

## LAB ASSIGNMENT 3

### OPERATING SYSTEMS (UCS303)

**Q1,** Round Robin Scheduling

**CODE:**

```
#include<iostream>
using namespace std;

void waitingtime(int process[], int n,int btime[], int wtime[], int quantum)
{
    int rem_btime[n]; //remaining burst time
    for (int i = 0 ; i < n ; i++)
        rem_btime[i] = btime[i];

    int t = 0;
    while (1)
    {
        bool done = true;

        for (int i = 0 ; i < n; i++)
        {
            if (rem_btime[i] > 0)
            {
                done = false; // There is a pending process

                if (rem_btime[i] > quantum)
                {
                    t += quantum;
                    rem_btime[i] -= quantum;
                }

                else
                {

```

```

        t = t + rem_btime[i];

        wtime[i] = t - btime[i];

        // As the process gets fully executed
        // make its remaining burst time = 0
        rem_btime[i] = 0;
    }
}

if (done == true)
    break;
}
}

void turnarounds_time(int process[], int n,int btime[], int wtime[], int ttime[])
{
    // calculating turnaround time by adding bt[i] + wt[i]
    for (int i = 0; i < n ; i++)
        ttime[i] = btime[i] + wtime[i];
}

// Function to calculate average time
void averagetime(int process[], int n, int btime[],int quantum)
{
    int wtime[n], ttime[n], total_wtime = 0, total_ttime = 0;

    // Function to find waiting time of all processes
    waitingtime(process, n, btime, wtime, quantum);

    // Function to find turn around time for all processes
    turnarounds_time(process, n, btime, wtime, ttime);

    // Display processes along with all details
    cout<<"\nprocesses  "<<"burst time  "<<"waiting time  "<<"turn around time\n ";

```

```

    // Calculate total waiting time and total turn
    // around time
    for (int i=0; i<n; i++)
    {
        total_wtime = total_wtime + wtime[i];
        total_tatime = total_tatime + tatime[i];
        cout<<"  "<<i+1<<"\t\t"<<btime[i]<<" \t  " <<wtime[i]<<"
\t\t  "<<tatime[i]<<endl;
    }

    cout << "Average waiting time = "
        << (float)total_wtime / (float)n;
    cout << "\nAverage turn around time = "
        << (float)total_tatime / (float)n;
}

int main()
{
    int processes[] = { 1, 2, 3};
    int n = sizeof processes / sizeof processes[0];

    int btime[] = {10, 5, 8};

    int quantum = 2;
    averagetime(processes, n, btime, quantum);
    return 0;
}

```

## OUTPUT:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\Jayan TIET\5th sem\UCS303\assignment 3> cd "e:\Jayan TIET\5th sem\UCS303\assignment 3\"; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile }; if ($?) { .\tempCodeRunnerFile }

processes burst time waiting time turn around time
1          10          13          23
2           5          10          15
3           8          13          21
Average waiting time = 12
Average turn around time = 19.6667
PS E:\Jayan TIET\5th sem\UCS303\assignment 3>
```

## Q2. Shortest Job First Scheduling

### CODE:

```
#include<bits/stdc++.h>
#include<iostream>
using namespace std;

struct process
{
    int id;
    int btime;
    int wtime;
};

void turnaroundsime(process proc[],int n,int wtime[],int ttime[])
{
    for(int i=0;i<n;i++)
    {
        ttime[i]=proc[i].btime+wtime[i];
    }
}

void waitingtime(process proc[],int n,int wtime[])
{
    wtime[0]=0;
```

```

    for(int i=1;i<n;i++)
    {
        wtime[i]=proc[i-1].btime+wtime[i-1];
    }
}

void averagetime(process proc[],int n)
{
    int wtime[n],tatetime[n],total_wtime=0,total_tatetime=0;

    waitingtime(proc,n,wtime);

    turnaroundtime(proc,n,wtime,tatetime);

    cout<<"\nprocesses"<<"burst time"<<"waiting time"<<"turn around time\n";

    for(int i=0;i<n;i++)
    {
        total_wtime=total_wtime+wtime[i];
        total_tatetime=total_tatetime+tatetime[i];

        cout<<"  "<<proc[i].id<<"\t\t"<<proc[i].btime <<" \t  "
<<wtime[i]<<" \t\t  "<<tatetime[i]<<endl;
    }

    cout<<"average waiting time is "<<(float)total_wtime/(float)n<<endl;
    cout<<"average turn around time is "<<(float)total_tatetime/(float)n;
}

int main()
{
    process proc[]={1,10,2},{2,5,0},{3,8,1}};
    int n=sizeof proc / sizeof proc[0];

```

