



UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

**FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING**

**SOFTWARE ENGINEERING DEPARTMENT**

# **OPERATING SYSTEMS LAB**

Lab 13

Shortest Job First CPU scheduling algorithm



**UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA**  
**FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING**

**SOFTWARE ENGINEERING DEPARTMENT**

## Discussion:

- Types of queues
- Process states
- Shortest Job First (SJF) scheduling algorithm

**Task:** Write a program for Shortest Job First (SJF) scheduling algorithm. And compare its outputs with previous implemented algorithms.

## Solution:

Source Code:

```
void main()
{
    // declare variable and initilize them
    // total number of process
    printf("Enter Number of process : ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter process Name | Arrival Time | Brust Time :");
        scanf("%s %d %d",&process[i],&Arrival_Time[i],&Burst_Time[i]);
    }
    /* Sort in Accending order Process and Arrival according to the
    BrustTime(execution TIME)*/
    for(i=0;i<n;i++)
    {
        if(i==0)
            Start_Time[i]=Arrival_Time[i];
        else
            //Start Time = Finish Time;

            //Wait Time = Start Time - Arrival Time;
            //Finish Time = ????;
            //Turn around Time = Finish Time - Arrival Time;
    }
    printf("\n Process ArrivalTime BrustTime StartTime FinishTime
    TurnArroundTime WaitTime");
    for(i=0;i<n;i++)
    {
        printf("\n  %s %10d %10d %10d %10d %13d
        %10d",process[i],Arrival_Time[i],Burst_Time[i],Start_Time[i],Finish_Time[i],TA
        _Time[i],Wait_Time[i]);
    }
    // Average Wait Time
    // Average Turn Around Time
}
```



# UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

## **FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING**

### **SOFTWARE ENGINEERING DEPARTMENT**

Outputs And Comparisons:

<b>Job Arrival Sequence</b>	<b>Length</b>	<b>Wait Time FCFS</b>	<b>SJF</b>
1	22	0	73
2	12	22	14
3	12	34	26
4	16	46	38
5	27	62	95
6	5	89	0
7	9	94	5
8	19	103	54
Average Waiting Time		56.25	38.125



**UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA**  
**FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING**

**SOFTWARE ENGINEERING DEPARTMENT**

```
[nauman@localhost Desktop]$ gcc SJFIRST.c
[nauman@localhost Desktop]$ ./a.out
Enter Number of process : 8
Enter process Name | Arrival Time | Brust Time :p1 0 22
Enter process Name | Arrival Time | Brust Time :p2 0 12
Enter process Name | Arrival Time | Brust Time :p3 0 12
Enter process Name | Arrival Time | Brust Time :p4 0 16
Enter process Name | Arrival Time | Brust Time :p5 0 27
Enter process Name | Arrival Time | Brust Time :p6 0 5
Enter process Name | Arrival Time | Brust Time :p7 0 9
Enter process Name | Arrival Time | Brust Time :p8 0 19

Process ArrivalTime BrustTime StartTime FinishTime TurnArroundTime WaitTime
p6      0      5      0      5      5      0
p7      0      9      5      14     14     5
p2      0     12     14     26     26     14
p3      0     12     26     38     38     26
p4      0     16     38     54     54     38
p8      0     19     54     73     73     54
p1      0     22     73     95     95     73
p5      0     27     95    122    122    95
Average Wait Time = 38.125000
Average Turn Arround Time = 53.375000
[nauman@localhost Desktop]$
```



**UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA**  
**FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING**

**SOFTWARE ENGINEERING DEPARTMENT**

The screenshot shows a Linux desktop environment with a window titled "SJF.c - Kate". The window contains a C program for implementing the Shortest Job First (SJF) scheduling algorithm. The program prompts the user to enter the number of processes, followed by the process name, arrival time, and execution time for each process. The output shows the execution order and the average waiting and turnaround times.

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

for(i=0;i<n;i++) |
{
    printf("Enter process name, arrival time & execution time:");
    scanf("%s%d%d",pn[i],&at[i],&et[i]);
}
for(i=0;i<n;i++)
```

Line: 13 Col: 20 INS LINE SJF.c

```
Enter the number of process:3
Enter process name, arrival time & execution time:p1 0 24
Enter process name, arrival time & execution time:p2 0 3
Enter process name, arrival time & execution time:p3 0 3

Pname  arrivaltime  executiontime  waitingtime  tatetime
p2      0              3              0              3
p3      0              3              3              6
p1      0              24             6              30
Average waiting time is:3.000000
Average turnaroundtime is:13.000000[nauman@localhost Desktop]$
```