Operating system

Unit-1

Program

A program is a collection or group or set of statements or instructions which performs a particular task.

Software

Software is a collection or set of programs which collectively performs a set of activities.

Types of software

1. System software
2. Application software

System software

Software which provides services to other software is called as system software. System Software is used for operating computer hardware. System Software is installed on the computer when operating system is installed. The user does not interact with system software because it works in the background. System software provides platform for running application software.

Example system softwares are: Operating Systems, Compilers, Linkers, Loaders, Assemblers, debugger, driver and so on.

Application software

Software which depends on other software for services is called as application software. Application software is used by user to perform specific task. Application softwares are installed according to user's requirements. The user interacts with application sofwares. Application software can't run independently. They can't run without the presence of system software.

Example application softwares are: VLC player, web browser, word processor and so on.

Computer

A computer contains number of hardware and software components. Each component of the system is called as a resource. Some of the hardware resources are: processor, HD, RAM, keyboard, monitor and so on. Some of the software resources are: operating system, compiler, interpreter, loader, assembler and so on.

2. What is an Operating System?

Operating System

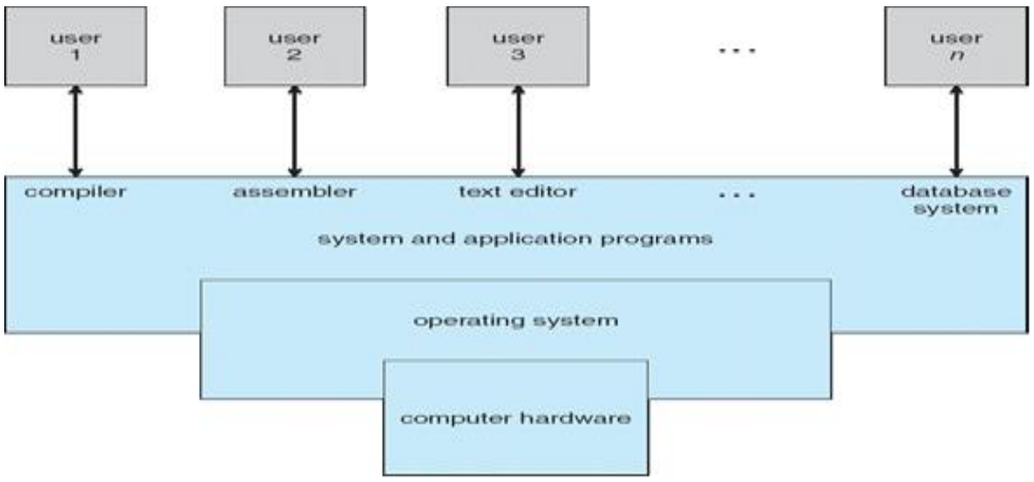
An Operating System is a program that manages the computer hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware. Some operating systems are designed to provide convenient communication between user and computer system, others for efficient utilization of resources, and others to provide both.

OS — Convinence ✓
Efficiency ✓ 2✓

Role of Operating System

The role of operating system in the computer system is illustrated with the help of following block diagram.

A computer system can be divided into four components: the hardware, the operating system, the application programs, and the users as shown in above figure. The **hardware** (the central processing unit, the memory and the I/O devices) provides the computing resources for the computer system. The **system programs** (compilers, assemblers and so on) and **application programs** (word processors, spreadsheets, web browsers and so on) help the user programs in their execution. The **operating system** controls the hardware and coordinates the use of hardware among various user programs.
An operating system is similar to a *government.* Like a government, it performs no useful function by itself. It simply provides an *environment* within which other programs can do useful work.



The role of operating system is fully explained by considering two views: user view and system view.

User view

Computer systems are divided into four categories from user point of view.

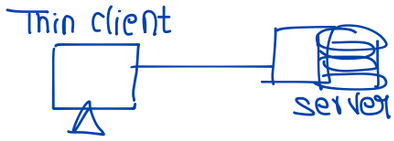
1) Personal computers

Used by single person for home applications. The Operating System used in personal computers should focus on the following

1. Easy use of the system
2. Performance of the system
3. Utilization of resources

2) Thin clients

A thin client is a system which do not have any processing capability and storage capacity. It is used for typing the program and displaying the output. Thin clients are connected to a server system which has storage capacity and execution capability. The programs typed in the thin clients are stored and executed in server system. The results of execution are passed to client systems.

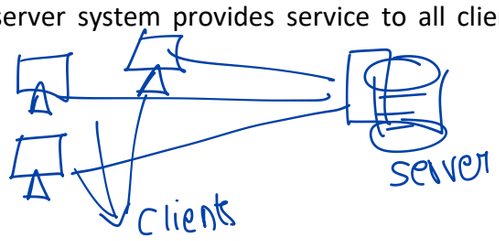


Ex: Oracle server

The operating system used in thin clients should concentrate on resource utilization.

3) Client – Server system

A number of client systems are connected to a server system. A client system makes a request to a server system for any service. The server system provides service to all client systems



connected to it. The Operating System used in client system should use resources of the client system as well as resources available at the server system.

4) Home devices or devices used in automobiles

The Operating System used in home devices should work automatically without any intervention of user. An example for home device is a microwave oven.

