Implementation of Multi-level feedback queue Scheduling using Scilab.

- Submitted by Sai Arun SR.

ABSTRACT-

Multi-level feedback queue is an extension of multi-level queue scheduling wherein two or more queues are executed based on the priority assigned to them. Each queue can implement any method of CPU scheduling (FCFS, SJF, Round-robin, etc). Multi-level feedback queue scheduling overcomes starvation for processes in lower-priority queues that require less execution/burst time, which is why it is preferred over multilevel scheduling in a multi-processing environment.

ALGORITHM-

In this example, we make use of 3 queues which are implemented based on Priority assigned to them.

- **HIGHEST PRIORITY:**

- ✓ Queue-1 to implement Round-robin scheduling with Time Quantum = 5 for a **single cycle.**
- ✓ Processes are queued based on their Arrival Time.
- ✓ Processes with burst time <=5 get executed.
- ✓ Other processes are added to the tail of the next queue to be executed with-

Remaining time(Pi) = burst time(Pi)-Time Quantum

Burst time(Pi) = Remaining time(Pi).

- **Queue-2 with Time Quantum = 8**

- ✓ Queue-2 with the next priority is scheduled in the same manner as Q1.
- ✓ The processes that could not complete execution are added to the tail of the next priority queue to be executed.

- **LOWEST PRIORITY:**

- ✓ Implements FCFS basis of scheduling.
- ✓ All the remaining processes complete their execution here.

SCILAB CODE:

```
clc

Q1=[],Q2=[],Q3=[];

n=input("Enter no of processes in HIGHEST PRIORITY queue");

for i=1:n

    Q1(i)=struct('pid',0,'AT',0,'BT',0,'TAT',0,'WT',0,'RT',0,'CT',0);

        /*SOME NOTATIONS FOR EACH PROCESS IN THE QUEUES*/

    // 'pid'- Process Number/ID, 'BT'- burst time, 'WT'- waiting time

  // 'TAT'- turnaround time, 'RT'- remaining time, 'CT'- Complete time.

end

r=1,time=0,tq1=5,tq2=8,flag=0,k=1,tot=0,wt=0,tat=0;

for i=1:n

Q1(i).pid=i;

Q1(i).AT=input("Enter   the   arrival   time   of   process   "+string(i)+":  ");
Q1(i).BT=input("Enter   the   burst   time   of   process   "+string(i)+":  ");
tot=tot+Q1(i).BT;

 Q1(i).RT=Q1(i).BT;            /*save INITIAL remaining time as burst time for each
                                                                          process*/

end

tmp=struct('PID',0,'AT',0,'BT',0,'TAT',0,'WT',0,'RT',0,'CT',0);

for i=1:n
```

```
          for j=i+1:n
             if Q1(i).AT > Q1(j).AT then
                tmp=Q1(i);
      Q1(i)=Q1(j);
      Q1(j)=tmp;
             end
       end
end
/*The ABOVE 10 Lines perform SORTING of Q1 based on Arrival time of the
processes.*/
 time=Q1(1).AT;      //Initial Total time assigned as least arrival time of the queued
                                                                              processes.
mprintf("\tProcess in first queue following RR with Time Quantum=5");
mprintf("\nProcess\t\tRT\t\tWT\t\tTAT\t\t");
for i=1:n
   if Q1(i).RT<=tq1 then
  time=time+Q1(i).RT;
  Q1(i).RT=0;
     Q1(i).WT=time-Q1(i).AT-Q1(i).BT; /*Calculating Waiting Time of the ith process
                                                                          in queue Q1*/
   wt=wt+Q1(i).WT;
   Q1(i).TAT=time-Q1(i).AT;                /*ith process time from arrival to execution
                                                                            completion*/
tat=tat+Q1(i).TAT;
mprintf("\n%d\t\t%d\t\t%d\t\t%d",Q1(i).pid,Q1(i).BT,Q1(i).WT,Q1(i).TAT);
```

