Operating System-Important questions

Introduction to Operating system

Q1 What is an operating system ?

- An operating system is a program that acts as an intermediary between the user and the computer hardware.
- The purpose of an OS is to provide a convenient environment in which users can execute programs in a convenient and efficient manner.
- It is a resource allocator responsible for allocating system resources and a control program which controls the operation of the computer hardware.

Q2 What are the types of operating systems ?

1. Batch OS – A set of similar jobs are stored in the main memory for execution. A job gets assigned to the CPU, only when the execution of the previous job completes.
2. Multiprogramming OS – The main memory consists of jobs waiting for CPU time. The OS selects one of the processes and assigns it to the CPU. Whenever the executing process needs to wait for any other operation (like I/O), the OS selects another process from the job queue and assigns it to the CPU. This way, the CPU is never kept idle and the user gets the flavor of getting multiple tasks done at once.
3. Multitasking OS – Multitasking OS combines the benefits of Multiprogramming OS and CPU scheduling to perform quick switches between jobs. The switch is so quick that the user can interact with each program as it runs
4. Time Sharing OS – Time-sharing systems require interaction with the user to instruct the OS to perform various tasks. The OS responds with an output. The instructions are usually given through an input device like the keyboard.
5. Real Time OS – Real-Time OS are usually built for dedicated systems to accomplish a specific set of tasks within deadlines.

Q3 Name some functions of the Operating system.

- Memory management

- Processor management
- Device management
- File management
- Security
- Job accounting
- Control over system performance
- Error detection
- Communicate between user and software
- Communication between software and hardware.

Q4. What is a kernel ?

Kernel is the core of every operating system. It connects applications to the actual processing of data. It also manages all communications between software and hardware components to ensure usability and reliability.

Q5. Define the two most famous types of Kernels.

Answer: Though there are many types of kernels, only two of them are considered in use.

- Monolithic Kernel
- MicroKernel

Monolithic Kernel: In this type of Kernel, all the User services and kernel services reside in the same memory space. The old operating system would use this type of Kernel. Some examples are Linux, Windows 95, 98, Unix, etc.

MicroKernel: This type of Kernel is small in size, and all the User and Kernel services reside in the different memory addresses. Operating systems like Mac OS X, windows use this type of Kernel.

Q6 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 turn 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.

Q7 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.compaction

Q8 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.

Q9 Number of processes with the fork (Microsoft)

```
main()

{

   fork();

   fork();

   fork();

}
```

How many new processes will be created?

A) 6

B) 7

C) 8

D) 5

At any point the number of processes running is 2^N (Where N being the number of fork calls).

So here 1st Fork creates 1 additional process - Total of 2 process

On 2nd fork call each of the 2 processes creates one additional process - Total of 4 process

On 3rd fork call each of the 4 processes creates new additional process - Total of 8 process

To get the count of new processes : Remove the root process from the count so 8 -1 = 7

Answer B

Q10 Define user mode and kernel mode. Why are two modes required?

User Mode

The system is in user mode when the operating system is running a user application such as handling a text editor. The transition from user mode to kernel mode occurs when the application requests the help of the operating system or an interrupt or a system call occurs.

The mode bit is set to 1 in the user mode. It is changed from 1 to 0 when switching from user mode to kernel mode.

Kernel Mode

The system starts in kernel mode when it boots and after the operating system is loaded, it executes applications in user mode. There are some privileged instructions that can only be executed in kernel mode.

These are interrupt instructions, input output management etc. If the privileged instructions are executed in user mode, it is illegal and a trap is generated.

The mode bit is set to 0 in the kernel mode. It is changed from 0 to 1 when switching from kernel mode to user mode.

Necessity of Dual Mode (User Mode and Kernel Mode) in Operating System

The lack of a dual mode i.e user mode and kernel mode in an operating system can cause serious problems. Some of these are:

- A running user program can accidentally wipe out the operating system by overwriting it with user data.
- Multiple processes can write in the same system at the same time, with disastrous results.

These problems could have occurred in the MS-DOS operating system which had no mode bit and so no dual mode.

Q11 What difference is between a loosely coupled and tightly coupled system.

BASIS FOR COMPARISON	LOOSELY COUPLED MULTIPROCESSOR SYSTEM	TIGHTLY COUPLED MULTIPROCESSOR SYSTEM
Basic	Each processor has its own memory module.	Processors have shared memory modules.
Efficient	Efficient when tasks running on different processors, has minimal interaction.	Efficient for high-speed or real-time processing.
Memory conflict	It generally does not encounter memory conflict.	It experiences more memory conflicts.
Interconnections	Message transfer system (MTS).	Interconnection networks PMIN, IOPIN, ISIN.
Data rate	Low.	High.
Expensive	Less expensive.	More expensive.

Q12 What Is Multitasking?

