Unit I: Introduction

Syllabus

Computer System - Elements and organization; Operating System Overview - Objectives and Functions - Evolution of Operating System; Operating System Structures - Operating System Services User Operating System Interface - System Calls - System Programs - Design and Implementation - Structuring methods.

Computer System

AU May-17, 18, 22, Dec-17

• Computer system consists of hardware device and software that are combined to provide a tool to user for solving problems.

• Fig. 1.1.1 shows modern computer system.

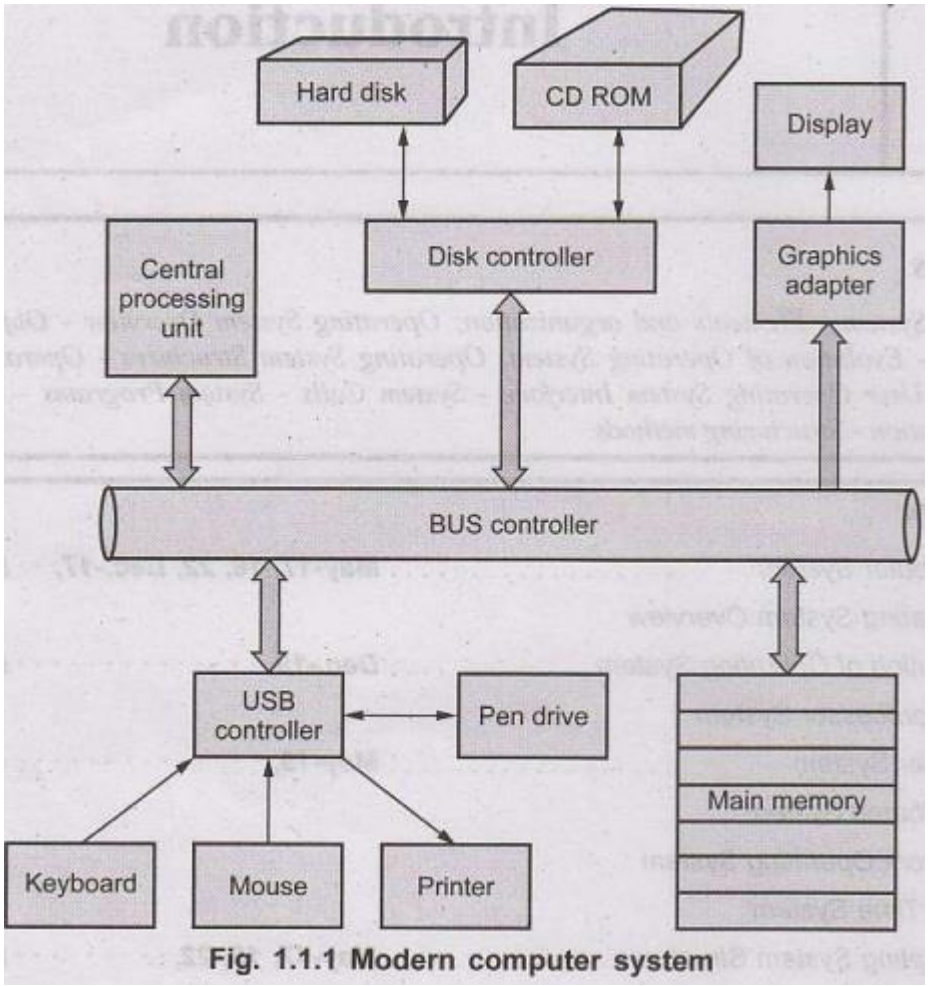


Fig. 1.1.1 Modern computer system

• Modern computer consists of one or two CPU with main memory and various I/O devices. Common bus is used for communication between these devices. Each device has its own device controller.

• CPU and device controller uses memory cycle for execution purposes. But memory cycle is only available to one device at a time.

• Bootstrap program is loaded when user start the computer. It initialies all the device connected to the computer system and then loads required device drivers.

• After this, operating system loads in the computer system. In UNIX OS, an 'init' is the first process which execute by OS.

• Interrupt is software and hardware. It is used to send signal to CPU. Software interrupt is sometime called system call.

• When interrupt is trigger, the CPU stops executing the instruction and control is transfer to the fixed location. Starting address is stored at fixed location where the service routine executes.

• Interrupts do not alter the control flow of the process executing on the processor.

Storage structure

• Processor access the data from main memory before executing any instruction. Main memory is also called Random Access Memory (RAM).

• DRAM is used in main memory. Fig. 1.1.2 shows hierarchy of storage device.

• At the top of the hierarchy, we have storage on the CPU registers. For accessing the CPU, it is fastest form of storage.

• Cache memory capacity is less than 1 MB.

• User program and data are stored in the main memory. Main memory is volatile, so it can not stored permanently.

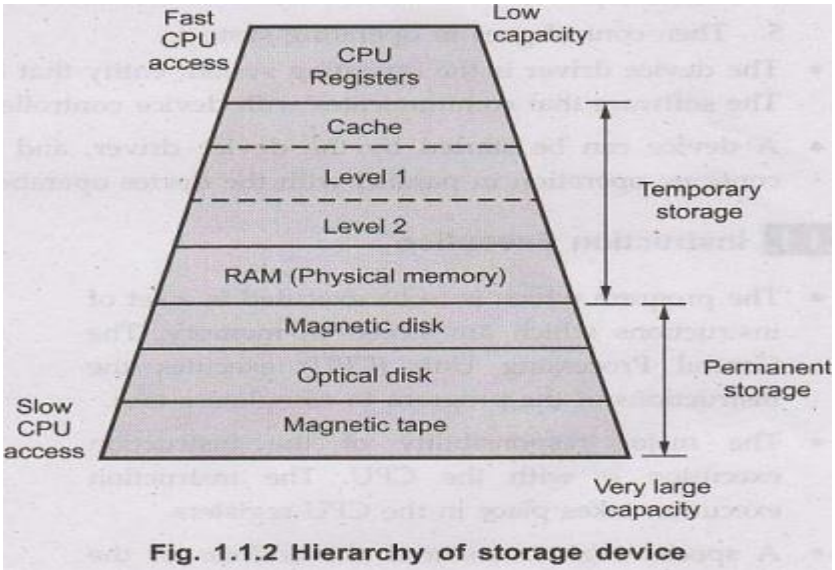


Fig. 1.1.2 Hierarchy of storage device

• Storage system is classified as temporary storage or permanent storage.

• Top level storage devices are low capacity with faster CPU access and bottom level storage devices having very large capacity with slow CPU access speed.

I/O structure

• Every device uses a device controller to connect it to the computer's address and data bus. Devices can be classified as a block oriented or character oriented, depending on the number of bytes transferred on an individual operation.

• Storage devices are used to store data while the computer is off.

• All the I/O devices are connected to each other by using common bus. CPU and main memory is also connected with this bus.

• Various types of controller is used in the computer system. Small Computer System Interface (SCSI) controller can handle upto seven devices. Each device controller have its own buffer.

• Device controller manage the data transfer between peripheral device and its controller. Device driver is handled by device controller.

I/O operation steps

1. Device driver loads the registers within the device controller.

2. Device controller takes action according to the data loaded into the register.

3. Data is transfer from device to its local buffer with the help of device controller.

4. After completion of data transfer, the device controller sends an interrupt signal to device driver about data transfer completion operation.

5. Then control goes to operating system.

• The device driver is the operating system entity that controls CPU-I/O parallelism. The software that communicates with device controller is called device driver.

• A device can be started by the device driver, and the application program can continue operation in parallel with the device operation.

Instruction Execution

• The program which is to be executed is a set of instructions which are stored in memory. The Central Processing Unit (CPU) executes the instructions of the program to complete a task.

• The major responsibility of the instruction execution is with the CPU. The instruction execution takes place in the CPU registers.

• A special register contains the address of the instruction. The CPU "fetches" the instruction from memory at that address.

• The CPU "decodes" the instruction to figure out what to do. The CPU "fetches" any data b(operands) needed by the instruction, from memory or registers.

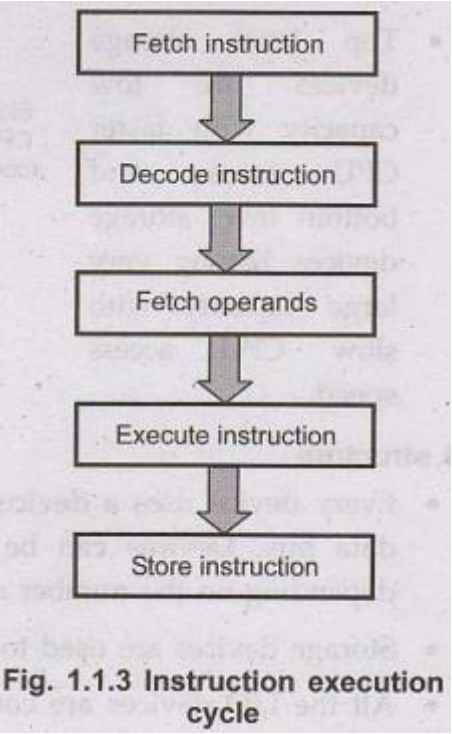


Fig. 1.1.3 Instruction execution cycle

• Fig 1.1.3 shows instruction execution cycle.

• The CPU "executes" the operation specified by the instruction on this data. The to CPU "stores" any results into a register or memory.

•The register used for instruction execution are as follows:

1. Memory Address Register (MAR): It specifies the address of memory location from which data or instruction is to be accessed (for read operation) or to best yr which the data is to be stored (for write operation).

2. Program Counter (PC): It keeps track of the instruction which is to be executed next, after the execution of an on-going instruction.

3. Instruction Register (IR): Here the instructions are loaded before their execution.

Interrupts

• **Definition:** It is an event external to the currently executing process that causes a change in the normal flow of instruction execution; usually generated by hardware not devices external to the CPU. They tell the CPU to stop its current activities and not execute the appropriate part of the operating system.

• Fig. 1.1.4 shows interrupts.

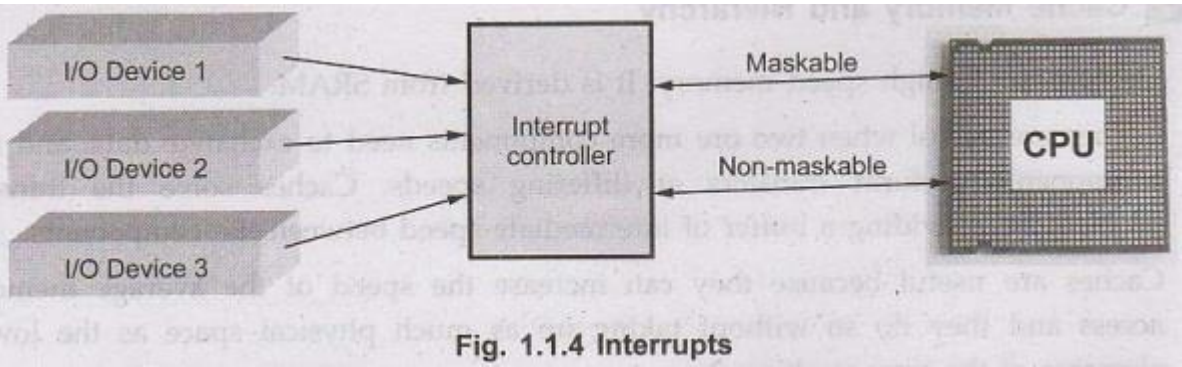


Fig. 1.1.4 Interrupts

• The CPU uses a table and the interrupt vector to find OS the code to execute in response to interrupts. When interrupt signaled, processor executes a routine called an interrupt handler to deal with the interrupt.

• Operating system may specify a set of instructions, called an interrupt handler, to be executed in response to each type of interrupt.

• Interrupt Service Routine (ISR) is the software code that is executed when the hardware requests interrupt. The design of the interrupt service routine requires careful consideration of many factors. Although interrupt handlers can create and use local variables, parameter passing between threads must be implemented using shared global memory variables.

• A private global variable can be used if an interrupt thread wishes to pass information to itself, e.g., from one interrupt instance to another. The execution of the main program is called the foreground thread, and the executions of the various interrupt service routines are called background threads.

• Interrupts are of three types: **Hardware, software and trap.**

• **Hardware Interrupts** are generated by hardware devices to signal that they need some attention from the OS. **Software Interrupts** are generated by programs when they want to request a system call to be performed by the operating system.

• Traps are generated by the CPU itself to indicate that some error or condition occurred for which assistance from the operating system is needed.

• Interrupt and trap numbers are defined by the hardware which is also responsible for calling the procedure in the kernel space. An interrupt handler is

called in response to a signal from another device while a trap handler is called in response to an instruction executed within the CPU.

• Synchronous interrupts occur when a process attempts to perform an illegal action, such as dividing by zero or referencing a protected memory location. Synchronous interrupt occurs due to software execution.

Cache Memory and Hierarchy

• Cache is small high speed memory. It is derived from SRAM.

• Caches are useful when two ore more components need to exchange data, and the components perform transfers at differing speeds. Caches solve the transfer problem by providing a buffer of intermediate speed between the components.

