OPERATING SYSTEM

An **Operating System** acts as a communication bridge (interface) between the user and computer hardware. The purpose of an operating system is to provide a platform on which a user can execute programs in a convenient and efficient manner.
An operating system is a piece of software that manages the allocation of computer hardware. The coordination of the hardware must be appropriate to ensure the correct working of the computer system and to prevent user programs from interfering with the proper working of the system.
Example: Just like a boss gives order to his employee, in the similar way we request or pass our orders to the Operating System. The main goal of the Operating System is to thus make the computer environment more convenient to use and the secondary goal is to use the resources in the most efficient manner.

What is Operating System

An operating system is a program on which application programs are executed and acts as an communication bridge (interface) between the user and the computer hardware.
The main task an operating system carries out is the allocation of resources and services, such as allocation of: memory, devices, processors and information. The operating system also includes programs to manage these resources, such as a traffic controller, a scheduler, memory management module, I/O programs, and a file system.

Important functions of an operating System:

1. **Security: –**

The operating system uses password protection to protect user data and similar other techniques. it also prevents unauthorized access to programs and user data.

2. **Control over system performance –**

Monitors overall system health to help improve performance. records the response time between service requests and system response to have a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems.

3. **Job accounting–**

Operating system Keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of user.

4. **Error detecting aids –**

Operating system constantly monitors the system to detect errors and avoid the malfunctioning of computer system.

5. Coordination between other software and users –

Operating systems also coordinate and assign interpreters, compilers, assemblers and other software to the various users of the computer systems.

6. Memory Management –

The operating system manages the Primary Memory or Main Memory. Main memory is made up of a large array of bytes or words where each byte or word is assigned a certain address. Main memory is a fast storage and it can be accessed directly by the CPU. For a program to be executed, it should be first loaded in the main memory An Operating System performs the following activities for memory management:

It keeps tracks of primary memory, i.e., which bytes of memory are used by which user program. The memory addresses that have already been allocated and the memory addresses of the memory that has not yet been used. In multi programming, the OS decides the order in which process are granted access to memory, and for how long. It Allocates the memory to a process when the process requests it and deallocates the memory when the process has terminated or is performing an I/O operation.

7. **Processor Management –**
   In a multi programming environment, the OS decides the order in which processes have access to the processor, and how much processing time each process has. This function of OS is called process scheduling. An Operating System performs the following activities for processor management.
   Keeps tracks of the status of processes. The program which perform this task is known as traffic controller. Allocates the CPU that is processor to a process. De-allocates processor when a process is no more required.

8. **Device Management –**
   An OS manages device communication via their respective drivers. It performs the following activities for device management. Keeps tracks of all devices connected to system. designates a program responsible for every device known as the Input/Output controller. Decides which process gets access to a certain device and for how long. Allocates devices in an effective and efficient way. Deallocates devices when they are no longer required.

9. **File Management –**
   A file system is organized into directories for efficient or easy navigation and usage. These directories may contain other directories and other files. An Operating System carries out the following file management activities. It keeps track of where information is stored, user access settings and status of every file and more… These facilities are collectively known as the file system.

   Moreover, Operating System also provides certain services to the computer system in one form or the other.
   The Operating System provides certain services to the users which can be listed in the following manner:

1. **Program Execution**: The Operating System is responsible for execution of all types of programs whether it be user programs or system programs. The Operating System utilises various resources available for the efficient running of all types of functionalities.
2. **Handling Input/Output Operations**: The Operating System is responsible for handling all sort of inputs, i.e, from keyboard, mouse, desktop, etc. The Operating System does all interfacing in the most appropriate manner regrading all kind of Inputs and Outputs.
   For example, there is difference in nature of all types of peripheral devices such as mouse or keyboard, then Operating System is responsible for handling data between them.
3. **Manipulation of File System**: The Operating System is responsible for making of decisions regarding the storage of all types of data or files, i.e, floppy disk/hard disk/pen drive, etc. The Operating System decides as how the data should be manipulated and stored.
4. **Error Detection and Handling**: The Operating System is responsible for detection of any types of error or bugs that can occur while any task. The well secured OS

sometimes also acts as countermeasure for preventing any sort of breach to the Computer System from any external source and probably handling them.

5. **Resource Allocation:** The Operating System ensures the proper use of all the resources available by deciding which resource to be used by whom for how much time. All the decisions are taken by the Operating System.
6. **Accounting:** The Operating System tracks an account of all the functionalities taking place in the computer system at a time. All the details such as the types of errors occurred are recorded by the Operating System.
7. **Information and Resource Protection:** The Operating System is responsible for using all the information and resources available on the machine in the most protected way. The Operating System must foil an attempt from any external resource to hamper any sort of data or information.

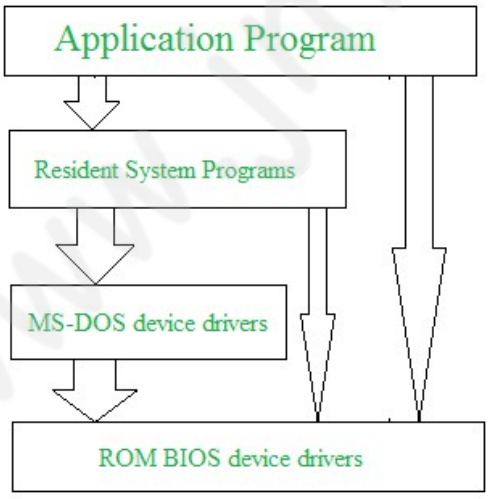
All these services are ensured by the Operating System for the convenience of the users to make the programming task easier. All different kinds of Operating System more or less provide the same services.

