**EXPERIMENT N0-5**

**A): First Come First Serve (FCFS)** scheduling algorithm

**CODE:**

#include<stdio.h>

#include<string.h>

int main()

{

char p[10][10],t[10];

int arr[10],burt[10],star[10],completion[10],tat[10],wt[10],i,j,n,temp;

int totwt=0,tottat=0;

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

scanf("%d",&n);

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

{

printf("Enter the ProcessName, Arrival Time& Burst Time:");

scanf("%s%d%d",&p[i],&arr[i],&burt[i]);

}

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

{

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

{

if(arr[i]<arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

temp=burt[i];

burt[i]=burt[j];

burt[j]=temp;

strcpy(t,p[i]);

strcpy(p[i],p[j]);

strcpy(p[j],t);

}

}

}

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

{

if(i==0)

star[i]=arr[i];

else

star[i]=completion[i-1];

wt[i]=star[i]-arr[i];

completion[i]=star[i]+burt[i];

tat[i]=completion[i]-arr[i];

}

printf("PName Arrtime Burtime WaitTime Start TAT completion");

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

{

printf("\n%s\t%3d\t%3d\t%3d\t%3d\t%6d\t%6d",p[i],arr[i],burt[i],wt[i],star[i],tat[i],completion[i]);

totwt+=wt[i];

tottat+=tat[i];

}

printf("\nTotal Waiting time:%f",(float)totwt);

printf("\nAverage Waiting time:%f",(float)totwt/n);

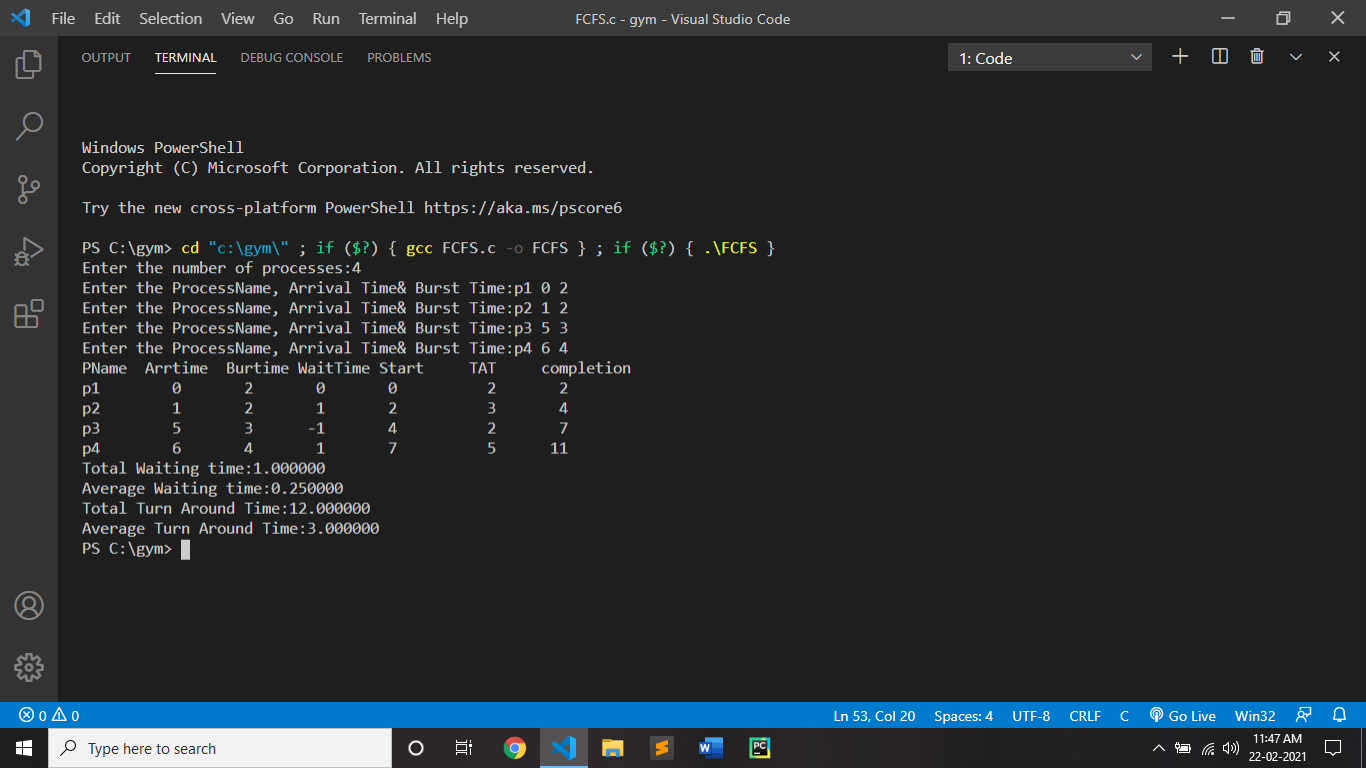
printf("\nTotal Turn Around Time:%f",(float)tottat);

printf("\nAverage Turn Around Time:%f",(float)tottat/n);

return 0;

