NAME:- KRUNAL RANK

ROLL NO:- U18C0081

CLASS:- BTECH III, Computer Eng.

SEM:- Semester 6

OS Tutorial

Ans 1:- Semaphore is simply a variable that is non negative and shared between threads.
A semaphore is a signaling mechanism and a thread that is waiting on a semaphore can be signaled by another thread.

Some types of semaphore are:-

- Counting Semaphore:-
This type of Semaphore uses count that helps task to be acquired or released numerous times. If initial count = 0, then counting semaphore creates an unavailable zone.

- Binary Semaphore:-
This type of semaphore only allows a resource to be accessed by a single process. (Called Mutex Lock).

For example,
When a process requires a resource, it calls the wait function which is as follows:-

```
wait(){
  while(s==0);
  s=s-1;
}
```

It waits until s=0 & When it releases a resource it calls wake function:-

```
wake(){
  while s=s+1);
}
```

It releases the resource.

U18CO084 KRUNAL RANK K.R.Rank

Ans 2: In distributed systems, we neither have shared memory nor a common physical clock and there for we can not solve mutual exclusion problem using shared variables. To eliminate the mutual exclusion problem in distributed system approach based on message passing is used.

A site in distributed system do not have complete information of state due to lack of shared memory and a common physical clock.

Solution for such a problem:-

- Token based Algorithm:-
A unique token can be shared among all the sites.
This is a similar solution as semaphores and mutex lock.
As explained the processes use wait and hold function to modify the token that allows them to access critical zone resources.

- Non token based approach:-
A site communicates with other sites in order to determine which sites would execute critical section next. This involves exchange of two or more successive round of message among site.

U18CO081 KRUNAL RANK

Ans 3: The ways in which CSCAN is better than SCAN algorithm are as follows:-

- It provides uniform waiting time in requesting the locations. This is because the requests are processed only in a single direction.
- This algorithm allows provides better response time in some cases.

Ans 4: The sharing of code among different processes can be done with the help of virtual memory.

- The programs are linked in the virtual address space that is much larger than the physical memory.
- The basic idea behind paging is that when a process is swapped in, the pager only loads into memory the processes that are expected to run right away.
- On the other hand, if the page is not found a Page Fault Trap signal is generated which handles the loading of required page.
- In this way, the code can be shared among different processes.

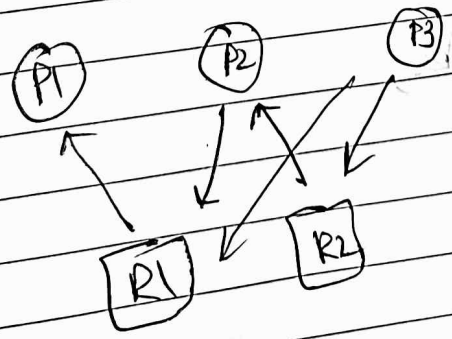
Ans 5: In some cases, Static linking may be preferred because of the following reasons:-

- Portability: The make files become portable.
- Insulation: By using default configurations of libraries which are dynamically linked, you are insulated from bugs or limitations.
- Quality: The performance of shared library improves over time and updated code is used.
- Stability: The combinations of library are too hard to reconstruct and use.

In these ways, Static linking may result in a better performance than dynamic linking.

Ans 6. Lets us consider 3 processes P1, P2 and P3 and 2 types of resources. They are having 1 instance each.

The below graph shows a no deadlock condition as no cycle is present.
If a cycle is present, then deadlock condition is faced.



U18CO081 KRUNAL RANK K.R.Rank

Ans 7: For mutual exclusion of more than 1 processes, we can use a counting semaphore.

~~int s = LIMIT // the token~~

```
int LIMIT = 5   // the total processes allowed
int s = LIMIT;
void wait() {
  while(s==0);
  s = s-1;
}

Void wake(){
    s=s+1;
}
```

Now for any process P, we can call it like:-

```
void P(){
    // some tasks;
    wait();
    // access critical zone;
    wake();
    // some other tasks;
}
```