Devanshu Surana
1032210755
PC-12

OS
Lab Assignment 6
Title: Process Synchronization (Deadlocks)

FAQ's

ans 1) A deadblock is a situation in which more than one process is blocked because ~~of~~ it is holding a resource and also requires some resource that is acquired by some other process.

Four neccessary ~~situations~~ conditions of a deadlock situation are:..

1) Mutual Exclusion:- only one process can use a resource at any given time.

2) Hold and wait:- A process is holding atleast one resource at a time and is waiting to acquire other resources held by some other process.

2/12/22

3) No preemption: Resource can be released by a process voluntarily.

4) Circular wait: A set of process are waiting in circular fashion for each other.

ans 2)

Deadlock	Starvation.
• All processes keep waiting for each other to complete and none of them get executed.	• High priority processes keep executing and low priority processes are blocked.
• Resources are blocked by processes.	Resource is continously utilized by high priority processes.

• Neccessary conditions : Mutal exclusion, Hold and wait, No preemption and cirular wait.	• Priorities are assigned to the process.
• Also known as circular wait.	• Also known as lived lock

ans 3.

	Resource Allocation Policy	Different schemes	Major Advanteges	Major Disadvanteges.
Detection	Very liberal: requested resources are granted where possible.	Invoke periodically to test for deadlock	Never delay process initiation	Inherent preemption losses.
Prevention	Comservative: undercommits resources	Requesting all ~~schemes~~ resources at once. Preemption and resource ordering.	Works well for process that perform single burst of activity	Inefficient and delays process initiation.
Avoidance	selects midway of Detection and preemption.	Manipulate to find atleast one safe path.	No preemption is necessary.	Future requiremen must be known Processes can b blocked for long periods.

ans 4. Safety algorithm is used to check whether or not a system is in ~~sta~~ Safe state or follows the safe sequence in Banker's Algorithm.

Safety algorithm.

```
1. let work and finish be vectors of length
   m and n respectively. Initialize
     work = available
     finish [i] = false  for  i = 1, 2, ... n

2. a) finish[i] = false
   b) Need i ≤ work
   if no such i exists, go to step 4.

3. Work = work + Allocation
   finish[i] = true
   go to step 2

4. If finish[i] = true for all i; then the system is in
   safe state.
```

8/
18/1/22