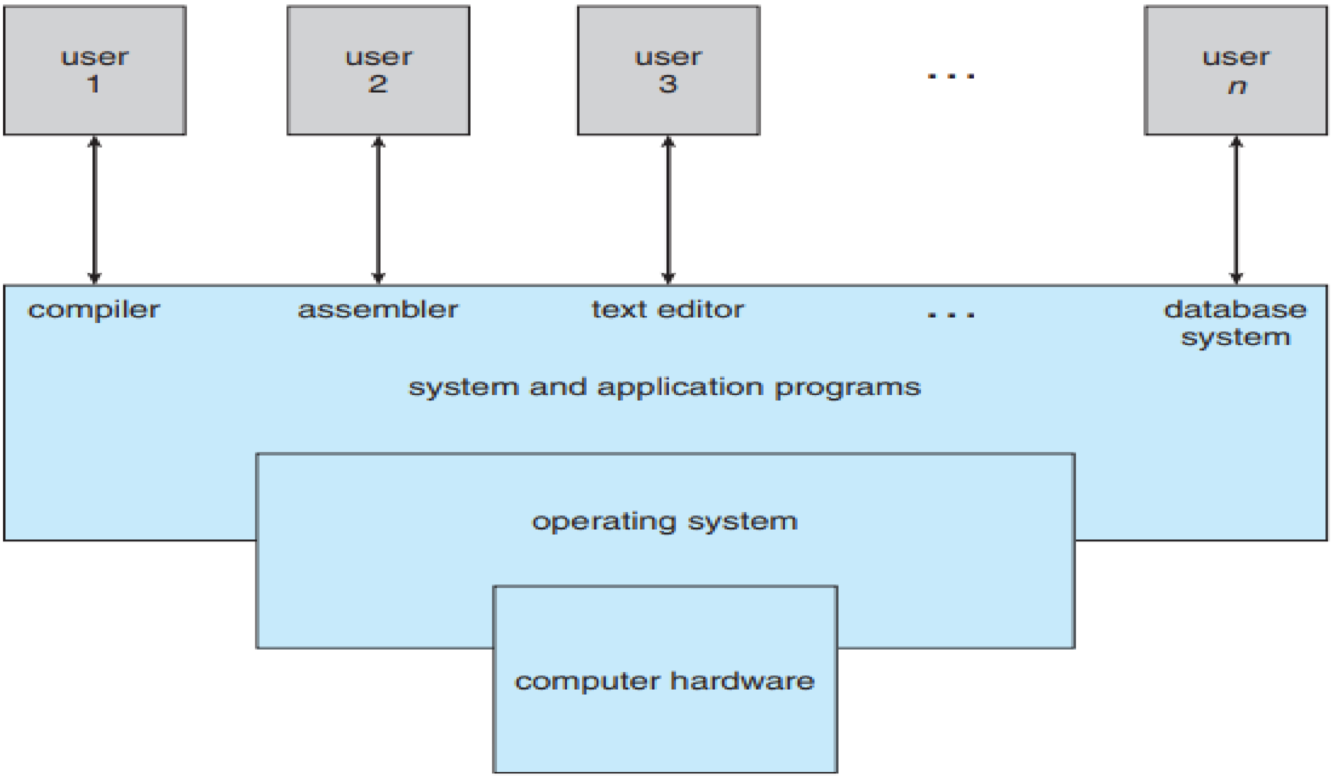
UNIT I

Introduction: Basics of Operating Systems: Definition, types of Operating Systems, OS Service, System Calls, OS structure: Layered, Monolithic, Microkernel Operating Systems – Concept of Virtual Machine.

Introduction to Operating System:

- An operating system is software that manages the basic resources of a computer i.e Processor, Memory, Files, Peripheral devices, disk storage units etc…
- It provides an environment in which user can execute programs in a convenient and efficient manner.
- Purpose of OS is to transfer control automatically from one job to next job.
- OS performs the job of allocating resources for various tasks such as program execution, file creation, command execution etc. and reclaims the allocated resources after their use.
- OS allocates Main Memory (MM) for the running program to the CPU, Schedules them and manage the data transfer to and from peripheral (I/O) devices to MM and Secondary storage (disks) units.



Abstract view of the components of a computer system.

User Views:-

The user view of the computer depends on the interface used.

- Some users may use PC's. In this the system is designed so that only one user can utilize the resources and mostly for ease of use where the attention is mainly on performances and not on the resource utilization.
- Some users may use a terminal connected to a mainframe or minicomputers.
- Other users may access the same computer through other terminals. These users may share resources and exchange information. In this case the OS is designed to maximize resource utilization-so that all available CPU time, memory & I/O are used efficiently.
- Other users may sit at workstations, connected to the networks of other workstation and servers. In this case OS is designed to compromise between individual visibility & resource utilization.

System Views:

