

UNIT 1: FUNDAMENTAL CONCEPTS OF OPERATING SYSTEMS

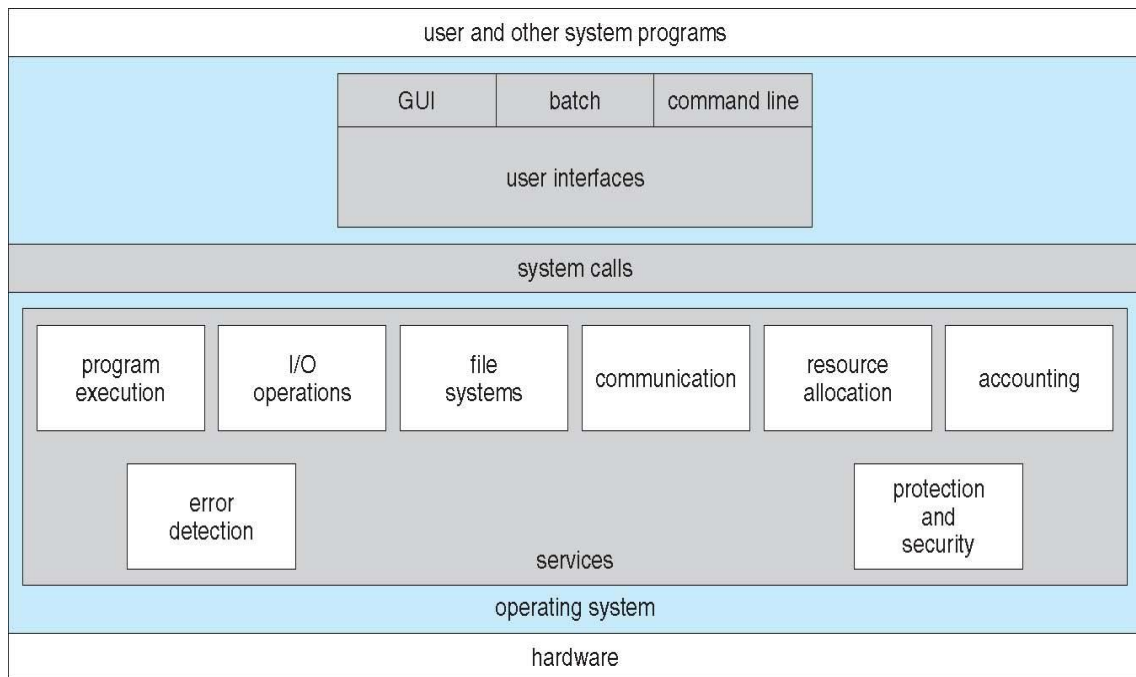
Operating systems services:

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

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

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

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



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**. 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.**

Some services exist not for helping the user but rather for ensuring the efficient operation of the system itself...

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 and Security

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.**

Accounting

- **To keep the track of users with the use of resources**
- **Simply used for accumulating usage statistic**

Types of Operating System (OS)

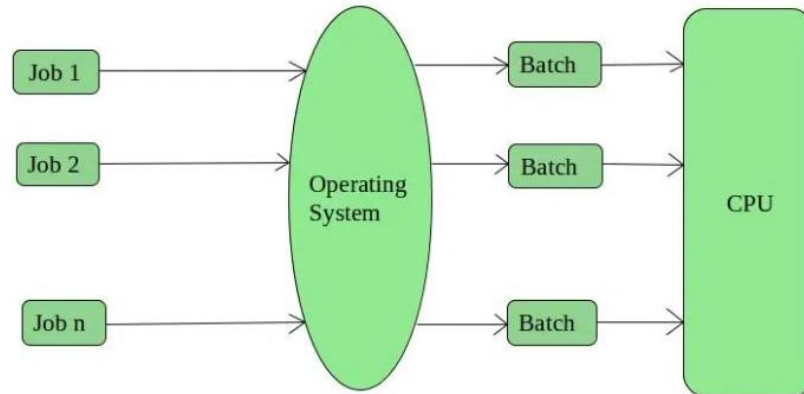
Following are the popular types of OS (Operating System):

- **Batch Operating System**

- Multitasking/Time Sharing OS
- Real Time OS
- Distributed OS
- Network OS

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 requirement and groups them into batches. It is the responsibility of the operator to sort jobs with similar needs.



