OBJECTIVES OF O.S (GOALS)

The OS has 3 main objectives.

- **Convenience.** An OS makes a computer more convenient to the user for using. (Easy-to-use commands, graphical user interface(GUI))
- **Efficiency.** An OS allows the computer system resources to be used in an efficient manner, to ensure good resource utilization efficiency, and provide appropriate corrective actions when it becomes low.
- **Ability to evolve.** An OS should be constructed in such a way as to permit the effective development, testing and introduction of new system functions without interfering with service.

Operating system performs the following functions:

1. Booting

Booting is a process of starting the computer operating system starts the computer to work. It checks the computer and makes it ready to work.

2. Memory Management

It is also an important function of operating system. The memory cannot be managed without operating system. Different programs and data execute in memory at one time. if there is no operating system, the programs may mix with each other. The system will not work properly.

3. Loading andExecution

A program is loaded in the memory before it can be executed. Operating system provides the facility to load programs in memory easily and then execute it.

4. Datasecurity

Data is an important part of computer system. The operating system protects the data stored on the computer from illegal use, modification or deletion.

5. Disk Management

Operating system manages the disk space. It manages the stored files and folders in a proper way.

6. Process Management

CPU can perform one task at one time. if there are many tasks, operating system decides which task should get the CPU.

7. Device Controlling

operating system also controls all devices attached to computer. The hardware devices are controlled with the help of small software called device drivers.

8. Printingcontrolling

Operating system also controls printing function. It a user issues two print commands at a time, it does not mix data of these files and prints them separately.

9. Providing interface

It is used in order that user interface acts with a computer mutually. User interface controls how you input data and instruction and how information is displayed on screen. The operating system offers two types of the interface to the user:

1. Graphical-line interface: It interacts with of visual environment to communicate with the computer. It uses windows, icons, menus and other graphical objects to issuescommands.

2. Command-lineinterface:itprovidesaninterfacetocommunicatewiththecomputerbytyping commands.

Computer System Architecture

Computer architecture means construction/design of a computer. A computer system may be organized in different ways. Some computer systems have single processor and others have multiprocessors. So based on the processors used in computer systems, they are categorized into the following systems.

1. Single-processor system

2. Multiprocessor system

3. Clustered Systems:

1. Single-Processor Systems:

Some computers use only one processor such as microcomputers (or personal computers PCs). On a single-processor system, there is only one CPU that performs all the activities in the computer system. However, most of these systems have other special purpose processors, such as I/O processorsthatmovedataquicklyamongdifferentcomponentsofthecomputers.Theseprocessors execute only a limited system programs and do not run the user program. Sometimes they are managed by the operating system. Similarly, PCs contain a special purpose microprocessor in the keyboard, which converts the keystrokes into computer codes to be sent to the CPU. The use of special purpose microprocessors is common in microcomputer. But it does not mean that this system is multiprocessor. A system that has only one general-purpose CPU, is considered as single-processor system.

2. Multiprocessor Systems:

In multiprocessor system, two or more processors work together. In this system, multiple programs (more than one program) are executed on different processors at the same time. This type of processing is known as multiprocessing. Some operating systems have features of multiprocessing. UNIX is an example of multiprocessing operating system. Some versions of Microsoft Windows also support multiprocessing.

Multiprocessor system is also known as parallel system. Mostly the processors of multiprocessor system share the common system bus, clock, memory and peripheral devices. This system is very fast in data processing.

Types of Multiprocessor Systems:

The multiprocessor systems are further divided into two types;

(i). Asymmetric multiprocessing system

(ii). Symmetric multiprocessing system

(i) Asymmetric Multiprocessing System(AMS):

The multiprocessing system, in which each processor is assigned a specific task, is known as Asymmetric Multiprocessing System. For example, one processor is dedicated for handling user's requests, one processor is dedicated for running application program, and one processor is dedicated for running image processing and so on. In this system, one processor works as master processor, while other processors work as slave processors. The master processor controls the operations of system. It also schedules and distributes tasks among the slave processors. The slave processors perform the predefined tasks.

(ii) Symmetric Multiprocessing System(SMP):

The multiprocessing system, in which multiple processors work together on the same task, is known as Symmetric Multiprocessing System. In this system, each processor can perform all types of tasks. All processors are treated equally and no master-slave relationship exists between the processors.

For example, different processors in the system can communicate with each other. Similarly, an I/O can be processed on any processor. However, I/O must be controlled to ensure that the data reaches the appropriate processor. Because all the processors share the same memory, so the input data given to the processors and their results must be separately controlled. Today all modern operating systems including Windows and Linux provide support for SMP.

It must be noted that in the same computer system, the asymmetric multiprocessing and symmetric multiprocessing technique can be used through different operating systems.

3. Clustered Systems:

Clustered system is another form of multiprocessor system. This system also contains multiple processors but it differs from multiprocessor system. The clustered system consists of two or more individual systems that are coupled together. In clustered system, individual systems (or clustered computers) share the same storage and are linked together ,via Local Area Network (LAN).