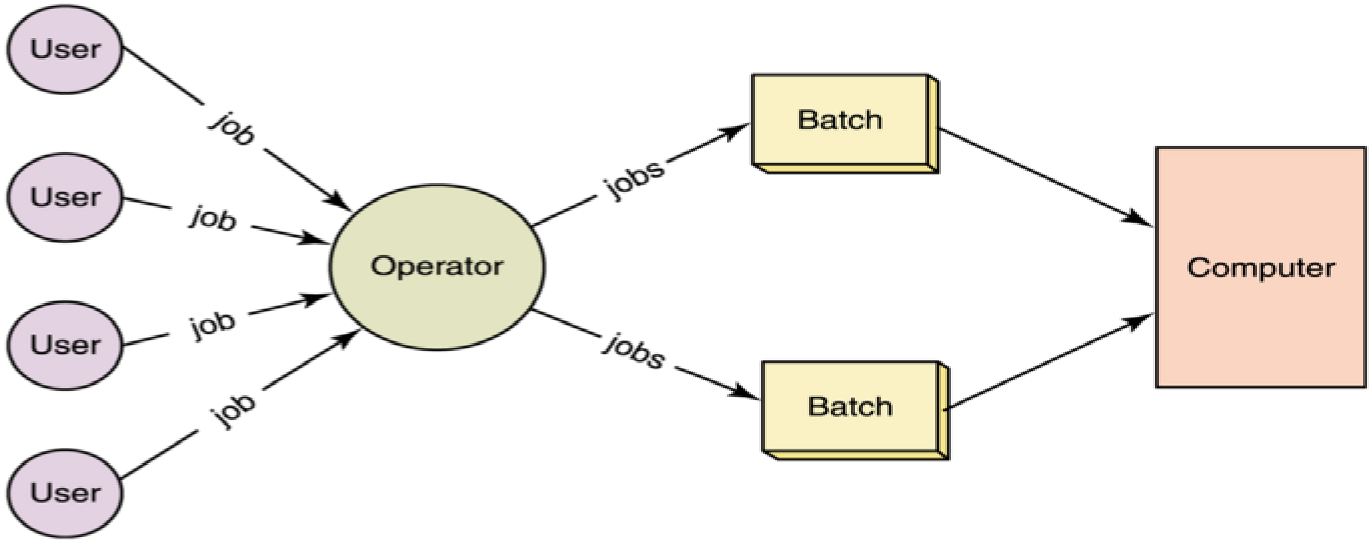
- We can view system as resource allocator i.e. a computer system has many resources that may be used to solve a problem. The OS acts as a manager of these resources. The OS must decide how to allocate these resources to programs and the users so that it can operate the Computer system efficiently and fairly.
- A different view of an OS is that it need to control various I/O devices & user programs i.e. an OS is a control program used to manage the execution of user program to prevent errors and improper use of the computer.
- Resources can be either CPU Time, memory space, file storage space, I/O devices and so on. The OS must support the following tasks

Types of Operating Systems

1. Batch Operating System.
2. Multiprogrammed Operating System
3. Time-Sharing Operating Systems
4. Parallel Systems
5. Distributed Operating System
6. Real-Time Operating System
7. Network Operating System

1. Batch Operating System:

This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having same requirement and group them into batches. It is the responsibility of operator to sort the jobs with similar needs.



Human operators would organize jobs into batches

- Early computers where physically large machines.
- The common I/P devices are card readers & tape drives.
- The common O/P devices are line printers, tape drives & card punches.
- The job was mainly in the form punched cards.
- At later time the O/P appeared and it consisted of result along with dump of memory and register content for debugging.
- The OS of these computers was very simple.
- Its major task was to transfer control from one job to the next.
- The OS was always resident in the memory.
- The processing of job was very slow. To improve the processing speed operators batched together the jobs with similar needs and processed it through the computers. This is called Batch Systems.

Drawback of First OS

The CPU may be idle for some time because the speed of the mechanical devices slower compared to the electronic devices.

In order to eliminate this drawback, a batch OS was used to perform the task of batching jobs.

Advantages:

- Simple, Sequential job Scheduling.
- Human interventions minimized.
- Increased performance & System throughput due to batching of jobs.

Disadvantages:

- Turn around time can be large from user point of view due to batching.
- Difficult to debug the program.
- A job can enter into infinite loop.
- A job could corrupt the monitor.
- Due to lack of protection scheme, one job may affect the pending jobs.

Example: Payroll System, Bank Statements etc

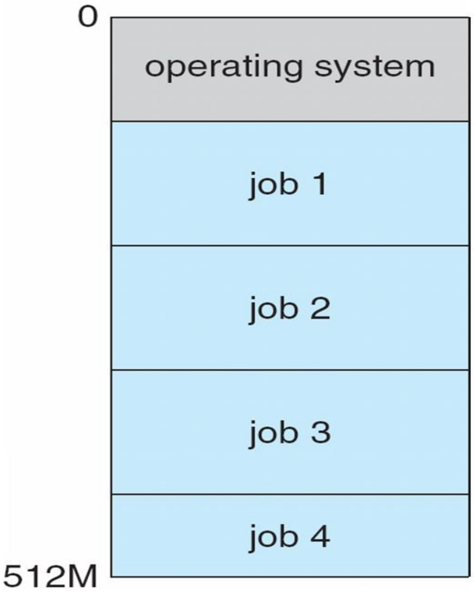
Note: Turn around time means time elapsed between the time of submission of a process or job by a user and the time of completion of that process or job.

Spooling:

- SPOOL(Simultaneous Peripheral Operation On-Line)
- Spooling is a process in which data is temporarily held to be used and executed by a device, program or the system.
- Data is sent to and stored in memory of other volatile (Temporary memory) storage until the program or computer requests it for execution.

2. Multiprogramming Operating Systems

- Multiprogramming concept increases CPU utilization by organization jobs so that the CPU always has one job to execute.
- The operating system keeps several jobs in memory simultaneously as shown in below figure.



Memory Layout for Multiprogrammed System

- This set of job is subset of the jobs kept in the job pool. The operating system picks and begins to execute one of the jobs in the memory.

- When a job needs to wait the CPU is simply switched to another job and so on.

Functions:

- The multiprogramming operating system is sophisticated because the operating system makes decisions for the user. This is known as Job scheduling.
- If several jobs are ready to run at the same time the system choose one among them. This is known as CPU scheduling.
- Other functions are Memory management, Device and File Management.

