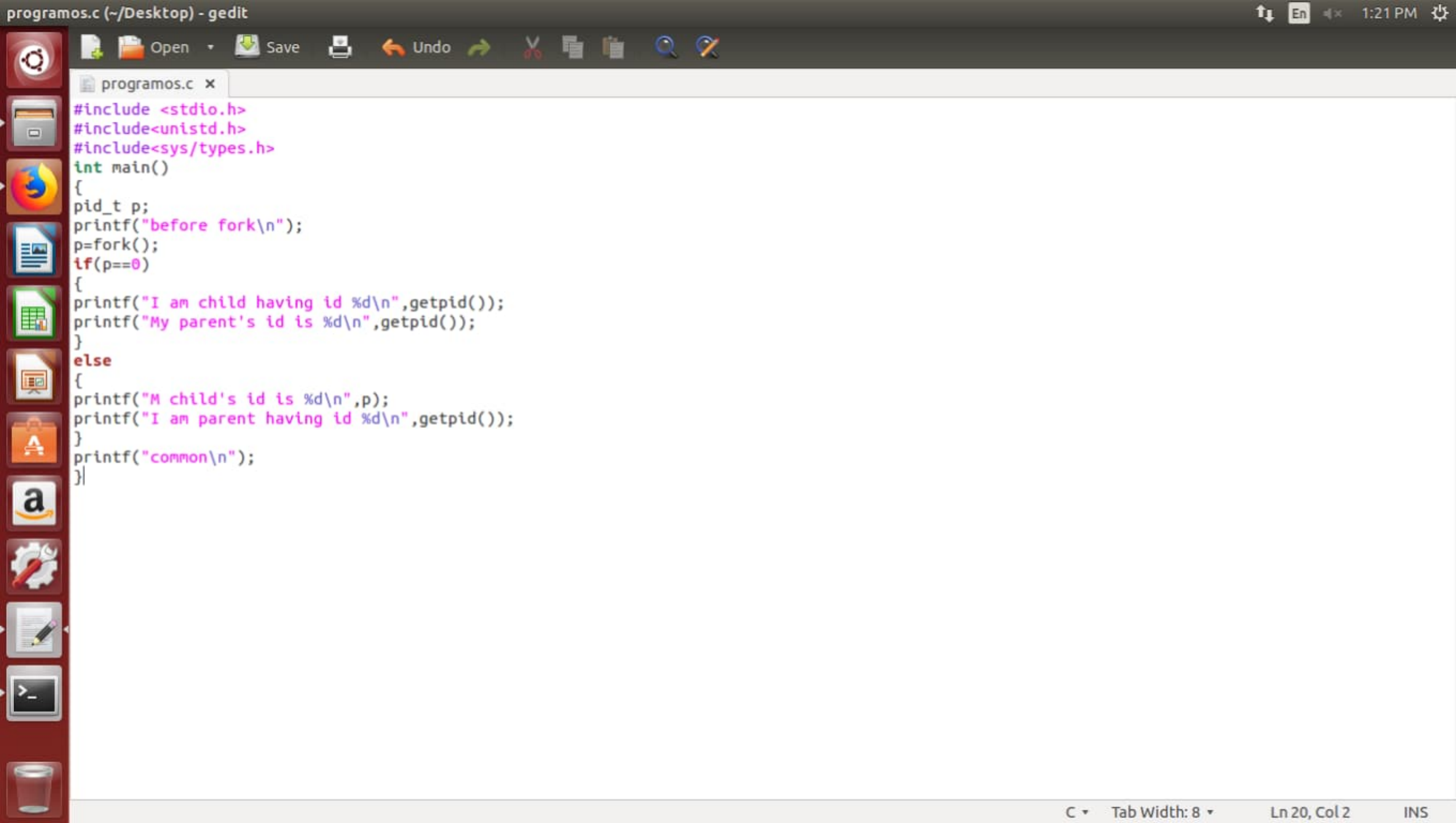
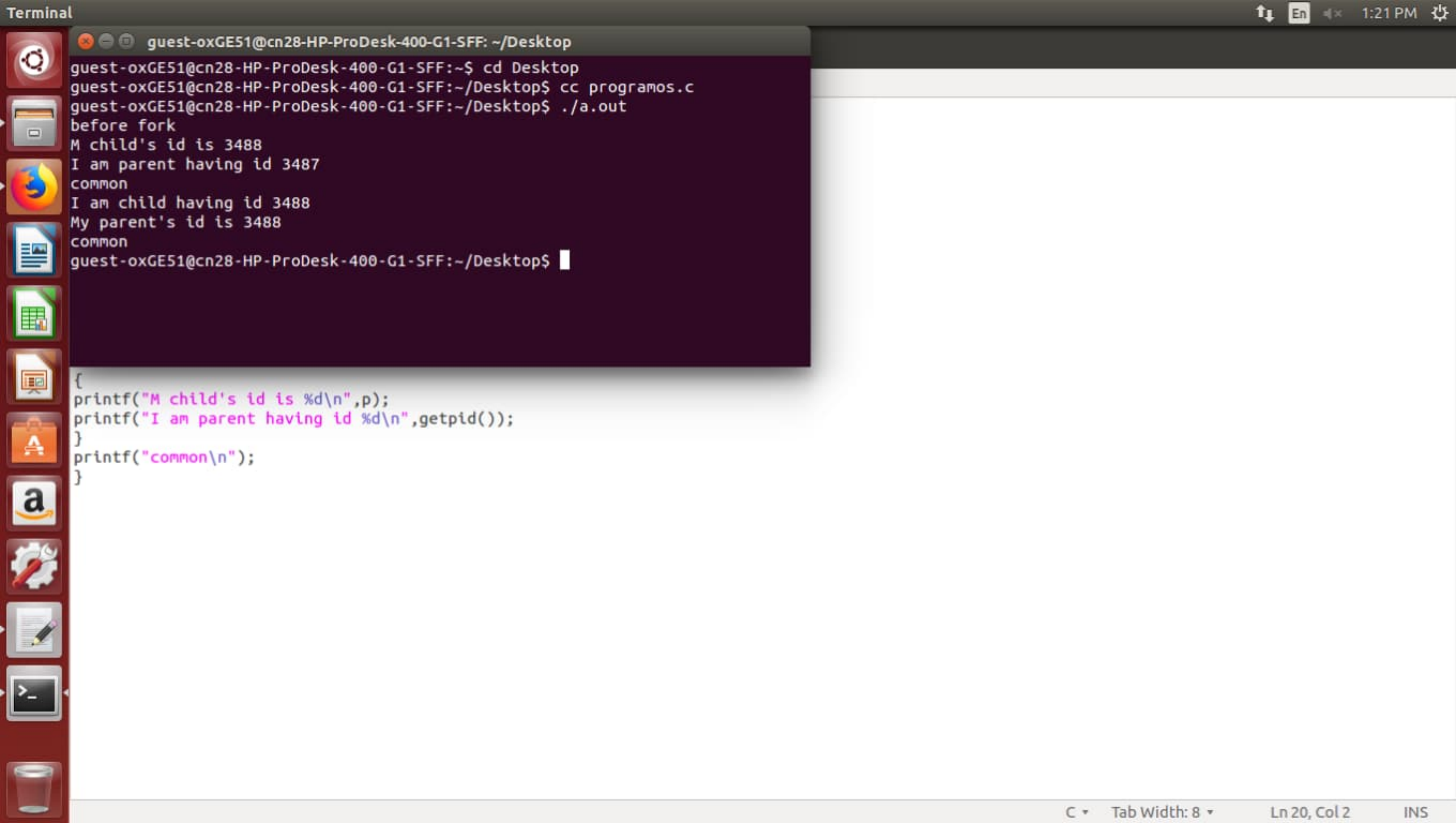
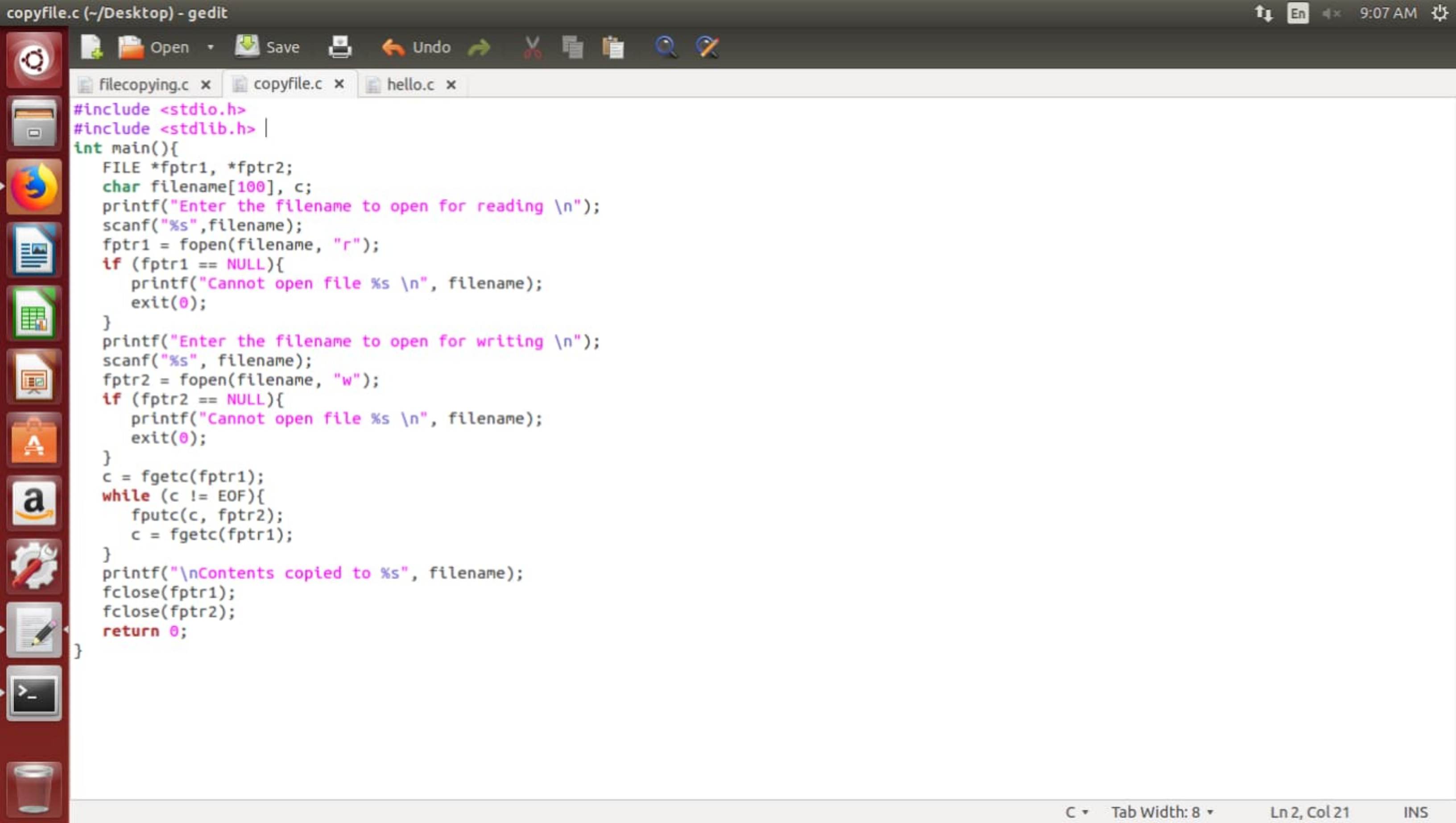


```c
#include <stdio.h>
#include<unistd.h>
#include<sys/types.h>
int main()
{
pid_t p;
printf("before fork\n");
p=fork();
if(p==0)
{
printf("I am child having id %d\n",getpid());
printf("My parent's id is %d\n",getpid());
}
else
{
printf("M child's id is %d\n",p);
printf("I am parent having id %d\n",getpid());
}
printf("common\n");
}
```

