VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

Assessment 2, FEBRUARY 2020

B.Tech, Winter-2019-2020

NAME	PRIYAL BHARDWAJ
REG. NO.	18BIT0272
COURSE CODE	ITE2002
COURSE NAME	OPERATING SYSTEMS
SLOT	L-37+L-38
FACULTY	Prof. SUDHA S.

(a) Implement the various process scheduling algorithms such as FCFS, SJF, Priority (Non Preemptive). **(Easy)**

FCFS:

Code:

```
#include<stdio.h>

int main()
{
    printf("18BIT0272-Priyal\n");
    int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;
    printf("Enter total number of processes(maximum 20):");
    scanf("%d",&n);

    printf("\nEnter Process Burst Time\n");
    for(i=0;i<n;i++)
    {
        printf("P[%d]:",i+1);
        scanf("%d",&bt[i]);
    }

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

    //calculating waiting time
    for(i=1;i<n;i++)
    {
        wt[i]=0;
        for(j=0;j<i;j++)
            wt[i]+=bt[j];
    }

    printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround
Time");

    //calculating turnaround time
    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];
        avwt+=wt[i];
        avtat+=tat[i];
        printf("\nP[%d]\t\t%d\t\t%d\t\t%d",i+1,bt[i],wt[i],tat[i]);
    }

    avwt/=i;
    avtat/=i;
    printf("\n\nAverage Waiting Time:%d",avwt);
```

```
    printf("\nAverage Turnaround Time:%d",avtat);
    printf("\n");

    return 0;
}
```

```
18bit0272@sjt120site053: ~
18bit0272@sjt120site053:~$ gcc fcfs2.c
18bit0272@sjt120site053:~$ ./a.out
18BIT0272-Priyal
Enter total number of processes(maximum 20):3

Enter Process Burst Time
P[1]:5
P[2]:8
P[3]:10

Process         Burst Time      Waiting Time    Turnaround Time
P[1]            5               0               5
P[2]            8               5               13
P[3]            10              13              23

Average Waiting Time:6
Average Turnaround Time:13
```

SJF (mom-preemptive):

Code:

```
#include<stdio.h>

void main()
{
    printf("18BIT0272-Priyal\n");
    int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
    float avg_wt,avg_tat;
    printf("Enter number of process:");
    scanf("%d",&n);

    printf("\nEnter Burst Time:\n");
    for(i=0;i<n;i++)
    {
        printf("p%d:",i+1);
        scanf("%d",&bt[i]);
        p[i]=i+1;           //contains process number
    }

    //sorting burst time in ascending order using selection sort
    for(i=0;i<n;i++)
```

```
    {
        pos=i;
        for(j=i+1;j<n;j++)
        {
            if(bt[j]<bt[pos])
                pos=j;
        }

        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 will be
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=(float)total/n;      //average waiting time
    total=0;

    printf("\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];

printf("\np%d\t\t  %d\t\t    %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
    }

    avg_tat=(float)total/n;     //average turnaround time
    printf("\n\nAverage Waiting Time=%f",avg_wt);
    printf("\nAverage Turnaround Time=%f\n",avg_tat);
}
```

```
18bit0272@sjt120site053: ~
18bit0272@sjt120site053:~$ gcc sjfnon2.c
18bit0272@sjt120site053:~$ ./a.out
18BIT0272-Priyal
Enter number of process:5

Enter Burst Time:
p1:4
p2:2
p3:8
p4:1
p5:9

Process     Burst Time          Waiting Time    Turnaround Time
p4                1                 0                  1
p2                2                 1                  3
p1                4                 3                  7
p3                8                 7                  15
p5                9                 15                 24

Average Waiting Time=5.200000
Average Turnaround Time=10.000000
```

Priority (non-preemptive):

Code:

```
#include<stdio.h>

int main()
{
    printf("18BIT0272-Priyal\n");
    int bt[20], p[20], wt[20], tat[20], pr[20], i, j, n, total=0,
pos,temp,avg_wt,avg_tat;
    printf("Enter Total Number of Process:");
    scanf("%d",&n);
    printf("\nEnter Burst Time and Priority\n");
    for(i=0;i<n;i++)
    {
        printf("\nP[%d]\n",i+1);
        printf("Burst Time:");
        scanf("%d",&bt[i]);
        printf("Priority:");
        scanf("%d",&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;

    printf("\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];

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

    avg_tat=total/n;     //average turnaround time
    printf("\n\nAverage Waiting Time=%d",avg_wt);
    printf("\nAverage Turnaround Time=%d\n",avg_tat);
      return 0;
}
```

```
18bit0272@sjt120site053: ~
18bit0272@sjt120site053:~$ gcc nonpri2.c
18bit0272@sjt120site053:~$ ./a.out
18BIT0272-Priyal
Enter Total Number of Process:5

Enter Burst Time and Priority

P[1]
Burst Time:4
Priority:2

P[2]
Burst Time:2
Priority:4

P[3]
Burst Time:3
Priority:3

P[4]
Burst Time:4
Priority:6

P[5]
Burst Time:6
Priority:8

Process     Burst Time          Waiting Time    Turnaround Time
P[1]              4                 0                  4
P[3]              3                 4                  7
P[2]              2                 7                  9
P[4]              4                 9                  13
P[5]              6                 13                 19
```

