

Operating System

Question 1: What is an operating system?

Answer: An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

Question 2: What are important functions of an operating system?

Answer:

- Memory Management (File System)
- Process Management (Scheduling)
- Device Management (Drivers)
- Security (Password access and restricted access)
- Error Detection (Handling Deadlocks)

Question 3: Explain different types of operating systems.

Answer:

1. Batch operating system
 - User prepares jobs offline and run it in system at one go
 - Lack of interaction between the user and the job.
 - CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
 - Difficult to provide the desired priority.
2. Multiprogramming operating system
 - A computer running more than one program at a time
3. Multiprocessing operating system
 - A computer using more than one CPU at a time.
4. Multitasking operating system
 - Tasks sharing a common resource (like 1 CPU).

Question 4: What is multithreading?

Answer: Multithreading is the ability of a central processing unit to provide multiple threads of execution concurrently, supported by the operating system.

Question 5: Disadvantages of Multithreading.

Answer:

1. Can lead to deadlock
2. Difficult to debug

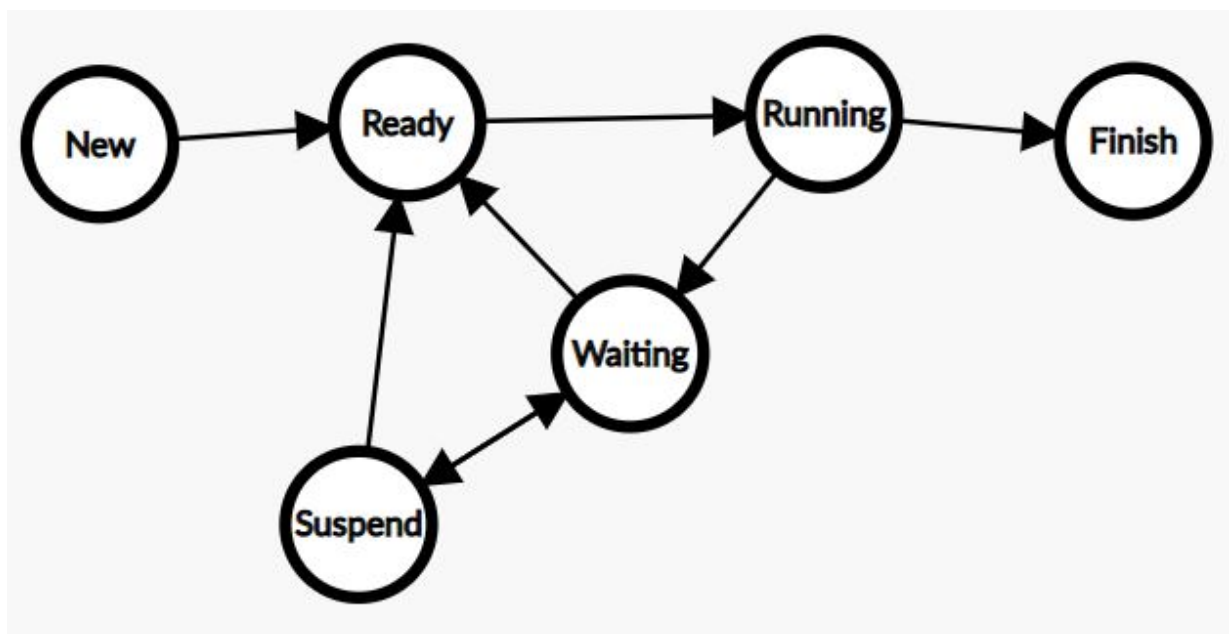
Question 6: Process vs Thread.

Answer:

- A process have its own code, data, heap and stack. Stack is required to manage function calls. Each thread has its own stack.
- All threads in a process share code, data and heap of the process.
- Multiple threads can be running simultaneously on multiprocessor system.
- Threads in a same process share same address space.

Question 7: What are different process states?