• Caches are useful because they can increase the speed of the average memory access and they do so without taking up as much physical space as the lower elements of the memory hierachy.

• Data is normally kept in storage system.. when it required, it is copied into a faster storage system i.e. cache memory for temporary basics.

• When user required a particular data, system check whether it is in the cache. If data found then, we use the data directly from the cache. It is not found then, use the data from the source.

• Internal programmable registers, such as index registers, provide a high-speed cache for main memory. The programmer implements the registor-allocation and register-replacement algorithms to decide which information to keep in registers and which to keep in main memory.

• If the fast device finds the data it needs in the cache, it need not wait for the 1er slower device. The data in the cache must be kept consistent with the data in the components.

• If a component has data value change, and the datum is also in the cache, the cache must also be updated. This is especially a problem on multiprocessor systems where more than one process may be accessing a datum.

• A component may be eliminated by an equal sized cache, but only if the cache and the component have equivalent state-saving capacity and the cache is affordable, because aster storage tends to be more expensive.

• Unfortunately, cache also introduce an additional level of complexity (coherency and consistency assurance).We also incur an economic and space penalty when we add a cache.

• Making a cache as large as a disk would be ineffective because it would be too costly, the immense size would slow it down and a cache is generally a volatile memory, while we want data on disks to be persistent..

• It holds data for temporary purpose to reduce the time required to service I/O requests from the host. Fig. 1.1.5 shows memory hierarchy.

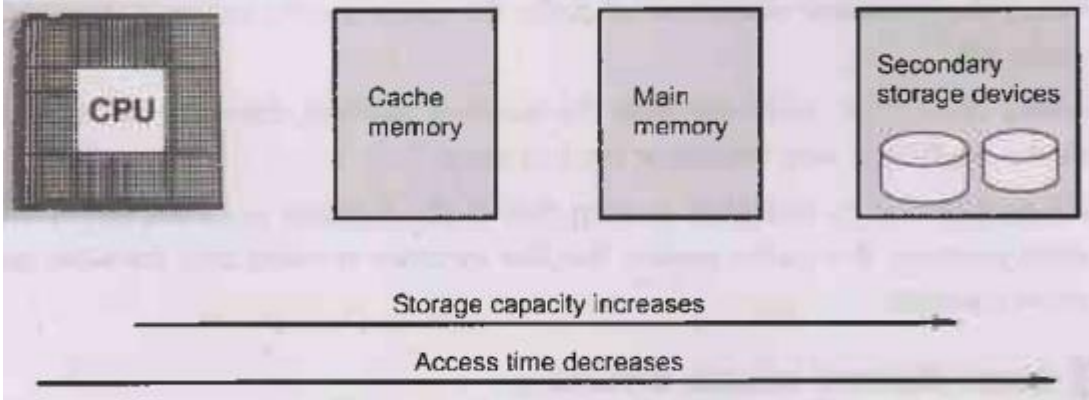


Fig. 1.1.5 Memory hierarchy

• Cache in CPU unit is referred as Level1 cache (L1 cache) and cache stored in a chip next to CPU is termed as Level2 cache (L2 cache) and it resides on the motherboard.

• The function of cache is to act as a buffer between a relatively fast device and a relatively slow one.

• Small unit of allocation in cache is page. Cache is arranged into slots or pages. It uses two components data store and tag RAM. The actual data is stored in a

different part of the cache, called the data store. The values stored in the tag RAM determine whether a cache lookup results in a hit or a miss.

• The size of the tag RAM determines what range of main memory can be cached. Tag RAM is a small piece of SRAM that stores the addresses of the data that is stored in the SRAM

• A cache address can be specified simply by index and offset. The tag is kept to allow the cache to translate from a cache address to a unique CPU address.

• A cache hit means that the CPU tried to access an address, and a matching cache block was available in cache. So, the cache did not need to access RAM. In a cache miss, the CPU tries to access an address, and there is no matching cache block. So, the cache is forced to access RAM. of

• Cache includes tags to identify which block of main memory is in each cache slot.

• Every cache block has associated with it at least the modify and valid bits, and a tag address. The valid bit says if the cache block is used or is unused. The modify bit makes sense only if the valid bit is set. The modify bit says whether the data in the cache block is different from RAM or is the same as RAM.

• The size of a page is dependent on the size of the cache and how the cache is organized. A cache page is broken into smaller pieces, each called a cache line. The size of a cache line is determined by both the processor and the cache design. .

• When the processor starts a read cycle, the cache checks to see if that address is a cache hit.

• **Cache Hit:** If the cache contains the memory location, then the cache will respond to the read cycle and terminate the bus cycle.

• **Cache Miss:** It is reference to item that is not resident in cache, but is resident in main memory. For cache misses, the fast memory is cache and the slow memory is main memory.

Direct Memory Access Structure

• Direct Memory Access (DMA) is one of several methods for coordinating the timing of data transfers between an input/output (I/O) device and the core processing unit or memory in a computer. DMA is one of the faster types of synchronization mechanisms.

• Without DMA, the processor is responsible for the physical movement of data between main memory and a device. A special control unit may be provided to allow transfer of a block of data directly between an external device and the main memory, without continuous intervention by the processor. This approach is called direct memory access.

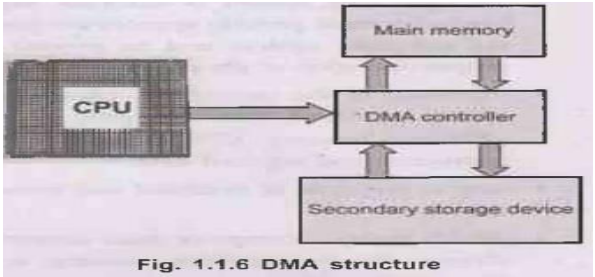


Fig. 1.1.6 DMA structure

• The DMA controller uses hold request (HOLD) and hold girl acknowledge (HOLD A) signals to ask the CPU to stop driving the address, data and control buses so that the DMA controller can drive them to carry out a transfer.

• A DMA controller implements direct memory access in a computer system. It connects directly to the I/O device at one end and to the system buses at the other end. It also interacts with the CPU, both via the system buses and two new direct connections.

• Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention. Only one interrupt is generated per block, rather than the one interrupt per byte.

• The DMA controller may either stop the CPU and access the memory or use the bus while the CPU is not using it

Operating System Overview

• **OS definition:** *Operating System is a program that controls the execution of application programs. It is an interface between applications and hardware.*

• OS provides different types of view. For user, it is abstract view because it provides features which are important for users. OS is intermediary between user and the computer system.

• The major design goals/ functions of an operating system are :

1. Efficient use of a computer system 2. User convenience 3. Ability to evolve.

• An operating system is software that manages the computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

• Efficiency is the one of the parameter for use of computer system. Operating system consumes some resources during its own operation. For example it uses CPU and memory. CPU is busy with scheduling and memory is occupied by instruction and required data.

• This is one type of overhead and because of this lesser resources are available to the user.

• An OS makes a computer more convenient to use. If the operating system can notallocates the free available resources to program or it over allocates the resources then efficiency is poor.

Efficient use

• For efficient use of resources, it must be monitored by operating system. Proper bold scheduling of resources is also required.

• Computer contains different type's resources like CPU, memory and I/O device etc. Proper monitoring is required on these resources to avoid the overhead. As per the resource, scheduling is required.

• Special attention to be given for CPU and memory. If memory is not free then user can not load new program into the memory. Then CPU will be busy with memory management.

User convenience

• User convenience is affected by computing environment of the computer system. The computing environment is comprised of computer system, its interfaces with other systems and nature of computations performed by its users.

• Computer architecture and use change the computing environment of the system. Following factors are considered while considering user convenience:

1. Good service 2. Ease of use 3. New programming model 4. Evolution 5. User friendly OS

Ability to evolve

• An OS should be constructed in such a way as to permit the effective en development, testing and introduction of new system functions without at the same time interfering with service.

• Task performed by operating systems :

1. Maintain the list of resources in the system
2. Maintain the list of authorized users
3. Initiate the execution of programs and process
4. Maintain resource usage list
5. Maintain the resource allocated list
6. Scheduling of resources (CPU, Secondary storage etc.)
7. Also maintain the protection information.

Operating System as a User Interface

• Computer system consists of software and hardware to solve specific problems. User, application program, operating system and the hardware are the components of the computer systems.

• Application program used to solve specific program. Student attendance no no monitoring is the example of application program.

• Operating system is a subset of the system software. OS interacts directly with the hardware to provide an interface to other system software.

• System software and hardware exist to support the creation and effective use application software.

• Fig. 1.2.1 shows conceptual view of a computer system.

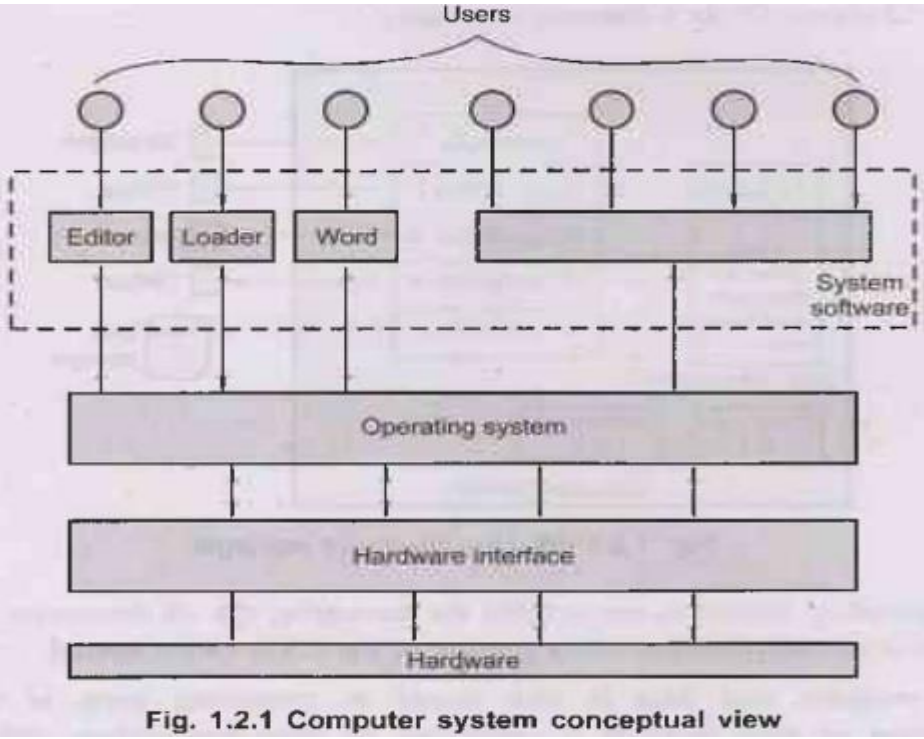


Fig. 1.2.1 Computer system conceptual view

• Resource sharing and resource abstraction are two key aspects of the operating system.

• Purposes of the operating system :

1. OS provides an interface between the computer hardware and computer user (programmer). It simplifies the programmer job like editing, coding, creation.

2. Allocation and use of computer resources among the programmer is controlled arnaldo it by OS.

Operating System as Resource Manager

• A computer is a set of resources. These resource provides various functions to the user. Functions like data movement, storing of data and program, operation on data are control by an operating system.

• Fig. 1.2.2 shows OS as a resource manager.

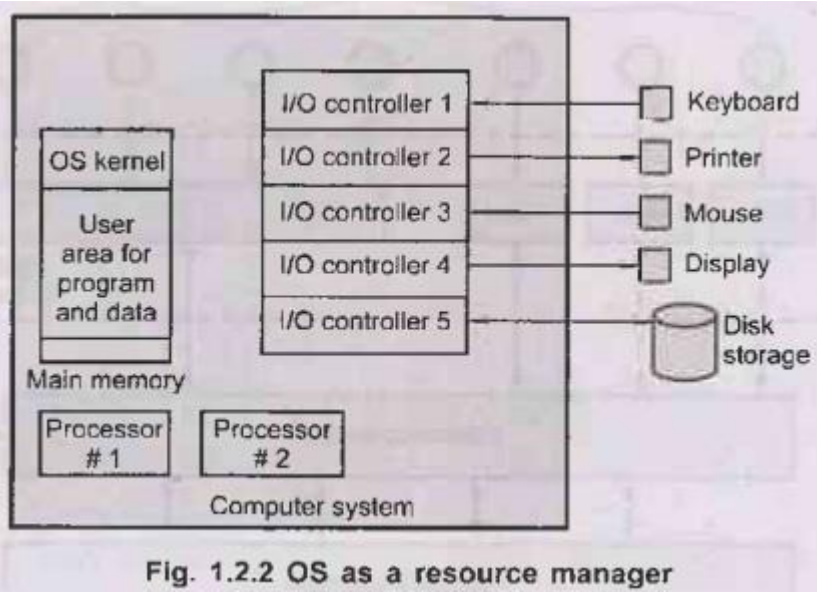


Fig. 1.2.2 OS as a resource manager

• The operating system is responsible for managing the all resources. A portion of the OS is in main memory. This portion of the OS is called kernel.