Advantages of Multiprogramming System:

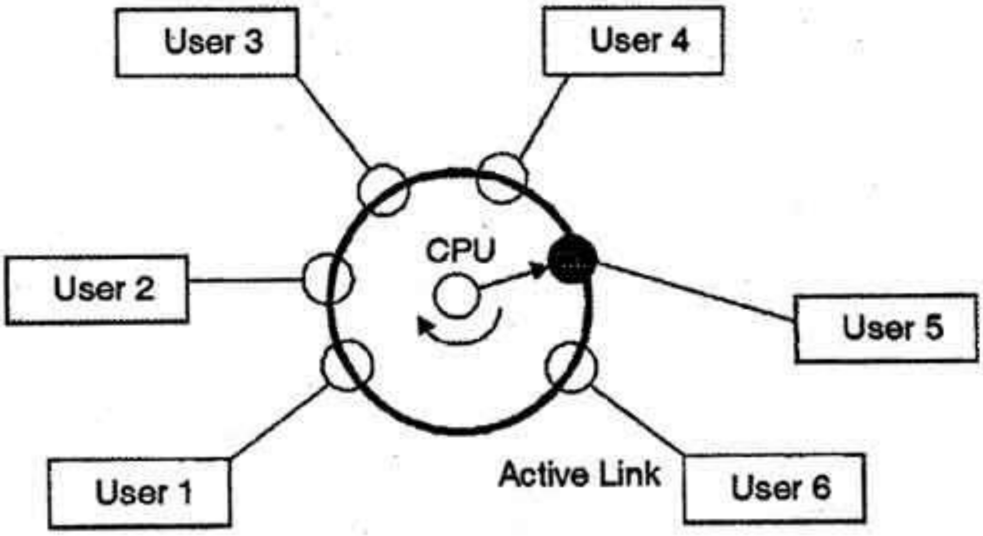
- Effective Resource Utilization (CPU, Memory, Peripheral devices).
- Elimination or Minimizing of the waste CPU idle time.
- Increased Throughput (The number of jobs executed against the number of jobs submitted for execution in a given interval of time).

The Disadvantages of Multiprogramming System:

- It does not provide user interaction with the computer system during the program execution.
- The introduction of disk technology solved these problems rather than reading the cards from card reader into disk. This form of processing is known as spooling.
- Complex and quite expensive.

3. Time-Sharing Operating Systems

- 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 communication between a user and system.
- The user gives instruction to the operating systems or 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.



- In above figure the user 5 is active but user 1, user 2, user 3, and user 4 are in waiting state whereas user 6 is in ready status.
- As soon as the time slice of user 5 is completed, the control moves on to the next ready user i.e. user 6. In this state user 2, user 3, user 4, and user 5 are in waiting state and user 1 is in ready state. The process continues in the same way and so on.

Advantages of time sharing system:

- Effective sharing and utilization of computer resources.
- Quick response time to many users.
- CPU idle time is eliminated completely.
- Suitable for on-line Data processing and user conversations.

Disadvantages of time sharing system:

- It is more complex than multiprogrammed operating system.
- The system must have memory management & protection, since several jobs are kept in memory at the same time.
- Time sharing system must also provide a file system, so disk management is required.
- It provides mechanism for concurrent execution which requires complex CPU scheduling schemes.

Example: Multics, UNIX etc.

4. Multiprocessor/Parallel/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:** Number of processes computed per unit time. By increasing the number 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:

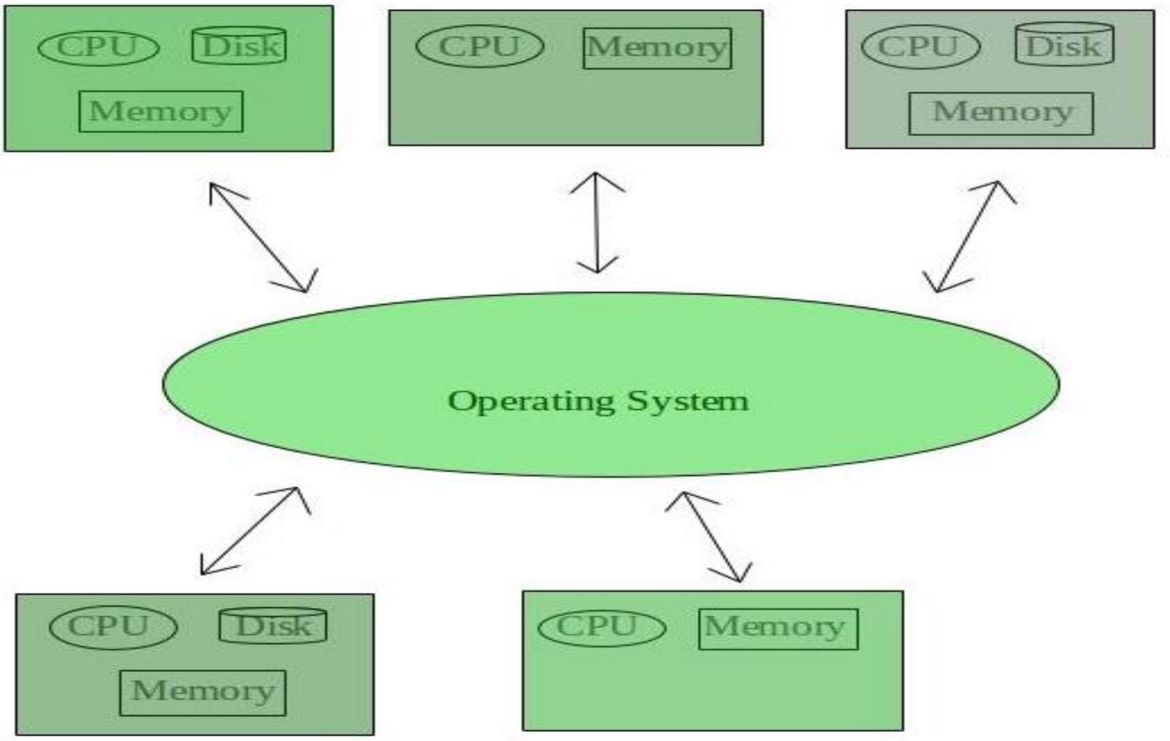
- **Symmetric Multiprocessing (SMP):** Each processor runs an identical copy of the operating system & these copies communicate with one another as required. Ex: Encore's version of UNIX for multi max computer. Virtually, all modern operating system including Windows NT, Solaris, Digital UNIX, OS/2 &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

- **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

5. Distributed/Loosely Coupled System

- 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 in a distributed system varies in size and function. It may be microprocessors, work stations, minicomputer, and large general purpose computers.

