



```
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF: ~/Desktop
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~$ cd Desktop
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ cc programas.c
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ ./a.out
before fork
M child's id is 3488
I am parent having id 3487
common
I am child having id 3488
My parent's id is 3488
common
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$
```

```
{
printf("M child's id is %d\n",p);
printf("I am parent having id %d\n",getpid());
}
printf("common\n");
}
```

 Open  Save   Undo      

filecopying.c x copyfile.c x hello.c x

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    FILE *fptr1, *fptr2;
    char filename[100], c;
    printf("Enter the filename to open for reading \n");
    scanf("%s", filename);
    fptr1 = fopen(filename, "r");
    if (fptr1 == NULL){
        printf("Cannot open file %s \n", filename);
        exit(0);
    }
    printf("Enter the filename to open for writing \n");
    scanf("%s", filename);
    fptr2 = fopen(filename, "w");
    if (fptr2 == NULL){
        printf("Cannot open file %s \n", filename);
        exit(0);
    }
    c = fgetc(fptr1);
    while (c != EOF){
        fputc(c, fptr2);
        c = fgetc(fptr1);
    }
    printf("\nContents copied to %s", filename);
    fclose(fptr1);
    fclose(fptr2);
    return 0;
}
```

guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF: ~/Desktop  
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~\$ cd Desktop  
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop\$ cc copyfile.c  
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop\$ ./a.out  
Enter the filename to open for reading  
file3.txt  
Cannot open file file3.txt  
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop\$ cc copyfile.c  
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop\$ ./a.out  
Enter the filename to open for reading  
filecopying.c  
Enter the filename to open for writing  
hello.c  
Contents copied to hello.c

```
c = fgetc(fp1);  
}  
printf("\nContents copied to %s", filename);  
fclose(fp1);  
fclose(fp2);  
return 0;  
}
```



TDM-GCC 9.2.0 64-bit Release

(globals)

Project Classes fcfs\_scheduling.cpp x sjfscheduling.cpp x priority scheduling.cpp x

```

1  #include<stdio.h>
2  int main()
3
4      int bt[10]={0},at[10]={0},tat[10]={0},wt[10]={0},ct[10]={0};
5      int n,sum=0;
6      float totalTAT=0,totalWT=0;
7      printf("Enter number of processes ");
8      scanf("%d",&n);
9      printf("Enter arrival time and burst time for each process\n\n");
10     for(int i=0;i<n;i++)
11     {
12         printf("Arrival time of process[%d] ",i+1);
13         scanf("%d",&at[i]);
14         printf("Burst time of process[%d] ",i+1);
15         scanf("%d",&bt[i]);
16         printf("\n");
17     }
18     for(int j=0;j<n;j++)
19     {
20         sum+=bt[j];
21         ct[j]=sum;
22     }
23     for(int k=0;k<n;k++)
24     {
25         tat[k]=ct[k]-at[k];
26         totalTAT+=tat[k];
27     }
28     for(int k=0;k<n;k++)
29     {
30         wt[k]=tat[k]-bt[k];
31         totalWT+=wt[k];
32     }
33     printf("Solution: \n\n");
34     printf("P#\t AT\t BT\t CT\t TAT\t WT\n\n");
35     for(int i=0;i<n;i++)
36     {
37         printf("P%d\t %d\t %d\t %d\t %d\t %d\n",i+1,at[i],bt[i],ct[i],tat[i],wt[i]);
38     }
39     printf("\n\nAverage Turnaround Time = %f\n",totalTAT/n);
40     printf("Average WT = %f\n\n",totalWT/n);
41     return 0;
42

```

C:\Users\DELL\OneDrive\Documents\fcfs\_scheduling.exe

Enter number of processes 4  
 Enter arrival time and burst time for each process

Arrival time of process[1] 0  
 Burst time of process[1] 34

Arrival time of process[2] 4  
 Burst time of process[2] 56

Arrival time of process[3] 7  
 Burst time of process[3] 67

Arrival time of process[4] 18  
 Burst time of process[4] 56

Solution:

P#	AT	BT	CT	TAT	WT
P1	0	34	34	34	0
P2	4	56	90	86	30
P3	7	67	157	150	83
P4	18	56	213	195	139

Average Turnaround Time = 116.250000

Average WT = 63.000000



Project Classes fcfsscheduling.cpp x sjfsscheduling.cpp x priorityscheduling.cpp x