As the name itself suggests, multitasking refers to execution of multiple tasks (say processes, programs, threads etc.) at a time. In the modern operating systems, we are able to play MP3 music, edit documents in Microsoft Word, surf Google Chrome all simultaneously, this is accomplished by means of multi tasking.

Multitasking is a logical extension of multiprogramming. The major way in which multitasking differs from multi programming is that multi programming works solely on the concept of context switching whereas multitasking is based on time sharing alongside the concept of context switching.

Q13 What is Multiprogramming ?

In a modern computing system, there are usually several concurrent application processes which want to execute. Now it is the responsibility of the Operating System to manage all the processes effectively and efficiently.

One of the most important aspects of an Operating System is to multi program.

In a computer system, there are multiple processes waiting to be executed, i.e. they are waiting when the CPU will be allocated to them and they begin their execution. These processes are also known as jobs. Now the main memory is too small to accommodate all of these processes or jobs into it. Thus, these processes are initially kept in an area called job pool. This job pool consists of all those processes awaiting allocation of main memory and CPU.

CPU selects one job out of all these waiting jobs, brings it from the job pool to main memory and starts executing it. The processor executes one job until it is interrupted by some external factor or it goes for an I/O task.

Q14 What is Multithreading ? (Adobe)

Multi threading is the ability of a process to manage its use by more than one user at a time and to manage multiple requests by the same user without having to have multiple copies of the program.

Q15 What is Multiprocessing ?

Multiprocessing is the use of two or more CPUs (processors) within a single Computer system. The term also refers to the ability of a system to support more than one processor within a single computer system. Now since there are multiple processors available, multiple processes can be executed at a time. These multi processors share the computer bus, sometimes the clock, memory and peripheral devices also.

Why use multi processing –

- The main advantage of a multiprocessor system is to get more work done in a shorter period of time. These types of systems are used when very high speed is required to process a large volume of data. Multi processing systems can save money in comparison to single processor systems because the processors can share peripherals and power supplies.
- It also provides increased reliability in the sense that if one processor fails, the work does not halt, it only slows down. e.g. if we have 10 processors and 1 fails, then the work does not halt, rather the remaining 9 processors can share the work of the 10th processor. Thus the whole system runs only 10 percent slower, rather than failing altogether.

Q16 What are different types of Multiprocessing ?

Sr. No.	Key	Asymmetric Multiprocessing	Symmetric Multiprocessing
1	**CPU**	All processors are not equal in precedence.	All processors are the same in precedence.
2	**OS Task**	OS task is done by the master processor.	OS tasks can be done by any processor.
3	**Communication Overhead**	No communication overhead between processors as they are controlled by the master processor.	All processors communicate to each

			other using shared memory.
4	**Process Scheduling**	Master-Slave approach is used.	A ready queue of processes is used.
5	**Cost**	Asymmetric multiprocessing is cheaper to implement.	Symmetric multiprocessing is costlier to implement.
6	**Design Complexity**	**Asymmetric multiprocessing is simpler to design.**	**Symmetric multiprocessing is complex to design.**

Q17 Difference between Hard Real Time and Soft Real Time Systems ?

Hard real time	Soft real time
Hard response time is required.	Soft response time is required.
Data integrity is short term.	Data integrity is long term.
Size of the data file is small or medium.	medium. Size of the data file is large.
Peak load performance is predictable.	Peak load performance is degraded.
Hard real-time systems have little laxity and generally provide full deadline compliance.	Soft real-time systems are more flexible. They have greater laxity and can tolerate certain amounts of deadline misses.
Safety critical systems are typically hard real-time system.	Linux and many OS provide a soft real time system.

Q18 What is IPC and state some of them?

IPC means inter process communication- process to process notification, process to process synchronization which allows a programmer to coordinate activities among different program processes that can run concurrently in an operating system

Some of the common IPC mechanism are:

- Message Queue:

A queue of messages that is maintained between processes used for exchange for messages and other important information among processes.

- Shared Memory:

In this memory (page) is shared among the processes in user space and one process can write into it and other can read.

- Pipe:

A pipe is a technique for passing information from one program process to another. Basically, a pipe passes a parameter such as the output of one process to another process which accepts it as input.

Example:

ps –ef | grep “skype”

What it does that the output of “ps –ef” is given as an input to “grep” command with the help of pipe and “ps” is used in Linux to get the running process on system and “grep” is used for search.

- Signal:

Signals come under IPC mechanisms that are used for notification – notification can be process to process – notification can be system to process.

Kill is the command by which one process can send a signal to other.

Syntax: kill <signal_name> <process_id>

Example:

1) Kill SIGINT 1234

2) Kill SIGQUIT 1234

Q19 What is graceful degradation?

In multiprocessor systems, failure of one processor will not halt the system, but only slow it down by sharing the work of the failure system with other systems. This ability to continue providing service is proportional to the surviving hardware is called graceful degradation.

