quality of service, provisioning of networks, and performance management.

Network management is the process of keeping your network healthy for an efficient communication between different computers.

Following are the features of network management:

- Network administration
- Network maintenance
- Network operation
- Network provisioning
- Network security

Main 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 which means it loses its contents in the case of system failure or as soon as system power goes down.

The main motivation behind Memory Management is to maximize memory utilization on the computer system.

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 deallocate memory space as needed.

Secondary Storage Management

The main purpose of a computer system is to execute programs. These programs, together with the data they access, must be in main memory during execution. Since the main memory is too small to permanently accommodate all data and program, the computer system must provide secondary storage to backup main memory.

Most modern computer systems use disks as the principle on-line storage medium, for both programs and data. Most programs, like compilers, assemblers, sort routines, editors, formatters, and so on, are stored on the disk until loaded into memory, and then use the disk as both the source and destination of their processing.

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

- Free space management
- Storage allocation
- Disk scheduling

Security Management

The operating system is primarily responsible for all task and activities happen in the computer system. The various processes in an operating system must be protected from each other's activities. For that purpose, various mechanisms which can be used to ensure that the files, memory segment, cpu and other resources can be operated on only by those processes that have gained proper authorization from the operating system.

Security Management refers to a mechanism for controlling the access of programs, processes, or users to the resources defined by a computer controls to be imposed, together with some means of enforcement.

For example, memory addressing hardware ensure that a process can only execute within its own address space. The timer ensure that no process can gain control of the CPU without relinquishing it. Finally, no process is allowed to do it's own I/O, to protect the integrity of the various peripheral devices.

Command Interpreter System

One of the most important component of an operating system is its command interpreter. The command interpreter is the primary interface between the user and the rest of the system.

Command Interpreter System executes a user command by calling one or more number of underlying system programs or system calls.

Command Interpreter System allows human users to interact with the Operating System and provides convenient programming environment to the users.

Many commands are given to the operating system by control statements. A program which reads and interprets control statements is automatically executed. This program is called the shell and few examples are Windows DOS command window, Bash of Unix/Linux or C-Shell of Unix/Linux.