OPERATING SYSTEM

Explain Booting the system and Bootstrap program in operating system.

The procedure of starting a computer by loading the kernel is known as booting the system.

When a user first turns on or booted the computer, it needs some initial program to run. This initial program is known as Bootstrap Program. It is stored in read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM). Bootstrap program locates the kernel and loads it into main memory and starts its execution.

Describe Main memory and Secondary memory storage in brief.

Main memory is also called random access memory (RAM). CPU can access Main memory directly. Data access from main memory is much faster than Secondary memory. It is implemented in a semiconductor technology, called dynamic random-access memory (DRAM).

Main memory is usually too small to store all needed programs. It is a volatile storage device that loses its contents when power is turned off. Secondary memory can stores large amount of data and programs permanently. Magnetic disk is the most common secondary storage device. If a user wants to execute any program it should come from secondary memory to main memory because CPU can access main memory directly.

What are the advantages of multiprocessor system?

Systems which have more than one processor are called multiprocessor system. These systems are also known as parallel systems or tightly coupled systems.

Multiprocessor systems have the following advantages.

- Increased Throughput: Multiprocessor systems have better performance than single processor systems. It has shorter response time and higher throughput. User gets more work in less time.

- Reduced Cost: Multiprocessor systems can cost less than equivalent multiple single processor systems. They can share resources such as memory, peripherals etc.

- Increased reliability: Multiprocessor systems have more than one processor, so if one processor fails, complete system will not stop. In these systems, functions are divided among the different processors.

Is it possible to have a deadlock involving only one process? Explain your answer.

Deadlock with one process is not possible. Here is the explanation.

A deadlock situation can arise if the following four conditions hold simultaneously in a system.

- Mutual Exclusion.

- Hold and Wait.

- No Preemption.

- Circular-wait.

It is not possible to have circular wait with only one process, thus failing a necessary condition for Circular wait. There is no second process to form a circle with the first one. So, it is not possible to have a deadlock involving only one process.

What is an operating system?

An operating system is a collection of software programs which control the allocation and usage of various hardware resources in the system. It is the first program to be loaded in the computer and it runs in the memory till the system is shut down.

Some of the popular Operating Systems are DOS, Windows, Ubuntu, Solaris etc.

What are its main functions?

The main functions of an OS are:

a. Process Management

b. Memory Management

c. Input/ Output Management

d. Storage/ File system management

What is a Kernel?

- Kernel is the part of OS which handles all details of sharing resources and device handling.

- It can be considered as the core of OS which manages the core features of an OS.

- Its purpose is to handle the communication between software and hardware

- Its services are used through system calls.

- A layer of software called shell wraps around the Kernel.

What are the main functions of a Kernel?

The main functions of a Kernel are:

- Process management

- Device management

- Memory management

- Interrupt handling

- I/O communication

- File system management

What are the different types of Kernel?

Kernels are basically of two types:

a. Monolithic Kernels - In this architecture of kernel, all the system services were packaged into a single system module which lead to poor maintainability and huge size of kernel.

b. Microkernels - They follow the modular approach of architecture. Maintainability became easier with this model as only the concerned module is to be altered and loaded for every function. This model also keeps a tab on the ever-growing code size of the kernel.

What are the disadvantages of Microkernels?

Following are the main disadvantages of Microkernels. Usually these disadvantages are situation based.

a. Larger running memory footprint

b. Performance loss due to the requirement of more software for interfacing.

c. Difficulty in fixing the messaging bugs.

d. Complicated process management.

What is a command interpreter?

It is a program that interprets the command input through keyboard or command batch file. It helps the user to interact with the OS and trigger the required system programs or execute some user application.

Command interpreter is also referred to as:

- Control card interpreter

- Command line interpreter

- Console command processor

- Shell

Explain Process.

A process is a program that is running and under execution. On batch systems, it is called as a "job" while on time sharing systems, it is called as a "task".

Explain the basic functions of process management.

Important functions of process management are:

- Creation and deletion of system processes.

- Creation and deletion of users.

- CPU scheduling.

- Process communication and synchronization.

What do you know about interrupt?

- Interrupt can be understood as a signal from a device causing context switch.

- To handle the interrupts, interrupt handlers or service routines are required.