Q20 What is multicore processor?

Hardware has been to place multiple processor cores on the same physical chip, resulting in a multicore processor. Each core maintains its architectural state and thus appears to the operating system to be a separate physical processor.

Q21 What is the Zombie process?

A zombie process is a process that has completed and in the terminated state but has its entry in the process table. It shows that the resources are held by the process and are not free.

Q22 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.

Q23 Explain the concept of Reentrancy?

It is a useful, memory-saving technique for multiprogrammed timesharing systems. A Reentrant Procedure is one in which multiple users can share a single copy of a program during the same period. Reentrancy has 2 key aspects: The program code cannot modify itself, and the local data for each user process must be stored separately. Thus, the permanent part is the code, and the temporary part is the pointer back to the calling program and local variables used by that program. Each execution instance is called activation. It executes the code in the permanent part, but has its own copy of local variables/parameters. The temporary part associated with each activation is the activation record. Generally, the activation record is kept on the stack.

Note: A reentrant procedure can be interrupted and called by an interrupting program, and still execute correctly on returning to the procedure.

Process

Q1 What is a process and process table?

A *process* is an instance of a program in execution. For example a Web Browser is a process, a shell (or command prompt) is a process.

The operating system is responsible for managing all the processes that are running on a computer and allocated each process a certain amount of time to use the processor. In addition, the operating system also allocates various other resources that processes will need such as computer memory or disks. To keep track of the state of all the processes, the operating system maintains a table known as the *process table*. Inside this table, every process is listed along with the resources the processes are using and the current state of the process.

Q2 Explain the different states of a process?

A process can go through following states in its lifecycle:

New: This is the first state when a process is created or just started. It resides in secondary memory.

Ready: This state signifies that the process is ready to be assigned to the processor that is ready to execute.

Running: This state signifies that process has been given to the processor and its instruction are getting executed.

Waiting: This state signifies that a process is waiting for some kind of resource or an event be it a user input or waiting for any file to become available.

Terminated: The process can be terminated normally or abnormally.

Normal termination means that process has done with its complete execution whereas abnormal means that process has been terminated without completing its task.

Note: Any process has to go minimum four states (new->ready->running->terminated).

Q3 What is the Process Control Block(PCB)?

Each process is represented in the operating system by a process control block alsocalled a task control block. It contains many pieces of information associated witha specific process.It simply acts as a repository for any information that may vary from process to process.It contains the following information:

Process state

Program counter

CPU registers

CPU-scheduling information

Memory-management information

Accounting information

I/O status information

Q4 Which is the first process to be created by OS?

Init process is the first process to be created by OS. It provides the environment for the other process to be created later.

Q5 What is starvation and aging?

Starvation:

Starvation is a resource management problem where a process is denied of resource or service for a long time or has been repeatedly denied services.

Aging:

This is a solution to starvation which involves gradually increasing the priority of processes that wait in the system for a long time.

The aging factor must increase the requests priority as time passes and must ensure that a request will eventually be the highest priority request (after it has waited long enough) and gets the chance to execute.

Note: It’s not same as deadlock.

Q6 Difference between mutex and semaphores ? (Adobe)

BASIS FOR COMPARISON	SEMAPHORE	MUTEX
Basic	Semaphore is a signalling mechanism.	Mutex is a locking mechanism.
Existence	Semaphore is an integer variable.	Mutex is an object.
Function	Semaphore allow multiple program threads to access a finite instance of resources.	Mutex allow multiple program thread to access a single resource but not simultaneously.
Ownership	Semaphore value can be changed by any process acquiring or releasing the resource.	Mutex object lock is released only by the process that has acquired the lock on it.

Categorize	Semaphore can be categorized into counting semaphore and binary semaphore.	Mutex is not categorized further.
Operation	Semaphore value is modified using wait() and signal() operation.	Mutex object is locked or unlocked by the process requesting or releasing the resource.
Resources Occupied	If all resources are being used, the process requesting for resource performs wait() operation and block itself till semaphore count become greater than one.	If a mutex object is already locked, the process requesting for resources waits and queued by the system till lock is released.

Q7 What problems are faced by use of semaphores ? (Adobe)

Q8 What is the use of fork and exec system calls?

Fork is a system call by which a new process is created. Exec is also a system call, which is used after a fork by one of the two processes to place the process memory space with a new program.

Q9 Define PThreads.

PThreads refers to the POSIX standard defining an API for thread creation and synchronization. This is a specification for thread behavior, not an implementation.

Q10 What are the requirements that a solution to the critical section problem must satisfy?

