

```
/* sjf scheduling (struct)(non preemptive)*/
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
typedef struct {
```

```
    int pid, btime, wtime, ttime; // Added ttime (turnaround time)
```

```
} sp; //sp=structure pointer(ptr)
```

```
int main() {
```

```
    int i, j, n, tbm = 0, totwtime = 0, totttime = 0; // total burst time = tbm
```

```
    sp *p, t; // p, t (temp) struct array is dynamic here
```

```
    printf("\n SJF scheduling ..\n");
```

```
    printf("Enter the number of processors: ");
```

```
    scanf("%d", &n);
```

```
    p = (sp *)malloc(n * sizeof(sp)); // Allocate memory for n processes
```

```
    printf("\n Enter the burst time:\n");
```

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

```
        printf("Process %d: ", i + 1);
```

```
        scanf("%d", &p[i].btime);
```

```
        p[i].pid = i + 1;
```

```
        p[i].wtime = 0;
```

```
    }
```

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

```
        for (j = i + 1; j < n; j++) {
```

```
            if (p[i].btime > p[j].btime) {
```

```
                t = p[i];
```

```
                p[i] = p[j];
```

```
                p[j] = t;
```

```

    }
}

printf("\n Process scheduling:\n");
printf(" Process \t Burst Time \t Waiting Time \t Turnaround Time\n");
for (i = 0; i < n; i++) {
    tbm += p[i].btime;
    p[i].ttime = tbm; //assigning total burst time to turn around time of p[i] as this whole is equal to
    //tt of p[i]
    p[i].wtime = tbm - p[i].btime; //as tt=bt+wt

    totwtime += p[i].wtime;
    totttime += p[i].ttime;

    printf(" %d\t\t %d\t\t %d\t\t %d\n", p[i].pid, p[i].btime, p[i].wtime, p[i].ttime);
}

printf("\n Total Waiting Time: %d\n", totwtime);
printf(" Average Waiting Time: %f\n", (float)totwtime / n);
printf(" Total Turnaround Time: %d\n", totttime);
printf(" Average Turnaround Time: %f\n", (float)totttime / n);

free(p); // Free allocated memory
return 0;
}

```

C Program for FCFS SchedulingOnline C CompilerSJF Scheduling Code Correction

tutorialspoint.com/compile_c_online.php

tutorialspointOnline C Compiler

ProjectEditSettingLogin

ExecuteBeautifyShareSource CodeHelp

```
1 /* SJF scheduling */
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 typedef struct {
6     int pid, btime, wtime, ttime; // Added ttime (turnaround time)
7 } sp; //sp=structure pointer(ptr)
8
9 int main() {
10     int i, j, n, tbn = 0, totwtime = 0, totttime = 0; // total burst time = tbn
11     sp *p, t; // p, t (temp) struct array is dynamic here
12
13     printf("\n SJF scheduling ..\n");
14     printf("Enter the number of processors: ");
15     scanf("%d", &n);
16     p = (sp *)malloc(n * sizeof(sp)); // Allocate memory for n processes
17
18     printf("\n Enter the burst time:\n");
19     for (i = 0; i < n; i++) {
20         printf("Process %d: ", i + 1);
21         scanf("%d", &p[i].btime);
22         p[i].pid = i + 1;
23         p[i].wtime = 0;
24     }
25
26     for (i = 0; i < n; i++) {
27         for (j = i + 1; j < n; j++) {
28             if (p[i].btime > p[j].btime) {
29                 t = p[i];
```

SJF scheduling ..

Enter the number of processors: 3

Enter the burst time:

Process 1: 10

Process 2: 5

Process 3: 15

Process scheduling:

Process	Burst Time	Waiting Time	Turnaround Time
2	5	0	5
1	10	5	15
3	15	15	30

Total Waiting Time: 20

Average Waiting Time: 6.666667

Total Turnaround Time: 50

Average Turnaround Time: 16.666666

29°

Search

ENG IN

20:34

09-09-2023