Unit-1

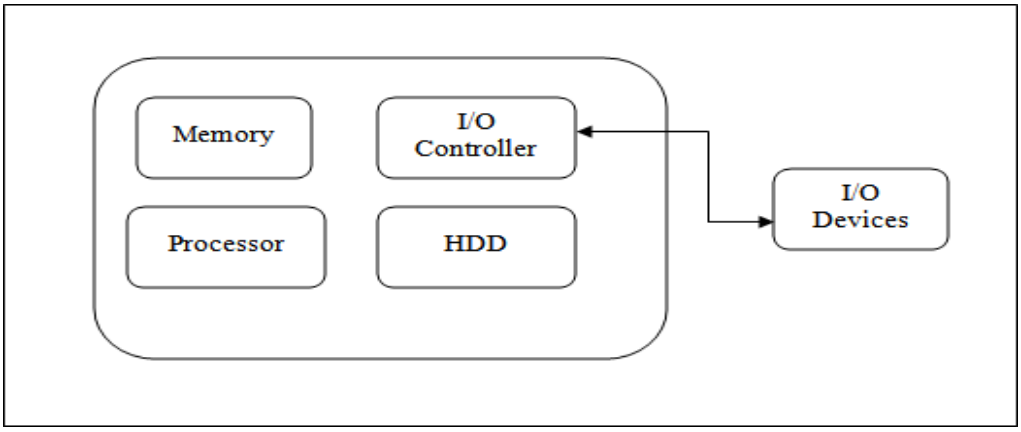
Overview of Operating Systems

Operating System:

An operating system is a program which manages all the computer hardware. It provides the base for application program and acts as an intermediary between a user and the computer hardware.

- The operating system has two objectives such as:
  - Firstly, an operating system controls the computer's hardware.
  - The second objective is to provide an interactive interface to the user and interpret commands so that it can communicate with the hardware.

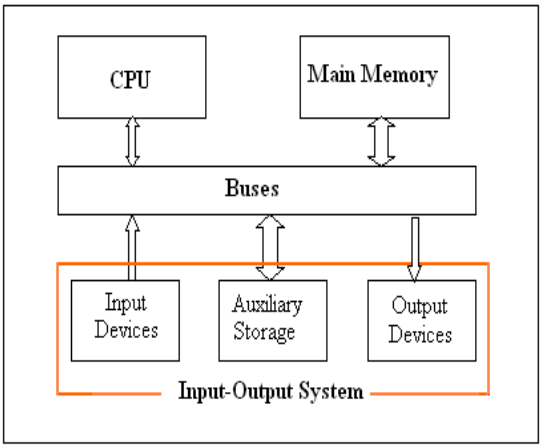
Managing Hardware



The prime objective of operating system is to manage & control the various hardware resources of a computer system. These hardware resources include processer, memory, and disk space, printer etc. The output result was display in monitor. In addition to communicating with the hardware the operating system provides on error handling procedure and display an error notification, if a device is not functioning properly and the operating system cannot communicate properly with the device.

Providing an Interface

The Operatin Syste Organizes Application so that users can easily access, use and store them.



Operating System provides a stable and consistent way for applications to deal with the hardware. The user is not required to have any knowledge of hardware. If the program is not working properly, the operating system takes control again, stopping the application and displaying the appropriate error message.

Computer system components are divided into five parts

- Computer hardware
- Operating system
- Utilities
- Application programs
- End user

The operating system controls and coordinates the use of hardware and various application programs for various users. It is a program that directly interacts with the hardware. The operating system is the first encoded with the Computer and it remains on the memory all time thereafter. The purpose of an operating system is to be provided an environment in which a user can execute its programs without any difficulty. Its primary goals are to make the computer system convenience for the user.

Its secondary goals are to use the computer hardware in efficient manner.