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.

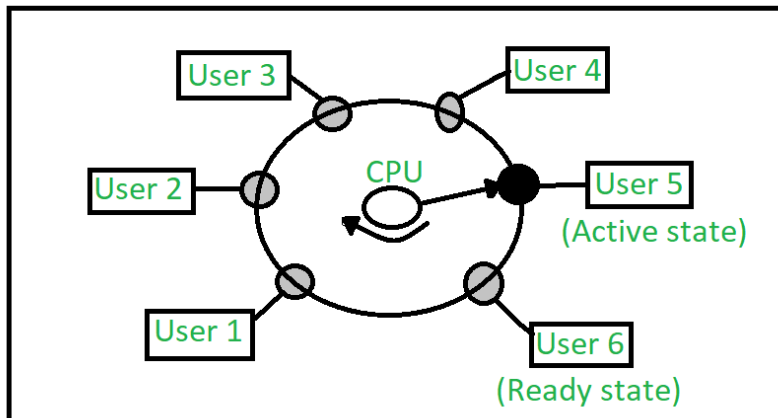
The other jobs will have to wait for an unknown time if any job fails.

Examples of Batch Operating Systems: **Payroll Systems**, Bank Statements, etc.

2. Time-sharing Operating systems

Time-sharing operating system enables people located at a different terminal (shell) to use a single computer system at the same time. **The processor time (CPU) which is shared among multiple users is termed as time sharing.** Multitasking Operating System is simply a multiprogramming Operating System with having facility of a Round-Robin Scheduling

Algorithm. It can run multiple programs simultaneously. **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.



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.

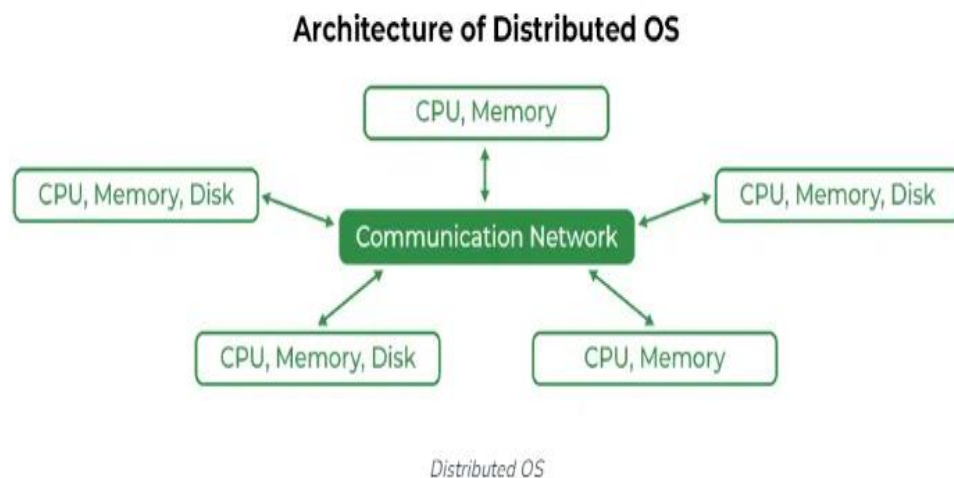
Examples of Time-Sharing OS with explanation

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.

3. Distributed Operating System

These types of operating system are 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.



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.

The distributed os must tackle the following issues:

Networking causes delays in the transfer of data between nodes of a distributed system.