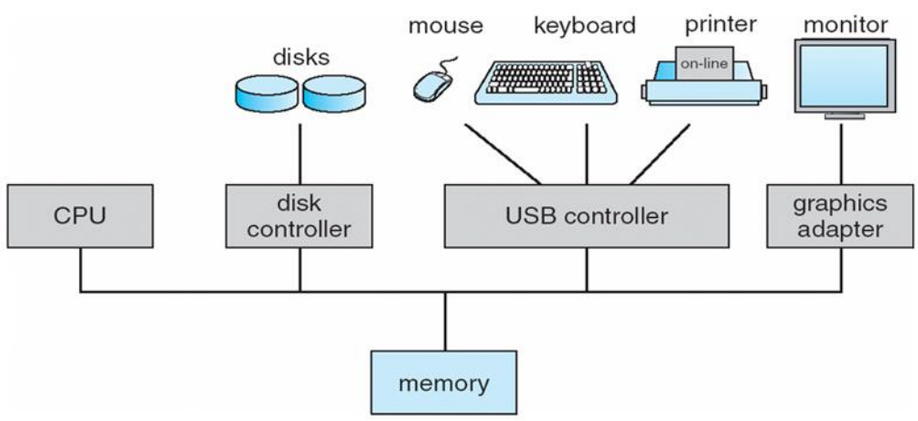
System view

From system point of view, the Operating System can be viewed as resource allocator. The Operating System allocates the resources of the computer system to the programs that are currently running in the system.

Computer System Operation

A modern computer system consists of one or more CPUs and number of devices. The CPUs and devices are connected to system bus as shown in following diagram. The devices are connected to the system bus through device controllers. Each device controller controls the operations of a device or a set of devices.

When the computer system is booted, a program called bootstrap program starts running. The bootstrap program resides in ROM or EEPROM. Role of bootstrap program is locating the operating system, loading operating system into RAM and starting execution of operating system. When operating system is started then operating system controls the operations of computer system.



When any work has to be done with the help of any hardware component then operating system assigns that work to the hardware component and then waits for some event to occur. The hardware component raises an interrupt when it finishes the assigned work. A software component invokes a system call when it requires any service from the operating system.

When an interrupt is raised or a system call is invoked then CPU stops the program that it is currently executing and transfers control to the corresponding interrupt service routine. The interrupt service routine processes the interrupt and returns control to CPU. The CPU then restarts the interrupted program.

Storage Structure (or) Storage Hierarchy

A storage device is either volatile or nonvolatile. A volatile storage device loses its contents when the power is switched off. A nonvolatile storage device retains its contents even the power is switched off.

Registers

A computer system contains number of registers. Registers are volatile in nature. During execution of any process, the instructions and data required by ~~instructions~~ process are all stored in registers.

Cache memory

A cache memory exists between CPU and main memory (RAM). It is volatile in nature. During execution of any process, the frequently executed instructions of the process are stored in cache memory.

Read Only Memory (ROM)

ROM is nonvolatile in nature. It contains the programs that are activated when the system is booted. One of the programs in ROM is boot strap program. The content in ROM cannot be erased.

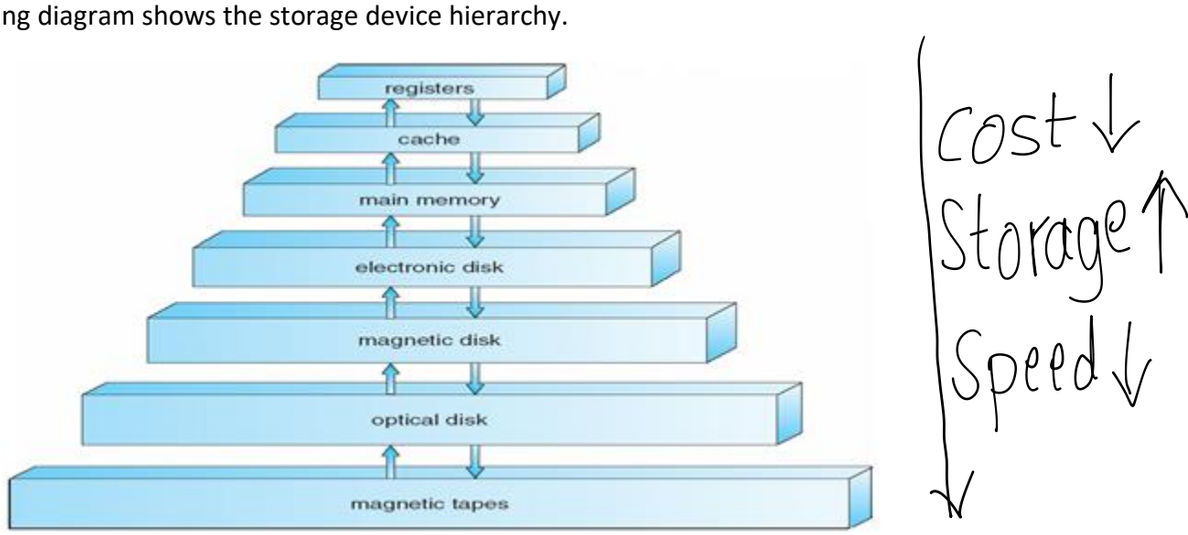
Main Memory (RAM)

Main memory is volatile in nature. The code of operating system is stored in one part of RAM. Other part of RAM is used to store user processes. When the user wants to execute a program the operating system loads that program into RAM.

Magnetic Disk (or) Hard disk

Magnetic disk is nonvolatile. It is permanent storage device. When the user saves any program, it will be stored in hard disk. When the user wants to execute his program then the operating system loads the program from disk to RAM.

The following diagram shows the storage device hierarchy.



In the hierarchy, from top to bottom: the cost of storage devices decreases, storage capacity increases; from bottom to top: accessing speed increases.

