

Important Short Questions of OS

Q1. What is multiprogramming? Describe the objectives of multiprogramming?

Ans: Sharing the processor, when two or more programs reside in memory at the same time, is referred as multiprogramming. Multiprogramming assumes a single shared processor. Objectives of multiprogramming are following

- Improved memory utilization.
- Increased CPU utilization.
- Shorter response time.

Q2. Explain long term and short term Scheduler? OR Differentiate between Short Term and Long Term Scheduler?

Ans:

LONG TERM SCHEDULER	SHORT TERM SCHEDULER
An OS scheduler that selects processes from the job queue and loads them into the main memory for execution	An OS scheduler that selects a process among multiple processes to be executed by the CPU
Also called job scheduler	Also called CPU scheduler
Selects a process from the job queue and brings that to the ready queue	Selects a process in the ready queue to be executed by the CPU
Slower	Faster
Controls the degree of multiprogramming	Provides minimum control on the degree of multiprogramming
Runs less frequently	Runs frequently
	Visit www.PEDIAA.com

Q3. What is Page fault? Under what circumstances do page faults occur?

Ans: A page fault is an interruption that occurs when a software program attempts to access a memory block not currently stored in the system's RAM. This exception tells the operating system to find the block in virtual memory so it can be sent from a device's storage (SSD or HD) to RAM.

Q4. Differentiate between LFU Algorithm and MFU algorithms?

Ans:

Least Frequently Used (LFU)	Most Frequently Used (MFU)
In this algorithm whenever a page is to be replaced, that page is chosen that is least frequently used or least intensively referenced i.e. the page with smallest count is replaced.	In this algorithm whenever a page is to be replaced that page is chosen which has been used most frequently i.e. the page with largest count.

Q5. What are threads? What are types of threads?

Ans: A thread is a lightweight process and forms the basic unit of CPU utilization. A process can perform more than one task at the same time by including multiple threads.

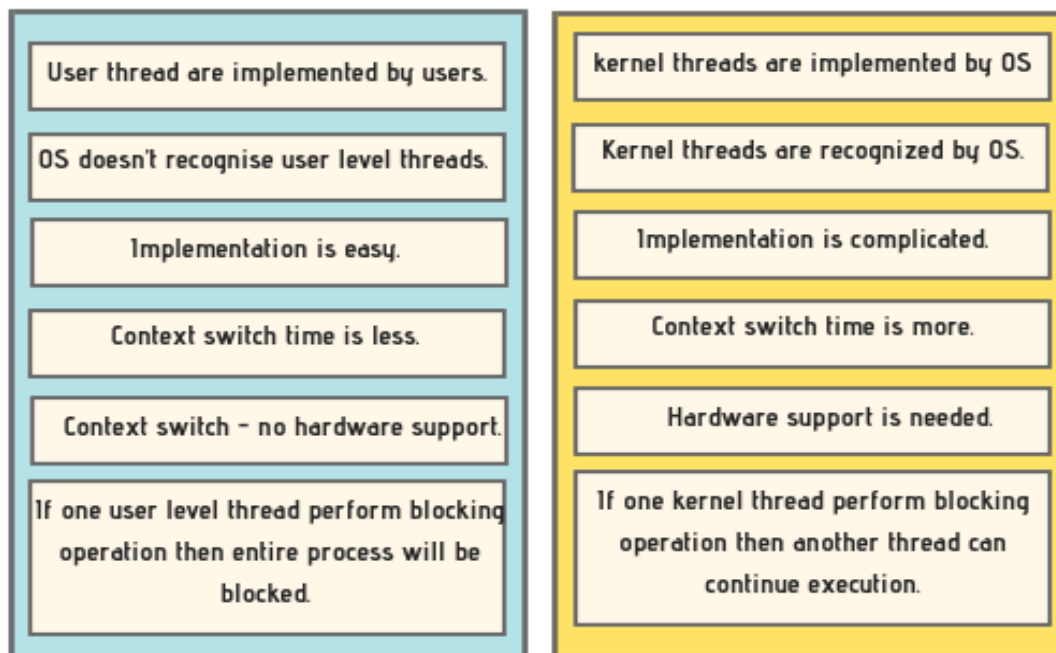
- A thread has its own program counter, register set, and stack
- A thread shares resources with other threads of the same process: the code section, the data section, files and signals.

There are two types of threads:

- User threads (User threads are implemented by users)
- Kernel threads (Kernel threads are implemented by OS)

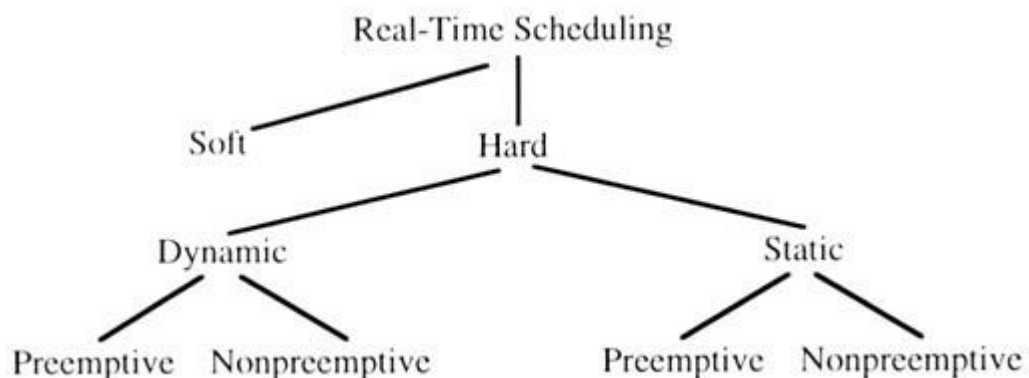
Q6. Write the differences between User-level and Kernel-level threads?