The three requirements are

Mutual exclusion

Progress

Bounded waiting

Q11 What is a Thread? What are the differences between process and thread? (Adobe)

A thread is a single sequence stream within a process. Because threads have some of the properties of processes, they are sometimes called *lightweight processes*. Threads are a popular way to improve application through parallelism. For example, in a browser, multiple tabs can be different threads. MS word uses multiple threads, one thread to format the text, other thread to process inputs, etc.

A thread has its own program counter (PC), a register set, and a stack space. Threads are not independent of one other like processes as a result threads share with other threads their code section, data section and OS resources like open files and signals

Q12 Write the code of dining philosopher problem ? (Adobe)

https://www.geeksforgeeks.org/dining-philosopher-problem-using-semaphores/

Q13 What are the different Scheduling Algorithms?

1. First Come First Serve (FCFS) : Simplest scheduling algorithm that schedules according to arrival times of processes.
2. Shortest Job First (SJF): Process which have the shortest burst time are scheduled first.
3. Shortest Remaining Time First (SRTF): It is a preemptive mode of SJF algorithm in which jobs are scheduled according to the shortest remaining time.
4. Round Robin (RR) Scheduling: Each process is assigned a fixed time, in a cyclic way.
5. Priority Based scheduling (Non Preemptive): In this scheduling, processes are scheduled according to their priorities, i.e., the highest priority process is scheduled first. If priorities of two processes match, then scheduling is according to the arrival time.
6. Highest Response Ratio Next (HRRN): In this scheduling, processes with the highest response ratio is scheduled. This algorithm avoids starvation. Response Ratio = (Waiting Time + Burst time) / Burst time
7. Multilevel Queue Scheduling (MLQ): According to the priority of process, processes are placed in the different queues. Generally high priority processes

are placed in the top level queue. Only after completion of processes from the top level queue, lower level queued processes are scheduled.

8. Multilevel Feedback Queue (MLFQ) Scheduling: It allows the process to move in between queues. The idea is to separate processes according to the characteristics of their CPU bursts. If a process uses too much CPU time, it is moved to a lower-priority queue.

Q14 Explain the different sections of a process.

There are mainly four sections in a process. They are as below:

1. Stack: contains local variables, returns address
2. Heap: Dynamically allocated memory via malloc, calloc,realloc
3. Data: contains global and static variables
4. Code or text: contains code, program counter and content of processor's register.

Note: Stack and the Heap section are extendible, that is stack can grow down and the heap can grow up.

Q15 List the different performance metrics for the scheduler.

a)CPU Utilization: Percentage of time that the CPU is doing useful work (I.e. not being idle). 100% is perfect.

b)Wait time: This is the time that a process spends for its turn to get executed.

c)Throughput: The number of processes completed / time unit.

d)Response Time: This is the time elapsed from when a process is submitted until a useful output is obtained.

e)Turnaround Time: This is considered to be the time elapsed from when a process is submitted to when it has completed.

Q16 Explain and differentiate between user level and kernel level thread.

USER LEVEL THREAD	KERNEL LEVEL THREAD
User threads are implemented by users.	kernel threads are implemented by OS.
OS doesn't recognize user level threads.	Kernel threads are recognized by OS.
Implementation of User threads is easy.	Implementation of Kernel thread is complicated.
Context switch time is less.	Context switch time is more.
Context switch requires no hardware support.	Hardware support is needed.
If one user level thread performs a blocking operation then the entire process will be blocked.	If one kernel thread performs a blocking operation then another thread can continue execution.

User level threads are designed as dependent threads.	Kernel level threads are designed as independent threads.
Example : Java thread, POSIX threads.	Example : Window Solaris.

Q17 What is the use of Job Queues, Ready Queue and Device Queues?

As a process enters a system, they are put into a job queue. This queue consists of all jobs in the system. The processes that are residing in main memory and are ready & waiting to execute are kept on a list called ready queue. The list of processes waiting for a particular I/O device is kept in the device queue.

Q18 What are System Calls?

System calls provide the interface between a process and the Operating system. System Calls are also called Monitor call or Operating-system function call. When a system call is executed, it is treated as by the hardware as software interrupt. Control passes through the interrupt vector toa service routine in the operating system, and the mode bit is set to monitor mode.

Q19 What is Spooling?

Spooling means Simultaneous Peripheral Operations OnLine. It is a high-speed device like a disk is interposed between a running program and a low–speed device involved with the program in input/output. It dissociates a running program from the slow operation of devices like printers

Q20 What are the possible threads a thread can have?

1. Ready
2. Standby
3. Running
4. Waiting
5. Transition
6. Terminated

Q21 What is process spawning?

When the OS at the explicit request of another process creates a process, this action is called process spawning.