A layer of cluster software runs on the cluster nodes. Each node can monitor one or more of the other nodes over the LAN. If the monitored machine fails due to some technical fault (or due to other reason), the monitoring machine can take ownership of its storage. The monitoring machine can also restart the applications that were running on the failed machine. The users of the applications see only an interruption of service.

Types of Clustered Systems:

Like multiprocessor systems, clustered system can also be of two types

(i). Asymmetric Clustered System

(ii). Symmetric Clustered System

(i). Asymmetric ClusteredSystem:

In asymmetric clustered system, one machine is in hot-standby mode while the other machine is running the application. The hot-standby host machine does nothing. It only monitors the active server. If the server fails, the hot-standby machine becomes the active server.

(ii). Symmetric ClusteredSystem:

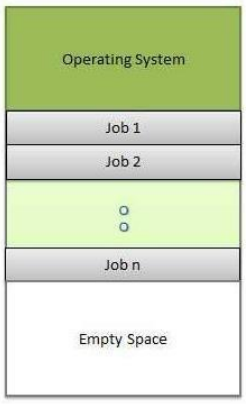
In symmetric clustered system, multiple hosts (machines) run the applications. They also monitor each other. This mode is more efficient than asymmetric system, because it uses all the available hardware. This mode is used only if more than one application be available to run.

Operating System – Structure

Multiprogramming

When two or more programs are residing in memory at the same time, then sharing the processor is referred to the multiprogramming. Multiprogramming assumes a single shared processor. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

Following figure shows the memory layout for a multiprogramming system.



Operating system does the following activities related to multiprogramming.

- The operating system keeps several jobs in memory at a time.
- This set of jobs is a subset of the jobs kept in the job pool.
- The operating system picks and begins to execute one of the job in the memory.
- Multiprogramming operating system monitors the state of all active programs and system resources using memory management programs to ensures that the CPU is never idle unless there are no jobs

Advantages

- High and efficient CPU utilization.
- User feels that many programs are allotted CPU almost simultaneously.

Disadvantages

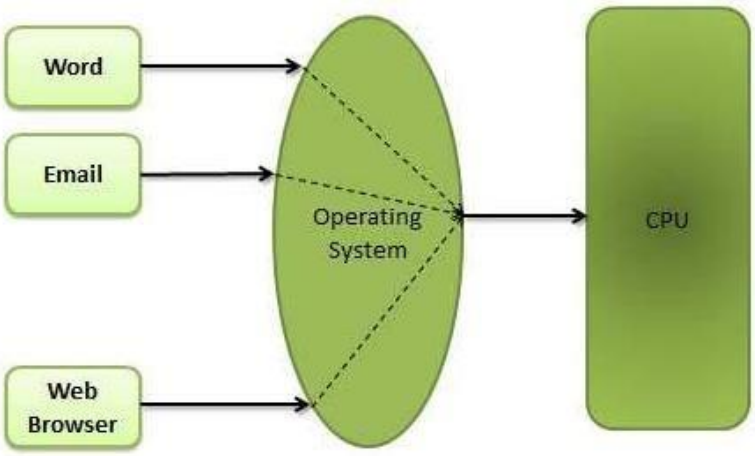
- CPU scheduling is required.
- To accommodate many jobs in memory, memory management is required.

2)

Multitasking

Multitasking refers to term where multiple jobs are executed by the CPU simultaneously by switching between them.Switches occur so frequently that the users may interact with each program while it is running. Operating system does the following activities related to multitasking.

- The user gives instructions to the operating system or to a program directly, and receives an immediate response.
- Operating System handles multitasking in the way that it can handle multiple operations / executes multiple programs at a time.
- Multitasking Operating Systems are also known as Time-sharing systems.
- These Operating Systems were developed to provide interactive use of a computer system at a reasonable cost.
- A time-shared operating system uses concept of CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared CPU.
- Each user has at least one separate program in memory.



- A program that is loaded into memory and is executing is commonly referred to as a process.
- When a process executes, it typically executes for only a very short time before it either finishes or needs to perform I/O.

- Since interactive I/O typically runs at people speeds, it may take a long time to completed. During this time a CPU can be utilized by another process.
- Operating system allows the users to share the computer simultaneously. Since each action or command in a time-shared system tends to be short, only a little CPU time is needed for each user.
- As the system switches CPU rapidly from one user/program to the next, each user is given the impression that he/she has his/her own CPU, whereas actually one CPU is being shared among many users.

Operating-system Operations

1)Dual-Mode Operation·

In order to ensure the proper execution of the operating system, we must be able to distinguish between the execution of operating-system code and user defined code. The approach taken by most computer systems is to provide hardware support that allows us to differentiate among various modes of execution.

At the very least we need two separate modes of operation.user mode and kernel mode.
A bit, called the mode bit is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1).with the mode bit we are able to distinguish between a task that is executed on behalf of the operating system and one that is executed on behalf of the user, When the

computer system is executing on behalf of a user application, the system is in user mode. However,when a user application requests a service from the operating system (via a.. system call), it must transition from user to kernel mode to fulfill the request.