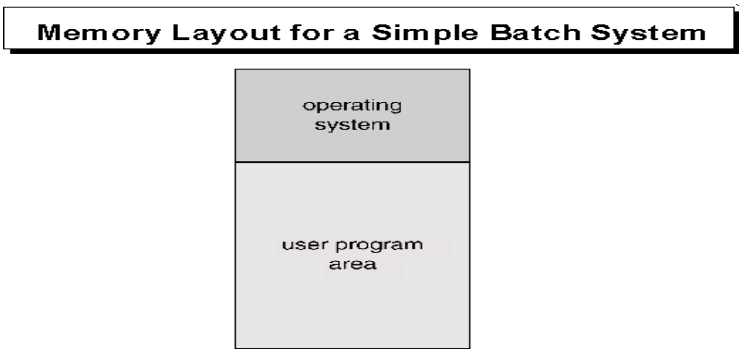
View of operating system

- **User view:** The user view of the computer varies by the interface being used. The examples are -windows XP, vista, windows 7 etc. Mostcomputer user sit in the in front of personal computer (pc) in this case the operating system is designed mostly for easy use with some attention paid to resource utilization. Some user sit at a terminal connected to a mainframe/minicomputer. In this case other users are accessing the same computer through the other terminals. There user are share resources and may exchange the information. The operating system in this case is designed to maximize resources utilization to assume that all available CPU time, memory and I/O are used efficiently and no individual user takes more than his/her fair and share. The other users sit at workstations connected to network of other workstations and servers. These users have dedicated resources but they share resources such as networking and servers like file, compute and print server. Here the operating system is designed to compromise between individual usability and resource utilization.
- System view: From the computer point of view the operating system is the program which is most intermediate with the hardware. An operating system has resources as hardware and software which may be required to solve a problem like CPU time, memory space, file storage space and I/O devices and so on. That's why the operating system acts as a manager of these resources. Another view of the operating system is it is a control program. A control program manages the execution of user programs to present the errors in proper use of the computer. It is especially concerned with the users operation and controlling of I/O devices.

Types of Operating System

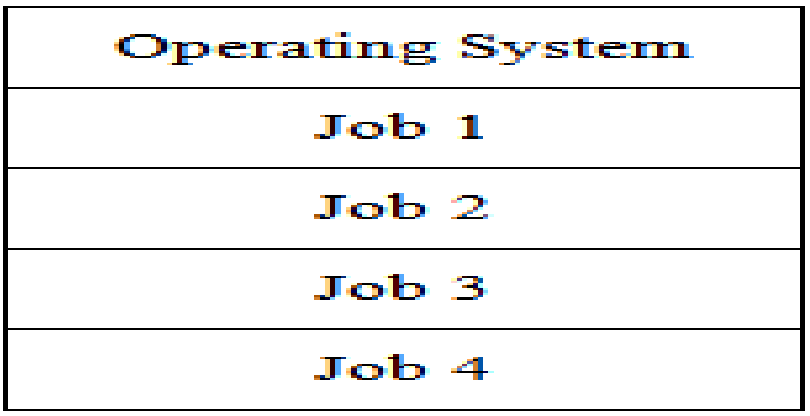
Batch /Early System: Early computers were physically large machine. The common input devices were card readers, tape drivers. The common output devices were line printers, tape drivers and card punches. In these systems

the user did not interact directly with the computer system. Instead the user preparing a job which consists of programming data and some controlling information and then submitted it to the computer operator after some time the output is appeared. The output in these early computer was fairly simple is main task was to transfer control automatically from one job to next. The operating system always resides in the memory. To speed up processing operators batched the jobs with similar needs and ran then together as a group. The disadvantages of batch system are that in this execution environment the CPU is often idle because the speed up of I/O devices is much slower than the CPU.



Multiprogramming System:

Multiprogramming concept increases CPU utilization by organization jobs so that the CPU always has one job to execute is the idea behind multiprogramming concept. The operating system keeps several jobs in memory simultaneously as shown in figure. This set of job is a subset of the jobs that kept a job pool. The operating system picks and starts to execute one of the jobs in the memory.In this environment the operating system simply switches and executes another job, when a job needs to wait the CPU simply switches to another job and so on. The multiprogramming operating system is sophisticated because the operating system makes decisions for the user. This is known as scheduling. If several jobs are ready to run at the same time the system choose one among them. This is known as CPU scheduling. The disadvantages of the multiprogramming system is



(Multiprogramming System)

- It does not provide user interaction with the computer system duringthe program execution.
- The introduction of disk technology solved these problems rather than reading the cards from card reader into disk. This form of processingis known as spooling.

SPOOL stands for simultaneous peripheral operations online. It uses the disk as a huge buffer for reading from input devices and for storing output data until the output devices accept them. It is also use for processing data at remote sides. The remote processing is done and its own speed with no CPU intervention. Spooling overlaps the input, output one job with computation of other jobs. Spooling has a beneficial effect on the performance of the systems by keeping both CPU and I/O devices working at much higher time.