Q22 What is a Dispatcher?
The dispatcher is the module that gives control of the CPU to the process selected by the short-term scheduler. This function involves:

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

Q23 Define Busy Waiting and Spinlock ?
When a process is in its critical section, any other process that tries to enter its critical section must loop continuously in the entry code. This is called as busy waiting and this type of semaphore is also called a spinlock, because the process while waiting for the lock

Q24 Difference between preemptive and non preemptive scheduling ? (Adobe)

PARAMETER	PREEMPTIVE SCHEDULING	NON-PREEMPTIVE SCHEDULING
Basic	In this resource(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 can not be interrupted until it terminates itself or its time is up.
Starvation	If a process having high priority frequently arrives in the ready queue, low	If a process with long burst time is running CPU, then later coming processes with less CPU burst time may starve.

	priority processes may starve.	
Overhead	It has overheads of scheduling the processes.	It does not have overheads.
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.

Q25 Write code for producer-consumer problem ? (Adobe)

https://www.geeksforgeeks.org/producer-consumer-problem-using-semaphores-set-1/

Q26 Can two threads in a process communicate? Is communication possible between two threads of two different processes?

https://www.geeksforgeeks.org/inter-thread-communication-java/
https://www.geeksforgeeks.org/inter-process-communication/

Q27 What is caching? (Amazon)

In computing, a cache is a high-speed data storage layer which stores a subset of data, typically transient in nature, so that future requests for that data are served up faster than is possible by accessing the data's primary storage location. Caching allows you to efficiently reuse previously retrieved or computed data.

How does Caching work?

The data in a cache is generally stored in fast access hardware such as RAM (Random-access memory) and may also be used in correlation with a software component. A cache's primary purpose is to increase data retrieval performance by reducing the need to access the underlying slower storage layer. Trading off capacity for speed, a cache typically stores a subset of data transiently, in contrast to databases whose data is usually complete and durable.

Q28 What is race condition ? (Visa)

A race condition is a situation that may occur inside a critical section. This happens when the result of multiple thread execution in critical section differs according to the order in which the threads execute.

Race conditions in critical sections can be avoided if the critical section is treated as an atomic instruction. Also, proper thread synchronization using locks or atomic variables can prevent race conditions.

Q29 Define mutual exclusion.

Mutual exclusion refers to the requirement of ensuring that no two process or threads are in their critical section at the same time. i.e. If process Pi is executing in its critical section, then no other processes can be executing in their critical sections.

Q30 Differentiate Long Term Scheduler and Short Term Scheduler ?
The long-term scheduler or job scheduler selects processes from the job pool and loads them into memory for execution.The short-term scheduler or CPU scheduler selects from among the processes that are ready to execute, and allocates the CPU to one of them.

Q31 What is meant by context switch?
Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. This task is known as context switch.

Deadlocks

Q1 What is Deadlock? How to avoid and prevent? (Adobe)
Deadlock is a situation when two or more processes wait for each other to finish and none of them ever finish. Consider an example when two trains are coming toward each other on same track and there is only one track, none of the trains can move once they are in front of each other. Similar situation occurs in operating systems when there are two or more processes hold some resources and wait for resources held by other(s).

There are three ways to handle deadlock

1) Deadlock prevention or avoidance: The idea is to not let the system into deadlock state.

2) Deadlock detection and recovery: Let deadlock occur, then do preemption to handle it once occurred.

3) Ignore the problem all together: If deadlock is very rare, then let it happen and reboot the system. This is the approach that both Windows and UNIX take.

Q2 What are the necessary conditions for deadlock?

Mutual Exclusion: There is a resource that cannot be shared.

Hold and Wait: A process is holding at least one resource and waiting for another resource which is with some other process.

No Preemption: The operating system is not allowed to take a resource back from a process until process gives it back.

Circular Wait: A set of processes are waiting for each other in circular form.

Q3. Explain deadlock avoidance using banker's algorithm in details.

https://www.studytonight.com/operating-system/bankers-algorithm

https://www.geeksforgeeks.org/bankers-algorithm-in-operating-system-2/

Q4 When is a system in safe state?

The set of dispatchable processes is in a safe state if there exists at least one temporal order in which all processes can be run to completion without resulting in a deadlock.

Memory Management

Q1 How are pages swapped of the memory ?(Adobe)

Q2 What is Belady's Anomaly?

Bélády's anomaly is an anomaly with some page replacement policies where increasing the number of page frames results in an increase in the number of page faults. It occurs with First in First Out page replacement is used. See the wiki page for an example and more details.

Q3 Differentiate Logical From Physical Address Space.?

Logical address refers to the address that is generated by the CPU. On the other hand, physical address refers to the address that is seen by the memory unit.

Q4 What is segmentation?

Segmentation is a memory management technique in which each job is divided into several segments of different sizes, one for each module that contains pieces that perform related functions. Each segment is actually a different logical address space of the program.

When a process is to be executed, its corresponding segmentation are loaded into non-contiguous memory though every segment is loaded into a contiguous block of available memory.