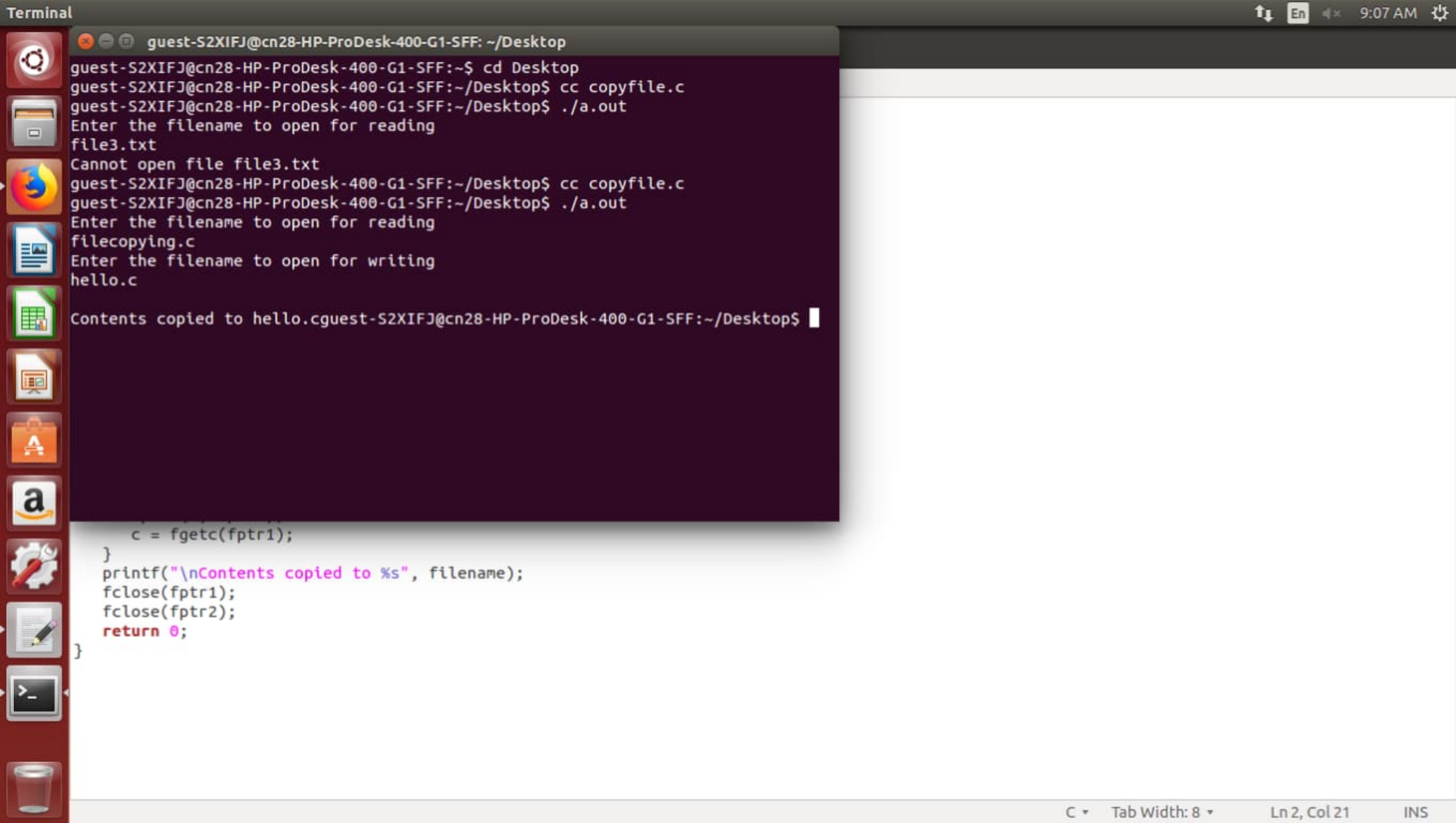


```
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~$ cd Desktop
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ cc programos.c
guest-oxGE51@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ ./a.out
before fork
M child's id is 3488
I am parent having id 3487
common
I am child having id 3488
My parent's id is 3488
common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

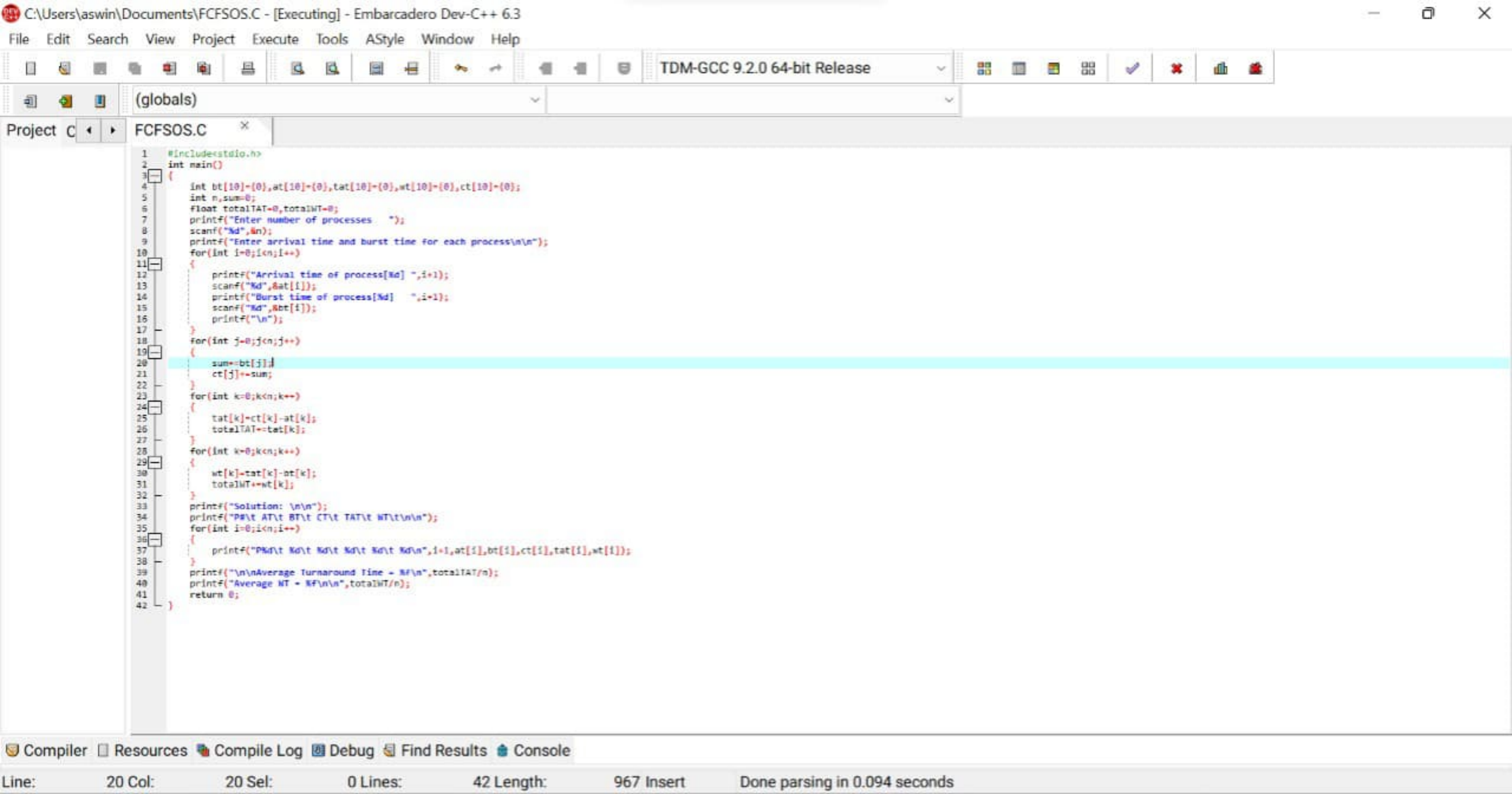