- The address of each Interrupt service routine is provided in a list which is maintained in interrupt vector.

What is a daemon?

- Daemon - Disk and execution monitor, is a process that runs in the background without user’s interaction. They usually start at the booting time and terminate when the system is shut down.

How would you identify daemons in Unix?

- The name of daemons usually ends with 'd' at the end in Unix.

- For e.g. httpd, named, lpd.

What do you mean by a zombie process?

- These are dead processes which are not yet removed from the process table.

- It happens when the parent process has terminated while the child process is still running. This child process now stays as a zombie.

What do you know about a Pipe? When is it used?

- It is an IPC mechanism used for one-way communication between two processes which are related.

- A single process doesn't need to use pipe. It is used when two process wish to communicate one-way.

What is a named pipe?

- A traditional pipe is unnamed and can be used only for the communication of related process. If unrelated processes are required to communicate - named pipes are required.

- It is a pipe whose access point is a file available on the file system. When this file is opened for reading, a process is granted access to the reading end of the pipe. Similarly, when the file is opened for writing, the process is granted access to writing end of the pipe.

- A named pipe is also referred to as FIFO or named FIFO.

What are the various IPC mechanisms?

IPC - Inter Process Communication.

Various IPC mechanisms are:

a. Sockets

b. Pipes

c. Shared memory

d. Signals

e. Message Queues

What is a semaphore?

- A semaphore is a hardware or a software tag variable whose value indicates the status of a common resource.

- Its purpose is to lock the common resource being used. A process which needs the resource will check the semaphore to determine the status of the resource followed by the decision for proceeding.

- In multitasking operating systems, the activities are synchronized by using the semaphore techniques.

What kind of operations are possible on a semaphore?

Two kind of operations are possible on a semaphore - 'wait' and 'signal'.

What is context switching?

- Context is associated with each process encompassing all the information describing the current execution state of the process

- When the OS saves the context of program that is currently running and restores the context of the next ready to run process, it is called as context switching.

- It is important for multitasking OS.

Tell us something about Mutex.

- Mutex - ‘Mutual Exclusion Lock’ is a lock which protects access to shared data resource.

- Threads can create and initialize a mutex to be used later.

- Before entering a critical region, the mutex is locked. It is unlocked after exiting the critical region. If any thread tries to lock the mutex during this time, it can't do so.

What is a critical section?

It is a section of code which can be executed only by one process at a time.

What is synchronization? What are the different synchronization mechanisms?

Synchronization means controlling access to a resource that is available to two or more threads or process. Different synchronization mechanisms are:

- Mutex

- Semaphores

- Monitors

- Condition variables

- Critical regions

- Read/ Write locks

What is the basic difference between pre-emptive and non-pre-emptive scheduling.

Pre-emptive scheduling allows interruption of a process while it is executing and taking the CPU to another process while non-pre-emptive scheduling ensures that a process keeps the CPU under control until it has completed execution.

Is non-pre-emptive scheduling frequently used in a computer? Why?

No, it is rarely used for the reasons mentioned below:

- It cannot ensure that each user gets a share of CPU regularly.

- The idle time with this increase reducing the efficiency and overall performance of the system.

- It allows program to run indefinitely which means that other processes have to wait for very long.

Explain condition variable.

- These are synchronization objects which help threads wait for particular conditions to occur.

- Without condition variable, the thread has to continuously check the condition which is very costly on the resources.

- Condition variable allows the thread to sleep and wait for the condition variable to give it a signal.

What are read-write locks?

- Read - write locks provide simultaneous read access to many threads while the write access stays with one thread at a time. They are especially useful in protecting the data that is not frequently written but read simultaneously by many threads.

- They are slower than mutexes.

What is a deadlock?

- It is a condition where a group of two or more waiting for the resources currently in use by other processes of the same group.

- In this situation every process is waiting for an event to be triggered by another process of the group.

- Since no thread can free up the resource a deadlock occurs and the application hangs.

What are the necessary conditions for deadlock to occur?

a. At least one resource should be occupied in a non-sharable condition.

b. A process holding at least one resource is waiting for more resources currently in use by other processes.

c. It is not possible to pre-empt the resource.

d. There exists a circular wait for processes.

Name the functions constituting the OS's memory management.

