

OPERATING SYSTEM STRIVER CHEET SHEET

1. What is the main purpose of an operating system? Discuss different types?

An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. So it manages the computer's memory, processes, devices, files, and security aspects of the system. It also allows us to communicate with the computer without knowing how to speak the computer's language. Without an operating system, a computer is not useful.

Types of operating system:

- Batch OS
- Distributed OS
- Multitasking OS
- Network OS
- Real-Time OS
- Mobile OS

2. What is a socket, kernel and monolithic kernel ?

Socket:

A socket is defined as an endpoint for communication, A pair of processes communicating over a network employ a pair of sockets ,one for each process. A socket is identified by an IP address concatenated with a port number.

The server waits for incoming client requests by listening to a specified port. Once a request is received, the server accepts a connection from the client socket to complete the connection.

Kernel is the central core component of an operating system that manages operations of computer and hardware. Kernel Establishes communication between user level application and hardware. Manages memory and CPU time. Decides state of incoming processes. Controls Disk, Memory, Task Management

Monolithic Kernel (provides good performance but lots of lines of code)

It is one of the types of kernel where all operating system services operate in kernel space. It has dependencies between system components. It has huge lines of code which is complex.

Example : Unix, Linux, Open VMS, XTS-400 etc.

3. Difference between process and program and thread? Different types of process.

Process:

Process is an instance of an executing program. For example, we write our computer programs in a text file and when we execute this program, it becomes a process which performs all the tasks mentioned in the program.

Program:

Program is a set of instructions to perform a certain task. Eg: chrome.exe, notepad.exe

Thread:

Thread is a path of execution within a process. A thread is also known as a lightweight process. The idea is to achieve parallelism by dividing a process into multiple threads. For example, Word processor uses multiple threads: one thread to format the text, another thread to process inputs.

4. Define virtual memory, thrashing, threads.

Virtual Memory:

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.

The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory serves two purposes. First, it allows us to extend the use of physical memory by using a disk. Second, it allows us to have memory protection, because each virtual address is translated to a physical address.

Thrashing:

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. High degree of multiprogramming (if number of processes keeps on increasing in the memory), lack of frames (if a process is allocated too few frames, then there will be too many and too frequent page faults) causes Thrashing.

Threads:

A thread is a single sequential flow of execution of tasks of a process so it is also known as thread of execution or thread of control.

5. What is RAID ? Different types.

RAID, or "Redundant Arrays of Independent Disks" is a technique which makes use of a combination of multiple disks instead of using a single disk for increased performance, data redundancy or both. Data redundancy, although taking up extra space, adds to disk reliability. This means, in case of disk failure, if the same data is also backed up onto another disk, we can retrieve the data and go on with the operation.

6. What is a deadlock? Different conditions to achieve a deadlock.

A Deadlock is a situation where each of the computer processes waits for a resource which is being assigned to some other process. In this situation, none of the processes gets executed since the resource it needs is held by some other process which is also waiting for some other resource to be released.

How deadlock is achieved: Deadlock happens when Mutual exclusion, hold and wait, No preemption and circular wait occurs simultaneously.

Necessary Conditions for deadlock:

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

7. What is fragmentation? Types of fragmentation.

Fragmentation: An unwanted problem in the operating system in which the processes are loaded and unloaded from memory, and free memory space is fragmented. Processes can't be assigned to memory blocks due to their small size, and the memory blocks stay unused. It is also necessary to understand that as programs are loaded and deleted from memory, they generate free space or a hole in the memory. These small blocks cannot be allotted to new arriving processes, resulting in inefficient memory use.

The conditions of fragmentation depend on the memory allocation system. As the process is loaded and unloaded from memory, these areas are fragmented into small pieces of memory that cannot be allocated to incoming processes. It is called fragmentation.

Types of fragmentation:

1. Internal
2. External

8. What is spooling ?

SPOOL is 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 system. In spooling, there is no interaction between the I/O devices and the CPU. That means there is no need for the CPU to wait for the I/O operations to take place. Such operations take a long time to finish executing, so the CPU will not wait for them to finish.

The biggest example of Spooling is printing. The documents which are to be printed are stored in the SPOOL and then added to the queue for printing. During this time, many processes can perform their operations and use the CPU without waiting while the printer executes the printing process on the documents one-by-one.

9. What is semaphore and mutex (Differences might be asked)? Define Binary semaphore.

10. Belady's Anomaly

Bélády's anomaly is the name given to the phenomenon where increasing the number of page frames results in an increase in the number of page faults for a given memory access pattern.