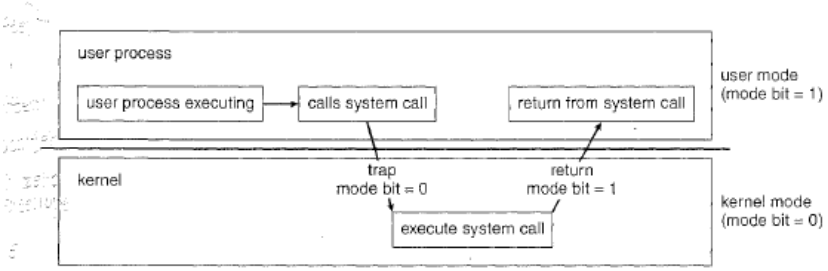


Figure Transition from user to kernel mode.

At system boot time, the hardware starts in kernel mode. The operating system is then loaded and starts user applications in user mode. Whenever a trap or interrupt occurs, the hardware switches from user mode to kernel mode (that is, changes the state of the mode bit to 0). Thus, whenever the operating system gains control of the computer, it is in kernel mode. The system always switches to user mode (by setting the mode bit to 1) before passing control to a user program.
The dual mode of operation provides us with the means for protecting the operating system from errant users-and errant users from one another. We accomplish this protection by designating some of the machine instructions that may cause harm as privileged instructions.the hardware allows privileged instructions to be executed only in kernel mode.If an attempt is made to execute a privileged instruction in user mode, the hardware does not execute the instruction but rather treats it as illegal and traps it to the operating system.The instruction to switch to kernel mode is an example of a privileged instruction. Some other examples include I/0 controt timer management and interruptmanagement.

2)

Timer

We must ensure that the operating system maintains control over the CPU. We must prevent a user program from getting stuck in an infinite loop or not calling system services and never returning control to the operating system. To accomplish this goal, we can use a **timer**. A timer can be set to interrupt the computer after a specified period.

Before turning over control to the user, the operating system ensures that the timer is set to interrupt. If the timer interrupts, control transfers automatically to the operating system, which may treat the interrupt as a fatal error or may give the program more time. Clearly, instructions that modify the content of the timer are privileged.

Thus, we can use the timer to prevent a user program from running too long.

EVOLUTION OF OPERATING SYSTEMS

Operating system and computer architecture have had a great deal of influence on each other. Operating systems were developed mainly to facilitate the use of the hardware and to bring it to the best advantage. Here we will briefly make a sketch of the evolutionary path of OS development.

Serial Processing

Before 1950's the programmers directly interact with computer hardware, there was no OS at that time. If the programmer want to execute the program on those days, he has to follow some serial steps:

- Type the program on punched card.
- Convert the punched card to card reader.
- Submit to the computing machine, if any error in the program, the error condition was indicated by lights.
- The programmer examine the registers and main memory to identify the cause of error.
- Take the output on the printers.
- Then the programmer is ready for the next program.

This type of processing is difficult for users, it takes much time and next program should wait for the completion of previous one. The programs are submitted to the machine one after the other. So, this method is called as "Serial processing".

Batch Processing

In olden days(before 1960's), it is difficult to execute a program using computer. Because the computer is located in different rooms, one room for card reader and one for executing the program and another room for printing the output. The user or machine operator, running between these three rooms to complete a job. This problem was solved by batch processing system.

In batch processing technique similar type of jobs batch together and execute at a time. The operator carries the group of jobs at a time from one room to another. Therefore the programmer need not run between these three rooms several times.

The batch processing had an advantage .In that for one batch, the compiler, assembler, the loader etc had to be loaded only once, thus reducing the setup time to some extent. For example, FORTRAN programs were grouped together as one batch say batch 1, the PASCAL programs into another batch say Batch 2, the COBOL programs into another batch say Batch 3, and so on. Now the operator can arrange for the execution of these source programs which has been batched together one by one. After the execution of batch1 was over, the operator would load the compiler, assembler and loader, etc for the batch 2 and so on.

Setup time for batch 1	Runtime for batch 1	Setup time for batch 2	Runtime for batch 2	...	...	...

Fig. *Batch Processing*

The main advantage of batch processing is setup time will be reduced to a large extent, but the disadvantage is that the CPU is idle for the time in between two batches.

If the programs were not batched up together, the set up time would be much more higher.

Setup time for program 1	Runtime for program 1	Setup time for program 2	Runtime for program 2	...	...	...

Multiprogramming

Multiprogramming is a rudimentary form of parallel processing in which several programs are run at the same time on a uniprocessor. Since there is only one processor, there can be no true simultaneous execution of different programs. Instead the processor executes part of one program, then part of another, and so on. But to the user it appears that all programs are executing at the same time.

In multiprogramming, number of processes are reside in main memory at a time. The OS picks and begins to execute one of the jobs in the main memory. For example, consider the main memory consisting of 5 jobs at a time, the CPU executes one by one.