I/O Structure

A large portion of operating system code is dedicated to manage i/o because of its importance and varying nature of i/o devices. Each device is connected to system bus through a device controller. A device controller can connect one or more devices to the system bus. Each device controller maintains a local buffer and a set of registers. The device controller moves data between the device and buffer. The operating system has a device driver for each device controller. To start an i/o operation, the operating system informs the corresponding device driver. The device driver instructs the device controller by loading the information regarding

the operation to be performed in the registers. The device controller then transfers data between the device and buffer. After completion of operation the controller informs the device driver via an interrupt.

Functions (or) Roles (or) Responsibilities (or) Activities of on Operating System

The following are the major functions of an operating system

1. Process Management
2. Memory Management
3. File Management
4. I/O Management
5. Protection
6. Security
7. Networking

Process Management

A program is a set of instructions. Program is a passive element and it resides in hard disk. A process is a program in execution and it resides in RAM. To run any process, some resources are required and that resources are allocated by the Operating System.

To mange processes, the Operating System performs the following activities

1 creating and deleting processes
2 suspending and resuming processes
3 providing synchronization
4 providing communication
5 handling deadlocks

Creating and deleting processes

To create a new process, the operating system has to do the following activities

1. Move the program from disk to RAM
2. Allocate CPU or processor to the program

To delete a process, the Operating System has to move the program form RAM to disk.

Suspending and resuming processes

Suspending a process is temporarily stopping the execution of the process due to some I/O request. Before suspending the process, the Operating System has to save the status of the execution of the process. Restarting the execution of a suspended process is called resuming the process. Before resuming the process, the Operating System has to restore the saved status of the process.

Providing synchronization

A sharable resource can be used by number of processes at a time. Ex: RAM, File etc. A non sharable resource has to be used by only one process at a time. Ex: CPU, printer. If number of processes requests a non sharable resource at the same time, then the Operating System has to provide synchronous access to the resource by allocating the resource to only one process at a time.

Providing commutation

If there are number of programs belonging to the same application then there should be some communication between the programs for sharing or accessing data. The Operating System has to provide communication between the programs. Two methods used for providing communication are:

1) Shared memory
2) Message passing

Handling deadlocks

A set of processes is said to be in deadlock state if each process in the set is waiting for another process. For example, if there are two processes P1 and P2 and two resources R1 and R2 and if P1 is using R1 and requesting R2, P2 is using R2 and requesting R1 then the two processes are in deadlock state. The Operating System has to allocate resources to processes such that the system should not go into dead lock state.

Memory Management

When a program is saved, it will be stored in hard disk. To execute a program, that program has to be moved to RAM. Operating system has to manage both primary memory (RAM) and secondary memory (Hard disk).

For managing primary memory, the Operating System is responsible for

1) Knowing which parts of the RAM are in use and by whom: RAM contains number of processes at any time. The Operating System must know which parts of the RAM are stored with which processes.
2) Deciding which process needs to be moved between RAM and Hard disk: while executing any process, if the process requests any i/o operation then the execution of that process will be stopped. If the RAM is full and if a new program has to be loaded into RAM, then the Operating System moves one of the waiting processes from RAM to hard disk in order to create space in RAM for loading the new program.
3) Allocating & de allocating space when required: to execute a program, the Operating System allocates space in RAM and moves the program from hard disk to RAM. When the execution of program is completed then the Operating System deallocates the space occupied by the program and moves the program from RAM to hard disk.

The hard disk contains number of storage blocks. At any time, some of the blocks contains programs or data and remaining blocks are free.

For managing secondary memory, the Operating System is responsible for

1) Free – space management
2) Storage allocation
3) Disk scheduling

Free – space Management

The Operating System maintains a list called “Free Block list” which contains the numbers of free blocks in the hard disk. The operating system updates this list whenever any data is stored or removed from disk.

Storage Allocation

When any data has to be stored in hard disk, the Operating System identifies free blocks from the free block list and allocates that free blocks for the data and updates the “Free Block List”.

Disk scheduling

If there are number of read and write requests for hard disk, then the Operating System has to schedule these requests i.e. the Operating System has to decide the order in which these requests are to be performed.

File Management

Computers use different types of storage devices like magnetic disk, optical disk, magnetic tape and so on. These storage devices have different physical properties and physical organization. To make the computer system convenient for users, the operating system hides the physical properties of storage devices from users by defining a logical storage unit called ‘file’. The Operating System maps files onto storage devices. Files are organized into directories to make them easier to use.

The Operating System is responsible for the following activities for managing files

1) Creating and deleting files
2) Creating and deleting directories
3) Supporting primitives for manipulating files and directories
4) Mapping files onto secondary storage
5) Backing up files on stable storage

I/O device Management

A computer system contains number of input and output devices like keyboard, mouse, monitor, printer and so on. Each device is controlled by a device driver. The Operating System has to manage all these device drivers.

Networking

A network is a collection of systems connected to each other. The Operating System used in a computer which is connected to a network must provide support

1) For sharing the resources available in the network.
2) Dividing the program into parts and distributing them to different systems in the network.
3) Providing transparency to the users.