Segmentation memory management works very similar to paging but here segments are of variable-length where as in paging pages are of fixed size.

A program segment contains the program's main function, utility functions, data structures, and so on. The operating system maintains a segment map table for every process and a list of free memory blocks along with segment numbers, their size and corresponding memory locations in main memory. For each segment, the table stores the starting address of the segment and the length of the segment. A reference to a memory location includes a value that identifies a segment and an offset.

Q5 There is a game which requires 4GB RAM memory and i have a system with 2GB RAM. Then which concept the system will use to run the game. (Amazon)

Q6 What Is The Translation Lookaside Buffer (tlb)?

In a cached system, the base addresses of the last few referenced pages is maintained in registers called the TLB that aids in faster lookup. TLB contains those page-table entries that have been most recently used. Normally, each virtual memory reference causes 2 physical

memory accesses- one to fetch appropriate page-table entry, and one to fetch the desired data. Using TLB in-between, this is reduced to just one physical memory access in cases of TLB-hit.

Q7 Define effective access time.

Let p be the probability of a page fault (0£p£1). The value of p is expected to be close to 0; that is, there will be only a few page faults. The effective access time is Effective access time = (1-p) * ma + p* page fault time. ma : memory-access time

Q8 Differentiate between Global and Local page replacement algorithms.

The number of frames allocated to a process can also dynamically change depending on whether you have used **global replacement** or **local replacement** for replacing pages in case of a page fault.

1. **Local replacement:** When a process needs a page which is not in the memory, it can bring in the new page and allocate it a frame from its own set of allocated frames only.
    - **Advantage:** The pages in memory for a particular process and the page fault ratio is affected by the paging behavior of only that process.
    - **Disadvantage:** A low priority process may hinder a high priority process by not making its frames available to the high priority process.
2. **Global replacement:** When a process needs a page which is not in the memory, it can bring in the new page and allocate it a frame from the set of all frames, even if that frame is currently allocated to some other process; that is, one process can take a frame from another.
    - **Advantage:** Does not hinder the performance of processes and hence results in greater system throughput.
    - **Disadvantage:** The page fault ratio of a process can not be solely controlled by the process itself. The pages in memory for a process depends on the paging behavior of other processes as well.

Q9 Differentiate a page from a segment.

In segmentation, the address space is typically divided into a preset number of segments like data segment (read/write), code segment(read-only), stack(read/write) etc. And the programs are divided into these segments accordingly. Logical addresses are represented as tuple . While with paging, the address space is divided into a sequence of fixed size units called "pages". And logical addresses take the form of a tuple .

Q10 What is address binding?

The process of associating program instructions and data to physical memory addresses is called address binding, or relocation.

Virtual Management

Q1 Explain Thrashing in Operating Systems.(Adobe)

Thrashing is a situation when the performance of a computer degrades or collapses. Thrashing occurs when a system spends more time processing page faults than executing transactions. While processing page faults is necessary to in order to appreciate the benefits of virtual memory, thrashing has a negative affect on the system. As the page fault rate increases, more transactions need processing from the paging device. The queue at the paging device increases, resulting in increased service time for a page fault

Now the effects of thrashing and also the extent to which thrashing occurs will be decided by the type of page replacement policy.

1. Global Page Replacement: The paging algorithm is applied to all the pages of the memory regardless of which process "owns" them. A page fault in one process may cause a replacement from any process in memory. Thus, the size of a partition may vary randomly.

2. Local Page Replacement: The memory is divided into partitions of a predetermined size for each process and the paging algorithm is applied independently for each region. A process can only use pages in its partition.

What happens after Thrashing starts?

If global page replacement is used, situations worsens very quickly. CPU thinks that CPU utilization is decreasing, so it tries to increase the degree of multiprogramming. Hence bringing more processes inside memory, which in effect increases the thrashing and brings down CPU utilization further down. The CPU notices that utilization is going further down, so it increases the degree of multiprogramming further and the cycle continues.

The solution can be local page replacement where a process can only be allocated pages in its own region in memory. If the swaps of a process increase also, the overall CPU utilization does not decrease much. If other transactions have enough page frames in the partitions they occupy, they will continue to be processed efficiently.

But local page replacement has its own disadvantages which you can now explore further.

Q2 What is virtual memory ? How is it implemented ? (Adobe)

Virtual memory creates an illusion that each user has one or more contiguous address spaces, each beginning at address zero. The sizes of such virtual address spaces is generally very high.

The idea of virtual memory is to use disk space to extend the RAM. Running processes don’t need to care whether the memory is from RAM or disk. The illusion of such a large amount of memory is created by subdividing the virtual memory into smaller pieces, which can be loaded into physical memory whenever they are needed by a process.

Q3 What are page replacement algorithms ?

1 FIFO (Belady's anomaly)

2 Optimal page replacement

3 LRU

Q4 What is demand paging?

