|  |
| --- |
| **Subject: Operating System Sub Teacher:Prof. S.S.Shete**  **Class :- S.E Computer Engg Roll no :-**  **Practical No: Date:-** |

**Title:-**Write a program to implement concept of threading.

**Aim:**-Study to implement concept of threading.

**Definition:**-Thread is light weight process.

**Why thread is called light weight process:-**

Thread is generated from a process, hence it utilizes all the resources of a process. No extra or additional resources are required to allocate for thread. Hence it is called light weight process.

**Thread advantages:-**

* Context switching time is minimized.
* Thread support for efficient communication.
* Resource sharing is possible using threading.
* A thread provides concurrency within a process.

**Types of threading:-**

There are two types of threading are as follows:

**1. Single thread**

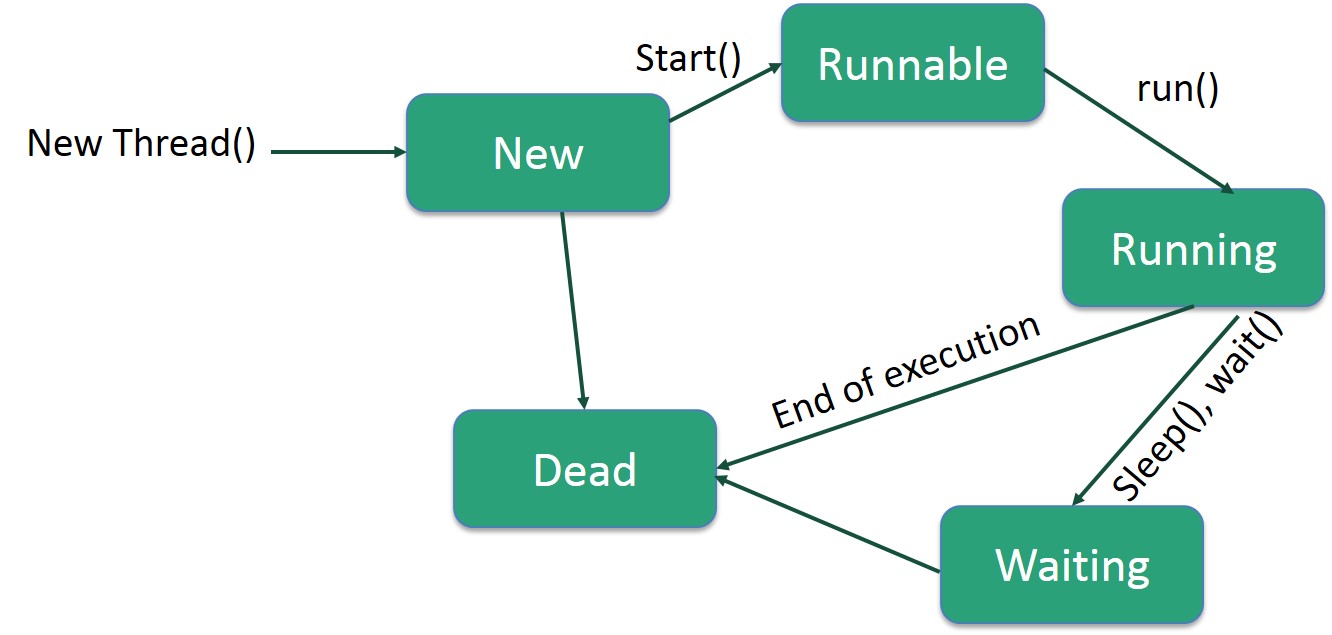
When our simple program starts one single thread begins running immediately .This is called our single main thread. The main threads create automatically when program is started. so imagine it as a single line going from the entry point of the application to its end.

**2.** **Multi-thread**

Java is a *multi-threaded programming language* which means we can develop multi-threaded program using Java. A multi-threaded program contains two or more parts that can run concurrently and each part can handle a different task at the same time making optimal use of the available resources specially when your computer has multiple CPUs. Imagine a tree: the whole app starts from 1 point, than it branches out more and more. Multi-threading enables you to write in a way where multiple activities can proceed concurrently in the same program.

**Life cycle of a thread:**

A thread goes through various stages in its life cycle. For example, a thread is born, started, runs, and then dies. The following diagram shows the complete life cycle of a thread.

**Fig:-Life cycle of thread**

**Following are the stages of the life cycle:**

* **New** − A new thread begins its life cycle in the new state. It remains in this state until the program starts the thread. It is also referred to as a **born thread**.
* **Runnable** − After a newly born thread is started, the thread becomes runnable. A thread in this state is considered to be executing its task.
* **Waiting** − Sometimes, a thread transitions to the waiting state while the thread waits for another thread to perform a task. A thread transitions back to the runnable state only when another thread signals the waiting thread to continue executing.
* **Timed Waiting** − A runnable thread can enter the timed waiting state for a specified interval of time. A thread in this state transitions back to the runnable state when that time interval expires or when the event it is waiting for occurs.
* **Terminated (Dead)** − A runnable thread enters the terminated state when it completes its task or otherwise terminates.

**Program: *\*****Write a Program to implement concept of Threading. \**/

class RunnableDemo implements Runnable

{

private Thread t;

private String threadName;

RunnableDemo( String name){

threadName = name;

System.out.println("Creating " + threadName );

}

public void run() {

System.out.println("Running " + threadName );

try {

for(int i = 4; i > 0; i--) {

System.out.println("Thread: " + threadName + ", " + i);

// Let the thread sleep for a while.

Thread.sleep(50);

}

} catch (InterruptedException e) {

System.out.println("Thread " + threadName + " interrupted.");

}

System.out.println("Thread " + threadName + " exiting.");

}

public void start ()

{

System.out.println("Starting " + threadName );

if (t == null)

{

t = new Thread (this, threadName);

t.start ();

}

}

}

public class Threading {

public static void main(String args[]) {

RunnableDemo R1 = new RunnableDemo( "Thread-1");

R1.start();

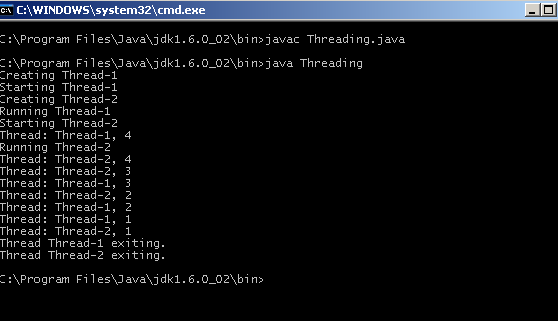
RunnableDemo R2 = new RunnableDemo( "Thread-2");

R2.start();

}

}

**Output:-**



**Conclusion:-**In this practical we have studies the implement concept of threading.