- 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.
- Computations speed up.
- Load sharing/Load Balancing.
- Reliability.
- Communication.

Disadvantages of Distributed Operating System:

- Failure of the main network will stop the entire communication.
- To establish distributed systems the language which are used are not well defined yet.
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet

Examples- LOCUS etc

6. Real-Time Operating System

- 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.

The disadvantages of real time system are:

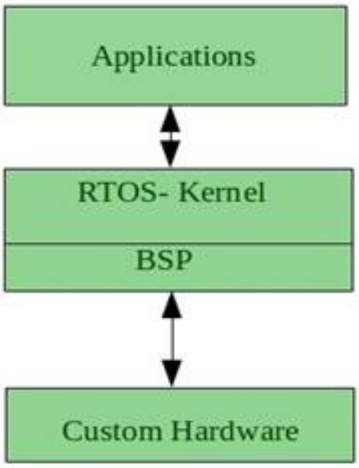
- A real time system is considered to function correctly only if it returns the correct result within the time constraints.
- Secondary storage is limited or missing instead data is usually stored in short term memory or ROM.
- Advanced OS features are absent.

Real time system is of two types such as:

1. **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.
2. **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 occasional deadline is acceptable

Example: QNX, VX works. Digital audio or multimedia is included in this category.

- 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.



Advantages of RTOS:

- **Maximum Consumption:** Maximum utilization of devices and system, thus more output from all the resources
- **Task Shifting:** Time assigned for shifting tasks in these systems are very less. For example in older systems it takes about 10 micro seconds in shifting one task to another and in latest systems it takes 3 micro seconds.
- **Focus on Application:** Focus on running applications and less importance to applications which are in queue.
- **Real time operating system in embedded system:** Since sizes of programs are small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error free.
- **Memory Allocation:** Memory allocation is best managed in these type of systems.

Disadvantages of RTOS:

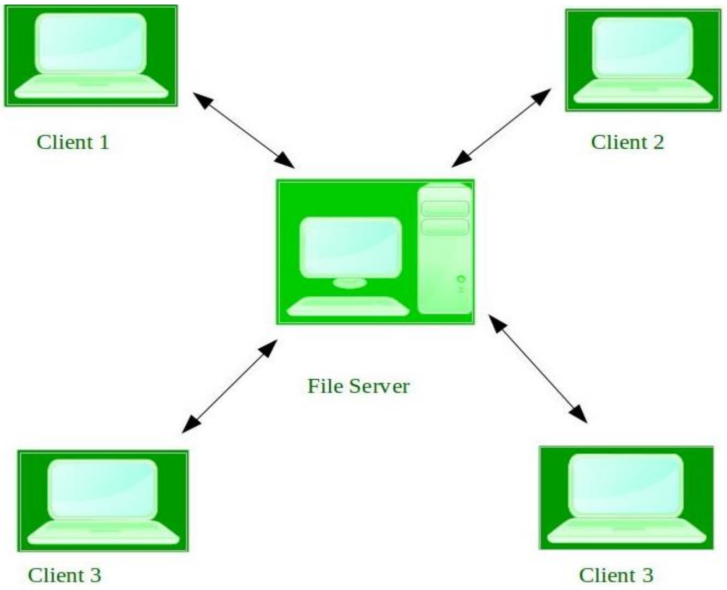
- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on few applications to avoid errors.
- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupts signals to response earliest to interrupts.
- **Thread Priority:** It is not good to set thread priority as these systems are very less pron to switching tasks.

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

7. Network Operating System

- These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions.
- This type of operating systems allows shared access of files, printers, security, applications, and other networking functions over a small private network.

- One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections etc. and that's why these computers are popularly known as **tightly coupled systems**.



Advantages of Network Operating System:

- Highly stable centralized servers
- Security concerns are handled through servers
- New technologies and hardware up-gradation are easily integrated to the system
- Server access are possible remotely from different locations and types of systems

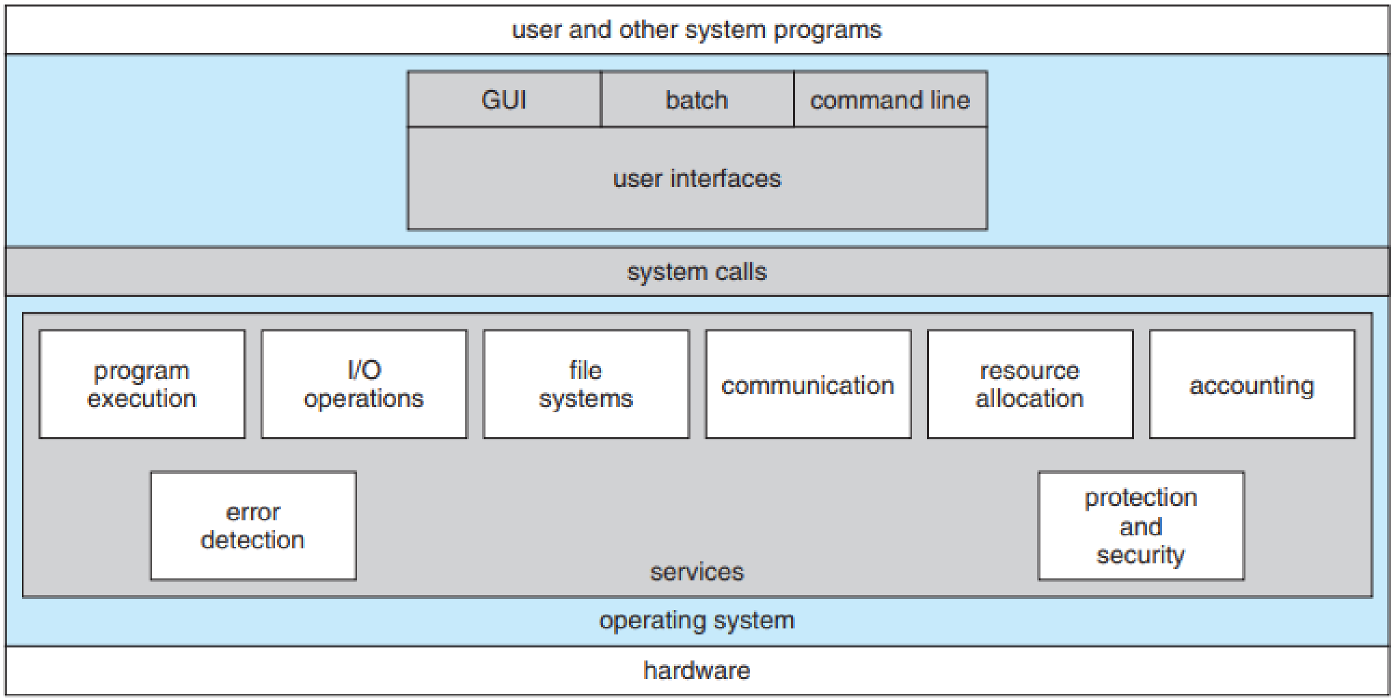
Disadvantages of Network Operating System:

- Servers are costly
- User has to depend on central location for most operations
- Maintenance and updates are required regularly

Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD etc.

❖ Operating System Services

- An operating system provides an environment for the execution of the program. It provides some services to the programs.
- The services provided by one OS may be different from other OS.
- OS makes the programming task easier.



A view of operating system services.

The various services provided by an operating system are as follows:

- **Program Execution:**
  - The system must be able to load a program into main memory partitions and to run that program.
  - The program must be able to terminate this execution normally for successful execution or abnormally for displaying errors.

- **I/O Operation:**
  - Accomplish the task of Device allocation and control I/O devices.
  - Provide for notifying device errors, device status, etc.

- **File System Manipulation:**
  - Accomplish the task of Opening a File, Closing a File.
  - The programs need to create and delete files by name
  - Allows file manipulations such as Reading a file, Writing a File, Appending a File.

- **Communication:**
  - Accomplish the task of inter-process communication either on the same computer system or between different computer systems on a computer network.
  - Provide for Message passing and shared memory access in safe mode.

- **Error detection:**
  - The operating system should take the appropriate actions for the occurrences of any type like arithmetic overflow; divide by zero error, access to the illegal memory location and too large user CPU time.
  - Accomplish the task of error detection and recovery if any. Ex: Paper jam or out of paper in a printer.
  - Keep track status of CPU, Memory, I/O devices, Storage devices, File system, networking, etc.
  - Abort execution in case of fatal errors such as RAM parity errors, power fluctuations, if any.

- **Resource Allocation:**
  - Accomplish the task of resource allocation to multiple jobs.
  - Reclaim the allocated resources after their use or as and when the job terminates.
  - 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.
  - Maintain logs of system activities for performance analysis and error recovery.

- **Protection:**
  - Accomplish the task of protecting the system resources against malicious use.
  - Provide for safe computing by employing security scheme against unauthorized access/users.
  - Authenticate legitimate users with login passwords and registrations.
  - 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.

❖ System Calls

- ➢ 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.
- ➢ System calls occur in different ways depending on the computer in use.

System calls can be roughly grouped into **5 major categories.**

1. Process Management and 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).
- **Allocate, Release Memory.**

2. File Management/File Manipulation/File Handling:

- **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/Management:

- **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 Management:

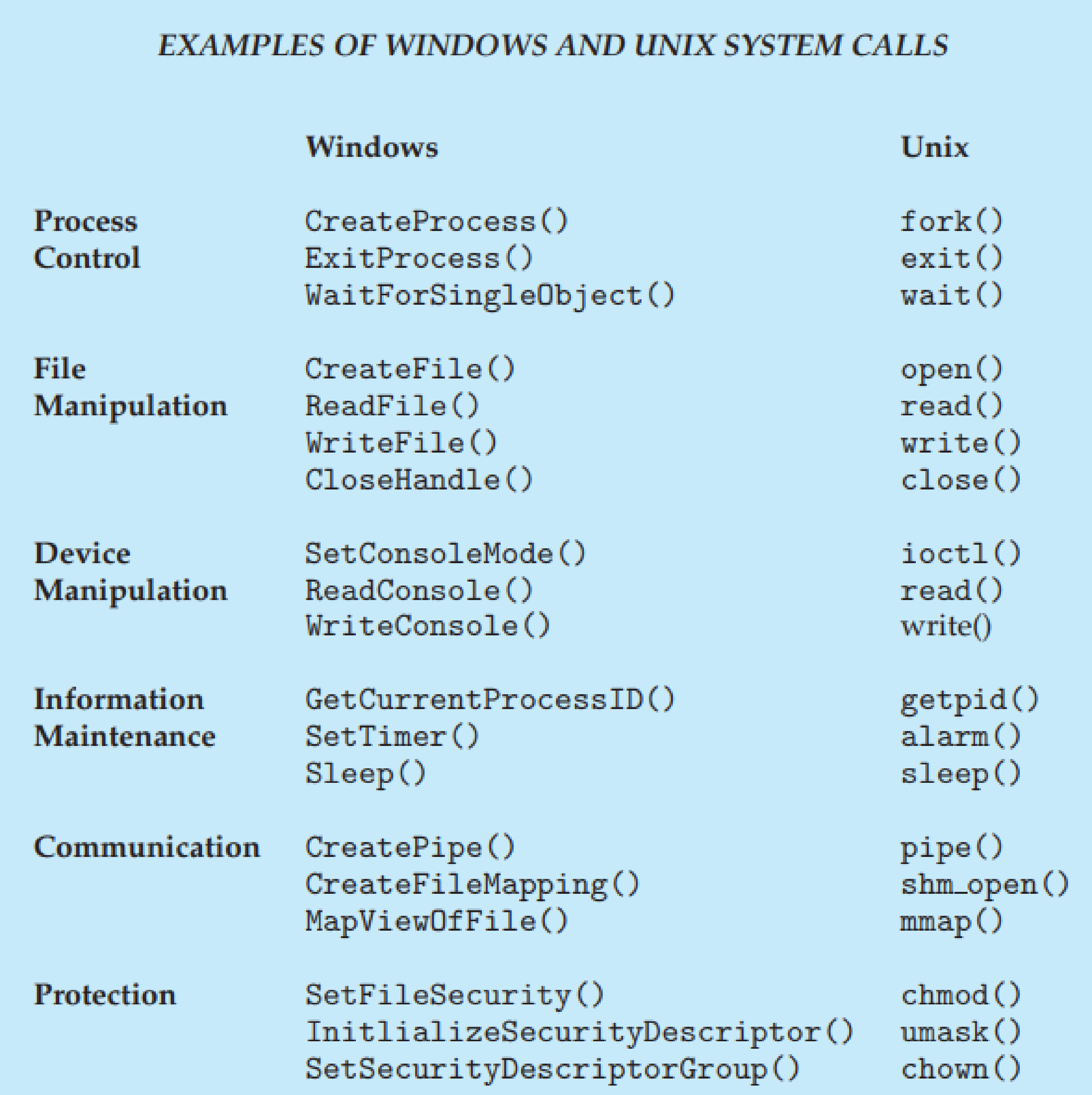
- Create, Delete Connection.
- Send, Receive Message.
- Attach, Detach remote device(Mounting/Remote Login)
- Transfer status information(byte)