```c
#include <stdio.h>
#include <stdlib.h>
int main(){
   FILE *fptr1, *fptr2;
   char filename[100], c;
   printf("Enter the filename to open for reading \n");
   scanf("%s",filename);
   fptr1 = fopen(filename, "r");
   if (fptr1 == NULL){
      printf("Cannot open file %s \n", filename);
      exit(0);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



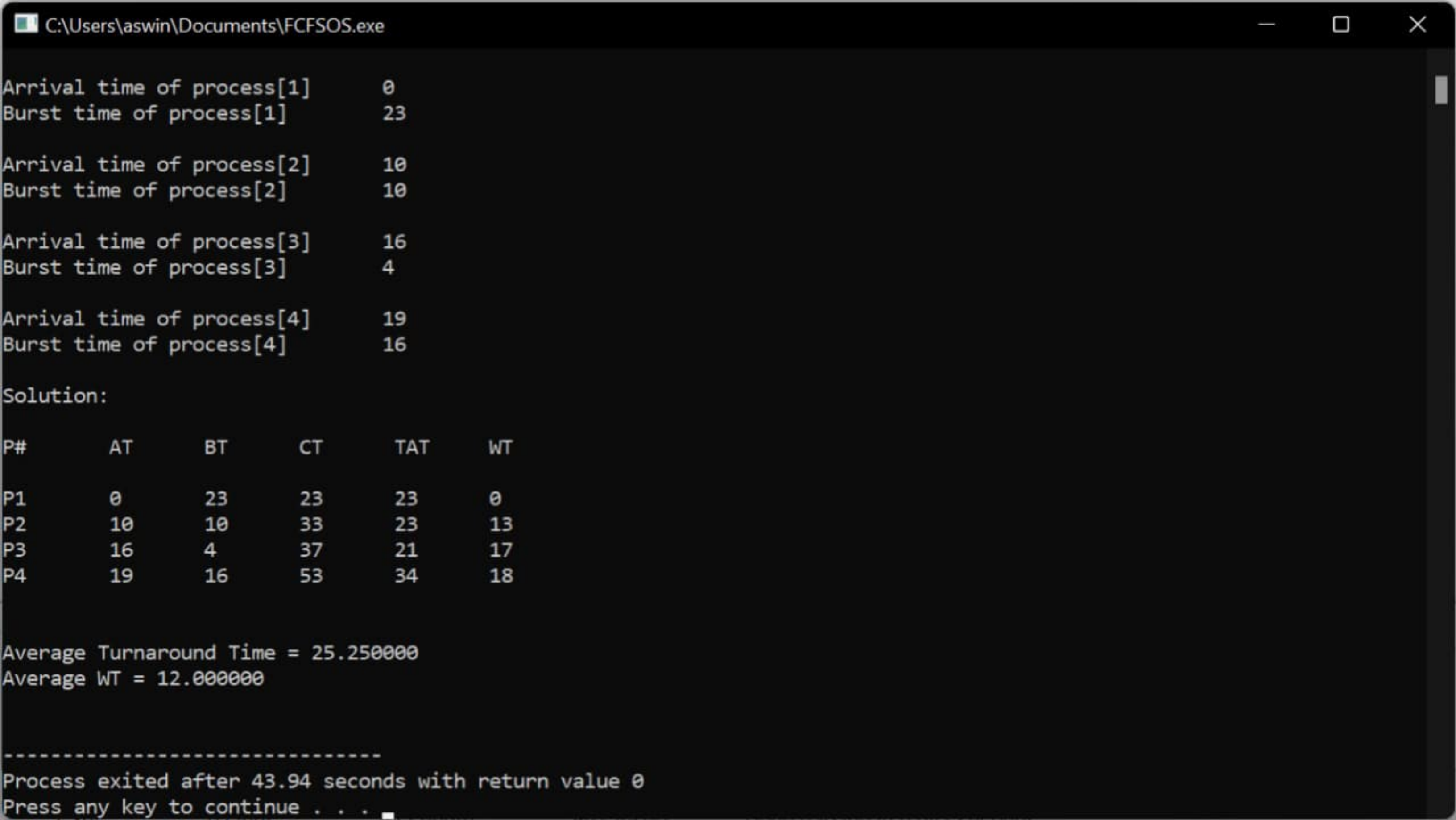
```
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~$ cd Desktop
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ cc copyfile.c
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$ ./a.out
Enter the filename to open for reading
file3.txt
Cannot open file file3.txt
guest-S2XIFJ@cn28-HP-ProDesk-400-G1-SFF:~/Desktop$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