Solution to fix Belady's Anomaly:

Implementing alternative page replacement algo helps eliminate Belady's Anomaly.. Use of stack based algorithms, such as Optimal Page Replacement Algorithm and Least Recently Used (LRU) algorithm, can eliminate the issue of increased page faults as these algorithms assign priority to pages

11. Starving and Aging in OS

Starving/Starvation(also called Lived lock): Starvation is the problem that occurs when low priority processes get jammed for an unspecified time as the high priority processes keep executing. So starvation happens if a method is indefinitely delayed.

Solution to Starvation : Ageing is a technique of gradually increasing the priority of processes that wait in the system for a long time.

12. Why does trashing occur?

High degree of multiprogramming(if number of processes keeps on increasing in the memory) , lack of frames(if a process is allocated too few frames, then there will be too many and too frequent page faults.) causes Thrashing.

13. What is paging and why do we need it?

Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. This scheme permits the physical address space of a process to be non – contiguous.

Paging is used for faster access to data. When a program needs a page, it is available in the main memory(RAM) as the OS copies a certain number of pages from your storage device to main memory. Paging allows the physical address space of a process to be noncontiguous.

14. Demand Paging, Segmentation

Demand paging is a method of virtual memory management which is based on the principle that pages should only be brought into memory if the executing process demands them. This is often referred to as lazy evaluation as only those pages demanded by the process are swapped from secondary storage to main memory.

So demand paging works opposite to the principle of loading all pages immediately.

Segmentation is a memory management technique in which the memory is divided into the variable size parts. Each part is known as a segment which can be allocated to a process.

The details about each segment are stored in a table called a segment table. Segment table is stored in one (or many) of the segments.

Segment table contains mainly two information about segment:

Base: It is the base address of the segment

Limit: It is the length of the segment.

15. Real Time Operating System, types of RTOS.

A real-time operating system (RTOS) is a special-purpose operating system used in computers that has strict time constraints for any job to be performed and is intended to serve real time applications that possess data as it comes in , typically without buffer delays.

Types of RTOS:

- Hard RTOS
- Firm RTOS
- Soft RTOS

16. Dynamic Binding

Static binding happens when the code is compiled, while dynamic bind happens when the code is executed at run time.

Static Binding:

When a compiler acknowledges all the information required to call a function or all the values of the variables during compile time, it is called “static binding”. As all the required information is known before runtime, it increases the program efficiency and it also enhances the speed of execution of a program. Static Binding makes a program very efficient, but it declines the program flexibility, as ‘values of variable’ and ‘function calling’ are predefined in the program.

Static binding is implemented in a program at the time of coding. Overloading a function or an operator is the example of compile time polymorphism i.e. static binding.

Dynamic Binding Calling a function or assigning a value to a variable, at run-time is called “Dynamic Binding”. Dynamic binding can be associated with run time ‘polymorphism’ and ‘inheritance’ in OOP. Dynamic binding makes the execution of a program flexible as it can decide what value should be assigned to the variable and which function should be called, at the time of program execution. But as this information is provided at run time it makes the execution slower as compared to static binding.

17. SRTF Scheduling

SRTF Scheduling is a preemptive version of SJF scheduling. In SRTF, the execution of the process can be stopped after a certain amount of time. At the arrival of every process, the short term scheduler schedules the process with the least remaining burst time among the list of available processes and the running process.

18. LRTF Scheduling

This is a preemptive version of Longest Job First (LJF) scheduling algorithm. In this scheduling algorithm, we find the process with the maximum remaining time and then process it. We check for the maximum remaining time after some interval of time(say 1 unit each) to check if another process having more Burst Time arrived up to that time.

19. Priority Scheduling

Priority Scheduling is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority.

The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority depends upon memory requirements, time requirements, etc.

20. Round Robin Scheduling

In Round-robin scheduling, 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. Widely used preemptive scheduling method in traditional OS. All the jobs get a fair allocation of CPU. Cons include : Finding a correct time quantum is a quite difficult task in this system, Round-robin scheduling doesn't give special priority to more important tasks.

21. Banker's Algorithm

It is a banker algorithm used to avoid deadlock and allocate resources safely to each process in the computer system. The 'S-State' examines all possible tests or activities before deciding whether the allocation should be allowed to each process. It also helps the operating system to successfully share the resources between all the processes. The banker's algorithm is named because it checks whether a person should be sanctioned a loan amount or not to help the bank system safely simulate allocation resources.

22. Explain Cache

Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.