```

1  #include <stdio.h>
2  int main()
3  {
4      int A[100][4];
5      int i, j, n, total = 0, index, temp; float avg_wt, avg_tat;
6      printf("Enter number of process: "); scanf("%d", &n);
7      printf("Enter Burst Time:\n");
8      for (i = 0; i < n; i++) {
9          printf("P%d: ", i + 1); scanf("%d", &A[i][1]); A[i][0] = i + 1;
10     }
11     for (i = 0; i < n; i++) {
12         index = i;
13         for (j = i + 1; j < n; j++)
14             if (A[j][1] < A[index][1]) index = j;
15         temp = A[i][1]; A[i][1] = A[index][1]; A[index][1] = temp;
16         temp = A[i][0];
17         A[i][0] = A[index][0]; A[index][0] = temp;
18     }
19     A[0][2] = 0;
20     for (i = 1; i < n; i++) {
21         A[i][2] = 0;
22         for (j = 0; j < i; j++)
23             A[i][2] += A[j][1];
24         total += A[i][2];
25     }
26     avg_wt = (float)total / n; total = 0;
27     printf("P BT WT TAT\n"); for (i = 0; i < n; i++) {
28         A[i][3] = A[i][1] + A[i][2];
29         total += A[i][3];
30         printf("P%d %d %d %d\n", A[i][0], A[i][1], A[i][2], A[i][3]);
31     }
32     avg_tat = (float)total / n;
33     printf("Average Waiting Time= %f", avg_wt); printf("\nAverage Turnaround Time= %f", avg_tat);
34 }

```

```

C:\Users\DELL\OneDrive\Documents\sjfsscheduling.exe
Enter number of process: 4
Enter Burst Time:
P1: 25
P2: 12
P3: 6
P4: 18
P BT WT TAT
P3 6 0 6
P2 12 6 18
P4 18 18 36
P1 25 36 61
Average Waiting Time= 15.000000
Average Turnaround Time= 30.250000
.....
Process exited after 17.19 seconds with return value 0
Press any key to continue . . .

```



TDM-GCC 9.2.0 64-bit Release

(globals)

Project Classes fcfsscheduling.cpp x sjfscheduling.cpp x priorityscheduling.cpp x

```

1  #include <stdio.h>
2  int main()
3  {
4      int bt[20], p[20], wt[20], tat[20], i, j, n, total=0, pos, temp;
5      float avg_wt, avg_tat;
6      printf("Enter number of process:");
7      scanf("%d", &n);
8      printf("\nEnter Burst Time:\n");
9      for(i=0; i<n; i++)
10     {
11         printf("p%d:", i+1);
12         scanf("%d", &bt[i]);
13         p[i]=i+1;
14     }
15     for(i=0; i<n; i++)
16     {
17         pos=i;
18         for(j=i+1; j<n; j++)
19         {
20             if(bt[j]<bt[pos])
21                 pos=j;
22         }
23         temp=bt[i];
24         bt[i]=bt[pos];
25         bt[pos]=temp;
26         temp=p[i];
27         p[i]=p[pos];
28         p[pos]=temp;
29     }
30     wt[0]=0;
31     for(i=1; i<n; i++)
32     {
33         wt[i]=0;
34         for(j=0; j<i; j++)
35             wt[i]=wt[j];
36     }
37     total=wt[i];
38     avg_wt=(float)total/n;
39     total=0;
40     printf("\nProcess\t\t Burst Time\t\t \tWaiting Time\tTurnaround Time");
41     for(i=0; i<n; i++)
42     {
43         tat[i]=bt[i]+wt[i];
44         total=tat[i];
45         printf("\np%d\t\t\t\t %d\t\t\t\t %d\t\t\t\t %d\t\t\t\t %d", p[i], bt[i], wt[i], tat[i]);
46     }
47     avg_tat=(float)total/n;
48     printf("\n\nAverage Waiting Time=%f", avg_wt);
49     printf("\n\nAverage Turnaround Time=%f\n", avg_tat);
50 }
51

```

```

Select C:\Users\DELL\OneDrive\Documents\priorityscheduling.exe
Enter number of process:4
Enter Burst Time:
p1:25
p2:12
p3:6
p4:18

Process      Burst Time      Waiting Time      Turnaround Time
p3           6                0                6
p2           12                6               18
p4           18                18              36
p1           25                36              61

Average Waiting Time=15.000000
Average Turnaround Time=30.250000

-----
Process exited after 45.25 seconds with return value 0
Press any key to continue . . .

```