

SCEDULING ALGORITHMS

JATIN KARTHIK TRIPATHY
17BCE7106

1. FCFS

```
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS  proc.sc  x  kts.c  x  sjf.c  x  priority.c  x  mc  x
  proc.sc
  kts.c
  kts.c
  prio
  C prio
  n
  C mc
  sjf
  G+ sjf.c
1 #include <stdio.h>
2 void findWT(int processes[], int n,
3             int bt[], int wt[])
4 {
5     wt[0] = 0;
6     for (int i = 1; i < n; i++)
7         wt[i] = bt[i-1] + wt[i-1];
8 }
9 void findTAT(int processes[], int n,
10             int bt[], int wt[], int tat[])
11 {
12     for (int i = 0; i < n; i++)
13         tat[i] = bt[i] + wt[i];
14 }
15 void findAvgT(int processes[], int n, int bt[])
16 {
17     int wt[n], tat[n], total_wt = 0, total_tat = 0;
18     findWT(processes, n, bt, wt);
19     findTAT(processes, n, bt, wt, tat);
20     printf("Processes  bt  wt  TAT\n");
21     for (int i = 0; i < n; i++)
22     {
23         total_wt = total_wt + wt[i];
24         total_tat = total_tat + tat[i];
25         printf(" %d ", i+1);
26         printf(" %d ", bt[i]);
27         printf(" %d ", wt[i]);
28         printf(" %d\n", tat[i]);
29     }
30     int s = (float)total_wt / (float)n;
31     int t = (float)total_tat / (float)n;
32     printf("Average waiting time = %d\n", s);
33     printf("Average turn around time = %d\n", t);
34 }
35
36 int main()
37 {
38     int processes[] = { 1, 2, 3};
39     int n = sizeof processes / sizeof processes[0];
40     int burst_time[] = {2, 1, 4};
41     findAvgT(processes, n, burst_time);
42     return 0;
43 }
```

2. SJF

```
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS  proc.sc  x  kts.c  x  sjf.c  x  priority.c  x  mc  x
  proc.sc
  kts.c
  kts.c
  prio
  C prio
  n
  C mc
  sjf
  G+ sjf.c
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 struct Process {
5     int pid;
6     int bt;
7     int art;
8 };
9 void findWT(Process proc[], int n, int wt[])
10 {
11     int rt[n];
12     for (int i = 0; i < n; i++)
13         rt[i] = proc[i].bt;
14
15     int complete = 0, t = 0, minm = INT_MAX;
16     int shortest = 0, finish_time;
17     bool check = false;
18     while (complete != n) {
19         for (int j = 0; j < n; j++) {
20             if ((proc[j].art <= t) &&
21                 (rt[j] < minm) && rt[j] > 0) {
22                 minm = rt[j];
23                 shortest = j;
24                 check = true;
25             }
26         }
27
28         if (check == false) {
29             t++;
30             continue;
31         }
32         rt[shortest]--;
33         minm = rt[shortest];
34         if (minm == 0)
35             minm = INT_MAX;
36         if (rt[shortest] == 0) {
37             complete++;
38             check = false;
39             finish_time = t + 1;
40             wt[shortest] = finish_time -
41                             proc[shortest].bt -
42                             proc[shortest].art;
43             if (wt[shortest] < 0)
44                 wt[shortest] = 0;
45             t++;
46         }
47     }
48 }
49
50 void findTAT(Process proc[], int n, int wt[], int tat[])
51 {
52 }
```


OUTPUTS

```
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/proc_scheduling$ gcc -o fcfs fcfs.c
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/proc_scheduling$ ./fcfs
Processes    BT    WT    TAT
1            2     0     2
2            7     2     9
3            4     9    13
Average waiting time = 3
Average turn around time = 8
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/proc_scheduling$ g++ -o sjf sjf.cpp
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/proc_scheduling$ ./sjf
Processes    Burst time    Waiting time    Turn around time
1            4            6            10
2            2            0             2
3            5            2             7
4            7            8            15
Average waiting time = 4
Average turn around time = 8.5
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/proc_scheduling$ gcc -o prio priority.c
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/proc_scheduling$ ./prio
Enter Total Number of Process:3

Enter Burst Time and Priority

P[1]
Burst Time:4
Priority:2

P[2]
Burst Time:5
Priority:1

P[3]
Burst Time:6
Priority:3

Process      Burst Time      Waiting Time      Turnaround Time
P[2]          5                0                 5
P[1]          4                5                 9
P[3]          6                9                15

Average Waiting Time=4
Average Turnaround Time=9
```

```
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/proc_scheduling$ gcc -o rr rr.c
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/proc_scheduling$ ./rr
Enter Total Process: 3
Enter Arrival Time and Burst Time for Process Number 1: 3 5
Enter Arrival Time and Burst Time for Process Number 2: 1 8
Enter Arrival Time and Burst Time for Process Number 3: 4 3
Enter Time Quantum: 2

Process |Turnaround Time|Waiting Time
P[3]    |      7      |      4
P[1]    |      9      |      4
P[2]    |     15      |      7

Average Waiting Time= 5.000000
Avg Turnaround Time = 10.333333
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/proc_scheduling$
```