- Memory allocation and de-allocation

- Integrity maintenance

- Swapping

- Virtual memory

Name the different types of memory?

a. Main memory also called primary memory or RAM

b. Secondary memory or backing storage

c. Cache

d. Internal process memory

Throw some light on Internal Process Memory.

- This memory consists of a set of high-speed registers. They work as temporary storage for instructions and data.

Explain compaction.

During the process of loading and removal of process into and out of the memory, the free memory gets broken into smaller pieces. These pieces lie scattered in the memory. Compaction means movement of these pieces close to each other to form a larger chunk of memory which works as a resource to run larger processes.

What are page frames?

Page frames are the fixed size contiguous areas into which the main memory is divided by the virtual memory.

What are pages?

- Pages are same sized pieces of logical memory of a program. Usually they range from 4 KB to 8 KB depending on the addressing hardware of the machine.

- Pages improve the overall system performance and reduces requirement of physical storage as the data is read in 'page' units.

Differentiate between logical and physical address.

- Physical addresses are actual addresses used for fetching and storing data in main memory when the process is under execution.

- Logical addresses are generated by user programs. During process loading, they are converted by the loader into physical address.

When does page fault error occur?

- It occurs when a page that has not been brought into main memory is accessed.

Explain thrashing.

- In virtual memory system, thrashing is a high page fault scenario. It occurs due to under-allocation of pages required by a process.

- The system becomes extremely slow due to thrashing leading to poor performance.

What are the basic functions of file management in OS?

- Creation and deletion of files/ directories.

- Support of primitives for files/ directories manipulation.

- Backing up of files on storage media.

- Mapping of files onto secondary storage.

Explain thread.

- It is an independent flow of control within a process.

- It consists of a context and a sequence of instructions for execution.

What are the advantages of using threads?

The main advantages of using threads are:

a.) No special communication mechanism is required.

b.) Readability and simplicity of program structure increases with threads.

c.) System becomes more efficient with less requirement of system resources.

What are the disadvantages of using threads?

The main disadvantages of using threads are:

- Threads cannot be re-used as they exist within a single process.

- They corrupt the address space of their process.

- They need synchronization for concurrent read-write access to memory.

What is a compiler?

A compiler is a program that takes a source code as an input and converts it into an object code. During the compilation process the source code goes through lexical analysis, parsing and intermediate code generation which is then optimized to give final output as an object code.

What is a library?

It is a file which contains object code for subroutines and data to be used by the other program.

What are the advantages of distributed system?

Advantages of distributed system are:

- Resources get shared

- Load gets shared

- Reliability is improved

- Provide a support for inter-process communication

What are the different types of scheduling algorithms?

The scheduling algorithms decide which processes in the ready queue are to be allocated to the CPU for execution. Scheduling algorithms can be broadly classified on the basis of:

- Preemptive algorithms

- Round Robin Scheduling

- Shortest Job First Scheduling (can be both)

- Priority Scheduling (can be both)

- Non-preemptive algorithms

- First Come First Served Scheduling

Non-Preemptive algorithms: In this type of scheduling once a CPU has been allocated to a process it would not release the CPU till a request for termination or switching to waiting state occurs.

Preemptive algorithms: In this type of scheduling a process maybe interrupted during execution and the CPU maybe allocated to another process.

Why is round robin algorithm considered better than first come first served algorithm?

The first come first served algorithm is the simplest scheduling algorithm known. The processes are assigned to the CPU on the basis of their arrival time in the ready queue. Since, it is non-preemptive once a process is assigned to the CPU, it will run till completion. Since a process takes the CPU till it is executed it is not very good in providing good response times. It can make other important processes wait un-necessarily.

On the other hand, the round robin algorithm works on the concept of time slice or also known as quantum. In this algorithm, every process is given a predefined amount of time to complete the process. In case, a process is not completed in its predefined time then it is assigned to the next process waiting in queue. In this way, a continuous execution of processes is maintained which would not have been possible in case of FCFS algorithm

Explain how a copying garbage collector works. How can it be implemented using semispaces?

The copying garbage collector basically works by going through live objects and copying them into a specific region in the memory. This collector traces through all the live objects one by one. This entire process is performed in a single pass. Any object that is not copied in memory is garbage.