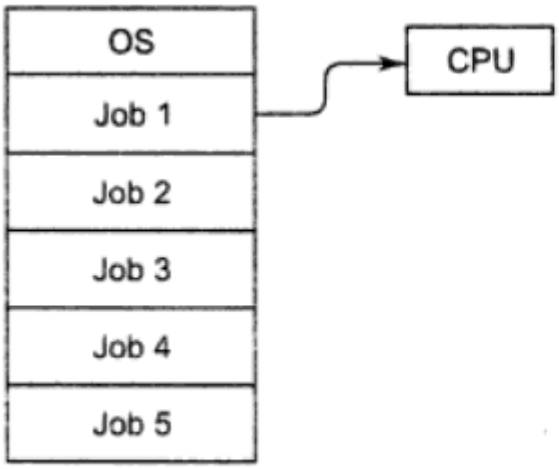


Fig. *Multiprogramming*

In non-multiprogramming system, the CPU can execute only one program at a time, if the running program waiting for any I/O device, the CPU becomes idle, so it will effect on the performance of the CPU.

But in multiprogramming environment, any I/O wait happened in a process, then the CPU switches from that job to another job in the job pool. If enough jobs could be held in main memory at once, the CPU is not idle at any time.

For Example: The idea is common in other life situations. The doctor does not have only one patient at a time, number of patients reside in the hospital under treatment. If the doctor has enough patients a doctor never needs to be idle.

Time Sharing Systems

Multiprogramming features were superimposed on batch processing to ensure good utilization of CPU but from the point of view of a user the service was poor as the response time, *i.e*, the time elapsed between submitting a job and getting the results was unacceptably high. Development of interactive terminals changed the scenario. Computation became an on-line activity. A user could provide inputs to a computation from a terminal and could also examine the output of the computation on the same terminal. Hence the response time needed to be drastically reduced. This was achieved by storing programs of several users in memory and providing each user a slice of time on CPU to process his/her program.

Time sharing or multitasking is a logical extension of multiprogramming. In time sharing environment, a number of jobs are loaded on to the memory and a number of users are communicating with the computer through different terminals. The OS allocates a fixed time interval (TIME SLICE) to each program in memory. Thus each program in memory is executed for a fixed interval of time.

As soon as the time allotted for a particular program is completed, the CPU starts executing the next program. This process is continued till all the programs in the memory are executed. A program may need number of time slices for its complete execution. Although the computer system is executing one job at a time, due to the speed of the CPU, every user on a terminal has the feeling that his program that is being executed continuously, because, after every time slice, the user gets a response from the computer. The user on the terminal is communicating with his running program, and is able to debug and experiment with his program.

Thus, the OS for a time sharing computer system has all the capabilities of a multiprogramming OS, but along with an additional capacity of allocating a fixed time slice of CPU to each program.

- Main advantage of time sharing system is efficient CPU utilization.
- The user can interact with the job while it is executing, but it is not possible in batch systems.

Personal-ComputerSystems(PCs)

A personal computer (PC) is a small, relatively inexpensive computer designed for an individual user. In price, personal computers range anywhere from a few hundred dollars to thousands of dollars. All are based on the microprocessor technology that enables manufacturers to put an entire CPU on one chip.

At home, the most popular use for personal computers is for playing games. Businesses use personal computers for word processing, accounting, desktop publishing, and for runningspreadsheet and database managementapplications.

Parallel Systems

Almost all the systems are uni-processor systems *i.e.*, they have only one CPU. Systems in which there are more than one CPU is called as Multi-processor systems. These systems have been developed to enhance the computing power of a computing system, and the features of this system is that, they share the memory, bus and the peripheral devices. These systems are referred as *"Tightly coupled systems"*. A system consisting of more than one processor and it is a tightly coupled, then the system is called "Parallel system".

In parallel systems number of processors executing their jobs in parallel (simultaneous process). Multi-processor systems are divided into following categories:

- Symmetric
- Asymmetric

In symmetric multi-processing, each processor runs a shared copy of operating system . The processors can communicate with each other and execute these copies concurrently. Thus, in a symmetric system, all the processors share an equal amount of load .Encore's version of UNIX for the Multimax computer is an example of symmetric multiprocessing. In this system various processors execute copies of UNIX operating system, thereby executing M processes if there are M processors.

Asymmetric multi-processing is based on the principle of master-slave relationship. In this system, one of the processors runs the operating system and that processor is called the master processor. Other processors run user processes and are known as slave processors. In other words, the master processor controls, schedules and allocates the task to the slave processors. Asymmetric multi-processing is more common in extremely large systems, where one of the time consuming tasks is processing I/O requests. In the asymmetric systems the processors do not share the equal load.

Advantages:

1. It results in saving money compared to the stand alone systems, since CPU'S can share memory, bus and peripherals.
2. Throughput can be increased
3. They increase the reliability.

Since there are more than one CPU, the failure of one or more of the CPU does not halt the entire system, but only slows down the work. For example, if there are five processors, all the five working together gives full efficiency. If two CPU's fail, then the system still works but only at 60% efficiency. This indicates increased aspect of reliability compared to stand alone systems.

Distributed Systems

A recent trend in computer system is to distribute computation among several processors. The processors in distribute system may vary in size and function, and referred by a number of different names such as sites, nodes, computers and so on depending on the context.

A distributed system is basically a collection of autonomous (independent by function) computer systems which co-operate with one another through their hardware and software interconnections.

In distributed systems, the processors cannot share memory or time, each processor has its own local memory. The processors communicate with one another through various communication lines such as high speed buses .These systems are also called as *"Loosely Coupled systems"*.