Different approaches or Structures of Operating Systems

Operating system can be implemented with the help of various structures. The structure of the OS depends mainly on how the various common components of the operating system are interconnected and melded into the kernel. Depending on this we have following structures of the operating system:

Simple structure:

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

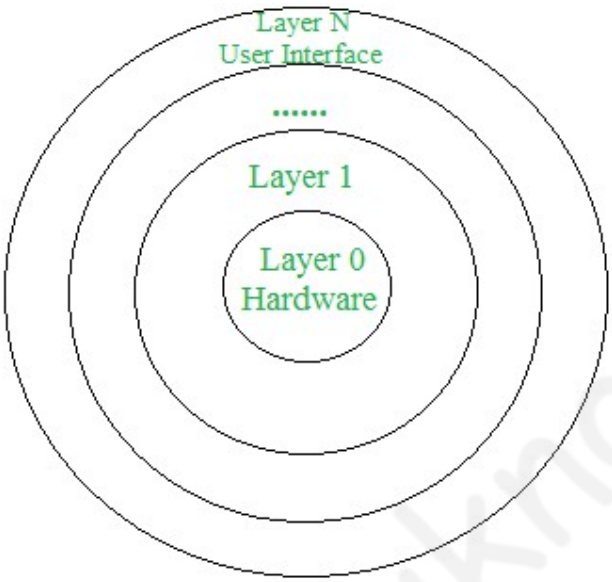


Layered structure:

An OS can be broken into pieces and retain much more control on system. In this

structure the OS is broken into number of layers (levels). The bottom layer (layer 0) is the hardware and the topmost layer (layer N) is the user interface. These layers are so designed that each layer uses the functions of the lower level layers only. This simplifies the debugging process as if lower level layers are debugged and an error occurs during debugging then the error must be on that layer only as the lower level layers have already been debugged.

The main disadvantage of this structure is that at each layer, the data needs to be modified and passed on which adds overhead to the system. Moreover careful planning of the layers is necessary as a layer can use only lower level layers. UNIX is an example of this structure.



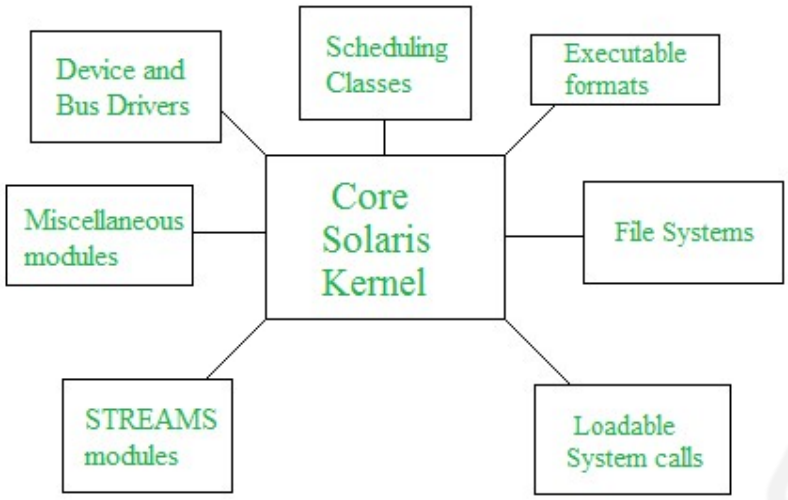
Micro-kernel:

This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs. This result in a smaller kernel called the micro-kernel.
Advantages of this structure are that all new services need to be added to user space and does not require the kernel to be modified. Thus it is more secure and reliable as if a service fails then rest of the operating system remains untouched. Mac OS is an example of this type of OS.

Modular structure or approach:

It is considered as the best approach for an OS. It involves designing of a modular kernel. The kernel has only set of core components and other services are added as dynamically loadable modules to the kernel either during run time or boot time. It resembles layered structure due to the fact that each kernel has defined and protected interfaces but it is more flexible than the layered structure as a module can call any other module.

For example, Solaris OS is organized as shown in the figure.



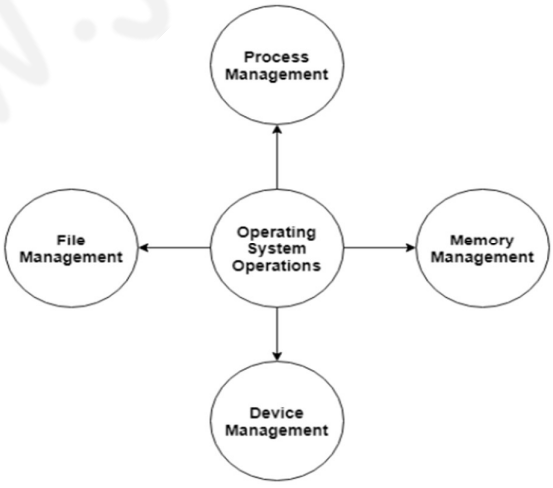
Important operations of an operating System