(b) Implement the various process scheduling algorithms such as Priority, Round Robin (preemptive). **(Medium)**

Priority (preemptive):

Code:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>

int main()
```

```
{
 int bt[20],at[10],n,i,j,temp,p[10],st[10],ft[10],wt[10],ta[10];
 int totwt=0,totta=0;
 float awt,ata;
 char pn[10][10],t[10];

 //clrscr();
 printf("Enter the number of process:");
 scanf("%d",&n);

 for(i=0; i<n; i++)
 {
 printf("Enter process name: ");
 scanf("%s",pn[i]);
 printf("Enter ArrivalTime: ");
 scanf("%d",&at[i]);
 printf("Enter BurstTime: ");
 scanf("%d",&bt[i]);
 printf("Enter Priority: ");
 //flushall();
 scanf("%d",&p[i]);
 printf("\n");
 }

 for(i=0; i<n; i++)
 for(j=0; j<n; j++)
 {
 if(p[i]<p[j])
 {
     temp=p[i];
     p[i]=p[j];
     p[j]=temp;
     temp=at[i];
     at[i]=at[j];
     at[j]=temp;
     temp=bt[i];
     bt[i]=bt[j];
     bt[j]=temp;
     strcpy(t,pn[i]);
     strcpy(pn[i],pn[j]);
     strcpy(pn[j],t);
 }
 }
 for(i=0; i<n; i++)
 {
 if(i==0)
 {
     st[i]=at[i];
     wt[i]=st[i]-at[i];
     ft[i]=st[i]+bt[i];
```

```
        ta[i]=ft[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=(float)totwt/n;
    ata=(float)totta/n;

printf("\nPname\tArrivalTime\tBurstTime\tPriority\tWaitingTime\tTota
lTurnAroundTime");
    for(i=0; i<n; i++)
        printf("\n%s\t%5d\t\t%5d\t\t%5d\t\t%5d\t\t%5d",pn[i],at[i],bt[
i],p[i],wt[i],ta[i]);

    printf("\nAverage waiting time is:%f",awt);
    printf("\nAverage turnaroundtime is:%f",ata);
    getch();
}
```

```
"C:\Users\PRIYAL BHARDWAJ\Desktop\DA's\ITE1002\DA's\Untitled1.exe"
Enter the number of process:5
Enter process name: 1
Enter ArrivalTime: 2
Enter BurstTime: 5
Enter Priority: 1

Enter process name: 3
Enter ArrivalTime: 9
Enter BurstTime: 4
Enter Priority: 3

Enter process name: 2
Enter ArrivalTime: 6
Enter BurstTime: 4
Enter Priority: 2

Enter process name: 4
Enter ArrivalTime: 6
Enter BurstTime: 1
Enter Priority: 5

Enter process name: 5
Enter ArrivalTime: 9
Enter BurstTime: 3
Enter Priority: 4


Pname   ArrivalTime     BurstTime       Priority        WaitingTime     TotalTurnAroundTime
1           2               5               1               0               5
2           6               4               2               1               5
3           9               4               3               2               6
5           9               3               4               6               9
4           6               1               5              12              13
Average waiting time is:4.200000
Average turnaroundtime is:7.600000
```

Round Robin:

Code:

```
#include<stdio.h>
int main()
{
 printf("18BIT0272-Priyal\n");
 int count,j,n,time,remain,flag=0,time_quantum;
 int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
 printf("Enter Total Process:\t ");
 scanf("%d",&n);
 remain=n;
 for(count=0;count<n;count++)
 {
 printf("Enter Arrival Time and Burst Time for Process Process
Number %d :",count+1);
 scanf("%d",&at[count]);
 scanf("%d",&bt[count]);
 rt[count]=bt[count];
 }
 printf("Enter Time Quantum:\t");
 scanf("%d",&time_quantum);
 printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");
 for(time=0,count=0;remain!=0;)
 {
 if(rt[count]<=time_quantum && rt[count]>0)
 {
 time+=rt[count];
 rt[count]=0;
 flag=1;
 }
 else if(rt[count]>0)
 {
 rt[count]-=time_quantum;
 time+=time_quantum;
 }
 if(rt[count]==0 && flag==1)
 {
 remain--;
 printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-
at[count]-bt[count]);
 wait_time+=time-at[count]-bt[count];
 turnaround_time+=time-at[count];
 flag=0;
```

