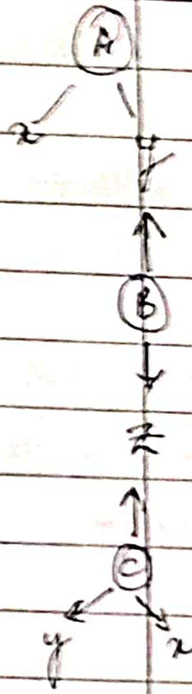
OS - ASSIGNMENT

21BCE7371
Radha Krishma Garg

_/_/_



four Conditionals of deadlock :-

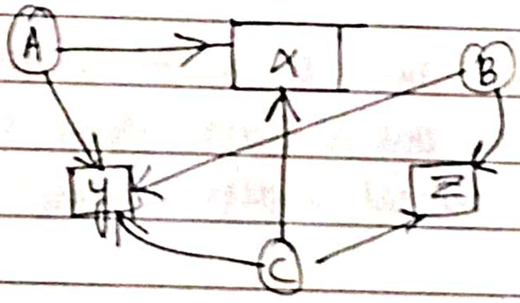
a) mutual exclusion: All resources are non-shareable

b) No-premption: process should not be prempted.

c) Hold and wait: All process are holding a resource and waiting for others.

d) circular wait

Diagram



Deadlock will not happen because:

1) The hold and wait conditions is not there because A is not waiting for any resource.

2) Circular wait is not happening

There are 4 ways to deal with deadlocks

1) Deadlock Avoidance
2) Deadlock Ignorance
3) Deadlock prevention
4) Deadlock detection and recovery

Ans-2 Critical Section :- This refer to the situation where multiple process or threads are accessing a shared resource or __/__/____ the critical section of a code.

Requirements

1) Mutual Exclusion :- Only process at a time should be allowed to enter the critical section.

2. Progress :- If no process is currently executing in the critical Section and there are some process waiting to enter then the selection of the next process to enter the critical selection cannot be postponed indefinitely.

3. Bounded wait :- A bound must exist on the no. of times other process are allowed to enter the critical section after a process has made a request to enter the critical section and before that request is granted.

Solution to 2:process critical section is Peterson's Algorithm

```
int turn =0;
bool flag [2] =~~ø~~{false, false};
// process 0
flag [0] = true;
turn =1;
while ( flag [1] ~~ad~~ and turn ~~##~~ ==1);
flag [0] = false;

~~8~~ // process 1
flag [1] = true;
turn =0;
while (flag [0] and turn ~~ttor~~ ==0);
flag [1] = false;
```

Each process gives priority to its own flag and set it to true when it is ready to enter the critical section. And then gives priority to others process id and waits until the other process finishes or given priority in critical section Then the flag is set to false which indicates completion of work by process This Algorithm satisfies all the requirment mentioned.

Ans-3 Place the given processes

Best Fit Algo

1. Process 215 kb in 215 kb partition
2. Process 423 kb in 520 kb partition
3. Process 112 kb in 215 kb partition
4. Process 426 kb in 600 kb partition

Worst fit Algorithm

1. Process 215 Kb in 600 Kb partition
2. Process 423 kb in 520 kb partition
3. Process 112 Kb in 360 kb partition
4. Process 426 kb in 600 kb partition

First fit Algo

1 Process 215 Kb in 215 Kb partition
2 Process 423 kb in 600 kb partition
3 Process 112 kb in 215 Kb partition
4 Process 426 kb in 600 Kb partition

Best Fit :- Algorithm minimizes the amount of wasted space in the memory partitions