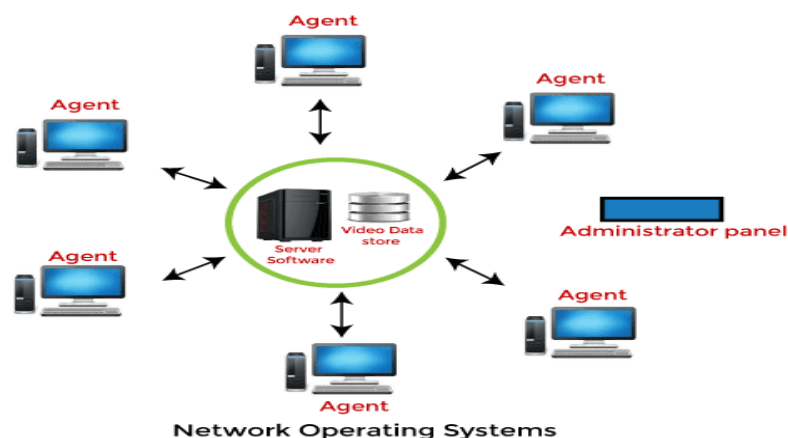
Such delays may lead to an inconsistent view of data located in different nodes, and make it difficult to know the chronological order in which events occurred in the system.

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.

4. 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.



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

5. 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..

Soft Real-Time Systems:

These OSs are for applications where time-constraint is less strict.

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.

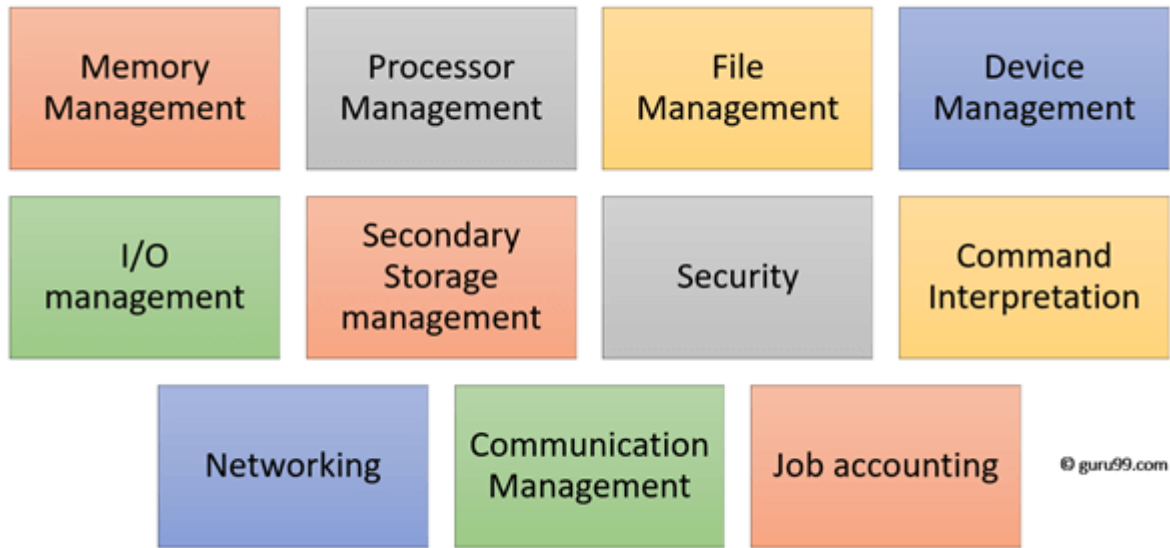
Thread Priority: It is not good to set thread priority as these systems are very less prone to switching tasks.

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

Functions of Operating System

Some typical operating system functions may include managing memory, files, processes, I/O system & devices, security, etc.