The copying garbage collector can be implemented using semispaces by splitting the heap into two halves. Each half is a contiguous memory region. All the allocations are made from a single half of the heap only. When the specified heap is half full, the collector is immediately invoked and it copies the live objects into the other half of the heap. In this way, the first half of the heap then only contains garbage and eventually is overwritten in the next pass.

How does reference counting manage memory allocated objects? When can it fail to reclaim objects?

Reference counting augments every object with a count of the number of times an object has been referenced. This count is incremented every time a reference to that object is made. Also every time a reference is destroyed the reference is decremented. This process is repeated till the reference count becomes zero. Once the reference count of an object reaches zero the object can be reclaimed. In this way, reference counting systems can perform automatic memory management by keeping a count in every object. Any object that does not have a reference count can be considered to be dead and that memory can be reclaimed.

The reference counting method can fail to reclaim objects in case of cyclic references. There are no concrete ways to avoid this problem and it is always suggested to create an architecture that does not use a circular reference.

What differences are there between a semaphore wait signal and a condition variable wait signal?

Semaphore wait signal:

- They can be used anywhere except in a monitor.

- The wait() function does not always blocks its caller.

- The signal() function increments the semaphore counter and can release a process.

- If the signal() releases a process, the released and the caller both continue.

Condition Variable wait signal:

- It can only be used in monitors.

- The wait() function always blocks its caller.

- The signal() can either release a process or it is lost as if it never occurred.

- On signal() releasing a process either the caller or the released continues but not both at the same time.

For a deadlock to occur what are the necessary conditions

In order for deadlocks to occur there are four necessary conditions:

- Mutual Exclusion: The resources available are not sharable. This implies that the resources used must be mutually exclusive.

- Hold and Wait: Any process requires some resources in order to be executed. In case of insufficient availability of resources a process can take the available resources, hold them and wait for more resources to be available.

- No Preemption: The resources that a process has on hold can only be released by the process itself voluntarily. This resource cannot be preempted by the system.

- Circular Waiting: A special type of waiting in which one process is waiting for the resources held by a second process. The second process is in turn waiting for the resources held by the first process.

Why is the context switch overhead of a user-level threading as compared to the overhead for processes? Explain.

This is due to the reason that a context switch implementation is done by the kernel. During this process the state information is copied between the processor and the PCB (process control block) or the TCB (thread control block). Since the kernel does not know anything about user-level threads, technically it is not possible for it to be a user level thread context switch. The user level scheduler can do some limited state copying on the behalf of a thread prior to the control being handed to that thread. But this copying of state information is smaller compared to that of a kernel-level process. Also the process does not involve going into the kernel mode with the help of a system call.

State the advantages of segmented paging over pure segmentation?

In broad terms paging is a memory management technique that allows a physical address space of a process to be non-contiguous.

Segmented paging has a certain set of advantages over pure segmentation such as:

- Segmented paging does not have any source of external fragmentation.

- Since a segment existence is not restricted to a contiguous memory range it can be easily grown and does not have to adjust into a physical memory medium.

- With segmented paging the addition of an offset and a base is simpler as it is only an append operation instead of it being a full addition operation.

When does the Belady's anomaly occur?

The Belady's anomaly is a situation in which the number of page faults increases when additional physical memory is added to a system. This anomaly arises in some algorithms that implement virtual memory. The virtual memory allows programs larger than the physical memory space to execute. An algorithm suffers from this problem when it cannot guarantee that a page will be kept when a small number of frames are available. An optimal algorithm would not suffer from this problem as it replaces the page not to be used for the longest time. The anomaly occurs when the page replacement algorithm will remove a page that will be needed in the immediate future. An optimal algorithm will not select such a page that will be required immediately. This anomaly is also stated to be unbounded.

What complications does concurrent processing add to an operating system?

There are various complications of concurrent processing such as:

- A time sharing method must be implemented to allow multiple processes to have an access to the system. This will involve the preemption of processes that do not give up CPU on their own i.e. more than one process may be executing kernel code simultaneously.

- The amount of resources that a process can use and the operations that it may perform must be limited. The system resources and the processes must be protected from each other.

- Kernel must be designed to prevent deadlocks between the various processes, i.e. Cyclic waiting or hold and waiting must not occur.

