OS SLYABUS

Unit1

: Introduction, Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Kernel Data Structures, Computing Environments, Open-Source Operating Systems, Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating-System Structure, Operating-System Debugging, Operating-System Generation, System Boot

Unit1 Introduction

An operating system is software that manages computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

Operating System: Definition:

- An operating system is a program that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware.
- A more common definition is that the operating system is the one program running at all times on the computer (usually called the kernel), with all else being application programs.
- An operating system is concerned with the allocation of resources and services, such as memory, processors, devices, and information. The operating system correspondingly includes programs to manage these resources, such as a traffic controller, a scheduler, a memory management module, I/O programs, and a file system.

Features of Operating system: Operating system has the following features:

1. **Convenience:** An OS makes a computer more convenient to use.
2. **Efficiency:** An OS allows the computer system resources to be used efficiently.
3. **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.
4. **Throughput:** An OS should be constructed so that It can give maximum **throughput**(Number of tasks per unit time).

Major Functionalities of Operating System:

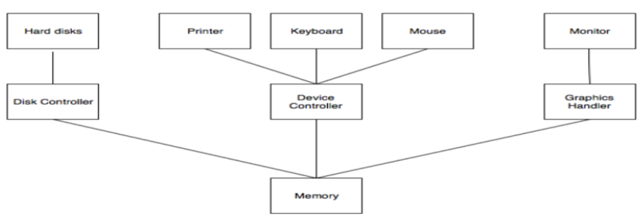
- **Resource Management:** When parallel accessing happens in the OS means when multiple users are accessing the system the OS works as Resource Manager, Its responsibility is to provide hardware to the user. It decreases the load in the system.
- **Process Management:** It includes various tasks like **scheduling and termination** of the process. It is done with the help of **CPU Scheduling** algorithms.
- **Storage Management:** The **file system** mechanism used for the management of the storage. **NIFS**, **CFS**, **CIFS**, **NFS**, etc. are some file systems. All the data is stored in various tracks of Hard disks that are all managed by the storage manager. It included **Hard Disk**.
- **Memory Management:** Refers to the management of primary memory. The operating system has to keep track of how much memory has been used and by whom. It has to decide which process needs memory space and how much. OS also has to allocate and deallocate the memory space.

The process operating system as User Interface:

1. User
2. System and application programs
3. Operating system
4. Hardware

Computer-System Organization

The computer system is a combination of many parts such as peripheral devices, secondary memory, CPU etc. This can be explained more clearly using a diagram.



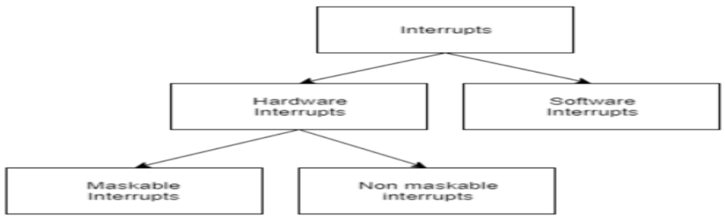
The salient points about the above figure displaying Computer System Organisation is −

- The I/O devices and the CPU both execute concurrently. Some of the processes are scheduled for the CPU and at the same time, some are undergoing input/output operations.
- There are multiple device controllers, each in charge of a particular device such as keyboard, mouse, printer etc.
- There is buffer available for each of the devices. The input and output data can be stored in these buffers.
- The data is moved from memory to the respective device buffers by the CPU for I/O operations and then this data is moved back from the buffers to memory.
- The device controllers use an interrupt to inform the CPU that I/O operation is completed.

Interrupt Handling

An interrupt is a necessary part of Computer System Organisation as it is triggered by hardware and software parts when they need immediate attention.

An interrupt can be generated by a device or a program to inform the operating system to halt its current activities and focus on something else. The types of interrupts are better explained using the following diagram −



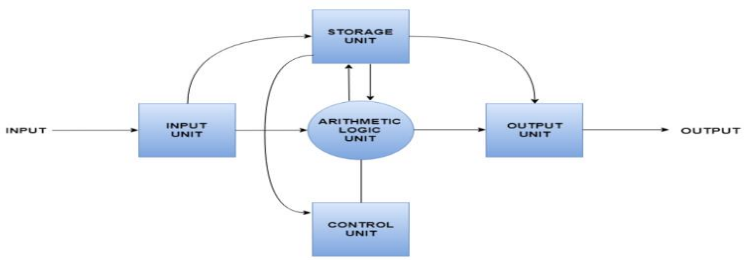
Hardware and software interrupts are two types of interrupts. Hardware interrupts are triggered by hardware peripherals while software interrupts are triggered by software function calls.

Hardware interrupts are of further two types. Maskable interrupts can be ignored or disabled by the CPU while this is not possible for non maskable interrupts.

Computer-System Architecture

computer system is basically a machine that simplifies complicated tasks. It should maximize performance and reduce costs as well as power consumption.The different components in the Computer System Architecture are Input Unit, Output Unit, Storage Unit, Arithmetic Logic Unit, Control Unit etc.

A diagram that shows the flow of data between these units is as follows −



The input data travels from input unit to ALU. Similarly, the computed data travels from ALU to output unit. The data constantly moves from storage unit to ALU and back again. This is because stored data is computed on before being stored again. The control unit controls all the other units as well as their data.

Details about all the computer units are −

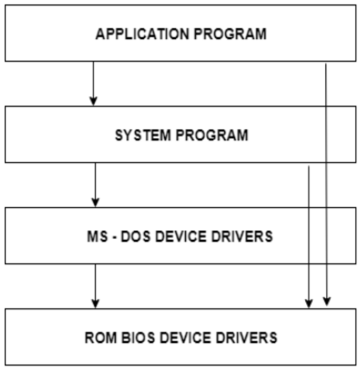
- **Input Unit**
  The input unit provides data to the computer system from the outside. So, basically it links the external environment with the computer. It takes data from the input devices, converts it into machine language and then loads it into the computer system. Keyboard, mouse etc. are the most commonly used input devices.
- **Output Unit**
  The output unit provides the results of computer process to the users i.e it links the computer with the external environment. Most of the output data is the form of audio or video. The different output devices are monitors, printers, speakers, headphones etc.
- **Storage Unit**
  Storage unit contains many computer components that are used to store data. It is traditionally divided into primary storage and secondary storage.Primary storage is also known as the main memory and is the memory directly accessible by the CPU. Secondary or external storage is not directly accessible by the CPU. The data from secondary storage needs to be brought into the primary storage before the CPU can use it. Secondary storage contains a large amount of data permanently.
- **Arithmetic Logic Unit**
  All the calculations related to the computer system are performed by the arithmetic logic unit. It can perform operations like addition, subtraction, multiplication, division etc. The control unit transfers data from storage unit to arithmetic logic unit when calculations need to be performed. The arithmetic logic unit and the control unit together form the central processing unit.
- **Control Unit**
  This unit controls all the other units of the computer system and so is known as its central nervous system. It transfers data throughout the computer as required including from storage unit to central processing unit and vice versa. The control unit also dictates how the memory, input output devices, arithmetic logic unit etc. should behave.

Operating-System Structure

An operating system is a construct that allows the user application programs to interact with the system hardware. Since the operating system is such a complex structure, it should be created with utmost care so it can be used and modified easily. An easy way to do this is to create the operating system in parts. Each of these parts should be well defined with clear inputs, outputs and functions.

Simple Structure

the structure of MS-DOS is as follows −



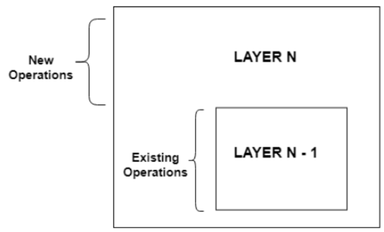
MS-DOS STRUCTURE

It is better that operating systems have a modular structure, unlike MS-DOS. That would lead to greater control over the computer system and its various applications. The modular structure would also allow the programmers to hide information as required and implement internal routines as they see fit without changing the outer specifications.

Layered Structure

One way to achieve modularity in the operating system is the layered approach. In this, the bottom layer is the hardware and the topmost layer is the user interface.

