

```
/* fcfs scheduling using at(struct) */
```

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

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

```
struct fcfs {
```

```
    int pid, btime, wtime, ttime, atime;
```

```
};
```

```
int main() {
```

```
    int i, n;
```

```
    int totwtime = 0, totttime = 0;
```

```
    printf("\n FCFS Scheduling\n");
```

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

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

struct fcfs p[n]; //struct fcfs p[10]; declares an array named p that can hold n elements, and each element is of the struct fcfs type.

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

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

```
        printf("\nEnter burst time of process %d: ", p[i].pid);
```

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

```
        printf("Enter arrival time of process %d: ", p[i].pid);
```

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

```
    }
```

```
// Sort the processes by their arrival times
```

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

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

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

```
            struct fcfs temp = p[i];
```

```

        p[i] = p[j];
        p[j] = temp;
    }
}

```

```

// Calculate waiting time and turnaround time
for (i = 0; i < n; i++) { //index of struct array p is starting from 0 but pid starts from 1
    if (i > 0) {
        p[i].wtime = p[i - 1].wtime + p[i - 1].btime - p[i].atime;
        if (p[i].wtime < 0) {
            p[i].wtime = 0; //if the prv waiting time is (-)ve then it becomes 0 as waiting time cant be
negative
        }
    } else {
        p[i].wtime = 0; // for the first process
    }

    p[i].ttime = p[i].wtime + p[i].btime; //as bt+wt==tt
    totwtime += p[i].wtime;
    totttime += p[i].ttime;
}

printf("\nProcesses\tBurst\tWaiting\tTurnaround\tArrival\n");

for (i = 0; i < n; i++) {
    printf("%d\t%d\t%d\t%d\t%d\n", p[i].pid, p[i].btime, p[i].wtime, p[i].ttime, p[i].atime);
}

printf("\nAverage Waiting time = %f", (float)totwtime / n);
printf("\nAverage Turnaround time = %f\n", (float)totttime / n);

```

```
return 0;
```

```
}
```

The screenshot shows a web browser window with the URL `tutorialspoint.com/compile_c_online.php`. The page title is "Online C Compiler". The code editor contains the following C code for FCFS scheduling:

```
1 /* fcfs scheduling using at */
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 struct fcfs {
6     int pid, btime, wtime, ttime, atime;
7 };
8
9 int main() {
10     int i, n;
11     int totwtime = 0, totttime = 0;
12     printf("\n FCFS Scheduling\n");
13     printf("Enter the number of processes: ");
14     scanf("%d", &n);
15
16     struct fcfs p[n]; //struct fcfs p[10]; declares an array named p that can
17                       //hold n elements, and each element is of the struct fcfs type.
18
19     for (i = 0; i < n; i++) {
20         p[i].pid = i + 1;
21         printf("\nEnter burst time of process %d: ", p[i].pid);
22         scanf("%d", &p[i].btime);
23         printf("Enter arrival time of process %d: ", p[i].pid);
24         scanf("%d", &p[i].atime);
25     }
26
27     // Sort the processes by their arrival times
28     for (i = 0; i < n; i++) {
29         for (int j = i+1; j < n; j++) {
```

The terminal output shows the execution of the program:

```
FCFS Scheduling
Enter the number of processes: 3
Enter burst time of process 1: 5
Enter arrival time of process 1: 2
Enter burst time of process 2: 10
Enter arrival time of process 2: 0
Enter burst time of process 3: 15
Enter arrival time of process 3: 1
Processes  Burst  Waiting Turnaround Arrival
2         10    0       10         0
3         15    9       24         1
1          5    22       27         2

Average Waiting time = 10.333333
Average Turnaround time = 20.333334
```

The Windows taskbar at the bottom shows the date and time as 09-09-2023, 19:50.