Answer:



- New and Suspend are in secondary memory
- Ready and waiting are in primary memory
- Running is in processor

Question 8: What is a process control block?

Answer: A process control block (PCB) is a data structure used by computer operating systems to store all the information about a process. It is also known as a process descriptor. When a process is created (initialized or installed), the operating system creates a corresponding process control block.

Its content vary from OS to OS, but few of the important ones are

- Process ID
- Process State
- CPU Register
- Accounts Information
- IO information
- CPU Scheduling Information
- Memory Information

Question 9: What is Process Scheduler.

Answer: Scheduler is a program which is responsible for loading, deleting and transferring processes between states in operating system.

There are 3 types of operating system

- **Long Term Scheduler** - It brings the new process to the 'Ready State'. It controls Degree of Multi-programming, i.e., number of process present in ready state at any point of time. It is important that the long-term scheduler make a careful selection of both IO and CPU bound process.
- **Short Term Scheduler** - It is responsible for selecting one process from ready state for scheduling it on the running state.
- **Medium Term Scheduler** - It is responsible for suspending and resuming the process. It mainly does swapping (moving processes from main memory to disk and vice versa).

Question 10: What is a dispatcher?

Answer: Dispatcher is responsible for loading the process selected by Short-term scheduler on the CPU (Ready to Running State) Context switching is done by dispatcher only.

A dispatcher does the following:

- Switching context.
- Switching to user mode.
- Jumping to the proper location in the newly loaded program.

Question 11: What is Job Queue and Ready Queue?

Answer:

- **Ready Queue:** All processes in ready queue.
- **Job Queue:** All programs which needs to be loaded on the primary memory to become ready/waiting.

Question 12: Advantages and Disadvantages of different scheduling algorithm.

Answer:

1. First Come First Serve:

- **Advantages**
 - Simple Algorithm.
 - Every process will run, no starvation.
- **Disadvantages:**
 - No option for pre-emption of a process.
 - Long waiting time.

2. Shortest Job First:

- **Advantages**
 - Increased throughput.
 - Shorter process complete quickly.
- **Disadvantages:**
 - Time taken by process must be known initially.
 - Longer processes may lead to starvation

3. Shortest Remaining Time First:

- **Advantages**
 - Increased throughput.
 - Pre-emption is allowed.
- **Disadvantages:**
 - Time taken by process must be known initially.

4. Round Robin:

- **Advantages**
 - Equal priority among processes.
 - Starvation does not occur.
- **Disadvantages:**
 - Big Quantum can lead to execution as FCFS.
 - If Quantum is short then context switching happens more often thus leads to inefficiency of processer.

Question 13: What is context switching?

Answer: Context Switching involves storing the context or state of a process so that it can be reloaded when required and execution can be resumed from the same point as earlier.

Question 14: What is race condition?

Answer: A race condition is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time, but because of

the nature of the device or system, the operations must be done in the proper sequence to be done correctly.

Question 15: What is critical section?

Answer: Snippets which causes race condition are known as critical section.

Question 16: What is consumer-producer problem?

Answer:

Consider 2 snippets

```
int itemCount = 0;
```

```
1. procedure producer() {  
2.     while (true) {  
3.         item = produceItem();  
4.         if (itemCount == BUFFER_SIZE) {  
5.             sleep();  
6.         }  
7.         putItemIntoBuffer(item);  
8.         itemCount = itemCount + 1;  
9.         if (itemCount == 1) {  
10.            wakeup(consumer);  
11.        }  
12.    }  
13. }
```

```
1. procedure consumer() {  
2.     while (true) {  
3.         if (itemCount == 0) {  
4.             sleep();  
5.         }  
6.         item = removeItemFromBuffer();  
7.         itemCount = itemCount - 1;  
8.         if (itemCount == BUFFER_SIZE - 1) {  
9.             wakeup(producer);  
10.        }  
11.        consumeItem(item);  
12.    }  
13. }
```

What can happen is that there comes a higher priority process after line 7 of producer, thus not increasing itemCount but item is there in buffer. So if consumer is called, it will see it as 0 and sleep. This causes inconsistency.