An image demonstrating the layered approach is as follows −



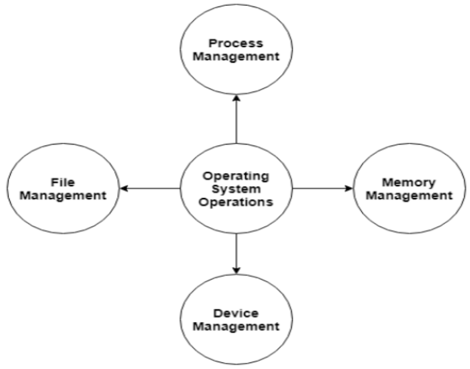
Layered Structure of Operating System

As seen from the image, each upper layer is built on the bottom layer. All the layers hide some structures, operations etc from their upper layers.

One problem with the layered structure is that each layer needs to be carefully defined. This is necessary because the upper layers can only use the functionalities of the layers below them.

Operating-System Operations

The major operations of the operating system are process management, memory management, device management and file management. These are given in detail as follows:



Process Management

The operating system is responsible for managing the processes i.e assigning the processor to a process at a time. This is known as process scheduling. The different algorithms used for process scheduling are FCFS (first come first served), SJF (shortest job first), priority scheduling, round robin scheduling etc.

There are many scheduling queues that are used to handle processes in process management. When the processes enter the system, they are put into the job queue. The processes that are ready to execute in the main memory are kept in the ready queue. The processes that are waiting for the I/O device are kept in the device queue.

Memory Management

Memory management plays an important part in operating system. It deals with memory and the moving of processes from disk to primary memory for execution and back again.

The activities performed by the operating system for memory management are −

- The operating system assigns memory to the processes as required. This can be done using best fit, first fit and worst fit algorithms.
- All the memory is tracked by the operating system i.e. it nodes what memory parts are in use by the processes and which are empty.

Device Management

There are many I/O devices handled by the operating system such as mouse, keyboard, disk drive etc. There are different device drivers that can be connected to the operating system to handle a specific device. The device controller is an interface between the device and the device driver. The user applications can access all the I/O devices using the device drivers, which are device specific codes.

File Management

Files are used to provide a uniform view of data storage by the operating system. All the files are mapped onto physical devices that are usually non volatile so data is safe in the case of system failure.

The files can be accessed by the system in two ways i.e. sequential access and direct access −

- **Sequential Access**
  The information in a file is processed in order using sequential access. The files records are accessed on after another. Most of the file systems such as editors, compilers etc. use sequential access.
- **Direct Access**
  In direct access or relative access, the files can be accessed in random for read and write operations. The direct access model is based on the disk model of a file, since it allows random accesses.

Process Management in OS

A Program does nothing unless its instructions are executed by a CPU. A program in execution is called a process. 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.

Some resources may need to be executed by one process at one time to maintain the consistency otherwise the system can become inconsistent and deadlock may occur.

The operating system is responsible for the following activities in connection with Process Management

1. Scheduling processes and threads on the CPUs.
2. Creating and deleting both user and system processes.
3. Suspending and resuming processes.
4. Providing mechanisms for process synchronization.
5. Providing mechanisms for process communication.

Attributes of a process

The Attributes of the process are used by the Operating System to create the process control block (PCB) for each of them. This is also called context of the process. Attributes which are stored in the PCB are described below.

1. Process ID

When a process is created, a unique id is assigned to the process which is used for unique identification of the process in the system.

2. Program counter

A program counter stores the address of the last instruction of the process on which the process was suspended. The CPU uses this address when the execution of this process is resumed.

3. Process State

The Process, from its creation to the completion, goes through various states which are new, ready, running and waiting. We will discuss about them later in detail.

4. Priority

Every process has its own priority. The process with the highest priority among the processes gets the CPU first. This is also stored on the process control block.

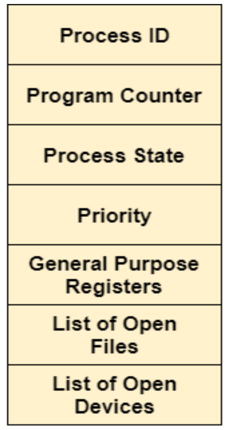
5. General Purpose Registers

Every process has its own set of registers which are used to hold the data which is generated during the execution of the process.

6. List of open files

During the Execution, Every process uses some files which need to be present in the main memory. OS also maintains a list of open files in the PCB.

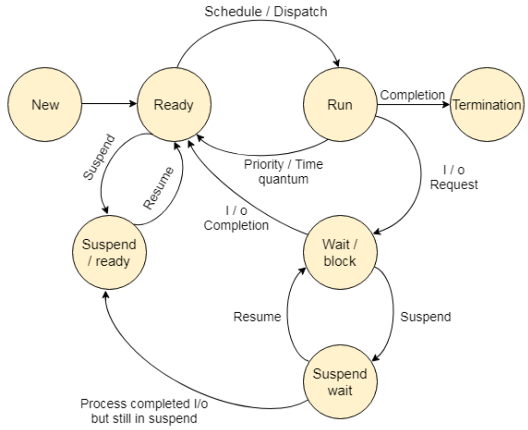
7. List of open devices
OS also maintain the list of all open devices which are used during the execution of the process.



Process Attributes

Process States

State Diagram



The process, from its creation to completion, passes through various states. The minimum number of states is five.

The names of the states are not standardized although the process may be in one of the following states during execution.

1. New **A program which is going to be picked up by the OS into the main memory is called a new process.**

2. Ready **Whenever a process is created, it directly enters in the ready state, in which, it waits for the CPU to be assigned. The OS picks the new processes from the secondary memory and put all of them in the main memory.**

The processes which are ready for the execution and reside in the main memory are called ready state processes. There can be many processes present in the ready state.

3. Running **One of the processes from the ready state will be chosen by the OS depending upon the scheduling algorithm. Hence, if we have only one CPU in our system, the number of running processes for a particular time will always be one. If we have n processors in the system then we can have n processes running simultaneously.**

4. Block or wait **From the Running state, a process can make the transition to the block or wait state depending upon the scheduling algorithm or the intrinsic behavior of the process.**

When a process waits for a certain resource to be assigned or for the input from the user then the OS move this process to the block or wait state and assigns the CPU to the other processes.

5. Completion or termination **When a process finishes its execution, it comes in the termination state. All the context of the process (Process Control Block) will also be deleted the process will be terminated by the Operating system.**

6. Suspend ready **A process in the ready state, which is moved to secondary memory from the main memory due to lack of the resources (mainly primary memory) is called in the suspend ready state.**

If the main memory is full and a higher priority process comes for the execution then the OS have to make the room for the process in the main memory by throwing the lower priority process out into the secondary memory. The suspend ready processes remain in the secondary memory until the main memory gets available.

7. Suspend wait **Instead of removing the process from the ready queue, it's better to remove the blocked process which is waiting for some resources in the main memory. Since it is already waiting for some resource to get available hence it is better if it waits in the secondary memory and make room for the higher priority process. These processes complete their execution once the main memory gets available and their wait is finished.**

Operations on the Process

1. Creation **Once the process is created, it will be ready and come into the ready queue (main memory) and will be ready for the execution.**

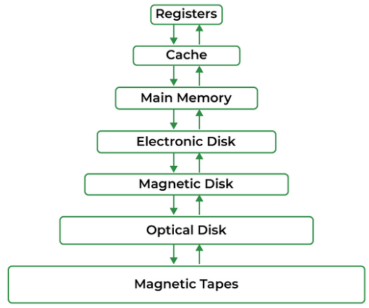
2. Scheduling **Out of the many processes present in the ready queue, the Operating system chooses one process and start executing it. Selecting the process which is to be executed next, is known as scheduling.**

3. Execution **Once the process is scheduled for the execution, the processor starts executing it. Process may come to the blocked or wait state during the execution then in that case the processor starts executing the other processes.**

4. Deletion/killing **Once the purpose of the process gets over then the OS will kill the process. The Context of the process (PCB) will be deleted and the process gets terminated by the Operating system.**

Memory Management The term Memory can be defined as a collection of data in a specific format. It is used to store instructions and process data. The memory comprises a large array or group of words or bytes, each with its own location. The primary motive of a computer system is to execute programs. These programs, along with the information they access, should be in the main memory during execution. The CPU fetches instructions from memory according to the value of the program counter.
To achieve a degree of multiprogramming and proper utilization of memory, memory management is important. Many memory management methods exist, reflecting various approaches, and the effectiveness of each algorithm depends on the situation.