Time Sharing System: The time sharing system is also known as multi user systems. The CPU executes multiple jobs by switching among them but the switches occurs so frequently that the user can interact with each program while it is running. An interactive computer system provides direct communicationbetween a user and system. The user gives instruction to the operating systemsor to a program directly using keyboard or mouse and wait for immediate results. So the response time will be short. The time sharing system allows many users to share the computer simultaneously. Since each action in this system is short, only a little CPU time is needed for each user. The system switches rapidly from one user to the next so each user feels as if the entire computer system is dedicated to his use, even though it is being shared by many users. The disadvantages of time sharing system are:

- It is more complex than multiprogramming operating system
- The system must have memory management & protection,

sinceseveral jobs are kept in memory at the same time.

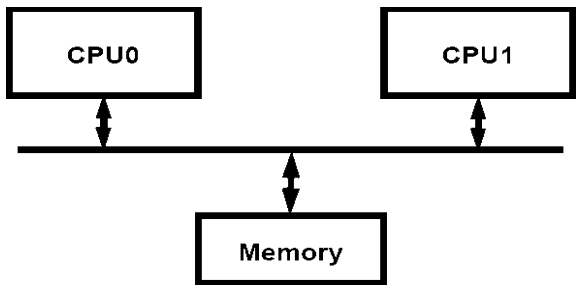
- Time sharing system must also provide a file system, so disk management isrequired.
- It provides mechanism for concurrent execution which requirescomplex CPU scheduling schemes.

Parallel Systems/ Tightly coupled Systems: These Systems have more than one processor in close communications which share the computer bus, clock, memory & peripheral devices. Ex: UNIX, LINUX. Multiprocessor Systems have 3 main advantages.

- **Increased throughput:** No. of processes computed per unit time. By increasing the no. Of processors move work can be done in less time. The speed up ratio with N processors is not N, but it is less than N. Because a certain amount of overhead is incurred in keeping all the parts working correctly.
- **Increased Reliability:** If functions can be properly distributed among several processors, then the failure of one processor will not halt the system, but slow it down. This ability to continue to operate in spite of failure makes the system fault tolerant.
- **Economic scale:** Multiprocessor systems can save money as they can share peripherals, storage & power supplies.

The various types of multiprocessing systems are:

- **Symmetric Multiprocessing (SMP):** Each processor runs an identical copy of the operating system & these copies communicate with one another as required. Ex: Encore's version of UNIX for multi max computer. Virtually, all modern operating system including Windows NT, Solaris, Digital UNIX, OS/2 & LINUX etc.



- **Asymmetric Multiprocessing (Master - Slave Processors):** Each processor is designed for a specific task. A master processor controls the system & schedules & allocates the work to the slave processors. Ex- Sun's operating system SUNOS version 4 provides asymmetric multiprocessing.

Distributed System/Loosely Coupled Systems: In contrast to tightly coupled systems, the processors do not share memory or a clock. Instead, each processor has its own local memory. The processors communicate with each other by various communication lines such as high speed buses or telephone lines. Distributed systems depend on networking for their functionalities. By being able to communicate distributed systems are able to share computational tasks and provide a rich set of features to the users. Networks vary by the protocols used, the distances between the nodes and transport media. TCP/IP is the most common network protocol. The processor is a distributed system varies in size and function. It may microprocessors, work stations, minicomputer, and large general purpose computer. Network types are based on the distance between the nodes such as LAN (within a room, floor or building) and WAN (between buildings, cities or countries). The advantages of distributed system are resource sharing, computation speed up, reliability, communication.

Real time Systems: Real time system is used when there are rigid time requirements on the operation of a processor or flow of data. Sensors bring data to the computers. The computer analyzes data and adjusts controls to modify the sensors inputs. System that controls scientific experiments, medical Imaging systems and some display systems are real time systems. It is a special purpose OS in which there are rigid time requirements on the operation of a processor. A real time OS has well defined fixed time constraints. Processing must be done within the time constraint or the system will fail. A real time system is said to function correctly only if it returns the correct result within the time constraint. These systems are characterized by having time as a key parameter.

The disadvantages of real time system are:

- A real time system is considered to function correctly only if it returns the correct result within the time constraints.