```

    averagetime(proc,n);

    return 0;
}

```

## OUTPUT:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER
PS E:\Jayan TIET\5th sem\UCS303\assignment 3>

> cd "e:\Jayan TIET\5th sem\UCS303\assignment 3\" ; if ($?) { g++ 2.cpp -o 2 } ; if ($?) { .\2 }

processesburst timewaiting timeturn around time
1           10           0           10
2            5           10           15
3            8           15           23
average waiting time is 8.33333
average turn around time is 16
PS E:\Jayan TIET\5th sem\UCS303\assignment 3>

```

## Q3. First Come First Serve Scheduling

### CODE:

```

def waitingtime(process,n,btime,wtime):
    wtime[0]=0 #initially waiting time is 0

    for i in range(1,n):
        wtime[i]=btime[i-1]+wtime[i-1]

def turnaroundtime(process,n,btime,wtime,tatime):
    for i in range(n):
        tatime[i]=btime[i]+wtime[i]

def averagetime(process,n,btime):
    wtime=[0]*n

```

```

tatetime=[0]*n
total_wtime=0
total_tatetime=0

waitingtime(process,n,btime,wtime)
turnaroundtime(process,n,btime,wtime,tatetime)

print("Burst time"+ "Waiting time"+"Turn around time")

for i in range(n):
    total_wtime=total_wtime+wtime[i]
    total_tatetime=total_tatetime+tatetime[i]

    print("
"+str(i+1)+"\t\t"+str(btime[i])+"\t"+str(wtime[i])+"\t\t"+str(tatetime[i]))

    print("Average waiting time= "+str(total_wtime/n))

    print("Average turn around time = "+str(total_tatetime/n))

process=[1,2,3]
n=len(process)
btime=[10,5,8]

averagetime(process,n,btime)

```

## OUTPUT:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
PS E:\Jayan TIET\5th sem\UCS303\assignment 3>

> python -u "e:\Jayan TIET\5th sem\UCS303\assignment 3\3.py"

Burst timeWaiting timeTurn around time
1      10      0      10
2       5     10     15
3       8     15     23
Average waiting time= 8.333333333333334
Average turn around time = 16.0
PS E:\Jayan TIET\5th sem\UCS303\assignment 3>
```

## Q4. Priority Scheduling

### CODE:

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;

// structure to input process id burst time and priority of each process

struct process
{
    int id;
    int btime;
    int priority;
};

//comparing different processes on basis of process id

bool comparison(process a, process b)
{
    return(a.priority>b.priority);
}

// to find the waiting time of each process
```



```

void waitingtime(process proc[],int n,int wtime[])
{
    //initially wt is 0
    wtime[0]=0;

    for(int i=1;i<n;i++)
    {
        wtime[i]=proc[i-1].btime+wttime[i-1];
    }
}

//to find turn around time

void turnaroundtime(process proc[],int n,int wtime[],int tatetime[])
{
    for(int i=0;i<n;i++)
    {
        tatetime[i]=proc[i].btime+wttime[i];
    }
}

void averagetime(process proc[],int n)
{
    int wtime[n],tatime[n],total_wtime=0,total_tatime=0;

    waitingtime(proc,n,wttime);

    turnaroundtime(proc,n,wttime,tatime);

    cout<<"\nprocesses  "<<"burst time  "<<"waiting time  "<<"turn around
time\n ";

    //for calculation of total waiting and turn around time

```

```

    for(int i=0;i<n;i++)
    {
        total_wtime=total_wtime+ wtime[i];
        total_tatime=total_tatime+tatime[i];

        cout<<" "<<proc[i].id<<"\t\t"<<proc[i].btime <<" \t "
<<wtime[i]<<" \t\t " <<tatime[i]<<endl;
    }

    cout<<"\nAverage time is "<<(float)total_wtime/(float)n;

    cout<<"\naverage turn around time "<<(float)total_tatime/(float)n;
}

//for priority scheduling

void priorityscheduling(process proc[],int n)
{
    sort(proc, proc+n ,comparison);
    cout<<"order of execution is "<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<proc[i].id<<" ";
    }

    averagetime(proc,n);
}

int main()
{
    process proc[]={1,10,2},{2,5,0},{3,8,1}};
    int n=sizeof proc / sizeof proc[0];
    priorityscheduling(proc,n);
    return 0;
}

```

## OUTPUT:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Code + - [ ] [ ] ^ x

order of execution is
1 3 2
processes burst time waiting time turn around time
1         10         0         10
3          8         10         18
2          5         18         23

Average time is 9.33333
average turn around time 17
PS E:\Jayan TIET\5th sem\UCS303\assignment 3>
```