```
        c = fgetc(fptr1);
    }
    printf("\nContents copied to %s", filename);
    fclose(fptr1);
    fclose(fptr2);
    return 0;
}
```

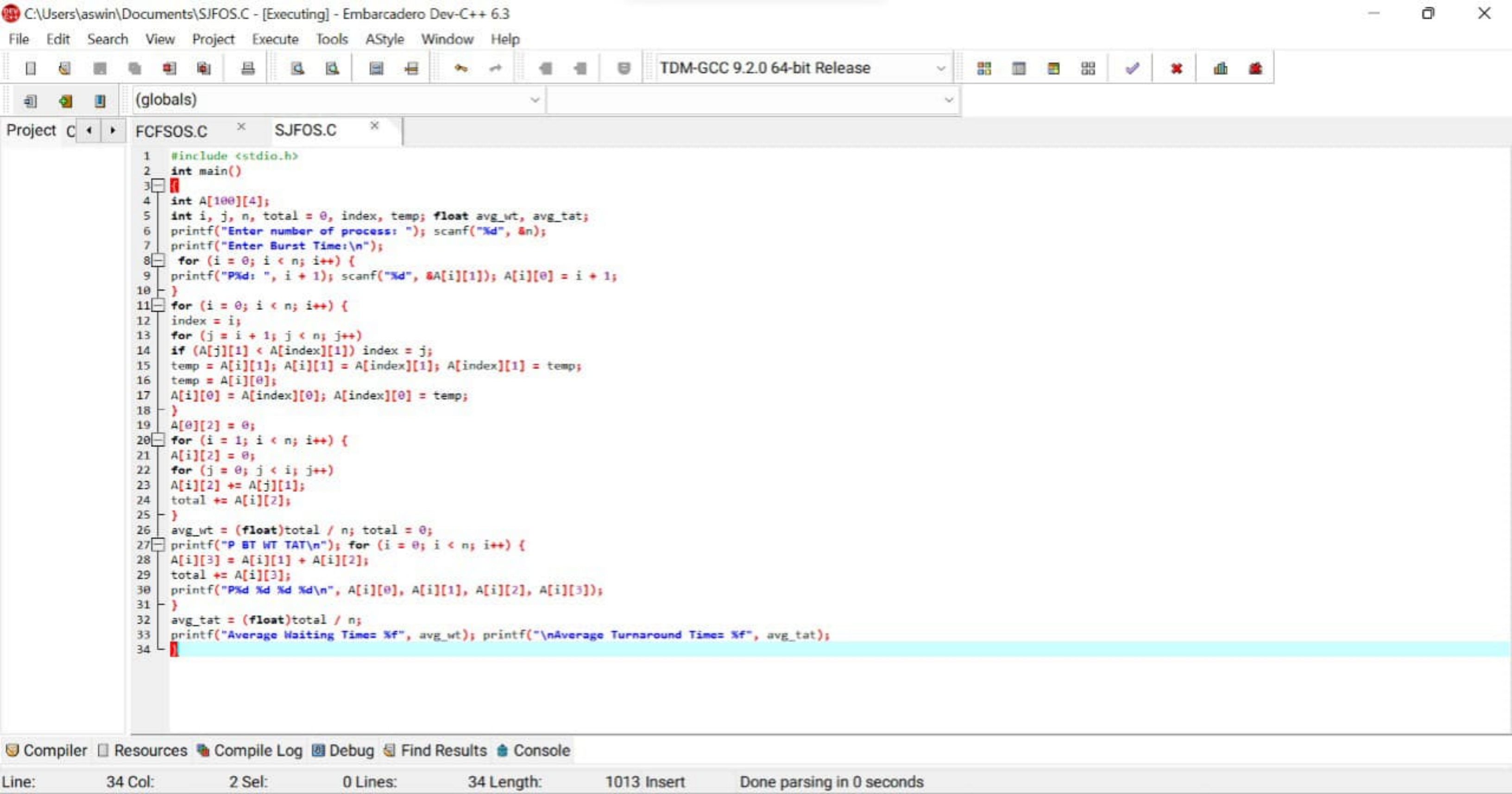

C:\Users\aswin\Documents\FCFSOS.C - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project
FCFSOS.C
Compiler Resources Compile Log Debug Find Results Console