- Effective memory management techniques must be used to better utilize the limited resources.

How can a VFS layer allow multiple file systems support?

The VFS layer also known as the virtual file system functions in many ways similar to object oriented programming techniques. It acts like an abstraction layer on top of a more specific file system. The VFS layer enables the OS to make system calls independent of the file system type used. Any file system that is used gives its function calls used and the data structures to the layer of VFS. The VFS layer translates a system call into the correct specific functions for the targeted file system. The program that is used for calling does not have a file system specific code also the system call structures used in upper levels are file system independent. The VFS layer translation translates the non-file system specific calls into a file system specific operation.

What are the pros and cons of using circuit switching?

The primary advantage of using circuit switching is that it ensures the availability of resources. That is it reserves the network resources required for a specific transfer prior to the transmission taking place. By doing so it ensures that no packet would be dropped and the required quality of service is met.

The disadvantage of using circuit switching is that it requires a round trip message to setup a reservation. By doing so as it provisions the resources ahead of the transmission it might lead to the suboptimal use of resources.

Circuit switching can be implemented for applications that have constant demand for network resources for long periods of time.

What problems are faced during the implementation of a network-transparent system?

A designer primarily faces two major problems while implementing a network-transparent system. They are as follows:

- The primary problem is to make all the processors and storage devices to appear transparent on the network. This implies that the distributed system should appear as a single centralized system to the users using the network.

There are two solutions to it:

- The Andrews files system

- The NFS system.

- Both these file systems (distributed) appear as a single file system to the user whereas in reality it may be distributed over the network.

- The secondary issue is regarding the user mobility. The designer would want any user to connect to the entire system overall rather than to a particular machine.

Explain the layers of a Windows XP system.

The layers of Windows XP system boot-up is as follows:

- A situation of operating system portability is created by the hardware abstraction layer by hiding hardware differences from the operating systems upper layers. A virtual machine interface is provided by the hardware abstraction layer to be used by the kernel dispatcher and the device drivers.

- The foundation provided by the kernel layer is used by the executive functions and the user mode sub systems. The kernel would always remain in memory and cannot be preempted. The functions of the kernel are thread scheduling, interrupt and exception handling etc.

- The executive layer is responsible for providing services to be used by all subsystems. These can be object manager, process manager, i/o manager etc.

Explain the booting process of a Windows XP system.

The steps involved are as follows:

- As the computer is powered on, the BIOS begins execution from ROM, it loads and executes the bootstrap loader.

- The NTLDR program is loaded from the root directory of the system disk and determines which boot disk contains the operating system.

- NTLDR loads the HAL library, kernel and system hive. The system hive indicates the required boot drivers and loads them one by one.

- Kernel execution begins by initializing the system and creating two processes: the system process containing all internal worker threads and the first user-mode initialization process: SMSS.

- SMSS further initializes the system by establishing paging files and loading device drivers.

- SMSS creates two processes: WINLOGON, which brings up the rest of the system and CSRSS, the Win32 subsystem process.

How are data structures handled by NTFS and how does it recover from a crash?

In an NTFS file system inside the transactions all the data structure updates are performed. Prior to the alteration of a data structure a transaction creates log record containing information on redo and undo functions. Once a transaction is completed commit record information is stored in the logs.

An NTFS system recovers from a crash by accessing information from the created log records. The first step is to redo operations of committed transactions and undoing those transactions which could not be successfully committed. Although the NTFS file system after recovering from a crash might not reflect the same user data prior to a crash but it can guarantee the file data structures are undamaged. It restores the structure to a pre-crash and consistent state.

What are the benefits and losses of placing the functionality in a device controller rather than in placing it in the kernel?

The benefits of placing functionality in the device controller are:

- System crasher due to the occurrence of a bug is greatly reduced.

- By the utilization of dedicated hardware and algorithms that are hard coded the performance can be improved greatly.

- Since the algorithms are hard coded the kernel gets simplified.

The banefits of placing functionality in the controller rather than the kernel are:

- Once a bug occurs they are difficult to fix, a new firmware or revision may be required.

- For performance improvement of algorithms hardware upgrades are required rather than a device driver update.

What are merits and demerits of systems supporting multiple file structure and systems supporting a stream of bytes?

