online c compiler - Search | Online C Compiler

https://www.programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Interactive C Course

main.c

Run

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

Output

Clear

```
/tmp/Bj9Lhhxdfa.o
before fork
My child's id is 1727
I am parent having id 1726
Common
I am child having id 1727
My parent's id is 1
Common
```

86°F
Mostly cloudy

ENG
IN

23:04
26-09-2022

online c compiler - Search | Online C Compiler

https://www.programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Interactive C Course

main.c

Run

```
6     printf("Enter the filename to open for reading \n");
7     scanf("%s",filename);
8     // Open one file for reading
9     fptr1 = fopen(filename, "r");
10    if (fptr1 == NULL){
11       printf("Cannot open file %s \n", filename);
12       exit(0);
13    }
14    printf("Enter the filename to open for writing \n");
15    scanf("%s", filename);
16    // Open another file for writing
17    fptr2 = fopen(filename, "w");
18    if (fptr2 == NULL){
19       printf("Cannot open file %s \n", filename);
20       exit(0);
21    }
22    // Read contents from file
23    c = fgetc(fptr1);
24    while (c != EOF){
25       fputc(c, fptr2);
26       c = fgetc(fptr1);
27    }
28    printf("\nContents copied to %s", filename);
29    fclose(fptr1);
30    fclose(fptr2);
31    return 0;
32 }
```

