UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

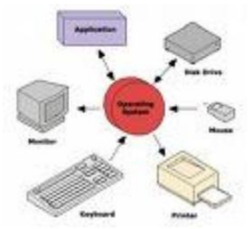
FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

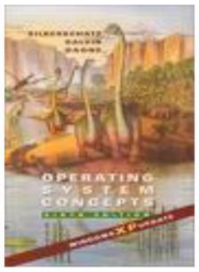
COMPUTER ENGINEERING DEPARTMENT

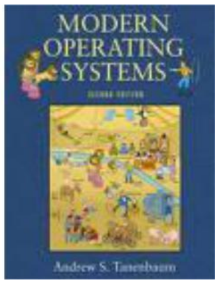
Operating systems

IMPLEMENTATION OF SJF ALGORITHM

LAB MANUAL 10







Date:	
Name:	
Reg#:	**Group:**
Marks:	**Signature:**

Operating systems

Lab Objective

Write a C program to implement the various process scheduling mechanisms such as SJF Scheduling.

Algorithm for SJF

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Start the Ready Q according the shortest Burst time by sorting according to lowest to highest burst time.

Step 5: Set the waiting time of the first process as '0' and its turnaround time as its burst time.

Step 6: For each process in the ready queue, calculate

(a) Waiting time for process(n)= waiting time of process (n-1) + Burst time of process(n-1)

(b) Turn around time for Process(n)= waiting time of Process(n)+ Burst time for process(n)

(a) Average waiting time = Total waiting Time / Number of process

(b) Average Turnaround time = Total Turnaround Time / Number of

process Step 7: Stop the process

/* SJF SCHEDULING ALGORITHM */

```
#include<stdio.h>
void main()
{
 int i,j,k,n,sum,wt[10],tt[10],twt,ttat;
 int t[10],p[10];
 float awt,atat;
 clrscr();

 printf("Enter number of process\n");
 scanf("%d",&n);

 for(i=0;i<n;i++)
```

```
{
   printf("\n Enter the Burst Time of Process %d",i);
   scanf("\n %d",&t[i]);
}

for(i=0;i<n;i++)
  p[i]=i;
  for(i=0;i<n;i++)
  {
    for(k=i+1;k<n;k++)
    {
      if(t[i]>t[k])
      {
              int temp;
              temp=t[i];
              t[i]=t[k];
              t[k]=temp;

              temp=p[i];
              p[i]=p[k];
              p[k]=temp;
      }
    }
    printf("\n\n SHORTEST JOB FIRST SCHEDULING ALGORITHM");
    printf("\n PROCESS ID \t BURST TIME \t WAITING TIME \t TURNAROUND TIME \n\n");
    wt[0]=0;
    for(i=0;i<n;i++)
    {
      sum=0;
      for(k=0;k<i;k++)
      {
            wt[i]=sum+t[k];
            sum=wt[i];
      }
    }
    for(i=0;i<n;i++)
    {
      tt[i]=t[i]+wt[i];
    }
    for(i=0;i<n;i++)
    {
      printf("%5d \t\t5%d \t\t %5d \t\t %5d \n\n",p[i],t[i],wt[i],tt[i]);
    }
    twt=0;
    ttat=t[0];
    for(i=1;i<n;i++)
    {
            twt=twt+wt[i];
```

```
        ttat=ttat+tt[i];
    }
    awt=(float)twt/n;
    atat=(float)ttat/n;

    printf("\n AVERAGE WAITING TIME %4.2f",awt);
    printf("\n AVERAGE TURN AROUND TIME
    %4.2f",atat); getch();
  }
}
```

OUTPUT:

```
Enter number of process
3

 Enter the Burst Time of Process 04

 Enter the Burst Time of Process 13

 Enter the Burst Time of Process 25


 SHORTEST JOB FIRST SCHEDULING ALGORITHM
   PROCESS ID    BURST TIME      WAITING TIME   TURNAROUND TIME

     1          3               0               3

     0          4               3               7

     2          5               7              12


 AVERAGE WAITING TIME 3.33
 AVERAGE TURN AROUND TIME 7.33
```