Main Memory?
The main memory is central to the operation of a Modern Computer. Main Memory is a large array of words or bytes, ranging in size from hundreds of thousands to billions. Main memory is a repository of rapidly available information shared by the CPU and I/O devices. Main memory is the place where programs and information are kept when the processor is effectively utilizing them. Main memory is associated with the processor, so moving instructions and information into and out of the processor is extremely fast. Main memory is also known as RAM (Random Access Memory). This memory is a volatile memory. RAM lost its data when a power interruption occurs.



Why Memory Management is Required?

- Allocate and de-allocate memory before and after process execution.
- To keep track of used memory space by processes.
- To minimize fragmentation issues.
- To proper utilization of main memory.
- To maintain data integrity while executing of process.

Now we are discussing the concept of Logical Address Space and Physical Address Space

Logical and Physical Address Space

- **Logical Address Space:** An address generated by the CPU is known as a "Logical Address". It is also known as a Virtual address. Logical address space can be defined as the size of the process. A logical address can be changed.
- **Physical Address Space:** An address seen by the memory unit (i.e the one loaded into the memory address register of the memory) is commonly known as a "Physical Address". A Physical address is also known as a Real address. The set of all physical addresses corresponding to these logical addresses is known as Physical address space. A physical address is computed by MMU. The run-time mapping from virtual to physical addresses is done by a hardware device Memory Management Unit(MMU). The physical address always remains constant.

Static and Dynamic Loading

Loading a process into the main memory is done by a loader. There are two different types of loading :

- **Static Loading:** Static Loading is basically loading the entire program into a fixed address. It requires more memory space.
- **Dynamic Loading:** The entire program and all data of a process must be in physical memory for the process to execute. So, the size of a process is limited to the size of physical memory. To gain proper memory utilization, dynamic loading is used. In dynamic loading, a routine is not loaded until it is called. All routines are residing on disk in a relocatable load format. One of the advantages of dynamic loading is that the unused routine is never loaded. This loading is useful when a large amount of code is needed to handle it efficiently.

Static and Dynamic Linking

To perform a linking task a linker is used. A linker is a program that takes one or more object files generated by a compiler and combines them into a single executable file.

- **Static Linking:** In static linking, the linker combines all necessary program modules into a single executable program. So there is no runtime dependency. Some operating systems support only static linking, in which system language libraries are treated like any other object module.
- **Dynamic Linking:** The basic concept of dynamic linking is similar to dynamic loading. In dynamic linking, "Stub" is included for each appropriate library routine reference. A stub is a small piece of code. When the stub is executed, it checks whether the needed routine is already in memory or not. If not available then the program loads the routine into memory.

Storage Management

Storage Management is defined as it refers to the management of the data storage equipment's that are used to store the user/computer generated data. Hence it is a tool or set of processes used by an administrator to keep your data and storage equipment's safe. Storage management is a process for users to optimize the use of storage devices and to protect the integrity of data for any media on which it resides and the category of storage management generally contain the different type of subcategories covering aspects such as security, virtualization and more, as well as different types of provisioning or automation, which is generally made up the entire storage management software market.

Storage management key attributes: Storage management has some key attribute which is generally used to manage the storage capacity of the system. These are given below:

1. Performance
2. Reliability
3. Recoverability
4. Capacity

Feature of Storage management: There is some feature of storage management which is provided for

Feature of Storage management: There is some feature of storage management which is provided for storage capacity. These are given below:

1. Storage management is a process that is used to optimize the use of storage devices.
2. Storage management must be allocated and managed as a resource in order to truly benefit a corporation.
3. Storage management is generally a basic system component of information systems.
4. It is used to improve the performance of their data storage resources.

Advantage of storage management: There are some advantage of storage management which are given below:

- It generally reduces the time consumption.
- It improves the performance of system.
- In virtualization and automation technologies, it can help an organization improve its agility.

Limitations of storage management:

- Limited physical storage capacity: Operating systems can only manage the physical storage space that is available, and as such, there is a limit to how much data can be stored.
- Performance degradation with increased storage utilization: As more data is stored, the system's performance can decrease due to increased disk access time, fragmentation, and other factors.
- Complexity of storage management: Storage management can be complex, especially as the size of the storage environment grows.
- Cost: Storing large amounts of data can be expensive, and the cost of additional storage capacity can add up quickly.
- Security issues: Storing sensitive data can also present security risks, and the operating system must have robust security features in place to prevent unauthorized access to this data.
- Backup and Recovery: Backup and recovery of data can also be challenging, especially if the data is stored on multiple systems or devices

Protection and Security

Protection and security requires that computer resources such as CPU, softwares, memory etc. are protected. This extends to the operating system as well as the data in the system. This can be done by ensuring integrity, confidentiality and availability in the operating system. The system must be protect against unauthorized access, viruses, worms etc.

Threats to Protection and Security

A threat is a program that is malicious in nature and leads to harmful effects for the system. Some of the common threats that occur in a system are −

Virus

Viruses are generally small snippets of code embedded in a system. They are very dangerous and can corrupt files, destroy data, crash systems etc. They can also spread further by replicating themselves as required.

Trojan Horse

A trojan horse can secretly access the login details of a system. Then a malicious user can use these to enter the system as a harmless being and wreak havoc.

Trap Door

A trap door is a security breach that may be present in a system without the knowledge of the users. It can be exploited to harm the data or files in a system by malicious people.

Worm

A worm can destroy a system by using its resources to extreme levels. It can generate multiple copies which claim all the resources and don't allow any other processes to access them. A worm can shut down a whole network in this way.

Denial of Service

These type of attacks do not allow the legitimate users to access a system. It overwhelms the system with requests so it is overwhelmed and cannot work properly for other user.

Protection and Security Methods

The different methods that may provide protect and security for different computer systems are −

Authentication

This deals with identifying each user in the system and making sure they are who they claim to be. The operating system makes sure that all the users are authenticated before they access the system. The different ways to make sure that the users are authentic are:

- **Username/ Password**
  Each user has a distinct username and password combination and they need to enter it correctly before they can access the system.
- **User Key/ User Card**
  The users need to punch a card into the card slot or use they individual key on a keypad to access the system.
- **User Attribute Identification**
  Different user attribute identifications that can be used are fingerprint, eye retina etc. These are unique for each user and are compared with the existing samples in the database. The user can only access the system if there is a match.

One Time Password

These passwords provide a lot of security for authentication purposes. A one time password can be generated exclusively for a login every time a user wants to enter the system. It cannot be used more than once. The various ways a one time password can be implemented are −

- **Random Numbers**
  The system can ask for numbers that correspond to alphabets that are pre arranged. This combination can be changed each time a login is required.
- **Secret Key**
  A hardware device can create a secret key related to the user id for login. This key can change each time.

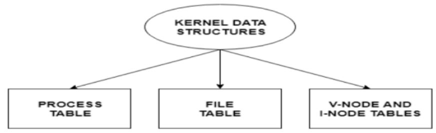
Kernel Data Structures

The kernel data structures are very important as they store data about the current state of the system. For example, if a new process is created in the system, a kernel data structure is created that contains the details about the process.

Most of the kernel data structures are only accessible by the kernel and its subsystems. They may contain data as well as pointers to other data structures.

Kernel Components

The kernel stores and organizes a lot of information. So it has data about which processes are running in the system, their memory requirements, files in use etc. To handle all this, three important structures are used. These are process table, file table and v node/ i node information.



Process Table

The process table stores information about all the processes running in the system. These include the storage information, execution status, file information etc.

When a process forks a child, its entry in the process table is duplicated including the file information and file pointers. So the parent and the child process share a file.

File Table

The file table contains entries about all the files in the system. If two or more processes use the same file, then they contain the same file information and the file descriptor number.

