Experiment No. 14

Aim: Create an application to demonstrate use of Multithreading in Java.

Problem Statement:

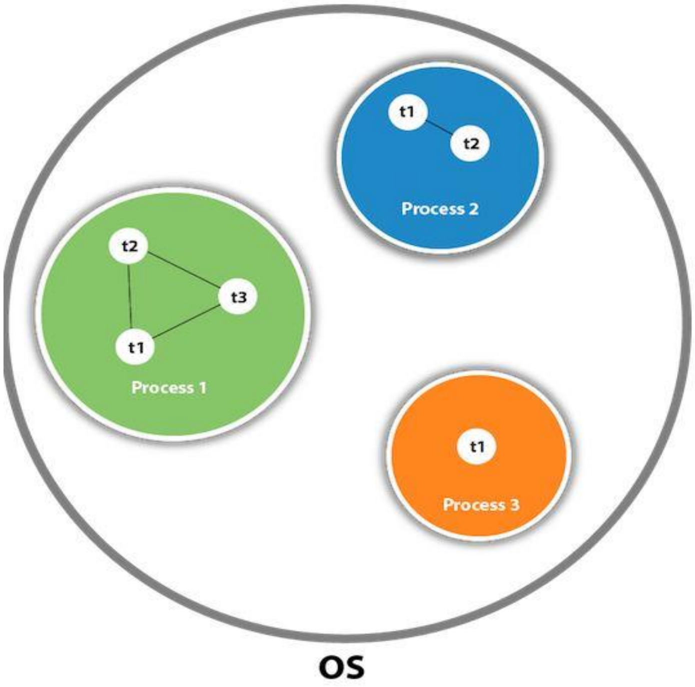
1. Write a java program to demonstrate execution of multiple threads.
2. Write a java program to pause execution of one or multiple threads using sleep ().

Theory:

Thread in java

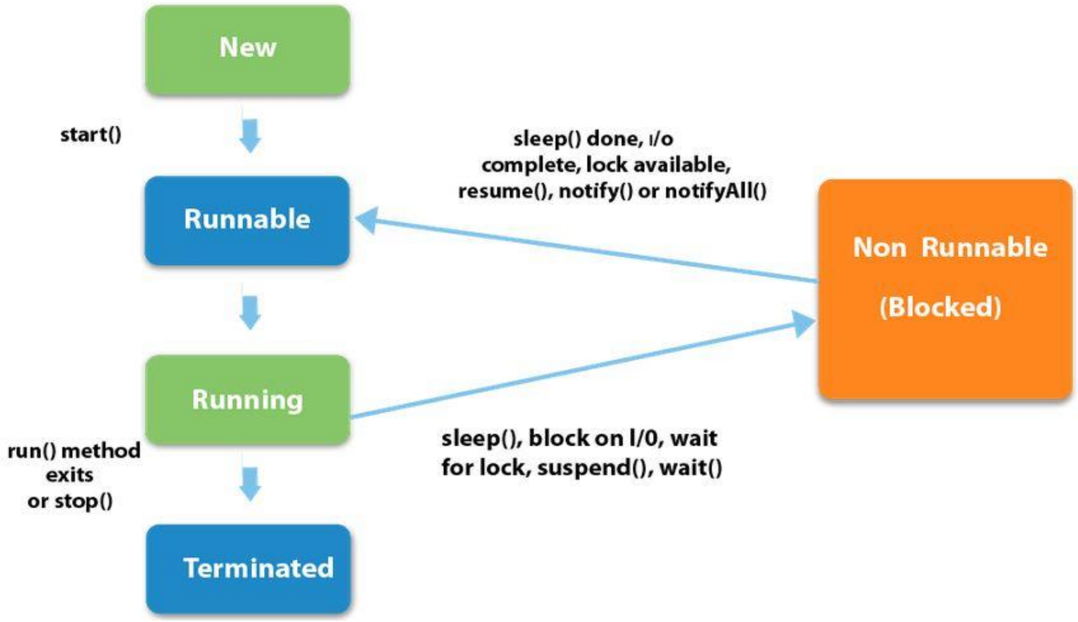
- Process is program in running. Program may have two or more parts. Each part of program is a called as thread.
- Thread-
  It is the smallest unit of processing. It is a lightweight subprocess. If exception in one thread, it doesn't affect other threads. It shares memory area and same address space of process. One process may have one or more threads.

Process and Threads



Life cycle of a Thread (Thread States)

- JVM controls life cycle of the thread in java.
- The five states of java thread-
  - New
  - Runnable
  - Running
  - Non-Runnable (Blocked)
  - Terminated


New
start()
Runnable
sleep() done, I/o complete, lock available, resume(), notify() or notifyAll()
Non Runnable
(Blocked)
Running
sleep(), block on I/0, wait for lock, suspend(), wait()
run() method exits or stop()
Terminated

Multithreading

- It is a process of executing multiple threads simultaneously.
- It is a Java feature that allows concurrent execution of two or more parts of a program.
- Multithreading is used to achieve multitasking.
- Used for maximum utilization of CPU.

How to Define/create and start thread

- Define new thread class that extends Thread class.
- Create object of new thread class
    - **MyThread t1=new MyThread();**
- Invoke start() method to start thread.
    - Threadobject.start()→t1.start();
- Run thread by run() method on thread object.
    - public void run()
- Use package **java.lang.Thread**
- **Two ways to create a thread:**
    - By extending Thread class
    - By implementing Runnable interface.

Conclusion:

Students successfully studied and implemented multithreading application.