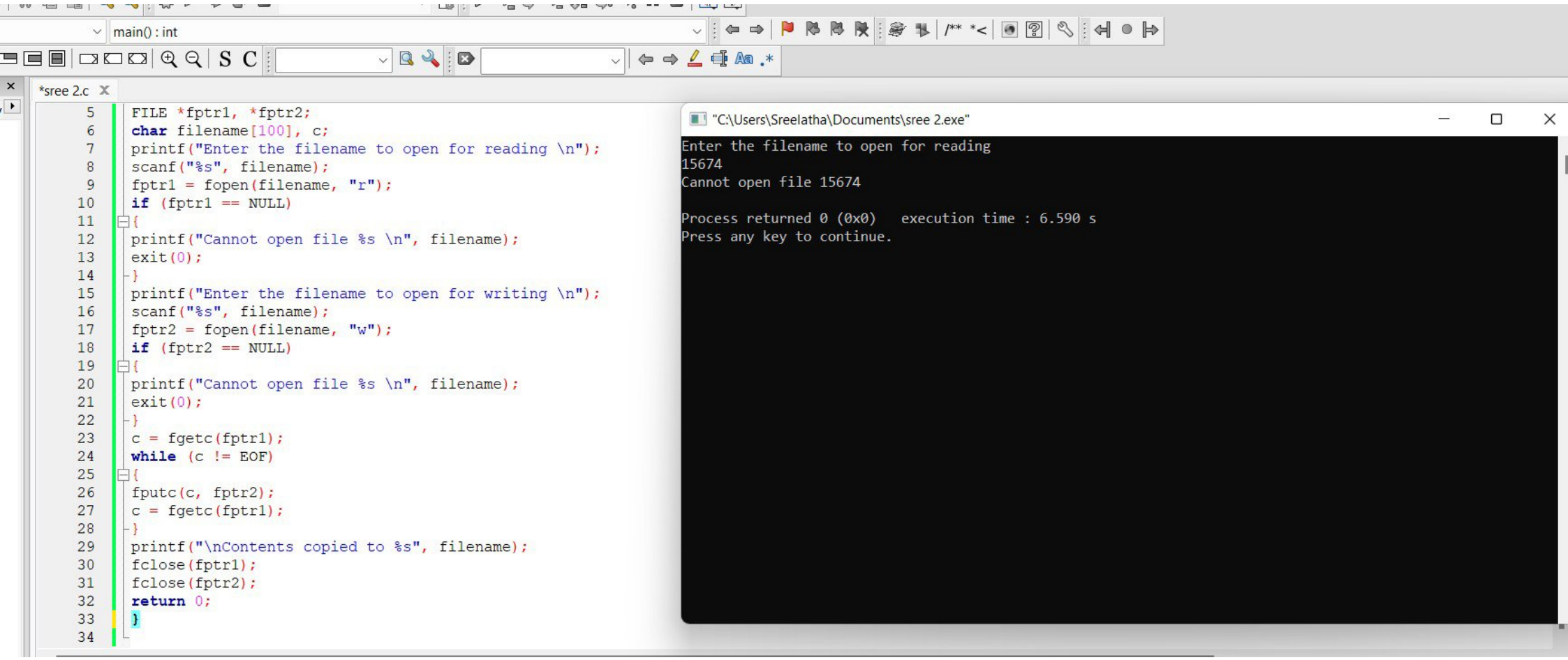
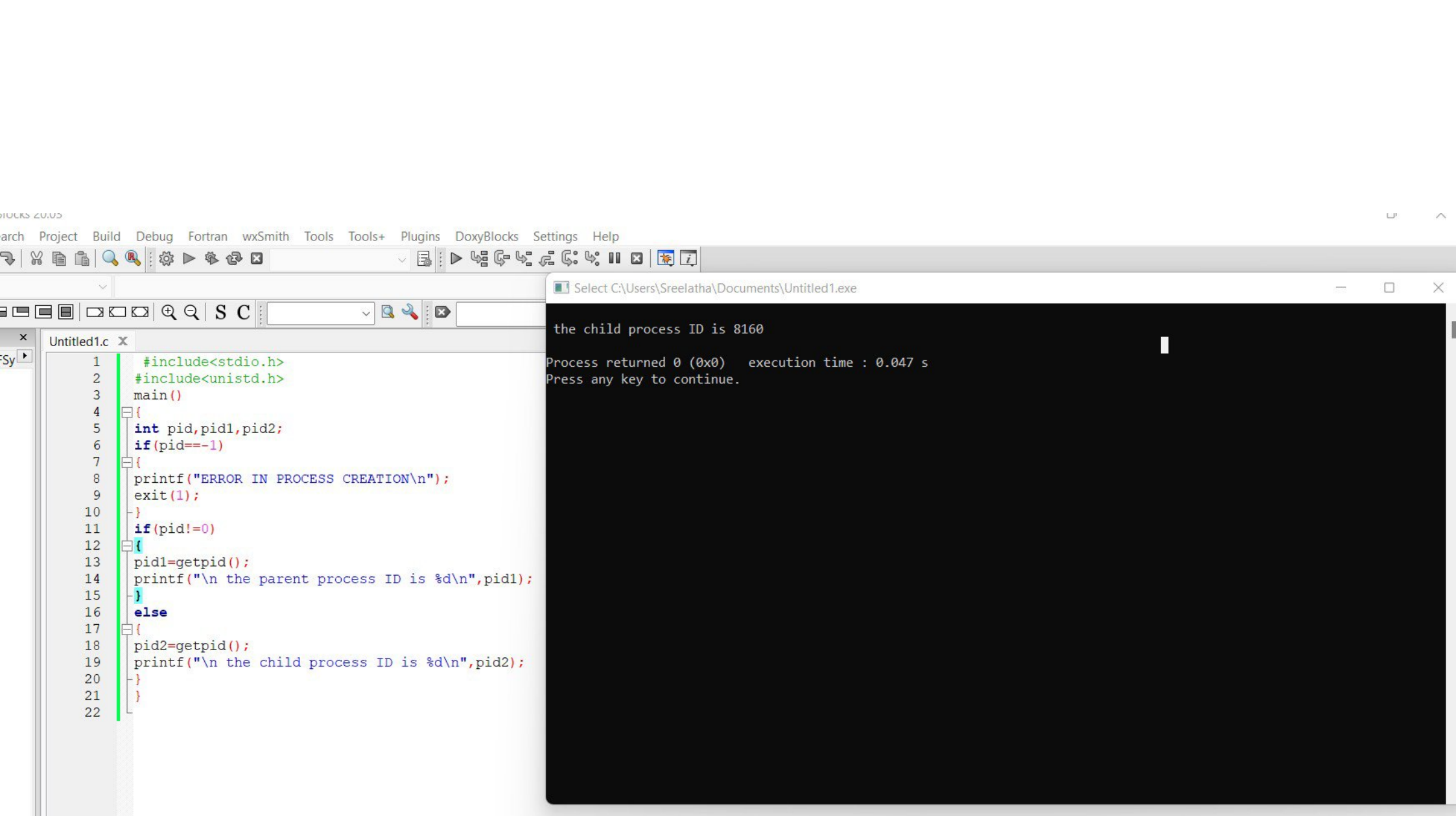
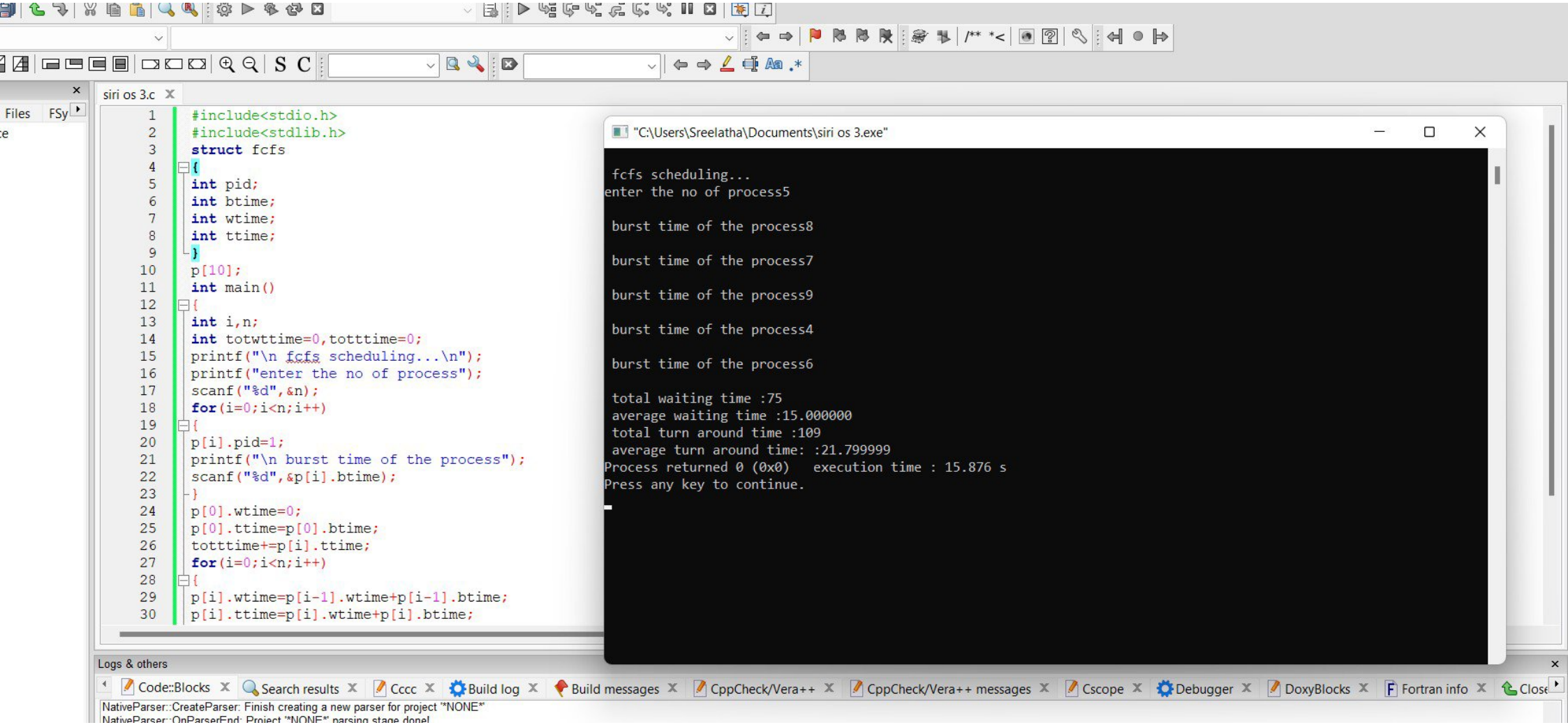
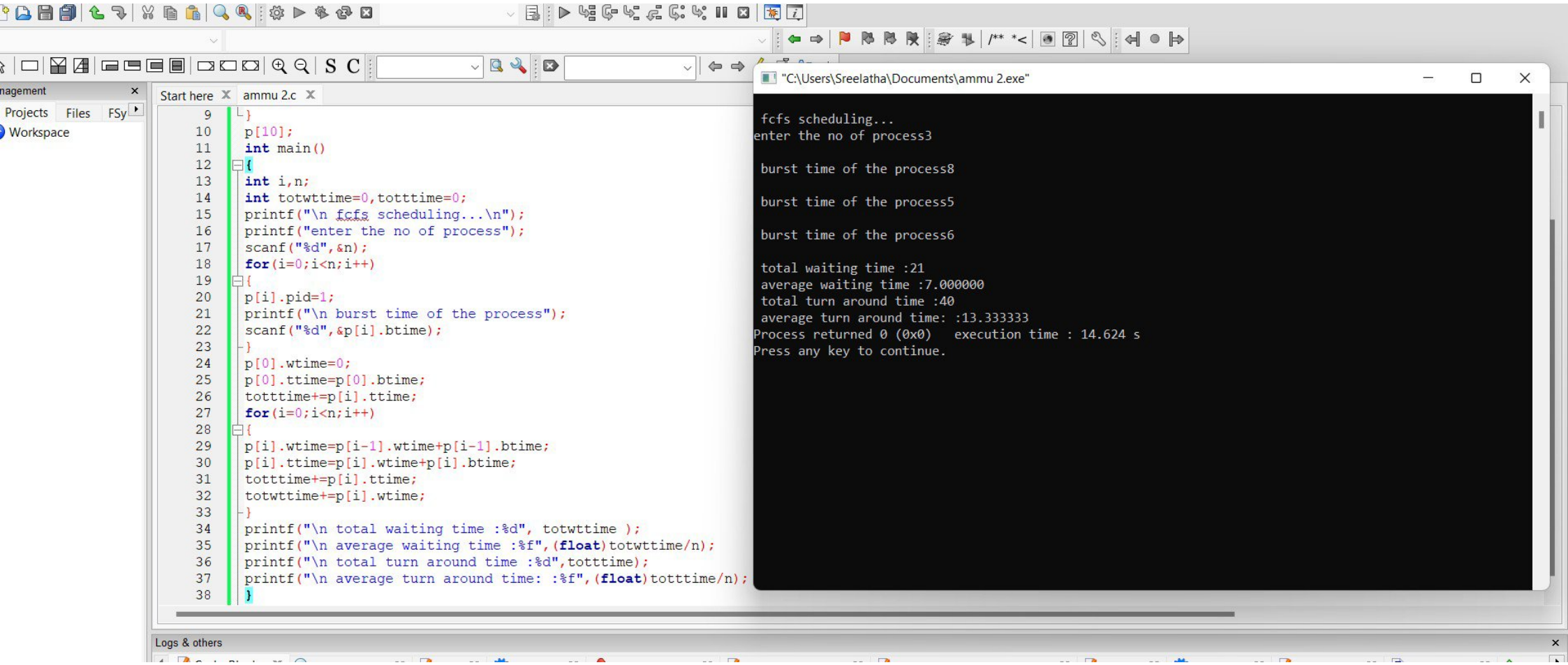
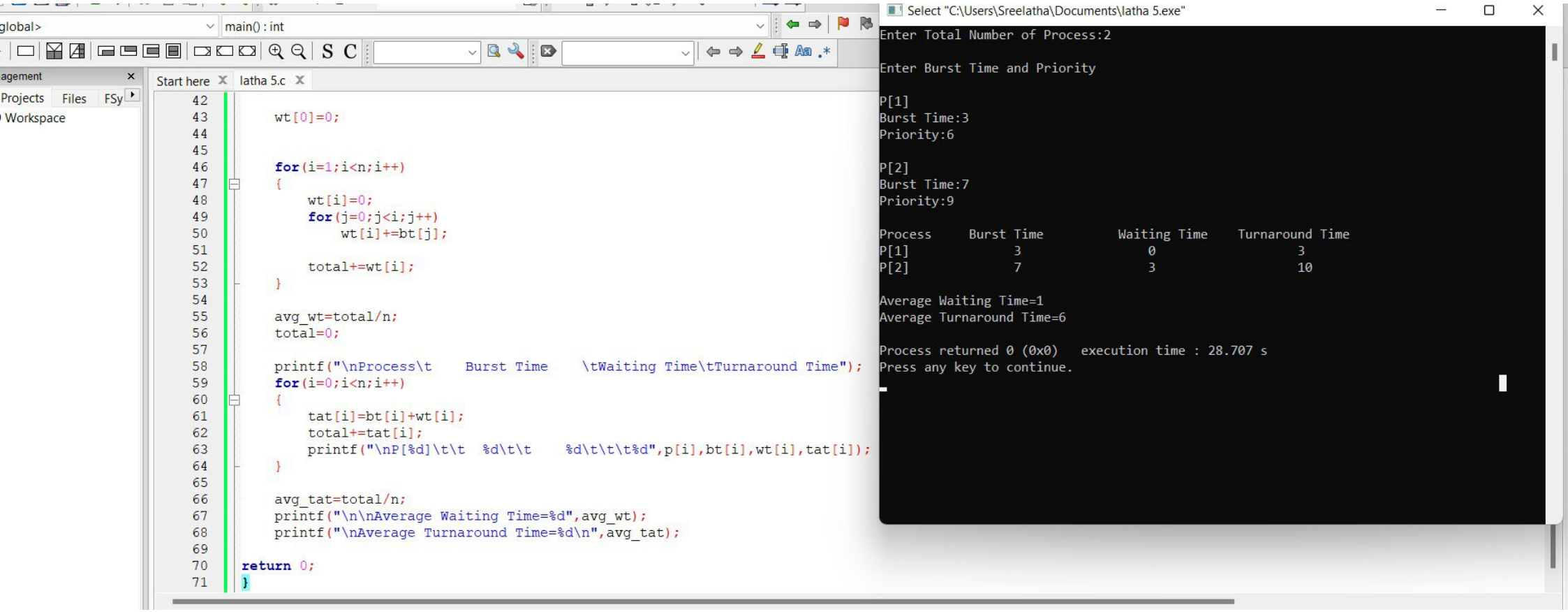

main() : int
*sree 2.c
5 FILE *fptr1, *fptr2;
6 char filename[100], c;
7 printf("Enter the filename to open for reading \n");
8 scanf("%s", filename);
9 fptr1 = fopen(filename, "r");
10 if (fptr1 == NULL)
11 {
12 printf("Cannot open file %s \n", filename);
13 exit(0);
14 }
15 printf("Enter the filename to open for writing \n");
16 scanf("%s", filename);
17 fptr2 = fopen(filename, "w");
18 if (fptr2 == NULL)
19 {
20 printf("Cannot open file %s \n", filename);
21 exit(0);
22 }
23 c = fgetc(fptr1);
24 while (c != EOF)
25 {
26 fputc(c, fptr2);
27 c = fgetc(fptr1);
28 }
29 printf("\nContents copied to %s", filename);
30 fclose(fptr1);
31 fclose(fptr2);
32 return 0;
33 }
34
"C:\Users\Sreelatha\Documents\sree 2.exe"
Enter the filename to open for reading
15674
Cannot open file 15674
Process returned 0 (0x0) execution time : 6.590 s
Press any key to continue.