Question 17: Goals of mutual exclusion?

Answer:

1. Mutual Exclusion for resource must happen
2. Progress must happen
3. Bounded waiting for fair CPU usage
4. Performance should improve

Question 18: Methods of synchronisation.

Answer:

1. Disabling Interrupts.
2. Mutex (Locks)
3. Semaphore
4. Monitor

Question 19: What is a mutex?

Answer: a mutex is locking mechanism used to synchronize access to a resource. Only one task (can be a thread or process based on OS abstraction) can acquire the mutex. It means there is ownership associated with mutex, and only the owner can release the lock (mutex).

Question 20: What is semaphore?

Answer: Semaphore is simply a variable which is non-negative and shared between threads. This variable is used to solve the critical section problem and to achieve process synchronization in the multiprocessing environment.

- Semaphore is signaling mechanism ("I am done, you can carry on" kind of signal). For example, if you are listening songs (assume it as one task) on your mobile and at the same time your friend calls you, an interrupt is triggered upon which an interrupt service routine (ISR) signals the call processing task to wakeup.
- **Binary Semaphore:** This is also known as mutex lock. It can have only two values – 0 and 1
- **Counting Semaphore:** Its value can range over an unrestricted domain. It is used to control access to a resource that has multiple instances.

Question 21: What are monitors?

Answer: Monitors are classes which have synchronized functions. They are used in Java. Example

```
class Account{  
    private int bal;  
    void synchronize deposit(int x){
```

```
        bal=bal+x;
    }
    void synchronize credit(int x){
        bal=bal-x;
    }
}
```

Question 22: What is a deadlock?

Answer: Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process

Question 23: What are necessary condition for deadlock?

Answer: There are 4 necessary condition for deadlock

1. **Mutual Exclusion-** If a resource is being used by one process, it cannot be used by another process.
2. **Hold and Wait-** Processes must be holding and waiting for some resource. If a process is not holding any resource or is not waiting for any resource then it cannot cause deadlock.
3. **No Preemption-** Operating system cannot take back allocated resources from the process
4. **Circular Wait-** Wait must be in circular order. Example
 - A holds P1 and waits for P2
 - B hold P2 and wait for P1

So there is a circular wait A->P2->B->P1->A

Question 24: What are methods of handling deadlock?

Answer: There are 4 methods to handle deadlock

1. **Prevent deadlock-** Do not let all 4 condition to possibly happen
2. **Avoid deadlock-** Do not grant a request which leads to deadlock
3. **Detect and Recover-** Detect Deadlock and recover
4. **Ignore it-** Interestingly it is used in Windows and Unix

Question 25: What is spooling?

Answer: SPOOL is an acronym for simultaneous peripheral operations on-line. It is a kind of buffering mechanism or 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.

In easier words, printer only prints one thing at a time. So what we can do is store future request in buffer and process them one by one. So whenever any processor asks for a printer in working state, it will be available and there is no mutual exclusion property for deadlock. This is called Spooling.

Question 26: How to prevent deadlock by making counter measures against hold and wait criteria?

Answer:

1. Ask process to declare its usage time (impractical)
2. Ask process to release all resources before making a new request

Question 27: Banker's algorithm for avoidance of deadlock and to even detect it.

Answer: Must Revise

Question 28: What are expectations from memory?

Answer:

1. Low access time
2. High Capacity
3. Low cost to store a bit

Question 29: What is Relative Address and Physical Address?

Answer:

- **Relative Address-** This address is used as a reference to access the physical memory location by CPU. This is actually distance from some base address.
 - Also called virtual address and logical address
- **Physical Address-** Physical Address identifies a physical location of required data in a memory. The user never directly deals with the physical address but can access by its corresponding logical address.
- **Memory Management Unit (MMU)-** This is a hardware device which maps logical address of process to actual physical address. This process is called **address binding**.