Output

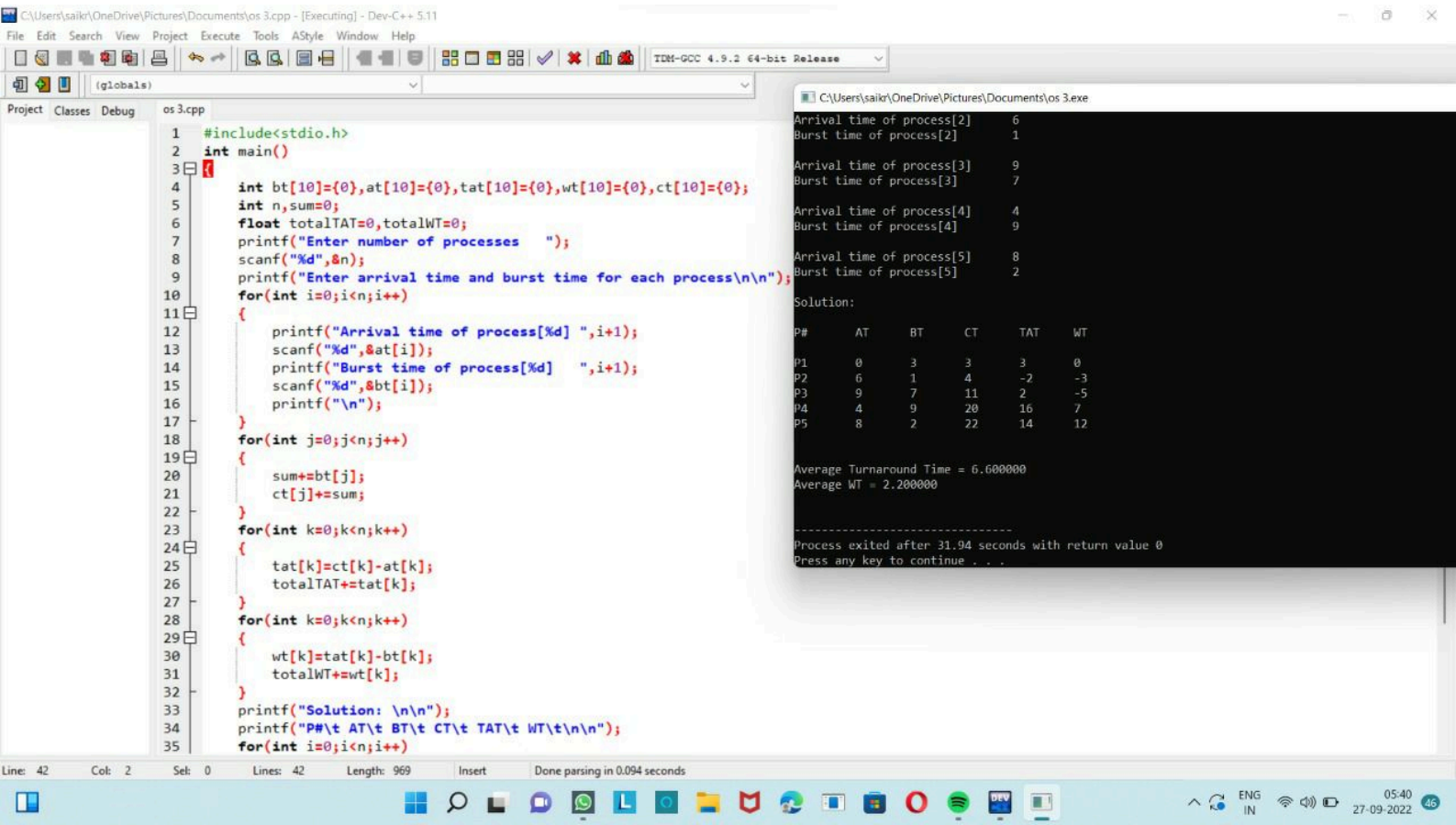
Clear

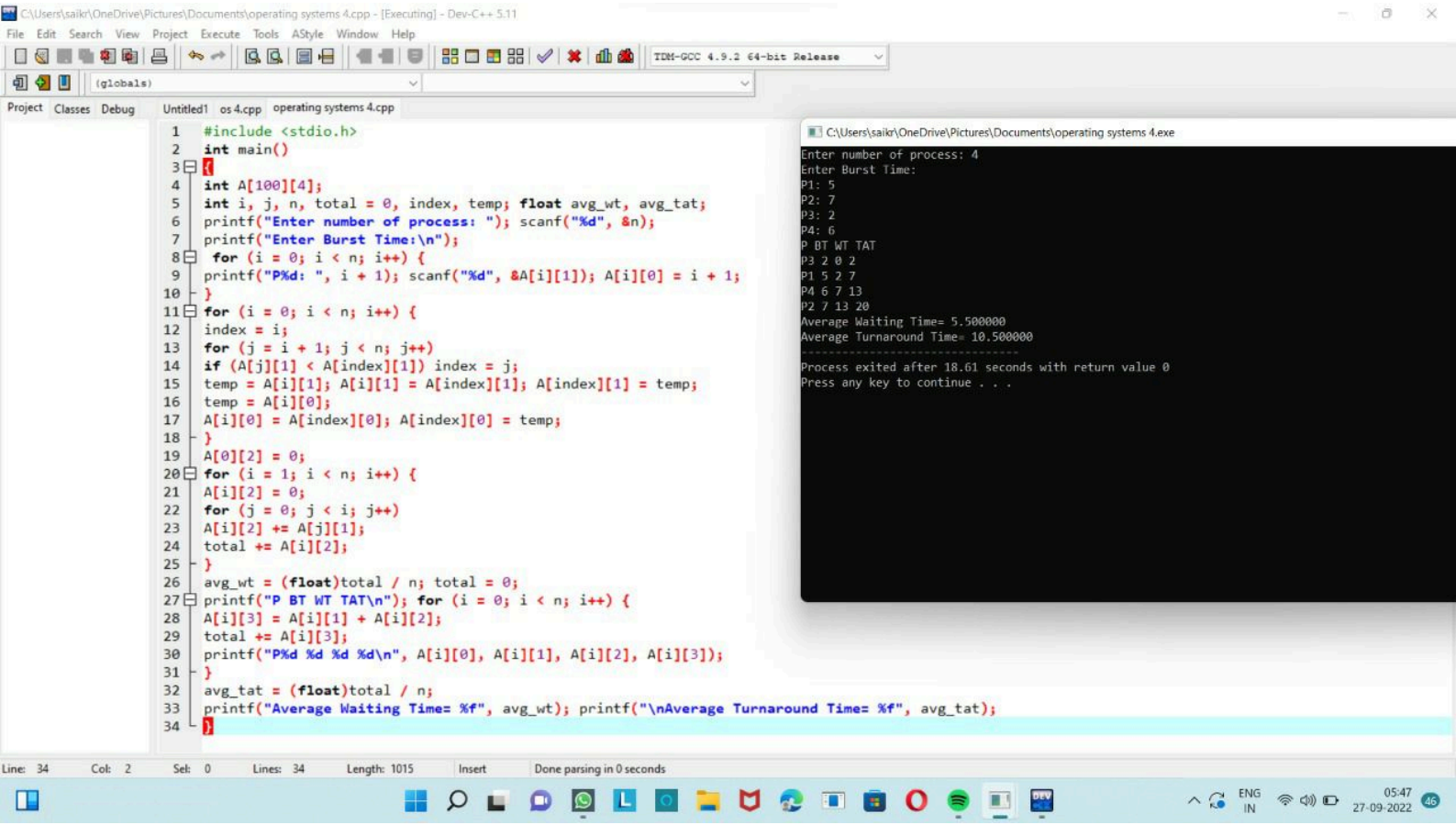
```
/tmp/LqKeOERUVc.o
Enter the filename to open for reading
addition matrix
Cannot open file addition
```