arch Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Untitled1.c
1 #include<stdio.h>
2 #include<unistd.h>
3 main()
4 {
5 int pid,pid1,pid2;
6 if(pid==-1)
7 {
8 printf("ERROR IN PROCESS CREATION\n");
9 exit(1);
10 }
11 if(pid!=0)
12 {
13 pid1=getpid();
14 printf("\n the parent process ID is %d\n",pid1);
15 }
16 else
17 {
18 pid2=getpid();
19 printf("\n the child process ID is %d\n",pid2);
20 }
21 }
22
Select C:\Users\Sreelatha\Documents\Untitled1.exe
the child process ID is 8160
Process returned 0 (0x0) execution time : 0.047 s
Press any key to continue.


siri os 3.c
#include<stdio.h>
#include<stdlib.h>
struct fcfs
{
int pid;
int btime;
int wtime;
int ttime;
}
p[10];
int main()
{
int i,n;
int totwttime=0,totttime=0;
printf("\n fcfs scheduling...\n");
printf("enter the no of process");
scanf("%d",&n);
for(i=0;i<n;i++)
{
p[i].pid=1;
printf("\n burst time of the process");
scanf("%d",&p[i].btime);
}
p[0].wtime=0;
p[0].ttime=p[0].btime;
totttime+=p[i].ttime;
for(i=0;i<n;i++)
{
p[i].wtime=p[i-1].wtime+p[i-1].btime;
p[i].ttime=p[i].wtime+p[i].btime;
"C:\Users\Sreelatha\Documents\siri os 3.exe"
fcfs scheduling...
enter the no of process5
burst time of the process8
burst time of the process7
burst time of the process9
burst time of the process4
burst time of the process6
total waiting time :75
average waiting time :15.000000
total turn around time :109
average turn around time: :21.799999
Process returned 0 (0x0) execution time : 15.876 s
Press any key to continue.
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
DoxyBlocks
Fortran info
NativeParser::CreateParser: Finish creating a new parser for project *NONE*


Start here
ammu 2.c
}
p[10];
int main()
{
int i,n;
int totwttime=0,totttime=0;
printf("\n fcfs scheduling...\n");
printf("enter the no of process");
scanf("%d",&n);
for(i=0;i<n;i++)
{
p[i].pid=1;
printf("\n burst time of the process");
scanf("%d",&p[i].btime);
}
p[0].wtime=0;
p[0].ttime=p[0].btime;
totttime+=p[i].ttime;
for(i=0;i<n;i++)
{
p[i].wtime=p[i-1].wtime+p[i-1].btime;
p[i].ttime=p[i].wtime+p[i].btime;
totttime+=p[i].ttime;
totwttime+=p[i].wtime;
}
printf("\n total waiting time :%d", totwttime);
printf("\n average waiting time :%f",(float)totwttime/n);
printf("\n total turn around time :%d",totttime);
printf("\n average turn around time: :%f",(float)totttime/n);
}
"C:\Users\Sreelatha\Documents\ammu 2.exe"
fcfs scheduling...
enter the no of process3
burst time of the process8
burst time of the process5
burst time of the process6
total waiting time :21
average waiting time :7.000000
total turn around time :40
average turn around time: :13.333333
Process returned 0 (0x0) execution time : 14.624 s
Press any key to continue.
Logs & others



```
    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=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=total/n;
    printf("\n\nAverage Waiting Time=%d",avg_wt);
    printf("\nAverage Turnaround Time=%d\n",avg_tat);

return 0;
}
```

```
Select "C:\Users\Sreelatha\Documents\latha 5.exe"
Enter Total Number of Process:2

Enter Burst Time and Priority

P[1]
Burst Time:3
Priority:6

P[2]
Burst Time:7
Priority:9

Process     Burst Time          Waiting Time    Turnaround Time
P[1]             3                   0                   3
P[2]             7                   3                   10

Average Waiting Time=1
Average Turnaround Time=6

Process returned 0 (0x0)   execution time : 28.707 s
Press any key to continue.
```