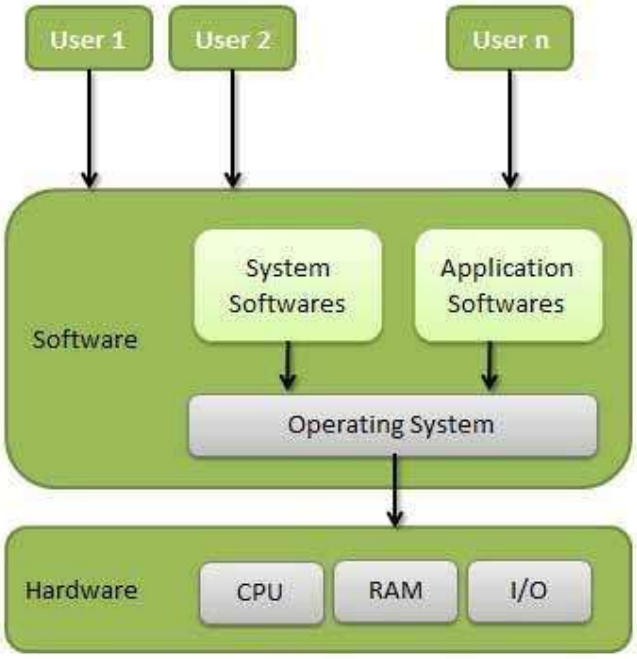
An operating system is a construct that allows the user application programs to interact with the system hardware. Operating system by itself does not provide any function but it provides an atmosphere in which different applications and programs can do useful work.

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

Definition

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





.

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

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management −

- Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
- In multiprogramming, the OS decides which process will get memory when and how much.
- Allocates the memory when a process requests it to do so.
- De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**. An Operating System does the following activities for processor management −

- Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
- Allocates the processor (CPU) to a process.
- De-allocates processor when a process is no longer required.

Device Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management −

- Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

An Operating System does the following activities for file management −

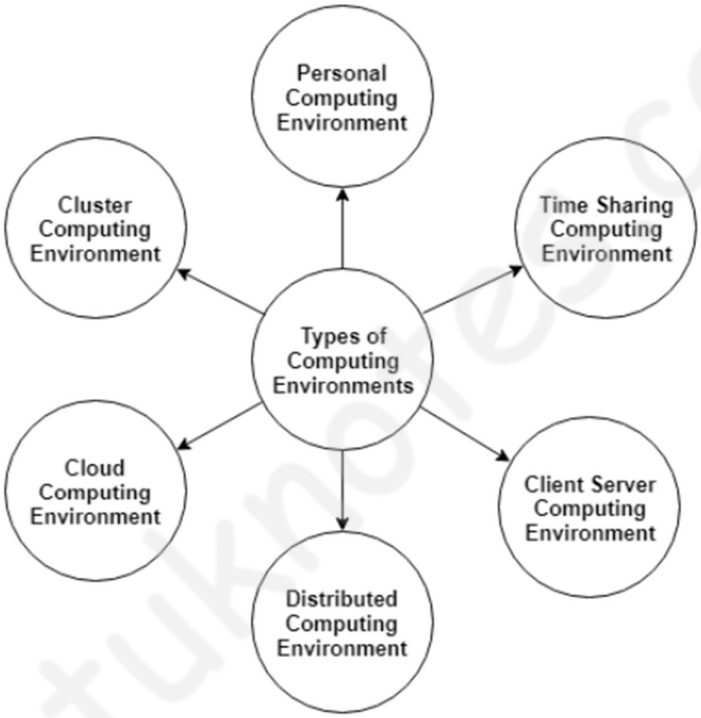
- Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

Other Important Activities

Following are some of the important activities that an Operating System performs −

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

- **Coordination between other softwares and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.
- A computer system uses many devices, arranged in different ways to solve many problems. This constitutes a computing environment where many computers are used to process and exchange information to handle multiple issues.
- The different types of Computing Environments are −



- Let us begin with Personal Computing Environment −

Personal Computing Environment

- In the personal computing environment, there is a single computer system. All the system processes are available on the computer and executed there. The different devices that constitute a personal computing environment are laptops, mobiles, printers, computer systems, scanners etc.

Time Sharing Computing Environment

- The time sharing computing environment allows multiple users to share the system simultaneously. Each user is provided a time slice and the processor switches rapidly among the users according to it. Because of this, each user believes that they are the only ones using the system.

Client Server Computing Environment

Distributed Computing Environment

- Cloud Computing Environment
- The computing is moved away from individual computer systems to a cloud of computers in cloud computing environment. The cloud users only see the service being provided and not the internal details of how the service is provided. This is done by pooling all the computer resources and then managing them using a software.
- Cluster Computing Environment
- The clustered computing environment is similar to parallel computing environment as they both have multiple CPUs. However a major difference is that clustered systems are created by two or more individual computer systems merged together which then work parallel to each other.

Open-source operating systems

Open source is a term that originally referred to **open source** software (OSS). **Open source** software is code that is designed to be publicly accessible—anyone can see, modify, and distribute the code as they see fit.

What's the difference between free software and open source?

These are two terms that get confused with one another in practice,and even get used as equivalents. All free software is open-source, but not all open-source software is free.

Open source is considered to have more flexible rules than free software, since it allows companies and developers to impose certain usage restrictions and licenses in order to protect the code's integrity. On the other hand, free software, strictly speaking, must literally adhere to the four points of freedom, according to Richard Stallman:

- There is freedom to execute the code however one wishes and for whatever purpose one wishes.
- The source code can be known and modified in its entirety.
- The code can be distributed freely (either without cost, or with a charge).
- Modifications to the code can also be freely distributed (with or without cost).

Examples of open source programs

Some widely used programs, platforms, and languages which are considered open source are:

- Linux operating system
- Android by Google
- Open office
- Firefox browser
- VCL media player

- Moodle
- ClamWinantivirus
- WordPress content management syste

Operating System Services

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

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

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

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

Program execution

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.

