

CS6401- Operating System**QUESTION BANK****UNIT-II****Part-A****1. Define process?**

A process is more than a program code, which is sometime known as the text section. IT also includes the current activity, as represented by the value of the program counter and the processor's registers.

2. What is meant by the state of the process?

The state of the process is defined in part by the current activity of that process. Each process may be in one of the following states.

- New: The process is being created.
- Running: Instruction are being executed
- Waiting: The process is waiting for some event to occur.
- Ready: The process is waiting to be assigned to a processor
- Terminated: The process has finished execution

3. Define process control block contain?

Each process is represented in the operating system by a process control block (PCB) – also called as task control block. The PCB simply serves as the repository for any information that may vary from process to process.

4. What does PCB contain?

- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory management information
- Accounting information

5. What are the 3 different types of scheduling queues?

- Job Queue: As process enters the system they are put into job queue.
- Ready Queue: The processes that are residing in the main memory and are ready and waiting to execute are kept in the queue
- Device Queue: The list of processes waiting for particular I/O device is called a device queue.

6. Define schedulers?

A process migrates between the various scheduling throughout its lifetime. The operating system must select, for scheduling purposes, processes from these queues in some fashion. The selection process is carried out by the appropriate scheduler.

7. What are the types of scheduler?

Long term scheduler or job scheduler selects processes from the pool and load them into the memory for execution. Short term scheduler or CPU scheduler, select among the processes that are ready to execute and allocates the CPU to one of them.

8. Define critical section?

If a system consist on n processes $\{P_0, P_1, \dots, P_{n-1}\}$. Each process has a segment of code called a critical section, in which the process may be changing common variables, updating a table , writing a file. The important feature of this system is that, when one process is in its critical section, no other process is to be allowed to execute in its critical section.

9. What requirement is to be satisfied for a solution of a critical section problem?

A solution to the critical section problem must satisfy the following 3 requirements.

- Mutual exclusion: If process P1 is executing in its critical section, then no other processes can be executing in their critical sections.
- Progress: If no process is executing in its critical section and some processes wish to enter their critical sections, then only those processes that are not executing in their remainder section can participate in the decision on which will enter its critical section next, and this selection cannot be postponed indefinitely.
- Bounded waiting: There exists a bound on the number of times that other processes are allowed to enter their critical section after a process has made a request to enter its critical section and before that request is granted.

10. Define semaphores.

Semaphore is a synchronization toll. A semaphore S is an integer variable that apart from initialization is accessed only through 2 standard atomic operations.

- Wait
- Signal

11. Define Starvation in deadlock?

A problem related to deadlock is indefinite blocking or starvation, a situation where processes wait indefinitely within a semaphore. Indefinite blocking may occur if we add and remove processes from the list associated with a semaphore in LIFO order.

12. Name some classic problem of synchronization?

- The Bounded – Buffer Problem
- The Reader – Writer Problem
- The Dining –Philosophers Problem

13. Define deadlock?

A process request resources; if the resource are not available at that time, the process enters a wait state. Waiting processes may never change state, because the resources they are requested are held by other waiting processes. This situation is called deadlock.

14. What is the sequence of operation by which a process utilizes a resource?

Under the normal mode of operation, a process may utilize a resource in only the following sequence:

- Request: If the request cannot be granted immediately, then the requesting process must wait until it can acquire the resource.
- Use: The process can operate on the resource.
- Release: The process releases the resource

15. Give the condition necessary for a deadlock situation to arise?

A deadlock situation can arise if the following 4 condition hold simultaneously in a system.

- Mutual Exclusion
- Hold and Wait
- No preemption
- Circular Wait

16. Define ‘Safe State’?

A state is safe if the system allocates resources to each process in some order and still avoid deadlock.