Protection and Security

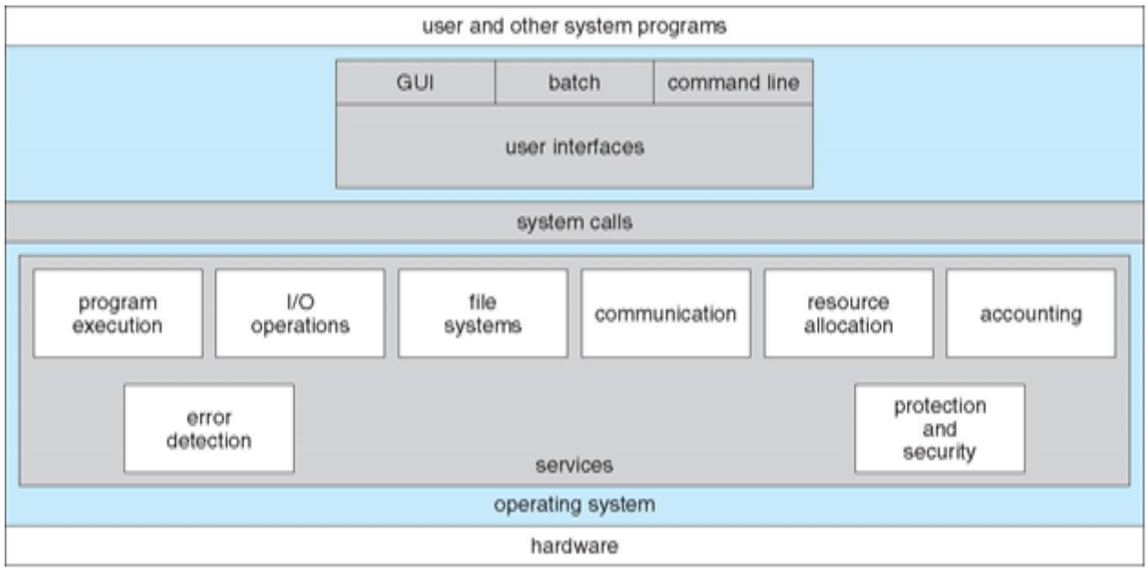
Protection: restricting access to the resources of the system from the processes that are running in the system. For example, if two or more processes request the CPU at the same time then the Operating System has to restrict the access to the CPU by allocating the CPU to only one process at a time.

Security: restricting access to the system by the users. Username and password are used to provide security.

b. With a neat sketch, List and explain Operating system Services

Operating System Services

The operating system provides an environment for the execution of programs. It provides certain services to programs and to the users of those programs. Following figure shows one view of the various operating-system services.



The services provided by an operating system for users and their programs are:

1) User interface
2) Program execution
3) I/O operations
4) File system manipulation
5) Communication
6) Error Detection

User interface

An operating system provides either one or combination of the following user interfaces.

1) Command line interface
2) Batch interface
3) Graphical user interface

With command line interface, interaction between user and computer takes place through commands.
With batch interface, interaction between user and computer takes place through commands. It supports both single command and batch of commands. The batch of commands is generally included into a file and that file is executed.
With graphical user interface, interaction between user and computer takes place through a window system and pointing devices like mouse.

Program Execution

Operating system has to load the program into memory and assign CPU to the program to start the execution of program and also allocate required resources during execution of program.

I/O Operation

A running program may need to perform i/o operations. The i/o operations are generally performed on either hardware devices or files. The operating system has to support all these i/o operations.

File system manipulation

An executing program may need to perform some operations on files. Some example operations performed on files are: creating new files, reading data from files, writing data to files and so on. The operating system has to provide support for performing all these operations.

Communication

A process needs to communicate with other processes to exchange some information. This communication can be established using either shared memory or message passing. This communication can be established between processes running on the same system or processes running on different systems.

Error detection

Errors may occur in the hardware or software of the computer system. Some example errors are: power failure, arithmetic overflow, connection failure on network and so on. Operating system has to take care of all these errors.

Another set of services provided by operating system for improving the efficiency of computer system are:

1) Resource allocation
2) Accounting
3) Protection and Security

Resource allocation

Each process requires number of resources. Operating system has to allocate resources to processes. To allocate some resources like CPU, the operating system uses allocation code like scheduling algorithms. Other resources are allocated without executing any allocation code.

Accounting

Accounting is recording which resources are used by which processes and for how much time the allocated resources are used by the processes. This accounting information is used for billing purpose and to know which resources are heavily used. This usage information is used to decide how many copies of each resource have to be maintained.

Protection and Security

Protection: restricting access to the resources of the system from the processes that are running in the system. For example, if two or more processes request the CPU at the same time then the Operating System has to restrict access to the CPU by allocating the CPU to only one process at a time.

Security: restricting access to the system by the users. Username and password are used to provide security.

4. What is a System Call? Explain about different types of System Calls?

System calls

The services provided by an operating system are made available to user programs through system calls. System calls are generally developed in C and C++ languages. Some system calls which directly interacts with hardware devices are developed in assembly language. System calls are executed in kernel of operating system when a user program requests the operating system for any service. For example, when a user program calls a scanf() function for reading data from keyboard then the system call ‘read()’ is invoked and executed in the kernel.

Number of system calls is invoked during execution of even a small program. For example, assume that there is a program for copying data from one file to other file. Some of the system calls invoked during the execution of this program are:

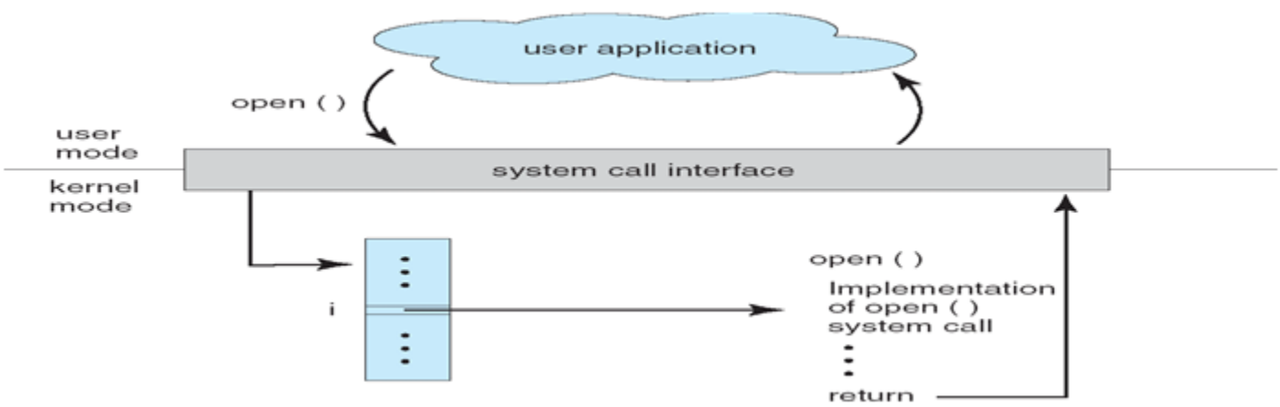
1) A system call to display the prompt message
2) A system call to read the name of source file
3) A system call to display the prompt message
4) A system call to read the name of destination file
5) A system call for opening the source file
6) A system call for opening the destination file
7) A sequence of system calls for reading data from the source file and writing data into destination file
8) A system call for closing the source file
9) A system call for closing the destination file