Question 30: Types of address binding.

Answer:

1. **Compile Time Binding:** For example in 8085, but that is not logical address rather itself a physical address. But if we use emulator then yeah that is the idea behind this.
2. **Load Time Binding-** Suppose a program is loaded to the memory for the first time. We assign it a base address. Then simply at that time convert all address with reference to base memory.
 - We cannot move the resources once allocated, so this is a problem
3. **Run Time Binding-** Here we need MMU.

Question 31: What is Fragmentation?

Answer: As processes are loaded and removed from memory, the free memory space is broken into little pieces. It happens after sometimes that processes cannot be allocated to memory blocks considering their small size and memory blocks remains unused. This problem is known as Fragmentation.

- **Internal Fragmentation-** Memory block assigned to process is bigger. Some portion of memory is left unused, as it cannot be used by another process.
- **External Fragmentation-** Total memory space is enough to satisfy a request or to reside a process in it, but it is not contiguous, so it cannot be used.

Question 32: Explain Dynamic Partitioning of memory.

Answer: Let's say our memory is divided into small blocks P_1, P_2, P_3, P_4 and P_5

Let's say all were occupied and then processes occupying P_2 and P_4 finishes, creating holes. Let's represent them as doubly linked list.

$$P_1 \rightleftharpoons P_2 \rightleftharpoons P_3 \rightleftharpoons P_4 \rightleftharpoons P_5$$

$$P_1 \rightleftharpoons H \rightleftharpoons P_3 \rightleftharpoons H \rightleftharpoons P_5$$

Now if P_3 becomes a hole as well, then we can combine 3 holes as a bigger hole.

$$P_1 \rightleftharpoons H \rightleftharpoons H \rightleftharpoons H \rightleftharpoons P_5$$

$$P_1 \rightleftharpoons H \rightleftharpoons P_5$$

Question 33: Explain First Fit, Best Fit and Next Fit memory Allocation.

Answer: As explained, in dynamic memory partition we can use a linked list memory mapping. Now which hole to allocate to which memory depends on the allocation algorithm. There are four allocation algorithms

1. First Fit Algorithm-

```
node cur = head;
while(cur && (cur.type == 'allocated' || cur.size() < req))
    cur = cur->next;
```

```
if(cur == null)throw no memory error;
Allocate(cur, req)
```

2. Best Fit Algorithm-

```
node cur = head, best;
while(cur)
    if(cur.type == 'allocated') continue
    if(cur.size() < req) continue
    if(best == null || cur.size() < best.size())
        best = cur;
    cur = cur->next;
if(best == null)throw no memory error;
Allocate(best, req)
```

3. Next Fit Algorithm-

```
node finish = cur;
if(cur.size() >= req)return Allocate(cur, req);
cur = cur->next;
while(cur != finish && (cur.type == 'allocated' || cur.size() < req))
    cur = cur->next;
if(cur == null)cur=head
if(cur == finish)throw no memory error;
Allocate(cur, req)
```

Question 34: Explain paging.

Answer:

- Primary memory is divided into slots of equal sizes called frames
- Process is divided into slots of equal sizes called pages.
- Individual pages are loaded into frames, 2 pages might be loaded into continuous frames or non-continuous frames
- Since CPU generated logical address is contiguous in nature, Static or Load Time Binding is not possible as stated in previous point.
- MMU has a page table base register pointing to page table of process, which is set during context-switching

Question 35: What is virtual memory?

Answer: Processes are told that they have been allocated memory they are demanding, even if there is unavailability. Instead the pages are stored in secondary memory. Pages are then loaded into frame when demand is there.

Question 36: What is page fault?

Answer: Suppose pages P_1 and P_2 are in frames while P_3 and P_4 are in secondary memory. Suppose there is a demand for memory by P_1 , then there is no issue as it is already there. But when P_3 demands memory then it is called a **page fault** as page has to be loaded from slower secondary memory. Reduces speed average throughput.

Question 37: What is Translation Lookaside Buffer?

