

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

Protected by PDF Anti-Copy Free

Lecture-16 By Vivek Dubey

Theory and Lab

PDF

COMPUTER SCIENCE AND ENGINEERING

CS403PC

COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

CS403PC

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Scheme of Operating System

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



S. No.	Course Code	Course Title	L	T	P	Credits
3	CS403PC	Operating Systems	3	0	0	3
6	CS406PC	Operating Systems Lab	0	0	3	1.5

UNIT-2:

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- **Process and CPU Scheduling –**
 - Process concepts and scheduling, Operations on Processes, Cooperating Processes, Threads, and Interposes Communication, Scheduling Criteria, **Scheduling Algorithms**, Multiple-Processor Scheduling.
- **System call interface for process management-**
 - fork, exit, wait, waitpid, exec

Scheduling Algorithms

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Different Scheduling Algorithms

- *First Come First Serve (FCFS)*
- *Shortest Job First(SJF)*
- *Longest Job First(LJF)*
- *Shortest Remaining Time First(SRTF)*
- *Longest Remaining Time First(LRTF)*
- *Round Robin Scheduling*
- *Priority Based scheduling (Non-Preemptive)*
- *Highest Response Ratio Next (HRRN)*
- *Multilevel Queue Scheduling*
- *Multi level Feedback Queue Scheduling*

Shortest Job First(SJF) Scheduling Algorithm

Characteristics of Shortest Job First Scheduling

1. SJF algorithm is helpful in batch operating *where the waiting time for job completion is not critical.*
2. SJF improves the throughput of the process by ensuring that the shorter jobs are executed first, *thus the possibility of less turnaround time.*
3. SJF enhances the output of the job by executing the process, *which is having the shortest burst time.*

Types of Shortest Job First (SJF) Scheduling



There are two types of Shortest Job First Scheduling.

1. Non-Preemptive SJF
2. Preemptive SJF

Non-Preemptive SJF

- In Non-Preemptive Scheduling, if a CPU is located to the process, *then the process will hold the CPU until the process enters into the waiting state or terminated.*

Preemptive SJF Scheduling

- In Preemptive Scheduling, jobs are moved into the ready queue *when they arrive.*
- Those Processes which have less burst time begins its execution first.
- When the process with less burst time arrives, then the current process stops execution, and the process having less burst time is allocated with the CPU first.

Advantages of SJF

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



1. It is the optimal algorithm ; it gives minimum average waiting time for a given set of processes.
2. By moving a short process before a long one, the waiting time of short process decreases.
3. It increases waiting time of the long process, so avg. waiting time decreases.

Disadvantages of SJF

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



1. The real difficulty with the SJF algorithm is to know the length of next CPU request.
2. SJF scheduling is used frequently in long term scheduling.
3. There is no way to know the length of next CPU burst. But we may predict the value of next CPU burst.
4. We accept that next CPU burst will be similar in length to the previous ones.
5. Thus, by computing an approximation at the length of next CPU burst, we can select the process with the shortest predicted CPU burst.
6. Next CPU burst is generally predicted as an exponential average of the measured length with respect to the previous CPU burst.

Case-1 Non-Preemptive SJF

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Lecture-16 By Vivek Dubey

Gantt Chart

Step-2

	P4	P1	P3	P2
0	3	9	16	24

Process ID	Burst Time	Waiting Time	Turnaround Time
P4	3	0	3
P1	6	3	9
P3	7	9	16
P2	8	16	24

Sort Based on BT

Step-1

Process ID	Burst Time
P1	6
P2	8
P3	7
P4	3

Step-3

Step-4

Average waiting time
 $= (3+16+9+0)/4$
 $= 28/4$
 $= 7$

Average turnaround time
 $= (9+24+16+3)/4$
 $= 52/4$
 $= 13$

Step-5

C Program for Non- preemptive SJF

```
#include<stdio.h>
int main()
{
    int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
    float avg_wt,avg_tat;
    printf("Enter number of process:");
    scanf("%d",&n);

    printf("\nEnter Burst Time:\n");
    for(i=0;i<n;i++)
    {
        printf("p%d:",i+1);
        scanf("%d",&bt[i]);
        p[i]=i+1;
    }
```

```
//sorting of burst times
for(i=0;i<n;i++)
{
    pos=i;
    for(j=i+1;j<n;j++)
    {
        if(bt[j]<bt[pos])
            pos=j;
    }

    temp=bt[i];
    bt[i]=bt[pos];
    bt[pos]=temp;

    temp=p[i];
    p[i]=p[pos];
    p[pos]=temp;
}
```

```
wt[0]=0;
```