Ans:



Q7. Different types of Real – Time Scheduling?

Ans:



Q8. What is Semaphores? Also give the operations for accessing Semaphores?

Ans: Semaphore is a protected integer variable used to control access to a common resource by multiple processes in a concurrent system such as a multitasking operating system. When a process requires to access the semaphore it performs **wait ()** and when releasing the resource it performs **signal ()** operation. **wait ()** operation is also called **P**, **sleep**, or **Down** operation, and **signal ()** operation is also called **V**, **wake-up**, or **up** operation.

Q9. What is demand paging?

Ans: Demand Paging is a technique in which a page is usually brought into the main memory only when it is needed or demanded by the CPU. Initially, only those pages are loaded that are required by the process immediately.

Q10. What is meant by pure demand paging?

Ans: In some cases when initially no pages are loaded into the memory, pages in such cases are only loaded when are demanded by the process by generating page faults. It is then referred to as Pure Demand Paging.

Q11. Define Arrival time, Burst time, Completion time, Waiting time, Turnaround time and Response time?

Ans:

Arrival Time – Time at which the process arrives in the ready queue.

Burst Time – Time required by a process for CPU execution.

Completion Time – Time at which process completes its execution.

Turn Around Time – Time Difference between completion time and arrival time.

- $Turn\ Around\ Time = Completion\ Time - Arrival\ Time$

Waiting Time – Waiting time is the total time spent by the process in the ready state waiting for CPU.

- $Waiting\ time = Turnaround\ time - Burst\ time$

Response Time – Response time is the time spent when the process is in the ready state and gets the CPU for the first time.

- $Response\ time =$
 $Time\ at\ which\ the\ process\ gets\ the\ CPU\ for\ the\ first\ time - Arrival\ time$

Q12. What is Processor Affinity?

Ans: Processor affinity or CPU pinning, enables binding and unbinding of a process or multiple processes to a specific CPU core in a way that the process(es) will run from that specific core only.

Q13. Differentiate between I/O bound processes and CPU bound processes?

Ans:

CPU-bound and I/O-bound

- ❑ A process is **CPU-bound** if it generates I/O requests infrequently, using more of its time doing computation.
- ❑ A process is **I/O-bound** if it spends more of its time to do I/O than it spends doing computation.
- ❑ A CPU-bound process might have a few **very long** CPU bursts.
- ❑ An I/O-bound process typically has **many short** CPU bursts.

3

Q14. What is meant by context switch?

Ans: When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process this task is known as context switching.

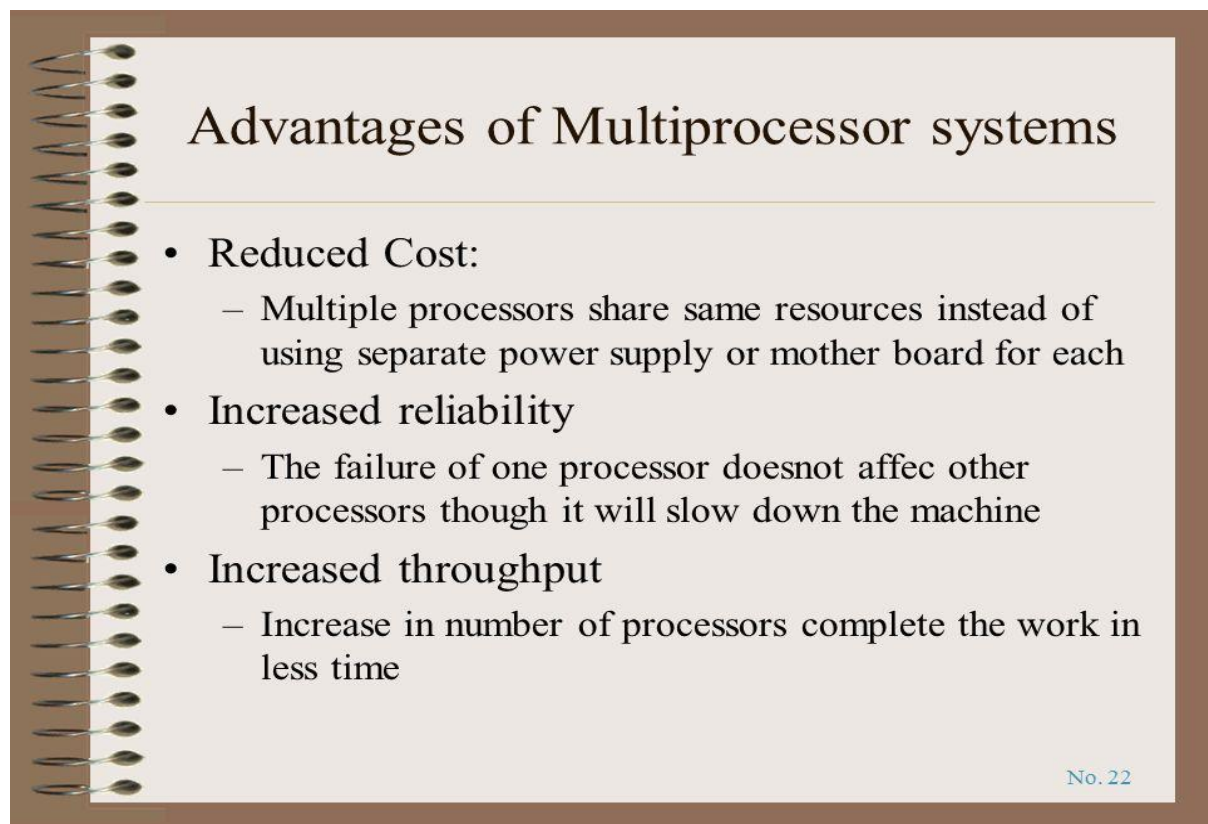
Q15. Discuss the benefits and drawbacks of short time quantum?

