OS PRACTICAL-EXAMINATION

Name: Amogh Garg

Roll Number: 2020UCO1688

Branch: Computer Science and Engineering

Date of Examination: 27.04.2022

Course: Operating System

Course Code: COCSC09

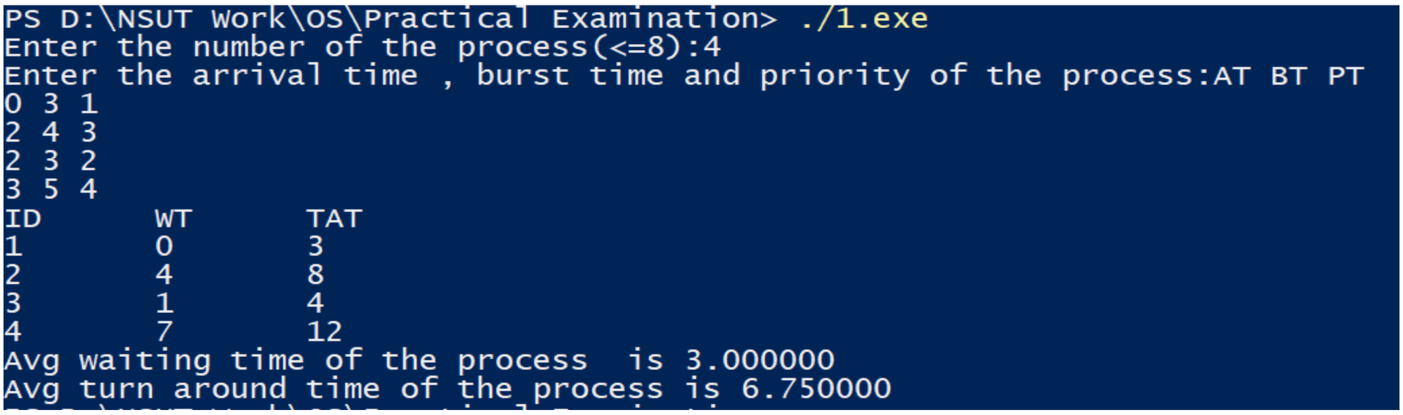
Question-1: Priority Scheduling Implementation

Code:

```
#include<stdio.h>

struct process{

   int WT,AT,BT,TAT,PT;

};

struct process a[10]; //Array of processes

//Higher number denotes lower priority

// Therefore 1 means highest priority

int main(){

   int n,temp[10],t,count=0,prior;

   float total_WT=0,total_TAT=0,Avg_WT,Avg_TAT;

   printf("Enter the number of the process(<=8):");

   scanf("%d",&n);

   printf("Enter the arrival time , burst time and priority of the process:");

   printf("AT BT PT\n");

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

      scanf("%d%d%d",&a[i].AT,&a[i].BT,&a[i].PT);

      temp[i]=a[i].BT;

   }

   a[9].PT=10000;  // 10000 here denotes very low priority

   for(t=0;count!=n;t++){ //t is time counter here

      prior=9;    // Assuming imaginary process 9 has highest priority
```

```
        //Loop for finding highest priority
        for(int i=0;i<n;i++){
            if(a[prior].PT>a[i].PT && a[i].AT<=t && a[i].BT>0){
                prior=i;
            }
        }
        a[prior].BT=a[prior].BT-1;
        //Count denotes number of completed processes
        if(a[prior].BT==0){
            count++;
            a[prior].WT=t+1-a[prior].AT-temp[prior];
            a[prior].TAT=t+1-a[prior].AT;
            total_WT=total_WT+a[prior].WT;
            total_TAT=total_TAT+a[prior].TAT;
        }
    }
    Avg_WT=total_WT/n;
    Avg_TAT=total_TAT/n;
    printf("ID WT TAT\n");
    for(int i=0;i<n;i++){
        printf("%d\t%d\t%d\n",i+1,a[i].WT,a[i].TAT);
    }
    printf("Avg waiting time of the process  is %f\n",Avg_WT);
    printf("Avg turn around time of the process is %f\n",Avg_TAT);
    return 0;
}
```

OUTPUT:



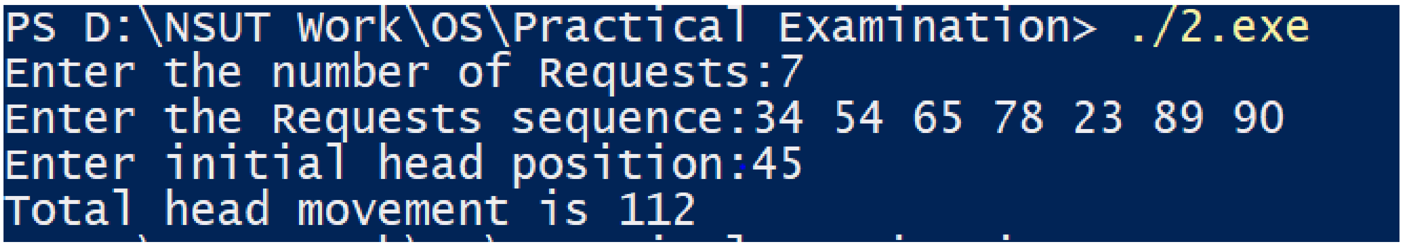
```
PS D:\NSUT Work\OS\Practical Examination> ./1.exe
Enter the number of the process(<=8):4
Enter the arrival time , burst time and priority of the process:AT BT PT
0 3 1
2 4 3
2 3 2
3 5 4
ID      WT      TAT
1       0       3
2       4       8
3       1       4
4       7       12
Avg waiting time of the process  is 3.000000
Avg turn around time of the process is 6.750000
```

Question-2: Shortest Seek Time First (SSTF) Implementation

Code:

```
#include<stdio.h>

#include<stdlib.h>

int main(){

   int RS[100],i,n,TotalHead=0,initial,count=0;

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

   scanf("%d",&n);

   printf("Enter the Requests sequence:");

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

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

   printf("Enter initial head position:");

   scanf("%d",&initial);

   //Count is the count of serviced requests

   while(count!=n){

      //1000 here denotes very large number

      int min=1000,d,index;

      //Finding shortest seek-time

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

        d=abs(RS[i]-initial);

        if(min>d){

           min=d;

           index=i;

        }

      }

      TotalHead=TotalHead+min;

      initial=RS[index];

      //So that this request doesn't get serviced again

      RS[index]=1000;

      count++;

   }

   printf("Total head movement is %d",TotalHead);

   return 0;

}
```

Output:



```
PS D:\NSUT Work\OS\Practical Examination> ./2.exe
Enter the number of Requests:7
Enter the Requests sequence:34 54 65 78 23 89 90
Enter initial head position:45
Total head movement is 112
```