Each file table entry contains information about the file such as file status (file read or file write), file offset etc. The file offset specifies the position for next read or write into the file.

The file table also contains v-node and i-node pointers which point to the virtual node and index node respectively. These nodes contain information on how to read a file.

V-Node and I-Node Tables

Both the v-node and i-node are references to the storage system of the file and the storage mechanisms. They connect the hardware to the software.

The v-node is an abstract concept that defines the method to access file data without worrying about the actual structure of the system. The i-node specifies file access information like file storage device, read/write procedures etc.

Computing Environments

Computing environments refer to the technology infrastructure and software platforms that are used to develop, test, deploy, and run software applications. There are several types of computing environments, including:

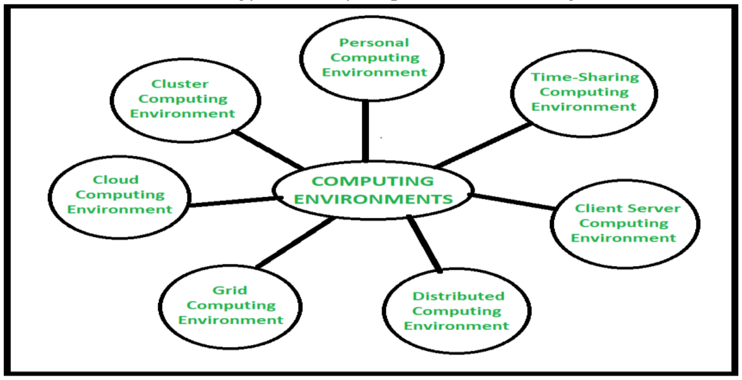
1. Mainframe: A large and powerful computer system used for critical applications and large-scale data processing.
2. Client-Server: A computing environment in which client devices access resources and services from a central server.
3. Cloud Computing: A computing environment in which resources and services are provided over the Internet and accessed through a web browser or client software.
4. Mobile Computing: A computing environment in which users access information and applications using handheld devices such as smartphones and tablets.
5. Grid Computing: A computing environment in which resources and services are shared across multiple computers to perform large-scale computations.
6. Embedded Systems: A computing environment in which software is integrated into devices and products, often with limited processing power and memory.

Each type of computing environment has its own advantages and disadvantages, and the choice of environment depends on the specific requirements of the software application and the resources available.

In the world of technology where every tasks are performed with help of computers, these computers have become one part of human life. Computing is nothing but process of completing a task by using this computer technology and it may involve computer hardware and/or software. But computing uses some form of computer system to manage, process, and communicate information. After getting some idea about computing now lets understand about **computing environments**.

Computing Environments : When a problem is solved by the computer, during that computer uses many devices, arranged in different ways and which work together to solve problems. This constitutes a computing environment where various number of computer devices arranged in different ways to solve different types of problems in different ways. In different computing environments computer devices are arranged in different ways and they exchange information in between them to process and solve problem. One computing environment consists of many computers other computational devices, software and networks that to support processing and sharing information and solving task. Based on the organization of different computer devices and communication processes there exists multiple types of **computing environments**.

Types of Computing Environments : There are the various types of computing environments. They are :



Computing Environments Types

1. **Personal Computing Environment :** In personal computing environment there is a stand-alone machine. Complete program resides on computer and executed there. Different stand-alone machines that constitute a personal computing environment are laptops, mobiles, printers, computer systems, scanners etc. That we use at our homes and offices.
2. **Time-Sharing Computing Environment :** In Time Sharing Computing Environment multiple users share system simultaneously. Different users (different processes) are allotted different time slice and processor switches rapidly among users according to it. For example, student listening to music while coding something in an IDE. Windows 95 and later versions, Unix, IOS, Linux operating systems are the examples of this time sharing computing environment.
3. **Client Server Computing Environment :** In client server computing environment two machines are involved i.e., client machine and server machine, sometime same machine also serve as client and server. In this computing environment client requests resource/service and server provides that respective resource/service. A server can provide service to multiple clients at a time and here mainly communication happens through computer network.
4. **Distributed Computing Environment :** In a distributed computing environment multiple nodes are connected together using network but physically they are separated. A single task is performed by different functional units of different nodes of distributed unit. Here different programs of an application run simultaneously on different nodes, and communication happens in between different nodes of this system over network to solve task.
5. **Grid Computing Environment :** In grid computing environment, multiple computers from different locations works on single problem. In this system set of computer nodes running in cluster jointly perform a given task by applying resources of multiple computers/nodes. It is network of computing environment where several scattered resources provide running environment for single task.
6. **Cloud Computing Environment :** In cloud computing environment on demand availability of computer system resources like processing and storage are availed. Here computing is not done in individual technology or computer rather it is computed in cloud of computers where all required resources

are provided by cloud vendor. This environment primarily comprised of three services i.e software-as-a-service (SaaS), infrastructure-as-a-service (IaaS), and platform-as-a-service (PaaS).

7. **Cluster Computing Environment :** In cluster computing environment cluster performs task where cluster is a set of loosely or tightly connected computers that work together. It is viewed as single system and performs task parallelly that's why also it is similar to parallel computing environment. Cluster aware applications are especially used in cluster computing environment.

Open-Source Operating System?

The term **"open source"** refers to computer software or applications where the owners or copyright holders enable the users or third parties to use, see, and edit the product's source code. The source code of an open-source OS is publicly visible and editable. The usually operating systems such as Apple's iOS, Microsoft's Windows, and Apple's Mac OS are closed operating systems. Open-Source Software is licensed in such a way that it is permissible to produce as many copies as you want and to use them wherever you like. It generally uses fewer resources than its commercial counterpart because it lacks any code for licensing, promoting other products, authentication, attaching advertisements, etc.

The open-source operating system allows the use of code that is freely distributed and available to anyone and for commercial purposes. Being an open-source application or program, the program source code of an open-source OS is available. The user may modify or change those codes and develop new applications according to the user requirement. Some basic examples of the open-source operating systems are **Linux, Open Solaris, Free RTOS, Open BDS, Free BSD, Minix,** etc.

In **1997**, the first Open-Source software was released. Despite the industry, there are now Open-Source alternatives for every Software program. Thanks to technological developments and innovations, many Open-Source Operating Systems have been developed since the dawn of the **21st** century.

How does Open-Source Operating System work?

It works similarly to a closed operating system, except that the user may modify the source code of the program or application. There may be a difference in function even if there is no difference in performance.

For instance, the information is packed and stored in a proprietary (closed) operating system. In open-source, the same thing happens. However, because the source code is visible to you, you may better understand the process and change how data is processed.

While the former operating system is secure and hassle-free, and the latter requires some technical knowledge, you may customize these and increase performance. There is no specific way or framework for working on the open-source OS, but it may be customized on the user requirements.

Best Open-Source Operating System

Most of the open-source operating systems are Linux based. Some of the best open-source operating systems are as follows:

1. Linux Kernel

Linux kernel was developed by Linus Torvalds. It offers the essential functions required for an operating system, such as data cancellation, memory processing, and interactions with computer hardware. It is open-source software, and many developers researched the source code and produced a plethora of helpful plug-ins and operating systems to meet their requirements.

2. Linux Lite

Linux Lite is another free and open-source operating system that can run on lower-end hardware. It is a lightweight operating system designed to help users who are unfamiliar with Linux-based operating systems. The operating system includes all of the required programs, capabilities, tools, and desktops. It has a minimal interface and is entirely based on the Ubuntu system. In the last five years, the operating system has been stable and has received regular updates. It is efficiently functional soon after installation. After installation, users are not required to install any further drivers. If you want a lightweight open-source operating system on your PC, go with Linux Lite.

3. Linux mint

Linux Mint is a powerful Linux-based operating system that exudes modernity and power. It is simple to use and includes complete multimedia capabilities, making it a user-friendly open-source operating system. It is an Ubuntu-based distribution that is popular among both beginners and experts. It is built on the Debian platform and includes one of the most powerful software managers. It is more stable and has better visual aesthetics than Ubuntu.