Ans:

Benefits	Drawbacks
Short quantum allows many processes to circulate through the processor quickly, each getting a brief chance to run. It provide better response time for short interactive processes.	Short quantum will cause too many process switches and will lower CPU efficiency.

Q16. State any three advantages of multiprocessor system?

Ans:



Q17. What is real time systems?

Ans:

What is a Real-Time System?

- Real-time systems have been defined as:
"those systems in which the correctness of the system depends not only on the logical result of the computation, but also on the time at which the results are produced";
 - J. Stankovic, "Misconceptions About Real-Time Computing," *IEEE Computer*, 21(10), October 1988.

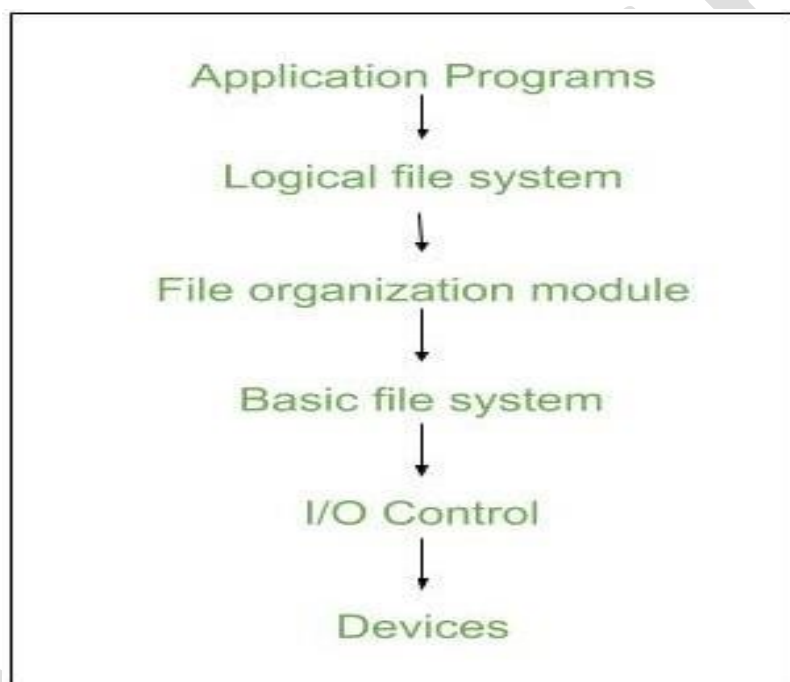
Q18. Define seek time and latency time/Rotational Latency?

Ans: Seek Time: Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or written.

Rotational Latency: Rotational Latency is the time taken by the desired sector of disk to rotate into a position so that it can access the read/write heads.

Q19. What are the various layers of a file system?

Ans: The image shown below, elaborates how the file system is divided in different layers



Q20. What are the advantages and disadvantages of Contiguous allocation?

Ans:

Advantages	Disadvantages
<ol style="list-style-type: none">1. It is simple to implement.2. We will get Excellent read performance.3. Supports Random Access into files.	<ol style="list-style-type: none">1. The disk will become fragmented.2. It may be difficult to have a file grow.

Q21. What is Spooling?

Ans: "Spool" is technically an acronym for "**Simultaneous peripheral operations online**". Spooling is a process in which data is temporarily held to be used and executed by a device, program or the system. Data is sent to and stored in memory or other volatile storage until the program or computer requests it for execution.

Q22. Define effective access time?

Ans: Effective access time is the average access time to memory items, where some items are cached in fast storage and other items are not cached.

$$EAT = \text{Hit Ratio} * TLB_hit_time + \text{Miss Ratio} * TLB_miss_time$$

Q23. Explain belady's anomaly?

Ans: In year 1970, Belady, Nelson and Shedler discovered that in FIFO page replacement, certain page reference patterns actually cause more page faults when the number of page frames allocated to a process is increased. This phenomenon is called the FIFO anomaly or Belady's anomaly.

Q24. What is Dispatcher? What is the responsibility of dispatcher?

Ans: The dispatcher is the module that gives a process control over the CPU after it has been selected by the short-term scheduler. Responsibilities of a dispatcher are following:

- Switching context
- Switching to user mode
- Jumping to the proper location in the user program to restart that program

Q25. What is meant by the term dispatch latency?

Ans: Dispatch latency is the time taken by the dispatcher to stop one process and start another. The lower the dispatch latency, the more efficient the software for the same hardware configuration.

Q26. What is a Critical Section? Give examples?

Ans: Critical Section is the portion of the code in the program where shared variables or resources are accessed and/or updated by various processes.

Examples:

- A piece of code that reads from or writes to a shared memory region.
- Or a code that modifies or traverses a linked list that can be accessed concurrently by another thread.

Q27. What is race condition?

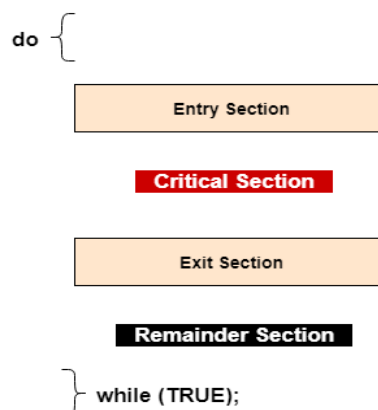
Ans:

Race Condition

- ☐ Incorrect behaviour of a program due to concurrent execution of critical sections by two or more threads.
- ☐ For example, if thread 1 deletes an entry in a linked list while thread 2 is accessing the same entry.

Q28. What is critical section problem?

Ans: The critical section problem is to implement a solution or design a set of protocols which can ensure that the Race condition among the processes will never arise.