```
 }
 if(count==n-1)
 count=0;
 else if(at[count+1]<=time)
 count++;
 else
 count=0;
 }
 printf("\nAverage Waiting Time= %f\n",wait_time*1.0/n);
 printf("Avg Turnaround Time = %f",turnaround_time*1.0/n);

 return 0;
}
```

```
"C:\Users\PRIYAL BHARDWAJ\Desktop\DA's\ITE1002\DA's\Untitled1.exe"
Enter Total Process:     6
Enter Arrival Time and Burst Time for Process Process Number 1 :0
9
Enter Arrival Time and Burst Time for Process Process Number 2 :3
5
Enter Arrival Time and Burst Time for Process Process Number 3 :6
5
Enter Arrival Time and Burst Time for Process Process Number 4 :9
1
Enter Arrival Time and Burst Time for Process Process Number 5 :6
8
Enter Arrival Time and Burst Time for Process Process Number 6 :4
2
Enter Time Quantum:     3


Process |Turnaround Time|Waiting Time

P[4]    |       1       |       0
P[6]    |       11      |       9
P[2]    |       17      |       12
P[3]    |       16      |       11
P[1]    |       28      |       19
P[5]    |       24      |       16

Average Waiting Time= 11.166667
Avg Turnaround Time = 16.166667
```

c) Consider a corporate hospital where we have n number of patients waiting for consultation. The amount of time required to serve a patient may vary, say 10 to 30 minutes. If a patient arrives with an emergency,he /she should be attended immediately before other patients, which may increase the waiting time of other patients. If you are given this problem with the following algorithms how would you devise an effective scheduling so that it optimizes the overall performance such as minimizing the waiting time of all patients. [Single queue or multi-level queue can be used].

• Consider the availability of single and multiple doctors • Assign top priority for patients with emergency case, women, children, elders, and youngsters. • Patients coming for review may take less time than others. This can be taken into account while using SJF.

1. FCFS
2. SJF (primitive and non-pre-emptive) **(High)**

FCFS & SJF(non-preemptive):

Code:

```
#include<iostream>
#include<conio.h>
#include<stdio.h>
using namespace std;
class cpuschedule
{int n,bu[20],pri[20];
float twt,awt,wt[20],tat[20];
public: void Getdata();
void fcfs();
void sjf();
};
//Getting no of processes and Burst time void
void cpuschedule::Getdata()
{int i; string s; cout<<"Enter the no of Patients:"; cin>>n;
for(i=1;i<=n;i++)
{cout<<"\nEnter The Time for Patient "<<i<<"=";
cin>>bu[i]; cout<<"\nEnter the type of Patient\n"; cin>>s;
if(s=="Emergency") pri[i]=1
; else if(s=="Woman") pri[i]=2;
else if(s=="Child") pri[i]=3;
else if(s=="Old") pri[i]=4; else if(s=="Young") pri[i]=5;
}}
```

```
//First come First served Algorithm
void cpuschedule::fcfs()
{ int i,b[10]; float sum=0.0; twt=0.0; for(i=1;i<=n;i++)
{b[i]=bu[i]; cout<<"\nTime for patient "<<i<<"=";
cout<<b[i];
}
wt[1]=0; for(i=2;i<=n;i++)
{
wt[i]=b[i-1]+wt[i-1];
}
for(i=1;i<=n;i++)
{
twt=twt+wt[i];
tat[i]=b[i]+wt[i]; sum+=tat[i];
}
awt=twt/n; sum=sum/n; cout<<"\nTotal Waiting Time="<<twt;
cout<<"\nAverage Waiting Time="<<awt;
cout<<"\nAverage Turnaround time="<<sum;
}
//Shortest job First Algorithm
void cpuschedule::sjf()
{
int i,j,temp,b[10]; float sum=0.0; twt=0.0;
for(i=1;i<=n;i++)
{
b[i]=bu[i]; cout<<"\nTime for patient "<<i<<"=";
cout<<b[i];
}
for(i=n;i>=1;i--)
{
for(j=2;j<=n;j++)
{ if(b[j-1]>b[j])
{
temp=b[j-1]; b[j-1]=b[j]; b[j]=temp;
}}}
wt[1]=0; for(i=2;i<=n;i++)
{
wt[i]=b[i-1]+wt[i-1];
}
for(i=1;i<=n;i++)
{
twt=twt+wt[i];
tat[i]=b[i]+wt[i]; sum+=tat[i];
}
awt=twt/n; sum=sum/n; cout<<"\nTotal Waiting Time="<<twt;
cout<<"\nAverage Waiting Time="<<awt;
cout<<"\nAverage turnaround time="<<sum;
}
int main()
{
```

