

OPERATING SYSTEM LABORATORY MANUAL



UNIVERSITY OF THE PUNJAB

FACULTY OF COMPUTING & INFORMATION TECHNOLOGY, LAHORE

DEPARTMENT OF COMPUTER SCIENCE

Course:	Operating System Lab	Date:
Course Code:	CC-217-3L	Max Marks: 40
Faculty/Instructor's Name & Email:	Dr. Ahmad Hassan Butt (ahmad.hassan@pucit.edu.pk)	

LAB MANUAL # 8 (SPRING 2023)

Name: _____ Enroll No: _____

Objective(s) :

To write a C program to implement CPU scheduling algorithm for Priority Scheduling.

Lab Tasks :

Task 01: Calculate the Average Time using Priority Scheduling.

Task 02: Write the algorithm for Priority Scheduling Algorithm.

Task 03 + 04: Write the output for program of Priority Scheduling.

Lab Grading Sheet :

Task	Max Marks	Obtained Marks	Comments(<i>if any</i>)
1.	10		
2.	10		
3.	10		
4.	10		
Total	40		Signature

Note : Attempt all tasks and get them checked by your Instructor

Lab 08: Priority Scheduling

Objective(s):

To write a C program to implement CPU scheduling algorithm for Priority Scheduling.

Tool(s) used:

Ubuntu, VIM Editor

CPU scheduler will decide which process should be given the CPU for its execution. For this it use different algorithm to choose among the process. One among that algorithm is FCFS algorithm. In this algorithm the process which arrives first is given the CPU after finishing its request only it will allow CPU to execute other process. In priority scheduling algorithm each process has a priority associated with it and as each process hits the queue, it is stored in based on its priority so that process with higher priority are dealt with first. It should be noted that equal priority processes are scheduled in FCFS order.

Task 01: Calculate the Average Time using Priority Scheduling.

Process	CPU Burst Time	Priority
P1	9	5
P2	4	3
P3	5	1
P4	7	2
P5	3	4
Total	28	

Task 02: Write the algorithm for Priority Scheduling Algorithm.

Task 03+04: Write the output for program of Priority Scheduling.

```
#include <stdio.h>
int main()
{
    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
    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
printf("\nProcess\t    Burst Time    \tWaiting Time ");
for(i=0;i<n;i++){
    printf("\nP[%d]\t\t %d\t\t %d",p[i],bt[i],wt[i]);
}
printf("\n\nAverage Waiting Time=%d",avg_wt);

return 0;
}
```

OUTPUT