```c
#include<stdio.h>
int main()
{
    int bt[10]={0},at[10]={0},tat[10]={0},wt[10]={0},ct[10]={0};
    int n,sum=0;
    float totalTAT=0,totalWT=0;
    printf("Enter number of processes   ");
    scanf("%d",&n);
    printf("Enter arrival time and burst time for each process\n\n");
    for(int i=0;i<n;i++)
    {
        printf("Arrival time of process[%d] ",i+1);
        scanf("%d",&at[i]);
        printf("Burst time of process[%d]   ",i+1);
        scanf("%d",&bt[i]);
        printf("\n");
    }
    for(int j=0;j<n;j++)
    {
        sum+=bt[j];
        ct[j]+=sum;
    }
    for(int k=0;k<n;k++)
    {
        tat[k]=ct[k]-at[k];
        totalTAT+=tat[k];
    }
    for(int k=0;k<n;k++)
    {
        wt[k]=tat[k]-bt[k];
        totalWT+=wt[k];
    }
    printf("Solution: \n\n");
    printf("P#\t AT\t BT\t CT\t TAT\t WT\t\n\n");
    for(int i=0;i<n;i++)
    {
        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]);
    }
    printf("\n\nAverage Turnaround Time = %f\n",totalTAT/n);
    printf("Average WT = %f\n\n",totalWT/n);
    return 0;
}
```

Line: 20 Col: 20 Sel: 0 Lines: 42 Length: 967 Insert Done parsing in 0.094 seconds



```
Arrival time of process[1]      0
Burst time of process[1]        23

Arrival time of process[2]      10
Burst time of process[2]        10

Arrival time of process[3]      16
Burst time of process[3]        4

Arrival time of process[4]      19
Burst time of process[4]        16

Solution:

P#      AT      BT      CT      TAT     WT

P1      0       23      23      23      0
P2      10      10      33      23      13
P3      16      4       37      21      17
P4      19      16      53      34      18