- Secondary storage is limited or missing instead data is usually stored in short term memory or ROM.
- Advanced OS features are absent.

Real time system is of two types such as:

- **Hard real time systems:** It guarantees that the critical task has been completed on time. The sudden task is takes place at a sudden instant of time.
- **Soft real time systems:** It is a less restrictive type of real time system where a critical task gets priority over other tasks and retains that priority until it computes. These have more limited utility than hard real time systems. Missing an occasion deadline is acceptable e.g. QNX, VX works. Digital audio or multimedia is included in this category.

Operating System Services

An operating system provides an environment for the execution of the program. It provides some services to the programs. The various services provided by an operating system are as follows:

- **Program Execution:** The system must be able to load a program into memory and to run that program. The program must be able to

terminate this execution either normally or abnormally.

- **I/O Operation:** A running program may require I/O. This I/O may Involve a file or I/O device for specific device. Some special function can be desired. Therefore the operating system must provide a means to do I/O.
- **File System Manipulation:** The programs need to create and delete files by name and read and write files. Therefore the operating system must maintain each and every file correctly.
- **Communication:** The communication is implemented via shared memory or by the technique of message passing in which packets of information are moved between the processes by the operating system.
- **Error detection:** The operating system should take the appropriate actions for the occurrences of any type like arithmetic overflow, access to the illegal memory location and too large user CPU time.
- **Resource Allocation:** When multiple users are logged on to the system the resources must be allocated to each of them. For current distribution of the resource among the various processes the operating system uses the CPU scheduling run times which determine which process will be allocated with the resource.
- **Accounting:** The operating system keep track of which users use how many and which kind of computer resources.
- **Protection:** The operating system is responsible for both hardware as well as software protection. The operating system protects the information stored in a multiuser computer system.

User and Operating System Interface

1. **CLI Command Line Interface (CLI)**

   CLI or Command Line Interface allows direct command entry Sometimes implemented in kernel, sometimes by systems program Sometimes multiple flavors implemented – shells Primarily fetches a command from user and executes it .Sometimes commands built-in, sometimes just names of programs If the latter, adding new features doesn't require shell modification

2. **GUI User-friendly interface**

Usually mouse, keyboard, and monitor 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 a folder)

- Invented at Xerox PARC Many systems now include both CLI and GUI interfaces Microsoft Windows is GUI with CLI "command" shell Apple Mac OS X as "Aqua" GUI interface with UNIX kernel underneath and shells available Solaris is CLI with optional GUI interfaces (Java Desktop, KDE)

Basic Functions of Operation System:

The various functions of operating system are as follows:

1. Process Management:

- A program does nothing unless their instructions are executed by a CPU. A process is a program in execution. A time shared user program such as a complier is a process. A word processing program being run by an individual user on a pc is a process.
- A system task such as sending output to a printer is also a process. A process needs certain resources including CPU time, memory files & I/O devices to accomplish its task.
- These resources are either given to the process when it is created or allocated to it while it is running. The OS is responsible for the following activities of process management.
- Creating & deleting both user & system processes.
- Suspending & resuming a processes.
- Providing mechanism for process synchronization.
- Providing mechanism for process communication.
- Providing mechanism for dead lock handling.

2. Main Memory Management:

The main memory is central to the operation of a modern computer system. Main memory is a large array of words or bytes ranging in size from hundreds of thousands to billions. Main memory stores the

quickly accessible data shared by the CPU & I/O device. The central processor reads instruction from main memory during instruction fetch cycle & it both reads & writes data from main memory during the data fetch cycle. The main memory is generally the only large storage device that the CPU is able to address & access directly. For example, for CPU to process any data, the data must first be transferred to main memory by CPU generated I/O calls. Instruction must be in memory for the CPU to execute them. The OS is responsible for the following activities in connection with memory management.

- Keeping track of which parts of memory are currently being used & by whom.
- Deciding which processes are to be loaded into memory when memory space becomes available.
- Allocating & deallocating memory space as needed.

3. File Management:

File management is one of the most important components of an OS computer can store information on several different types of physical media magnetic tape, magnetic disk & optical disk are the most common media. Each medium is controlled by a device such as disk drive or tape drive those has unique characteristics. These characteristics include access speed, capacity, data transfer rate & access method (sequential or random).For convenient use of computer system the OS provides a uniform logical view of information storage. The OS abstracts from the physical properties of its storage devices to define a logical storage unit the file. A file is collection of related information defined by its creator. The OS is responsible for the following activities of file management.

