PRACTICAL 2

AIM: Demonstrate Priority Based Scheduling

Theory behind: Priority Based Scheduling

In the Shortest Job First scheduling algorithm, the priority of a program of a process is generally the inverse of the CPU burst time, i.e. the larger the burst time the lower is the priority of that process.

In case of priority scheduling the priority is not always set as the inverse of the CPU burst time, rather it can be internally or externally set, but yes the scheduling is done on the basis of priority of the process where the process which is most urgent is processed first, followed by the ones with lesser priority in order.

Processes with same priority are execute in FCFS manner.

The priority of process, when internally defined, can be decided based on memory requirements, time limits, number of open files, ratio of I/O burst to CPU burst etc.

Whereas, external priorities are set based on criteria outside the operating system, like the importance of the process.

Types of Priority Scheduling Algorithm

Priority scheduling can be of two types:

1. Preemptive Priority Scheduling: If the new process arrived at the ready queue has a higher priority than the currently running process, the CPU is preempted, which means the processing of the current process is stopped and the incoming new process with higher priority gets the CPU for its execution.

2. Non-Preemptive Priority Scheduling: In case of non-preemptive priority scheduling algorithm if a new process arrives with a higher priority than the current running process, the incoming process is put at the head of the ready queue, which means after the execution of the current process it will be processed.

CODE:

```
import java.io.*;
```

```
class job implements Runnable {
    int process_id, no_of_instr, priority_value, time_quantum;
    Thread t;

    job(int pid, int instr, int prio, int tq) {
        process_id = pid;
        no_of_instr = instr;
        time_quantum = tq;
        priority_value = prio;
        t = new Thread(this);
        t.setPriority(priority_value);
        t.start();
    }

    public void run() {
        try {
            for (int i = 1; i < no_of_instr; i++) {
                System.out.println("Executing instr no " + i + " of process " +
process_id);
                Thread.sleep(time_quantum);
            }
            System.out.println("Job " + process_id + " is over");
        } catch (InterruptedException e) {
            System.out.println("The job has been interrupted...");
        }
    }
```

```
}


class OS {
   public static void main(String args[]) {
      try {
         int process_id = 100, time_quantum = 100;
         BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
         System.out.println("Enter a user process starting number:");
         process_id = Integer.parseInt(br.readLine());
         System.out.println("Enter a time quantum(in millis):");
         time_quantum = Integer.parseInt(br.readLine());
         job j1 = new job(++process_id, 8, 9, time_quantum);
         job j2 = new job(++process_id, 8, 1, time_quantum);
         job j3 = new job(++process_id, 8, 2, time_quantum);
      } catch (Exception e) {
         System.out.println("Some process failed to complete...");
         System.out.println("Please contact system admin");
      }
   }
}
```

OUTPUT:

```
Enter a user process starting number:
100
Enter a time quantum(in millis):
20
Executing instr no 1 of process 101
```

Executing instr no 1 of process 102

Executing instr no 1 of process 103

Executing instr no 2 of process 101

Executing instr no 2 of process 102

Executing instr no 2 of process 103

Executing instr no 3 of process 102

Executing instr no 3 of process 103

Executing instr no 3 of process 101

Executing instr no 4 of process 101

Executing instr no 4 of process 102

Executing instr no 4 of process 103

Executing instr no 5 of process 101

Executing instr no 5 of process 103

Executing instr no 5 of process 102

Executing instr no 6 of process 101

Executing instr no 6 of process 102

Executing instr no 6 of process 103

Executing instr no 7 of process 101

Executing instr no 7 of process 103

Executing instr no 7 of process 102

Job 102 is over