4. Fedora

Fedora is another popular Linux-based operating system, and it is widely considered the best open-source operating system after Ubuntu. It is an RPM-based general-purpose operating system that is supported by Red Hat and built by the Fedora Project community. Its purpose is to develop and share cutting-edge open-source technology for free. As a result, Fedora developers prefer to make upstream improvements rather than create fixes specifically for Fedora. Fedora developers' updates are available to all Linux distributions.

It has a GNOME-based desktop that may be customized. Fedora comes with a customizable GNOME-based desktop. Its Fedora Spins feature allows you to customize and run several user interfaces and desktop environments.

5. React OS

ReactOS is another free and open-source operating system that has nearly 1 million downloads in over **100** countries. This community-based OS may run Windows apps, making it an excellent alternative to the Windows operating system. Although ReactOS is still growing, users, who love highly customizable operating systems, can select ReactOS. However, the operating system is developer-focused.

Advantages and Disadvantages of Open-Source Operating System

Various advantages and disadvantages of the open-source operating system are as follows:

Advantages

1. Reliable and efficient

The open-source operating systems are most reliable and efficient. Thousands of eyes monitor these because the source code is public. As a result, if there are any bugs or errors, they are fixed by the best developers worldwide.

2. Cost-efficient

Most of the open-source operating systems are free. And some of them are far less expensive than commercially closed products.

3. Flexibility

The great advantage is you may customize it as per your requirement. And there is creative freedom.

Disadvantages

1. Complicated

It is not as user-friendly as the ones that are closed. To use this software, you must have a basic understanding of technology.

2, Security risk

Despite the defects having been detected, there is a risk of assaults because the attackers have access to the source code.

3. No support

If you run across an issue, there is no customer support available to assist you.

Operating-System Services

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. 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. 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.

User and Operating-System Interface

in computers there are different interfaces. These interfaces are not necessarily used but can be used in computers whenever it is needed. So, different types of tasks can be performed by the help of different interfaces.

Command line interface

The command-line interface is an interface whenever the user needs to have different commands regarding the input and output and then a task is performed so this is called the command-line argument and it is used to execute the output and create, delete, print, copy, paste, etc.

All these operations are performed with the help of the command-line interface.

The interface is always connected to the OS so that the command given by the user directly works by the OS and a number of operations can be performed with the help of the command line interface because multiple commands can be interrupted at same time and execute only one.

The command line interface is necessary because all the basic operations in the computer are performed with the help of the OS and it is responsible for memory management. By using this we can divide the memory and we can use the memory.

Command Line Interface advantages −

- Controls OS or application
- faster management
- ability to store scripts which helps in automating regular tasks.
- Troubleshoot network connection issues.

Command Line Interface disadvantages −

- The steeper learning curve is associated with memorizing commands and a complex syntax.
- Different commands are used in different shells.

Graphical user interface

The graphical user interface is used for playing games, watching videos, etc. these are done with the help of GUI because all these applications require graphics.

The GUI is one of the necessary interfaces because only by using the user can clearly see the picture, play videos.

So we need GUI for computers and this can be done only with the help of an operating system.

When a task is performed in the computer then the OS checks the task and defines the interface which is necessary for the task. So, we need GUI in the OS.

The basic components of GUIs are −

- Start menu with program groups
- Taskbar which showing running programs
- Desktop screen
- Different icons and shortcuts.

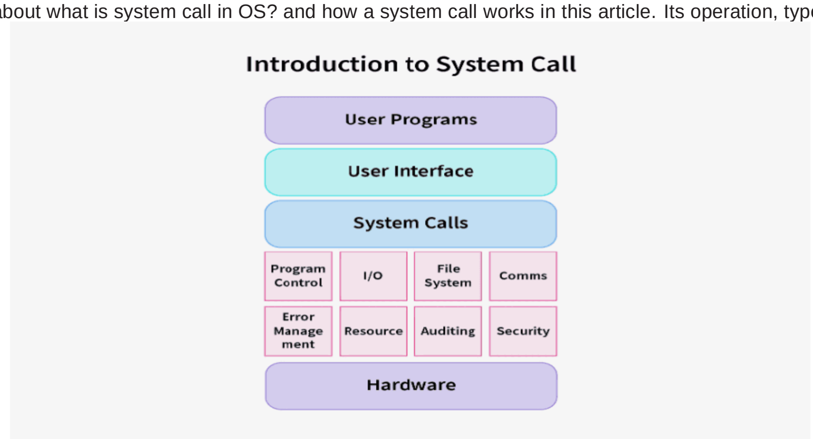
Choice of interface

The interface that is used with the help of OS for a particular task and that task can be performed with minimum possible time and the output is shown on the screen in that case we use the choice of interface.

The choice of interface means the OS checks the task and finds out which interface can be suitable for a particular task. So that type of interface is called the choice of interface and this can be done with the help of an OS.

System Calls

System calls serve as the interface between an operating system and a process. System calls can typically be found as assembly language instructions. They are also covered in the manuals that the programmers working at the assembly level use. When a process in user mode needs access to a resource, system calls are typically generated. The resource is then requested from the kernel via a system call. System calls are always carried out in heoperating system's kernel mode. We will talk about what is system call in OS? and how a system call works in this article. Its operation, types, and how



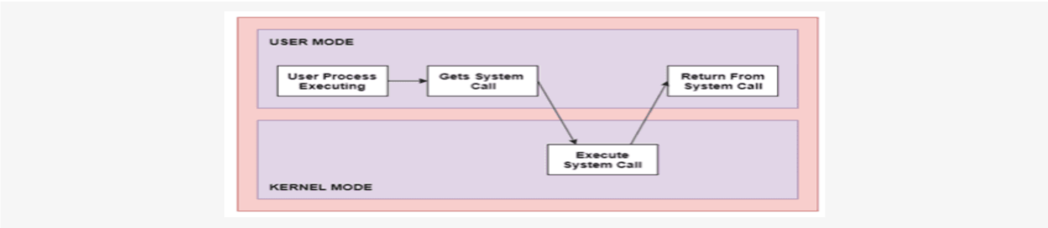
to pass parameters to such calls will be covered.

What is a System Call in Operating System (OS)?

A system calls in computing is the programmatic method by which a computer programme asks the kernel of the operating system it is running on for a service. Programs can interact with the operating system by making a system call. When a computer programme requests something from the kernel of the operating system, it performs a system call.

System call uses Application Programming Interfaces to provide operating system services to user programmes (API). It offers a point of contact between processes and the operating system so that user-level processes can ask for the operating system's assistance. The only ways to access the kernel system are through system calls. System calls are required for all programmes that require resources.

The following is a diagram that illustrates what is system call in an operating system and how the system call is executed:



This diagram shows how the processes normally run in user mode until a system call interrupts them. The system call is then carried out in kernel mode according to priority. The control switches back to user mode after the system call has been completed, allowing user processes to continue running.

Example of System Calls

Following are the examples of system calls:

The most common system calls used on Unix system calls, Unix-like, and other POSIX-compliant operating systems are open, read, write, close, wait, exec, fork, exit, and kill. There are hundreds of systems calls in many contemporary operating systems. For instance, Linux and OpenBSD each have over 300 different calls, NetBSD and FreeBSD both have over 500, and Windows has close to 2000 system calls, split between win32k and ntdll.

Tools like strace, ftrace, and truss allow a process to run from the beginning and report every system call it makes. They can also attach to an already running process and intercept any system call made by that process if the action does not conflict with the user's permissions.

If we need to copy data from one file into another, we can do so by writing programme code. The names of the two files, the input and output files, are the first piece of data that the programme needs. This kind of programme execution requires some system calls from the OS in an interactive system.

- Writing a prompting message on the screen comes first.
- Second, read the characters that define the two files from the keyboard.

Types of System Calls

Types of System Calls in OS

There are primarily 5 different types of system calls in the operating system:

1. Process Control

The following services are provided by Process Control System calls that are used to control a process.

- To forcefully abort the process, simply end it normally.
- Execute a process after loading it into the main memory.
- Terminate the current process before starting a new one.
- Wait for a process to complete running. Wait until a specific event happens, then announce it once it has.
- Allocate memory to a process, then release the memory if the process is terminated.