Q29. What is mutual exclusion condition, progress and bounded-waiting?

Ans: These three are the conditions that a solution to the critical section problem must satisfy.

- **Mutual Exclusion**

Mutual exclusion implies that only one process can be inside the critical section at any time. If any other processes require the critical section, they must wait until it is free.

- **Progress**

Progress means that if a process is not using the critical section, then it should not stop any other process from accessing it. In other words, any process can enter a critical section if it is free.

- **Bounded Waiting**

Bounded waiting means that each process must have a limited waiting time. It should not wait endlessly to access the critical section.

Q30. What is a Job queue, ready queue and device queue? OR What are various scheduling queues?

Job queue – This queue keeps all the processes in the system.

Ready queue – This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.

Device queues – The processes which are blocked due to unavailability of an I/O device constitute this queue.

Q31. Explain how Demand Paging affects the performance of a computer system?

Ans: If one access out of 1000 causes a page fault, the effective access time is 8.2 microseconds. The computer will be slowed down by a factor of 40 because of demand paging. Therefore, it is important to keep the page-fault rate low in a demand paging system. Otherwise, the effective access time increases slowing process execution.

Q32. Is it a wise approach to reserve an array in zero-capacity buffer?

Ans: No it is not a wise approach because we can't store data in zero capacity buffer. So implementing array does not make sense.

Q33. Describe the Safe, unsafe, and deadlock state spaces?

Ans: Safe State – If the system can allocate resources to the process in such a way that it can avoid deadlock. Then the system is in a safe state.

Unsafe State – If the system can't allocate resources to the process safely, then the system is in an unsafe state.

Deadlock State – If a process is in the waiting state and is unable to change its state because the resources required by the process is held by some other waiting process, then the system is said to be in Deadlock state.

Q34. Define Multi-threading? Explain its benefits?

Ans: Multithreading is a CPU (central processing unit) feature that allows two or more instruction threads to execute independently while sharing the same process resources.

Benefits of Multithreaded Programming

- **Responsiveness**
Threads don't block each other
- **Resource Sharing**
Can access same resources
- **Economy**
Cheaper to create a new thread
- **Utilization of Multi-processors**
Program runs faster

Q35. Explain the importance of Real-Time Embedded systems?

Ans: Real-time embedded systems are required to perform their assigned function or furnish the output under strict time constraints. These systems are used in many advance technologies like Air traffic control, Missile Launching system and Heart pacemaker etc.

Q36. What is multitasking?

Ans: Multitasking is when multiple jobs are executed by the CPU simultaneously by switching between them. Switches occur so frequently that the users may interact with each program while it is running. Multitasking is based on time sharing alongside the concept of context switching.

Q37. What are the Time Sharing Systems?

Ans:

Time Sharing Systems:

- Time sharing, or multitasking, is a logical extension of multiprogramming.
- Multiple jobs are executed by switching the CPU between them.
- In this, the CPU time is shared by different processes, so it is called as "Time sharing Systems".
- Time slice is defined by the OS, for sharing CPU time between processes.
- Examples: Multics, Unix, etc.,

Q38. What is interrupt?

Ans: Interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention. It alerts the processor to a high-priority process requiring interruption of the current working process.

Q39. Write down the name of primary thread libraries?

Ans:

Thread Libraries

Thread libraries provide programmers with an API for creating and managing threads.

Thread libraries may be implemented either in user space or in kernel space.

There are three main thread libraries in use today:

- ❖ POSIX Pthreads .
- ❖ Win32 threads.
- ❖ Java threads .

Q40. Differences between logical and physical addresses?

Ans:

Logical Address	Physical Address
An address at which an item such as memory cell, storage element appears to reside from the perspective of an executing program	A memory address that allows accessing a particular storage cell in the main memory.
User can view the logical address of a program.	User can never view physical address of program.
Generated by CPU	Computed by MMU

Q41. What are necessary conditions which can lead to a deadlock situation in a system?

Ans: Mutual exclusion – If a resource is shareable and can be accessed by more than one process at the same time. Then it leads to a deadlock.

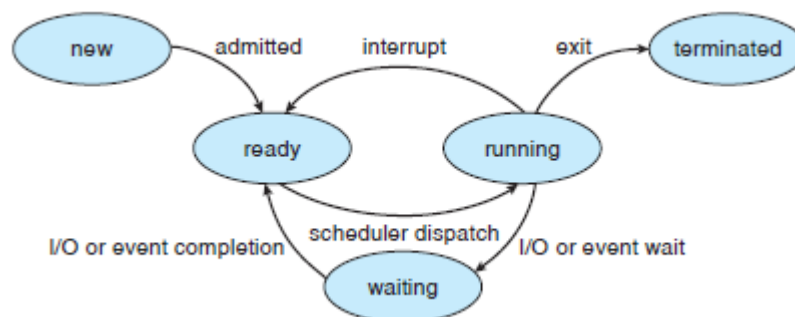
Hold and wait – If one process holding a resource and waiting for another resource that is held by another process. Then it leads to a deadlock.

Circular wait – One process is waiting for the resource, which is held by the second process, which is also waiting for the resource held by the third process etc. This will continue until the last process is waiting for a resource held by the first process. This creates a circular chain.

No preemption – If we have set no priority for all processes, then every process demands that it will execute first and utilize the resources. Then it leads to a deadlock.

Q42. What are the states of a process? OR List down the process states?

Ans:



Q43. What is thrashing?

Ans: Thrashing is a condition or a situation when the system is spending a major portion of its time in servicing the page faults, but the actual processing done is very negligible. As a result, no useful work would be done by the CPU and the CPU utilization is reduced.

Q44. What is cause of thrashing?

Ans: Thrashing is caused by under allocation of the minimum number of pages required by a process, forcing it to continuously page fault. Due to higher page fault rate CPU utilization is low.