Demand paging is referred to when not all of a process's pages are in the RAM, then the OS brings the missing(and required) pages from the disk into the RAM.

Q5 What is fragmentation?

Fragmentation is memory wasted. It can be internal if we are dealing with systems that have fixed-sized allocation units, or external if we are dealing with systems that have variable-sized allocation units

Q6 Explain difference between internal external fragmentations in detail.

S.NO	INTERNAL FRAGMENTATION	EXTERNAL FRAGMENTATION
1.	In internal fragmentation fixed-sized memory, blocks square measure appointed to process.	In external fragmentation, variable-sized memory blocks square measure appointed to method.

2.	Internal fragmentation happens when the method or process is larger than the memory.	External fragmentation happens when the method or process is removed.
3.	The solution of internal fragmentation is best-fit block.	Solution of external fragmentation is compaction, paging and segmentation.
4.	Internal fragmentation occurs when memory is divided into fixed sized partitions.	External fragmentation occurs when memory is divided into variable size partitions based on the size of processes.
5.	The difference between memory allocated and required space or memory is called Internal fragmentation.	The unused spaces formed between non-contiguous memory fragments are too small to serve a new process, is called External fragmentation .

Q7 What are Overlays ?

Overlays are used to enable a process to be larger than the amount of memory allocated to it. The basic idea of this is that only instructions and data that are needed at any given time are kept in memory.

Q8 Define Lazy swapper ?

Rather than swapping the entire process into main memory, a lazy swapper is used. A lazy swapper never swaps a page into memory unless that page will be needed.

Q9 What is Pure Demand paging ?

When starting execution of a process with no pages in memory, the operating system sets the instruction pointer to the first instruction of the process, which is on a non-memory resident page, the process immediately faults for the page.After this page is brought into memory, the process continues to execute, faulting as necessary until every page that it needs is in memory.At that point, it can execute with no more faults.This schema is pure demand paging.

Q10 Define Dynamic Loading.

To obtain better memory-space utilization dynamic loading is used. With dynamic loading, a routine is not loaded until it is called. All routines are kept on disk in a relocatable load format. The main program is loaded into memory and executed. If the routine needs another routine, the calling routine checks whether the routine has been loaded. If not, the relocatable linking loader is called to load the desired program into Memory.

Q11 Define Dynamic Linking.

Dynamic linking is similar to dynamic loading, rather that loading being postponed until execution time, linking is postponed. This feature is usually used with system libraries, such as language subroutine libraries. A stub is included in the image for each library-routine reference. The stub is a small piece of code that indicates how to locate the appropriate memory-resident library routine, or how to load the library if The routine is not already present.

Q12 What do you mean by Compaction?

Compaction is a solution to external fragmentation. The memory contents are shuffled to place all free memory together in one large block. It is possible only if relocation is dynamic, and is done at execution time.

Q13 What is cycle stealing?

We encounter cycle stealing in the context of Direct Memory Access (DMA). Either the DMA controller can use the data bus when the CPU does not need it, or it may force the CPU to temporarily suspend operation. The latter technique is called cycle stealing. Note that cycle stealing can be done only at specific break points in an instruction cycle.

Q14 What is time-stamping?

It is a technique proposed by Lamport, used to order events in a distributed system without the use of clocks. This scheme is intended to order events consisting of the transmission of messages. Each system 'i' in the network maintains a counter Ci. Every time a system transmits a message, it increments its counter by 1 and attaches the time-stamp Ti to the message. When a message is received, the receiving system 'j' sets its counter Cj to 1 more than the maximum of its current value and the incoming time-stamp Ti. At each site, the ordering of messages is determined by the following rules: For messages x from site i and y from site j, x precedes y if one of the following conditions holds....(a) if Ti<Tj or (b) if Ti=Tj and i<j.

Disk Scheduling

Q1 What are Disk Scheduling Algorithms ?

1. FCFS: FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue.
2. SSTF: In SSTF (Shortest Seek Time First), requests having shortest seek time are executed first. So, the seek time of every request is calculated in advance in a queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk arm will get executed first.
3. SCAN: In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of the disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works like an elevator and hence also known as elevator algorithm.
4. CSCAN: In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.
5. LOOK: It is similar to the SCAN disk scheduling algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only. Thus it prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.
6. CLOOK: As LOOK is similar to SCAN algorithm, in a similar way, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm in spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end’s last request. Thus, it also prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

Q2 What is the use of Boot block ?

A program at some fixed location on a hard disk, floppy disk or other media, which is loaded when the computer is turned on or rebooted and which controls the next phase of loading

the actual operating system. The loading and execution of the boot block is usually controlled by firmware in ROM or PROM.

Q3 What is sector sparing ?

