**ROUND-ROBIN TIME SCHEDULING**

**STATEMENT:**

UNIX Operating system uses round robin time slicing with multilevel feedback for process scheduling. Assume that the processes arrive at random. assume that the processes (PCB’s)arrive at random. Assume that time slice is 3 units. Simulate Round robin time slicing with fixed and variable (Randomized) burst times. Run the processes for fixed amount of time and find the average waiting time, average turn-around time of the processes completed.

**EXPLANATION:**

Round robin scheduling Algorithm is designed especially for time sharing systems. Preemption is added to enable the system to switch between process. A small unit of time called time quantum/slice is defined. A time quantum is generally from 10-100(ms)in length. During time quantum each process is allowed to run if process is still running at the end of the quantum, the CPU is pre-empted and given to another process has blocked/finished before the quantum has elapsed, the CPU switching is done.

**ALGORITHM:**

1.Create an array rem\_bt[] to keep track of remaining

burst time of processes. This array is initially a

copy of bt[] (burst times array)

2- Create another array wt[] to store waiting times

of processes. Initialize this array as 0.

3- Initialize time : t = 0

4- Keep traversing the all processes while all processes

are not done. Do following for i&#39;th process if it is

160117733052 P.Srijay

not done yet.

a- If rem\_bt[i] &gt; quantum

(i) t = t + quantum

(ii) bt\_rem[i] -= quantum;

c- Else // Last cycle for this process

(i) t = t + bt\_rem[i];

(ii) wt[i] = t - bt[i]

(ii) bt\_rem[i] = 0;