**Indian Institute of Information Technology Vadodara**

**CS266 Operating System Lab**

**Lab Assignment 4**

**Roll No. 201951105 Name: Nishant Andoriy**

**Problem 1**

Implement following CPU scheduling algorithms:

• First Come First Serve

• Shortest Job First

• Priority Based Scheduling

Find the following performance objectives:

• Average Waiting Time

• Average Turnaround Time

• CPU efficiency

**Input:-**

• Number of Processes

• Process Id

• Arrival time of all the processes

**–** All arrive at the same time (say 0).

**–** All arrive at different time

• Burst time of all the processes

**Output:-**

Table 1: Output Table

|  |  |  |  |
| --- | --- | --- | --- |
| PID Arrival Time | Burst Time | Waiting Time | Turnaround Time |
|  |  |  |  |

• Average Turnaround Time:-

• Average Waiting Time:-

1

**• First Come First Serve**

**Ans:-**

#include<iostream>

using namespace std;

// Function to find the waiting time for all

// processes

void findWaitingTime(int processes[], int n, int bt[],

 int wt[], int at[])

{

 int service\_time[n];

 service\_time[0] = 0;

 wt[0] = 0;

 // calculating waiting time

 for (int i = 1; i < n ; i++)

 {

 // Add burst time of previous processes

 service\_time[i] = service\_time[i-1] + bt[i-1];

 wt[i] = service\_time[i] - at[i];

 if (wt[i] < 0)

 wt[i] = 0;

 }

}

// Function to calculate turn around time

void findTurnAroundTime(int processes[], int n, int bt[],

 int wt[], int tat[])

{

 // Calculating turnaround time by adding bt[i] + wt[i]

 for (int i = 0; i < n ; i++)

 tat[i] = bt[i] + wt[i];

}

// Function to calculate average waiting and turn-around

// times.

void findavgTime(int processes[], int n, int bt[], int at[])

{

 int wt[n], tat[n];

 // Function to find waiting time of all processes

 findWaitingTime(processes, n, bt, wt, at);

 // Function to find turn around time for all processes

 findTurnAroundTime(processes, n, bt, wt, tat);

 // Display processes along with all details

 cout << "Processes " << " Burst Time " << " Arrival Time "

 << " Waiting Time " << " Turn-Around Time \n";

 int total\_wt = 0, total\_tat = 0;

 for (int i = 0 ; i < n ; i++)

 {

 total\_wt = total\_wt + wt[i];

 total\_tat = total\_tat + tat[i];

 int compl\_time = tat[i] + at[i];

 cout << " " << i+1 << "\t\t" << bt[i] << "\t\t"

 << at[i] << "\t\t" << wt[i] << "\t\t "

 << tat[i] << endl;

 }

 cout << "Average waiting time = "

 << (float)total\_wt / (float)n;

 cout << "\nAverage turn around time = "

 << (float)total\_tat / (float)n;

 cout << "\nAverage turn around time = "

 << (float)total\_tat / (float)n;

}

// Driver code

int main()

{

 cout << "Enter total number of processes:\n";

 int n;

 cin>>n;

 int processes[n];

 int burst\_time[n],arrival\_time[n];

 cout << "Enter process ID, arrival time and burst time:\n";

 for(int i=0;i<n;i++)

 {

 cin>>processes[i]>>arrival\_time[i]>>burst\_time[i];

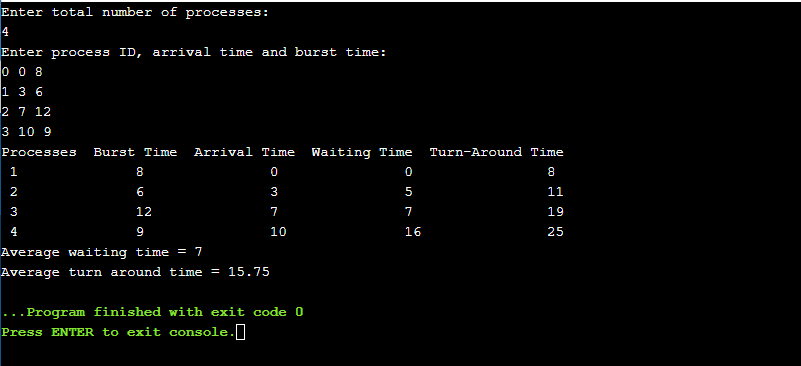
}

 findavgTime(processes, n, burst\_time, arrival\_time);

 return 0;

