UT2 Questions

Questions for 2 marks

1. Define terms Critical Conditions and Race conditions.

Critical Section:

- A critical section is group of instructions/statements or region of code that need to be executed atomically (read this post for atomicity), such as accessing a resource (file, input or output port, global data, etc.).
- Problems only arise if one or more of the threads write to the same resources. It is safe to let multiple threads read the same resources

Race Condition:

- The situation where two threads compete for the same resource, where the sequence in which the resource is accessed is significant, is called race conditions.
- A code section that leads to race conditions is called a critical section.

2. Compare pre-emptive vs Non Pre-emptive CPU scheduling.

Basis for Comparison	Preemptive Scheduling	Non Preemptive Scheduling
Basic	The resources are allocated to a process for a limited time.	Once resources 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 till it terminates or switches to waiting state.
Starvation	If a high priority process frequently arrives in the ready queue, low priority process may starve.	If a process with long burst time is running CPU, then another process with less CPU burst time may starve.
Overhead	Preemptive scheduling has overheads of scheduling the processes.	Non-preemptive scheduling does not have overheads.
Flexibility	Preemptive scheduling is flexible.	Non-preemptive scheduling is rigid.
Cost	Preemptive scheduling is cost associated.	Non-preemptive scheduling is not cost associative.

3. What are different file access methods ?

The information stored in the file needs to be accessed and read into the computer memory. There are different methods available to do it. Some of them are:

- **Sequential Access:**
    - This is the most common method.
    - Here the information present in the file is accessed in a sequential fashion, one record after the other.
    - It is a very common approach which is used by editors and compilers usually.

    - The Read and Write operations form the major part of the operations done on a file.
    - A read operation reads the next portion of the file and automatically advances the file pointer, which tracks the I/O location.
    - A write operation appends to the end of the file and advances to the end of the newly written material.
- **Direct Access**
    - This type of access method provides a speedy access to the file. It provides immediate access to large amount of information.
    - Here a file is made up of logical records that allow programs to read and write.
    - It allows the programs to read and write the records in a rapid manner in no particular (or pre-defined) order.
    - It is based on the disk-model of a file, as a disk allows random access to any block.
    - For direct access, we can view the file as a numbered sequence of blocks or records.
    - This method is usually used in databases.
- **Indexed access:**
    - This method is built on top of Direct access method.
    - Here an index contains the pointers to various blocks of the file.
    - So, to find a record inside a file, we firstly search the index and later use the pointer obtained to access the file directly and find the record we have been searching for.

4. Explain Producer Consumer problem ?

The **producer–consumer problem (also known as the bounded-buffer problem)** is an example of a multi-process synchronization problem.

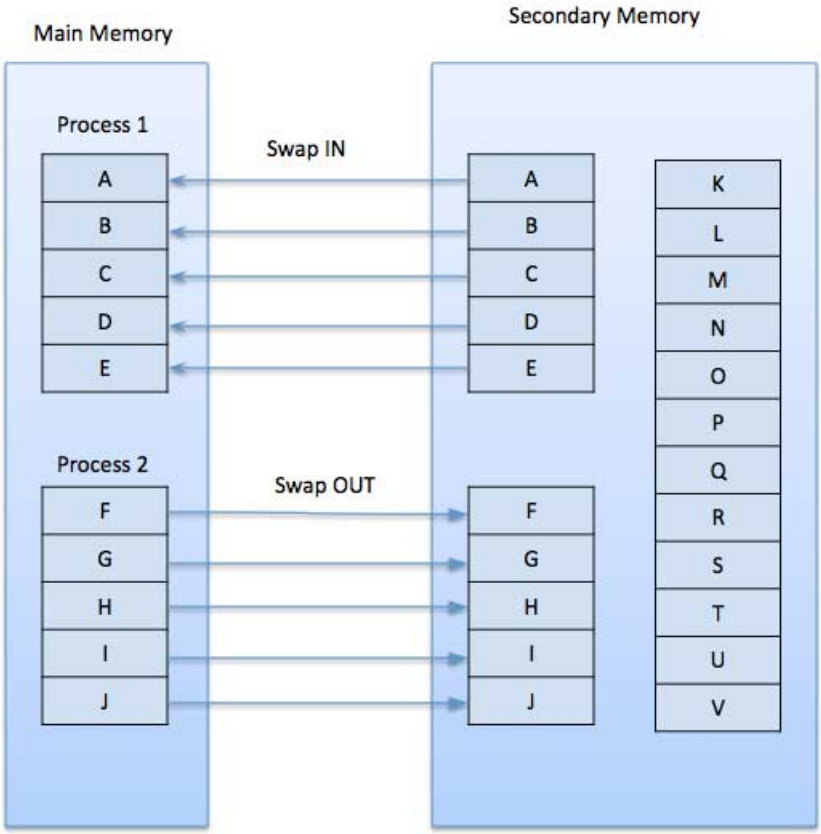
- The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer used as a queue.
- The producer's job is to generate a piece of data, put it into the buffer and start again. At the same time, the consumer is consuming the data (i.e., removing it from the buffer) one piece at a time.
- The problem is to make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer.