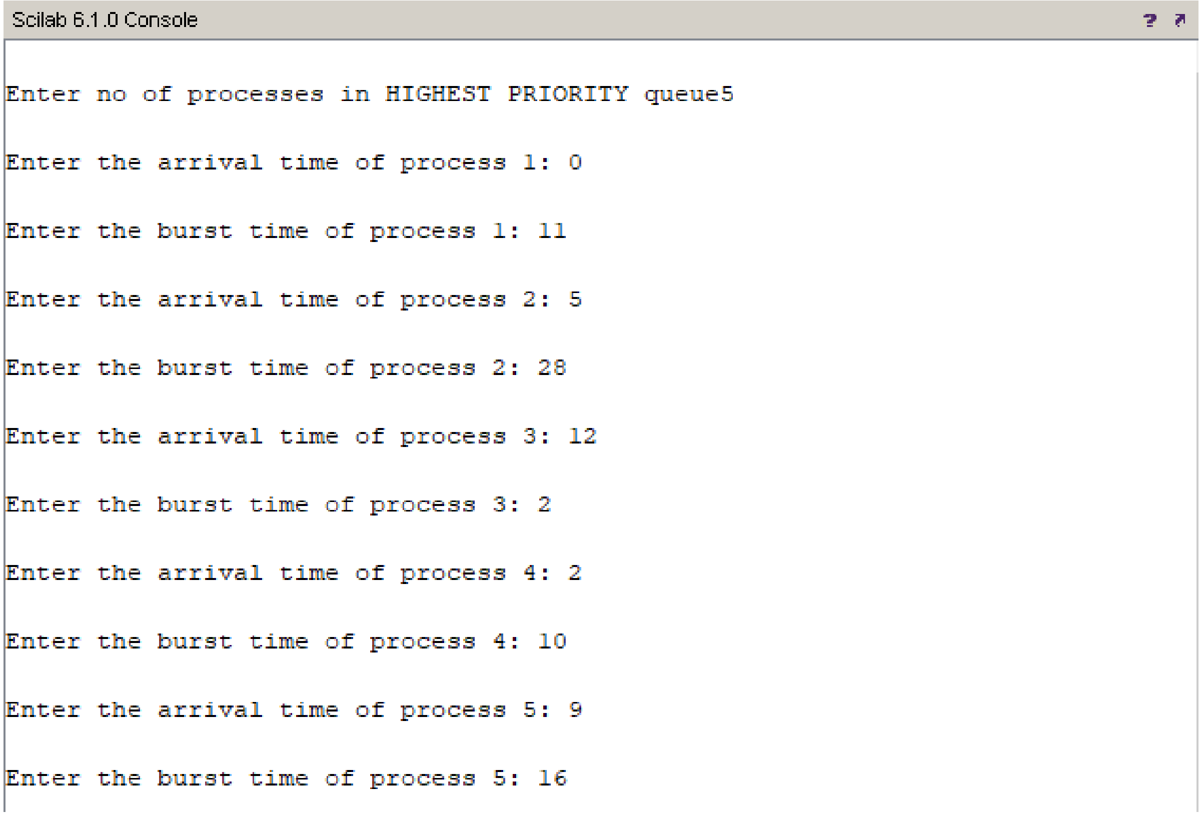
```
else          /*process moves to queue 2 with  qt=8*/
Q2(k)=struct('pid',0,'AT',0,'BT',0,'TAT',0,'WT',0,'RT',0,'CT',0);
Q2(k).WT=time;
time=time+tq1;
Q1(i).RT=Q1(i).RT-tq1;
Q2(k).BT=Q1(i).RT;
Q2(k).RT=Q2(k).BT;
Q2(k).pid=Q1(i).pid;
k=k+1;
flag=1;
  end
end
if flag==1 then
 mprintf("\n\nProcess in second queue following RR with Time Quantum=8");
mprintf("\nProcess\t\tRT\t\tWT\t\tTAT\t\t");
end
for i=1:k-1
    if Q2(i).RT<=tq2 then
            time=time+Q2(i).RT;     /*from arrival time of first process +BT of this process*/
  Q2(i).RT=0;
 Q2(i).WT=time-tq1-Q2(i).BT;
   wt=wt+Q2(i).WT;
Q2(i).TAT=time-Q2(i).AT;
tat=tat+Q2(i).TAT;
mprintf("\n%d\t\t%d\t\t%d\t\t%d",Q2(i).pid,Q2(i).BT,Q2(i).WT,Q2(i).TAT);
```

```
    else          /*process moves to queue 3 with FCFS*/
Q3(r)=struct('pid',0,'AT',0,'BT',0,'TAT',0,'WT',0,'RT',0,'CT',0);
Q3(r).AT=time;

time=time+tq2;

Q2(i).RT=Q2(i).RT-tq2;

   Q3(r).BT=Q2(i).RT;

Q3(r).RT=Q3(r).BT;

Q3(r).pid=Q2(i).pid;    r=r+1;    flag=2;

end

end

if(flag==2) then

mprintf("\n\n\tProcess in third queue following FCFS");
mprintf("\nProcess\t\tRT\t\tWT\t\tTAT\t\t");

end

for i=1:r-1

if i==1 then

Q3(i).CT=Q3(i).BT+time-tq1-tq2;

else       Q3(i).CT=Q3(i-1).CT+Q3(i).BT;

end

end

for i=1:r-1

Q3(i).TAT=Q3(i).CT;

tat=tat+Q3(i).TAT;

Q3(i).WT=Q3(i).TAT-Q3(i).BT;
```

```
wt=wt+Q3(i).WT;
mprintf("\n%d\t\t%d\t\t%d\t\t%d\t\t",Q3(i).pid,Q3(i).BT,Q3(i).WT,Q3(i).TAT);end

mprintf("\nAVG Execution time: %f",tot/n);

mprintf("\nAVG Waiting time: %f",wt/n);

mprintf("\nAVG Turnaround time: %f",tat/n);
```