Answer: TLB is cache in processor for Page Table. We need to look at page table billions of time. So it is optimal to have it cached.

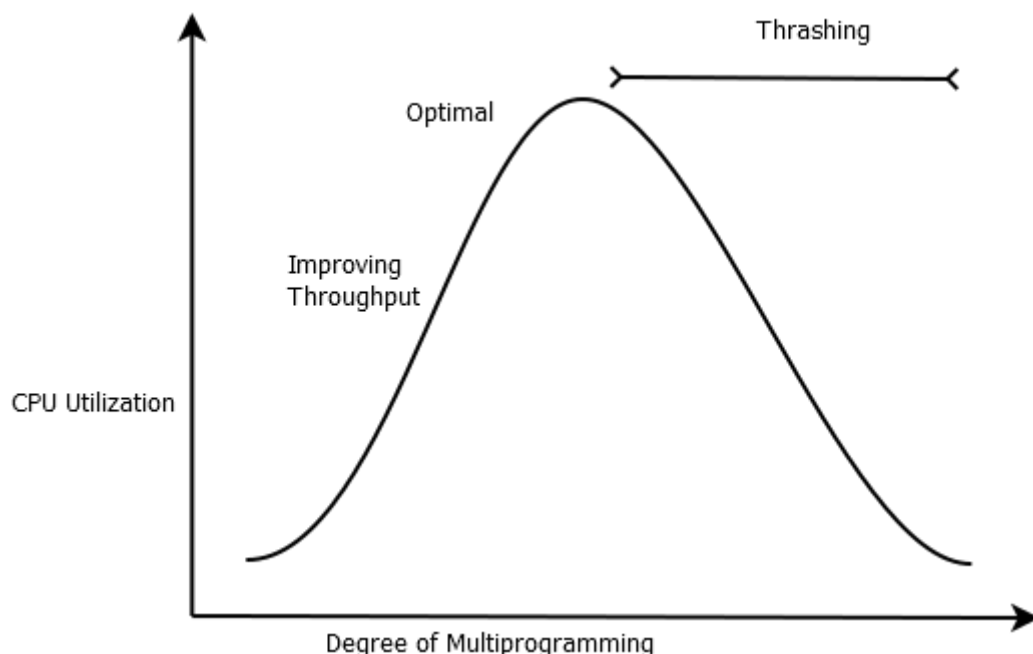
A translation lookaside buffer is a memory cache that is used to reduce the time taken to access a user memory location. It is a part of the chip's memory-management unit.

Question 38: What is demand paging?

Answer: In a system that uses demand paging, the operating system copies a disk page into physical memory only if an attempt is made to access it and that page is not already in memory.

Question 39: What is thrashing?

Answer: When we increase degree of multiprogramming, then CPU utilization is bound to go better. But this is not the case actually. After sometime, since disk pages needs to be loaded from secondary memory more, hence page faults increases thus increasing IO operation. This leads to slower process and CPU is thus kept at hold.



Question 40: Page Replacement Algorithms.

Answer: First of all there are two types of place replacements.

1. **Local Replacement-** Page which is pre-empted from frame belongs to the same process. It is better for single process.

2. **Global Replacement-** Page which is pre-empted from frame can belong to the any process. It is better for overall system.

Three Page Replacement Algorithms

1. **First In First Out**
2. **Optimal-** We find the process which will be used after long time and remove it from frame. This is impractical but optimal.
3. **Least Recently Used-** Process with oldest timestamp is removed. After use timestamp is updated. (Not actual implementation but idea to understand)

Question 41: What is Belady's Anomaly?

Answer: We can assume that if we increase number of frames in physical memory then page faults decreases but this may not be true. For example: Run this is for 3 and 4 frames in FIFO = 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

Question 42: What are segments?

Answer: Segmentation is a memory management technique in which, the memory is divided into the variable size parts. Each part is known as segment which can be allocated to a process. The details about each segment are stored in a table called as segment table.