• User program and data is also stored in remaining parts of the memory. Allocation of main memory is controlled by operating system with the help of memory management hardware.

• I/O device is controlled by OS and it decides when an I/O device can be used by program in execution. Processor is one type of resource and OS control the execution of user program on the processor.

• Modern OS allows multiple programs to run at the same time. If multiple users are using computer then there is need of managing and protecting the memory, I/O devices and other resources.

• Resource management includes sharing resources in different ways. Time and space are the two concept for resource sharing.

1. Time: Time slot is allocated to each program first one gets to use the resource then another and so on.

2. Space: Consider the example of main memory. Main memory is normally divided up among several running programs, so each one can be resident at the same time.

Characteristics of Modern Operating Systems

1. Microkernel architecture assigns only a few major functions to the kernel. It includes basic CPU scheduling, IPC, allocation of address space etc.

2. Process: Collection of threads.

3. Thread Thread is light weight process. A thread is a flow of execution through the process code. It contains program counter, stack pointer and stack for data storage.

4. Multithreading: Process creates number of multiple threads. These all thread can run simultaneously.

5. Distributed operating systems: It provides transparency to the end user.

Evolution of Operating System

AU: DEC-19

• Types of an operating system are a grouping that differentiates or identifies the operating system based on how it works the type of hardware it controls and the application it supports.

• Evolution of an operating systems are as follows:

a) Early systems/serial processing (mid 1940s to mid 1950s)

b) Simple batch systems (1960)

c) Multi-programmed batch systems (1970)

d) Time-sharing and real-time systems (1970)

e) Multiprocessor systems (1980)

f) Networked/distributed systems (1980)

Batch System

• Batch system process a collection of jobs, called a batch. Batch is a sequence of user jobs.

• Job is a predefined sequence of commands, programs and data that are combined into a single unit.

• Each job in the batch is independent of other jobs in the batch. A user can define a job control specification by constructing a file with a sequence of commands.

• Jobs with similar needs were batched together to speed up processing. Card readers and tape drives are the input device in batch systems. Output devices are tape drives, card punches and line printers.

• Primary function of the batch system is to service the jobs in a batch one after another without requiring the operator's intervention. There is no need for human/user interaction with the job when it runs, since all the information required to complete job is kept in files.

• Some computer systems only did one thing at a time. They had a list of instructions to carry out and these would be carried out one after the other. This is called a serial system. The mechanics of development and preparation of programs in such environments are quite slow and numerous manual operations non involved in the process.

• Batch monitor is used to implement batch processing system. Batch monitor is also called kernel. Kernel resides in one part of the computer main memory.

• The memory allocator managed the main memory space. Batch monitor controls the sequence of events. Main memory store the batch monitor and users program and data (jobs). Fig. 1.3.1 shows memory layout for a batch system.

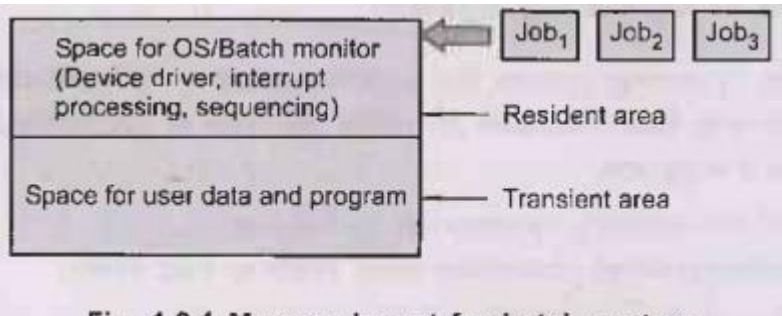


Fig. 1.3.1 Memory layout for batch system

• Computer operator gives a command to start the processing of a batch, the kernel sets up the processing of the first job. Job was selected from the job queue and loaded into main memory. When a job completed execution, its memory was released and the output for the job was copied.

• When a job is completed, it returns control to the monitor, which immediately reads in the next job.

• Fig. 1.3.2 shows concept of batch system.

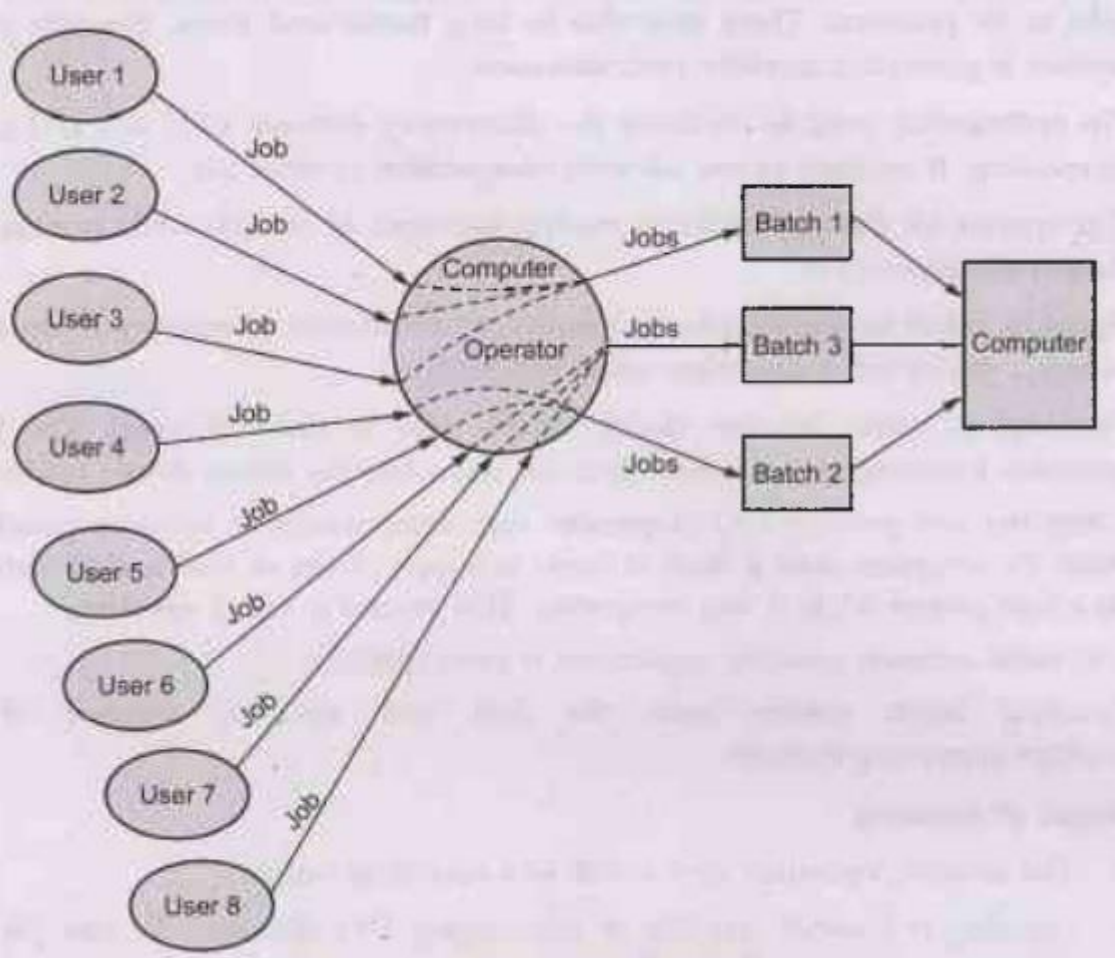


Fig. 1.3.2 Concept of batch system

• Scheduling is also simple in batch system. Jobs are processed in the order of submission i.e. first come first served fashion.

• When a job completes execution, its memory is released and the output for the job gets copied into an output spool for later printing.

Spooling

• Spooling an acronym for simultaneous peripheral operation on line. Spooling uses the disk as a large buffer for outputting data to printers and other devices. It can also be used for input, but is generally used for output.

• Its main use is to prevent two users from alternating printing lines to the line printer on the same page, getting their output completely mixed together. It also helps in reducing idle time and overlapped I/O and CPU.

• Batch system often provides simple forms of file management. Access to file is serial. Batch systems do not require any time critical device management.

• Batch systems are inconvenient for users because users cannot interact with their jobs to fix problems. There may also be long turnaround times. Example of this system is generating monthly bank statement.

• An optimization used to minimize the discrepancy between CPU and I/O speeds is spooling. It overlaps of one job with computation of other job.

• The spooler for instance could be reading the input of one job while printing the output of different job.

• Spooling refers to putting jobs in a buffer, a special area in memory or on a disk where a device can access them when it is ready.

• Spooling is useful because device access data at different rates. The buffer provides a waiting station where data can rest while the slower device catches up.

• Computer can perform I/O in parallel with computation, it becomes possible to have the computer read a deck of cards to a tape, drum or disk and to write out to a tape printer while it was computing. This process is called spooling.

• The most common spooling application is print spooling.

• Spooling batch system were the first and are the simplest of the multiprogramming systems.

Advantages of spooling

1. The spooling operation uses a disk as a very large buffer.

2. Spooling is however capable of overlapping I/O operation for one job with processor operations for another job.

Advantages and Disadvantages of Batch System

Advantages of batch system

1. Move much of the work of the operator to the computer.

2. Increased performance since it was possible for job to start as soon as the previous job finished.

Disadvantages of batch system

1. Turn around time can be large from user standpoint.

2. Program debugging is difficult.

3. There was possibility of entering jobs in infinite loop.

4. A job could corrupt the monitor, thus affecting pending jobs.

5. Due to lack of protection scheme, one batch job can affect pending jobs.

Multi-programing Operating System

• CPU remains idle in batch system. At any time either CPU or I/O device was idle in batch system. To keep CPU busy, more than one program/job must be loaded for execution. It increases the CPU utilizations. So multiprogramming increases the CPU utilization.

• Resource management is the main aim of multiprogramming operating system. File system, command processor, I/O control system and transient area are the essential components of a single user operating system.

• Multiprogramming operating system divides the transient area to store the multiple programs and provides resource management to the operating system.

• The concurrent execution of programs improves the utilization of system resources. A program in execution is called a "Process", bini "Job" or a "Task".

• When two or more programs are in the memory at the same time, sharing the processor is referred to the multiprogramming operating system.

• Fig. 1.3.3 shows the memory layout for a multiprogramming operating system.

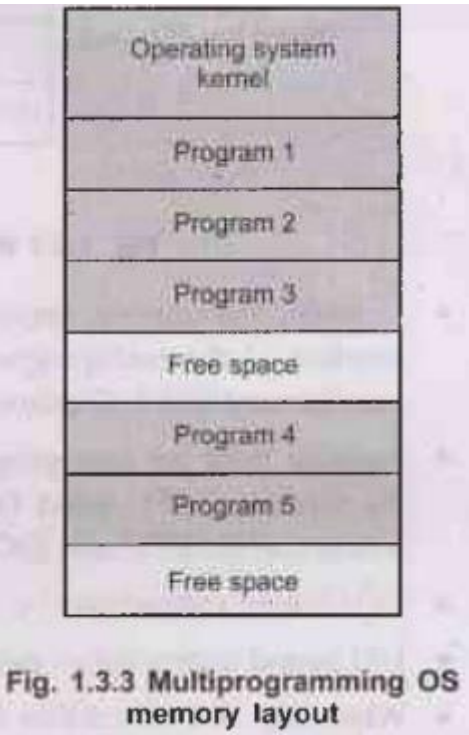


Fig. 1.3.3 Multiprogramming OS memory layout

• Operating system keeps number of programs into the memory. It selects one program from the memory and executes it. All the programs that enter the system are kept in the job pool.

• Job pool consists of all processes residing on disk and waiting to allocate primary memory. Job scheduling concept is used if there is no space for process in the primary memory.

• Memory management is also required to manage the memory for process. CPU scheduling is applied for selecting process from memory.

• When computer loads more than one program in to the memory, CPU executes one program and I/O system is busy with other programs.

• Multiprogramming operating system do not provide user interaction with the program.

• Fig. 1.3.4 shows working of multiprogramming OS.

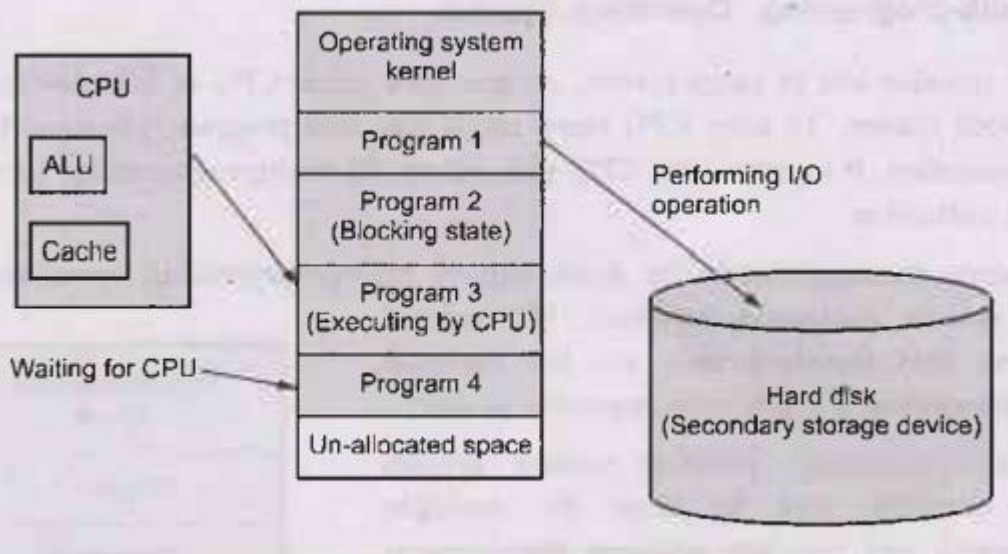


Fig. 1.3.4 Working of multiprogramming OS

• In multiprogramming operating system, programs are competing for resources. A function of the multiprogramming operating system is CPU scheduling, memory management and I/O management.