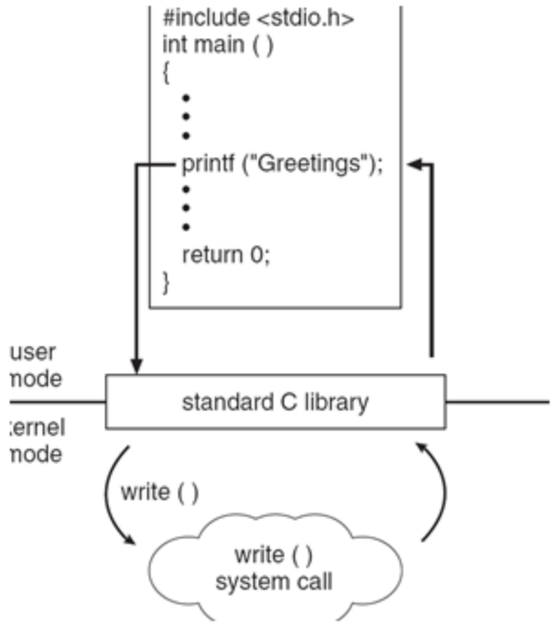
The programmers develop programs using Application Programming Interface (API). The API contains set of functions/methods using which the programmer develops programs. The API also specifies the parameters that need to be passed to each function and also the values returned by each function. Java API is one example. Java API is used for designing programs that run on the Java virtual machine. Each function or method in the API is linked to a system call. When any API function or method is executed in the program then the corresponding system call is invoked.

There are two main reasons for writing programs using API instead of system calls directly.

1. Program written using API can be executed on any system that supports the same API.
2. Working with system calls is more difficult than API.

The run-time support system for most programming languages provides a system-call interface that serves as the link to system calls made available by the operating system. During execution of user program, when any API function is executed then the system-call interface identifies the corresponding system call and activates that system call within the operating system and returns the status of the system call and any return values. The system-call interface maintains a table with one row for each system call. Each row is indexed by a number.





System calls may require parameters as they are basically functions. The parameters required by system calls are passed in any one of the following three ways. The simplest approach is to pass the parameters in *registers.* If there are more parameters than available registers then the parameters are generally stored in a *block* in memory and the address of the block is passed as a parameter in a register. This is the approach taken by Linux and Solaris. Parameters also can be passed by storing on the *stack.* Some operating systems prefer the block or stack method because those approaches do not limit the number or length of parameters being passed.

Types of system calls

1. Process control system calls
2. File management system calls
3. Device management system calls
4. Information maintenance system calls
5. Communication maintenance system calls
6. Protection and security maintenance system calls

Some important system calls under each category are:

Process control system calls

The various system calls for controlling processes are

1) end, abort
2) load, execute
3) create process , terminate process
4) get process attributes , set process attributes
5) wait, signal
6) lock

End system call is invoked when the process is terminated normally. Abort system call is invoked when the process is terminated abnormally due to logical errors. Load system call is invoked when a process is loaded in to RAM. Execute system call is invoked when CPU is allocated to the process. Create process system call is invoked when a process creates a child process. Terminate process system call is invoked when the created child process is terminated. Set process attributes is invoked to set the attributes to the process. Get process attributes is invoked to get the attributes of the process. Wait system call is invoked when parent process

has to wait for the completion of child process. Signal system call is invoked when child process completes its execution.

File management system calls

1) create, delete
2) open, close
3) read, write, reposition
4) get file attributes, set file attributes

Device management system calls

1) request device, release device
2) read, write, reposition
3) get attributes, set attributes

Information maintenance system calls

1) get time or date, set time or date
2) get system data, set system data
3) dump memory
4) time profile

Communication maintenance system calls

1) open connection, close connection
2) get hostid, get process id
3) send message, receive message
4) shared memory create, shared memory atttach

protection

1) set permission, get permission
2) allow user, deny user

3. Describe different types of Operating system structures.

Types of Operating Systems

1) Batch processing operating system
2) Multiprogramming operating system
3) Timesharing operating system
4) Distributed operating system
5) Real time operating system
6) Multi user operating system
7) Multi tasking operating system
8) Embedded operating system
9) Mobile operating system

Batch processing operating system

If the computer system is running with batch processing operating system then the execution of programs in the computer system is as follows:
The programs that users want to execute and the input data required for executing the programs are collected into a batch. The batch of programs is then loaded into computer system for execution. The batch of programs is then executed in sequential manner. There is no interaction between users and computer during execution of the programs. The outputs generated after execution of programs is collected into a batch and then distributed to users. IBM's Z/OS is an example Batch operating system.