Q45. Define Preemptive and Non Preemptive Scheduling?

Ans: Preemptive Scheduling – The scheduling in which a running process can be interrupted if a high priority process enters the queue and is allocated to the CPU is called preemptive scheduling.

Non Preemptive Scheduling – The scheduling in which a running process cannot be interrupted by any other process is called non-preemptive scheduling. Any other process which enters the queue has to wait until the current process finishes its CPU cycle.

Q46. What is difference between preemptive and non-preemptive scheduling?

Ans:

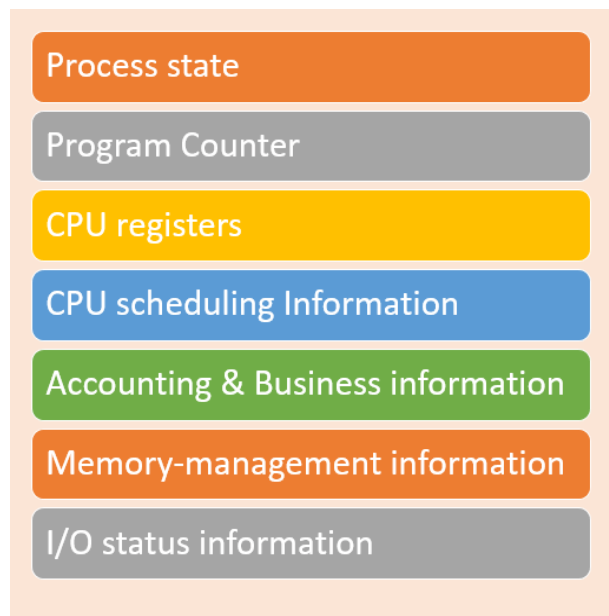
Parameter	Preemptive Scheduling	Non-Preemptive Scheduling
Basic	In this resources (CPU cycle) are allocated to a process for a limited time.	Once resources (CPU cycle) are allocated to a process, the process holds it till it completes its burst time or switches to waiting state
Interrupt	Process can be interrupted in between.	Process cannot be interrupted until it terminates itself or its time is up.
Flexibility	Flexible	rigid
Cost	Cost associated	No cost associated
CPU Utilization	In preemptive scheduling, CPU utilization is high.	It is low in non -preemptive scheduling.
Examples	Examples of preemptive scheduling are Round Robin and Shortest Remaining Time First.	Examples of non-preemptive scheduling are First Come First Serve and Shortest Job First.

Q47.What is system call? What is the purpose of system Calls?

Ans: A system call is a method for a computer program to request a service from the kernel of the operating system on which it is running. A system call is a method of interacting with the operating system via programs. Purpose of System call is to provide the services of the operating system to the user programs via Application Program Interface(API).

Q48. What is PCB? What is the purpose of PCB?

Ans: A Process Control Block or simple PCB is a data structure that is used to store the information of a process that might be needed to manage the scheduling of a particular process. The process control block typically contains:



Q49. Define Starvation?

Ans: Starvation is the problem that occurs when high priority processes keep executing and low priority processes get blocked for indefinite time. A steady stream of higher-priority methods will stop a low-priority process from ever obtaining the processor.

Q50. What is virtual memory?

Ans: A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard disk that's set up to emulate the computer's RAM.

Q51. What is Process Synchronization?

Ans: Process Synchronization means sharing system resources by processes in such a way that, Concurrent access to shared data is handled thereby minimizing the chance of inconsistent data.

Q52. What is a garbage collection?

Ans: Garbage collection (GC) is a dynamic approach to automatic memory management and heap allocation that processes and identifies dead memory blocks and reallocates storage for reuse. The primary purpose of garbage collection is to reduce memory leaks.

Q53. What is disk controller?

Ans: A hard disk controller (HDC) is an electrical component within a computer hard disk that enables the processor or CPU to access, read, write, delete and modify data to and from the hard disk. Essentially, an HDC allows the computer or its processor to control the hard disk.

Q54. What are the main purpose of an operating system?

Ans: An operating system has three main functions:

- Manage the computer's resources, such as the central processing unit, memory, disk drives, and printers.
- Establish a user interface.
- Execute and provide services for applications software.

Q55. What is Paging?

Ans: Paging is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main memory frames in the form of pages. A frame is basically a place where a page can be placed.

Q56. What is the purpose of Paging and page tables?

Ans: The main purpose of paging is to allow the physical address space of the process to be non-contiguous, so that a process can be allocated memory anywhere wherever a free frame is available in main memory. Purpose of Page table is to store the mapping between physical and logical addresses.

Q57. Define Spin clock?

Ans: A spinlock is a lock that causes a thread trying to acquire it to simply wait in a loop ("spin") while repeatedly checking whether the lock is available. Since

the thread remains active but is not performing a useful task, the use of such a lock is a kind of busy waiting.

Q58. Which of the following scheduling algorithms can lead to starvation? FIFO, Shortest Job First, Priority, Round Robin?

Ans:

- Shortest Job First => if short processes are continuously added.
- Priority Scheduling => if high priority processes are continuously added.

Q59. What is fragmentation?

Ans: Fragmentation is a phenomenon of memory wastage. It reduces the capacity and performance because space is used inefficiently. There are two types of fragmentation,

- Internal fragmentation.
- External fragmentation

Q60. Differentiate between internal and external fragmentation?

Ans:

Internal Fragmentation	External Fragmentation
A form of fragmentation that arises When the memory assigned to the process is slightly larger than the memory requested by the process this creates free space in the allocated block causing internal fragmentation.	A form of fragmentation that arises when there is enough memory available to allocate for the process but that available memory is not contiguous.
Fixed-sized memory blocks are designated for internal fragmentation.	Variable-sized memory blocks are designated for external fragmentation.
The best-fit block is the solution to internal fragmentation.	Paging and compaction are solutions to external fragmentation.

