

15/6/23

FCFS

#include <stdio.h>

```
void waitingtime ( int proc[], int n, int burst_time[],  
int wait_time[] ) {
```

```
    wait_time[0] = 0;
```

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

```
        wait_time[i] = burst_time[i - 1] +
```

```
        wait_time[i - 1];
```

```
}
```

3

```
void turnaroundtime ( int proc[], int n, int burst_ -  
time[], int wait_time[], int tat[] ) {
```

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

```
        tat[i] = burst_time[i] + wait_time[i];
```

3

3

```
void avgtime ( int proc[], int n, int burst_time[] ) {
```

```
    int wait_time[n], tat[n], total_wt = 0,
```

```
    total_wt = 0;
```

```
    waitingtime ( proc, n, burst_time, wait_time );
```

```
    turnaroundtime ( proc, n, burst_time, wait_time,  
            tat );
```

```
    printf (" process \t burst_time \t waiting_time  
           \t turnaround_time \n");
```

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

```
        total_wt += wait_time[i];
```

```
        total_tat += tat[i];
```

```
    printf ("%d\t%d\t%d\t%.2f\t%.2f\n", proc[i],  
           burst_time[i], wait_time[i], tat[i]);
```

3

1

float avg_wt = (float) total_wt / n;
float avg_tat = (float) total_tat / n;
printf ("Avg. waiting time = %0.2f",
 avg_wt);

printf ("Avg. Turnaround time = %0.2f",
 avg_tat);

3 do calculation

```
#include <stdio.h>  
int proc[3] = {1, 2, 3};  
int n = size(proc) / size(proc[0]);  
int burst_time[3] = {10, 5, 8};  
avg_time(proc, n, burst_time);  
return 0;
```

int main () {
 printf(proc[0]);

int n = 10, i;
int burst_time[10];
for (i=0; i<n; i++)

int burst_time[i] = i+1;

for (i=0; i<10; i++) {

printf("enter val %d:", i+1);
 scanf("%d", &burst_time[i]);

3 do calculation

avgtime(proc, n, burst_time);
return 1;

3 do calculation

:(D) exit, (D) wait for read (W) with found

OUTPUT -

702

FEB 11/2020

→ enter burst time for 3 processes

2 5 7

<1.0ms> burst time

processes burst time waiting time turnaround time

processes	burst time	waiting time	turnaround time
1	2	0	2
2	5	2	7
3	7	7	14

average waiting time = $\frac{3}{3} = 1$ ms

average turnaround time = $\frac{7}{3} = 2.33$ ms

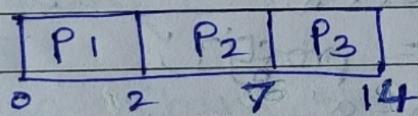
returned O(0x0) execution time = 9.166 s

CP unit used = 0.0001 sec

W = CP unit used

W = CP unit used

graph →



$\sqrt{6/27}$

(i) (C) arrival times first (J) with priorities due

3 (C) arrival times due

1 2 3

2 3 1 0 1 2 3

3 (C) arrival times due

(B) arrival times due + D = D arrival times

(E) arrival times due + D

1 2 3 0 1 2 3

and (C) arrival times due + D = D arrival times

(D) arrival times due + D = D arrival times

1 2 3 0 1 2 3

3 (C) arrival times due

SJF

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

```
void sort (int burst_time[], int n, int proc[])
{
    int i, j, a;
    for (i=0; i<n; i++) {
        for (j=i+1; j<n; j++) {
            if (burst_time[i] > burst_time[j]) {
                a = burst_time[i];
                burst_time[i] = burst_time[j];
                burst_time[j] = a;
                a = proc[i];
                proc[i] = proc[j];
                proc[j] = a;
            }
        }
    }
}
```

```
int waiting_time (int wait_time[], int n,
                  int burst_time[])
{
    int i;
    int tot_wt = 0;
    for (i=1; i<n; i++) {
        wait_time[i] = wait_time[i-1] + burst_time[i-1];
        tot_wt += wait_time[i];
    }
    return tot_wt;
}
```

```
int turnaround (int wait_time[], int burst_time[])
{
    int n, int num_around[][];
    int i, tot_tt;
    for (i=0; i<n; i++) {
        tot_tt = wait_time[i] + burst_time[i];
        num_around[i][0] = tot_tt;
    }
}
```

$\text{turn-around}[i] = \text{wait-time}[i] + \text{burst-time}[i];$
 $\text{tot-tt} = \text{turn-around}[i];$
 } return tot-tt;
 }

```

int main() {
    int n, i, tot-wt, tot-tt;
    int proc[100], burst-time[100];
    wait-time[100], turn-around[100];
    printf("enter no. of processes:");
    scanf("%d", &n);
    for (i=0; i<n; i++) {
        proc[i] = i+1;
        printf("enter burst time %d:", i+1);
        scanf("%d", &burst-time[i]);
    }
    sort(burst-time, n, proc);
    wait-time[0]=0;
    tot-wt = waiting-time(wait-time, burst-time,
                           n);
    tot-tt = turnaround(wait-time, burst-time, n,
                         turn-around);
    printf("In Process\tBurst time\tWait time\t
           Turn around time\n");
    for (i=0; i<n; i++) {
        printf("%d\t%d\t%d\t%d\n",
               proc[i], burst-time[i], wait-time[i],
               burst-time[i]+wait-time[i]);
    }
    printf("\n");
}

printf("average waiting time: %d\n", tot-wt/n);
printf("average turnaround time: %d\n", tot-tt/n);
}
```

22/6/2

Ques. - 6.0 with time = 5.0 burst - ready

enter no. of processes - 3

enter the burst time of each process - 12 3 4

3.00 4.00 12.000

3.00 ms ready

process no. = 1 (unit time) waiting time

burst time = 3.000 ms (5.000 ms org)

waiting time = 0.000 ms (5.000 ms org)

turnaround time = 3.000 ms (5.000 ms org)

process no. = 2 (unit time) waiting time

burst time = 4.000 ms (5.000 ms org)

waiting time = 3.000 ms (5.000 ms org)

turnaround time = 7.000 ms (5.000 ms org)

process no. = 3 (unit time) waiting time

burst time = 12.000 ms (5.000 ms org)

waiting time = 7.000 ms (5.000 ms org)

turnaround time = 19.000 ms (5.000 ms org)

22/6/27

average waiting time = 3.333

average turnaround time = 9.666

graph

P ₁	P ₂	P ₃
0	3	7