**Collage : Vishwakarma Institute of Technology**

**Course Name : Operating System**

**Name : Vedika Vikas Sontakke**

**Roll no : 37**

**PRN NO 12220206**

**Assignment No – 3**

**Scheduling algorithm**

1. **FIRST COME FIRST SERVE**

**Program :**

// First Come, First Served (FCFS) is a scheduling algorithm that schedules processes based on their arrival time.

// The process that arrives first will be executed first, and so on.

// Here is an example implementation of the FCFS scheduling algorithm in C:

#include<stdio.h>

int main()

{

    int  p[10],at[10],bt[10],ct[10],tat[10],wt[10],i,j,temp=0,n;

    float awt=0,atat=0;

    printf("enter no of proccess you want:");

    scanf("%d",&n);

    for(i=0;i<n;i++)

    {

       printf("enter process no :");

       scanf("%d",&p[i]);

       printf("enter arrival time for process %d:",p[i]);

       scanf("%d",&at[i]);

       printf("enter burst time for process %d:",p[i]);

       scanf("%d",&bt[i]);

    }

    for(i=0;i<n;i++)

    {

     for(j=0;j<(n-i);j++)

    {

      if(at[j]>at[j+1])

     {

        temp=p[j+1];

        p[j+1]=p[j];

        p[j]=temp;

        temp=at[j+1];

        at[j+1]=at[j];

        at[j]=temp;

        temp=bt[j+1];

        bt[j+1]=bt[j];

        bt[j]=temp;

      }

     }

    }

    /\* calculating 1st ct \*/

    ct[0]=at[0]+bt[0];

    /\* calculating 2 to n ct \*/

    for(i=1;i<n;i++)

    {

      //when proess is ideal in between i and i+1

      temp=0;

     if(ct[i-1]<at[i]) temp=at[i]-ct[i-1];

     ct[i]=ct[i-1]+bt[i]+temp;

    }

    /\* calculating tat and wt \*/

    printf("\np\t A.T\t B.T\t C.T\t TAT\t WT");

    for(i=0;i<n;i++)

    {

        tat[i]=ct[i]-at[i];

        wt[i]=tat[i]-bt[i];

        atat+=tat[i];

        awt+=wt[i];

    }

    atat=atat/n;

    awt=awt/n;

    for(i=0;i<n;i++)

      printf("\nP%d\t %d\t %d\t %d \t %d \t %d",p[i],at[i],bt[i],ct[i],tat[i],wt[i]);

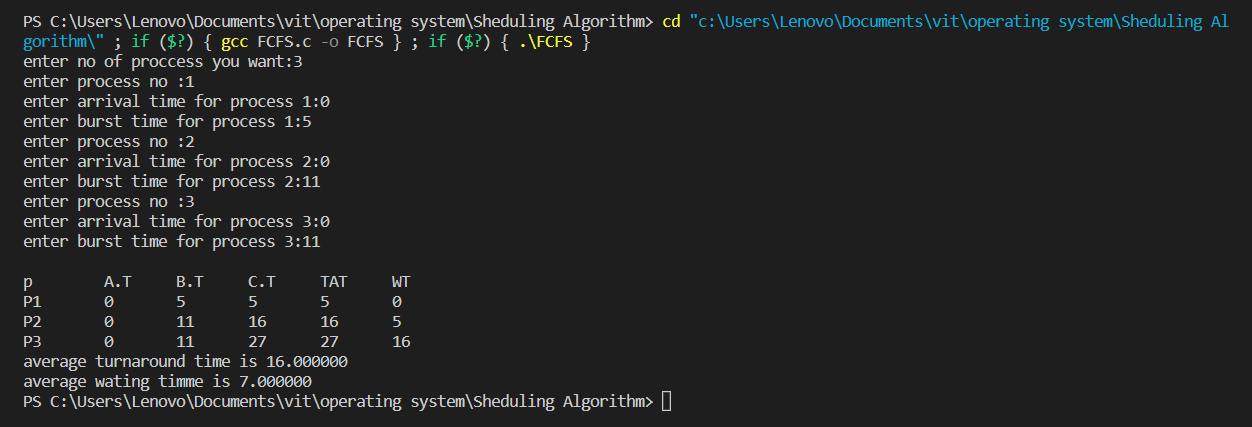
    printf("\naverage turnaround time is %f",atat);

    printf("\naverage wating timme is %f",awt);

    return 0;

