***Lab No-10***

***Name of the lab:*****Implementation of Round Robin Scheduling Algorithm**

***Name:******Shuvo Biswas ID:******IT-16014***

***Objective:***

**(1) What is Round Robin Scheduling Algorithm ?**

**Ans:**

Round Robin is a CPU Scheduling Algorithm where each process is assigned a fixed time slot in a cyclic way.

* It is simple, easy to implement, and starvation-free as all processes get fair share of CPU.
* One of the most commonly used technique in CPU scheduling as a core.
* It is preemptive as processes are assigned CPU only for a fixed slice of time at most.
* The disadvantage of it is more overhead of context switching.

**(2) How to implementation in C?**

**Ans :**

**Source code:**

#include<stdio.h>

int main()

{

int i,j,n,time,remain,flag=0,ts;

int sum\_wait=0,sum\_turnaround=0,at[10],bt[10],rt[10];

printf("Enter no of Processes : ");

scanf("%d",&n);

remain=n;

for(i=0; i<n; i++)

{

printf("Enter arrival time and burst time for Process P%d :",i+1);

scanf("%d",&at[i]);

scanf("%d",&bt[i]);

rt[i]=bt[i];

}

printf("Enter time slice");

scanf("%d",&ts);

printf("\n\nProcess\t|Turnaround time|waiting time\n\n");

for(time=0,i=0; remain!=0;)

{

if(rt[i]<=ts && rt[i]>0)

{

time+=rt[i];

rt[i]=0;

flag=1;

}

else if(rt[i]>0)

{

rt[i]-=ts;

time+=ts;

}

if(rt[i]==0 && flag==1)

{

remain--;

printf("P[%d]\t|\t%d\t|\t%d\n",i+1,time-at[i],time-at[i]-bt[i]);

sum\_wait+=time-at[i]-bt[i];

sum\_turnaround+=time-at[i];

flag=0;

}

if(i==n-1)

i=0;

else if(at[i+1]<=time)

i++;

else i=0;

}

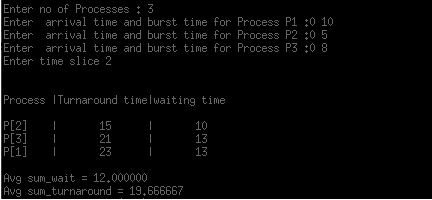
printf("\nAvg sum\_wait = %f\n",sum\_wait\*1.0/n);

printf("Avg sum\_turnaround = %f",sum\_turnaround\*1.0/n);

return 0;

}

**Output :**



**Conclusion :**

By doing this lab, I have implemented Round Robin scheduling algorithm.

Firstly I solve this algorithm in codeblocks during this time I faced many problems.

But later I solve this problem in my pc.