Distributed system = Network + Transparency(Invisible)

Advantages

1. **Resource sharing:** If a number of sites connected by a high speed communication lines, it is possible to share the resources from one site to another site.
   For example, S_1 and S_2 are two sites, these are connected by some communication lines, the site S_1 having the printer, but S_2 does not having the printer. Then the system can use the printer at S_1 without moving from S_2 to S_1.Therefore resource sharing is possible in distributed systems.
2. **Computation speedup:** A big computation is partitioned into number of partitions, these sub-partitions run concurrently in distributed systems.
   For example, site S_1 need to execute a big computation, this computation is divided into sub computations and these are executed by some other machines in different sites.
3. **Reliability:** If a resource or a system failed in one site due to technical problems. We can use other systems or other resources in some other sites.
4. **Communication:** Distributed systems provides communication which is not at all possible, that much in a centralized system. For Example, E-mail

REAL-TIME OS

In a time shared computer system, generally the computer response time is of the order of 0.5 to 2 seconds, which means a user will get computers attention after this much of time. Longer response times may be irritating but not hazardous.

However a real-time OS is needed for the computer systems controlling a process or a real time situation, such as a machine or a satellite. In this case two important points to be noticed are:

- The OS should provide for interactive processing.
- The response time should be very small.

The sensors bring in the data from a device, the OS instructs the computer to analyze the data and send appropriate signals back to the device. Any delay on the part of the computer system or the OS can be catastrophic. Thus, the real-time OS have to work strict time limits and have to be quick. Apart from this, these systems must be highly reliable to avoid failure of the system being controlled.

Here the main job of OS is instant handling of the signals or interrupts sent by the device which is being controlled by the computer system.

Real-time systems are systems that have in-built characteristics as supplying immediate response. A primary objective of the real-time system is to provide quick response time. User convenience and resource utilization are of secondary concern to real-time systems.

Real time System is of two types:

- **Hard real-time**
  - Guarantees that critical tasks complete within time.
  - All the delays in the system are bounded.
  - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - Conflicts with time-sharing systems, not supported by general-purpose operating systems.
- **Soft real-time**
  - Critical time tasks gets priority over other tasks, and retails that priority until it completes.
  - Limited utility in industrial control of robotics
  - Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.

Special purpose systems

a)Real-Time EmbeddedSystems

These devices are found everywhere, from car engines and manufacturing robots to DVDs and microwave ovens. They tend to have very specific tasks.

They have little or no user interface, preferring to spend their time monitoring and managing hardware devices, such as automobile engines and robotic arms.

b)Multimedia Systems

Most operating systems are designed to handle conventional data such as text files, programs, word-processing documents, and spreadsheets. However, a recent trend in technology is the incorporation of multimedia data into computer systems. Multimedia data consist of audio and video files as well as conventional files. These data differ from conventional data in that multimedia data-such as frames of video-must be delivered (streamed) according to certain time restrictions (for example, 30 frames per second). Multimedia describes a wide range of applications in popular use today. These include audio files such as MP3, DVD movies, video conferencing, and short video clips of movie previews or news stories downloaded over the Internet. Multimedia applications may also include live webcasts (broadcasting over the World Wide Web)

c)HandheldSystems

Handheld Systems include personal digital assistants (PDAs, cellular telephones. Developers of handheld systems and applications face many challenges, most of which are due to the limited size of such devices. For example, a PDA is typically about 5 inches in height and 3 inches in width, and it weighs less than one-half pound. Because of their size, most handheld devices have small amounts of memory, slow processors, and small display screens.

Operating System services

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

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

Following are few common services provided by operatingsystems.

- Programexecution
- I/Ooperations
- File Systemmanipulation
- Communication
- ErrorDetection
- ResourceAllocation
- Protection

Programexecution

Operating system handles 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 programmanagement.

- Loads a program intomemory.
- Executes theprogram.
- Handles program'sexecution.
- Provides a mechanism for processsynchronization.
- Provides a mechanism for processcommunication.
- Provides a mechanism for deadlock handling.

I/OOperation

I/O subsystem comprised of I/O devices and their corresponding driver software. Drivers hides the peculiarities of specific hardware devices from the user as the device driver knows the peculiarities of the specificdevice.

Operating System manages the communication between user and device drivers. Following are the major activities of an operating system with respect to I/O Operation.

- I/O operation means read or write operation with any file or any specific I/Odevice.
- Program may require any I/O device whilerunning.
- Operating system provides the access to the required I/O device whenrequired.

Filesystemmanipulation

A file represents a collection of related information. Computer can store files on the disk (secondary storage), for long term storage purpose. Few examples of storage media are 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 accessmethods.

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 afile.
- The operating system gives the permission to the program for operation onfile.
- Permission varies from read-only, read-write, denied and soon.
- Operating System provides an interface to the user to create/deletefiles.
- Operating System provides an interface to the user to create/deletedirectories.
- 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, operating system manages communications between processes. Multiple processes with one another through communication lines in the network.

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 betweenthem.
- The both processes can be on the one computer or on different computer but are connected through computernetwork.
- Communication may be implemented by two methods either by Shared Memory or by MessagePassing.