Q61. Define Hit Ratio?

Ans: Hit ratio is defined as the percentage of times that a page number is found in the associative registers.

- $\text{Hit ratio} = \text{hit} / (\text{hit} + \text{miss}) = \text{no. of hits} / \text{total accesses}$

Q62. What are the capacities of queues in message passing system?

Ans:

1. **Zero capacity** – no messages are queued on a link.
 - Sender must wait for receiver (rendezvous)
2. **Bounded capacity** – finite length of n messages
 - Sender must wait if link full
3. **Unbounded capacity** – infinite length
 - Sender never waits

Q63. What is a batch system?

Ans: Batch systems are those systems in which user who is using a batch operating system do not interact with the computer directly. There is an operator which takes the jobs and create groups of the jobs that perform similar functions. These job groups are treated as a batch and executed simultaneously.

Q64. What is the “degree of multiprogramming”?

Ans: The degree of multiprogramming describes the maximum number of processes that a single-processor system can accommodate efficiently. The primary factor affecting the degree of multiprogramming is the amount of memory available to be allocated to executing processes.

Q65. What is cascading termination?

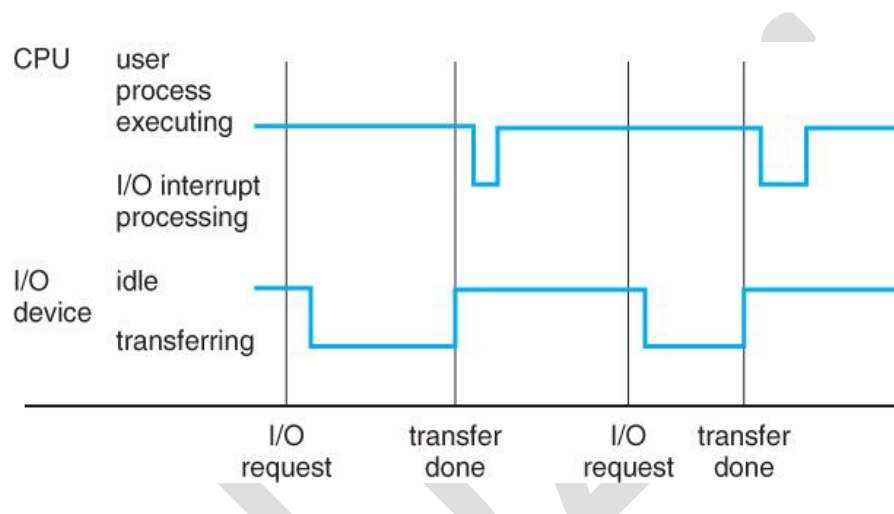
Ans: When a process creates a new process, the identity of the newly created process is passed to its parent. When a parent process is terminating, then all of its children process is also terminated. This phenomenon is known as "Cascading Termination" and is normally initiated by the operating system.

Q66. What is round robin scheduling?

Ans: Round-robin is a CPU scheduling algorithm in which each ready task runs turn by turn only in a cyclic queue for a limited time slice. This algorithm also offers starvation free execution of processes.

Q67. Draw Interrupt Timeline clearly mentioning interrupt occurred by a process?

Ans:



The I/O device (controller) is busy transferring data from the device buffer to the device. It goes from idle to transferring. This is the peak for I/O device. It goes back to idle when the transfer is done, until the next request.

The CPU curve shows a peak when the transfer is done because the CPU is notified by the device (through an interrupt).

Q68. What is virtual machine?

Ans: A virtual machine (VM) is a virtual environment that works like a computer within a computer. It runs on an isolated partition of its host computer with its own CPU power, memory, operating system (such as Windows, Linux, macOS), and other resources. End users can run applications on VMs and use them as they normally would on their workstation.

Q69. What is dynamic linking?

Ans: When one program is dependent on some other program. In such a case, rather than loading all the dependent programs, CPU links the dependent

programs to the main executing program when its required. This mechanism is known as Dynamic Linking. Dynamic linking refers to the linking that is done during load or run-time and not when the exe is created.

Q70. What is compaction? Why use it?

Ans: Compaction is a process in which the free space is collected in a large memory chunk to make some space available for processes. Compaction is used to minimize the probability of external fragmentation.

Q71. With what type of fragmentation does Paging and Segmentation suffers from?

Ans:

- Paging suffer from internal fragmentation
- Segmentation suffer from external fragmentation.

Q72. What does “preemptive” mean?

Ans: Preemption as used with respect to operating systems means the ability of the operating system to preempt (that is, stop or pause) a currently scheduled task in favour of a higher priority task.

Q73. What are the disadvantages of single Contiguous Memory Allocations?

Ans:

- Internal fragmentation occurs in the contiguous memory allocation.
- CPU time is wasted when the program has to wait for some resources.

Q74. What is swapping?

Ans: Swapping is a memory management scheme in which any process can be temporarily swapped from main memory to secondary memory so that the main memory can be made available for other processes. It is used to improve main memory utilization.