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.



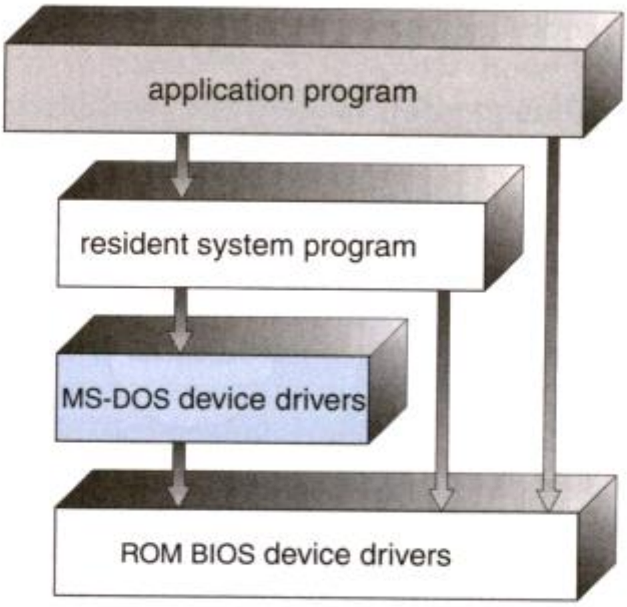
EXAMPLES OF WINDOWS AND UNIX SYSTEM CALLS

	Windows	Unix
Process Control	CreateProcess()	fork()
	ExitProcess()	exit()
	WaitForSingleObject()	wait()
File Manipulation	CreateFile()	open()
	ReadFile()	read()
	WriteFile()	write()
	CloseHandle()	close()
Device Manipulation	SetConsoleMode()	ioctl()
	ReadConsole()	read()
	WriteConsole()	write()
Information Maintenance	GetCurrentProcessID()	getpid()
	SetTimer()	alarm()
	Sleep()	sleep()
Communication	CreatePipe()	pipe()
	CreateFileMapping()	shm_open()
	MapViewOfFile()	mmap()
Protection	SetFileSecurity()	chmod()
	InitlializeSecurityDescriptor()	umask()
	SetSecurityDescriptorGroup()	chown()

❖ <u>OPERATING SYSTEM STRUCTURES</u>

The design of operating system architecture traditionally follows the *separation of concerns* principle. This principle suggests structuring the operating system into relatively independent parts that provide simple individual features, thus keeping the complexity of the design manageable.

1. Simple Structure



In MS-DOS, applications may bypass the operating system.

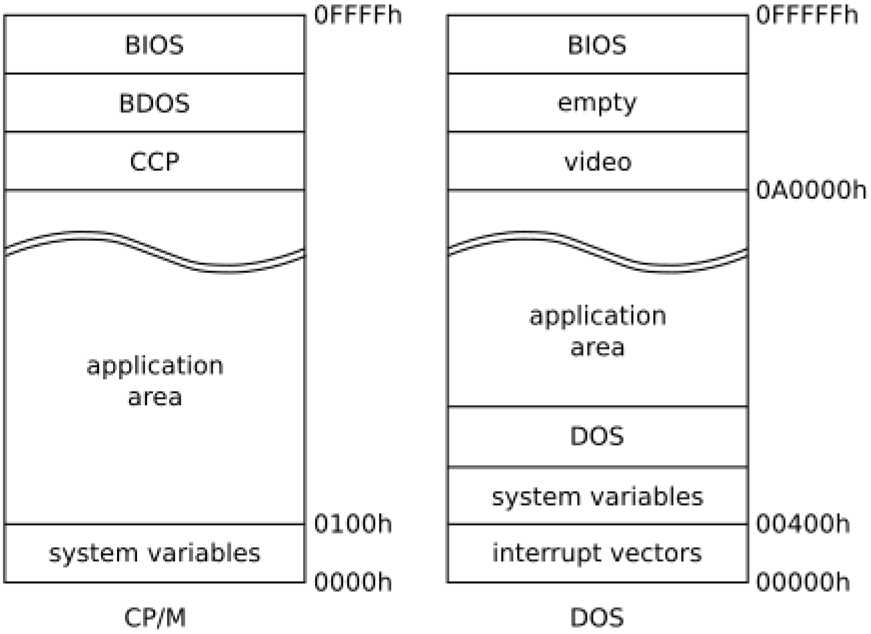
- There are several commercial system that don't have a well- defined structure such operating systems begins as small, simple & limited systems and then grow beyond their original scope.
- MS-DOS is an example of such system. It was not divided into modules carefully.
- Another example of limited structuring is the UNIX operating system.
- There was no CPU Execution Mode (user and kernel), so errors in applications could cause the whole system to crash.