Average Turnaround Time = 25.250000
Average WT = 12.000000



--------------------------------
Process exited after 43.94 seconds with return value 0
Press any key to continue . . .
```



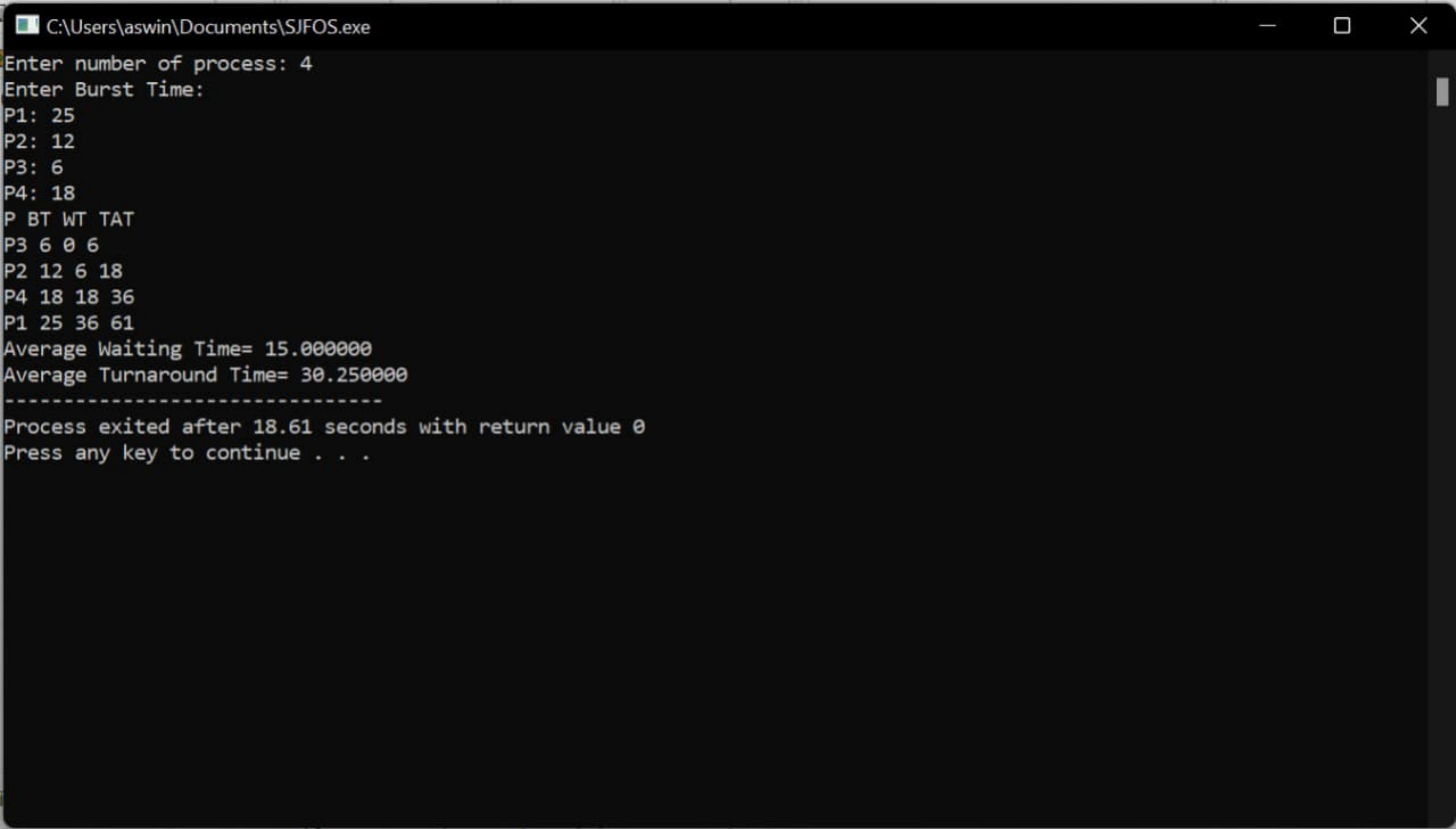
C:\Users\aswin\Documents\SJFOS.C - [Executing] - Embarcadero Dev-C++ 6.3

TDM-GCC 9.2.0 64-bit Release

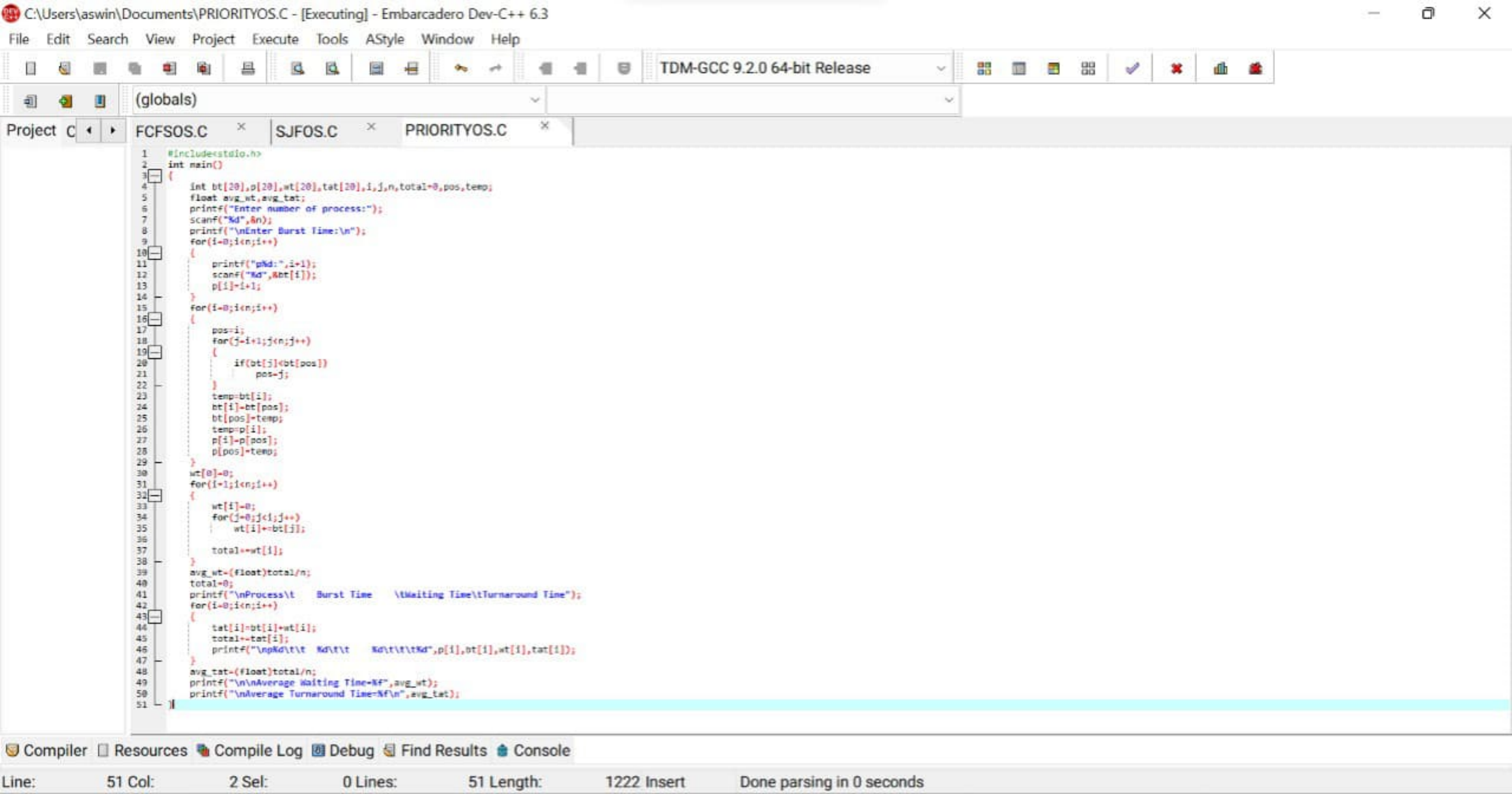
(globals)

```c
#include <stdio.h>
int main()
{
int A[100][4];
int i, j, n, total = 0, index, temp; float avg_wt, avg_tat;
printf("Enter number of process: "); scanf("%d", &n);
printf("Enter Burst Time:\n");
 for (i = 0; i < n; i++) {
printf("P%d: ", i + 1); scanf("%d", &A[i][1]); A[i][0] = i + 1;
}
for (i = 0; i < n; i++) {
index = i;
for (j = i + 1; j < n; j++)
if (A[j][1] < A[index][1]) index = j;
temp = A[i][1]; A[i][1] = A[index][1]; A[index][1] = temp;
temp = A[i][0];
A[i][0] = A[index][0]; A[index][0] = temp;
}
A[0][2] = 0;
for (i = 1; i < n; i++) {
A[i][2] = 0;
for (j = 0; j < i; j++)
A[i][2] += A[j][1];
total += A[i][2];
}
avg_wt = (float)total / n; total = 0;
printf("P BT WT TAT\n"); for (i = 0; i < n; i++) {
A[i][3] = A[i][1] + A[i][2];
total += A[i][3];
printf("P%d %d %d %d\n", A[i][0], A[i][1], A[i][2], A[i][3]);
}
avg_tat = (float)total / n;
printf("Average Waiting Time= %f", avg_wt); printf("\nAverage Turnaround Time= %f", avg_tat);
}
```

Line: 34 Col: 2 Sel: 0 Lines: 34 Length: 1013 Insert Done parsing in 0 seconds