Multiprogramming operating system

When only one program is loaded into main memory and if that program requires any i/o operation during its execution then the CPU will be in idle state until the i/o operation is completed. In this case the utilization of resources (CPU, i/o devices) is low. To increase the utilization of resources, number of programs is loaded into memory at a time. During execution of program, if the program requires any i/o operation then the CPU is switched to another program, so that the CPU is busy at all times. Loading number of programs into main memory is called **multiprogramming.** Multiprogramming operating system allows loading of number of programs at a time into RAM. Multiprogramming operating system switches CPU from current program to another program when the current process is completed or goes to the waiting state. Windows and LINUX are examples for multiprogramming operating systems.

Timesharing operating system

Time sharing operating system loads number of programs at a time into main memory. Allow each program to execute for a certain amount of time only. After that the CPU is switched to another program. So, each program has equal chance of getting CPU. Timesharing operating system switches CPU from current program to another program when the current process is completed or goes to the waiting state or the allocated time slice is over. The user can interact with his program while it is running. A time-shared operating system allows many users to share the computer simultaneously. As the system switches rapidly from one user to the next, each user is given the impression that the entire computer system is dedicated to his use though it is being shared among many users. Windows and LINUX are examples for timesharing operating systems.

Distributed operating system

Distributed operating system manages a group of independent computers and makes them appear to be a single computer. The group of computers is connected through a network. Following are the features of distributed operating systems:

Resource sharing

Resources can be shared by the computer systems connected to the network. For example, if a printer is connected to the network then all systems in the network can share the printer.

Reliability of resources

If any resource of any system fails then the resource of other system in the network can be used.

Speed up of computations

Programs can be executed in less amount of time by dividing the program into number of parts and executing these parts on different systems in the network.

Communication between systems

If more number of persons is involved in development of any project then there should be some communication between the systems that are being used by the persons. This communication can be provided easily by the distributed operating system.

Providing Transparency

The distributed operating system hides the details of execution of the program from the user.

Amoeba and LOCUS are examples for distributed operating systems.

Real time operating system

Rea time operating systems are used in computer systems which executes real time applications. Real time applications have fixed time constraints on processing. For example, if there is a satellite which sends some data for every 100 seconds and if a computer system receives and stores that data then the operating system used in that computer system is a real time operating system. In this example, the operating system has to receive and store the data within 99 seconds. Windows CE and Symbian are examples for real time operating systems.

Multiuser operating system

Multi user operating system allows multiple users to access a computer system simultaneously. Time-sharing systems can be classified as multi-user systems as they enable multiple user access to a computer through time sharing. UNIX and open VMS are examples for multi user operating systems.

Multitasking operating system

Multitasking operating system allows execution of multiple tasks at a time. There are two versions of multitasking: pre-emptive and co-operative. In pre-emptive multitasking, the operating system slices the CPU time and dedicates one slot to each of the programs. Unix-like operating systems such as Solaris and Linux support pre-emptive multitasking. In co-operative multitasking, CPU is switched to another process when the current process is completed or goes to the waiting state. MS Windows prior to Windows 95 used to support cooperative multitasking.

Embedded operating system

The operating systems designed for being used in embedded computer systems are known as embedded operating systems. They are designed to operate on small machines like PDAs with less autonomy. They are able to operate with a limited number of resources. They are very compact and extremely efficient. Windows CE and FreeBSD are some examples of embedded operating systems.

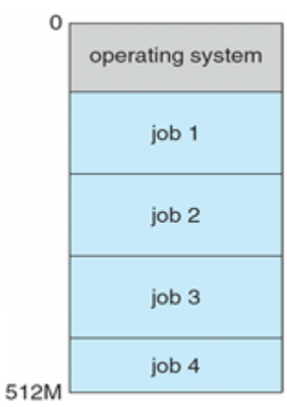
Mobile operating system

A mobile OS controls a mobile device and its design supports wireless communication and mobile applications. Tablet PCs and smart phones run on mobile operating systems. Blackberry OS, Google's Android and Apple's iOS are some of the most known names of mobile operating systems.

5. a. Explain the Dual-Mode operation of an operating system.

Dual mode operation of the system

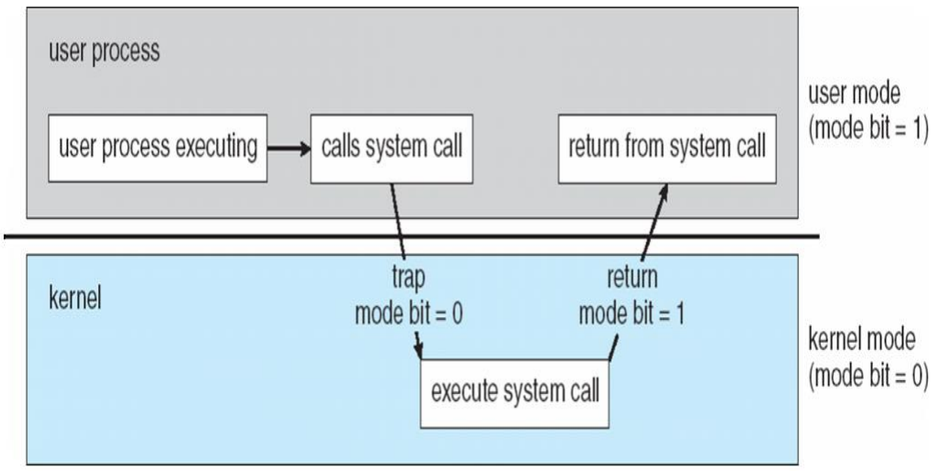
A modern computer system operates in 2 modes in order to protect the operating system code from user processes and also the code of each process from other processes.