Errorhandling

Error can occur anytime and anywhere. 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.

- OS constantly remains aware of possibleerrors.
- OS takes the appropriate action to ensure correct and consistentcomputing.

ResourceManagement

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.

- OS manages all kind of resources usingschedulers.
- CPU scheduling algorithms are used for better utilization of CPU.

Protection

Considering a computer systems having multiple users the concurrent execution of multiple processes, then the various processes must be protected from each another's activities.

Protection refers to mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer systems. Following are the major activities of an operating system with respect to protection.

- OS ensures that all access to system resources iscontrolled.
- OS ensures that external I/O devices are protected from invalidaccess attempts.
- OS provides authentication feature for each user by means of a password.

Accounting

The operating system must keep track of :

Which users are requesting for what resource.

Which resource is allocated to which process

What are resources that are not allocated.

User operating system interface

1)Command interpreter

The main function of the command interpreter is to get and execute the next user-specified command. Many of the commands given at this level manipulate files: create, delete, list, print, copy, execute, and so on.

2) GUI

- User-friendly desktopinterface
- Usually mouse, keyboard, andmonitor
- Icons represent files, programs, actions,etc
- Various mouse buttons over objects in the interface cause various actions(provide information, options, execute function, open directory (known as afolder)

System calls

As we know that for performing any Operation as user must have to specify the Operation which he wants to Operate on the Computer. We can say that For Performing any Operation a user must have to Request for a Service from the System. For Making any Request a user will prepare a Special call which is also known as the **System Call**.

The System Call is the Request for Running any Program and for Performing any Operation on the System. When a user First Time Starts the System then the **System is in the user Mode and When he request For a Service then the User Mode will be Converted into the Kernel Mode** Which just Listen the Request of the user and Process the Request and Display the Results those are Produced after the Processing. When a user Request for Opening any Folder or When a Moves his Mouse his Mouse ontheScreen,thenthisiscalledastheSystemcallwhichheisusingforperforminganyOperation.

To read data from one file and copy to another file. For this first input to the program is two file names (*i.e.*, input and output files). These file names can be specified in many ways.

1. Ask the user for the names of the two files.
2. Specify the file names with control statements.
3. Use the mouse to select the source name from a list of files and fill the destination.

If there is already an output file with the same name as input file then the program may abort using one "system call" or delete the existing file by using another "system call" and create a new file

using one more "system call". Finally after the entire file is copied, the program may close both files using another "system call". Write a message to the console (more "system calls") and terminate normally by a last "system call".

Types of SystemCalls

Types of System Call

A system call is made using the system call machine language instruction. System calls can be grouped into five major categories.

1. File management
2. Interprocess communication
3. Process management
4. I/O device management
5. Information maintenance.

a. Create : Create a new file and open it.
b. Delete : Delete a file.
c. Open : Open a file to read or write.
d. Close : Close a file, indicating that file is no longer using it.
e. Read : Read a byte from an open file.
f. Write : Write a byte to an open file.
g. Stat : Get information about a file.
h. Unlink : Remove a file from a directory.
i. Get file attribute and set file attribute - Attribute includes file name, file type, protection codes and accounting information.

2. Interprocess communication :

a. Create message queue : Create a queue to hold message.
b. Send message : Send a message to a message queue.
c. Receive message : Receive a message from a message queue.
d. Close connection : Terminates the communication.

3. Process management system call :

a. Create process : Create a new process.
b. Terminate process : Terminate the process making the system call.
c. Wait : Wait for another process to exit.
d. Fork : Create a duplicate of the process making a
e. End : Halt the process execution normally.
f. Abort : Halt the process execution abnormally.
g. Load : Load the process into memory.
h. Execute : Execute the loaded process.
i. Get process attributes and set process attributes.
j. Allocate and free memory.

4. I/O device system calls :

a. Request device : To ensure exclusive use of device.
b. Release device : Release the device after finished with the device.
c. Read, write : Same as file system call.
d. Stat : Get information about an I/O device.

5. Information maintenance :

a. Get time and date
b. Set time and date
c. Set process, file or device attributes.
d. Get process, file or device attributes
e. Get system data
f. Set system data.

The operating system provides a set of operations which are called system calls. A **system call interface** is the description of the set of system calls implemented by the operating system.

systemprograms

System Programs

System programs provide basic functioning to users so that they do not need to write their own environment for program development (editors, compilers) and program execution (shells). In some sense, they are bundles of useful system calls. Different operating systems provide the services which have been discussed above in different ways. The basic methods used are with the help of System calls or with system programs. The system programs can be classified into the following groups:

- File management – These system program facilitate creation, deletion, copying, renaming, printing and other related file operations.
- Program execution – After compiling the program, for execution of the program it must be loaded into memory. The system programs developed for this purpose helps in loading and resolving links for the program ready for execution.
- Status information – These system programs helps in providing the information of status of the system, such as available memory or disk space, other status information.
- Support for Programming Language – Common programming languages compilers, interpreters such as C, C++, and Java etc., are often provided as part of the operating system.
- File Modification – The support for text editors for creation and storing of files on disk or tape etc. are supported by the operating system.
- Communications – These system programs are responsible for providing environment for establishing communication between different users, processes and different computer systems.