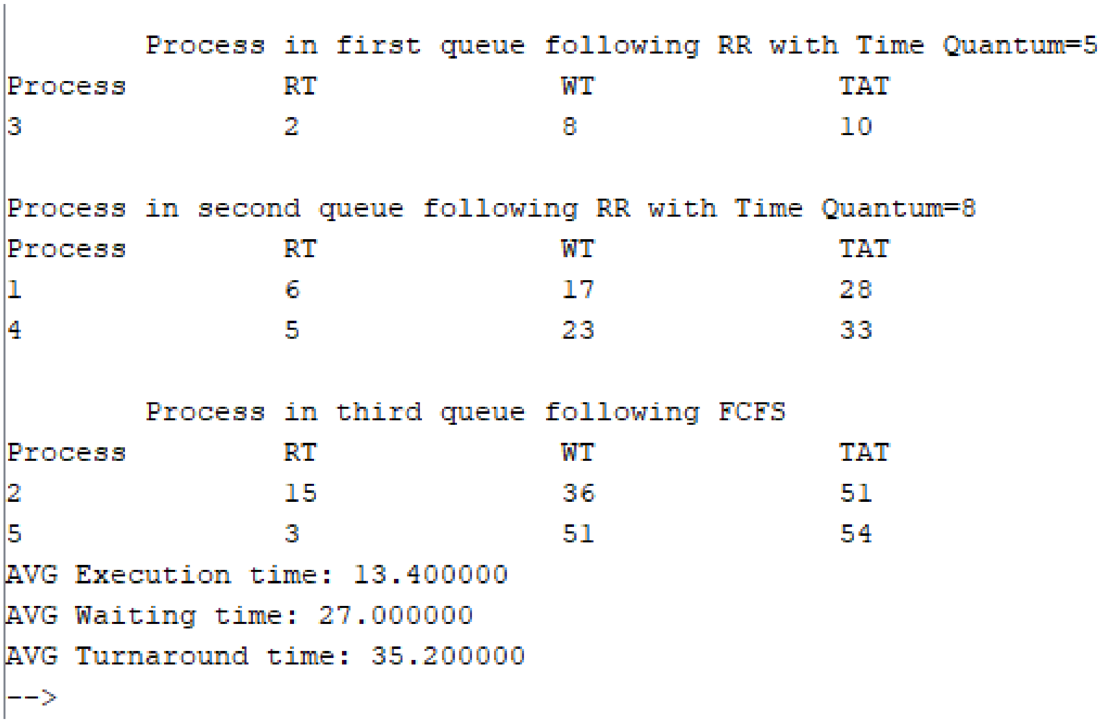
SCREENSHOTS-

INPUT-



```
Enter no of processes in HIGHEST PRIORITY queue5

Enter the arrival time of process 1: 0

Enter the burst time of process 1: 11

Enter the arrival time of process 2: 5

Enter the burst time of process 2: 28

Enter the arrival time of process 3: 12

Enter the burst time of process 3: 2

Enter the arrival time of process 4: 2

Enter the burst time of process 4: 10

Enter the arrival time of process 5: 9

Enter the burst time of process 5: 16
```

OUTPUT-



```
        Process in first queue following RR with Time Quantum=5
Process         RT              WT              TAT
3               2               8               10

Process in second queue following RR with Time Quantum=8
Process         RT              WT              TAT
1               6               17              28
4               5               23              33

        Process in third queue following FCFS
Process         RT              WT              TAT
2               15              36              51
5               3               51              54
AVG Execution time: 13.400000
AVG Waiting time: 27.000000
AVG Turnaround time: 35.200000
-->
```