```
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 18.61 seconds with return value 0
Press any key to continue . . .
```



C:\Users\aswin\Documents\PRIORITYOS.C - [Executing] - Embarcadero Dev-C++ 6.3

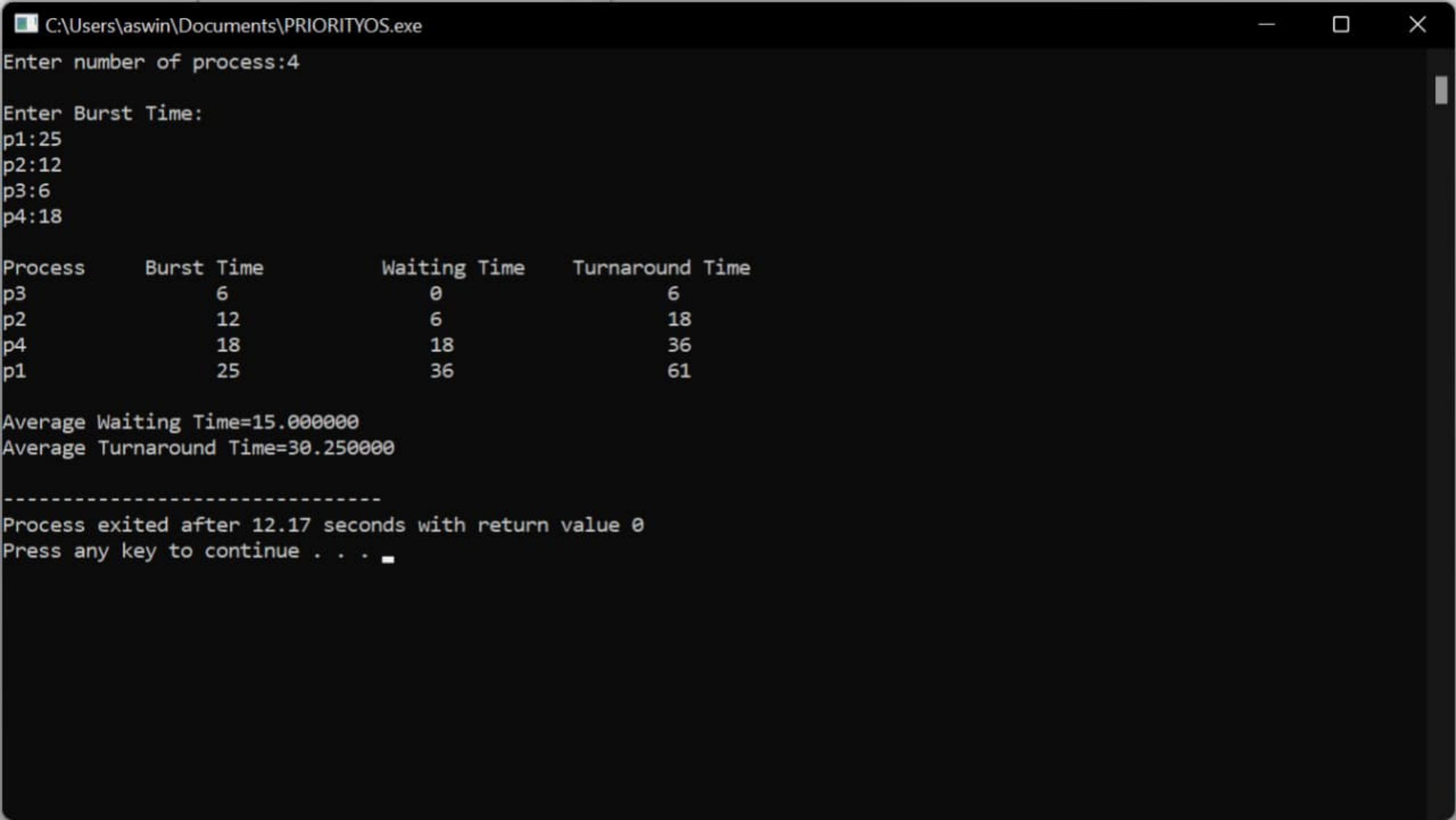
TDM-GCC 9.2.0 64-bit Release

(globals)

FCFSOS.C SJFOS.C PRIORITYOS.C

```c
#include<stdio.h>
int main()
{
    int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
    float avg_wt,avg_tat;
    printf("Enter number of process:");
    scanf("%d",&n);
    printf("\nEnter Burst Time:\n");
    for(i=0;i<n;i++)
    {
        printf("p%d:",i+1);
        scanf("%d",&bt[i]);
        p[i]=i+1;
    }
    for(i=0;i<n;i++)
    {
        pos=i;
        for(j=i+1;j<n;j++)
        {
            if(bt[j]<bt[pos])
                pos=j;
        }
        temp=bt[i];
        bt[i]=bt[pos];
        bt[pos]=temp;
        temp=p[i];
        p[i]=p[pos];
        p[pos]=temp;
    }
    wt[0]=0;
    for(i=1;i<n;i++)
    {
        wt[i]=0;
        for(j=0;j<i;j++)
            wt[i]+=bt[j];

        total+=wt[i];
    }
    avg_wt=(float)total/n;
    total=0;
    printf("\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time");
    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];
        total+=tat[i];
        printf("\np%d\t\t  %d\t\t    %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
    }
    avg_tat=(float)total/n;
    printf("\n\nAverage Waiting Time=%f",avg_wt);
    printf("\nAverage Turnaround Time=%f\n",avg_tat);
}
```

Line: 51 Col: 2 Sel: 0 Lines: 51 Length: 1222 Insert Done parsing in 0 seconds



C:\Users\aswin\Documents\PRIORITYOS.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 12.17 seconds with return value 0
Press any key to continue . . .
```