}

**Output :**



1. **Priority basic scheduling**

/\*

 \* C program to implement priority scheduling

 \*/

#include <stdio.h>

//Function to swap two variables

void swap(int \*a,int \*b)

{

    int temp=\*a;

    \*a=\*b;

    \*b=temp;

}

int main()

{

    int n;

    printf("Enter Number of Processes: ");

    scanf("%d",&n);

    // b is array for burst time, p for priority and index for process id

    int b[n],p[n],index[n];

    for(int i=0;i<n;i++)

    {

        printf("Enter Burst Time and Priority Value for Process %d: ",i+1);

        scanf("%d %d",&b[i],&p[i]);

        index[i]=i+1;

    }

    for(int i=0;i<n;i++)

    {

        int a=p[i],m=i;

        //Finding out highest priority element and placing it at its desired position

        for(int j=i;j<n;j++)

        {

            if(p[j] > a)

            {

                a=p[j];

                m=j;

            }

        }

        //Swapping processes

        swap(&p[i], &p[m]);

        swap(&b[i], &b[m]);

        swap(&index[i],&index[m]);

    }

    // T stores the starting time of process

    int t=0;

    //Printing scheduled process

    printf("Order of process Execution is\n");

    for(int i=0;i<n;i++)

    {

        printf("P%d is executed from %d to %d\n",index[i],t,t+b[i]);

        t+=b[i];

    }

    printf("\n");

    printf("Process Id     Burst Time   Wait Time    TurnAround Time\n");

    int wait\_time=0;

    for(int i=0;i<n;i++)

    {

        printf("P%d          %d          %d          %d\n",index[i],b[i],wait\_time,wait\_time + b[i]);

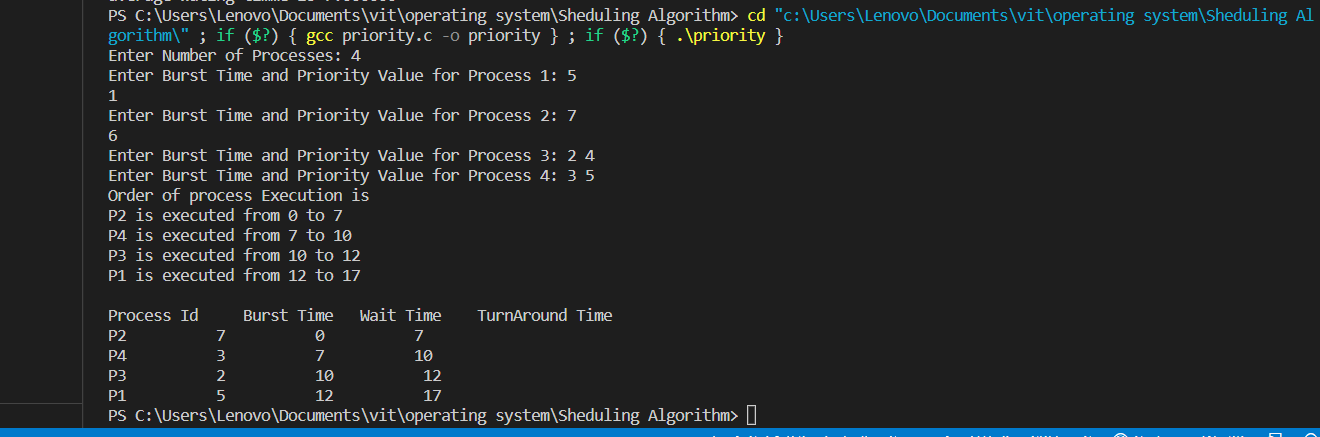
        wait\_time += b[i];

    }

    return 0;

}

**Output :**

****

1. **Round Robin**

**Program :**

#include<stdio.h>

#include<conio.h>

void main()