The main advantage of having a system that supports multiple file structures is that the support for it is provided by the system itself no other individual application is required to provide the multiple structure support. Since the support is provided by the system itself the implementation is much more efficient as compared to application level.

A demerit of such kind of implementation is that it can increase the overall size of the system. Also, since the support is provided by the system, for an application that requires a different file type may not be executable on such a system.

A good alternative for this is that the OS does not define any support for file structures instead all files are considered to be a series of bytes. By doing so the support for file systems is simplified as the OS does not have to specify the different structures for the file systems. It allows the applications to define the file structures. This kind of implementation can be found in UNIX.

What do you understand by transaction atomicity?

The transaction process can be considered to be a series of read and write operations upon some data which is followed by a commit operation. By transaction atomicity it means that if a transaction is not completed successfully then the transaction must be aborted and any changes that the transactions did while execution must be roll backed. It means that a transaction must appear as a single operation that cannot be divided. This ensures that integrity of the data that is being updated is maintained. If the concept of atomicity in transaction is not used any transaction that is aborted midway may result in data to be inconsistent as there might be a possibility two transactions may be sharing the same data value.

Why is a single serial port managed with a single interrupt-driven I/O but a front-end processor is managed using a polling I/O, such as a terminal concentrator?

When the I/O is frequent and of very short durations polling is considered to be more efficient than an interrupt driven I/O. Although, a serial port individually can have fairly infrequent number of I/O and hence should ideally use interrupts the case of serial ports in a terminal concentrator is different.

A terminal concentrator consists of multiple serial ports and this can lead to the creation of multiple short I/O instances this can create un-necessary load on the system in case of interrupts usage.

Instead, if a polling loop is used it can greatly reduce the amount of load on the system by looping through without the requirement of I/O.

Due to this reason interrupts are used for single ports as the frequency of I/O on such a port is less and can be managed effectively, whereas we use polling for multiple ports as the frequency of I/O increases and are of short durations which suits polling.

What is graceful degradation?

- It is the ability to continue providing service proportional to level of hardware.

- Systems designed for graceful degradation are called fault tolerant.

- If we have several processors connected together, then failure of one would not stop the system.

- Then the entire system runs only 10% slower.

- This leads to increased reliability of the system.

What are loosely coupled systems?

- These systems are also called as the distributed systems.

- It consist of collection of processors that do not share memory or clock.

- The processors communicate through high speed buses or telephone lines.

- It can be a centralized system where the server responds to client requests.

- It can also be a peer to peer system.

Explain SMP.

- It is called as symmetric multiprocessing which is multiprocessor system.

- In it each processor runs an identical copy of the operating system.

- These copies communicate with one another as needed.

- These processor systems lead to increased throughput.

- These systems are also called parallel systems or tightly coupled systems.

What is DLM?

- It is the service called as distributed lock manager.

- In cluster systems to avoid file sharing the distributed systems must provide the access control and file locking.

- This ensures that no conflicting operations occur in the system.

- Here the distributed file systems are not general purpose therefore it requires locking.

Explain the handheld systems. List the issues related to the handheld system.

- Handheld devices are palm tops and cellular telephones with connectivity to a network.

- These devices are of limited size which leads to limited applications.

- They use a memory 512KB to 16MB as a result the operating system and applications must use the memory efficiently.

- The speed of the processors is only a fraction of speed of the PC processors and for faster processors larger battery is required.

- These devices use very small display screens so reading mails and browsing must be condensed to smaller displays.

Why is interrupt vector used in operating systems?

- The operating system these days are interrupt driven and this requires the interrupt vector.

- This interrupt vector contains the addresses of the interrupt service routines for various devices.

- Here the interrupts can be indirectly called through the table with no intermediate routine needed.

- This leads to interrupt handling at a faster rate.

- Operating systems like MS DOS and UNIX are using the interrupt vector.

What is the need of device status table?

- This table gives the device type, its address and status.

- It is required to keep a track of many input output requests at the same time.

- The state of the device can be functioning, idle or busy.

- If a device is busy, type of request and other parameters are stored in the table entry.

- If more than one processor issues request for the same device then a wait queue is maintained.

How can the speed of interrupt driven input output systems be improved?

- Direct memory access is used to enhance the speed of the input output systems.