17. What is the use of cooperating processes?

- Information sharing: Since several users may be interested in the same piece of information, we must provide an environment to allow concurrent access to these type of resources.
- Computation speedup: If we want a particular task to run faster, we must break it into subtask, each of which executing in parallel with others.
- Modularity: We may want to construct the system in a modular fashion, dividing the system functions into separate processes or thread.
- Convenience: Even an individual user may have many tasks on which to work at one time. For instance a user is editing, printing and computing in parallel.

18. Define deadlock-avoidance algorithm?

A deadlock-avoidance algorithm dynamically examines the resource allocation state to ensure that a circular wait condition can never exist. The resource allocation state is defined by the number of available and allocated resources, and the maximum demand of the processes.

19. What are the benefits of multithreaded programming?

- Responsiveness
- Resource sharing
- Economy
- Utilization of multiprocessor architecture

20. Define deadlock detection diction?

If a system does not employ either a deadlock-prevention or a deadlock avoidance algorithm, then a deadlock situation may occur. In this environment, the system must provide:

- An algorithm that examines the state of the system to determine whether a deadlock has occurred
- An algorithm to recover from the deadlock.

21. Define race condition.

When several process access and manipulate same data concurrently, then the outcome of the execution depends on particular order in which the access takes place is called race condition. To avoid race condition, only one process at a time can manipulate the shared variable

22. What is critical section problem?

Consider a system consists of 'n' processes. Each process has segment of Code called a critical section, in which the process may be changing common variables, updating a table, writing a file. When one process is executing in its critical section, no other process can allowed to execute in its critical section.

23. 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.

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

The three requirements are

- Mutual Exclusion
- Progress
- Bounded waiting

25. Define entry section and exit section.

The critical section problem is to design a protocol that the processes can use to cooperate. Each process must request permission to enter its critical section. The section of the code implementing this

request is the entry section. The critical section is followed by an exit section. The remaining code is the remainder section.

26. What are conditions under which a deadlock situation may arise?

A deadlock situation can arise if the following four conditions hold Simultaneously in a system:

- a. Mutual exclusion
- b. Hold and wait
- c. No pre-emption
- d. Circular wait

27. What is a resource-allocation graph?

Deadlocks can be described more precisely in terms of a directed graph called a system resource allocation graph. This graph consists of a set of vertices V and a set of edges E . The set of vertices V is partitioned into two different types of nodes; P the set consisting of all active processes in the system and R the set consisting of all resource types in the system.

PART-B

1. What is the important feature of critical section? State the dining philosopher's problem and show how to allocate the several resources among several processes in a deadlock and starvation free manner.
2. How can deadlock be detected? Explain.
3. Write about the various CPU scheduling algorithms.
4. Write notes about multiple-processor scheduling and real-time scheduling.
5. Write about critical regions and monitors.
6. Consider the following five processes, with the length of the CPU burst time given in milliseconds.

| Process | Burst time |
|---------|------------|
| P1 | 10 |
| P2 | 29 |
| P3 | 3 |
| P4 | 7 |
| P5 | 12 |

Consider the First come First serve (FCFS), Non Preemptive Shortest Job First(SJF), Round Robin(RR) (quantum=10ms) scheduling algorithms. Illustrate the scheduling using Gantt chart. Which algorithm will give the minimum average waiting time? Discuss.

7. Consider the following page reference string
7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

How many page faults would occur for the following replacement algorithms, assuming three frames that all frames are initially empty?

8. i) Construct a Resource Allocation Graph for the following scenario. At time 't' Process P1 request for a resource X, process P2 requests for a resource Y. Both the resources are Available and they are allocated to the requesting process. At time t_1 where $t_1 > t_2$ both the processes are still holding the resources, however process P1 request for Y which is held by P2, process P2 request for X held by P1. Will there be a deadlock? If there is a deadlock discuss the four necessary conditions for deadlock, else justify there is no deadlock .
ii) With relevant example show that the implementation of a semaphore with a waiting queue may result in deadlock
9. Discuss about the issues to be considered with multithreaded programs.
10. Write about the three common types of threading implementation.
11. Explain the two solutions of recovery from deadlock.