Below are the main functions of Operating System:



Functions of Operating System

In an operating system software performs each of the function:

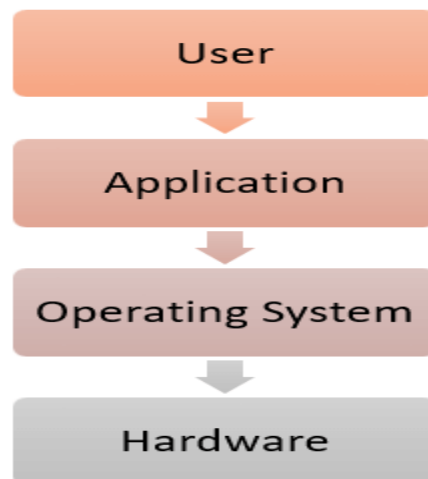
1. **Process management:** Process management helps OS to create and delete processes. It also provides mechanisms for synchronization and communication among processes.
2. **Memory management:** Memory management module performs the task of allocation and de-allocation of memory space to programs in need of this resources.
3. **File management:** It manages all the file-related activities such as organization storage, retrieval, naming, sharing, and protection of files.
4. **Device Management:** Device management keeps tracks of all devices. This module also responsible for this task is known as the I/O controller. It also performs the task of allocation and de-allocation of the devices.
5. **I/O System Management:** One of the main objects of any OS is to hide the peculiarities of that hardware devices from the user.
6. **Secondary-Storage Management:** Systems have several levels of storage which includes primary storage, secondary storage, and cache storage. Instructions and data must be stored in primary storage or cache so that a running program can reference it.
7. **Security:** Security module protects the data and information of a computer system against malware threat and authorized access.
8. **Command interpretation:** This module is interpreting commands given by the and acting system resources to process that commands.

9. **Networking:** A distributed system is a group of processors which do not share memory, hardware devices, or a clock. The processors communicate with one another through the network.
10. **Job accounting:** Keeping track of time & resource used by various job and users.
11. **Communication management:** Coordination and assignment of compilers, interpreters, and another software resource of the various users of the computer systems.

Features of Operating System (OS)

Here is a list of important features of OS:

- Protected and supervisor mode
- Allows disk access and file systems Device Drivers
- Networking Security
- Program Execution
- Memory management Virtual Memory Multitasking
- Handling I/O operations
- Manipulation of the file system
- Error Detection and handling
- Resource allocation
- Information and Resource Protection



Advantage of Operating System

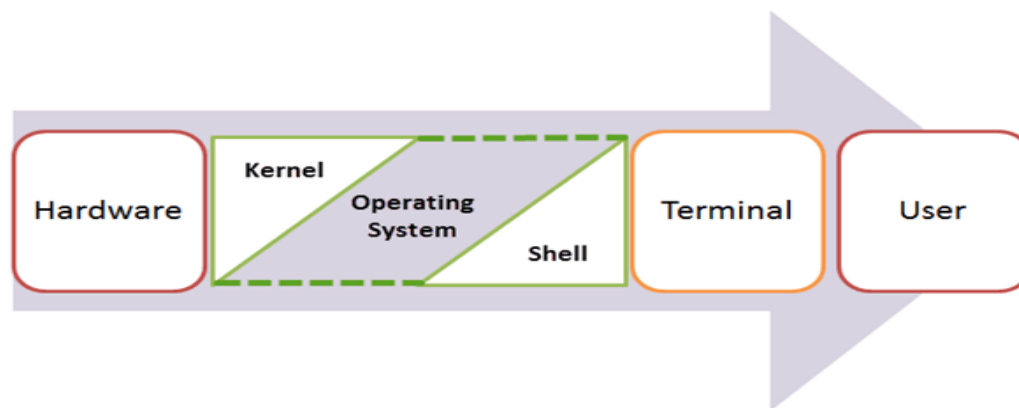
- Allows you to hide details of hardware by creating an abstraction
- Easy to use with a GUI
- Offers an environment in which a user may execute programs/applications
- The operating system must make sure that the computer system convenient to use
- Operating System acts as an intermediary among applications and the hardware components
- It provides the computer system resources with easy to use format
- Acts as an intermediary between all hardware's and software's of the system

Disadvantages of Operating System

- If any issue occurs in OS, you may lose all the contents which have been stored in your system
- Operating system's software is quite expensive for small size organization which adds burden on them. Example Windows
- It is never entirely secure as a threat can occur at any time

What is Kernel in Operating System?