The two modes are:

1 User mode
2 Kernel mode (supervisor mode, privileged mode or system mode)

A bit called 'mode bit' indicates the current mode. The value of mode bit is '0' for kernel mode and '1' for user mode. When the system is booted, the system runs in kernel mode. When the operating system starts the execution of any user program then the mode of system is switched to user mode. During execution of the program, if the program requests any service from the operating system by calling or invoking a system call then the mode is switched to kernel mode. After executing the system call, the mode is switched to user mode by setting mode bit to '1' before passing the control to user program.



To protect the operating system code, some of the machine instructions are designated as privileged instructions. These privileged instructions are executed only in kernel mode. The instruction used change the mode bit is an example for privileged instruction. When any privileged instruction is executed in user mode then the hardware informs to the operating system.

1. a. Explain the operating system structure and its functions.

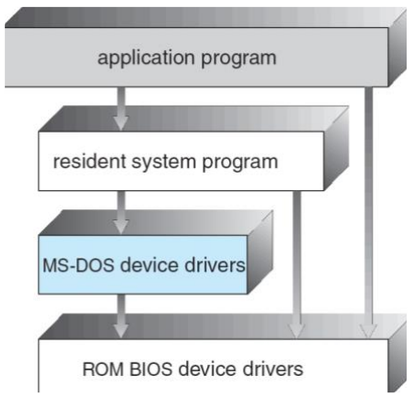
Operating System Structure

Here we discuss different ways used for designing and implementing an operating system. The better approach is implementing the functionalities of operating system through number of small modules instead of implementing all functionalities in one module. There are four approaches for developing operating systems:

1) Simple Structure Approach
2) Layered Approach
3) Microkernel Approach
4) Modules Approach

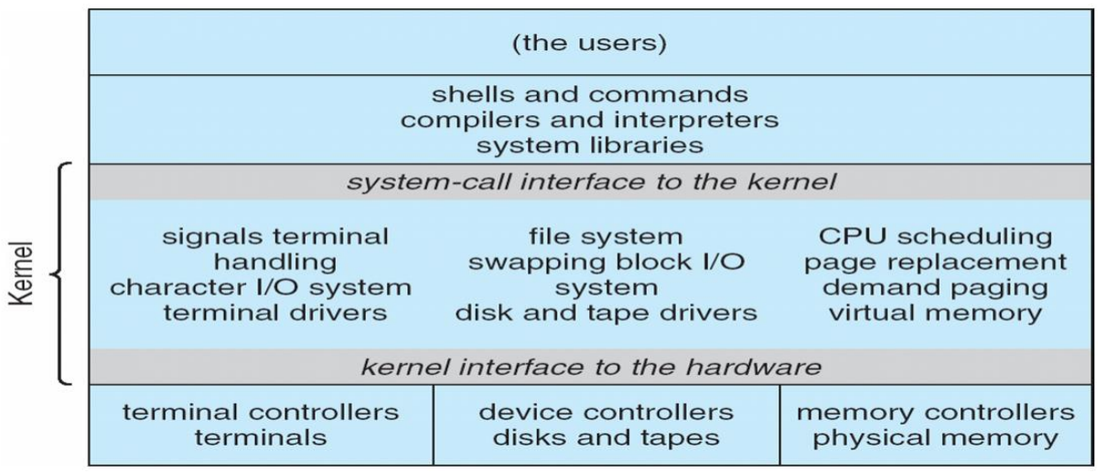
Simple Structure Approach

MS–DOS and Original UNIX operating systems were developed with simple structure. In MS-DOS, all functionalities are implemented in one module. MS-DOS is not a fully implemented operating system. The structure of MS-DOS is shown in following figure.



The application programs can directly interact with the hardware components. This direct interaction may crash the entire system when the application programs fail.

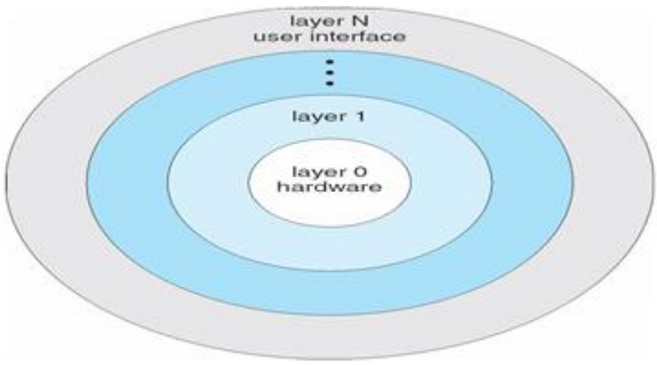
The structure of original unix is shown in the following diagram.



All functionalities are implemented in the kernel. Extension of operating system to support new functionalities is difficult. Maintenance also becomes difficult.

Layered Approach

In layered approach, the functionalities of operating system are implemented through number of layers. The following diagram shows a layered operating system. Each layer implements only few functionalities of the operating system. Each layer provides functionalities/services to the layers above on it. Each layer uses the functionalities/services provided by layers below it.



Advantages

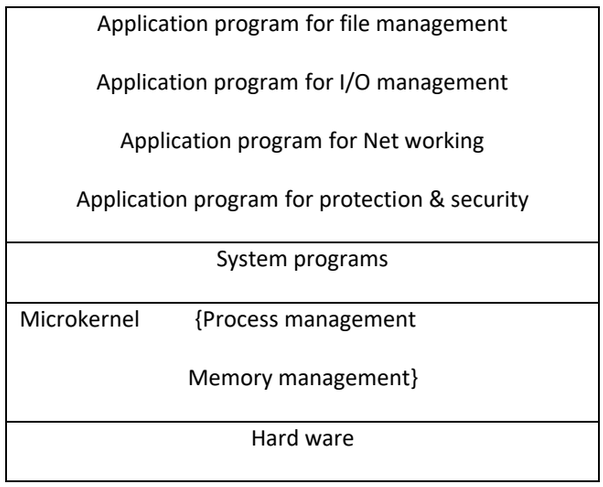
1) Easy to implement, debug and maintain.
2) Easy to extend.