{

    // initlialize the variable name

    int i, NOP, sum=0,count=0, y, quant, wt=0, tat=0, at[10], bt[10], temp[10];

    float avg\_wt, avg\_tat;

    printf(" Total number of process in the system: ");

    scanf("%d", &NOP);

    y = NOP; // Assign the number of process to variable y

    // Use for loop to enter the details of the process like Arrival time and the Burst Time

    for(i=0; i<NOP; i++)

    {

       printf("\n Enter the Arrival and Burst time of the Process[%d]\n", i+1);

       printf(" Arrival time is: \t");  // Accept arrival time

       scanf("%d", &at[i]);

       printf(" \nBurst time is: \t"); // Accept the Burst time

       scanf("%d", &bt[i]);

       temp[i] = bt[i]; // store the burst time in temp array

    }

    // Accept the Time qunat

    printf("Enter the Time Quantum for the process: \t");

    scanf("%d", &quant);

    // Display the process No, burst time, Turn Around Time and the waiting time

    printf("\n Process No \t\t Burst Time \t\t TAT \t\t Waiting Time ");

    for(sum=0, i = 0; y!=0; )

    {

      if(temp[i] <= quant && temp[i] > 0) // define the conditions

      {

         sum = sum + temp[i];

         temp[i] = 0;

         count=1;

        y--; //decrement the process no.

        printf("\nProcess No[%d] \t\t %d\t\t\t\t %d\t\t\t %d", i+1, bt[i], sum-at[i], sum-at[i]-bt[i]);

        wt = wt+sum-at[i]-bt[i];

        tat = tat+sum-at[i];

        count =0;

      }

       else if(temp[i] > 0)

       {

         temp[i] = temp[i] - quant;

         sum = sum + quant;

       }

   if(at[i+1]<=sum) i++;

   else i=0;

}

// represents the average waiting time and Turn Around time

avg\_wt = wt \* 1.0/NOP;

avg\_tat = tat \* 1.0/NOP;

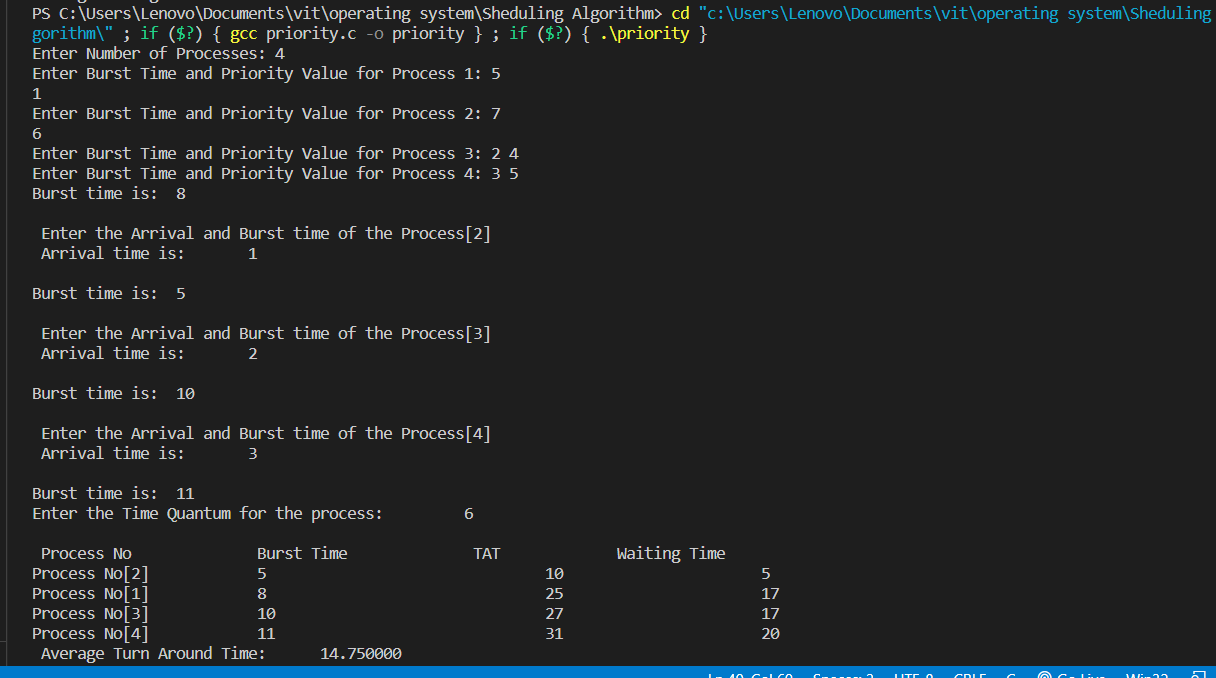
printf("\n Average Turn Around Time: \t%f", avg\_wt);

