Yash Mahajan SE IT B 04

6) Post Experiment Exercise :-

A] Extended Theory

(1) Explain deadlock situation & how can it be eliminated.

Deadlock is a situation where a set of processes are blocked because each process is holding a resource & waiting for another resource acquired by some other process.
For eg, In OS when there are two or more resources that hold some resource & wait for resources held by other.
We can prevent deadlock by eliminating the condition.
a) Eliminate mutual exclusion.
It is not possible to dis-satisfy the mutual exclusive
a) Avoid nested locks :- You must avoid giving locks to multiple threads, this is the main reason for deadlock condition.
b) Avoid unnecessary locks - The locks should be given to the important threads. Giving locks to unnecessary threads - that cause the deadlock condition.
c) Using thread join - Deadlock usually happens when one thread is waiting for the other to finish. In this case, we can use Thread.join with a maximum time that a thread will take.

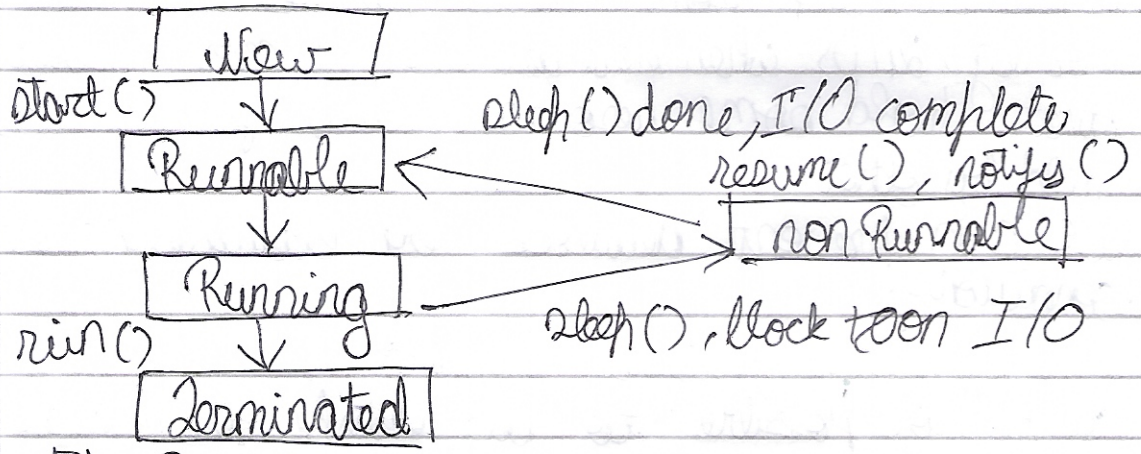
Page No :
Date :

Yash Mahajan SE IT B 04

2) Explain with diagram Java thread model and thread life cycle model.

The Java language & it's runtime system was designed keeping in mind about multi-threading. Java provides synchronous thread environment this helps to increase the utilization of CPU.

Life cycle of a thread :-



(i) New :- A new thread begins in its cycles in new state. It remains there until program starts the thread.

(ii) Runnable :- After a newly born thread is started, the thread becomes runnable. A thread in this state is considered to be executing tasks.

Yash Mahajan SE IT B 04

(iii) Waiting/Blocked :- Sometimes a thread transitions to waiting state while the thread waits for another thread to perform a task.

(iv) Terminated :- A runnable thread enters the terminated state when it completes its task or otherwise terminates.

D] Conclusion :-

In this experiment we have implemented programs related to Multithreading and Thread Synchronization.

Multithreading can be used to increase parallelism to make most of the available CPU resources & to improve application responsiveness. Java also provides thread synchronization to control access to shared resources and maintain consistency of data.

A web server will utilize multiple threads to simultaneously process requests for data at the same time. Multithreading also leads to minimizing & more efficient use of computing resources.