COMSATS UNIVERSITY ISLAMABAD ATTOCK CAMPUS

Lab Report 9 : Operating System

Submitted to : Sir Fayyaz Ali

Group Members	Muaaz Shoaib FA20-BCS-074 Shahzeb Shaheen FA20-BCS-040

Rubrics Assessment Sheet for Operating System		
Lab #:	**Lab no 9**	
Lab Title:	**Multi Level Queue Scheduling Algorithm**	
Submitted by:		

Names	Registration
Muaaz Shoaib	**FA20-BCS-074**
Shahzeb Shaheen	**FA20-BCS-040**

Rubrics name & number		Marks	
		ln-Lab	Post lab
Engineering Knowledge	*R2*:Use of Engineering Knowledge and follow Experiment Procedures: *Ability to follow experimental procedure,control variables,and record Procedural steps on lab report.*		
Problem Analysis	*R6: Experimental Data Analysis : Ability to interept findings,compare them to values in the literature,identify weaknesses and limitations*		
Design	*RS: Best Coding Staudards: Ability to follow the coding standards and programming practices*		
Modem Tools Usage	*R9: Undcrsta11d Tools: Ability to describe and explain the principles behind applicability of engineering tools.*		
Individual and Teamwork	*R9:Management of Team Work: Ability to appreciate, understand and work multidisciplinary team members*		

Rubrics #	R2	R6	RS	R9	Rl3
Jn -Lab					
Post- Lab					

Description :

In Multi Level Queue Scheduling Algorithm the processes in the ready queue can be divided into different classes where each class has its own scheduling needs.

IN-LAB QUESTIONS

Q : Write a C program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into three categories – system processes, user processes and I/O processes. System processes are to be given higher priority than user processes and I/O. Consider each process priority to be from 1 to 3. Use priority scheduling for the processes in each queue

Answer :

Code :

```c
Untitled2.c
1   main()
2   {
3   int p[20],bt[20],su[20],wt[20],tat[20],i,k,n,temp;
4   float wtavg, tatavg;
5
6   printf("Enterthenumberofprocesses --- ");
7   scanf("%d",&n);
8   for(i=0;i<n;i++)
9   {
10      p[i] = i;
11      printf("EntertheBurstTimeofProcess%d--- ",i);
12      scanf("%d",&bt[i]);
13      printf("System/User Process (0/1) ? --- ");
14      scanf("%d", &su[i]);
15   }
16   for(i=0;i<n;i++)
17   for(k=i+1;k<n;k++)
18   if(su[i] > su[k])
19   {
20       temp=p[i];
21   p[i]=p[k];
22   p[k]=temp;
23   temp=bt[i];
24   bt[i]=bt[k];
25   bt[k]=temp;
26   temp=su[i];
```

```c
Untitled2.c
22   p[k]=temp;
23   temp=bt[i];
24   bt[i]=bt[k];
25   bt[k]=temp;
26   temp=su[i];
27   su[i]=su[k];
28   su[k]=temp;
29   }
30   wtavg = wt[0] = 0;
31   tatavg = tat[0] = bt[0];
32   for(i=1;i<n;i++)
33   {
34       wt[i] = wt[i-1] + bt[i-1];
35   tat[i] = tat[i-1] + bt[i];
36   wtavg = wtavg + wt[i];
37   tatavg=tatavg+tat[i];
38   }
39   printf("\nPROCESS\t\t SYSTEM/USER PROCESS \tBURST TIME\tWAITINGTIME\tTURNAROUND TIME");
40
41   for(i=0;i<n;i++)
42   printf("\n%d \t\t %d \t\t %d \t\t %d \t\t %d ",p[i],su[i],bt[i],wt[i],tat[i]);
43   printf("\nAverage Waiting Time is --- %f",wtavg/n);
44   printf("\nAverage Turnaround Time is --- %f",tatavg/n);
45   getch();
46   }
47
```

Output :


```
C:\Users\HP\OneDrive\Desktop\Project1.exe                                                    —    □    >

Enterthenumberofprocesses --- 4
EntertheBurstTimeofProcess0--- 4
System/User Process (0/1) ? --- f
EntertheBurstTimeofProcess1--- System/User Process (0/1) ? --- EntertheBurstTimeofProcess2--- System/User Process (0/1)
? --- EntertheBurstTimeofProcess3--- System/User Process (0/1) ? ---
PROCESS          SYSTEM/USER PROCESS     BURST TIME     WAITINGTIME      TURNAROUND TIME
1                0              0                0              0
3                0              0                0              0
0                1573712                4              0        4
2                6487332                268501009          4              268501013
Average Waiting Time is --- 1.000000
Average Turnaround Time is --- 67125256.000000
```