• Suppose there are four programs for execution. All four programs are loaded into the memory. CPU select first program for execution. Normally programs contain instruction for CPU and I/O operation.

• CPU bound instructions: c = a + b

• I/O bound instructions: printf, scanf etc.

• When any I/O instruction is encounter in the program, CPU select next program for processing. CPU select second program for execution and I/O system select first program for performing I/O operation. Multiprogramming operating system monitors the state of all active programs and system resources.

Advantages:

1. CPU utilization is high.

2. It increases the degree of multiprogramming.

Disadvantages :

1. CPU scheduling is required.

2. Memory management is also required.

Time Sharing System

• Time sharing is also called multitasking operating system. It is logical extension of the multiprogramming operating systems.

• User interaction with program is possible in time sharing operating system. During execution of the program, user interacts directly with the program, supplying information to the program.

• Multi-tasking means that the computer can work with more than one program at a time. For example, user could be working with information from one database on the screen analyzing data, while the computer is sorting information from another database, while a excel sheet is performing calculations on a separate worksheet.

• Many users share the computer system simultaneously in time sharing operating system. Time sharing system uses multiprogramming and CPU scheduling. Each user has at least one separate program in memory.

• In time sharing system, each user is given a time slice for executing his/her job in round robin fashion. Job continues until the time slice ends.

• Fig. 1.3.5 shows multitasking OS.

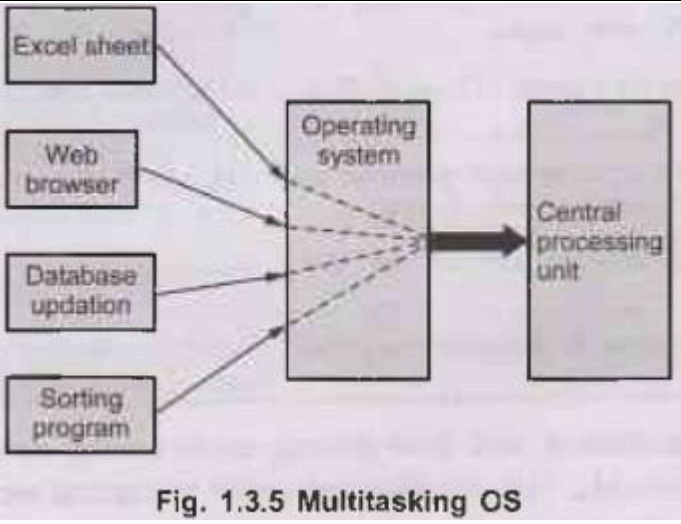


Fig. 1.3.5 Multitasking OS

• Concept of virtual machine is used in time sharing system. It creates virtual machine one per user. User interaction with system by using virtual machine. User enters the command for virtual machine and result will received back to user.

• Time sharing system is more complex than multiprogramming operating system. It So disk sni also takes help of file system. File system is stored on the disk management is also required.

• Major problem with time sharing system is protection and security of data.

• Time sharing system uses medium term scheduling such as round robin for the foreground. Background process uses can use a different scheduling method.

• Difference between multiprogramming and multitasking operating system is context switching. In multiprogramming system a context switching occurs only when the currently executing process stalls for some reasons.

• Time sharing system gives each user the impression that the entire system is dedicated to his use. Context switching simply allows several applications to be open, but only one is working at a time.

• Truly speaking, even in true multi-tasking, only one application program is ever running at anyone instant. Because the computer automatically switches from one program to the next program so quickly, all the programs seem to run don simultaneously.

Difference between Time sharing Os and multiprogramming Os

Sr. No.	Time sharing OS	Multiprogramming OS
1.	It enables execution of multiple tasks and processes at the same processes to increases CPU performance.	Multiple programs reside in the main memory simultaneously to improve CPU utilization.
2.	It is based on the concept of time sharing.	It is based on the concept of context switching.
3.	The idea is to allow multiple processes to run simultaneously via time sharing.	The idea is to reduce the CPU idle time for as long as possible.
4.	It takes less time to execute the task allocation.	It takes more time to execute the process.
5.	Principle objective is minimize response time.	Principle objective is maximize processor use.

• In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation can result in various security problems. Such problems are privacy, integrity and denial of service.

a) Privacy: One user can read the private data of another user.

b) Integrity: One user can corrupt the private data of another user**.**

c) Denial of Service: One user can prevent another user from getting anything done.

• Can we get the same degree of security in a time-shared machine as we have in a dedicated machine? Solution to this is Yes/No. There are two answers, either one correct.

a) Yes: If we can ensure that the operating system prevents any sharing of data between users, either for reading or writing and fairly shares the computer, then we can achieve the same level of security.

b) No: We can never be sure that our software doesn't have bugs, so we can never be sure that we prevent all sharing of data and fairly allocate computer resources.

Multiprocessor System

• Multiprocessor system means more than one processor in close communication. All the processor share common bus, clock, memory and peripheral devices.

• Multiprocessor system is also called parallel system or tightly coupled systems. Multiprocessor operating systems are just regular operating systems. They handle system calls, do memory management, provide a file system, and manage I/O devices.

• Multiprocessor systems is also known as parallel systems or multicore systems.

•Features of Multiprocessor Systems :

1.The processor should support efficient context switching operation.

2. It supports large physical address space and larger virtual address space.

3. If one processor fails, then other processors should retrieve the interrupted process state so that execution of the process can continue.

4.The inter-process communication mechanism should be provided and implemented in hardware as it becomes efficient and easy.

• Multiprocessors can be used in different ways:

1. Uniprossesors (single-instruction, single-data or SISD)

2. Within a single system to execute multiple, independent sequences of instructions in multiple contexts (multiple-instruction, multiple-data or MIMD);
3. 3.A single sequence of instructions in multiple contexts (single-instruction, multiple-data or SIMD, often used in vector processing);

4. Multiple sequences of instructions in a single context (multiple-instruction, single-data or MISD, used for redundancy in fail-safe systems and sometimes applied to describe pipelined processors or hyper threading).

Advantages and Disadvantages of Multiprocessor Systems

Advantages of Multiprocessor Systems:

1. Throughput : Throughput increases because number of processor is increases.

2. Cost: Multiprocessor system is cheaper than the multiple single processor Ang system.

3. Reliability: If one processor fails, it has no effect on whole system operation.

4. Response time : Response time is less because of increased number of processor.

• Disadvantages of Multiprocessor Systems :

1. Multiprocessor systems are more complex in both hardware and software.

2. Additional CPU cycles are required to manage the cooperation, so per-CPU efficiency goes down.

• Multiprocessor systems are of two types: Symmetric multiprocessing and asymmetric multiprocessing.

Symmetric Multiprocessing

• The simplest multiprocessors are based on a single bus. In symmetric multiprocessing, number of homogeneous processor are running independently without affecting other programs.

• Systems that treat all CPUs equally are called symmetric multiprocessing (SMP) systems.

• Each processor uses 'different data and program but sharing some common resources like memory, I/O device etc. Fig. 1.4.1 shows multiprogramming system.

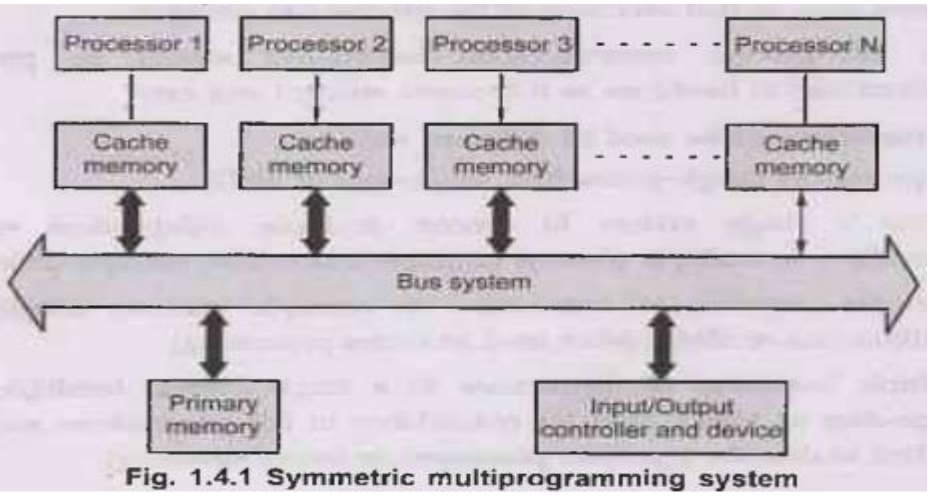


Fig. 1.4.1 Symmetric multiprogramming system

• In SMP, all processor are at the same level. They work in peers. There is no master-slave relationship in between the processor. Each processor shares a single copy of operating system.

• Symmetric multiprocessing is easier to implement in operating systems. Examples of symmetric multiprocessing operating systems are Windows NT 4.0, Novell Netware 2.1 etc.

Asymmetric Multiprocessor

• If all CPUs are not equal, system resources may be divided in a number of ways, including asymmetric multiprocessing (ASMP), non-uniform memory access

(NUMA) multiprocessing, and clustered multiprocessing.

• All CPUs are not equal. There is one master processor and remaining is the slave processor. Each processor has its own memory address space.

• A master processor controls the whole operations. It gives the instruction to the slave processor or slave processor uses some predefined set of tasks.

• Asymmetric multiprocessing has one master CPU and the remainder CPUs are slaves. The Master distributes takes among the slaves and I/O is usually done by the master only.

• Each processor has own task which assign by master processor. Asymmetric multiprocessing is difficult to implement.

Difference between Symmetric and Asymmetric Multiprocessor

Sr. No.	Symmetric Multiprocessor	Asymmetric Multiprocessor
1.	Symmetric multiprocessing treats all processors are equals, an I/O can be processed on any CPU.	Asymmetric multiprocessing has one master CPU and the remainder CPUs are slaves.
2.	Symmetric multiprocessing is easier to implement in operating systems.	Asymmetric multiprocessing is difficult to implement.
3.	Single OS manages all processor cores simultaneously.	Separate OS, or separate copy of same OS manage each core.
4.	Master-slave concept is not used.	It uses concept of master-slave concept.
5.	The processors communicate with each other through shared memory.	Shared memory is not used for communication.

Cluster System

AU:May19

• Clustered system is a group of computer system connected with a high speed communication link. Each computer system has its own memory and peripheral devices. Clustering provides high availability.

• Clustering systems are integrated with hardware cluster and software cluster. Hardware cluster support sharing of high performance disks. Software cluster is in the form of unified control of the computer system in a cluster.

• Cluster node contains layer of cluster software and it run on the node. Each node monitors the network whether the cluster is working properly or not. If the monitored computer fails, then the monitoring computer takes the control and ownership of its storage and other resources. It restarts the application of failed computer. The users and other clients will get the resources. The failed computer can remain down until it repair.

• Clustered systems are divided into two groups:

1. Asymmetric clustering 2. Symmetric clustering.

• **Asymmetric clustering:** In this type of clustering, one machine is always in the hot standby mode and other machines are running its applications. The hot standby machine only monitors the active server. When active server fails, then hot standby machine takes the control and becomes the active server.

• **Symmetric clustering:** more than one host runs the applications. It uses all of the available hardware for operation. Host monitors working of other host for smooth operation of the cluster.

• Fig. 1.5.1 shows structure of cluster system.

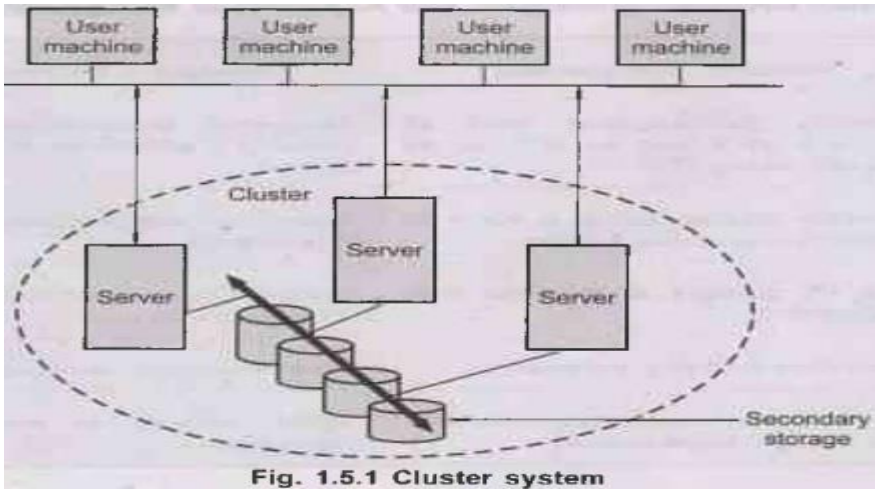


Fig. 1.5.1 Cluster system

• Clustered system and symmetric multiprocessor provide a support for high demand applications.