}

**OUTPUT:**

****

**B): Round robin scheduling algorithm**

**CODE:**

#include<stdio.h>

#include<conio.h>

void main()

{

int i,a , 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", &a);

y = a;

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

{

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

printf("Arrival time is:");

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

printf("Burst time is:");

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

temp[i] = bt[i];

}

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

scanf("%d", &quant);

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

for(sum=0, i = 0; y!=0; )

{

if(temp[i] <= quant && temp[i] > 0)

{

sum = sum + temp[i];

temp[i] = 0;

count=1;

}

else if(temp[i] > 0)

{

temp[i] = temp[i] - quant;

sum = sum + quant;

}

if(temp[i]==0 && count==1)

{

y--;

printf("\nProcess No[%d] \t\t %d\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;

}

if(i==a-1)

{

i=0;

}

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

{

i++;

}

else

{

i=0;

}

}

avg\_wt = wt \* 1.0/a;

avg\_tat = tat \* 1.0/a;

printf("\n\nTotal Waiting Time: %f",avg\_wt\*a);

printf("\nAverage Waiting Time: %f", avg\_wt);

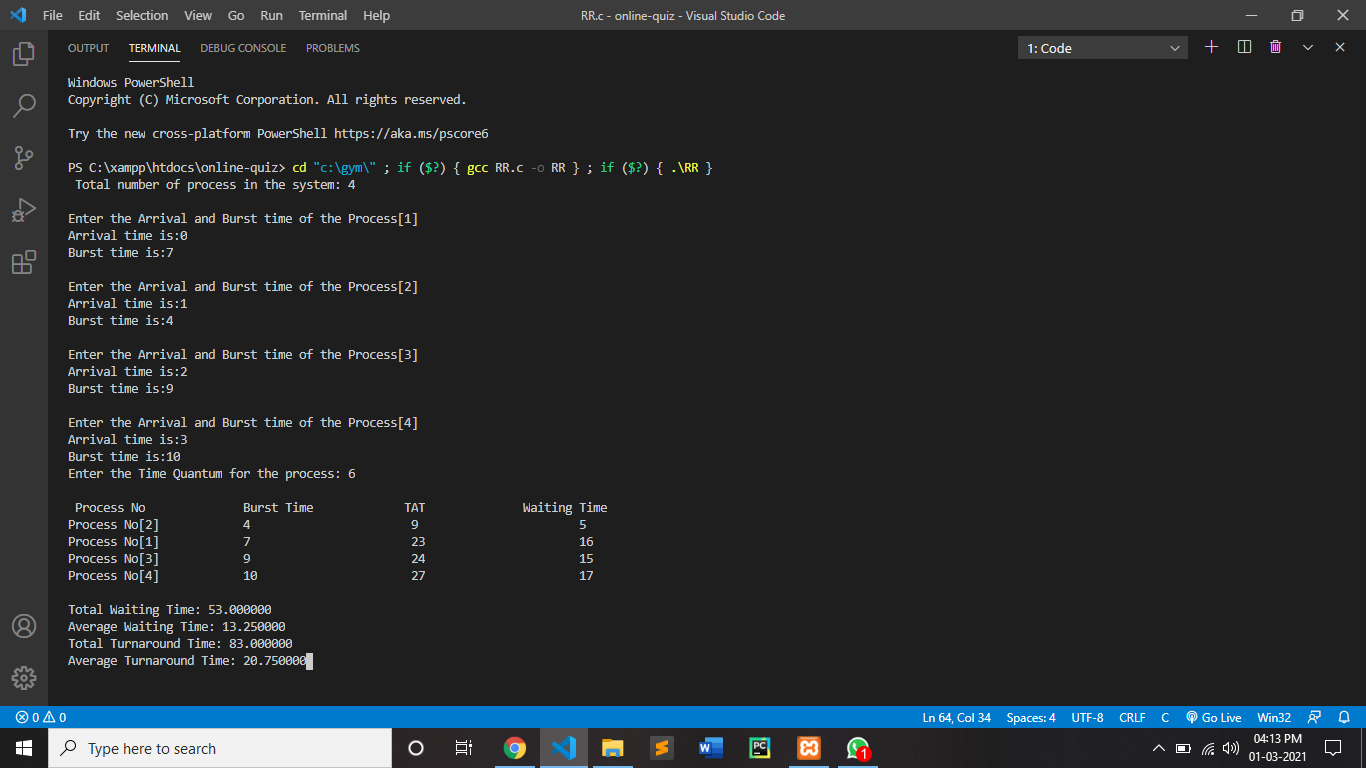
printf("\nTotal Turnaround Time: %f",avg\_tat\*a);

printf("\nAverage Turnaround Time: %f", avg\_tat);

getch();

}

**OUTPUT:**

****