2. File Management

The following services are provided by system call for file management:

- Making and erasing files
- Open the file, then close it.
- Write to a specific file, read from a specific file.
- To obtain a file's attribute and to change a file's attribute

3. Device Management

The following services are offered by a system call that controls I/O devices:

- Devices might be needed while a process is running. such as access to the file system, I/O devices, main memory, etc.
- As a result, it can ask for a device and then release it once the task is complete.
- when a requested device is granted access by the process. It is capable of reading, writing, and repositioning operations.
- In order to obtain or modify a specific device's attribute.
- To detach a device from the processor that is currently executing a command, the call can be made.

4. Information Maintenance

The system calls for information maintenance call moves data from the user programme to the operating system. Considering this, the services offered by this type of system call are:

- Obtain the system's time or date. Set the system's time or date.
- Obtain system-related information. Configure the system data.
- Obtain the characteristics of a specific operating system process. Alternatively, of a specific file on the system or on any attached devices.
- Set the characteristics of a specific operating system process. Alternatively, of a specific file on the system or on any attached devices.

5. Communication

Such a system call facilitates the system's network connection. The services that these system calls offer are:

- Open a fresh connection to send the data. After the transmission is finished, disconnect from the connection.
- On a particular connection, send a message. Obtain communication from a specific connection.
- Identify and connect a specific remote device to the network. Remove a specific remote computer or device from the network.

Example of System Calls in Windows and Unix

Types of System Calls	Windows	Linux
Process Control	**CreateProcess()** **ExitProcess()** **WaitForSingleObject()**	**fork()** **exit()** **wait()**
File Management	**CreateFile()** **ReadFile()** **WriteFile()** **CloseHandle()**	**open()** **read()** **write()** **close()**
Device Management	**SetConsoleMode()** **ReadConsole()** **WriteConsole()**	**ioctl()** **read()** **write()**

Types of System Calls	Windows	Linux
Information Maintenance	**GetCurrentProcessID()** **SetTimer()** **Sleep()**	**getpid()** **alarm()** **sleep()**
Communication	**CreatePipe()** **CreateFileMapping()** **MapViewOfFile()**	**pipe()** **shmget()** **mmap()**

Rules for Passing Parameters to the System Call

Following are the **rules for passing parameters to the system call**:

- The system call cannot accept the floating-point parameters as a parameter.
- In the system call, a finite number of arguments may be passed.
- If there are additional arguments, they should be stored in the memory block, and the register contains the address of that memory block.
- Only the operating system has the ability to push parameters and pop them from the stack.

Important System Calls Used in OS

1. wait()

In certain systems, a process must wait for another process to finish running before continuing. When a parent process creates a child process, the parent process's execution is suspended until the child process has finished running.

With a wait() system call, the parent process is automatically suspended. Control returns to the parent process once the child process has completed its execution.

2. fork()

The fork system call in OS is used by processes to make copies of themselves. By using this system, the Call parent process can create a child process, and the parent process's execution will be halted while the child process runs.

3. exec()

When an executable file replaces an earlier executable file within the context of an active process, this system call is executed. The original process identifier is still present even though a new process is not created; instead, the new process replaces the old one's stack, data, head, data, etc.

4. kill()

The OS uses the kill() system call to urge processes to terminate by sending them a signal. A kill system call can have different meanings and is not always used to end a process.

5. exit()

An exit system call is used when the programme must be stopped. When the exit system call is used, the resources that the process was using were released.

Why Do You Need System Calls in Operating System?

The following circumstances involve the use of system calls in OS:

- System calls are necessary for reading and writing from files.
- System calls are necessary for a file system to add or remove files.
- New processes are created and managed using system calls.
- System calls are required for packet sending and receiving over network connections.
- A system call is required to access hardware devices like scanners and printers.

What is the Purpose of System Calls in OS?

The purpose of system calls serves as the interface between an operating system and a process. System calls can typically be found as assembly language instructions. They are also covered in the manuals that the programmers working at the assembly level use. When a process in user mode needs access to a resource, system calls are typically generated. The resource is then requested from the kernel via a system call.

System calls are required in in following circumstances:

- If a file system needs files to be created or deleted. A system call is also necessary for reading and writing from files.
- development and administration of new procedures.
- System calls are also required for network connections. This also applies to packet sending and receiving.
- A system call is necessary to access hardware such as a printer, scanner, etc.

Conclusion

The interface by which the process interacts with the system call is called a system call. There are two operating modes for computers:

- kernel mode
- user mode

When a system call is made, the process transitions from user mode to kernel mode. Once the system call has finished running, the control is returned to the user mode process. Sending the kernel, a trap signal causes it to read the system call code from the register and carry out the system call. Process control, file management system calls in OS, device management, information maintenance, and communication are the main types of system calls in OS. Several crucial system calls used by our computer system include wait(), fork(), exec(), kill(), and exit(). Enrol in KnowledgeHut's Full Stack Web Developer Course to learn the full stack developer skill set.

System Programs

The system programs are responsible for the development and execution of a program and they can be used by the help of system calls because system calls define different types of system programs for different tasks.

- **File management** − These programs create, delete, copy, rename, print, exit and generally manipulate the files and directory.
- **Status information** − It is the information regarding input, output process, storage and the CPU utilization time how the process will be calculated in how much memory required to perform a task is known by status information.
- **Programming language supports** − compiler, assembler, interrupt are programming language support used in the operating system for a particular purpose in the computer.
- **Programming Loading and execution** − The needs to enter the program and after the loading of a program it needs to execute the output of the programs and this task is also performed by system calls by the help of system programs.
- **Communication** − These services are provided by the user because by using this number of devices communicates with each other by the help of device or wireless and communication is necessary for the operating system.
- **Background services** − There are different types of services available on the operating system for communication and background service is used to change the background of your window and it also works for scanning and detecting viruses in the computer.

Purpose of using system program

System programs communicate and coordinate the activities and functions of hardware and software of a system and also controls the operations of the hardware. An operating system is one of the examples of system software. Operating system controls the computer hardware and acts like an interface between the application software's.

Types of System programs

The types of system programs are as follows −

Utility program

It manages, maintains and controls various computer resources. Utility programs are comparatively technical and are targeted for the users with solid technical knowledge.

Few examples of utility programs are: antivirus software, backup software and disk tools.

Device drivers

It controls the particular device connected to a computer system. Device drivers basically act as a translator between the operating system and device connected to the system.

Example − printer driver, scanner driver, storage device driver etc.

Directory reporting tools

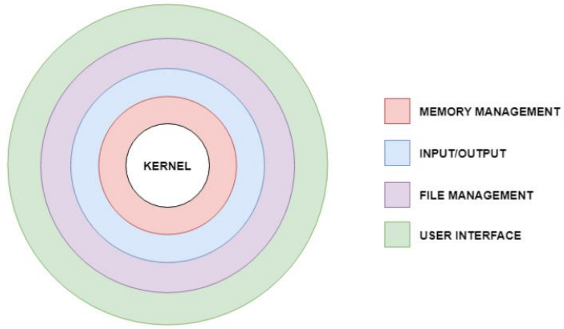
These tools are required in an operation system to have some software to facilitate the navigation through the computer system.

Example − dir, ls, Windows Explorer etc.

Operating-System Design and Implementation

An operating system is a construct that allows the user application programs to interact with the system hardware. Operating system by itself does not provide any function but it provides an atmosphere in which different applications and programs can do useful work.

There are many problems that can occur while designing and implementing an operating system. These are covered in operating system design and implementation.



Layered Operating System Design

Operating System Design Goals

It is quite complicated to define all the goals and specifications of the operating system while designing it.The design changes depending on the type of the operating system i.e if it is batch system, time shared system, single user system, multi user system, distributed system etc.

There are basically two types of goals while designing an operating system. These are −

User Goals

The operating system should be convenient, easy to use, reliable, safe and fast according to the users. However, these specifications are not very useful as there is no set method to achieve these goals.

System Goals

The operating system should be easy to design, implement and maintain. These are specifications required by those who create, maintain and operate the operating system. But there is not specific method to achieve these goals as well.

Operating System Mechanisms and Policies

There is no specific way to design an operating system as it is a highly creative task. However, there are general software principles that are applicable to all operating systems.