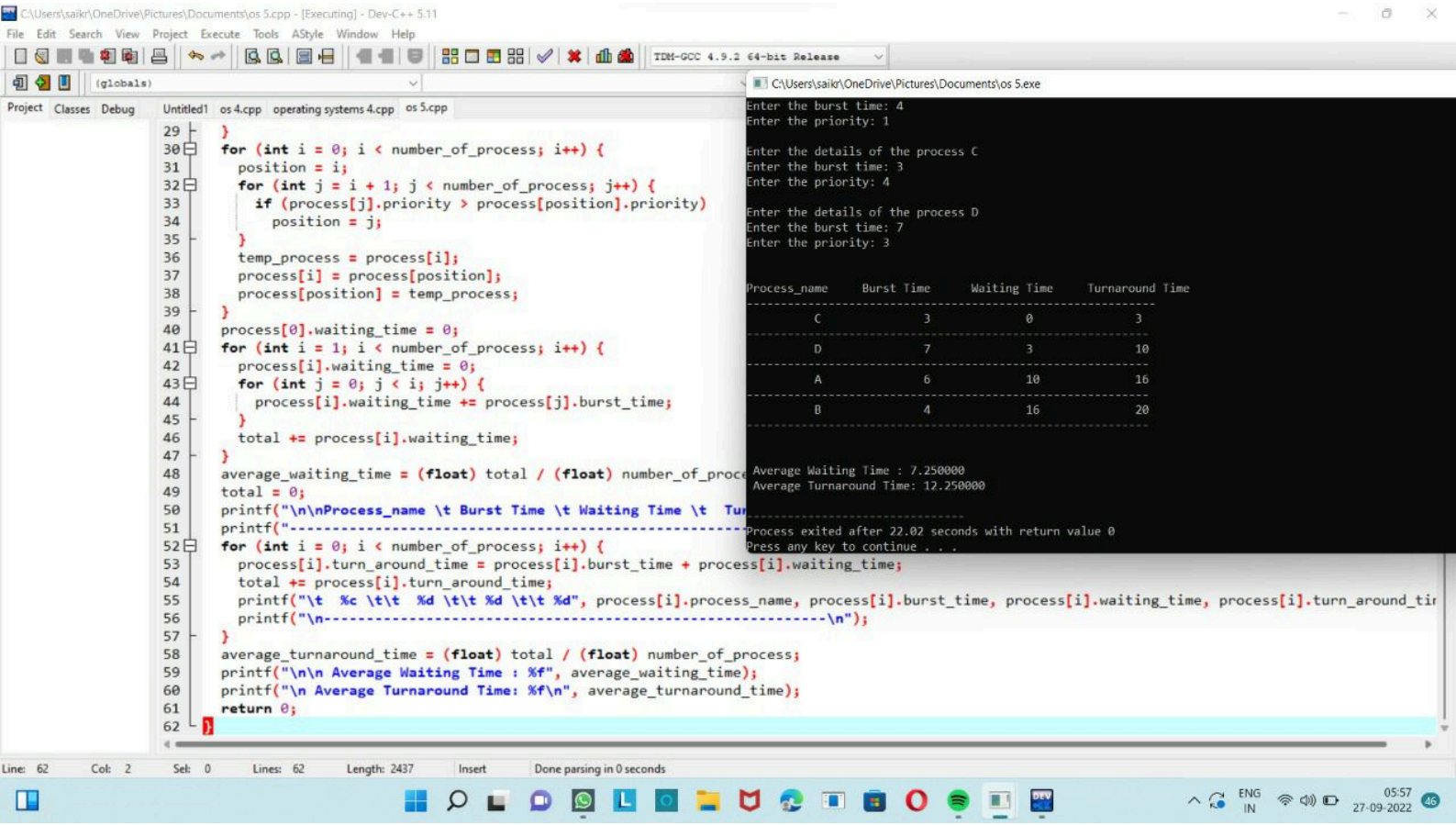
86°F
Mostly cloudy

ENG
IN

23:16
26-09-2022


C:\Users\saikr\OneDrive\Pictures\Documents\os 3.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
os 3.cpp
#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++)
Line: 42 Col: 2 Sel: 0 Lines: 42 Length: 969 Insert Done parsing in 0.094 seconds
C:\Users\saikr\OneDrive\Pictures\Documents\os 3.exe
Arrival time of process[2] 6
Burst time of process[2] 1
Arrival time of process[3] 9
Burst time of process[3] 7
Arrival time of process[4] 4
Burst time of process[4] 9
Arrival time of process[5] 8
Burst time of process[5] 2
Solution:
P# AT BT CT TAT WT
P1 0 3 3 3 0
P2 6 1 4 -2 -3
P3 9 7 11 2 -5
P4 4 9 20 16 7
P5 8 2 22 14 12
Average Turnaround Time = 6.600000
Average WT = 2.200000
Process exited after 31.94 seconds with return value 0
Press any key to continue . . .
ENG IN
05:40 27-09-2022


C:\Users\saikr\OneDrive\Pictures\Documents\operating systems 4.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
Untitled1 os 4.cpp operating systems 4.cpp
#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);
}
C:\Users\saikr\OneDrive\Pictures\Documents\operating systems 4.exe
Enter number of process: 4
Enter Burst Time:
P1: 5
P2: 7
P3: 2
P4: 6
P BT WT TAT
P3 2 0 2
P1 5 2 7
P4 6 7 13
P2 7 13 20
Average Waiting Time= 5.500000
Average Turnaround Time= 10.500000
Process exited after 18.61 seconds with return value 0
Press any key to continue . . .
Line: 34 Col: 2 Sel: 0 Lines: 34 Length: 1015 Insert Done parsing in 0 seconds
ENG IN
05:47
27-09-2022


C:\Users\saikr\OneDrive\Pictures\Documents\os 5.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
Untitled1 os 4.cpp operating systems 4.cpp os 5.cpp
}
for (int i = 0; i < number_of_process; i++) {
position = i;
for (int j = i + 1; j < number_of_process; j++) {
if (process[j].priority > process[position].priority)
position = j;
}
temp_process = process[i];
process[i] = process[position];
process[position] = temp_process;
}
process[0].waiting_time = 0;
for (int i = 1; i < number_of_process; i++) {
process[i].waiting_time = 0;
for (int j = 0; j < i; j++) {
process[i].waiting_time += process[j].burst_time;
}
total += process[i].waiting_time;
}
average_waiting_time = (float) total / (float) number_of_proc
total = 0;
printf("\n\nProcess_name \t Burst Time \t Waiting Time \t Tu
printf("----------
for (int i = 0; i < number_of_process; i++) {
process[i].turn_around_time = process[i].burst_time + process[i].waiting_time;
total += process[i].turn_around_time;
printf("\t %c \t\t %d \t\t %d \t\t %d", process[i].process_name, process[i].burst_time, process[i].waiting_time, process[i].turn_around_tir
printf("\n----------\n");
}
average_turnaround_time = (float) total / (float) number_of_process;
printf("\n\n Average Waiting Time : %f", average_waiting_time);
printf("\n Average Turnaround Time: %f\n", average_turnaround_time);
return 0;
}
Line: 62 Col: 2 Sel: 0 Lines: 62 Length: 2437 Insert Done parsing in 0 seconds
C:\Users\saikr\OneDrive\Pictures\Documents\os 5.exe
Enter the burst time: 4
Enter the priority: 1
Enter the details of the process C
Enter the burst time: 3
Enter the priority: 4
Enter the details of the process D
Enter the burst time: 7
Enter the priority: 3
Process_name Burst Time Waiting Time Turnaround Time
C 3 0 3
D 7 3 10
A 6 10 16
B 4 16 20
Average Waiting Time : 7.250000
Average Turnaround Time: 12.250000
Process exited after 22.02 seconds with return value 0
Press any key to continue . . .
ENG IN
05:57
27-09-2022