5. What is Thrashing in memory management ?

Thrashing is computer activity that makes little or no progress, usually because memory or other resources have become exhausted or too limited to perform needed operations.
In a virtual storage system (an operating system that manages its logical storage or memory in units called pages), thrashing is a condition in which excessive paging operations are taking place.

6. Explain Demand Paging.

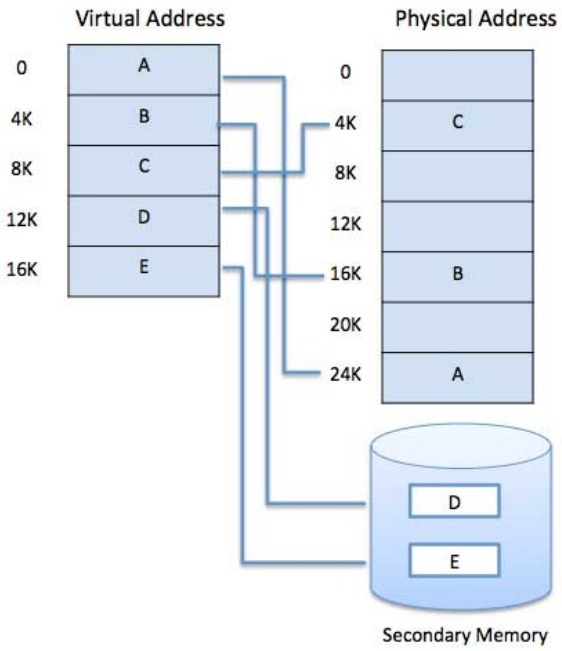
A demand paging system is quite similar to a paging system with swapping where processes reside in secondary memory and pages are loaded only on demand, not in advance. When a context switch occurs, the operating system does not copy any of the old program’s pages out to the disk or any of the new program’s pages into the main memory Instead, it just begins executing the new program after loading the first page and fetches that program’s pages as they are referenced.



7. What is Virtual Memory in Operating System.

Virtual memory is a memory management capability of an OS that uses hardware and software to allow a computer to compensate for physical memory shortages by temporarily transferring data from random access memory (RAM) to disk storage.
Computers have a finite amount of RAM so memory can run out, especially when multiple programs run at the same time. A system using virtual memory can load larger programs or multiple programs running at the same time, allowing each one to operate as if it has infinite memory and without having to purchase more RAM.

As part of the process of copying virtual memory into physical memory, the OS divides memory into pagefiles or swap files that contain a fixed number of addresses. Each page is stored on a disk and when the page is needed, the OS copies it from the disk to main memory and translates the virtual addresses into real addresses.



8. What are the ways of Handling Deadlocks ?

1.Avoidance

- Deadlock avoidance methods use some advance knowledge of the resource usage of process to predict the future state of the system for avoiding allocations that can eventually lead to a deadlock.
- Deadlock avoidance algorithms are usually in the following steps:

i.When a process requests for a resource, if the resource is available for allocation it is not immediately allocated to the process rather the system assumes that the request is granted.

ii.Using advance knowledge of resource usage of processes and assumptions of step 1 the system analysis whether granting the process request is safe or unsafe.

iii.The resource is allocated to the process only if the analysis of step 2 shows that it is safe to do so otherwise the request is deferred.

2.Prevention

- This approach is based on the idea of designing the system in such a way that deadlocks become impossible.
- Mutual exclusion, hold-and-wait, no pre-emption and circular-wait are the four necessary conditions for a deadlock to occur.

3.Deadlock Detection

- In this approach for deadlock detection, the system does not make any attempt to prevent deadlock but allows processes to request resources and wait for each other in uncontrolled manner.

4.Recovery from Deadlock
i.Ask for operator intervention

ii.Termination of Process

iii.Rollback of processes

9. Explain Logical Versus Physical Address Space ?

Definition of Logical Address

Address generated by **CPU** while a program is running is referred as **Logical Address**. The logical address is virtual as it does not exist physically. Hence, it is also called as **Virtual Address**. This address is used as a reference to access the physical memory location. The set of all logical addresses generated by a programs perspective is called **Logical Address Space**.

Definition of Physical Address

Physical Address identifies a physical location in a memory. MMU (**Memory-Management Unit)** computes the physical address for the corresponding logical address. MMU also uses logical address computing physical address. The user never deals with the physical address.

