Operating System Concepts

Sessional Assignment Spring 2020

Marks : 20

NAME SAEEDA NAZ

ID 14332

DEPARTMENT BS(CS) 5TH SEMESTER

1. Explain the necessary conditions that may lead to a deadlock situation. 2. What are the various methods for handling deadlocks?

ANSWER:

NECESSARY CONDITION

A deadlock situation can arise if and only if the following four condition hold simultaneously in a system-

Mutual Exclusion: At least one resource is held in a non-sharable mode that is only one process at a time can use the resource. If another process requests that resource, the requesting process must be delayed until the resource has been released.

Hold and Wait: There must exist a process that is holding at least one resource and is waiting to acquire additional resources that are currently being held by other processes.

No Preemption: Resouces cannot be preempted; that is, a resource can only be released voluntarily by the process holding it, after the process has completed its task.

Circular Wait: There must exist a set {p0, p1,.....pn} of waiting processes such that p0 is waiting for a resource which is held by p1, p1 is waiting for a resource which is held by p2,..., pn-1 is waiting for a resource which is held by pn and pn is waiting for a resource which is held by p0

handling deadlocks:

There are mainly four methods for handling deadlock.

1. Deadlock ignorance

It is the most popular method and it acts as if no deadlock and the user will restart. As handling deadlock is expensive to be called of a lot of codes need to be altered which will decrease the performance so for less critical jobs deadlock are ignored. Ostrich algorithm is used in deadlock Ignorance. Used in windows, Linux etc

2. Deadlock prevention

It means that we design such a system where there is no chance of having a deadlock.

Mutual exclusion:

Hold and wait:

Active approach:

Wait time out:

Circular wait:

No pre-emption:

3. Deadlock avoidance

Here whenever a process enters into the system it must declare maximum demand. To the deadlock problem before the deadlock occurs. This approach employs an algorithm to access the possibility that deadlock would occur and not act accordingly. If the necessary condition of deadlock is in place it is still possible to avoid feedback by allocating resources carefully.

4. Detection and recovery

When the system is in deadlock then one method is to inform the operates and then operator deal with deadlock manually and the second method is system will automatically recover from deadlock. There are two ways to recover from deadlock:

Process termination:

Resources preemption:

2. Is it possible to have a deadlock involving only one single process? Explain your answer.

ANSWER:

No. This follows directly from the hold-and-wait condition.
A deadlock situation can arise if the following four conditions hold simultaneously in a system.
Mutual Exclusion.
Hold and Wait.
No Preemption.
Circular-wait.
It is not possible to have circular wait with only one process, thus failing a necessary condition for Circular wait. There is no second process to form a circle with the first one. So it is not possible to have a deadlock involving only one process.

3 . Consider a system consisting of 4 resources of the same type that are shared by 3 processes, each of which needs at most 2 resources. Show that the system is deadlock free.

ANSWER:

Suppose the system is deadlocked. This implies that each process is holding one resource and is waiting for one more. Since there are three processes and four resources, on process must be able to obtain two resources. This process requires no more resources and therefore it will return its resources when done.

4 .What is a resource allocation graph? How do you obtain a wait-for graph from it? Explain their uses.

ANSWER:

The resource allocation graph is **the** pictorial representation of **the** state of a system. As **its** name suggests, **the resource allocation graph** is **the** complete information about all **the** processes which **are** holding some **resources** or **waiting** for some **resources**.

5.Can a system detect that some of its processes are starving? If you answer "yes," explain how it can. If you answer "no," explain how the system can deal with the starvation problem.

ANSWER:

Detection of starvation in effect requires future knowledge since no amount of record-keeping statistics on processes can determine if it is making "progress"or not.However, starvation can be prevented by "aging"a process. This means maintaining a roll back count for each process, and including this as part of the cost factor in the selection process for a victim for preemption/rollback.

6. On a disk with 1000 cylinders, number 0 to 999, compute the number of tracks the disk arm must move to satisfy all the requests in the disk queue. Assume the last request serviced was at

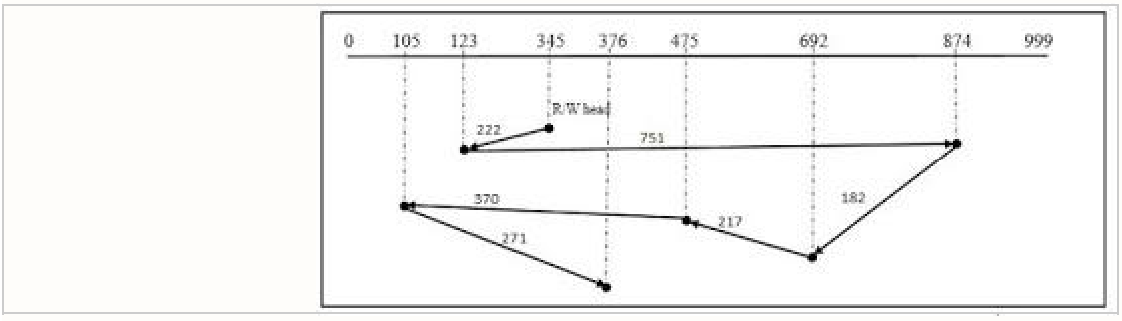
track 345 and the head is moving toward track 0. The queue in FIFO order contains requests for the following tracks: 123, 847, 692, 475, 105, 376. Perform the computations for the following disk scheduling algorithms:

- FCFS
- SSTF

ANSWER:

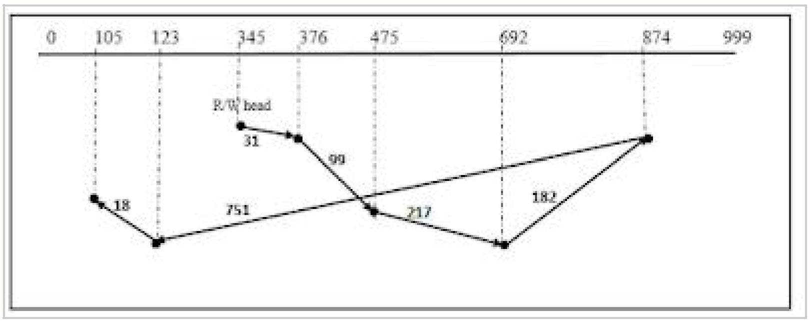
Perform the computation for the above scheduling algorithms.

FCFS:



Total R/W head Movement:
222+751+182+217+370+271 =2013 tracks

SSTF:



Total R/W head Movement:
31+99+182+217+182+751+18=1298 tracks