M.Rohith
3122 21 5001 085

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
(An Autonomous Institution, Affiliated to Anna University, Chennai)
SSN College of Engineering

Department of Computer Science and Engineering

UCS2412 – Operating Systems Laboratory

II Year CSE – A & B Section (IV Semester)

Academic Year 2022-23

Exercise – 4 – Scheduling Algorithms – Round Robin, Priority

Lab Exercise 4 Implementation of Scheduling Algorithms (– Round Robin, Priority)

Aim:

To develop a C program to implement the Scheduling Algorithms. (K4, CO1)

Scenario:

Consider a Traffic signal system, which is used to control the movement of vehicles and passengers to ensure smooth and safe traffic. The traffic light changes every 2 minutes and the vehicles arrive at different times at the signal junction.

a. Identify the suitable scheduling algorithm for this scenario and develop a solution to schedule the vehicles on a particular lane.
b. Compute the waiting time, turnaround time and response time and tabulate the results with the Gantt chart.

In the above scenario, let us assume the arrival of an ambulance or any VIP vehicles that overrule the traffic light system.

c. Identify the suitable scheduling algorithm for this scenario by giving preferences to the above vehicles and develop a solution to schedule them.
d. Compute the waiting time, turnaround time and response time and tabulate the results with the Gantt chart.
e. Analyze the algorithms and identify the best algorithm with respect to the scheduling criteria.

Algorithm:

1. Read the following
    a. Number of processes
    b. Process IDs
    c. Arrival time for each process
    d. Burst Time for each process
    e. Priority / Time quantum
2. Design a menu with Priority and Round Robin options
3. Upon selection of menu option apply the corresponding algorithm.
4. Compute the Turnaround Time, Average waiting Time for each of the algorithm.
5. Tabularize the results.
6. Display the Gantt Chart.

Sample Input & Output:

CPU SCHEDULING ALGORITHMS

1. ROUND ROBIN
2. PRIORITY
3. EXIT

Enter your option: 1

ROUND ROBIN CPU SCHEDULER

Number of Processes: 5

Time Quantum: 2

Process ID: P1

Arrival Time: 0

Burst Time: 4

-

-

-

-

Process ID: P5

Arrival Time: 6

Burst Time: 3

Output:

Process ID	Arrival Time	Burst Time	Turnaround Time	Waiting Time
P1	0	4	***	***
***	***	***	***	***
***				
***				
		Average	***	***

Want to Continue (Y/N): Y

CPU SCHEDULING ALGORITHMS

1. ROUND ROBIN
2. PRIORITY
3. EXIT

Enter your option: 2

PRIORITY CPU SCHEDULER

a. Non preemptive PRIORITY

b. Pre emptive PRIORITY

Enter your option: a

Number of Processes: 5

Process ID: P1

Arrival Time: 0

Burst Time: 4

Priority : 3

-

-

-

-

Process ID: P5

Arrival Time: 6

Burst Time: 3

Priority : 1

M.Rohith
3122 21 5001 085

Output:

Process ID	Arrival Time	Burst Time	Priority	Turnaround Time	Waiting Time
***	***	***	***	***	***
***	***	***	***	***	***
***					
***					
		Average		***	***

M.Rohith
3122 21 5001 085

Program Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>


float average(int arr[],int n)
{
    float avg;
    float sum=0;
    for(int i=0;i!=n;i++)
    {
        sum=sum+arr[i];
    }
    avg=sum/n;
    return avg;
}


void sort(int arr[],int n,int prio[])
{
    int temp;
    for(int i=0;i!=n;i++)
    {
        for(int j=i+1;j!=n;j++)
        {
            if(prio[arr[j]]<prio[arr[i]])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}


int in_queue(int n,int time,int at[],int bt[],int st[],int et[],int
q[],int exec[])
{
```

M.Rohith
3122 21 5001 085