- Creating & deleting files.
- Creating & deleting directories.
- Supporting primitives for manipulating files &directories.
- Mapping files into secondary storage.
- Backing up files on non-volatile media.

4. I/O System Management:

One of the purposes of an OS is to hide the peculiarities of specific

hardware devices from the user. For example, in UNIX the peculiarities of I/O devices are hidden from the bulk of the OS itself by the I/O subsystem. The I/O subsystem consists of: Memory management component that includes buffering, catching & spooling.

- A general device- driver interfaces drivers for specific hardware devices. Only the device driver knows the peculiarities of the specific device to which it is assigned.

5. **Secondary Storage Management:**

   The main purpose of computer system is to execute programs. These programs along with the data they require must be in main memory during execution. As the main memory is too small to accommodate all data & programs & because the data that it holds are lost when power is lost. The computer system must provide secondary storage to back-up main memory. Most modern computer systems are disks as the storage medium to store data & program. The operating system is responsible for the following activities of disk management.

   - Free space management.
   - Storage allocation.
   - Disk Scheduling

   Because secondary storage is used frequently it must be used efficiently.

Networking:

A distributed system is a collection of processors that don't share memory peripheral devices or a clock. Each processor has its own local memory & clock and the processor communicate with one another through various communication lines such as high speed buses or networks. The processors in the system are connected through communication networks which are configured in a number of different ways. The communication network design must consider message routing & connection strategies are the problems of connection &security.

Protection or security:

If a computer system has multi users & allow the concurrent execution of

multiple processes then the various processes must be protected from one another's activities. For that purpose, mechanisms ensure that files, memory segments, CPU & other resources can be operated on by only those processes that have gained proper authorization from the OS.

Command interpretation:

One of the most important functions of the OS is connected interpretation where it acts as the interface between the user & the OS.

System Call:

System calls provide the interface between a process & the OS. These are usually available in the form of assembly language instruction. Some systems allow system calls to be made directly from a high level language program like C, BCPL and PERL etc. systems calls occur in different ways depending on the computer in use. System calls can be roughly grouped into five major categories.

1. **Process Control:**
   - **End, abort:** A running program needs to be able to has its execution either normally (end) or abnormally (abort).
   - **Load, execute:** A process or job executing one program may want to load and executes another program.
   - **Create Process, terminate process:** There is a system call specifying for the purpose of creating a new process or job (create process or submit job). We may want to terminate a job or process that we created (terminates process, if we find that it is incorrect or no longer needed).
   - **Get process attributes, set process attributes:** If we create a new job or process we should able to control its execution. This control requires the ability to determine & reset the attributes of a job or processes (get process attributes, set process attributes).
   - **Wait time:** After creating new jobs or processes, we may need to wait for them to finish their execution (wait time).
   - **Wait event, signal event:** We may wait for a specific event to occur (wait event). The jobs or processes then signal when that event has

occurred (signal event).

2. **File Manipulation:**
   - **Create file, delete file:** We first need to be able to create & delete files. Both the system calls require the name of the file & some of its attributes.
   - **Open file, close file:** Once the file is created, we need to open it & use it. We close the file when we are no longer using it.
   - **Read, write, reposition file:** After opening, we may also read, write or reposition the file (rewind or skip to the end of the file).
   - **Get file attributes, set file attributes:** For either files or directories, we need to be able to determine the values of various attributes & reset them if necessary. Two system calls get file attribute & set file attributes are required for their purpose.
3. **Device Management:**
   - **Request device, release device:** If there are multiple users of the system, we first request the device. After we finished with the device, we must release it.
   - **Read, write, reposition:** Once the device has been requested & allocated to us, we can read, write & reposition the device.

4. **Information maintenance:**
   - **Get time or date, set time or date:** Most systems have a system call to return the current date & time or set the current date & time.
   - **Get system data, set system data:** Other system calls may return information about the system like number of current users, version number of OS, amount of free memory etc.
   - **Get process attributes, set process attributes:** The OS keeps information about all its processes & there are system calls to access this information.
5. **Communication:** There are two modes of communication such as:
   - **Message passing model:** Information is exchanged through an inter process communication facility provided by operating system. Each computer in a network has a name by which it is known. Similarly, each

process has a process name which is translated to an equivalent identifier by which the OS can refer to it. The get host id and get processed systems calls to do this translation. These identifiers are then passed to the general purpose open & close calls provided by the file system or to specific open connection system call. The recipient process must give its permission for communication to take place with an accept connection call. The source of the communication known as client & receiver known as server exchange messages by read message & write message system calls. The close connection call terminates the connection.