Operating system design and implementation

Designing a system involves defining goals and specifications of a system.the designing of system is dependent on type of hardware that supports the type of operating system like batch,time sharing,single-user,multi-user,realtime,etc.

The requirements are categorized into two:

- Usergoals
- Systemgoals

The goals of user are the system should be convenient to use,easy to learn,easy to use,reliable,secure,fast.
The goal of the system ,who areresponsible to develop and maintain it are thesystemi.e.operating system must be easy to design,easy to implement,easy to maintain,feasible error free,efficient,etc.

Mechanisms and policies

Mechanisms determine how to do something.policy determine what should be done.
Example:CPU protection the mechanism is adjusting the timer,policy is the value to be set for the timer.
The difference between policy and mechanism is flexibility.policy can change from time to time.the value of the timer is different for different type of operating system.mechanism never changes as policy.

Implementation

Once operating system is designed it has to be implemented.early operating system were written in assembly language.

Now a days operating system are written in high level languages.

UNIX OS written in C Language,except 900 lines of code written in assembly language.

The advantage in high level language is code can be written faster,easy to understand,debug.

MS-DOS OS was developed in 8088 assembly language can run on intel CPU's.but UNIX OS developed in C can run on any CPU.

OS Structure

An operating system might have many structure. According to the structure of the operating system; operating systems can be classified into many categories.

Some of the main structures used in operating systems are:

1. **Monolithic architecture of operating system**

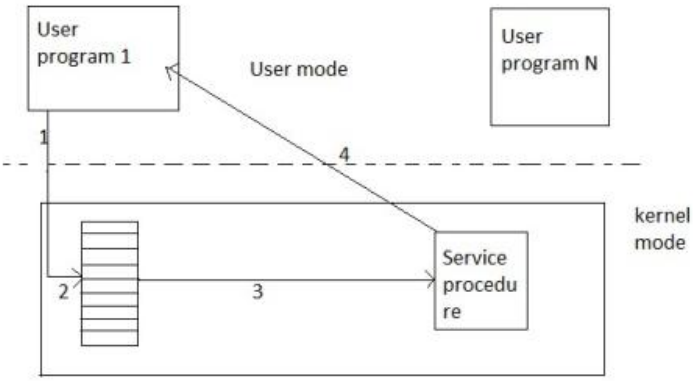


fig:- monolithic structure of os

monolithic sturucture of operating system

It is the oldest architecture used for developing operating system. Operating system resides on kernel for anyone to execute. System call is involved i.e. Switching from user mode to kernel mode and transfer control to operating system shown as event 1. Many CPU has two modes, kernel mode, for the operating system in which all instruction are allowed and user mode for user program in which I/O devices and certain other instruction are not allowed. Two operating system then examines the parameter of the call to determine which system call is to be carried out shown in event 2. Next, the operating system index's into a table that contains procedure that carries out system call. This operation is shown in events. Finally, it is called when the work has been completed and the system call is finished, control is given back to the user mode as shown in event 4.

2. **Layered Architecture of operating system**

The layered Architecture of operating system was developed in 60's in this approach; the operating system is broken up into number of layers. The bottom layer (layer 0) is the hardware layer and the highest layer (layer n) is the user interface layer as shown in the figure.

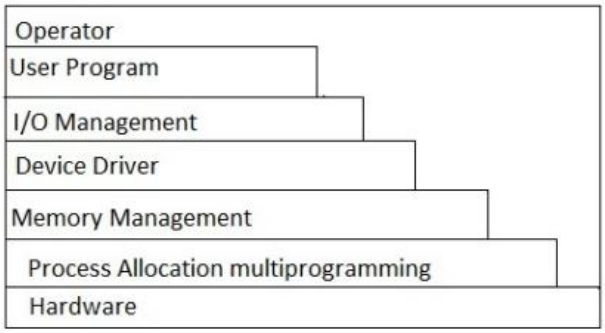


fig:- layered Architecture

layered architecture

The layered are selected such that each user functions and services of only lower level layer. The first layer can be debugged wit out any concern for the rest of the system. It user basic hardware to implement this function once the first layer is debugged., it's correct functioning can be assumed while the second layer is debugged & soon . If an error is found during the debugged of particular layer, the layer must be on that layer, because the layer below it already debugged. Because of this design of the system is simplified when operating system is broken up into layer.

Os/2 operating system is example of layered architecture of operating system another example is earlier version of Windows NT.

The main disadvantage of this architecture is that it requires an appropriate definition of the various layers & a careful planning of the proper placement of the layer.

3)Microkernel

Layered OS are less efficient.Designers of the OS started adding more features,the kernel became 'too heavy ' and unmanageable. These obstacles gave birth to a totally new approach for designing Operating Systems-the Microkernel approach. Microkernel has become a real buzzword in the world of Operating Systems today with more and more Operating Systems subscribing to this approach.

