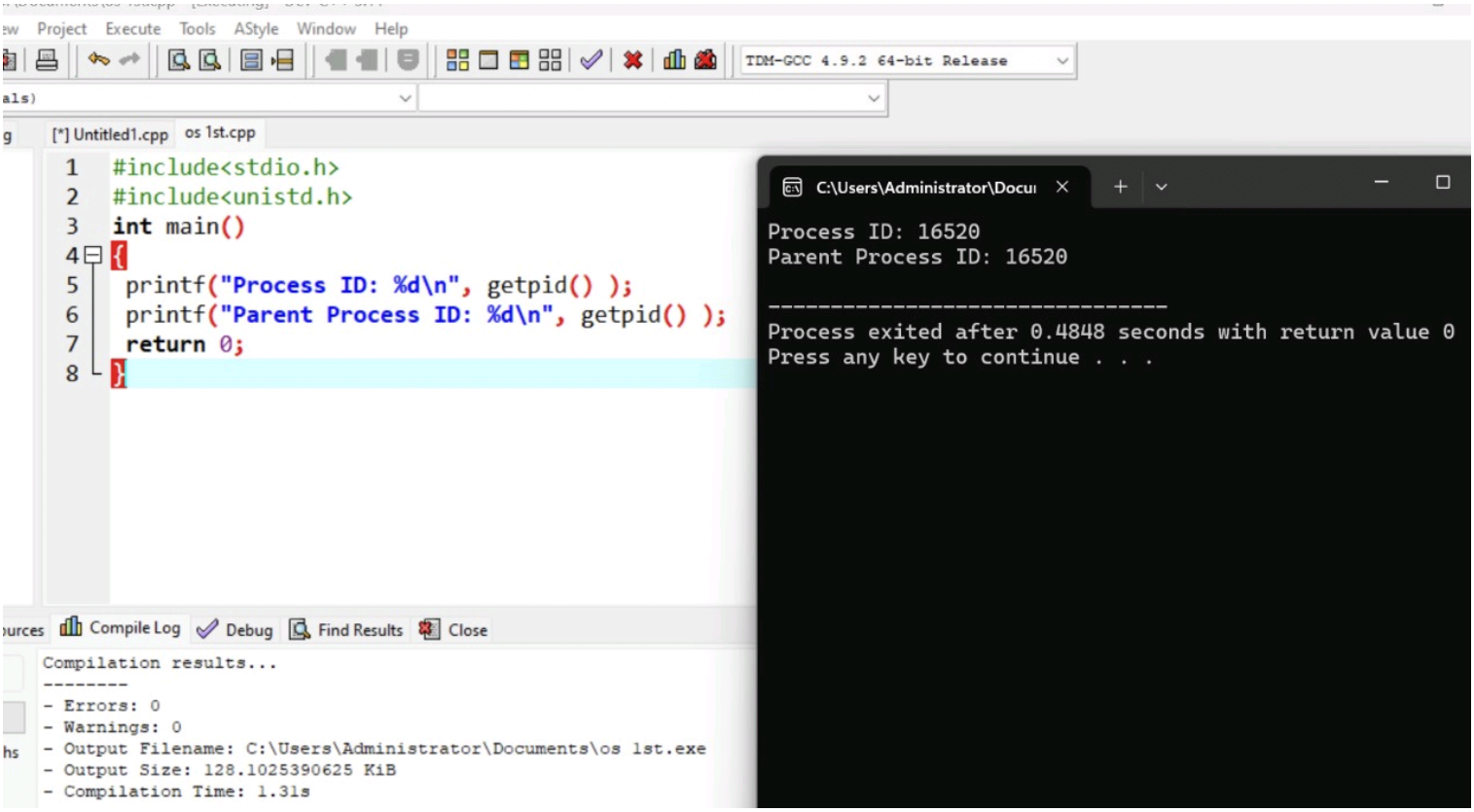
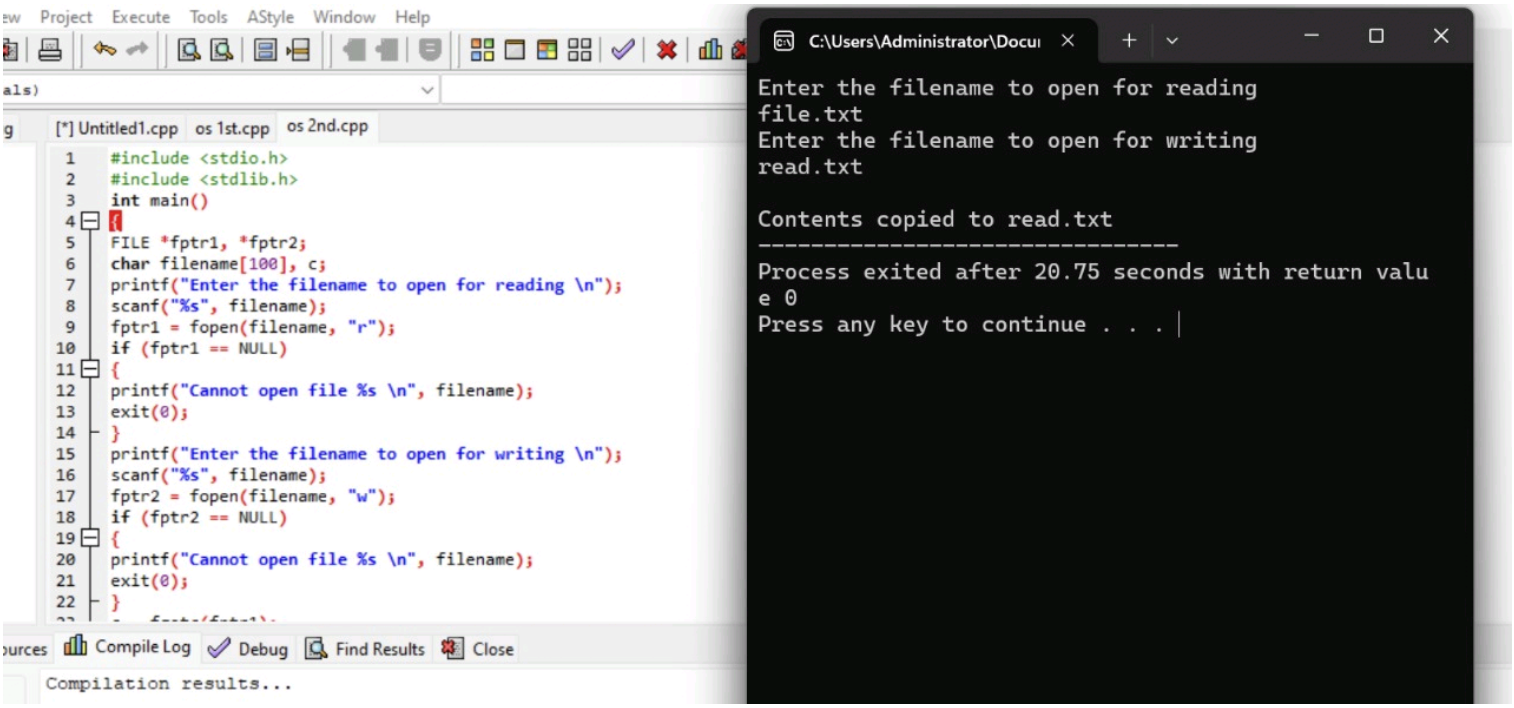
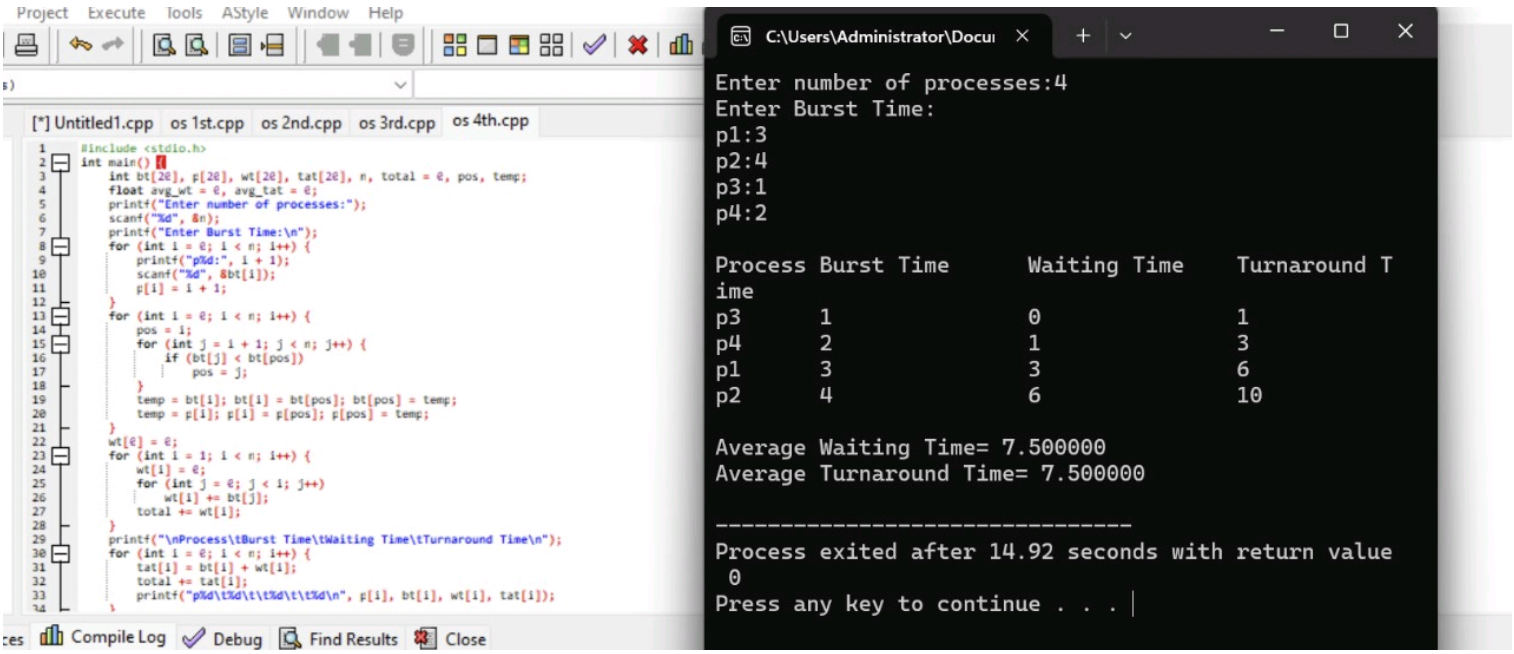
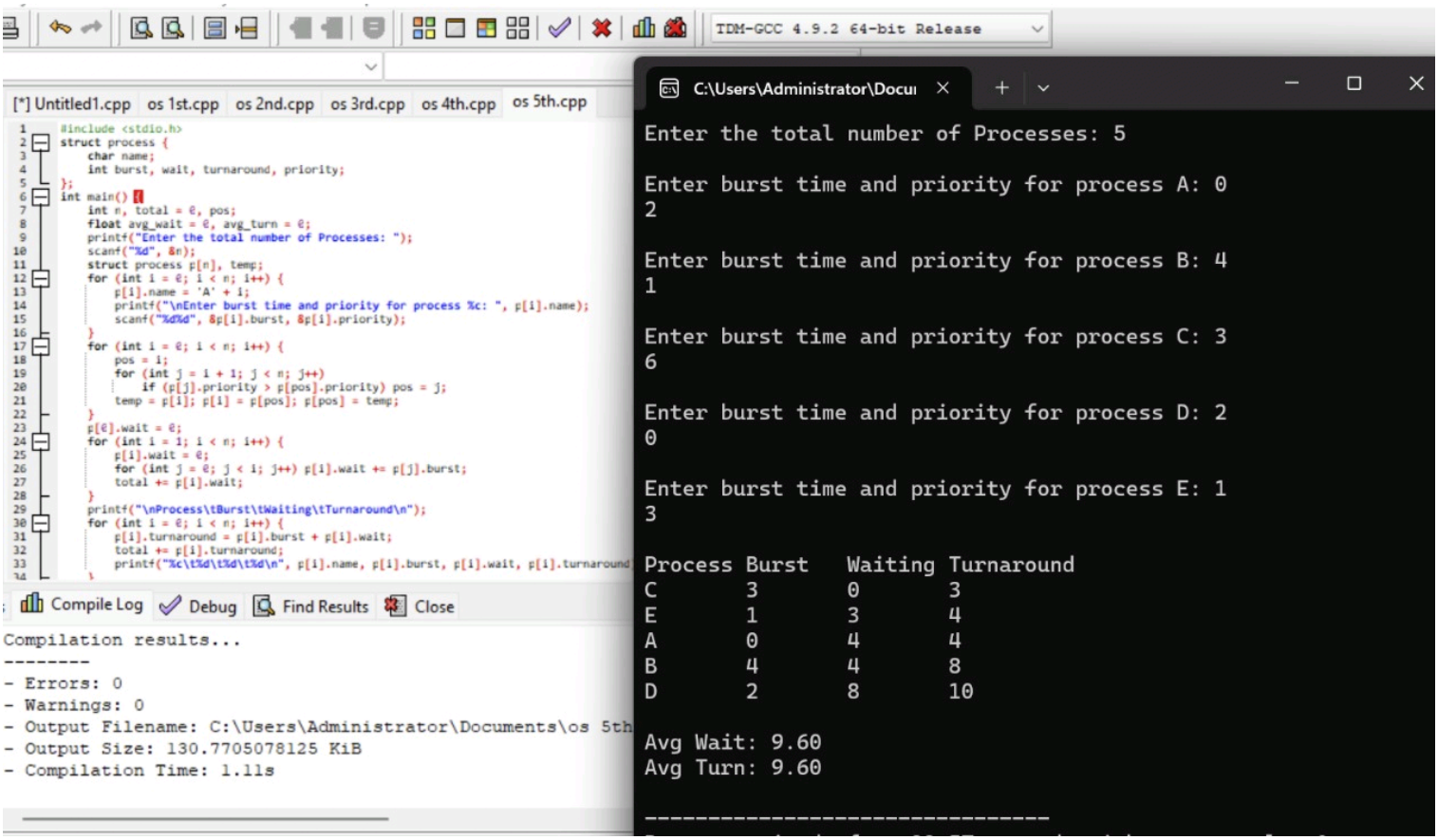
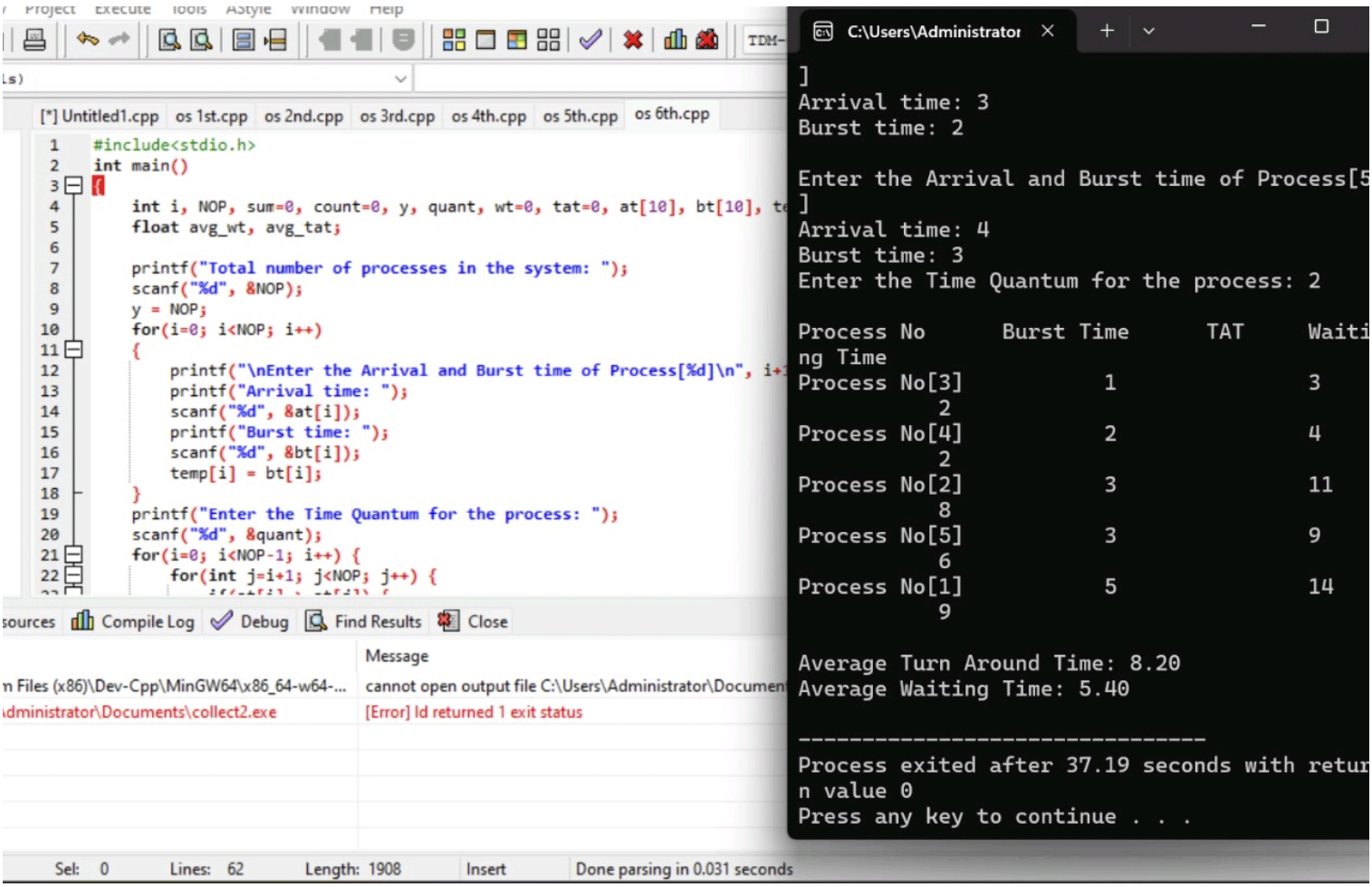

Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
[*] Untitled1.cpp
os 1st.cpp
#include<stdio.h>
#include<unistd.h>
int main()
{
printf("Process ID: %d\n", getpid());
printf("Parent Process ID: %d\n", getpid());
return 0;
}
Compile Log
Debug
Find Results
Close
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Documents\os 1st.exe
- Output Size: 128.1025390625 KiB
- Compilation Time: 1.31s
C:\Users\Administrator\Docu
Process ID: 16520
Parent Process ID: 16520
Process exited after 0.4848 seconds with return value 0
Press any key to continue . . .


Project Execute Tools AStyle Window Help
[*] Untitled1.cpp
os 1st.cpp
os 2nd.cpp
#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);
}
Compile Log
Debug
Find Results
Close
Compilation results...
C:\Users\Administrator\Docu
Enter the filename to open for reading
file.txt
Enter the filename to open for writing
read.txt
Contents copied to read.txt
Process exited after 20.75 seconds with return valu
e 0
Press any key to continue . . .