OPERATING SYSTEM AND USER INTERFACE

As already mentioned, in addition to the hardware, a computer also needs a set of programs—an operating system—to control the devices. This page will discuss the following:

- **There are different kinds of operating systems**: such as Windows, Linux and Mac OS
- **There are also different versions of these operating systems,** e.g. Windows 7, 8 and 10
- **Operating systems can be used with different user interfaces (UI)**: text user interfaces (TUI) and graphical user interfaces (GUI) as examples
- **Graphical user interfaces have many similarities in different operating systems**: such as the start menu, desktop etc.

When you can recognize the typical parts of each operating system's user interface, you will mostly be able to use both Windows and Linux as well as e.g. Mac OS.

THE ROLE OF OPERATING SYSTEM IN THE COMPUTER

An operating system (OS) is a set of programs which ensures the interoperability of the hardware and software in your computer. The operating system enables, among other things,

- the identification and activation of devices connected to the computer,
- the installation and use of programs, and
- the handling of files.

What happens when you turn on your computer or smartphone?
– The computer checks the functionality of its components and any devices connected to it, and starts to look for the OS on a hard drive or other memory media.
– If the OS is found, the computer starts to load it into the RAM (Random Access Memory).
– When the OS has loaded, the computer waits for commands from you.

DIFFERENT OPERATING SYSTEMS

Over the years, several different operating systems have been developed for different purposes. The most typical operating systems in ordinary computers are Windows, Linux and Mac OS.

WINDOWS

The name of the Windows OS comes from the fact that programs are run in "windows": each program has its own window, and you can have several programs open at the same time. Windows is the most popular OS for home computers, and there are several versions of it. The newest version is Windows 10.

LINUX AND UNIX

Linux is an open-source OS, which means that its program code is freely available to software developers. This is why thousands of programmers around the world have developed Linux, and it is considered the most tested OS in the world. Linux has been very much influenced by the commercial Unix OS.

In addition to servers, Linux is widely used in home computers, since there are a great number of free programs for it (for text and image processing, spreadsheets, publishing, etc.). Over the years, many different versions of Linux have become available for distribution, most of which are free for the user (such as Ubuntu, Fedora and Mint, to name a few). See the additional reading material for more information on Linux.

MAC OS X

Apple's Mac computers have their own operating system, OS X. Most of the programs that are available for PCs are also available for Macs running under OS X, but these two types of computers cannot use the exact same programs: for example, you cannot install the Mac version of the Microsoft Office suite on a Windows computer. You can install other operating systems on Mac computers, but the OS X is only available for computers made by Apple. Apple's lighter portable devices (iPads, iPhones) use a light version of the same operating system, called iOS.

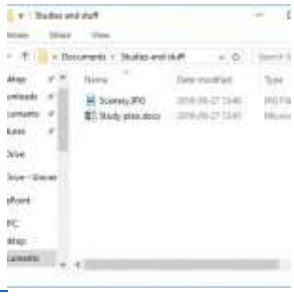
Mac computers are popular because OS X is considered fast, easy to learn and very stable and Apple's devices are considered well-designed—though rather expensive. See the additional reading material for more information on OS X.

ANDROID

Android is an operating system designed for phones and other mobile devices. Android is not available for desktop computers, but in mobile devices it is extremely popular: more than a half of all mobile devices in the world run on Android.

USER INTERFACES

A user interface (UI) refers to the part of an operating system, program, or device that allows a user to enter and receive information. A **text-based user interface** (see the image to the left) displays text, and its commands are usually typed on a command line using a keyboard. With a **graphical user interface** (see the right-hand image), the functions are carried out by clicking or moving buttons, icons and menus by means of a pointing device.



Larger image: text UI | graphical UI

The images contain the same information: a directory listing of a computer. You can often carry out the same tasks regardless of which kind of UI you are using.

TEXT USER INTERFACE (TUI)

Modern graphical user interfaces have evolved from text-based UIs. Some operating systems can still be used with a text-based user interface. In this case, the commands are entered as text (e.g., "cat story.txt").

To display the text-based Command Prompt in Windows, open the **Start** menu and type **cmd**. Press **Enter** on the keyboard to launch the command prompt in a separate window. With the command prompt, you can type your commands from the keyboard instead of using the mouse.

GRAPHICAL USER INTERFACE

In most operating systems, the primary user interface is graphical, i.e. instead of typing the commands you manipulate various graphical objects (such as icons) with a pointing device. The underlying principle of different graphical user interfaces (GUIs) is largely the same, so by knowing how to use a Windows UI, you will most likely know how to use Linux or some other GUI.

Most GUIs have the following basic components:

- a start menu with program groups
- a taskbar showing running programs
- a desktop
- various icons and shortcuts.

What is System Call in Operating System?

A **system call** is a mechanism that provides the interface between a process and the operating system. It is a programmatic method in which a computer program requests a service from the kernel of the OS.

System call offers the services of the operating system to the user programs via API (Application Programming Interface). System calls are the only entry points for the kernel system.