• Clustered system is different than parallel system.

• Other type of clustered system is parallel clusters. It uses shared storage of data. Parallel cluster requires some special purpose software and special type of applications. Example of parallel cluster is Oracle Parallel Server.

• Cluster technology changes rapidly. New developments are added to this cluster technology. Now a day, global cluster concept is used for large organizations.

Pros:

1. Increased resource availability.

2. Cluster computing is manageable and easy to implement.

3. Clustered systems are quite fault-tolerance and the loss of a single node does not result in the system's failure.

4. There is a high availability of resources meaning that if one node fails, another node takes over.

Cons:

1. Extra hardware is needed to maintain system.

2.Difficult to manage and organize a large number of computers.

Distributed System

• **Definition:** A distributed system is a collection of autonomous hosts that are connected through a computer network.

• A distributed system is a collection of independent computers that appears to its users as a single coherent system. Each host executes components and operates a distribution middleware.

• Middleware enables the components to coordinate their activities. Users perceive the system as a single, integrated computing facility.

• A distributed computer system consists of multiple software components that are on multiple computers, but run as a single system. The computers that are in a distributed system can be physically close together and connected by a local network, or they can be geographically distant and connected by a wide area network.

• A distributed system can consist of any number of possible configurations, such as

mainframes, personal computers, workstations, minicomputers and so on, such a

• Distributed operating systems depend on networking for their operation. Distributed OS runs on and controls the resources of multiple machines. It sts provides resource sharing across the boundaries of a single computer system. It looks to users like a single machine OS.

• Distributing OS owns the whole network and makes it look like a virtual otauni-processor or may be a virtual multiprocessor.

• **Definition:** A distributed operating system is one that looks to its users like an ordinary operating system but runs on multiple, independent CPU.

• Distributed systems depend on networking for their functionality. Fig. 1.6.1 shows the distributed system.

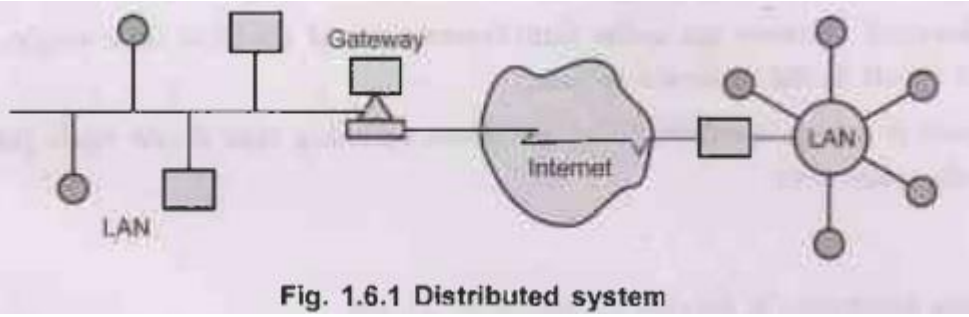


Fig. 1.6.1 Distributed system

• Examples of distributed operating system are Amoeba, chrous, mach and v-system

• A **Local Area Network** (LAN) is a network that is confined to a relatively small area. It is generally limited to a geographic area such as a college, lab or building.

• **WAN** provides long distance transmission of data and voice. Computers connected to a wide-area network are often connected through public networks, such as the telephone system. They can also be connected through leased lines or satellites.

• A MAN typically covers an area of between 5 and 50 km diameter. Many MANS cover an area the size of a city, although in some cases MANS may be as small as a group of buildings.

• A MAN often acts as a high speed network to allow sharing of regional resources. MAN provides the transfer rates from 34 to 150 Mbps.

Advantages and Disadvantages of DS

a) Advantages of distributed OS:

1. Resource sharing: Sharing of software resources such as software libraries, database and hardware resources such as hard disks, printers and CDROM can also be done in a very effective way among all the computers and the users.

2. Higher reliability: Reliability refers to the degree of tolerance against errors and component failures. Availability is one of the important aspects of reliability. Availability refers to the fraction of time for which a system is available for use.

3. Better price performance ratio: Reduction in the price of microprocessor and increasing computing power gives good price-performance ratio.

4. Shorter responses times and higher throughput.

5. Incremental growth: To extend power and functionality of a system by simply adding additional resources to the system.

b) Disadvantages :

1. There are no current commercially successful examples.

2. Protocol overhead can dominate computation costs.

3. Hard to build well.

4. Probably impossible to build at the scale of the internet.

Client-Server Computing

• The system is structured as a set of processes, called servers that offer services to the users, called clients.

• Server systems are of two types: Compute servers and file servers.

• **Compute-server system:** It provides an interface to which a client can send a request to perform an action. In response, the server performs some operation and sends the results to the client. Server contains database.

• **File-server system:** Client can performs various operation like create, read, update and delete file on file server. Web server is best example of this type.

• Fig. 1.6.2 shows the client server model.

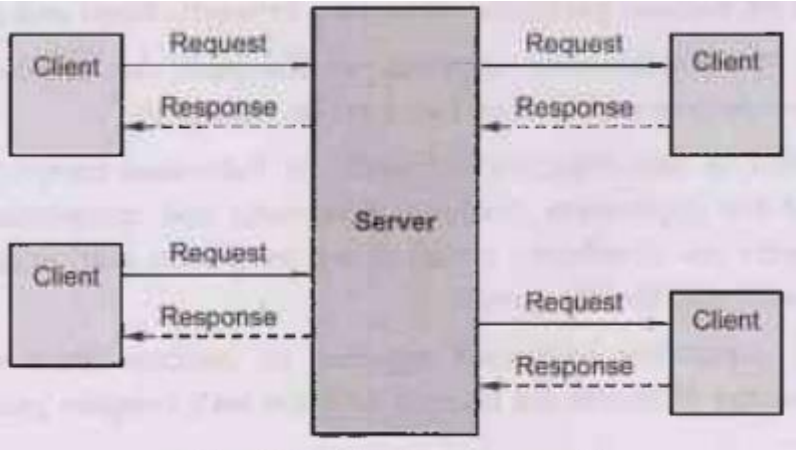


Fig. 1.6.2 Client server model

• The client-server model is usually based on a simple request/reply protocol, implemented with send/receive primitives or using Remote Procedure Calls (RPC) or Remote Method Invocation (RMI) :

1. The client sends a request (invocation) message to the server asking for some service;

2. The server does the work and returns a result (e.g. the data requested) or an error code if the work could not be performed.

• Server is a process; Client is a process. Clients invoke servers, servers send results to clients. Servers can be clients of other servers.

• HTTP (Web) server is a server process to its client processes (web browsers).

• HTTP server may be a client of a database server. Service may be provided by multiple servers, as is most often the case within a large enterprise.

• Cache is a repository of recently accessed objects (files, graphics) that is physically closer to the client than the server from which it originated. Proxy server sits in between clients and servers and can play many mitigation roles.

Advantages:

1. Simple to implement.

2. Provides good security.

3. All files are stored in a central location.

Disadvantages:

1. Single point of failure.

2. A specialist network operating system is needed.

Peer-to-Peer System

• All processes (objects) play similar role. Do not require a server process. Processes (objects) interact without particular distinction between clients and servers.

• The pattern of communication depends on the particular application. Fig. 1.6.3 shows the peer-to-peer model. (See Fig. 1.6.3 on next page.)

• Processing and communication loads for access to objects are distributed across many computers and access links. This is the most general and flexible model.

• A group of computers connected together to combine their computing and 'processing abilities to search the Internet or solve very complex problems

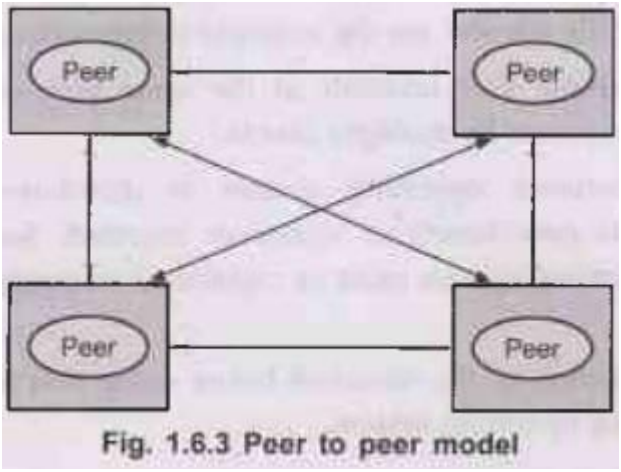


Fig. 1.6.3 Peer to peer model

• Problems with peer-to-peer : High complexity due to,

1. Cleverly place individual objects

2. Retrieve the objects

3. Maintain potentially large number of replicas.

Distinguish between Client Server and Peer-to-Peer Model

Sr. No.	Client–server model	Peer-to-peer model
1.	The client-server model firmly distinguishes the roles of the client and server. Under this model, the client requests services that are provided by the server.	The peer-to-peer model doesn't have such strict roles. In fact, all nodes in the system are considered peers and thus may act as either clients or servers or both.
2.	Single point of failure.	No single point of failure.
3.	Need for dedicated application and database servers.	No need for dedicated application and database servers.
4.	Resources may be removed at any time.	Resources are not removed from the network until they are no longer being requested.
5.	Storage and bandwidth must be provided by the host.	Storage and bandwidth are distributed and provided by the entire network.
6.	Provides good security.	Provides Poor security.

Network Operating System

• Network operating system is designed by using existing operating system.

• In network operating system, user can login to the remote resources and then access that resource. There are multiple machines are connected by

communication link. User can transfer data from their own machine to remote machine and vice

versa.

• Telnet and remote file transfer are the example of network operating system.

• NOS supports multiple user accounts at the same time and enables concurrent access to shared resources by multiple clients.

•A function of network operating system is distributed over the network computers. It adds new functions whenever required. Server machine support multitasking. Operating system must be capable of executing multiple tasks at the same time.

• User knows the location of the resources before using that resource. Each machine has its own network operating system.

Real Time System

• Time constraints is the key parameter is real time systems. It controls autonomous system such as robots, satellites, air traffic control and hydroelectric dams.

• When user gives an input to the system, it must process within the time limit and result is sent back. Real time system fails if it does not give result within the time limits.

• Real time systems are of two types: **Hard real time and soft real time**.

• Critical task is completed within the time limit in hard real time system. All the delay in the system is fixed and time bounded. Existing general purpose operating system does not support the hard real time system functions. Real time task cannot keep waiting for longer time without allocating kernel.

• Soft real time system is less restrictive type. Soft real time cannot guarantee that it will be able to meet deadline under all condition. Example of soft real time system is digital telephone and digital audio.

• Real time operating system uses priority scheduling algorithm to meet the response requirement of a real time application. General real time applications with their example are listed below:

1. Transportation: Air traffic control and traffic light system

2. Communication: Digital telephone

3. Process control: Petroleum and paper mill

4. Detection: Burglar system and radar system

5. Flight simulation: Auto pilot shuttle mission simulator.

• Memory management in real time system is less demanding than in other type of multiprogramming operating systems. Time critical device management is one of non the main characteristics of the real time systems.

Operating System Structures

AU: May-17, 19, 22

• Modern operating system is complex in nature. Here we study different structure motto of the operating systems.

Simple Structure

• Microsoft-Disk Operating System (MS-DOS) is example of simple structure of operating system. Most of the commercial systems do not have well defined structure. Initially DOS is small and simple in size. When new versions are introduced, size goes on increasing.

• There is no CPU Execution Mode (user and kernel), and so errors in applications can cause the whole system to crash.

• MS-DOS is layered structure operating system. It consists of following layers:

1.Application program layer

2. System program layer for resident

3. Device driver layer

4. ROM BIOS device driver layer.

• In DOS, application program directly interact with BIOS device driver. If user makes any changes in the BIOS device driver, it creates the problem and affects the whole system. Memory size is also limited so after use memory must be made free for other users.

• Another example of this type is UNIX operating system. Initially it provides limited hardware functionality. UNIX is divided into two parts : Kernel and system programs.

• Kernel provides system calls for CPU scheduling, I/O management, file management and memory management. System call uses application program interface in UNIX. API defines user interface by using set of system programs. Kernel supports an API and user interface.

• To support more advanced version of hardware, new UNIX operating system was developed. It is consists of different modules for controlling the hardware.

Module will communicate with each other for proper operation. Designers are free to design operating systems which support all new versions of hardware and user requirements.

