OS THREAD ASSIGNMENT

Name: Aditya Aggarwal   **Roll Number: IIT2019210**

Q1) What Is the Difference Between a Process and a Thread?

Solution: The process can be referred to as a program in execution whereas thread is part of the process. Process has its own address space whereas multiple threads share the same address space of process. Each thread has its own stack. Processes can have multiple threads but thread is the smallest unit which can execute concurrently with other threads. You do not require synchronization in case of process. Threads require synchronization to avoid unexpected scenarios. Processes can communicate to each other using inter-process communication only whereas thread can communicate directly as thread shares the same address space. You can easily create new threads by calling thread's start method but you need to copy resources of the parent process to create a new child process.

Q2) What is the difference between start and run method in Java Thread?

Solution: Start is used to create new thread and run method is used to start the execution of newly created thread. We cannot invoke start multiple times (can be invoked once) and we can invoke run method any number of times. Start is defined in java.lang. Thread class whereas run is defined in java.lang.runnable Interface and must be overridden in the implementing class.

Q3) Which one is better to implement thread in Java? Extending Thread class or implementing Runnable?

Solution: We can define a thread in the following two ways by extending Thread class or By implementing Runnable interface In the first approach, Our class always extends Thread class. There is no chance of extending any other class. Hence, we are missing Inheritance benefits. In the second approach, while implementing Runnable interface we can extend any other class. Hence, we are able to use the benefits of Inheritance. Because of the above reasons, implementing Runnable interface approach is recommended rather than extending Thread class.

Q4) What is deadlock?

Solution: Deadlock is a condition in java threads in which when there is synchronization failure, two or more threads gets blocked for infinite time. This condition happens when multiple threads wait for each other to resume their execution but because all are waiting so they all get blocked, waiting for each other. Deadlock occurs when multiple threads need the same locks but obtain them in different order.

Q5) What is thread starvation?

Solution: Starvation is also similar to deadlock where threads are also waiting for each other. But here waiting time is not infinite after some interval of time, the waiting thread always gets the resources whatever is required to execute the thread run() method.

Q6) what is the difference between the wait() and sleep() methods?

Solution: The basic difference between wait() and sleep() method is that sleep() method is used to block the thread for some time and wait() method is mostly used whenever we need to wait for other thread to complete the execution and then do some work with both threads. Moreover sleep() takes an argument t(in milli seconds) and when sleep() is invoked, thread is blocked for t milli seconds and whenever wait() is invoked, thread is blocked until notify() is invoked.

Q7) What is the difference between synchronous and asynchronous programming?

Solution: Synchronous way: It waits for each operation to complete, after that only it executes the next operation.
When you make a request or have a task to run, a thread is allocated to handle it. If the response to your request involves something time-consuming like an I/O operation or a database query, the thread remains idle until the operation is complete.
Asynchronous way: It never waits for each operation to complete, rather it executes all operations in the first GO only.
Asynchronous programming is a method of programming that allows more than one task to be executed at the same time. When a task has completed its execution, the main thread is notified to collect the result.

Q8 and 9)Codes are attached.