A subtle difference between mechanism and policy is that mechanism shows how to do something and policy shows what to do. Policies may change over time and this would lead to changes in mechanism. So, it is better to have a general mechanism that would require few changes even when a policy change occurs.

For example - If the mechanism and policy are independent, then few changes are required in mechanism if policy changes. If a policy favours I/O intensive processes over CPU intensive processes, then a policy change to preference of CPU intensive processes will not change the mechanism.

Operating System Implementation

The operating system needs to be implemented after it is designed. Earlier they were written in assembly language but now higher level languages are used. The first system not written in assembly language was the Master Control Program (MCP) for Burroughs Computers.

Advantages of Higher Level Language

There are multiple advantages to implementing an operating system using a higher level language such as: the code is written more fast, it is compact and also easier to debug and understand. Also, the operating system can be easily moved from one hardware to another if it is written in a high level language.

Disadvantages of Higher Level Language

Using high level language for implementing an operating system leads to a loss in speed and increase in storage requirements. However in modern systems only a small amount of code is needed for high performance, such as the CPU scheduler and memory manager. Also, the bottleneck routines in the system can be replaced by assembly language equivalents if required.

Operating-System Structure

What is an operating System Structure?

We want a clear structure to let us apply an operating system to our particular needs because operating systems have complex structures. It is easier to create an operating system in pieces, much as we break down larger issues into smaller, more manageable subproblems. Every segment is also a part of the operating system. Operating system structure can be thought of as the strategy for connecting and incorporating various operating system components within the kernel. Operating systems are implemented using many types of structures, as will be discussed below:

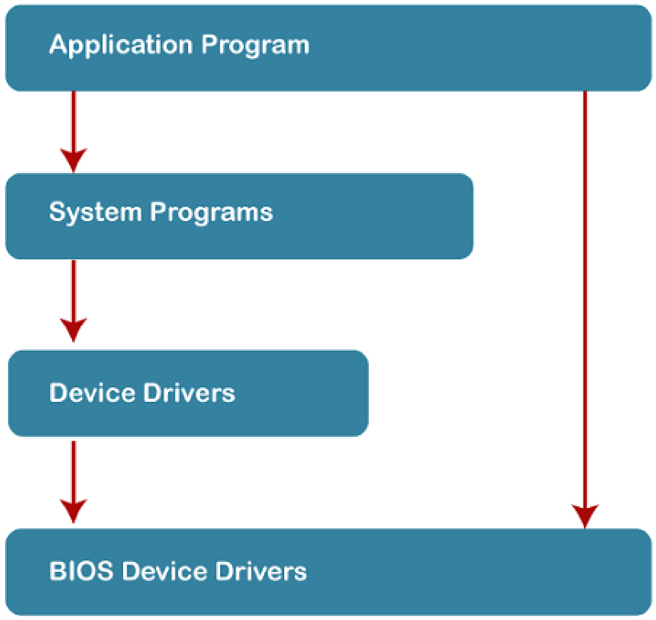
SIMPLE STRUCTURE

It is the most straightforward operating system structure, but it lacks definition and is only appropriate for usage with tiny and restricted systems. Since the interfaces and degrees of functionality in this structure are clearly defined, programs are able to access I/O routines, which may result in unauthorized access to I/O procedures.

This organizational structure is used by the MS-DOS operating system:

- There are four layers that make up the MS-DOS operating system, and each has its own set of features.
- These layers include ROM BIOS device drivers, MS-DOS device drivers, application programs, and system programs.
- The MS-DOS operating system benefits from layering because each level can be defined independently and, when necessary, can interact with one another.
- If the system is built in layers, it will be simpler to design, manage, and update. Because of this, simple structures can be used to build constrained systems that are less complex.
- When a user program fails, the operating system as whole crashes.
- Because MS-DOS systems have a low level of abstraction, programs and I/O procedures are visible to end users, giving them the potential for unwanted access.

The following figure illustrates layering in simple structure:



Advantages of Simple Structure:

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

Disadvantages of Simple Structure:

- The entire operating system breaks if just one user program malfunctions.
- Since the layers are interconnected, and in communication with one another, there is no abstraction or data hiding.
- The operating system's operations are accessible to layers, which can result in data tampering and system failure.

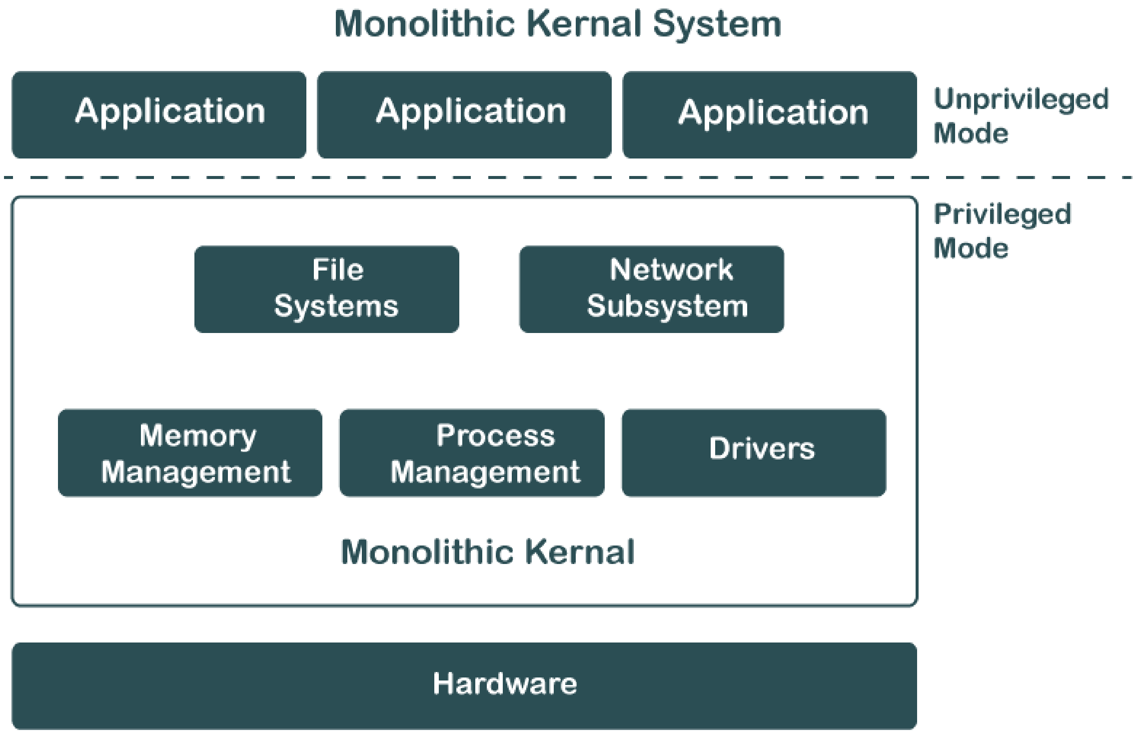
MONOLITHIC STRUCTURE

The monolithic operating system controls all aspects of the operating system's operation, including file management, memory management, device management, and operational operations.

The core of an operating system for computers is called the kernel (OS). All other System components are provided with fundamental services by the kernel. The operating system and the hardware use it as their main interface. When an operating system is built into a single piece of hardware, such as a keyboard or mouse, the kernel can directly access all of its resources.

The monolithic operating system is often referred to as the monolithic kernel. Multiple programming techniques such as batch processing and time-sharing increase a processor's usability. Working on top of the operating system and under complete command of all hardware, the monolithic kernel performs the role of a virtual computer. This is an old operating system that was used in banks to carry out simple tasks like batch processing and time-sharing, which allows numerous users at different terminals to access the Operating System.

The following diagram represents the monolithic structure:



Advantages of Monolithic Structure:

- Because layering is unnecessary and the kernel alone is responsible for managing all operations, it is easy to design and execute.
- Due to the fact that functions like memory management, file management, process scheduling, etc., are implemented in the same address area, the monolithic kernel runs rather quickly when compared to other systems. Utilizing the same address speeds up and reduces the time required for address allocation for new processes.

Disadvantages of Monolithic Structure:

- The monolithic kernel's services are interconnected in address space and have an impact on one another, so if any of them malfunctions, the entire system does as well.