In this operating system tutorial, you will learn:

System Calls in Operating System

Example of System Call

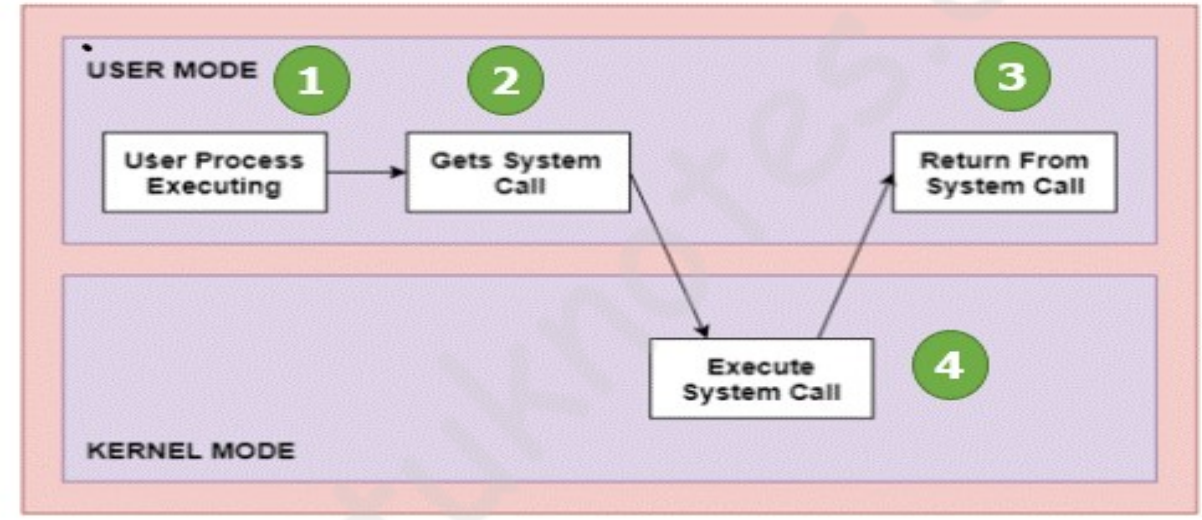
For example, if we need to write a program code to read data from one file, copy that data into another file. The first information that the program requires is the name of the two files, the input and output files.

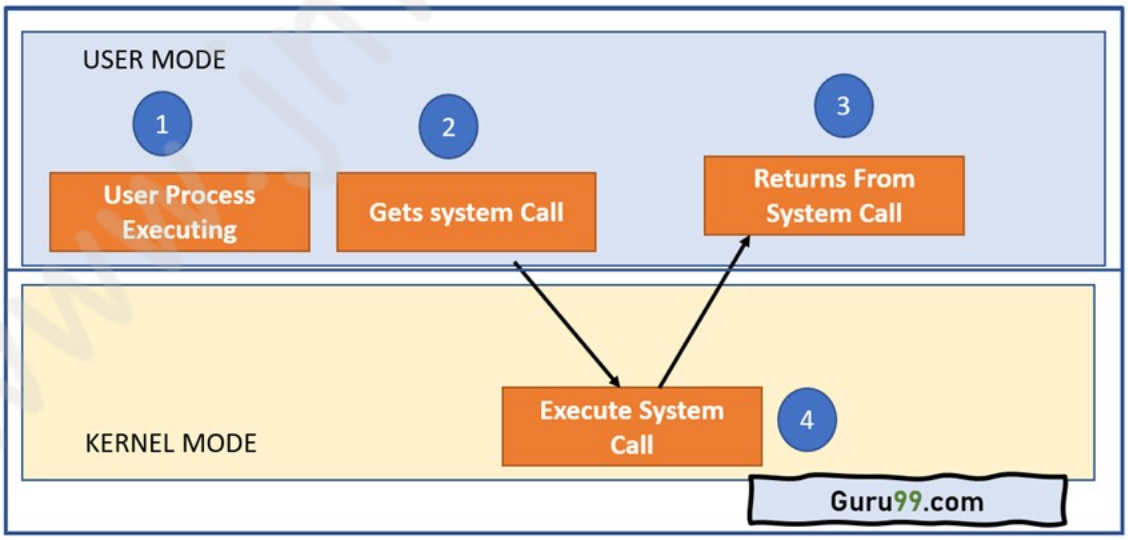
In an interactive system, this type of program execution requires some system calls by OS.

- First call is to write a prompting message on the screen
- Second, to read from the keyboard, the characters which define the two files.

How System Call Works?

Here are steps for System Call:





Architecture of the System Call

As you can see in the above-given diagram.

Step 1) The processes executed in the user mode till the time a system call interrupts it.

Step 2) After that, the system call is executed in the kernel-mode on a priority basis.

Step 3) Once system call execution is over, control returns to the user mode.,

Step 4) The execution of user processes resumed in Kernel mode.

Why do you need System Calls in OS?