- **Shared memory model:** processes use map memory system calls to access regions of memory owned by other processes. They exchange information by reading & writing data in the shared areas. The processes ensure that they are not writing to the same location simultaneously.

SYSTEM PROGRAMS:

System programs provide a convenient environment for program development & execution. They are divided into the following categories.

- **File manipulation:** These programs create, delete, copy, rename, print & manipulate files and directories.
- **Status information:** Some programs ask the system for date, time & amount of available memory or disk space, no. of users or similar status information.
- **File modification:** Several text editors are available to create and modify the contents of file stored on disk.

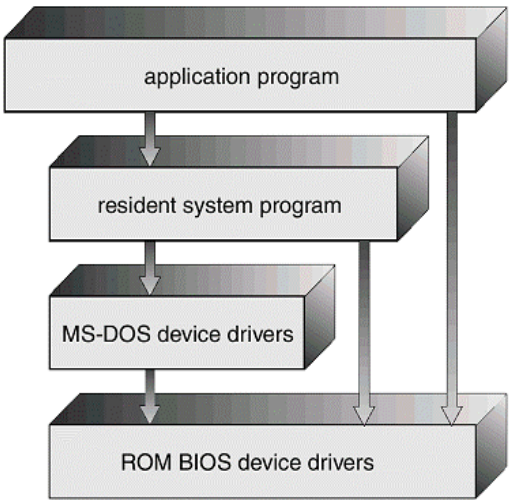
- **Programming language support:** compliers, assemblers & interpreters are provided to the user with the OS.
- **Programming loading and execution:** Once a program is assembled or compiled, it must be loaded into memory to be executed.
- **Communications:** These programs provide the mechanism for creating virtual connections among processes user's and different computer systems.

- **Application programs:** Most OS are supplied with programs that are useful to solve common problems or perform common operations. Ex: web browsers, word processors & text formatters etc.

Operating System Structure:

1. Simple structure:

There are several commercial system that don't have a well- defined structure such operating systems begins as small, simple & limited systems and then grow beyond their original scope.

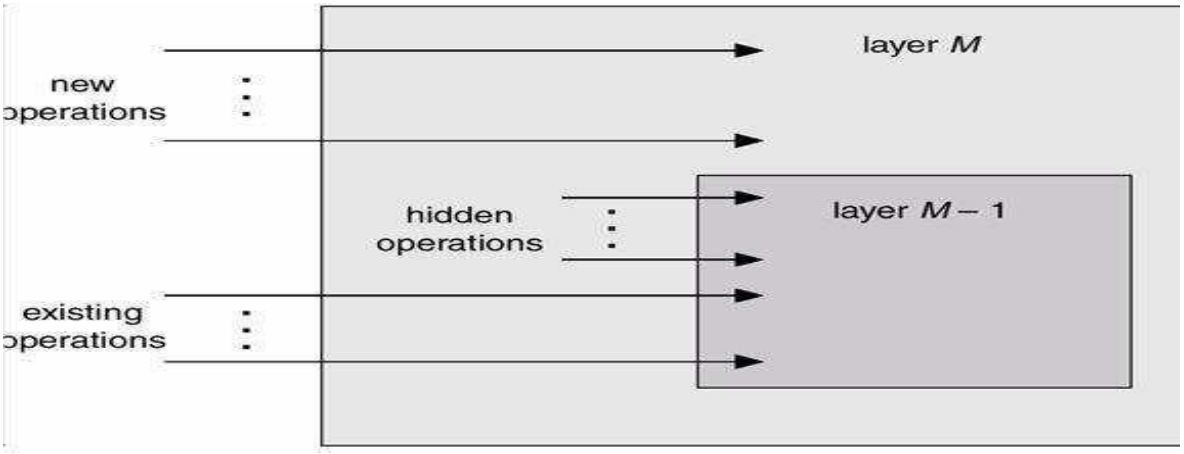


MS DOS Structure

MS-DOS is an example of such system. It was not divided into modules carefully. Another example of limited structuring is theUNIX operating system.