Layered Approach

• Second type of operating system design is layered approach. Operating system is divided into number of layers. Each layer boundary is properly defined. Bottom layer is called layer 0 and top most layer is called layer N. Layer N provides user interface.

• A function of layer is also fixed. Fig. 1.9.1 shows the operating system layer. Each layer consists of data structure and a set of routines. Layer provides services to upper and lower layers.

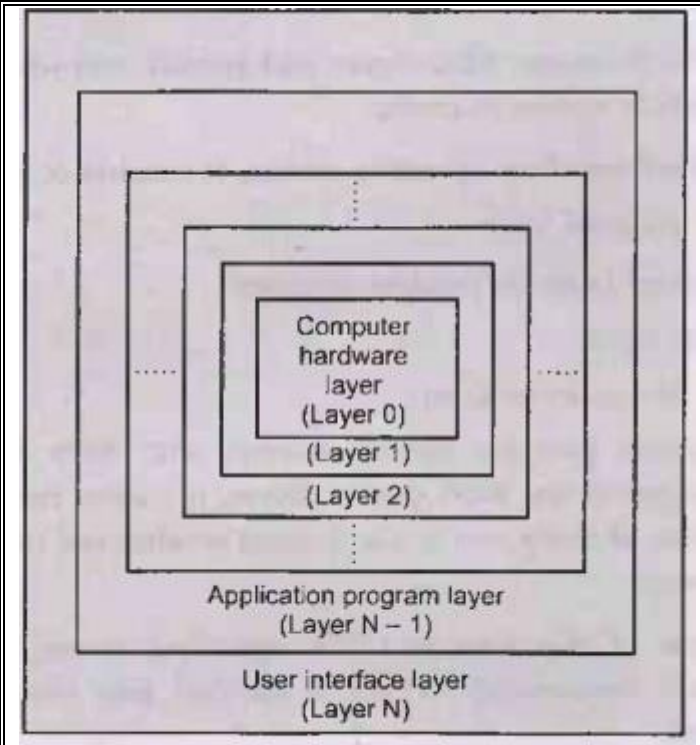


Fig. 1.9.1 Operating system layers

• Modularity is the advantage of the layered system. Number of layers are selected properly and each layer uses the services and functions of lower level layer (Below layer of the current layer). This type of design simplifies the debugging and system verification.

• First layer contains only basic hardware to implement function. So only first layer is debugged for checking the whole system. If error is not found then the system will works properly. If error is encounter while debugging second layer, then error is related to the second layer only. In this operating system is designed and implemented. It simplifies the design task of the operating system.

• Care should be taken while designing the layers. Which functions are added to which layer must be designing properly? Layer boundaries must be defined properly. Some of the function includes only in the lower layer. Device driver, memory management and input/output operation function must be included in the lower layer.

• Secondary storage is required for all operation. When CPU changes the process as per scheduling method, currently executing process is stored on the secondary storage. Disk space is required for this purpose. So CPU scheduling is includes in the above layer of the secondary storage layer.

• Each layer uses system calls for performing their operation. Layer adds overhead to the system call.

Monolithic Structure

• Traditional UNIX operating system uses monolithic kernel architecture. The entire operating system runs as a single program in kernel mode. Program contains operating system core function and device drivers.

• Most of the operation performed by kernel is via system call. Required system calls are made within programs and a checked copy of the request is passed through a system call. Fig. 1.9.2 shows monolithic kernel.

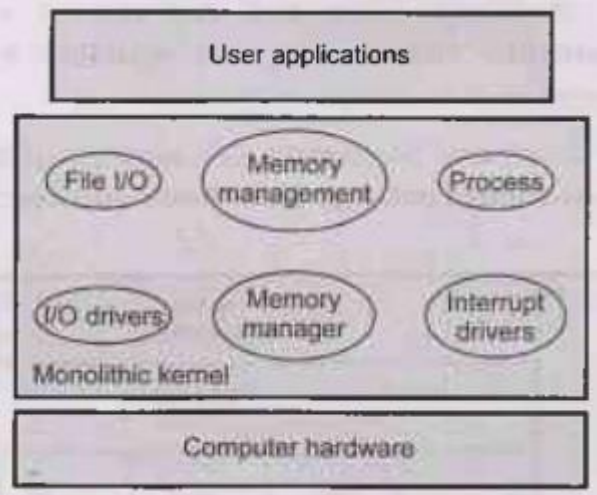


Fig. 1.9.2 Monolithic kernel

• LINUX operating system and FreeBSD uses modern monolithic kernel architecture. It loads the modules at run time. There is easy access of kernel function as required and minimize the code running in kernel space.

• Monolithic kernel used in Windows 95, Windows 98, Linux and FreeBSD etc.

Advantages:

1. Simple to design and implement.
2. Simplicity provides speed on simple hardware.
3. It can be expanded using module system.
4. Time tested and design well known.

Disadvantages :

1.Runtime loading and unloading is not possible because of module system.
2. If code base size increases, maintain is difficult.
3. Fault tolerance is low.

Microkernel

• Microkernel provides minimal services like defining memory address space, IPC and process management. It is small operating core. Hardware resource management is implemented whenever process is executing.

• The function of microkernel is to provide a communication facility between the client programs. It also provides facility to various services which are running in user space. Message passing method is for communication two processes.

• Microkernel runs in kernel mode and rest run in normal user processes. Microkernel also provides more security and reliability. Most of the services are running as user rather than kernel processes.

• By running device driver and file system as a separate user process, a error in one can crash only single component. Fig. 1.9.3 shows microkernel.

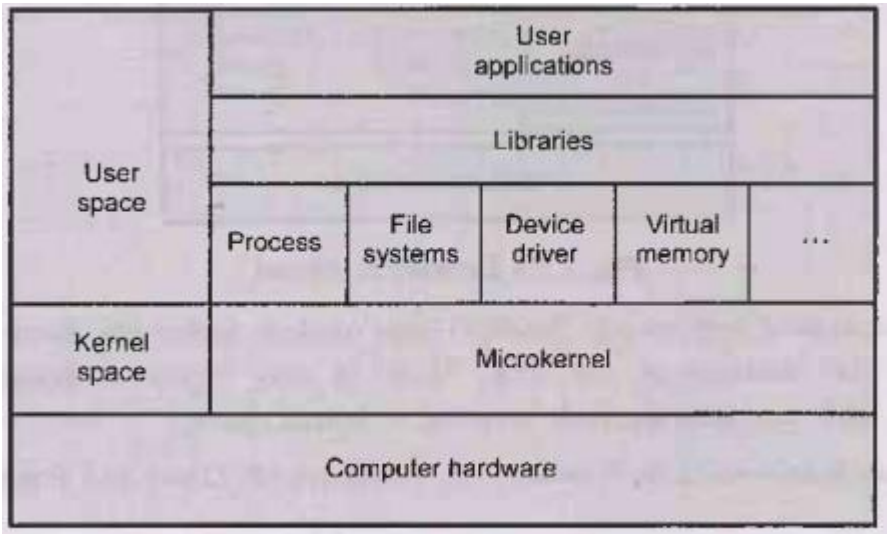


Fig. 1.9.3 Microkernel

• Mach operating system uses microkernel architecture. Microkernel in Windows NT operating system provides portability and modularity. The kernel is surrounded by a number of compact subsystems so that the task of implementing Windows NT on a variety of platform is easy.

• Microkernel architecture assigns only a few essential functions to the kernel, including address space, IPC and basic scheduling. QNX is a real time operating system that is also based upon the microkernel design.

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

Advantages of microkernel:

1. Microkernel allows the addition of new services.

2. Microkernel architecture design provides a uniform interface on requests made by a process.

3. Microkernel architecture supports object oriented operating system.

4. Modular design helps to enhance reliability.

5. Microkernel lends itself to distributed system support.

6. Microkernel architecture support flexibility. User can add or subtract services according to the requirement.

Virtual Machine

• In a pure virtual machine architecture the operating system gives each process the illusion that it is the only process on the machine. The user writes an application as if only its code were running on the system.

• Each user interacts with the computer by typing commands to the virtual machine on a virtual system console and receiving results back from the machine as soon as they are computed.

• Each user directs the virtual machine to perform different commands. These commands are then executed on the physical machine in a multiprogramming environments.

• Virtualization is an abstraction layer that decouples the physical hardware from the operating system to deliver greater IT resource utilization and flexibility.

• It allows multiple virtual machines, with heterogeneous operating systems to run in isolation, side-by-side on the same physical machine. Fig. 1.9.4 shows virtual machine.

• Each virtual machine has its own set of virtual hardware (e.g., RAM, wol CPU, NIC, etc.) upon which an o operating system and applications The operating system creates the illusion of multiple processes, each executing on its own processor with its own (virtual) memory.

• The main components of virtual machine are the control program, conversational monitor system,remote spooling communication interactive problem control and system.

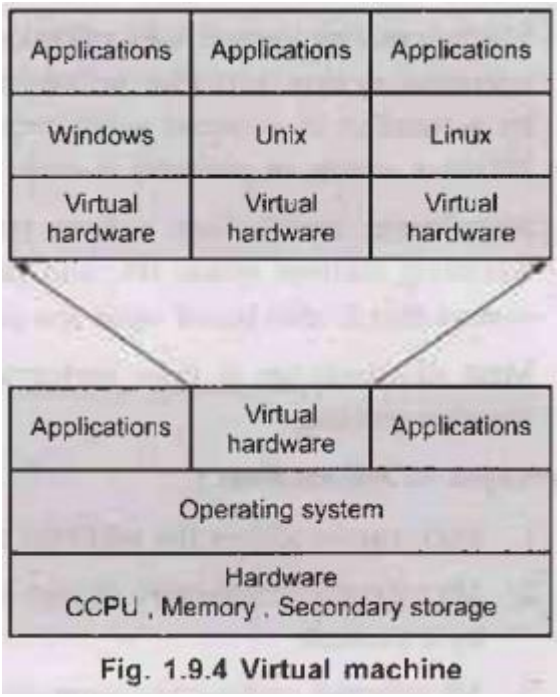


Fig. 1.9.4 Virtual machine

• The control program creates the environments in which virtual machines can executes. It also manages the real machines underlying the virtual machine environment.

• The java virtual machine is one of the most widely used virtual machines.

• Virtual machines tend to be less efficient than real machines because they access the hardware indirectly.

Benefits

1. There is no overlap amongst memory as each virtual memory has its own memory space.

2. Virtual machines are completely isolated from the host machine and other virtual machines.

3. Data does not leak across virtual machines.

Virtualization

•Virtualization means running multiple machines on a single hardware. The "Real" hardware invisible to operating system. OS only sees an abstracted out picture. Only Virtual Machine Monitor (VMM) talks to hardware.

• It is "a technique for hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources. This includes making a single physical resource appear to function as multiple logical resources; or it can include making multiple physical resources appear as a single logical resource."

• It is divided into two main categories :

1. Platform virtualization involves the simulation of virtual machines.

2. Resource virtualization involves the simulation of combined, fragmented, or simplified resources.

Comparison between Monolithic Kernel and Microkernel

Sr. No.	Monolithic Kernel	Microkernel
1.	Kernel size is large.	Kernel size is small.
2.	OS is complex to design.	OS is easy to design, implement and install.
3.	Request may be serviced faster.	Request may be serviced slower than monolithic Kernel.
4.	All the operating system services are included in the Kernel.	Kernel provides only IPC and low level device management services.
5.	No message passing and no context switching are required while the Kernel is performing the job.	Microkernel requires message passing and context switches.

Operating System Services

• Operating system provides different types of services to different users. It provides services to program like load of data into memory, allocating disk for storage, files or directory for open or read etc.

• Services will change according to the operating system. May be the two different operating system provides same type of services with different names. OS makes programmer job easy by providing different services.

Following are the list of services provided by operating systems :

1. Program execution
2. Input-output operation
3. Error detection
4. File and directory operation
5. Communication
6. Graphical user interface.

1. Program execution: Before executing the program, it is loaded into the memory by operating system. Once program loads into memory, its start

execution. Program finishes its execution with error or without error. It is up to the user for next operation.

2. Input- output operation: Program is combination of input and output statement. While executing the program, it requires I/O device. OS provides the I/O devices to the program.

3. Error detection: Error is related to the memory, CPU, I/O device and in the user program. Memory is full, stack overflow, file not found, directory not exist, printer is not ready, attempt to access illegal memory are the example of error detection.

4. File and directory operation: User wants to read and writes the file and directory. User wants to search the file/directory, rename file, and modify the file etc. user also create the file or directory. All these operation is performed by user by using help of operating system.

5. Communication: Communication may be inter-process communication and any other type of communication. Data transfer is one type of communication. Communication is in between two process of same machine or two process of different machine. Pipe, shared memory, socket and message passing are the different methods of communication.

6. Graphical user interface: User interacts with operating system by using user interface. All operating system provides user interface. Command line interface and batch interface are two types of user interface used in the operating system. In command line interface, user enters the text command for performing operation. Batch interface uses files. A file contains the command for various operations. When file executes, command output is displayed on the screen.