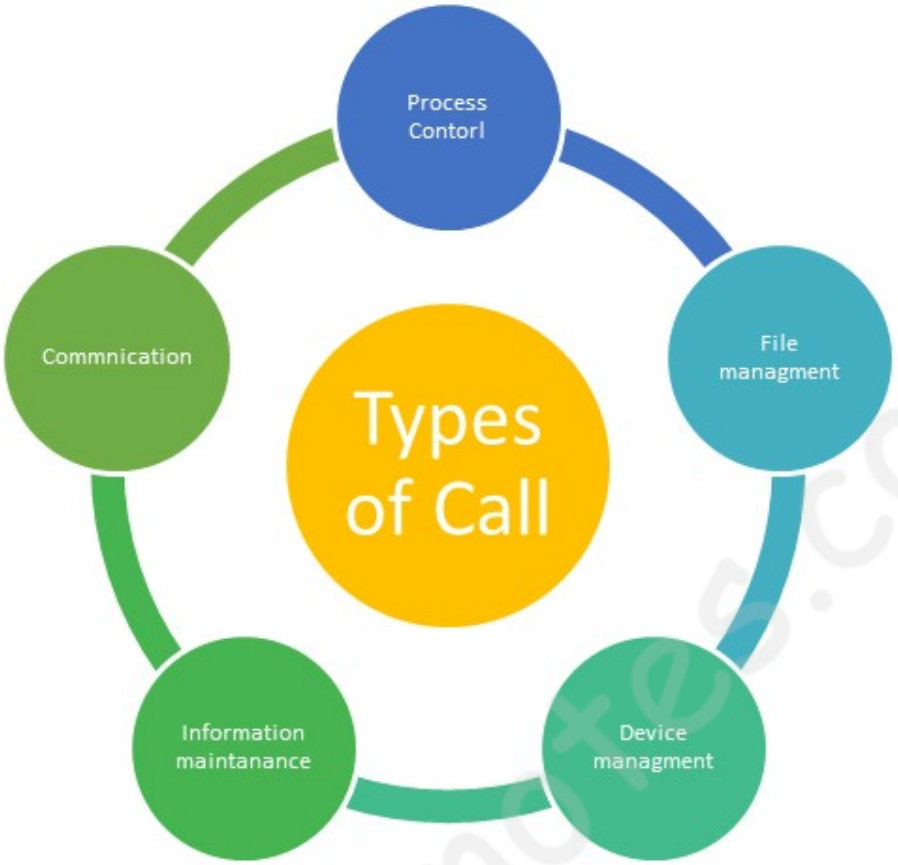
Following are situations which need system calls in OS:

- Reading and writing from files demand system calls.
- If a file system wants to create or delete files, system calls are required.
- System calls are used for the creation and management of new processes.
- Network connections need system calls for sending and receiving packets.
- Access to hardware devices like scanner, printer, need a system call.

Types of System calls

Here are the five types of system calls used in OS:

- Process Control
- File Management
- Device Management
- Information Maintenance
- Communications



Process Control

This system calls perform the task of process creation, process termination, etc.

Functions:

- End and Abort
- Load and Execute
- Create Process and Terminate Process
- Wait and Signed Event
- Allocate and free memory

File Management

File management system calls handle file manipulation jobs like creating a file, reading, and writing, etc.

Functions:

- Create a file
- Delete file
- Open and close file
- Read, write, and reposition
- Get and set file attributes

Device Management

Device management does the job of device manipulation like reading from device buffers, writing into device buffers, etc.

Functions

- Request and release device
- Logically attach/ detach devices
- Get and Set device attributes

Information Maintenance

It handles information and its transfer between the OS and the user program.

Functions:

- Get or set time and date
- Get process and device attributes

Communication:

These types of system calls are specially used for interprocess communications.

Functions:

- Create, delete communications connections
- Send, receive message
- Help OS to transfer status information
- Attach or detach remote devices

Rules for passing Parameters for System Call

Here are general common rules for passing parameters to the System Call:

- Parameters should be pushed on or popped off the stack by the operating system.
- Parameters can be passed in registers.
- When there are more parameters than registers, it should be stored in a block, and the block address should be passed as a parameter to a register.

Important System Calls Used in OS

wait()

In some systems, a process needs to wait for another process to complete its execution. This type of situation occurs when a parent process creates a child process, and the execution of the parent process remains suspended until its child process executes.

The suspension of the parent process automatically occurs with a wait() system call. When the child process ends execution, the control moves back to the parent process.

fork()

Processes use this system call to create processes that are a copy of themselves. With the help of this system Call parent process creates a child process, and the execution of the parent process will be suspended till the child process executes.

exec()

This system call runs when an executable file in the context of an already running process that replaces the older executable file. However, the original process identifier remains as a new process is not built, but stack, data, head, data, etc. are replaced by the new process.

kill():

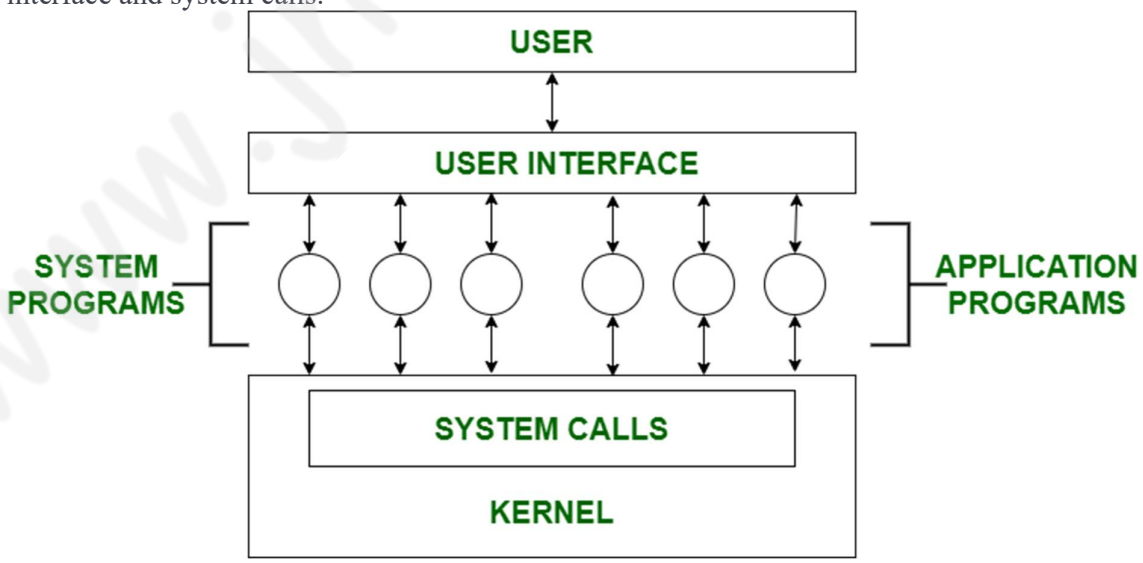
The kill() system call is used by OS to send a termination signal to a process that urges the process to exit. However, a kill system call does not necessarily mean killing the process and can have various meanings.

exit():

The exit() system call is used to terminate program execution. Specially in the multi-threaded environment, this call defines that the thread execution is complete. The OS reclaims resources that were used by the process after the use of exit() system call.