printf("\n Average Waiting Time: \t%f", avg\_tat);

getch();

}

**Output :**

****

**Shortest job first**

**Program :**

// SJF scheduling program in c

// https://www.prepbytes.com/blog/c-programming/sjf-scheduling-program-in-c/

#include<stdio.h>

int main()

{

    int n , totwt=0 , totta=0;

    double awt , ata;

    printf("Enter the number of process:");

    scanf("%d",&n);

    int pn[n], bt[n],at[n],i,j,temp,st[n],ft[n],wt[n],ta[n];

    for(i=0; i<n; i++)

    {

        printf("Enter process name, arrival time& burst time:");

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

    }

    for(i=0; i<n; i++)

        for(j=0; j<n; j++)

        {

            if(bt[i]<bt[j])

            {

                temp=at[i];

                at[i]=at[j];

                at[j]=temp;

                temp=bt[i];

                bt[i]=bt[j];

                bt[j]=temp;

                temp = pn[i];

                pn[i] = pn[j];

                pn[j] = temp;

            }

        }

    for(i=0; i<n; i++)

    {

        if(i==0) st[i]=at[i];

        else st[i]=ft[i-1];

        wt[i]=st[i]-at[i];

        ft[i]=st[i]+bt[i];

        ta[i]=ft[i]-at[i];

        totwt+=wt[i];

        totta+=ta[i];

    }

    awt=(double)totwt/n;

    ata=(double)totta/n;

    printf("\nProcess\_id \t arrivaltime \t bursttime \t waitingtime \t turnaroundtime");

    for(i=0; i<n; i++)

    {

        printf("\n%d \t\t %d \t\t %d \t\t %d \t\t %d",pn[i],at[i],bt[i],wt[i],ta[i]);

    }

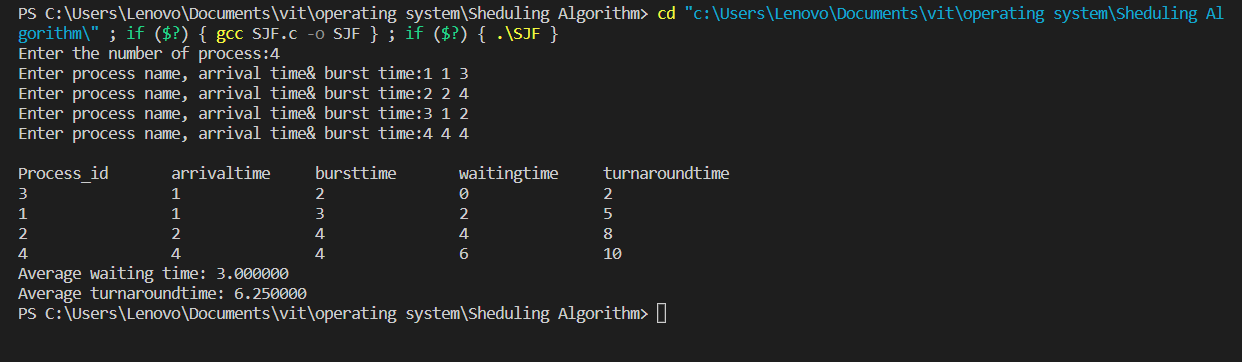
    printf("\nAverage waiting time: %f",awt);

    printf("\nAverage turnaroundtime: %f",ata);

    return 0;

}

**Output :**

****