When a hard drive encounters a sector that it cannot write to, it will write to a sector that was previously allocated as a spare sector. However, if the previously allocated spare sector is unavailable due to a previous allocation, this can lead to disk failure.

Q4 What is meant by RAID ?

RAID stands for Redundant Array of Independent Disks. It is used to store the same data redundantly to improve the overall performance.

Following are the different RAID levels:

RAID 0 - Stripped Disk Array without fault tolerance

RAID 1 - Mirroring and duplexing

RAID 2 - Memory-style error-correcting codes

RAID 3 - Bit-interleaved Parity

RAID 4 - Block-interleaved Parity

RAID 5 - Block-interleaved distributed Parity

RAID 6 - P+Q Redundancy

Q5 Define Seek Time and Latency Time.
The time taken by the head to move to the appropriate cylinder or track is called seek time. Once the head is at right track, it must wait until the desired block rotates under the read-write head. This delay is latency time.

Q6 What are the Allocation Methods of a Disk Space?
Three major methods of allocating disk space which are widely in use are
Contiguous allocation
Linked allocation
Indexed allocation

Q7 What is Direct Access Method?

Direct Access method is based on a disk model of a file, such that it is viewed as a numbered sequence of blocks or records. It allows arbitrary blocks to be read or written. Direct access is advantageous when accessing large amounts of information.

Q8 What is meant by polling?

Polling is the process where the computer waits for an external device to check for its readiness. The computer does not do anything else than checking the status of the device .Polling is often used with low-level hardware. Example: when a printer connected via a parallel port the computer waits until the next character has been received by the printer. These processes can be as minute as only reading 1 Byte. Polling is the continuous (or frequent) checking by a controlling device or process of other devices, processes, queues, etc.

Q9 What is storage area networks?

A storage area network (SAN) is a dedicated network that provides access to consolidated, block level data storage. SANs are primarily used to make storage devices, such as disk arrays, tape libraries, and optical jukeboxes, accessible to servers so that the devices appear like locally attached devices to the operating system.

File Management

Q1 What is a File?
A file is a named collection of related information that is recorded on secondary storage. A file contains either programs or data. A file has certain “structure” based on its type.
File attributes: Name, identifier, type, size, location, protection, time, date
File operations: creation, reading, writing, repositioning, deleting,truncating, appending, renaming
File types: executable, object, library, source code etc.

Q2 What is the Access Control List (ACL)?
Access-list (ACL) is a set of rules defined for controlling the network traffic and reducing network attack. ACLs are used to filter traffic based on the set of rules defined for the incoming or out going of the network.

ACL features –

1. The set of rules defined are matched serial wise i.e matching starts with the first line, then 2nd, then 3rd and so on.
2. The packets are matched only until it matches the rule. Once a rule is matched then no further comparison takes place and that rule will be performed.
3. There is an implicit deny at the end of every ACL, i.e., if no condition or rule matches then the packet will be discarded.

Q3 What are the Functions of Virtual File System(VFS)?
It has two functions, It separates file-system-generic operations from their implementation defining a clean VFS interface. It allows transparent access to different types offile systems Mounted locally. VFS is based on a file representation structure, called a vnode. It contains a numerical value for a network-wide unique file .The kernel maintains one vnode structure for each active file or directory.

Q4 What are the different accessing methods of a file?
The different types of accessing a file are: Sequential access: Information in the file is accessed sequentially Direct access: Information in the file can be accessed without any particular order. Other access methods: Creating index for the file, indexed sequential access method (ISAM) etc.

Q5 What is a Directory?
The device directory or simply known as directory records information-such as name, location, size, and type for all files on that particular partition. The directory can be viewed as a symbol table that translates file names into their directory entries.

Q6 What are the most common schemes for defining the logical structure of a directory?
The most common schemes for defining the logical structure of directory Single-Level Directory Two-level Directory Tree-Structured Directories Acyclic-Graph Directories General Graph Directory

Q7 Define UFD and MFD.
In the two-level directory structure, each user has her own user file directory (UFD). Each UFD has a similar structure, but lists only the files of a single user. When a job starts the system's master file directory (MFD) is searched. The MFD is indexed by the user name or account number, and each entry points to the UFD for that user.

Q8 What are the various layers of a file system?
The file system is composed of many different levels. Each level in the design uses the feature of the lower levels to create new features for use by higher levels.

- Application programs
- Logical file system
- File-organization module
- Basic file system
- I/O control
- Devices

Q9 What is inode?

Inode is a data structure which holds all the attributes of a file. It is also called index number.

Some of the attributes of file are:

- File type
- Permission
- File size
- Time when last it is modified

Note: Please while reading thread concept, try to relate all this with process, thread will have its own, all those attributes as that of a process like its id(tid), scheduling parameter/policy etc. and concept like context switching, different section etc. Thus try to relate all thread's concept with that of a process.