System Programs in Operating System

System Programming can be defined as act of building Systems Software using System Programming Languages. According to Computer Hierarchy, one which comes at last is Hardware. Then it is Operating System, System Programs, and finally Application Programs. Program Development and Execution can be done conveniently in System Programs. Some of System Programs are simply user interfaces, others are complex. It traditionally lies between user interface and system calls.



So here, user can only view up-to-the System Programs he can't see System Calls.

System Programs can be divided into these categories :

1. **File Management –**

   A file is a collection of specific information stored in memory of computer system. File management is defined as process of manipulating files in computer system, it management includes process of creating, modifying and deleting files.

   - It helps to create new files in computer system and placing them at specific locations.
   - It helps in easily and quickly locating these files in computer system.
   - It makes process of sharing of files among different users very easy and user friendly.
   - It helps to stores files in separate folders known as directories.
   - These directories help users to search file quickly or to manage files according to their types or uses.
   - It helps user to modify data of files or to modify he name of file in directories.

2. **Status Information –**

   Information like date, time amount of available memory, or disk space is asked by some of users. Others providing detailed performance, logging and debugging information which is more complex. All this information is formatted and displayed on output devices or printed. Terminal or other output devices or files or a window of GUI is used for showing output of programs.

3. **File Modification –**

   For modifying contents of files we use this. For Files stored on disks or other storage devices we used different types of editors. For searching contents of files or perform transformations of files we use special commands.

4. **Programming-Language support –**
   For common programming languages we use Compilers, Assemblers, Debuggers and interpreters which are already provided to user. It provides all support to users. We can run any programming languages. All languages of importance are already provided.

5. **Program Loading and Execution –**
   When program is ready after Assembling and compilation, it must be loaded into memory for execution. A loader is part of an operating system that is responsible for loading programs and libraries. It is one of essential stages for starting a program. Loaders, relocatable loaders, linkage editors and Overlay loaders are provided by system.

6. **Communications –**
   Virtual connections among processes, users and computer systems are provided by programs. User can send messages to other user on their screen, User can send e-

mail, browsing on web pages, remote login, transformation of files from one user to another.

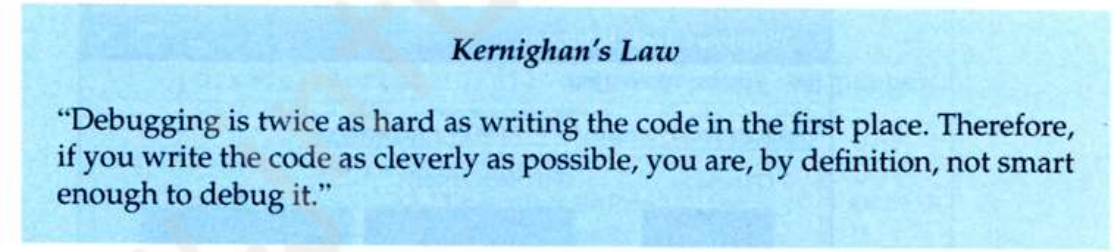
Some examples of system programs in O.S. are –

- Windows 10
- Mac OS X
- Ubuntu
- Linux
- Unix
- Android
- Anti-virus
- Disk formatting
- Computer language translator

Operating-System Debugging

Kernighan's Law

"Debugging is twice as hard as writing the code in the first place. Therefore,
if you write the code as cleverly as possible, you are, by definition, not smart
enough to debug it."



- Debugging here includes both error discovery and elimination and performance tuning.

2.8.1 Failure Analysis

- Debuggers allow processes to be executed stepwise, and provide for the examination of variables and expressions as the execution progresses.
- Profilers can document program execution, to produce statistics on how much time was spent on different sections or even lines of code.
- If an ordinary process crashes, a memory dump of the state of that process's memory at the time of the crash can be saved to a disk file for later analysis.
    - The program must be specially compiled to include debugging information, which may slow down its performance.

- These approaches don't really work well for OS code, for several reasons:
    - The performance hit caused by adding the debugging (tracing) code would be unacceptable. (Particularly if one tried to "single-step" the OS while people were trying to use it to get work done!)
    - Many parts of the OS run in kernel mode, and make direct access to the hardware.
    - If an error occurred during one of the kernel's file-access or direct disk-access routines, for example, then it would not be practical to try to write a crash dump into an ordinary file on the filesystem.
        - Instead the kernel crash dump might be saved to a special unallocated portion of the disk reserved for that purpose.

2.8.2 Performance Tuning

- Performance tuning (debottlenecking) requires monitoring system performance.
- One approach is for the system to record important events into log files, which can then be analyzed by other tools. These traces can also be used to evaluate how a proposed new system would perform under the same workload.
- Another approach is to provide utilities that will report system status upon demand, such as the unix "top" command. (w, uptime, ps, etc.)
- System utilities may provide monitoring support.

