Thread Pool Example

In the following tutorial, we will look at a basic example of thread pool executor-FixedThreadPool.

Steps to be followed

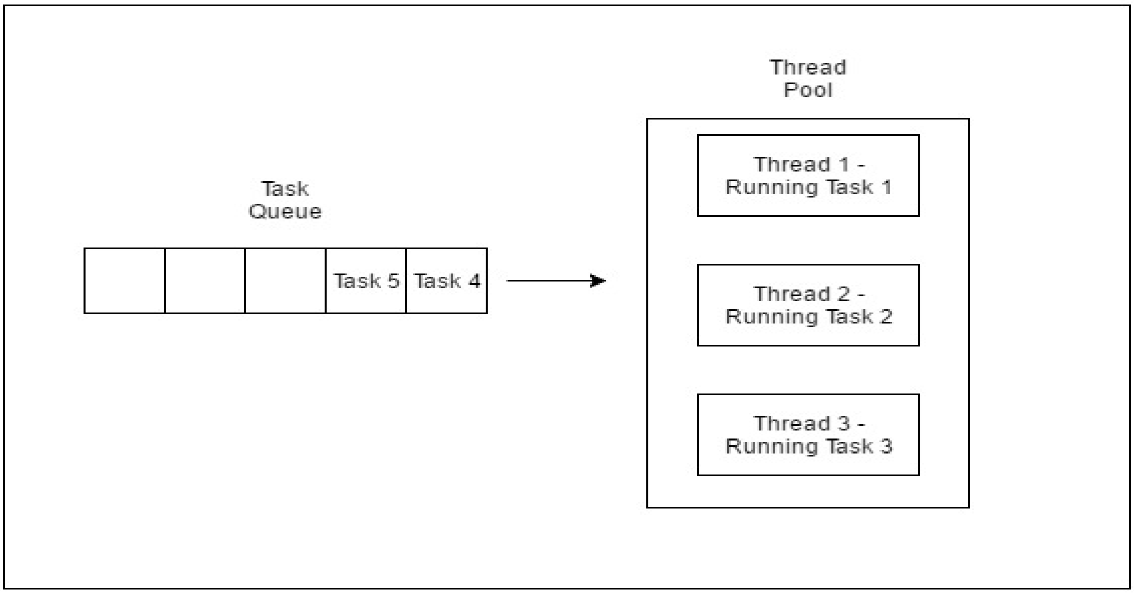
```
1. Create a task(Runnable Object) to execute
2. Create Executor Pool using Executors
3. Pass tasks to Executor Pool
4. Shutdown the Executor Pool
```

```java
// Java program to illustrate
// ThreadPool
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

// Task class to be executed (Step 1)
class Task implements Runnable
{
    private String name;

    public Task(String s)
    {
        name = s;
    }

    // Prints task name and sleeps for 1s
    // This Whole process is repeated 5 times
    public void run()
    {
        try
        {
            for (int i = 0; i<=5; i++)
            {
                if (i==0)
                {
                    Date d = new Date();
                    SimpleDateFormat ft = new SimpleDateFormat("hh:mm:ss");
                    System.out.println("Initialization Time for"
                            + " task name - "+ name +" = " +ft.format(d));
                    //prints the initialization time for every task
                }
                else
                {
                    Date d = new Date();
                    SimpleDateFormat ft = new SimpleDateFormat("hh:mm:ss");
                    System.out.println("Executing Time for task name - "+
                            name +" = " +ft.format(d));
                    // prints the execution time for every task
                }
                Thread.sleep(1000);
            }
            System.out.println(name+" complete");
        }

        catch(InterruptedException e)
        {
            e.printStackTrace();
        }
    }
}
public class Test
{
     // Maximum number of threads in thread pool
    static final int MAX_T = 3;

    public static void main(String[] args)
    {
        // creates five tasks
        Runnable r1 = new Task("task 1");
        Runnable r2 = new Task("task 2");
        Runnable r3 = new Task("task 3");
        Runnable r4 = new Task("task 4");
        Runnable r5 = new Task("task 5");

        // creates a thread pool with MAX_T no. of
        // threads as the fixed pool size(Step 2)
        ExecutorService pool = Executors.newFixedThreadPool(MAX_T);

        // passes the Task objects to the pool to execute (Step 3)
        pool.execute(r1);
        pool.execute(r2);
        pool.execute(r3);
        pool.execute(r4);
        pool.execute(r5);

        // pool shutdown ( Step 4)
        pool.shutdown();
    }
}
```