• All above services provided by operating system is only for single user. If suppose there are multiple user in the system, then operating system provides some Po additional services. These services are resource allocation, accounting, protection of data and security.

• Different types of resources are managed by the operating system. Multiple users require different resource for their execution. One type of resource may be required by two different users. So resource allocation is necessary.

• Accounting means keeping information of resources and users. Which types of resources are allocated to which user and whether particular user has permission sostro for using this type of resources.

User and Operating System Interface

• Various methods are used to interface with the operating system. They are command-line interface, Graphical User Interface and Touch-Screen Interface.

Command Interpreters

• A command interpreter is a system software that understands and executes commands that are entered interactively by a human or from another program. Command interpreter is an important part of any operating system. It provides an interface between the user and the computer.

• Command interpreter is also called command-line interface.

• Most operating systems, including Linux, UNIX and Windows, treat the command interpreter as a special program that is running when a process is initiated or when a user first logs on.

• A command interpreter is often also called a command shell or simply a shell. A command shell also usually provides a set of programs or utilities which allows ad users to perform file management functions.

• The shell utility provides a user interface to many system services. For example, user requests such as listing file names in a directory, running a program, logging out, may all be handled by the shell. The shell may invoke other utilities to actually do the work; for example, directory file listing is sometimes a utility program itself.

• The command interpreter itself contains the code to execute the command. For example, a command to delete a file may cause the command interpreter to jump to a section of its code that sets up the parameters and makes the appropriate system call.

• User may choose among several different shells, including the C shell, Bourne-Again shell, Korn shell, and others. Third-party shells and free user-written shells are also available. A very popular shell on most commercial variants of Unix is the Korn shell.

• Command line is text based interface of the Linux system. Terminal is another name in Linux system. User can communicate with system is by using command. You can combine more than one command and get output.

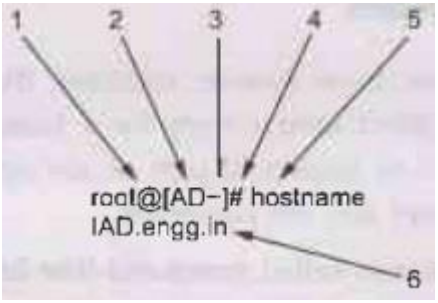
• Commands are executed at the prompt. Command line interface gives the user more control and options. There are many varieties of Linux, but almost all of them use similar commands that can be entered from a command-line interface terminal.

• The syntax of command line is as follows:

#command options

• Example :

[root@IAD~]# hostname



Here we will explain all of above arrows one by one.

1. Logged in user name, as now currently we are logged in with root user.

2. After the @ sign we have hostname of our Linux.

3. This is current working directory, "~" it is user home directory sign in Linux.

4. "#" hash sign or number sign means we are logged in with super user such as "root" it will be changed in case if we are logged in with normal Linux user e.g "$"

5. Just for example we give "hostname" command to shell, hostname is used to know the fully qualified hostname of your Linux machine.

6. This is output of our previously entered command.

Graphical User Interface

• GUI provides a graphical interface for the user to interact with the computer. It is the common user interface that includes graphical representation like buttons and Marie icons and communication can be performed by interacting with these icons rather than the usual text-based or command-based communication.

• The GUI actually translates user language, which comprises simple one-line commands, single click and double clicks to machine language or assembly language. The machine understands machine language and hence the machine responds to the task initiated, which is translated to use language and communicated to the user via GUI.

• The first commercially available GUI was called "PARC," and was developed by Xerox for its 8010 Information System, released in 1981.

• The actions in a GUI are usually performed through direct manipulation of graphical elements like buttons and icons. Communication can be performed by interacting with these icons rather than the usual text-based or command-based communication.

• Graphical user interface applications are self-descriptive, feedback is typically immediate and visual cues encourage and steer discoverability

Touch-Screen Interface

• The first touch screen phone was launched in 1992 by IBM and was named IBM Simon. iPhone was the latest invention of the touch screen phone which gets its popularity in no time.

• Whether it is touch screen microwave, touch screen car LCD, touch screen monitor, touch screen ATMs and others. The interface of touch screen devices is made in such a way that it is easy for people to use them.

• Some touch screen interface is fully controlled by the touch screen while others are partially controlled by the screen and another function is controlled by the physical keyboard.

• A touch user interface is a type of interaction between computer-based device. This interaction is by doing a physical touch on the screen and the computer responds to this touch interaction.

ystem Calls

AU May-17, 22

• System calls provide the interface between a running program and the operating system. Any single CPU computer can execute only one instruction at a time. If a process is running a user program in user mode and needs a system service, such UTD as reading a data from a file, it has to execute a instruction to transfer control to the operating system.

• Operating system provides services and system call provides interface to these services. System call is written in language C and C++ as routines. System calls are performed in a series of steps.

• System call is a technique by which a program executing in user mode can request the kernel's service.

•An application programmer interface is a function definition that specifies how to

obtain a given service.

• Fig. 1.12.1 shows the working of system call.

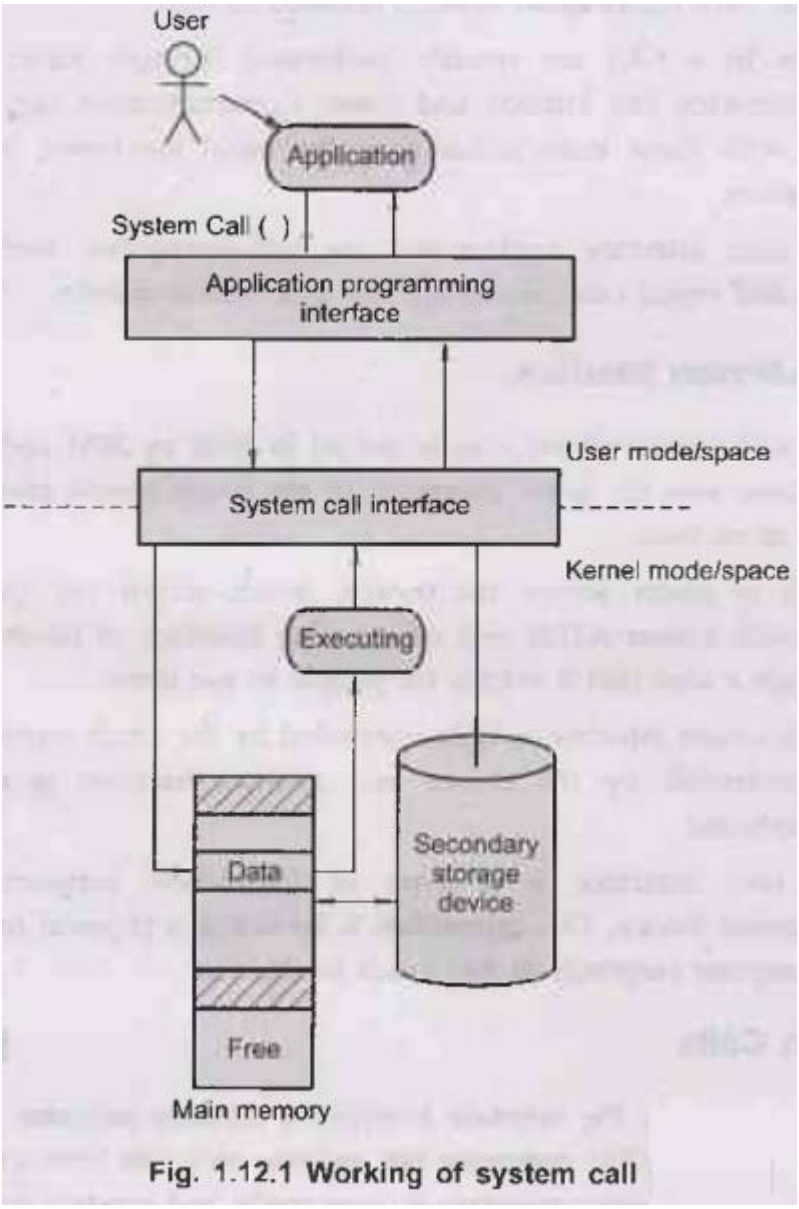


Fig. 1.12.1 Working of system call

• When application program calls the stub, trap instruction is executed and CPU switches to supervisor mode. Each system call contains its identification number.

• OS maintains the table of system call number. Operating system executes the system call using that number.

• When the function completes, it switches the processor to user mode and then returns control to the user process.

A system call is an explicit request to the kernel mode via a software interrupt. When user mode process invokes a system call, the CPU switches to kernel mode and starts the execution of the kernel function.

• Making a system call is like making a special kind of procedure call. Only system call enters the kernel and procedure call does not enter into the kernel.

• Kernel implements many different types of system calls. The user mode process must pass a parameter called the system call number to identify the required system call. All system calls return an integer value. In the kernel, positive or 0 values denote a successful termination of the system call and negative values denote an error condition.

• An API does not necessarily correspond to a specific system calls.

1. API could offer its services directly in user mode.

2. Single API function could make several system calls.

3. Several API functions could make the same system call.

• API supports a set of functions for application programmer. It includes passing parameter to each function and return values. API used by application programmer are as follows:

1. Windows system: win32 API

2. POSIX system : POSIX API

3. Virtual machine: Java API.

• User function uses trap instruction for using kernel services. Application programmer uses ordinary procedure call. Operating system provides a library of user functions with name corresponding to each actual system call. Each of these stub function contains a trap to the operating system function.

• Trap is also called system call interface.

• Three general methods are used to pass parameters to the operating system.

a. Pass parameters in registers

b. Registers pass starting addresses of blocks of parameters

C. Parameters can be placed, or pushed, onto the stack by the program, and popped off the stack by the OS

• Fig. 1.12.2 shows passing of parameters as a table

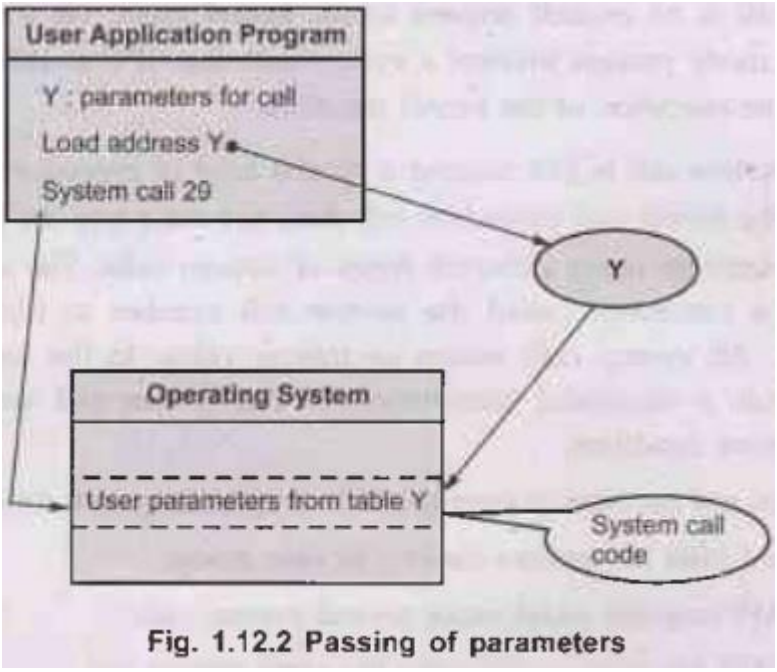


Fig. 1.12.2 Passing of parameters

Classification of System Call

• System call is divided into following types:

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

1. File management

• File management system calls are create file, delete file, open file, close file, read file, write file, get and set file attribute.

• User can create a file using create() system call. File name with attributes are required for creating and deleting a file through system call. After creating a file, user can performs various operations on the file.

• Read, write, reposition are the operation performed on the file. File is closed after finished using. Same type of operation is performed on directory.

• Every file has file attributes. File attributes includes name of file, type of file, accounting information etc. To perform any operation on the file, set file attribute and get file attribute executed to check the attributes.

2. Process management

• System calls for process management are create, terminate, load, execute, abort, set and get process attributes. Other system call for process management is wait for time, wait event, allocate and free memory.

• In some situation user want to terminate the currently running process abnormally then system call used. Other reasons for abnormal process termination are error message generated, memory dump, error trap.

• Operating system provides debugging facility to determine the problem of dump. Dump is written to secondary storage disk.

• Debugger is a one type of system program. It provides facility for finding and correcting bug.

3. Interprocess communication

• Pipe, socket, message passing and shared memory are used for interprocess communication. Send message, receive message, create and delete connection, attach remote device are the system calls used in interprocess communication. •Shared memory: A process uses memory for communication. One process will create a memory portion which other processes can access. A shared segment can be attached multiple times by the same process. Shared memory is the fastest form of IPC because data does not need to be copied between processes.

• A socket is a bidirectional communication device that can be used to communicate with another process on the same machine or with a process running on other machine.

• Message passing: Two processes communicate with each other by passing messages. Message passing is direct and indirect communication. Indirect communication uses mailbox for sending receiving message from other process.

4. I/O device management

• System calls for device management are request () device, release ( ) device, get of nail and set device attributes, read, write etc.