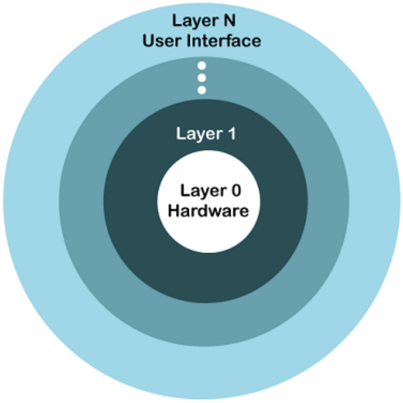
- It is not adaptable. Therefore, launching a new service is difficult.

LAYERED STRUCTURE

The OS is separated into layers or levels in this kind of arrangement. Layer 0 (the lowest layer) contains the hardware, and layer 1 (the highest layer) contains the user interface (layer N). These layers are organized hierarchically, with the top-level layers making use of the capabilities of the lower-level ones.

The functionalities of each layer are separated in this method, and abstraction is also an option. Because layered structures are hierarchical, debugging is simpler, therefore all lower-level layers are debugged before the upper layer is examined. As a result, the present layer alone has to be reviewed since all the lower layers have already been examined.

The image below shows how OS is organized into layers:



Advantages of Layered Structure:

- Work duties are separated since each layer has its own functionality, and there is some amount of abstraction.
- Debugging is simpler because the lower layers are examined first, followed by the top layers.

Disadvantages of Layered Structure:

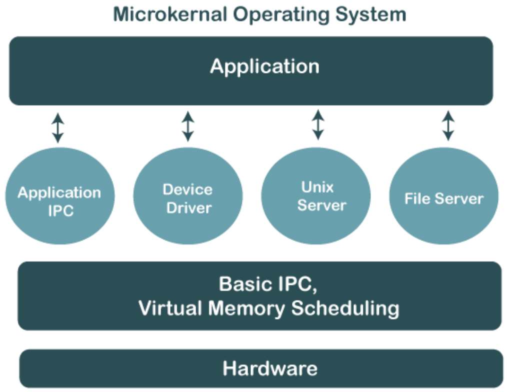
- Performance is compromised in layered structures due to layering.
- Construction of the layers requires careful design because upper layers only make use of lower layers' capabilities.

MICRO-KERNEL STRUCTURE

The operating system is created using a micro-kernel framework that strips the kernel of any unnecessary parts. Systems and user applications are used to implement these optional kernel components. So, Micro-Kernels is the name given to these systems that have been developed.

Each Micro-Kernel is created separately and is kept apart from the others. As a result, the system is now more trustworthy and secure. If one Micro-Kernel malfunctions, the remaining operating system is unaffected and continues to function normally.

The image below shows Micro-Kernel Operating System Structure:



Advantages of Micro-Kernel Structure:

- It enables portability of the operating system across platforms.
- Due to the isolation of each Micro-Kernel, it is reliable and secure.
- The reduced size of Micro-Kernels allows for successful testing.
- The remaining operating system remains unaffected and keeps running properly even if a component or Micro-Kernel fails.

Disadvantages of Micro-Kernel Structure:

- The performance of the system is decreased by increased inter-module communication.
- The construction of a system is complicated.

EXOKERNEL

An operating system called Exokernel was created at MIT with the goal of offering application-level management of hardware resources. The exokernel architecture's goal is to enable application-specific customization by separating resource management from protection. Exokernel size tends to be minimal due to its limited operability.

Operating-System Debugging

Debugging is the process of finding the problems in a computer system and solving them. There are many different ways in which operating systems perform debugging. Some of these are −

Log Files

The log files record all the events that occur in an operating system. This is done by writing all the messages into a log file. There are different types of log files. Some of these are given as follows −

Event Logs

These stores the records of all the events that occur in the execution of a system. This is done so that the activities of all the events can be understood to diagnose problems.

Transaction Logs

The transaction logs store the changes to the data so that the system can recover from crashes and other errors. These logs are readable by a human.

Message Logs

These logs store both the public and private messages between the users. They are mostly plain text files, but in some cases they may be HTML files.

Core Dump Files

The core dump files contain the memory address space of a process that terminates unexpectedly. The creation of the core dump is triggered in response to program crashes by the kernel. The core dump files are used by the developers to find the program’s state at the time of its termination so that they can find out why the termination occurred.

The automatic creation of the core dump files can be disabled by the users. This may be done to improve performance, clear disk space or increase security.

Crash Dump Files

In the event of a total system failure, the information about the state of the operating system is captured in crash dump files. There are three types of dump that can be captured when a system crashes. These are −

Complete Memory Dump

The whole contents of the physical memory at the time of the system crash are captured in the complete memory dump. This is the default setting on the Windows Server System.

Kernel Memory Dump

Only the kernel mode read and write pages that are present in the main memory at the time of the system crash are stored in the kernel memory dump.

Small Memory Dump

This memory dump contains the list of device drivers, stop code, process and thread information, kernel stack etc.

Trace Listings

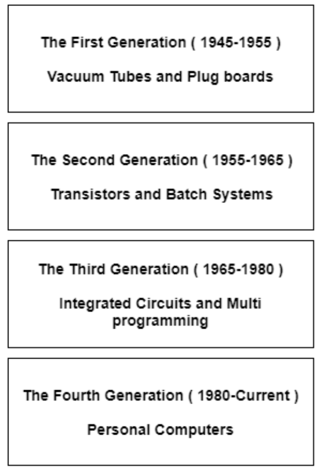
The trace listing record information about a program execution using logging. This information is used by programmers for debugging. System administrators and technical personnel can use the trace listings to find the common problems with software using software monitoring tools.

Profiling

This is a type of program analysis that measures various parameters in a program such as space and time complexity, frequency and duration of function calls, usage of specific instructions etc. Profiling is done by monitoring the source code of the required system program using a code profiler.

Operating-System Generation

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. These can be described as follows −



OPERATING SYSTEM GENERATIONS

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.

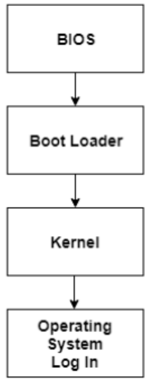
The advent of personal computers also led to the growth of networks. This created network operating systems and distributed operating systems. The users were aware of a network while using a network operating system and could log in to remote machines and copy files from one machine to another

System Boot

he BIOS, operating system and hardware components of a computer system should all be working correctly for it to boot. If any of these elements fail, it leads to a failed boot sequence.

System Boot Process

The following diagram demonstrates the steps involved in a system boot process −



Here are the steps −

- The CPU initializes itself after the power in the computer is first turned on. This is done by triggering a series of clock ticks that are generated by the system clock.
- After this, the CPU looks for the system's ROM BIOS to obtain the first instruction in the start-up program. This first instruction is stored in the ROM BIOS and it instructs the system to run POST (Power On Self Test) in a memory address that is predetermined.
- POST first checks the BIOS chip and then the CMOS RAM. If there is no battery failure detected by POST, then it continues to initialize the CPU.
- POST also checks the hardware devices, secondary storage devices such as hard drives, ports etc. And other hardware devices such as the mouse and keyboard. This is done to make sure they are working properly.
- After POST makes sure that all the components are working properly, then the BIOS finds an operating system to load.
- In most computer system's, the operating system loads from the C drive onto the hard drive. The CMOS chip typically tells the BIOS where the operating system is found.
- The order of the different drives that CMOS looks at while finding the operating system is known as the boot sequence. This sequence can be changed by changing the CMOS setup.
- After finding the appropriate boot drive, the BIOS first finds the boot record which tells it to find the beginning of the operating system.
- After the initialization of the operating system, the BIOS copies the files into the memory. Then the operating system controls the boot process.
- In the end, the operating system does a final inventory of the system memory and loads the device drivers needed to control the peripheral devices.
- The users can access the system applications to perform various tasks.

Without the system boot process, the computer users would have to download all the software components, including the ones not frequently required. With the system boot, only those software components need to be downloaded that are legitimately required and all extraneous components are not required. This process frees up a lot of space in the memory and consequently saves a lot of time.