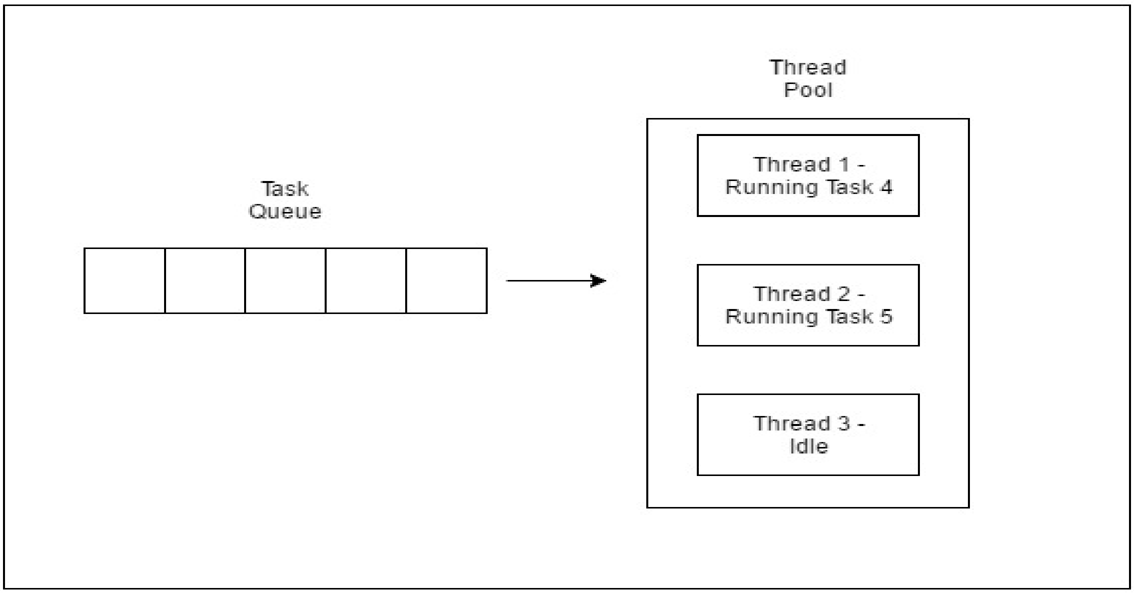
Sample Execution

```
Output:
Initialization Time for task name - task 2 = 02:32:56
Initialization Time for task name - task 1 = 02:32:56
Initialization Time for task name - task 3 = 02:32:56
Executing Time for task name - task 1 = 02:32:57
Executing Time for task name - task 2 = 02:32:57
Executing Time for task name - task 3 = 02:32:57
Executing Time for task name - task 1 = 02:32:58
Executing Time for task name - task 2 = 02:32:58
Executing Time for task name - task 3 = 02:32:58
Executing Time for task name - task 1 = 02:32:59
Executing Time for task name - task 2 = 02:32:59
Executing Time for task name - task 3 = 02:32:59
Executing Time for task name - task 1 = 02:33:00
Executing Time for task name - task 3 = 02:33:00
Executing Time for task name - task 2 = 02:33:00
Executing Time for task name - task 2 = 02:33:01
Executing Time for task name - task 1 = 02:33:01
Executing Time for task name - task 3 = 02:33:01
task 2 complete
task 1 complete
task 3 complete
Initialization Time for task name - task 5 = 02:33:02
Initialization Time for task name - task 4 = 02:33:02
Executing Time for task name - task 4 = 02:33:03
Executing Time for task name - task 5 = 02:33:03
Executing Time for task name - task 5 = 02:33:04
Executing Time for task name - task 4 = 02:33:04
Executing Time for task name - task 4 = 02:33:05
Executing Time for task name - task 5 = 02:33:05
Executing Time for task name - task 5 = 02:33:06
Executing Time for task name - task 4 = 02:33:06
Executing Time for task name - task 5 = 02:33:07
Executing Time for task name - task 4 = 02:33:07
task 5 complete
task 4 complete
```

As seen in the execution of the program, the task 4 or task 5 are executed only when a thread in the pool becomes idle. Until then, the extra tasks are placed in a queue.


Thread Pool
Task Queue
Task 5
Task 4
Thread 1 - Running Task 1
Thread 2 - Running Task 2
Thread 3 - Running Task 3

Thread Pool executing first three tasks



Thread Pool executing task 4 and 5