2. Layered approach:

In the layered approach, the OS is broken into a number of layers (levels) each built on top of lower layers. The bottom layer (layer 0) is the hardware. Top most layers (layer N) are the user interface. The main advantage of the layered approach is modularity. The layers are selected such that each users functions (or operations) & services of only lower layer.



Layerd Structure

- This approach simplifies debugging & system verification, i.e. the first layer can be debugged without concerning the rest of the system. Once the first layer is debugged, its correct functioning is assumed while the 2nd layer is debugged & soon.
- If an error is found during the debugging of a particular layer, the error must be on that layer because the layers below it are already debugged.Thus the design & implementation of the system are simplified when the system is broken down into layers.
- Each layer is implemented using only operations provided by lower layers. A layer doesn't need to know how these operations are implemented; it only needs to know what these operations do.
- The layer approach was first used in the WINDOWS NT operating system. It was defined in six layers.

Layers	Functions
5	User Program
4	I/O Management
3	Operator Process Communication
2	Memory Management
1	CPU Scheduling
0	Hardware

The main disadvantage of the layered approach is:

- The main difficulty with this approach involves the careful definition of the layers, because a layer can use only those layers below it. For example, the device driver for the disk space used by virtual memory algorithm must be at a level lower than that of the memory management routines, because memory management requires the ability to use the disk space.
- It is less efficient than a non layered system (Each layer adds overhead to the system call & the net result is a system call that take longer time than on a non layered system).

3. Micro Kernel

A microkernel is a type of operating system kernel that is designed to provide only the most basic services required for an operating system to function, such as memory management and process scheduling. Other services, such as device drivers and file systems, are implemented as user-level processes that communicate with the microkernel via message passing. This design allows the operating system to be more modular and flexible than traditional monolithic kernels, which implement all operating system services in kernel space.

The main advantage of microkernel architecture is that it provides a more secure and stable operating system. Since only the most essential services run in kernel space, the attack surface of the operating system is reduced, making it more difficult for an attacker to exploit vulnerabilities. Additionally, if a user-level process crashes, it will not affect the stability of the entire system, since the microkernel is responsible only for managing processes and memory.

Another advantage of a microkernel architecture is that it makes the operating system more modular and flexible. Since services are implemented as user-level processes, it is easier to add, remove, or replace services without affecting other parts of the system. This makes it easier to customize the operating system to meet specific requirements.

However, there are also some disadvantages to a microkernel architecture. One major disadvantage is that message passing between user-level processes can be slower than direct system calls in a monolithic kernel. This can affect the performance of the operating system, especially in high-performance applications. Additionally, the modular design of a microkernel can lead to increased complexity, which can make it more difficult to develop and maintain the operating system.

Overall, a microkernel architecture can provide a more secure and flexible operating system, but it may also come with some performance and complexity trade-offs. The choice between a microkernel and monolithic kernel architecture depends on the specific needs and requirements of the operating system being developed.

4. Exokernel

Significant advancements have been made in the field of operating systems (OS) over the years. The idea of an Exokernel is one such innovation that has evolved in recent decades. This ground-breaking method of OS design challenges conventional monolithic and microkernel architectures by giving system designers and users more flexibility, performance, and control. The following are Exokernel Architecture core principles:

1. Resource Exposure

In an Exokernel, programs have direct access to hardware resources such the CPU, memory, and I/O units. Applications may now handle these resources with greater precision because they have fine-grained control over them.

2. Protection and Isolation

Exokernels provide robust isolation and protection features despite exposing resources. Applications are protected from illegal access and intervention by running in separate address spaces.

3. Minimalism

Exokernels take a straightforward stance. They leave high-level services and resource management to application-level libraries and only offer the absolute necessities in abstractions. This minimalism encourages adaptability and effectiveness.

4. Flexibility

Exokernels adopt a direct posture. They only provide the bare minimum in abstractions, leaving high-level services and resource management to application-level libraries. This minimalism promotes efficacy and adaptability.

Advantages of Exokernels

- Performance: Because applications have direct control over resources, exokernels frequently run better than conventional kernels.