Disadvantages

1) Before implementing the functionalities of operating system, the implementer has to decide the number of layers for implementing the functionalities.
2) The implementer has to decide the functionalities that will be implemented in each layer.
3) When a user program makes a request to a hardware component then that request has to be passed through number of layers. This leads to more waiting time by the user program.

Microkernel Approach

In microkernel approach, only the essential functionalities are implemented in the kernel instead of implementing all functionalities of operating system in the kernel. Remaining functionalities are implemented as either system programs or application programs. Here, the kernel is called microkernel as the size of the kernel is reduced. The following diagram shows the microkernel approach



The main function of the microkernel is providing communication between user programs and the services that are implemented as user or application programs in user space.

If any user program request for any service that has been implemented as application program then the microkernel will

1. Receive the request from the user program
2. Provides communication between the user program and the application program which is implementing the required service.

Advantage

Extension of operating system for providing new services is easy. When a new functionality is required then an application program for providing that functionality is developed and added to the list of application programs and no modification is required in the microkernel.

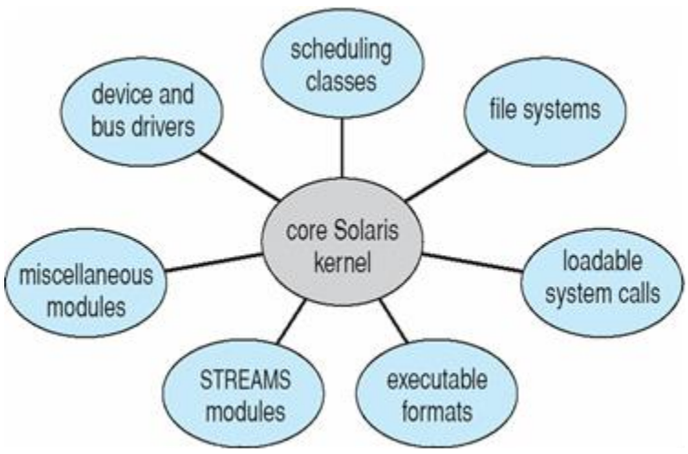
Disadvantage

Performance of the system is decreased due to more over head of the system.

Modules approach

In modules approach, a core kernel similar to microkernel is used. This core kernel implements the most essential functionalities of operating system. Remaining functionalities of operating system are implemented as separate modules. This modular approach supports dynamic linking

and loading of modules in the kernel when a request comes from any user program for any service. Most of the operating systems like LINUX, MAC OS and Solaris were developed using this approach.



Computing Environments

There are four computing environments

1) Traditional computing
2) Client-Server computing
3) Peer-to-Peer computing
4) Web-Based computing

Traditional computing

Traditional computing indicates an office environment. An office environment consists of number of personal computer systems connected to a network. A server system is also connected to this network. This server system contains the data that will be accessed by all systems in the network. To achieve portability, laptops are used in place of personal computers. Now a day, companies are using web technologies concept and creating portals. Through this portal the employees of company can access the data from anywhere.

Client-Server Computing

In client-server computing, number of client systems and one or more server systems are connected in a network. The client systems request for services from server system. The server system provides services to the client systems. There are two types of server systems

1) Compute-server system: perform actions like reading data when a client makes a request. Database server is an example for compute-server.
2) File-server system: allow clients to create, delete, read and update files.

Peer-to-Peer computing

In peer-to-peer computing, number of systems is connected through a network. Each system in the network is called a peer. Each system in the network may act as either a client or a server, depending on whether it is requesting or providing a service. In client-server computing, the server has to provide services to all clients. So, the server may block. In peer-to-peer computing, there is no block of server as many systems are providing services.

When a node joins in the network, it can begin by requesting services from or providing services to other systems in the network. When a node or system requires a service, it

broadcasts the request to other nodes or systems in the network. The nodes which provide the requested service will respond to the node which is making the request.

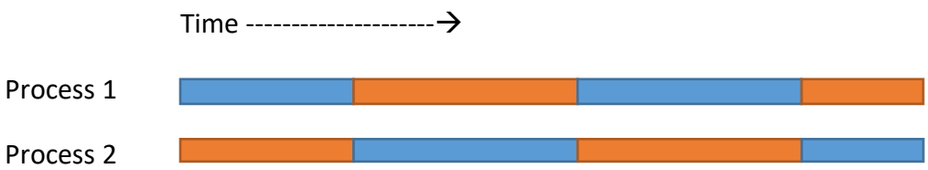
Another way to get service from other nodes is: when a system joins in the network, the system registers its service with a central system. Any node requesting for a service will contact this central system to know which nodes are providing the required service. Then, communication takes place directly between the client and the service provider.

Web-Based computing

In web-based computing, number of systems is connected through a wired or wireless network. The systems may be at different locations. One or more systems in the network may act as servers and provides services to other systems.

Interleaving in Multiprogramming

To deal with multiple processes, the operating system needs to switch control between them from time to time and accordingly save and restore their contexts. In a uniprocessor multiprogramming system, processes are interleaved in time to yield the appearance of simultaneous execution as illustrated in following figure.



Overlapping in Multitasking or Multiprocessing

Multitasking allow to execute a number of processes at a time with multiple processors. With a multiprocessor system, it is possible not only to interleave processes but also to overlap them as illustrated in following figure.