```
    int count=0;
    for(int i=0;i!=n;i++)
    {
        if(at[i]<=time && exec[i]!=1)
        {
            q[count++]=i;
        }
    }
    return count;
}


void start_end_time(int n,int time,int at[],int bt[],int st[],int
et[],int q[],int exec[],int prio[])
{
    int m;
    int min;
    for(int i=0;i!=n;i++)
    {
        m=in_queue(n,time,at,bt,st,et,q,exec);
        sort(q,m,prio);
        min=q[0];
        st[min]=time;
        et[min]=time+bt[min];
        time=time+bt[min];
        exec[min]=1;
    }
}


void waiting_time(int n,int wt[],int st[],int at[])
{
    for(int i=0;i!=n;i++)
    {
        wt[i]=st[i]-at[i];
    }
}


void response_time(int n,int rt[],int st[],int at[])
{
    for(int i=0;i!=n;i++)
    {
```

M.Rohith
3122 21 5001 085

```
        rt[i]=st[i]-at[i];
    }
}


void turn_around_time(int n,int tat[],int wt[],int bt[])
{
    for(int i=0;i!=n;i++)
    {
        tat[i]=wt[i]+bt[i];
    }
}


void main()
{
    int option=0;
    while(option!=3)
    {
        printf("ENTER 1 FOR PRIORITY SCHEDULING\nENTER 2 FOR ROUND
ROBIN SCHEDULING\nENTER 3 TO EXIT\n");
        printf("ENTER THE OPTION:");
        scanf("%d",&option);
        if(option==1)
        {
            int n;
            int time=0;
            printf("ENTER THE NUMBER OF PROCESSES:");
            scanf("%d",&n);
            int pid[n];
            int at[n];
            int bt[n];
            int st[n];
            int et[n];
            int wt[n];
            int tat[n];
            int rt[n];
            int exec[n];
            int q[n];
            int prio[n];
            float avw,avt,avr;
            for(int i=0;i!=n;i++)
```

M.Rohith
3122 21 5001 085

```
            {
                exec[i]=0;
                printf("ENTER THE ARRIVAL TIME OF P%d:",i+1);
                scanf("%d",&at[i]);
                printf("ENTER THE BURST TIME OF P%d:",i+1);
                scanf("%d",&bt[i]);
                printf("ENTER THE PRIORITY OF P%d:",i+1);
                scanf("%d",&prio[i]);
                printf("\n");
            }
            start_end_time(n,time,at,bt,st,et,q,exec,prio);
            waiting_time(n,wt,st,at);
            response_time(n,rt,st,at);
            turn_around_time(n,tat,wt,bt);
            printf("process_id\tarrival_time\tburst_time\tstart_time
\tend_time\twait_time\ttat\tresponse_time\n");
            for(int i=0;i!=n;i++)
            {
                printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%
d\n",i+1,at[i],bt[i],st[i],et[i],wt[i],tat[i],rt[i]);
            }
            avw=average(wt,n);
            avt=average(tat,n);
            avr=average(rt,n);
            printf("AVERAGE WAITING TIME:%f\n",avw);
            printf("AVERAGE TURN AROUND TIME:%f\n",avt);
            printf("AVERAGE RESPONSE TIME:%f\n",avr);
        }
        else if(option==2)
        {
            int i, limit, total = 0, x, counter = 0, time_quantum;
            int wait_time = 0, turnaround_time = 0,
arrival_time[10], burst_time[10], temp[10];
            float average_wait_time, average_turnaround_time;
            printf("Enter Total Number of Processes:");
            scanf("%d", &limit);
            x = limit;
            for(i = 0; i < limit; i++)
            {
                printf("Enter Details of Process[%d]:", i +
1);
                printf("\nArrival Time:");
```

M.Rohith
3122 21 5001 085

