STF(with arrival time)

Programiz
C Online Compiler

main.c

```c
1  #include<stdio.h>
2
3  int main() {
4      int time, burst_time[10], at[10], sum_burst_time = 0, smallest, n, i;
5      int sumt = 0, sumw = 0;
6      printf("enter the no of processes : ");
7      scanf("%d", & n);
8      for (i = 0; i < n; i++) {
9          printf("the arrival time for process P%d : ", i + 1);
10         scanf("%d", & at[i]);
11         printf("the burst time for process P%d : ", i + 1);
12         scanf("%d", & burst_time[i]);
13         sum_burst_time += burst_time[i];
14     }
15     burst_time[9] = 9999;
16     for (time = 0; time < sum_burst_time;) {
17         smallest = 9;
18         for (i = 0; i < n; i++) {
19             if (at[i] <= time && burst_time[i] > 0 && burst_time[i] <
                   burst_time[smallest])
20                 smallest = i;
21         }
22         printf("P[%d]\t|\t%d\t|\t%d\n", smallest + 1, time + burst_time[smallest]
               - at[smallest], time - at[smallest]);
23         sumt += time + burst_time[smallest] - at[smallest];
24         sumw += time - at[smallest];
25         time += burst_time[smallest];
```

Output

```
/tmp/mYgPKDSoMZ.o
enter the no of processes : 4
the arrival time for process P1 : 0
the burst time for process P1 : 5
the arrival time for process P2 : 1
the burst time for process P2 : 3
the arrival time for process P3 : 2
the burst time for process P3 : 3
the arrival time for process P4 : 4
the burst time for process P4 : 1
P[1]    |    5    |    0
P[4]    |    2    |    1
P[2]    |    8    |    5
P[3]    |   10    |    7


average waiting time = 3.250000


average turnaround time = 6.250000
```

---

main.c

```c
9       printf("the arrival time for process P%d : ", i + 1);
10      scanf("%d", & at[i]);
11      printf("the burst time for process P%d : ", i + 1);
12      scanf("%d", & burst_time[i]);
13      sum_burst_time += burst_time[i];
14  }
15  burst_time[9] = 9999;
16  for (time = 0; time < sum_burst_time;) {
17      smallest = 9;
18      for (i = 0; i < n; i++) {
19          if (at[i] <= time && burst_time[i] > 0 && burst_time[i] <
                burst_time[smallest])
20              smallest = i;
21      }
22      printf("P[%d]\t|\t%d\t|\t%d\n", smallest + 1, time + burst_time[smallest]
            - at[smallest], time - at[smallest]);
23      sumt += time + burst_time[smallest] - at[smallest];
24      sumw += time - at[smallest];
25      time += burst_time[smallest];
26      burst_time[smallest] = 0;
27  }
28  printf("\n\n average waiting time = %f", sumw * 1.0 / n);
29  printf("\n\n average turnaround time = %f", sumt * 1.0 / n);
30  return 0;
31 }
32
33
```

Output

```
/tmp/mYgPKDSoMZ.o
enter the no of processes : 4
the arrival time for process P1 : 0
the burst time for process P1 : 5
the arrival time for process P2 : 1
the burst time for process P2 : 3
the arrival time for process P3 : 2
the burst time for process P3 : 3
the arrival time for process P4 : 4
the burst time for process P4 : 1
P[1]    |    5    |    0
P[4]    |    2    |    1
P[2]    |    8    |    5
P[3]    |   10    |    7


average waiting time = 3.250000


average turnaround time = 6.250000
```