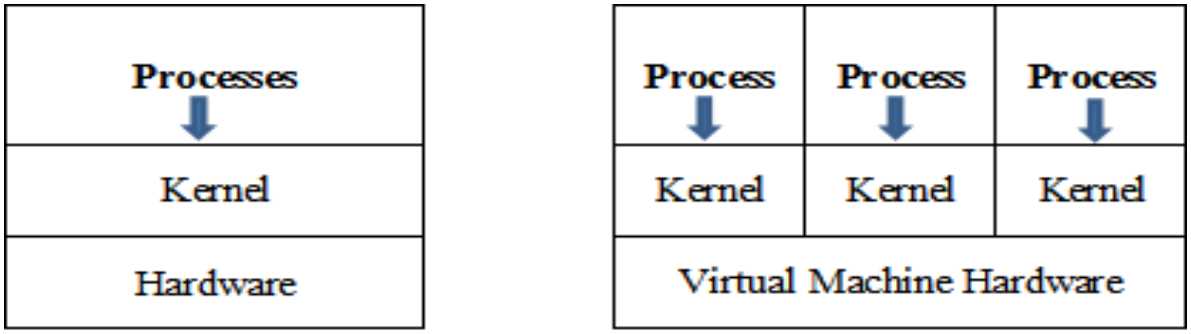
- Flexibility: Resource management can be tailored by developers to be more efficient for particular workloads.
- Resource Use: Exokernels allow for effective resource use, which lowers overhead.

Challenges and Considerations

- Exokernel development can be complicated since it requires precise resource control.
- Security: Direct resource exposure to applications raises questions about security. Strong isolation techniques are necessary.
- Compatibility: Moving existing applications from conventional OS architectures to Exokernels might necessitate their modification or adaptation.

5. Virtual Machine:

By using CPU scheduling & virtual memory techniques an operating system can create the illusion of multiple processes, each executing on its own



(Non virtual Machine) (Virtual Machine)

processors &

Own virtual memory. Each processor is provided a virtual copy of underlying computer. The resources of the computer are shared to create the virtual machines. CPU scheduling can be used to create the appearance that users have their own processor.

Implementation: Although the virtual machine concept is useful, it is difficult to implement since much effort is required to provide an exact duplicate of the underlying machine. The CPU is being multiprogrammed among several virtual machines, which slows down the virtual machines in various ways.

Difficulty: A major difficulty with this approach is regarding the disk system. The solution is to provide virtual disks, which are identical in all respects except size. These are known as mini disks in IBM's VM OS. The sum of sizes of all mini disks should be less than the actual amount of physical disk space available.

Benefits of using virtual machines

Personalization:

There is great flexibility and personalization with virtual machines. No two environments have to be exactly the same; finance can have a different virtual environment than that of marketing. Likewise, depending on the user and their title, a virtual machine can be configured to maintain user defined settings for each session or can be configured to reset those settings at the beginning of each session.

Virtualization:

Virtualization involves using software to creating a "virtual", non-physical form or copy of something. This includes the reproduction of hardware in a digitally accessible manner. VMs can transform a workplace into a more mobile, flexible workplace where traditional hardware is replaced with virtual environments, infrastructure, and storage. Employees can access these virtual environments from anywhere at any time, allowing businesses to forgo the upfront costs of expensive hardware.

Backup by VM at a time:

With virtual machines, businesses can backup or store data one VM at a time. Instead of backing up individual files, the entire virtual machine can be backed up altogether. Remember, the VM is not locally hosted on the computer itself; instead all of the data and files used within the VM are accessed over a network like the internet. This model allows businesses to replicate or backup entire VMs at a time.

Easy Recover/Failover:

Since virtual machines can be backed up one VM at a time, the recovery/failover process is made much easier. Businesses not only can seamlessly failover to VMs, but can restore those VM's quickly and efficiently, making file and application access simple. This can help reduce downtime in the face of error or disaster.

Run multiple Operating Systems:

VM's allow users to work in multiple environments. For example, Microsoft Office users can run a Mac operating system or vice versa thanks to virtual machines. It's not just about running multiple operating systems, either. Users can utilize certain applications within the VM without having to provision it on their local computers. This saves IT professional's time, allowing them to work on patching, updating and testing the application, or even focus on other core competency projects.

OS and Application Updates:

Software updates are performed by the service provider that administers the VM. If there are new updates to the operating system, the service provider also manages those updates so that the end user's environment is always up to date. Applications that your employees use within the VM are also updated by the service provider, not your own IT team. This is what is referred to as desktop as a Service and can save your IT team hours of monotonous updates.