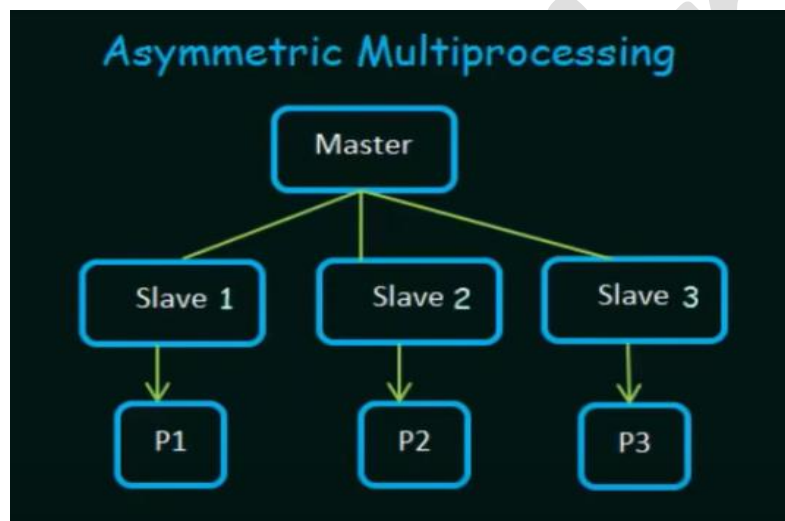
Q75. What is system call to create child process?

Ans: ‘fork()’ system call is used to create processes. It takes no arguments and returns a process ID. The purpose of the fork() is to create a new process, which

becomes the child process of the caller. After a new child process is created, both processes will execute the next instruction following the fork() system call.

Q76. Suppose there is an organization which hired a person for dividing tasks among other persons which multiprocessing environment does this organization depict?

Ans: This organization depicts asymmetric multiprocessing environment because the processors in asymmetric multiprocessing have a master slave relationship i.e. a processor assign processes to other processors.



Q77. What is the difference between deadlock avoidance, prevention and detection?

Ans: Deadlock Prevention – Deadlock prevention means to block at least one of the four conditions required for deadlock to occur.

Deadlock Avoidance – Deadlock avoidance means checking the state of the system, in advance and sees if the allocation of a resource will lead to a deadlock condition, if so it should wait, so that system does not go in unsafe state.

Deadlock detection – In this method, the OS assumes that a deadlock will occur in the future. So it runs a deadlock detection mechanism with a certain interval of time, and when it detects the deadlock, it starts a recovery approach.

Q78. FIFO and LRU both use previous information in page replacement policy. How is one different from another then?

Ans: In FIFO When a page needs to be replaced the oldest page which is at the front of the queue is selected for removal. Where as in LRU, whenever page replacement happens, the page which has not been used for the longest amount of time is replaced.

Q79. Differentiate between concurrency and parallelism with example?

Ans: Concurrency – Concurrency is when two or more tasks can start, run, and complete in overlapping time periods. It doesn't necessarily mean they'll ever both be running at the same instant. For example, *multitasking* on a single-core machine.

Parallelism is when tasks *literally* run at the same time, e.g., on a multicore processor.

Q80. Mention at least 4 system calls when you enter a command that copies a file from one path to another?

Ans:

Windows	Linux
CreateFile()	open()
ReadFile()	read()
WriteFile()	write()
CloseHandle()	close()

Q81. Why Peterson solution is not violation Bounded-wait?

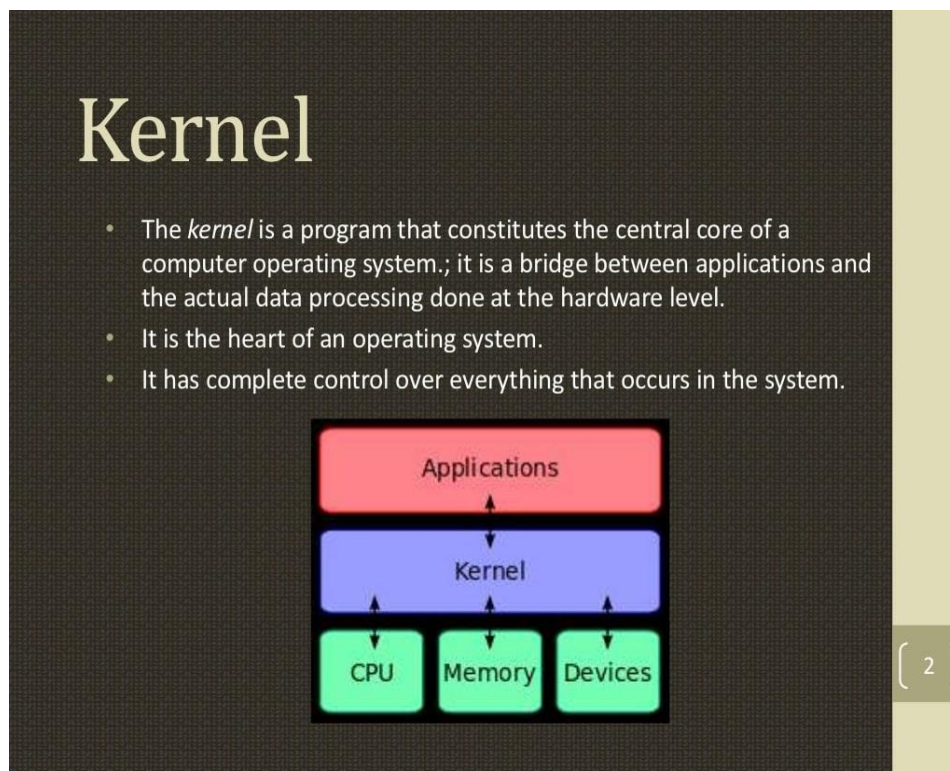
Ans: In Peterson's algorithm, a process will never wait longer than one turn for entrance to the critical section that's why Peterson solution is not violating bounded-wait.

Q82. Why do we call a program passive entity and a process active entity?

Ans: Program is a passive entity as it resides in the secondary memory such as a file containing a list of instructions stored on disk. Whereas process is an active entity as it is created during execution and loaded into the main memory.

Q83. What is kernel?

Ans:



Q84. Why SJF can't be used in real-time environment when you don't have execution history of the programs?

Ans: To successfully implement SJF, the burst time/duration time of the processes should be known to the processor in advance, which is practically not feasible in real time systems.