23. Producer Consumer Problem

About Producer-Consumer problem: The Producer-Consumer problem is a classic problem that is used for multi-process synchronisation i.e. synchronisation between more than one processes.

The job of the Producer is to generate the data, put it into the buffer, and again start generating data. While the job of the Consumer is to consume the data from the buffer.

What's the problem here?

The following are the problems that might occur in the Producer-Consumer:

The producer should produce data only when the buffer is not full. If the buffer is full, then the producer shouldn't be allowed to put any data into the buffer.

The consumer should consume data only when the buffer is not empty. If the buffer is empty, then the consumer shouldn't be allowed to take any data from the buffer.

The producer and consumer should not access the buffer at the same time.

We can solve this problem by using semaphores.

ADDITIONAL QUESTIONS

Difference between main memory and secondary memory.

Primary memory	Secondary memory
Primary memory is temporary.	Secondary memory is permanent.
Primary memory is directly accessible by Processor/CPU.	Secondary memory is not directly accessible by the CPU.
Nature of Parts of Primary memory varies, RAM- volatile in nature. ROM- Non-volatile.	It's always Non-volatile in nature.
Primary memory devices are more expensive than secondary storage devices.	Secondary memory devices are less expensive when compared to primary memory devices.
The memory devices used for primary memory are semiconductor memories.	The secondary memory devices are magnetic and optical memories.
Primary memory is also known as Main memory or Internal memory.	Secondary memory is also known as External memory or Auxiliary memory.
Examples: RAM, ROM, Cache memory, PROM, EPROM, Registers, etc.	Examples: Hard Disk, Floppy Disk, Magnetic Tapes, etc.

FCFS Scheduling

SJF Scheduling

Diff between direct mapping and associative mapping

Direct-mapping	Associative Mapping	Set-Associative Mapping
Needs only one comparison because of using direct formula to get the effective cache address.	Needs comparison with all tag bits, i.e., the cache control logic must examine every block's tag for a match at the same time in order to determine that a block is in the cache/not.	Needs comparisons equal to number of blocks per set as the set can contain more than 1 blocks.
Main Memory Address is divided into 3 fields : TAG, BLOCK & WORD. The BLOCK & WORD together make an index. The least significant WORD bits identify a unique word within a block of main memory, the BLOCK bits specify one of the blocks and the Tag bits are the most significant bits.	Main Memory Address is divided into 1 fields : TAG & WORD.	Main Memory Address is divided into 3 fields : TAG, SET & WORD.
There is one possible location in the cache organization for each block from main memory because we have a fixed formula.	The mapping of the main memory block can be done with any of the cache block.	The mapping of the main memory block can be done with a particular cache block of any direct-mapped cache.
If the processor need to access same memory location from 2 different main memory pages frequently, cache hit ratio decreases.	If the processor need to access same memory location from 2 different main memory pages frequently, cache hit ratio has no effect.	In case of frequently accessing two different pages of the main memory if reduced, the cache hit ratio reduces.
Search time is less here because there is one possible location in the cache organization for each block from main memory.	Search time is more as the cache control logic examines every block's tag for a match.	Search time increases with number of blocks per set.
The index is given by the number of blocks in cache.	The index is zero for associative mapping.	The index is given by the number of sets in cache.
It has least number of tag bits.	It has the greatest number of tag bits.	It has less tags bits than associative mapping and more tag bits than direct mapping.

Diff between multitasking and multiprocessing

Difference between Multiprogramming and Multi-tasking

Sr.no	Multiprogramming	Multi-tasking
1.	It includes the single CPU to execute the program.	It uses multiple tasks for the task allocation.
2.	Concept of Context Switching is used.	Concept of Context Switching and Time Sharing is used.
3.	In multiprogrammed system, the operating system simply switches to, and executes, another job when current job needs to wait.	The processor is typically used in time sharing mode. Switching happens when either allowed time expires or where there other reason for current process needs to wait (example process needs to do IO).
4.	Multi-programming increases CPU utilization by organizing jobs .	In multi-tasking also increases CPU utilization, it also increases responsiveness.
5.	The idea is to reduce the CPU idle time for as long as possible.	The idea is to further extend the CPU Utilization concept by increasing responsiveness Time Sharing.
6.	It uses job scheduling algorithms so that more than one program can run at the same time.	Time sharing mechanism is used so that multiple tasks can run at the same time.
7.	In community edition, personalized shopping experiences is not created.	Promotions, personalized shopping experiences can be displayed in enterprise edition products.
8.	Execution of process takes more time.	Execution of process takes less time.