2. Monolithic Approach

- A monolithic design of the operating system architecture makes no special accommodation for the special nature of the operating system. Although the design follows the separation of concerns, no attempt is made to restrict the privileges granted to the individual parts of the operating system.
- The entire operating system executes with maximum privileges.

- The communication overhead inside the monolithic operating system is the same as the communication overhead inside any other software, considered relatively low.

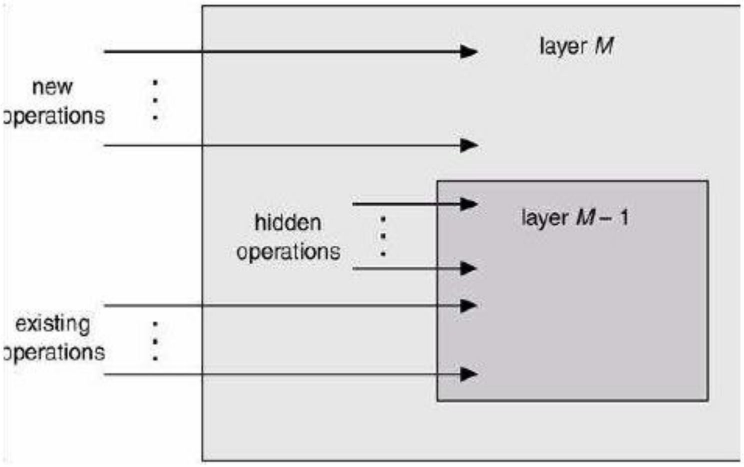
CP/M and DOS are simple examples of monolithic operating systems. Both CP/M and DOS are operating systems that share a single address space with the applications. In CP/M, the 16 bit address space starts with system variables and the application area and ends with three parts of the operating system, namely CCP (Console Command Processor), BDOS (Basic Disk Operating System) and BIOS (Basic Input/Output System). In DOS, the 20 bit address space starts with the array of interrupt vectors and the system variables, followed by the resident part of DOS and the application area and ending with a memory block used by the video card and BIOS.



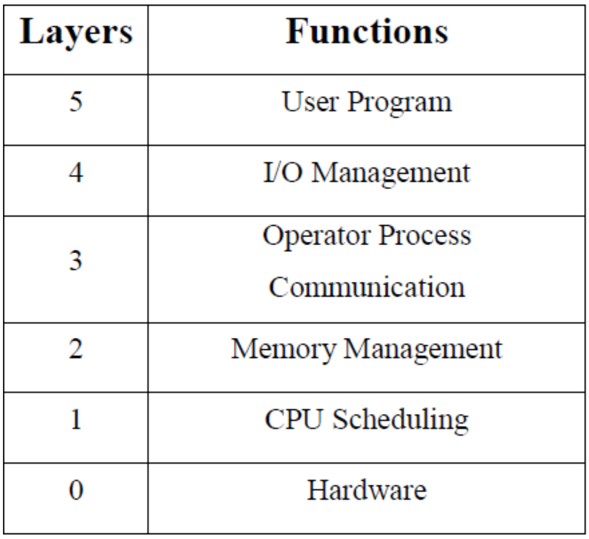
Simple Monolithic Operating Systems Example

3. 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 layer (layer N) is 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.
- 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 & so on.
- 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 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).

4. Microkernel approach:

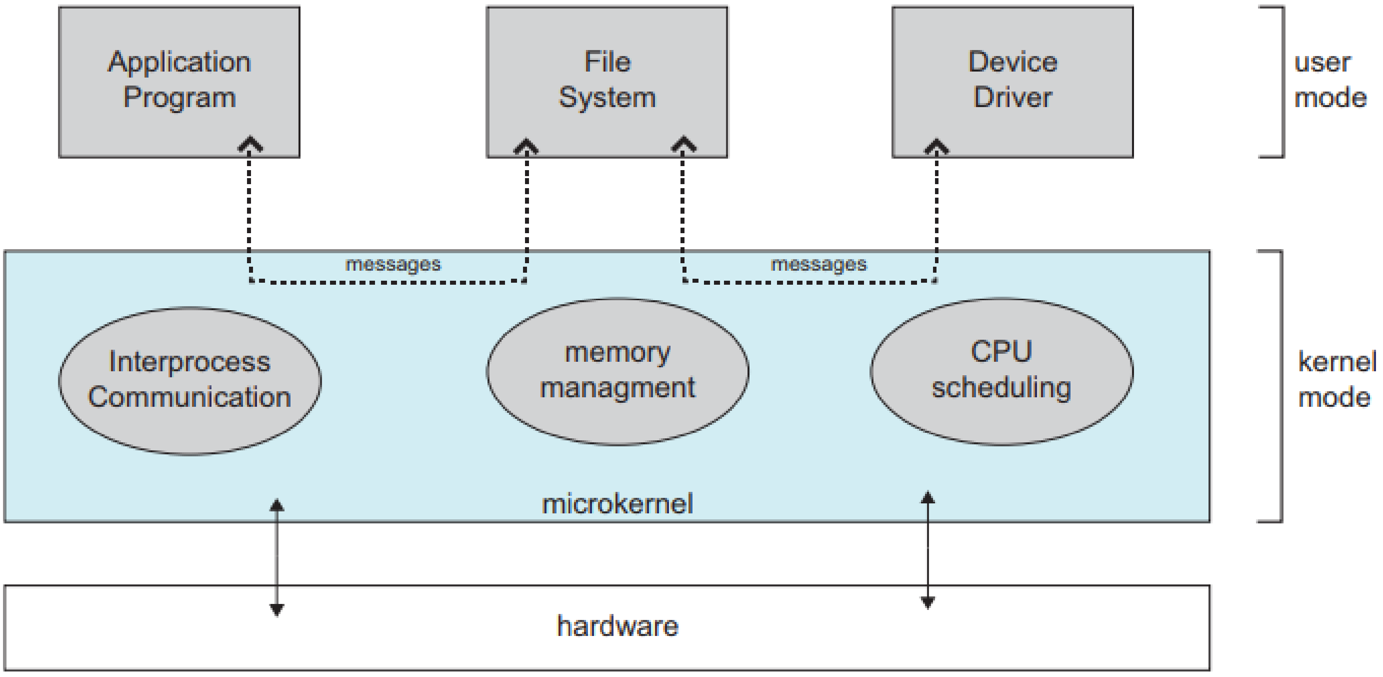
This structures the operating system by removing all nonessential portions of the kernel and implementing them as system and user level programs.