The kernel is the central component of a computer operating systems. The only job performed by the kernel is to manage the communication between the software and the hardware. A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.



Introduction to Kernel

Features of Kernel

- Low-level scheduling of processes
- Inter-process communication
- Process synchronization
- Context switching

Types of Kernel

There are many types of kernels that exist, but among them, the two most popular kernels are:

1. Monolithic

A monolithic kernel is a single code or block of the program. It provides all the required services offered by the operating system. It is a simplistic design which creates a distinct communication layer between the hardware and software.

2. Microkernels

Microkernel manages all system resources. In this type of kernel, services are implemented in different address space. The user services are stored in user address space, and kernel services are stored under kernel address space. So, it helps to reduce the size of both the kernel and operating system.

Difference between Firmware and Operating System

Below are the Key Differences between Firmware and Operating System:

Firmware	Operating System
Define Firmware: Firmware is one kind of programming that is embedded on a chip in the device which controls that specific device.	Define Operating System: OS provides functionality over and above that which is provided by the firmware.
Firmware is programs that been encoded by the manufacture of the IC or something and cannot be changed.	OS is a program that can be installed by the user and can be changed.
It is stored on non-volatile memory.	OS is stored on the hard drive.

Difference between 32-Bit and 64-Bit Operating System

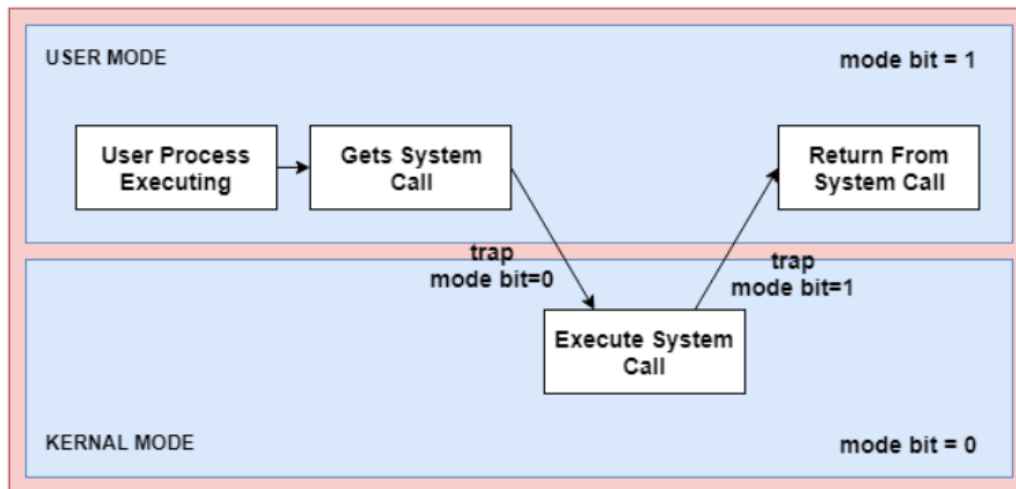
Below are the Key Differences between 32-Bit and 64-Bit Operating System:

Parameters	32. Bit	64. Bit
Architecture and Software	Allow 32 bit of data processing simultaneously	Allow 64 bit of data processing simultaneously
Compatibility	32-bit applications require 32-bit OS and CPUs.	64-bit applications require a 64-bit OS and CPU.
Systems Available	All versions of Windows 8, Windows 7, Windows Vista, and Windows XP, Linux, etc.	Windows XP Professional, Vista, 7, Mac OS X and Linux.
Memory Limits	32-bit systems are limited to 3.2 GB of RAM.	64-bit systems allow a maximum 17 Billion GB of RAM.

Dual-Mode and Multi-Mode Operation in Operating System

There are two modes of operation in the operating system to make sure it works correctly. These are user mode and kernel mode.

A diagram that illustrates the transition from user mode to kernel mode and back again is as follows



User Mode:

The system is in user mode when the operating system is running a user application such as handling a text editor. The transition from user mode to kernel mode occurs when the application requests the help of operating system or an interrupt or a system call occurs.