• Process needs several resources in its life time. When process request for resources, request is granted if it is free otherwise request is rejected. Once request is granted, control is transfer to the process.

5. Information processing and maintenance

• System calls for this category is set time and date, get time and date, get system data, get system data, get and set process, file and device.

• Most of the operating system provides system call for set and get the time and date.

• This type of system call is used for transferring information from user to operating system and vice versa.

System Programs

• Modern operating system consists of a collection of system programs. System program that provides an application programming environment on top of the hardware. Some of them are simply user interfaces to system calls. They can be divided into these categories:

1. File management
2. Status information
3. File modification
4. Programming language support
5. Program loading and execution
6. Communications

• File management programs are used to create, delete, copy, rename, list, dump on files and directory. Status information system programs covers the date, time, disk in space, available memory and users. All this information is formatted and to displayed on output device or printed.

• Text editors are used for file modification. In this, new file is created and content of file is modified. Programming language support includes the compilers, assemblers, interpreters for common programming language like C, C++, Java, Visual Basic.

• For program loading and execution, it is loaded into memory then program is executed by processor. Operating system provides different types loaders and linkers to complete execution operation.

• Debugging facility is provided by the operating system with the help of application program. It is used for checking errors.

• Communication between two devices are performed by creating temporary connection. Communication is in between process, users and other I/O devices. File transfer, remote login, electronic mail, browsing web, downloading multimedia data are the example of communication.

Design and Implementation

Design:

• Before designing the system, first define goals and specification of the system. Goals and specification will give the idea about the system and it also understands the requirement of the user. At the higher level, system design is dominated by the choice of hardware and system type.

• System type may be batch system, multiprogramming, multitasking, real time and distributed system. Hardware must support all these type of operating system.

• Once the goal and specification is clearly understood, then a requirement is related to user and system. These requirements are called user goals and system goals.

• User goals are as follows:

1. Simple to use and understand.

2. Provides reliability.

3. Provides security.

4. Fast speed.

• System goals are as follows:

1. Easy to design

2. Implementation and maintenance is also simple and easy

3. Free from error

4. Efficient to use.

• Operating system requirements is not fixed. It will change according to the user requirements. Single user operating system is required and multiple user operating systems is also required.

Implementation

• At first, operating systems were written in assembly, but now a days C/C++ is the language commonly used.

• Small blocks of assembly code are still needed, especially related to some low level I/O functions in device drivers, turning interrupt on and off and the Test and Set instruction for synchronization facilities.

• Using higher level language allow code to be written faster, is easier to read, and can be debugged easier. It also makes the OS much easier to port to different hardware platforms.

Two Marks Questions with Answers

Q.1 Define operating systems.

Ans. : An operating system is a program that manages the computer hardware.

Q.2 What are the goals of OS?

Ans. : Operating system goals :

a. Execute user programs and make solving user problems easier.

b. Make the computer system convenient to use.

Q.3 List down the functions of operating systems.

Ans. : Functions of operating systems are convenience, efficiency and ability to evolve.

Q.4 What are the objective of operating systems ? AU: [CSE]: May-17, Dec.-17

Ans. : The objective of operating systems are efficient use, user convenience, ability to evolve.

Q.5 Compare and contrast DMA and Cache memory.

Ans. :• DMA technology provides special channels for CPU and I/O devices to exchange I/O data, and the memory is used for buffering the I/O data. When the CPU wants to handle I/O data, it triggers the DMA write operations that transfer the I/O data from I/O devices to the memory.

• Caches are "automatically" managed in that the hardware, when the requested memory contents are not in the cache, fetches that data from main memory.

Q.6 How does an interrupt differ from trap?AU: [EIE]: May-18

Ans. : Trap is a software-generated interrupt caused either by an error or by a specific request from a user program that an operating-system service be performed. A trap usually results in a switch to kernel mode. Interrupt signals can cause a program to suspend itself temporarily to service the interrupt. An interrupt is a hardware-generated signal that changes the flow within the system.

Q.7 What are the three main purposes of an operating system ?

Ans. :

• To provide an environment for a computer user to execute programs on computer hardware in a convenient and efficient manner.

• To allocate the separate resources of the computer as needed to solve the problem given. The allocation process should be as fair and efficient as possible.

• As a control program it serves two major functions: 1) Supervision of the execution of user programs to prevent errors and improper use of the computer and 2) Management of the operation and control of I/O devices.

Q.8 Explain what is batch processing.

Ans. : Here jobs with similar requirements are batched together and run through the computer as a group. Thus a batch operating system reads a stream of separate jobs, each with its own control cards that predefine what the job does,

feed the batches one after another and send the output of each job to the appropriate destination.

Q.9 What is spooling ?

Ans. : The use of secondary memory as buffer storage to reduce processing delays when transferring data between peripheral equipment and the processors of a computer.

Q.10 Define the degree of multiprogramming.

Ans. : Degree of multiprogramming is the number of processes in the memory.

Q.11 What is the main advantage of multiprogramming ?

Ans. : Multiprogramming makes efficient use of the CPU by overlapping the demands for the CPU and its I/O devices from various users. It attempts to increase CPU utilization by always having something for the CPU to execute.

Q.12 State the function of Printer Daemon.

Ans. : The Printer Daemon provides printer services for local and remote users. It manages the printer spool area and the print queues. Printer Daemon is started at boot time from a startup script. It is generally included in the startup of Linux and BSD systems by default so you might not need to add it to your startup script.

Q.13 What do you mean by multiprogramming?

Ans. : Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

Q.14 List the disadvantages of multiprogrammed batched systems.

Ans. :Disadvantages

1. Users cannot interact with their jobs, while executing.

2. A programmer cannot modify a program as it executes to study its behavior

Q.15. Do time sharing differs from multiprogramming? If so, how?

Ans. : Time sharing is the sharing of resources among several processes at the same time. Multiprogramming is the allocation of more than one process on a computer system and its resources. Time sharing minimizes the response time and multiprogramming maximizes the processor use. Time sharing systems use the concept of multiprogramming to share the CPU time between multiple users at the same time.

Q.16 Write the difference between batch system and time sharing systems.

Ans. : A batch system executes jobs, whereas a time-shared system has user programs, or tasks.

• Batch systems are inconvenient for users because users cannot interact with their jobs to fix problems. User interacts with system in time sharing system.

Q.17 Describe the differences between symmetric and asymmetric multiprocessing.

Ans. : Symmetric multiprocessing treats all processors as equals and I/O can be processed on any CPU. Asymmetric multiprocessing has one master CPU and the remainder CPUs are slaves. The master distributes tasks among the slaves and I/O is usually done by the master only.

Q.18 What is tightly coupled system ?

Ans. : Processors share memory and a clock; communication usually takes place through the shared memory.

Q.19 What is graceful degradation ?

Ans. : In multiprocessor systems, failure of one processor will not halt the system, but only slow it down. If there are 15 processors and if one fails the remaining 14 processors pick up the work of the failed processor. This ability to continue providing service is proportional to the surviving hardware is called graceful degradation.

Q.20 What are the disadvantages of multiprocessor systems ?

Ans. : Multiprocessor systems are more complex in both hardware and software. Additional CPU cycles are required to manage the cooperation, so per-CPU efficiency goes down.

Q.21 Distinguish between tightly coupled system and loosely coupled system.

Ans. :

Sr. No.	Tightly coupled system	Loosely coupled system
1.	Also known as parallel system.	Also known as distributed system.
2.	Shares computer bus, clock and sometimes memory.	Do not share memory or a clock.
3.	More than one processor in close communication.	Single processor system.

Q.22 What is difference between networked O.S. and distributed O.S. ?

Ans. : Difference between networked O.S. and distributed O.S. is an follows:

Sr. No.	Disributed O.S.	Networked O.S.
1.	Degree of transparency is high.	Degree of transparency is low.
2.	Same O.S. on all the nodes.	May be different or same O.S. on all the nodes.
3.	Shared memory is used for communication.	Files are used for communication.
4.	Scalability is less.	Scalability is high.
5.	Resource management is done globally or central.	Resource management is done by per node.

Q.23 Mention the advantages in the design of distributed operating systems. Ans. : Advantages are as follows:

1. Resource sharing 2. Higher reliability 3. Better price performance ratio

4. Shorter response times and higher throughput.

Q.24 What are the advantages of peer-to-peer systems over client server systems?Ans. : Peer-to-peer system is more reliable as central dependency is eliminated. All the resources and contents are shared by all the peers, unlike server-client architecture where server shares all the contents and resources.

Q.25 Define real time system.

Ans. : Real time system is one that must react to inputs and responds to them quickly. A real time system has well defined, fixed time constraints.

Q.26 What is meant by "hard real systems and soft real systems"?

Ans. Hard real systems guarantee that critical tasks complete on time. In soft real system a critical task get priority over other tasks and remains that priority until it completes.

Q.27 What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment ?

Ans. : The main difficulty is keeping the operating system within the fixed time constraints of a real-time system. If the system does not complete a task in a certain time frame, it may cause a breakdown of the entire system it is running. Therefore when writing an operating system for a real-time system, the writer must be sure that his scheduling schemes don't allow response time to exceed the time constraint.

Q.28 What are the main advantages of layered approach ?

Ans. : a. Each layer is implemented using those operations provided by lower-level layers.

b. A layer does not need to know how the low-level operations are implemented; it needs to know what these operations o.

c. Each layer hides the existence of data structures, operations and hardware from higher-level layer.

Q.29 Mention the purpose of system calls. AU: May-18

Ans. : System calls allow user-level processes to request services of the operating system.

Q.30 What is the purpose of system programs ? AU: May-16

Ans.: System program that provides an application programming environment on top of the hardware.

Q.31 What are the five major categories of system calls ? AU CSE/IT: May-11

Ans. : Five main categories of system calls are File management, IPC, process management, I/O devices management, information management.

Q.32 What is the function of system programs? Write the name of the categories in which the system programs can be divided.

Ans. : programs provide a convenient environment for program development and execution. System programs are divided into these categories: File management, status management, file modification, programming language support, program loading and execution, communications.

Q.33 What is meant by a system call ?

Ans. : System calls provide the interface between a process and the OS.

Q.34 What does the CPU do when there are no user programs to run?

Ans. : The CPU will always do processing. Even though there are no application programs running, the OS is still running and the CPU will still have to process many system processes during the operation of computer.

Q.35 List general methods used to pass parameters in system call.

Ans. : Three general methods are used to pass parameters between a running programand the operating system.

a. Pass parameters in registers.

b. Store the parameters in a table in memory and the table address is passed as a parameter in a register.

c. Push (store) the parameters onto the stack by the program and pop off the stack by operating system.

Q.36 Why APIs need to be used rather than system calls?

Ans. : An application programmer designing a program using an API can expect his program to compile and run on any system that supports the same API. Actual system calls can often be more detailed and difficult to work with the API available to an application programmer.

Q.37 List the advantages and disadvantages of writing an operating system in high level languages such as C. AU: May-18

Ans. : Advantages:

• The code can be written faster and more compact.

• Easier to understand and debug

• Recompilation is simple.

Disadvantages :

• There could be a performance overhead introduced by the compiler and runtime system used for the high-level language.

• Certain operations and instructions that are available at the machine-level might not be accessible from the language level, thereby limiting some of the functionality available to the programmer.

Q.38 What is dual mode operation and what is the need of it? AU [CSE] May-19

Ans.:

• Two modes are provided by the hardware - user and kernel mode. Mode bit is added to computer hardware to indicate the current mode. The OS is loaded and then user process is started in user mode. When an interrupt occurs, hardware switches to kernel mode.

• In order to ensure the proper execution of the OS, user must be able to distinguish between the execution of OS code and user defined code.

Q.39 List out some system calls required to control the communication system. AU [CSE]: May-19

Ans. : System calls for communication system: Open connection and close connection, read message and write message, get hosted and get processed.

University Question

1. Discuss about direct memory access.

2. Explain cache memory and its mapping. **Marks 6**

3. Give reason why caches are useful. What problems do they solve? What problems do they cause?

If a cache can be made as large as the device for which it is caching why not make it that large and eliminated the device? **Marks 8**

4. Illustrate the flow of control with and without interrupts. **Marks 7**

5. Brief about the various types of memories in memory hierarchy.

6.Distinguish multiprogramming and time-sharing environment. In a multiprogramming and od time-sharing environment, several users share the system simultaneously. This situation can result in various security problems.

i) What are two such problems?

ii) Can we ensure the same degree of security in a time-shared machine as in a dedicated machine? Explain your answer.

Discuss the pros and cons of single processor system, multi core system and clustered system.

1. Explain the various structures of an operating system.

2. State the operating system structure. Describe the operating system operations in detail. Justify the reason why the lack of a hardware-supported dual mode can cause serious shortcoming in an operating system?

3. Explain in detail the different architectures of OS starting from simple structure, layered structure, micro kernels, modules and hybrid system with suitable example OS structures, including google's Android.

1. Discribe system calls and system programs in detail with neat sketch.

2. List and explain five types of system calls.

APEC