- Here, buffers, counters and pointers are set for the devices.

- The device controller transfers the block of data directly from own buffer storage to memory.

- The data is not given to the CPU for further transfer between CPU and input output devices or CPU and memory.

- Only one interrupt is generated per block than one interrupt per byte which enhances the speed.

Explain the execution cycle for a von Neumann architecture.

- Initially the system will fetch the instruction and stores it in instruction register.

- Instruction is then decoded and may cause operands to be fetched from memory.

- After execution the result is stored in the memory.

- Here the memory unit sees only the memory addresses irrespective of how they are generated.

- Memory unit is also unaware of what addresses are for.

Explain the positioning time for a disk.

- It is also called as the random-access time used by a disk to perform operations.

- It consists of time to move the disk arm to the desired cylinder called the seek time.

- The time required for the desired sector to rotate to the disk head is called rotational latency.

- Typical disks can transfer megabytes of data per second.

- Seek time and rotational latency is always in milliseconds.

What is EIDE?

- EIDE is a bus called enhanced integrated drive electronics.

- The input output devices are attached to the computer by a set of wires called the bus.

- The data transfer on a bus are carried out by electronic processes called controllers.

- The host controller sends messages to device controller and device controller performs the operations.

- These device controllers consist of built in cache so that data transfer occurs at faster speed.

Differentiate between the user mode and monitor mode.

- User mode and monitor mode are distinguished by a bit called the mode bit.

- User mode uses bit 1 and monitor mode uses bit 0.

- At the boot time hardware starts with the monitor mode.

- Also, at the time of interrupt user mode is shifted to the transfer mode.

- System always switches to the user mode before passing control to the user program.

- Whenever system gains control of the computer it works in monitor mode otherwise in user mode.

What is time slice?

- The timer in CPU is set to interrupt every N milliseconds where this N is called the time slice.

- It is the time each user gets to execute before control is given to next user.

- At the end of each time slice the value of N is incremented and the record is maintained.

- It also maintains the record of the total time user program has executed thus far.

- This method helps in time sharing among the various users.

What are the activities related to the Time Shared User Program Management?

- An Operating System is responsible for the creation and deletion of both user and system processes.

- It also provides mechanism for the process synchronization.

- Suspending and resuming of windows is done by the operating system itself.

- Program needs resources like CPU time, memory, files, input output devices to complete the task **which is provided by the operating system.**

- Mechanisms are also provided for deadlock handling.

When an input file is opened, what are the possible errors that may occur?

- 1st condition may be that the file is protected against access, here it terminates abruptly.

- 2nd condition may be that file exists, then we need to create the output file.

- If file with the same name exists then it may be deleted or program may be aborted.

- In another case the system may ask the user to replace the existing file or abort the program.

Explain PCB.

- PCB, process control block, is also called as the task control block.

- It contains information about the process state like new, ready, running, waiting and halt.

- It also includes the information regarding the process priority and pointers to scheduling queues .

- Its counter indicates the address of the next instruction to be executed for the process.

- It basically serves as the storage for any information that may vary from process to process.

What is context switching ?

- It is the process of switching the CPU from one process to another.

- This requires to save the state of the old process and loading the saved state for the new process.

- The context of the process is represented in the process control block.

- During switching the system does no useful work.

- How the address space is preserved and what amount of work is needed depends on the memory management.

What is cascading termination?

- If one process is terminated, its related processes are also terminated abnormally then it is called cascade termination.

- It occurs in the case of parent child process.

- If the parent process is terminated normally or abnormally then all its child processes must be terminated.

- The parent is existing and the operating system does not allow a child to continue if its parent terminates.

- This child process is the new process created by the process called the parent process.

Explain IPC.

- It is called as the inter process communication.

- The scheme requires that processes share a common buffer pool and code for implementing the buffer.

- It allows processes to communicate and to synchronize their actions.

- Example: chat program used on the world wide web.

- It is useful in distributed computer systems where communicating processes reside on different computers connected with a network.

What are sockets?

- A socket is defined as endpoint for communication, a pair of sockets is used by the pair of processes.

- It is made of IP address chained with a port number.

- They use the client server architecture.

- Server waits for incoming client requests by listening to specified port.

- On reception of request, server accepts connection from client socket to complete the connection.