The mode bit is set to 1 in the user mode. It is changed from 1 to 0 when switching from user mode to kernel mode.

Kernel Mode:

The system starts in kernel mode when it boots and after the operating system is loaded, it executes applications in user mode. There are some privileged instructions that can only be executed in kernel mode. These are interrupt instructions, input output management etc. If the privileged instructions are executed in user mode, it is illegal and a trap is generated.

The mode bit is set to 0 in the kernel mode. It is changed from 0 to 1 when switching from kernel mode to user mode.

The concept of modes of operation in operating system can be extended beyond the dual mode. This is known as the multimode system. In those cases, the more than 1 bit is used by the CPU to set and handle the mode.

An example of the multimode system can be described by the systems that support virtualisation. These CPU's have a separate mode that specifies when the virtual machine manager (VMM) and the virtualisation management software is in control of the system.

For these systems, **the virtual mode has more privileges than user mode but less than kernel mode.**

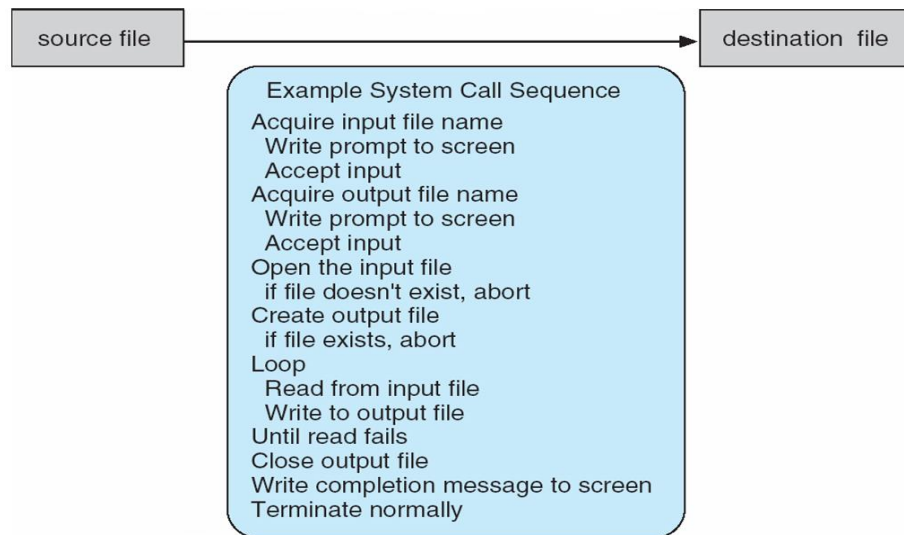
What is the need of dual and multimode operations in an OS.?

Without the mode bits' user level processes can operate the kernel level processes and they can overwrite the data present at the kernel level. So, to avoid this interference of both the processes into each other's level & to maintain the safety and protection of user and kernel level processes

mode bits are assigned to distinguish among each level. All modern versions of OS opt for dual mode operation.

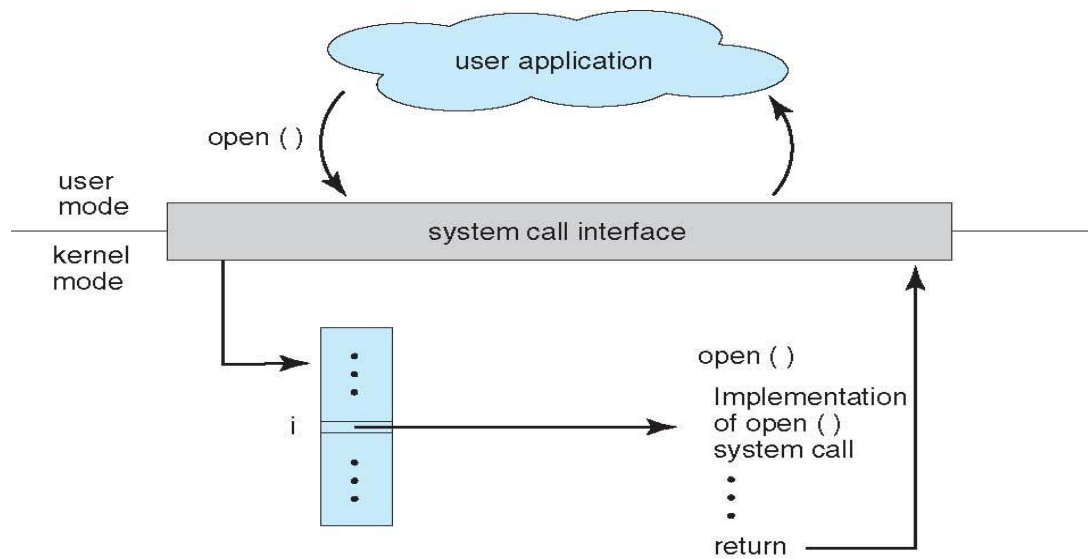
Systems Calls:

System call sequence to copy the contents of one file to another file (page 61-Galvin)



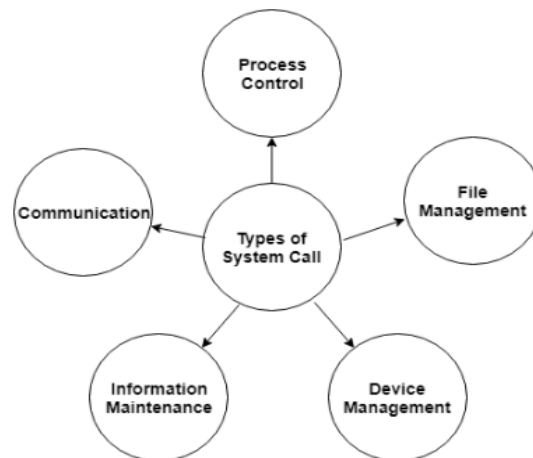
The interface between a process and an **operating system** is provided by **system calls**. In general, system calls are available as **assembly language** instructions. They are also included in the manuals used by the assembly level programmers. System calls are usually made when a process in user mode requires access to a resource. Then it requests the kernel to provide the resource via a system call.

System Call Implementation: Handling of user application invoking the open() system call (Page 63- Galvin)



Types of System Calls

There are mainly five types of system calls. These are explained in detail as follows –



Here are the types of system calls –

Process Control

These system calls deal with processes such as process creation, process termination etc.

File Management

These system calls are responsible for file manipulation such as creating a file, reading a file, writing into a file etc.

Device Management

These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.

Information Maintenance

These system calls handle information and its transfer between the operating system and the user program.

Communication

These system calls are useful for **interprocess communication**. They also deal with creating and deleting a communication connection.

Some of the examples of all the above types of system calls in Windows and Unix are given as follows –

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()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()

There are many different system calls as shown above. Details of some of those system calls are as follows –

`wait()`

In some systems, a process may wait for another process to complete its execution. This happens when a parent process creates a child process and the execution of the parent process is suspended until the child process executes. The suspending of the parent process occurs with a `wait()` system call. When the child process completes execution, the control is returned back to the parent process.

`exec()`

This system call runs an executable file in the context of an already running process. It replaces the previous executable file. This is known as an overlay. The original process identifier remains since a new process is not created but data, heap, stack etc. of the process are replaced by the new process.

`fork()`

Processes use the `fork()` system call to create processes that are a copy of themselves. This is one of the major methods of process creation in operating systems. When a parent process creates a child process and the execution of the parent process is suspended until the child process executes. When the child process completes execution, the control is returned back to the parent process.

`exit()`

The `exit()` system call is used by a program to terminate its execution. In a multithreaded environment, this means that the thread execution is complete. The operating system reclaims resources that were used by the process after the `exit()` system call.

`kill()`

The `kill()` system call is used by the operating system to send a termination signal to a process that urges the process to exit. However, `kill` system call does not necessarily mean killing the process and can have various meanings.