}

**Output:-**



**• Shortest Job First**

**Ans:-**

#include<iostream>

using namespace std;

int mat[10][6];

void swap(int \*a, int \*b)

{

 int temp = \*a;

 \*a = \*b;

 \*b = temp;

}

void arrangeArrival(int num, int mat[][6])

{

 for(int i=0; i<num; i++)

 {

 for(int j=0; j<num-i-1; j++)

 {

 if(mat[j][1] > mat[j+1][1])

 {

 for(int k=0; k<5; k++)

 {

 swap(mat[j][k], mat[j+1][k]);

 }

 }

 }

 }

}

void completionTime(int num, int mat[][6])

{

 int temp, val;

 mat[0][3] = mat[0][1] + mat[0][2];

 mat[0][5] = mat[0][3] - mat[0][1];

 mat[0][4] = mat[0][5] - mat[0][2];

 for(int i=1; i<num; i++)

 {

 temp = mat[i-1][3];

 int low = mat[i][2];

 for(int j=i; j<num; j++)

 {

 if(temp >= mat[j][1] && low >= mat[j][2])

 {

 low = mat[j][2];

 val = j;

 }

 }

 mat[val][3] = temp + mat[val][2];

 mat[val][5] = mat[val][3] - mat[val][1];

 mat[val][4] = mat[val][5] - mat[val][2];

 for(int k=0; k<6; k++)

 {

 swap(mat[val][k], mat[i][k]);

 }

 }

}

int main()

{

 int num, temp;

 int wait=0,tat=0;

 cout<<"Enter number of Process: ";

 cin>>num;

 cout<<"...Enter the process ID...\n";

 for(int i=0; i<num; i++)

 {

 cout<<"...Process "<<i+1<<"...\n";

 cout<<"Enter Process Id: ";

 cin>>mat[i][0];

 cout<<"Enter Arrival Time: ";

 cin>>mat[i][1];

 cout<<"Enter Burst Time: ";

 cin>>mat[i][2];

 }

 cout<<"Before Arrange...\n";

 cout<<"Process ID\tArrival Time\tBurst Time\n";

 for(int i=0; i<num; i++)

 {

 cout<<mat[i][0]<<"\t\t"<<mat[i][1]<<"\t\t"<<mat[i][2]<<"\n";

 }

 arrangeArrival(num, mat);

 completionTime(num, mat);

 cout<<"Final Result...\n";

 cout<<"Process ID\tArrival Time\tBurst Time\tWaiting Time\tTurnaround Time\n";

 for(int i=0; i<num; i++)

 {

cout<<mat[i][0]<<"\t\t"<<mat[i][1]<<"\t\t"<<mat[i][2]<<"\t\t"<<mat[i][4]<<"\t\t"<<mat[i][5]<<"

\n";

wait=wait+mat[i][4];

tat=tat+mat[i][5];