Q85. Write names of all preemptive and non-preemptive scheduling policies?

Ans:

Preemptive Policies	Non Preemptive Policies
<ul style="list-style-type: none">✓ SRTF (Shortest remaining time first)✓ LRTF (Longest remaining time first)✓ Round Robin✓ Priority Based	<ul style="list-style-type: none">✓ FCFS (First come first serve)✓ SJF (Shortest job first)✓ LJF (Longest job first)✓ HRRN (Highest Response Ratio Next)

Q86. Suppose that we have free segments with sizes 6, 17, 25, 14, and 19. Place a program with size 13kb in the free segment using first-fit, best-fit and worst fit?

Ans:

- ✓ **First Fit:** 13KB put in 17KB partition
- ✓ **Best Fit:** 13KB put in 14KB partition
- ✓ **Worst Fit:** 13KB put in 25KB partition

Q87. How to implement hold and wait which can ensure that a deadlock will not occur?

Ans: Allocate all required resources to the process before the start of its execution, this way hold and wait condition is eliminated and deadlock will not occur. However, its Practical implementation is not possible because a process can't determine necessary resources initially.

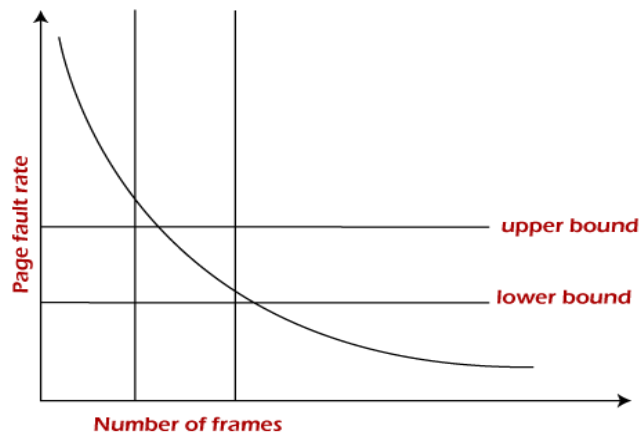
Q88. What is difference between microkernel and layered operating system structure?

Ans: Micro-Kernel: This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs. This result in a smaller kernel called the micro-kernel.

Layered OS: In this structure the OS is broken into number of layers (levels) each of these layers performs some kind of functionality. This simplifies the debugging process and increases modularity.

Q89. How page fault frequency can be used as a method of thrashing?

Ans: Page fault frequency method in an approach to prevent thrashing. The concept here is to control the page fault rate. Upper and lower limits are established on the desired page fault rate. If the page fault rate falls below the lower limit, frames can be removed from the process. Similarly, if the page fault rate exceeds the upper limit, more number of frames can be allocated to the process.



Q90. Page table can be placed either in CPU registers or main memory. What will be the criteria to place the page table in CPU registers?

Ans: The hardware implementation of page table can be done by using dedicated registers. But the usage of register for the page table is satisfactory only if page table is small. If page table contain large number of entries then we can use main memory.

Q91. Define file system?

Ans: A file system is a process that manages how and where data on a storage disk, typically a hard disk drive (HDD), is stored, accessed and managed. It is a logical disk component that manages a disk's internal operations as it relates to a computer and is abstract to a human user.

Q92. What is API?

Ans: An application program interface (API) is code that allows two software programs to communicate with each other. An API defines the correct way for a developer to request services from an operating system (OS) or other application.

Q93. What are the Deadlock Characterization?

Ans:



Deadlock Characterization

Deadlock can arise if four conditions hold simultaneously:

1. Mutual Exclusion
2. Hold and Wait
3. No Preemption
4. Circular Wait

Q94. What are the two types of Multiprocessing?

Ans:

- Symmetric Multiprocessing
- Asymmetric Multiprocessing

Q95. Why page table is needed to be paged?

Ans: In certain situations the page tables could become large enough that by paging the page tables, one could simplify the memory allocation problem by ensuring that everything is allocated as fixed-size pages as opposed to variable-sized chunks and also enable the swapping of portions of page table that are not currently used.

Q96. When cycle is both necessary and sufficient to detect deadlock?

Ans: If there is a cycle in the graph and each resource has only one instance, then there is deadlock. In this case, a cycle is a necessary and sufficient condition for deadlock. If there is a cycle in the graph, and each resource has more than one instance, there may or may not be deadlock.

Q97. What is processor Affinity?

Ans: Processor affinity, or CPU pinning or “Cache Affinity”, enables the binding and unbinding of a process or a thread to a central processing unit (CPU) or a

range of CPU's so that the process or thread will execute only on the designated CPU or CPU's rather than any CPU.

Q98. Differentiate between Soft affinity and Hard affinity?

Ans:

Engineered for Tomorrow

Soft affinity

- OS has a policy for keeping the process run on the same processor, but doesn't guarantee its implementation.
- Ex: Linux provides system calls, which support migration of process

Hard affinity

- OS specifies the process, not to migrate to other processors.
- Ex: Solaris limits processes to processor sets.

11/9/2018

mvi COLLEGE ENGINEER

Q99. When and for what purpose Banker's algorithm is used?

Ans: Banker's algorithm is used to avoid deadlock and allocate resources safely to each process in the computer system. It is named as Banker's algorithm on the banking system where a bank never allocates available cash in such a manner that it can no longer satisfy the requirements of all of its customers.

Q100. Thread Control Block?

Ans: Thread Control Block (TCB) is a data structure in the operating system kernel which contains thread-specific information needed to manage it. The TCB is the manifestation of a thread in an operating system. An example of information contained within a TCB is:

Thread ID
Thread state
CPU information : Program counter Register contents
Thread priority
Pointer to process that created this thread
Pointer(s) to other thread(s) that were created by this thread

Written by
M.Uzair