```
cout<<"18BIT0272-Priyal\n";
int ch=0,cho; cpuschedule c; do
{
switch(ch)
{
case 0:
cout<<"\n0.Menu";
cout<<"\n1.Get Patient Time";
cout<<"\n2.FCFS";
cout<<"\n3.SJF";
cout<<"\n4.Exit";
break; case 1:
c.Getdata(); break; case 2:
cout<<"FCFS";
c.fcfs(); break; case 3:
cout<<"SJF";
c.sjf(); break; case 4:
break;
}
cout<<"\nEnter your choice:";
cin>>ch; }while(ch<4);
return 0;
}
```

```
"C:\Users\PRIYAL BHARDWAJ\Desktop\DA's\ITE1002\DA's\Untitled1.exe"
18BIT0272-Priyal

0.Menu
1.Get Patient Time
2.FCFS
3.SJF
4.Exit
Enter your choice:1
Enter the no of Patients:4

Enter The Time for Patient 1=12

Enter the type of Patient
Old

Enter The Time for Patient 2=3

Enter the type of Patient
Child

Enter The Time for Patient 3=8

Enter the type of Patient
Young

Enter The Time for Patient 4=15

Enter the type of Patient
Male

Enter your choice:2
FCFS
Time for patient 1=12
Time for patient 2=3
Time for patient 3=8
Time for patient 4=15
Total Waiting Time=50
Average Waiting Time=12.5
Average Turnaround time=22
Enter your choice:3
SJF
Time for patient 1=12
Time for patient 2=3
Time for patient 3=8
Time for patient 4=15
Total Waiting Time=37
Average Waiting Time=9.25
Average turnaround time=18.75
Enter your choice:4
```

SJF (preemptive):

Code:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
printf("18BIT0272-Priyal\n");
int arrival_time[10], service_time[10], temp[10];
int i, smallest, count = 0, time, limit;
double wait_time = 0, turnaround_time = 0, end;
float average_waiting_time, average_turnaround_time;
printf("\nEnter the Number of Patients:");
scanf("%d", &limit);
printf("\nEnter Details of %d Patients\n", limit);
for(i = 0; i < limit; i++)
{
 printf("\nEnter Patient Arrival Time:");
 scanf("%d", &arrival_time[i]);
 printf("Enter Patient Service Time:");
 scanf("%d", &service_time[i]);
 temp[i] = service_time[i];
}
service_time[9] = 9999;
for(time = 0; count != limit; time++)
{
 smallest = 9;
 for(i = 0; i < limit; i++)
 {
 if(arrival_time[i] <= time && service_time[i] <
service_time[smallest]
&& service_time[i] > 0)
 smallest = i;
 }
 service_time[smallest]--;
 if(service_time[smallest] == 0)
 {
 count++;
 end = time + 1;
 wait_time = wait_time + end - arrival_time[smallest] -
temp[smallest];
 turnaround_time = turnaround_time + end - arrival_time[smallest];
 }
}
average_waiting_time = wait_time / limit;
average_turnaround_time = turnaround_time / limit;
printf("\n\nAverage Waiting Time:%lf\n", average_waiting_time);
printf("Average Turnaround Time:%lf\n", average_turnaround_time);
return 0;
```

}

```
"C:\Users\PRIYAL BHARDWAJ\Desktop\DA's\ITE1002\DA's\Untitled1.exe"
18BIT0272-Priyal

Enter the Number of Patients:5

Enter Details of 5 Patients

Enter Patient Arrival Time:4
Enter Patient Service Time:10

Enter Patient Arrival Time:9
Enter Patient Service Time:6

Enter Patient Arrival Time:6
Enter Patient Service Time:8

Enter Patient Arrival Time:5
Enter Patient Service Time:12

Enter Patient Arrival Time:3
Enter Patient Service Time:5


Average Waiting Time:10.600000
Average Turnaround Time:18.799999
```