 }

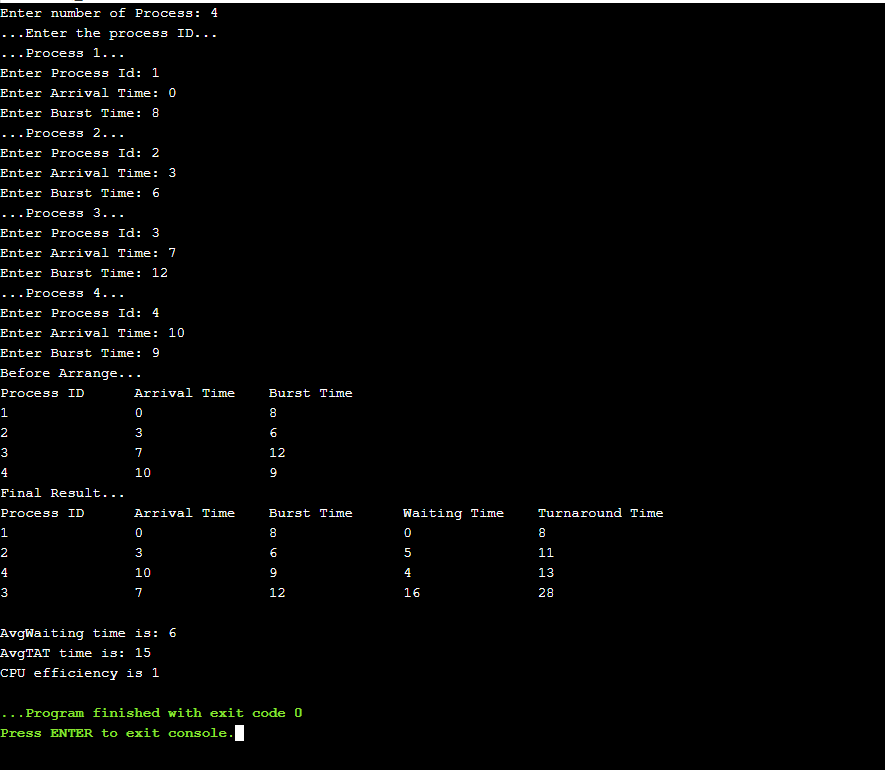
 cout << "\nAvgWaiting time is: " << wait/num;

 cout << "\nAvgTAT time is: " << tat/num;

 cout << "\nCPU efficiency is 1";

}

**Output:-**



• Priority Based Scheduling

Ans:-

#include<iostream>

using namespace std;

int main()

{

    int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg\_wt,avg\_tat;

    cout<<"Enter Total Number of Process:";

    cin>>n;

    cout<<"\nEnter Burst Time and Priority\n";

    for(i=0;i<n;i++)

    {

        cout<<"\nP["<<i+1<<"]\n";

        cout<<"Burst Time:";

        cin>>bt[i];

        cout<<"Priority:";

        cin>>pr[i];

        p[i]=i+1;           //contains process number

    }

    //sorting burst time, priority and process number in ascending order using selection sort

    for(i=0;i<n;i++)

    {

        pos=i;

        for(j=i+1;j<n;j++)

        {

            if(pr[j]<pr[pos])

                pos=j;

        }

        temp=pr[i];

        pr[i]=pr[pos];

        pr[pos]=temp;

        temp=bt[i];

        bt[i]=bt[pos];

        bt[pos]=temp;

        temp=p[i];

        p[i]=p[pos];

        p[pos]=temp;

    }

    wt[0]=0;            //waiting time for first process is zero

    //calculate waiting time

    for(i=1;i<n;i++)

    {

        wt[i]=0;

        for(j=0;j<i;j++)

            wt[i]+=bt[j];

        total+=wt[i];

    }

    avg\_wt=total/n;      //average waiting time

    total=0;

    cout<<"\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time";

    for(i=0;i<n;i++)

    {

        tat[i]=bt[i]+wt[i];     //calculate turnaround time

        total+=tat[i];

        cout<<"\nP["<<p[i]<<"]\t\t  "<<bt[i]<<"\t\t    "<<wt[i]<<"\t\t\t"<<tat[i];

    }

    avg\_tat=total/n;     //average turnaround time

    cout<<"\n\nAverage Waiting Time="<<avg\_wt;

    cout<<"\nAverage Turnaround Time="<<avg\_tat;

    return 0;

}

**Output:-**