```
                scanf("%d", &arrival_time[i]);
                printf("Burst Time:");
                scanf("%d", &burst_time[i]);
                temp[i] = burst_time[i];
            }
            printf("Enter Time Quantum::");
            scanf("%d", &time_quantum);
            printf("Process ID\t\tBurst Time\t Turnaround
Time\tWaiting Time\n");
            for(total = 0, i = 0; x != 0;)
            {
                if(temp[i] <= time_quantum && temp[i] > 0)
                {
                    total = total + temp[i];
                    temp[i] = 0;
                    counter = 1;
                }
                else if(temp[i] > 0)
                {
                    temp[i] = temp[i] - time_quantum;
                    total = total + time_quantum;
                }
                if(temp[i] == 0 && counter == 1)
                {
                    x--;
                    printf("Process[%d]\t\t%d\t\t%d\t\t\t %d\n", i +
1, burst_time[i], total -
                    arrival_time[i], total - arrival_time[i] -
burst_time[i]);
                    wait_time = wait_time + total - arrival_time[i]
- burst_time[i];
                    turnaround_time = turnaround_time + total -
arrival_time[i];
                    counter = 0;
                }
                if(i == limit - 1)
                {
                    i = 0;
                }
                else if(arrival_time[i + 1] <= total)
                {
                    i++;
```

M.Rohith
3122 21 5001 085

```
                }
                else
                {
                    i = 0;
                }
            }
            average_wait_time = wait_time * 1.0 / limit;
            average_turnaround_time = turnaround_time * 1.0 / limit;
            printf("Average Waiting Time:%f\n", average_wait_time);
            printf("Avg Turnaround Time:%f\n",
average_turnaround_time);
        }
        else
        {
            break;
        }
    }
}
```

Ex- no:4 M.Rohith
Date: 06-04-2023 3122 21 5001 085

Output:

```
PS C:\Rohith\Backup\Desktop\SEM 4\Operating Systems lab\Exercise-4> gcc Assignment_4.c -o run
PS C:\Rohith\Backup\Desktop\SEM 4\Operating Systems lab\Exercise-4> ./run
ENTER 1 FOR PRIORITY SCHEDULING
ENTER 2 FOR ROUND ROBIN SCHEDULING
ENTER 3 TO EXIT
ENTER THE OPTION:1
ENTER THE NUMBER OF PROCESSES:4
ENTER THE ARRIVAL TIME OF P1:0
ENTER THE BURST TIME OF P1:8
ENTER THE PRIORITY OF P1:3

ENTER THE ARRIVAL TIME OF P2:1
ENTER THE BURST TIME OF P2:4
ENTER THE PRIORITY OF P2:1

ENTER THE ARRIVAL TIME OF P3:2
ENTER THE BURST TIME OF P3:9
ENTER THE PRIORITY OF P3:4

ENTER THE ARRIVAL TIME OF P4:3
ENTER THE BURST TIME OF P4:5
ENTER THE PRIORITY OF P4:2

process_id      arrival_time    burst_time      start_time      end_time        wait_time       tat     response_time
P1              0               8               0               8               0               8               0
P2              1               4               8               12              7               11              7
P3              2               9               17              26              15              24              15
P4              3               5               12              17              9               14              9
AVERAGE WAITING TIME:7.750000
AVERAGE TURN AROUND TIME:14.250000
AVERAGE RESPONSE TIME:7.750000
```

```
ENTER 1 FOR PRIORITY SCHEDULING
ENTER 2 FOR ROUND ROBIN SCHEDULING
ENTER 3 TO EXIT
ENTER THE OPTION:2
Enter Total Number of Processes:3
Enter Details of Process[1]:
Arrival Time:0
Burst Time:24
Enter Details of Process[2]:
Arrival Time:10
Burst Time:3
Enter Details of Process[3]:
Arrival Time:15
Burst Time:3
Enter Time Quantum::4
Process ID              Burst Time       Turnaround Time        Waiting Time
Process[2]              3               5                       2
Process[3]              3               3                       0
Process[1]              24              30                      6
Average Waiting Time:2.666667
Avg Turnaround Time:12.666667
ENTER 1 FOR PRIORITY SCHEDULING
ENTER 2 FOR ROUND ROBIN SCHEDULING
ENTER 3 TO EXIT
ENTER THE OPTION:3
```