The set of all physical addresses corresponding to the logical addresses in a Logical address space is called **Physical Address Space**.

10. How to recover from Deadlock ?
Process Terminatation:

- Abort all deadlocked processes
  It will break the deadlock but at a great wxpense
- Abort one process at a time until deadlock cycle is eliminated
  Considerable overhead since after each process is aborted, a deadlock-detection algorithm must be invoked.

Resource Preemption:
We successiveliy preempt some resources from processes and give these resources to other processes until the deadlock cycle is broken

- Selecting a Victim:
  We must determinr the order of the preemption to minimize cost
- Roll Back:
  The preempted process must roll back to some safe state and restart again
- Strarvation:
  Grauranting resource wont be preemted from same process

11. How Mutex Lock works.

In computer programming, a ***mu****tual* ***ex****clusion object (mutex)* is a program object that allows multiple program threads to share the same resource, such as file access, but not simultaneously. When a program is started, a mutex is created with a unique name. After this stage, any thread that

needs the resource must lock the mutex from other threads while it is using the resource. The mutex is set to unlock when the data is no longer needed or the routine is finished.

12. Explain following terms
Throughput, Turnaround time, Waiting time, Response time.

Throughput: number of processes completed per unit time.
Turnaround Time: mean time from submission to completion of process.
Waiting Time: Amount of time spent ready to run but not running.
Response Time: Time between submission of requests and first response to the request.

13. How to detect deadlock ?

Deadlock Detection

- If deadlocks are not avoided, then another approach is to detect when they have occurred and recover somehow.

Single Instance of Each Resource Type

- If each resource category has a single instance, then we can use a variation of the resource-allocation graph known as a ***wait-for graph***.
- A wait-for graph can be constructed from a resource-allocation graph by eliminating the resources and collapsing the associated edges
- As before, cycles in the wait-for graph indicate deadlocks.

Several Instances of Resourse type

- The wait-for graph is not applicable to resourse allocation system with multiple instances of each resourse type.
- We turn now to deadlock detection algorithm that is applicable to such a system
- Key Features:
  - Available: A vector of length m indicates available resource.
  - Allocation: A n x m matrix defines the no. of resources of each type allocated.
  - Request: A n x m matrix indicates the current request of each process.

14. Explain features of Distributed system
Resource Sharing:
It provides a mechanism for sharing files at remote sites, processing information in a distributed database, printing files at remote sites, using remote specialized hardware devices and performing other operations.
Computing speedup:
A distributed system allows us to distribute the subcomputations among the various sites the subcomputation can be run concurrently and thus provide computing speedup.
Reliability:
If one site fails in a Distributed system the remaining sites can continue operating giving better reliabilty
Communication:
When several sites are connected to one another by a communication network users at various sites have the opportunity to exchange information
Communication happens internally.

15. Explain Stateful Versus Stateless Service

Parameters	Stateful	Stateless
1. State	A Stateful server remember client data (state) from one request to the next.	A Stateless server keeps nostate information
2. Programming	Statefulserver is harder to code	Statelessserver is straightforward to code
3.Efficiency	MoreBecause clients do not have to provide full file information every time they perform an operation	Less because information needs to be provided
7.Operations	Open, Read, Write, Seek, Close	Read, Write

Questions for 5 Marks

1. Short Note on RAID (Redundant array of Independent Disk)

2. Short Note of Distributed System

3. Explain Paging in detail with necessary diagrams

4. Explain Segmentation with the help of diagram

5. Calculate Hit and Miss using LRU, Optimal, FIFO, page replacement policies for following sequence. Page frame size is 3.

6. What is deadlock ? State necessary conditions for the deadlock. How to prevent the deadlock?

7. Explain semaphore ?

8. Explain bankers algorithms in detail.

9. What is internal and External Fragmentation.

10. Explain Different File allocation techniques in detail.

11. On the disk with 1000 cylinders, number 0-999. Computer the number of tracks the disk arm must satisfy all request in the disk queue. Assume the last request was at track 345 and the head is moving toward track 0. The queue in the FIFO order contains request for the following tracks. 123, 874, 692, 475, 105, 376.

Perform following computation for the following scheduling algorithms
i. FIFO ii. SSTF iii. SCAN

12 . Use following scheduling algorithms to calculate the ATAT and AWT for the following process.
1 . FCFS 2. Pre-emptive and non-premptive SJF 3. Preemptive Priority

Process, Arrival Time, Burst Time, Priority
P1, 0, 8, 3
P2, 1, 1, 1
P3, 2, 3, 2
P4, 3, 2, 3
P5, 4, 6, 4

13 . Explain bankers algorithm for safe state.

14 . Short note NFS ?

15 . Explain role of Swapping in Operating System.