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

```
for(i=1;i<n;i++)
{
    wt[i]=0;
    for(j=0;j<i;j++)
        wt[i]+=bt[j];

    total+=wt[i];
}
```



```
avg_wt=(float)total/n;
total=0;

printf("\nProcess\t Burst Time\t \tWaiting Time\tTurnaround Time");
for(i=0;i<n;i++)
{
    tat[i] = bt[i] + wt[i];
    total = total + tat[i];
    printf("\np%d\t\t %d\t\t %d\t\t\t %d",p[i],bt[i],wt[i],tat[i]);
}

avg_tat=(float)total/n;
printf("\n\nAverage Waiting Time=%f",avg_wt);
printf("\n\nAverage Turnaround Time=%f\n",avg_tat);
}
```

Preemptive SJF

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Process ID	Burst Time	Arrival Time
P1	9	0
P2	4	1
P3	5	2
P4	7	3
P5	3	4

Ms.	PDF	P2	P3	P4	P5
0.ms					
1.ms	8	4			
2.ms	8	3	5		
3.ms	8	2	5	7	
4.ms	8	1	5	7	3
5.ms	8	0	5	7	3
8.ms	8	×	5	7	0
13.ms	8	×	0	7	×
20.ms	8	×	×	0	×
28.ms	0	×	×	×	×

Waiting Time = Total waiting Time – No.of Milisec. Process executed – Arrival Time

Process waiting time:

P1 = 20 – 1 – 0 = 19 ms,
P2 = 4 – 3 – 1 = 0 ms,
P3 = 8 - 0 – 2 = 6ms
P4 = 13 – 0 – 3 = 10 ms,
P5 = 5 - 0 - 4 = 1 ms.

Average waiting time
= (19+0+6+10+1)/5
=36/5
=7.2 ms

Turnaround Time = Complition Time- Arrival Time

Process Turnaround Time:

P1 = 28 – 0 =28 ms,
P2 = 5 – 1 = 4,
P3 = 13 – 2 = 11,
P4 = 20 – 3 = 17,
P5 = 8 – 4 = 4

Average turnaround time
= (31+4+12+4)/4
=12.75

C Program - Preemptive SJF

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



```
#include <stdio.h>
int main()
{
    int arrival_time[10], burst_time[10], temp[10];
    int i, smallest, count = 0, time, limit;
    double wait_time = 0, turnaround_time = 0, end;
    float average_waiting_time, average_turnaround_time;
    printf("\nEnter the Total Number of Processes:\t");
    scanf("%d", &limit);
```

```
printf("\nEnter Details of %d Processes\n", limit);  
for(i = 0; i < limit; i++)  
{  
    printf("\nEnter Arrival Time:\t");  
    scanf("%d", &arrival_time[i]);  
    printf("Enter Burst Time:\t");  
    scanf("%d", &burst_time[i]);  
    temp[i] = burst_time[i];  
}
```

```
burst_time[9] = 9999;
```

```
for(time = 0; count != limit; time++)
```

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

```
{
```

```
    smallest = 9;
```

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

```
    {
```

```
        if(arrival_time[i] <= time && burst_time[i] < burst_time[smallest] && burst_time[i] > 0)
```

```
            smallest = i;
```

```
    }
```

```
    burst_time[smallest]--;
```

```
    if(burst_time[smallest] == 0)
```

```
    {
```

```
        count++;
```

```
        end = time + 1;
```

```
        wait_time = wait_time + end - arrival_time[smallest] - temp[smallest];
```

```
        turnaround_time = turnaround_time + end - arrival_time[smallest];
```

```
    }
```

```
}
```



```
average_waiting_time = wait_time / limit;  
average_turnaround_time = turnaround_time / limit;  
printf("\n\nAverage Waiting Time: %lf\n", average_waiting_time);  
printf("Average Turnaround Time: \t%lf\n", average_turnaround_time);  
return 0;  
}
```

Output

Enter the Total Number of Processes: 5

Enter Details of 5 Processes

Enter Arrival Time: 0

Enter Burst Time: 9

Enter Arrival Time: 1

Enter Burst Time: 4

Enter Arrival Time: 2

Enter Burst Time: 5

Enter Arrival Time: 3

Enter Burst Time: 7

Enter Arrival Time: 4

Enter Burst Time: 3

Average Waiting Time: 7.200000

Average Turnaround Time: 12.800000