The basic concept of a Microkernel Operating Systems is very easy to understand. In this approach, the kernel provides only the most essential Operating System functions like process management, communication primitives and low-level memory management. System programs or user level programs, implemented outside the kernel, provide the remaining Operating System services. These programs are known as *servers*. As a result, the size of the kernel reduces dramatically, making it a Microkernel. The application programs and various servers communicate with each other using messages that pass through the Microkernel. The Microkernel validates the messages, passes them between the various modules of the Operating System and permits access to the hardware. Figure illustrates the working of a Microkernel Operating System. The Mach Operating System developed at the Carnegie Mellon University in the 1980s was the first Operating System built using the Microkernel approach. Andrew Tanenbaum's Minix is another example of a Microkernel Operating System.

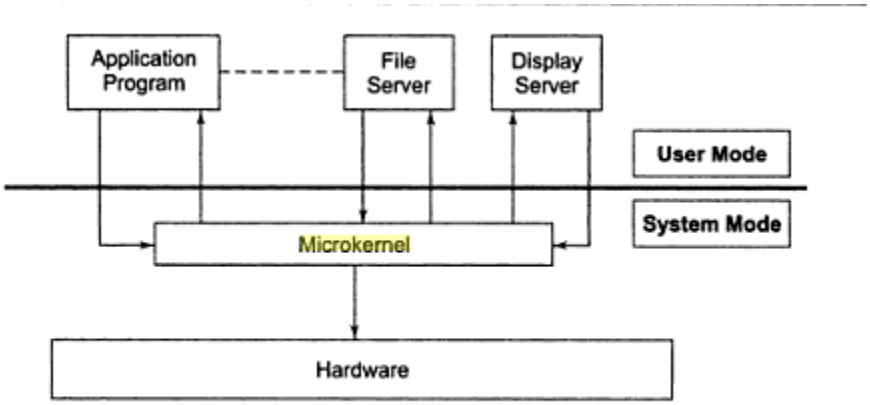


Fig. Microkernel Operating System

VirtualMachines

The concept of **virtual machine** came about in the 1960s. The background to this is quite interesting.

IBM had developed the System/360 operating system, which had become quite popular on the Mainframe computers. However, the major concern regarding this operating system was that it was batch-oriented in nature. There was no concept of online computing or timesharing. Users were increasingly feeling a need of timesharing, since only batch processing was not adequate.

In order add the timesharing features to System/360, IBM appointed a dedicated team, which started to work on a solution to this problem. This team came up with a new operating system called as TSS/360, which was based on the System/360, but also had timesharing features. Although, technically this solution was acceptable, as it turned out, the development of TSS/360 took a lot of time, and when it finally arrived, people thought that it was too bulky and *heavy*. Therefore, a better solution was warranted.

Soon, IBM came up with another operating system, called as CP/CMS, which was later renamed as VM/370. The VM/370 operating system is quite interesting. It contains a *virtual machine monitor*. The term *virtual machine* indicates a machine (i.e. a computer), which does not physically exist, and yet, makes the user believe that there is a machine. This virtual machine monitor runs on the actual hardware, and performs the multiprogramming functions. The idea is shown in Fig. Here, we assume that three application programs A, B and C are executing with their own operating systems (again A, B and C, shown as Virtual Machine A, B and C, respectively).

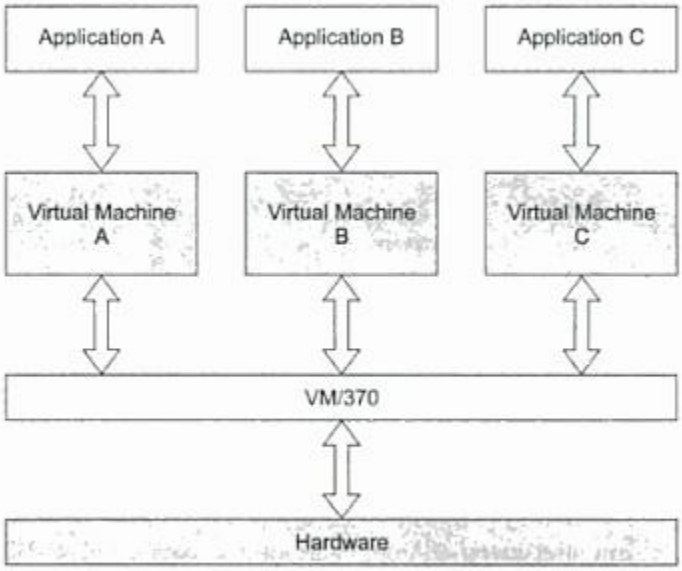


Fig. Virtual Machine (VM) concept

The virtual machine, in this case, is an exact copy of the hardware. In other words, it provides support for the kernel/user mode, input/output instructions, interrupts, etc. What significance does this have? It means that there can actually be more than one Operating System running on the computer! The way this works as follows:

1. Each application program is coded for one of the available operating systems. That is, the programmer issues system calls for a particular operating system.
2. The system call reaches its particular operating system, from all those available (depending on which operating system the programmer wants to work with).
3. At this stage, the system call of the program's operating system is mapped to the system call of VM/370 (i.e. the actual system call to be executed on the real hardware).

The virtual machine now makes the actual system call, addressed to the physical hardware.