

# OPERATING SYSTEMS

Course Code : SWE3001

Slot : F2+TF2

## Assignment - 4

Name : S.Deepan

Reg No : 19MIS0102

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.

Develop an effective scheduling algorithm, so that it optimizes the overall performance such as minimizing the waiting time of all patients.

- Consider the availability of single and multiple doctors.
- Assign top priority for patients with emergency case, women, children, elders, and youngsters.

### **Code :**

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

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

```
void main()
```

```
{
```

```
    // initialize 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("Enter the Total number of patient : ");
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("\nEnter the Arrival time and Burst time of the patient %d \n", i+1);
printf("Enter the  Arrival time : \t"); // Accept arrival time
scanf("%d", &at[i]);
printf(" \nEnter the Burst time : \t\t"); // Accept the Burst time
scanf("%d", &bt[i]);
temp[i] = bt[i]; // store the burst time in temp array
}
// Accept the Time quantat
printf("Enter the Time Quantum for the Patient: \t");
scanf("%d", &quant);
// Display the process No, burst time, Turn Around Time and the waiting time
printf("\nPatient 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;
}
else if(temp[i] > 0)
{
    temp[i] = temp[i] - quant;
    sum = sum + quant;
}
if(temp[i]==0 && count==1)
{
    y--; //decrement the process no.
    printf("\nPatient 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==NOP-1)
{
    i=0;
}
else 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("\nAverage Turn Around Time: \t%f", avg_wt);
printf("\nAverage Waiting Time: \t%f", avg_tat);
printf("\n\n");
}

```

### Output :

```

deepan2001@ubuntu:~/Desktop$ gcc Ass4.c
deepan2001@ubuntu:~/Desktop$ ./a.out
Enter the Total number of patient : 5

Enter the Arrival time and Burst time of the patient 1
Enter the  Arrival time :      0

Enter the Burst time :      8

Enter the Arrival time and Burst time of the patient 2
Enter the  Arrival time :      1

Enter the Burst time :      4

Enter the Arrival time and Burst time of the patient 3
Enter the  Arrival time :      2

Enter the Burst time :      5

Enter the Arrival time and Burst time of the patient 4
Enter the  Arrival time :      3

Enter the Burst time :      2

Enter the Arrival time and Burst time of the patient 5
Enter the  Arrival time :      4

Enter the Burst time :      1
Enter the Time Quantum for the Patient:      2

Patient No      Burst Time      TAT      Waiting Time
Patient No 4      2      5      3
Patient No 5      1      5      4
Patient No 2      4      12      8
Patient No 3      5      16      11
Patient No 1      8      20      12
Average Turn Around Time:      7.600000
Average Waiting Time:      11.600000

deepan2001@ubuntu:~/Desktop$ █

```