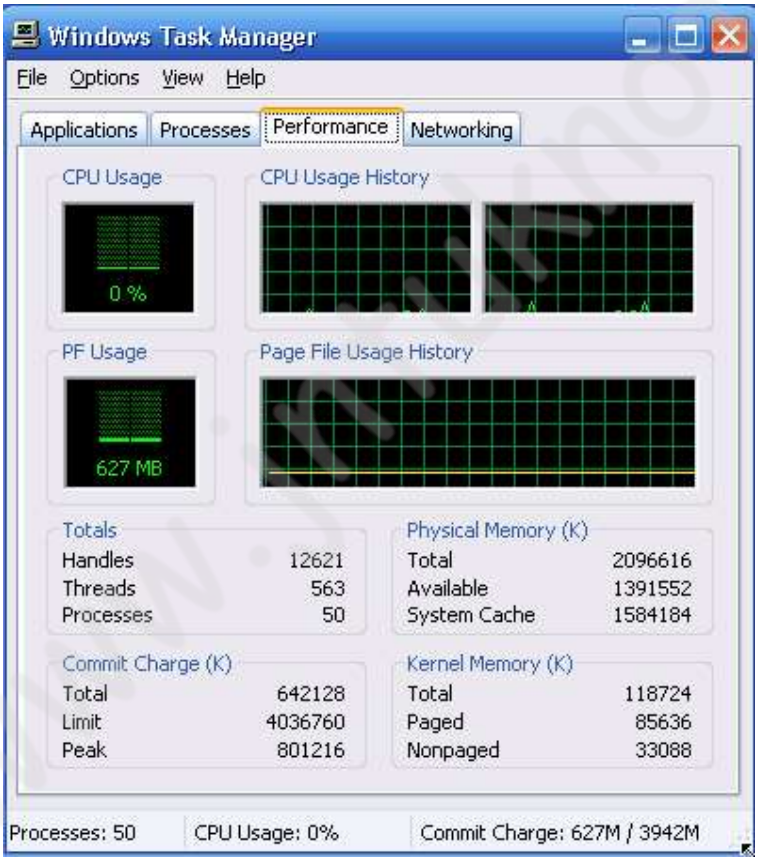


Figure 2.19 - The Windows task manager.

2.8.3 DTrace

- DTrace is a special facility for tracing a running OS, developed for Solaris 10.

- DTrace adds "probes" directly into the OS code, which can be queried by "probe consumers".
- Probes are removed when not in use, so the DTrace facility has zero impact on the system when not being used, and a proportional impact in use.
- Consider, for example, the trace of an ioctl system call as shown in Figure 2.22 below.

System Boot

The general approach when most computers boot up goes something like this:

- When the system powers up, an interrupt is generated which loads a memory address into the program counter, and the system begins executing instructions found at that address. This address points to the "bootstrap" program located in ROM chips (or EPROM chips) on the motherboard.
- The ROM bootstrap program first runs hardware checks, determining what physical resources are present and doing power-on self tests (POST) of all HW for which this is applicable. Some devices, such as controller cards may have their own on-board diagnostics, which are called by the ROM bootstrap program.
- The user generally has the option of pressing a special key during the POST process, which will launch the ROM BIOS configuration utility if pressed. This utility allows the user to specify and configure certain hardware parameters as where to look for an OS and whether or not to restrict access to the utility with a password.
    - Some hardware may also provide access to additional configuration setup programs, such as for a RAID disk controller or some special graphics or networking cards.
- Assuming the utility has not been invoked, the bootstrap program then looks for a non-volatile storage device containing an OS. Depending on configuration, it may look for a floppy drive, CD ROM drive, or primary or secondary hard drives, in the order specified by the HW configuration utility.
- Assuming it goes to a hard drive, it will find the first sector on the hard drive and load up the fdisk table, which contains information about how the physical hard drive is divided up into logical partitions, where each partition starts and ends, and which partition is the "active" partition used for booting the system.
- There is also a very small amount of system code in the portion of the first disk block not occupied by the fdisk table. This bootstrap code is the first step that is not built into the hardware, i.e. the first part which might be in any way OS-specific. Generally this code knows just enough to access the hard drive, and to load and execute a (slightly) larger boot program.
- For a single-boot system, the boot program loaded off of the hard disk will then proceed to locate the kernel on the hard drive, load the kernel into memory, and then transfer control over to the kernel. There may be some opportunity to specify a particular kernel to be loaded at this stage, which may be useful if a new kernel has just been generated and doesn't work, or if the system has multiple kernels available with different configurations

for different purposes. (Some systems may boot different configurations automatically, depending on what hardware has been found in earlier steps.)

- For dual-boot or multiple-boot systems, the boot program will give the user an opportunity to specify a particular OS to load, with a default choice if the user does not pick a particular OS within a given time frame. The boot program then finds the boot loader for the chosen single-boot OS, and runs that program as described in the previous bullet point.
- Once the kernel is running, it may give the user the opportunity to enter into single-user mode, also known as maintenance mode. This mode launches very few if any system services, and does not enable any logins other than the primary log in on the console. This mode is used primarily for system maintenance and diagnostics.
- When the system enters full multi-user multi-tasking mode, it examines configuration files to determine which system services are to be started, and launches each of them in turn. It then spawns login programs (gettys) on each of the login devices which have been configured to enable user logins.
    - (The getty program initializes terminal I/O, issues the login prompt, accepts login names and passwords, and authenticates the user. If the user's password is authenticated, then the getty looks in system files to determine what shell is assigned to the user, and then "execs" (becomes) the user's shell. The shell program will look in system and user configuration files to initialize itself, and then issue prompts for user commands. Whenever the shell dies, either through logout or other means, then the system will issue a new getty for that terminal device.)