- Generally they provide minimal process and memory management, and a communications facility.
- Communication between components of the OS is provided by message passing.

The ***benefits*** of the microkernel are as follows:

- Extending the operating system becomes much easier.
- Any changes to the kernel tend to be fewer, since the kernel is smaller.
- The microkernel also provides more security and reliability.

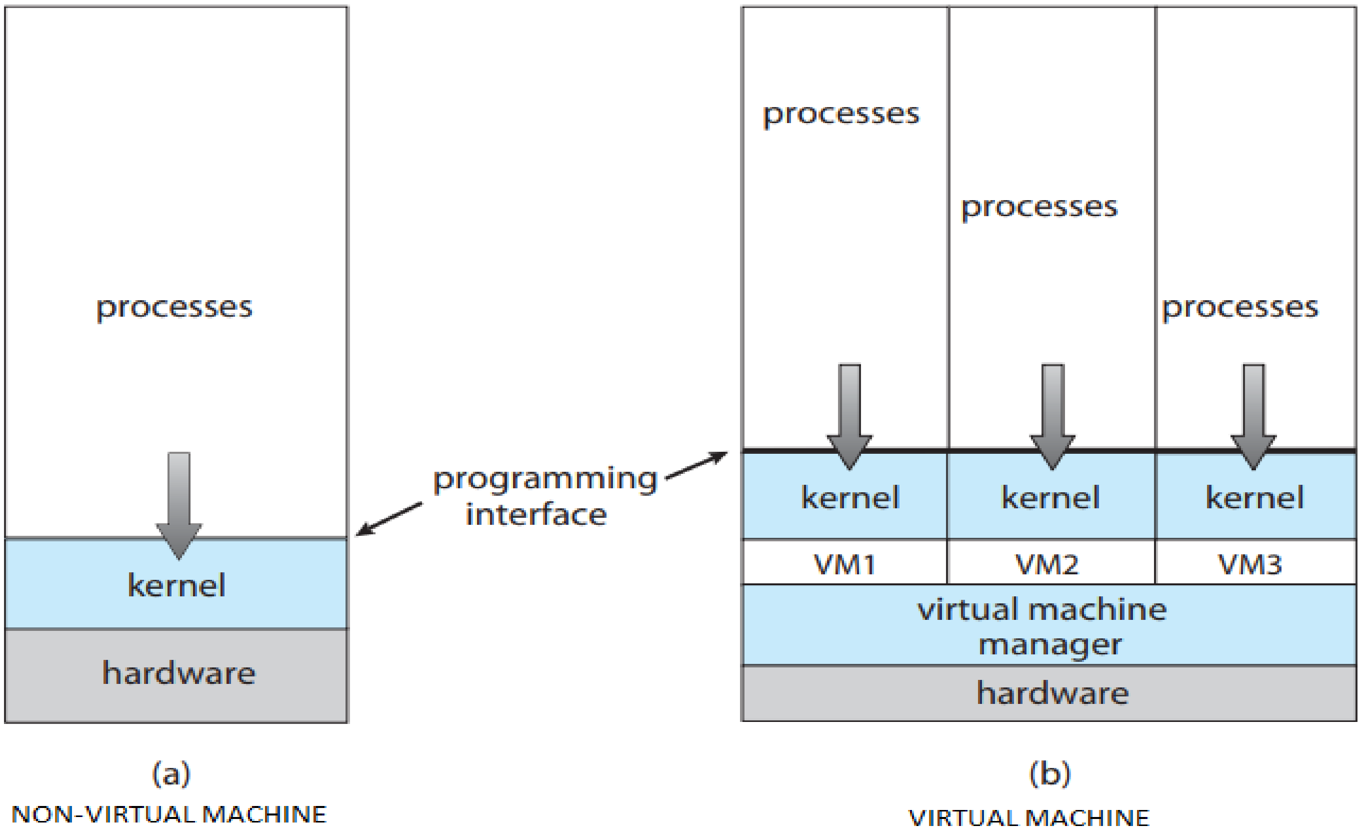
Main ***disadvantage*** is poor performance due to increased system overhead from message passing.



Architecture of a typical microkernel.

❖ Virtual Machines

- A virtual machine takes the layered approach to its logical conclusion.
- It treats hardware and the operating system kernel as though they were all hardware.
- A virtual machine provides an interface identical to the underlying bare hardware.
- The operating system creates the illusion of multiple processes, each executing on its own processor with its own (virtual) memory.
- The resources of the physical computer are shared to create the virtual machines.
- CPU scheduling can create the appearance that users have their own processor.
- Spooling and a file system can provide virtual card readers and virtual line printers.
- A normal user time-sharing terminal serves as the virtual machine operator's console.



VMware

Advantages and Disadvantages of Virtual Machines:

- The virtual-machine concept provides complete protection of system resources since each virtual machine is isolated from all other virtual machines. This isolation, however, permits no direct sharing of resources.

- A virtual-machine system is a perfect vehicle for operating-systems research and development. System development is done on the virtual machine, instead of on a physical machine and so does not disrupt normal system operation.
- The virtual machine concept is difficult to implement due to the effort required to provide an *exact* duplicate to the underlying machine.

Drawback:

- Virtual Machine includes the increased system overhead which is due to simulation of virtual machine operation heavily.
- The efficiency of VM OS depends upon the number of operations that must be simulated by VM monitor.