Project Execute Tools AStyle Window Help
[*] Untitled1.cpp os 1st.cpp os 2nd.cpp os 3rd.cpp os 4th.cpp
#include <stdio.h>
int main() {
int bt[20], p[20], wt[20], tat[20], n, total = 0, pos, temp;
float avg_wt = 0, avg_tat = 0;
printf("Enter number of processes:");
scanf("%d", &n);
printf("Enter Burst Time:\n");
for (int i = 0; i < n; i++) {
printf("p%d:", i + 1);
scanf("%d", &bt[i]);
p[i] = i + 1;
}
for (int i = 0; i < n; i++) {
pos = i;
for (int 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 (int i = 1; i < n; i++) {
wt[i] = 0;
for (int j = 0; j < i; j++)
wt[i] += bt[j];
total += wt[i];
}
printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");
for (int i = 0; i < n; i++) {
tat[i] = bt[i] + wt[i];
total += tat[i];
printf("p%d\t%d\t\t%d\t\t%d\n", p[i], bt[i], wt[i], tat[i]);
}
Compile Log
Debug
Find Results
Close
C:\Users\Administrator\Docu
Enter number of processes:4
Enter Burst Time:
p1:3
p2:4
p3:1
p4:2
Process Burst Time Waiting Time Turnaround T
ime
p3 1 0 1
p4 2 1 3
p1 3 3 6
p2 4 6 10
Average Waiting Time= 7.500000
Average Turnaround Time= 7.500000
Process exited after 14.92 seconds with return value
0
Press any key to continue . . .


TDM-GCC 4.9.2 64-bit Release
[*] Untitled1.cpp
os 1st.cpp
os 2nd.cpp
os 3rd.cpp
os 4th.cpp
os 5th.cpp
#include <stdio.h>
struct process {
char name;
int burst, wait, turnaround, priority;
};
int main() {
int n, total = 0, pos;
float avg_wait = 0, avg_turn = 0;
printf("Enter the total number of Processes: ");
scanf("%d", &n);
struct process p[n], temp;
for (int i = 0; i < n; i++) {
p[i].name = 'A' + i;
printf("\nEnter burst time and priority for process %c: ", p[i].name);
scanf("%d%d", &p[i].burst, &p[i].priority);
}
for (int i = 0; i < n; i++) {
pos = i;
for (int j = i + 1; j < n; j++)
if (p[j].priority > p[pos].priority) pos = j;
temp = p[i]; p[i] = p[pos]; p[pos] = temp;
}
p[0].wait = 0;
for (int i = 1; i < n; i++) {
p[i].wait = 0;
for (int j = 0; j < i; j++) p[i].wait += p[j].burst;
total += p[i].wait;
}
printf("\nProcess\tBurst\tWaiting\tTurnaround\n");
for (int i = 0; i < n; i++) {
p[i].turnaround = p[i].burst + p[i].wait;
total += p[i].turnaround;
printf("%c\t%d\t%d\t%d\n", p[i].name, p[i].burst, p[i].wait, p[i].turnaround
Compile Log
Debug
Find Results
Close
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Documents\os 5th
- Output Size: 130.7705078125 KiB
- Compilation Time: 1.11s
C:\Users\Administrator\Docu
Enter the total number of Processes: 5
Enter burst time and priority for process A: 0
2
Enter burst time and priority for process B: 4
1
Enter burst time and priority for process C: 3
6
Enter burst time and priority for process D: 2
0
Enter burst time and priority for process E: 1
3
Process Burst Waiting Turnaround
C 3 0 3
E 1 3 4
A 0 4 4
B 4 4 8
D 2 8 10
Avg Wait: 9.60
Avg Turn: 9.60



[*] Untitled1.cpp | os 1st.cpp | os 2nd.cpp | os 3rd.cpp | os 4th.cpp | os 5th.cpp | os 6th.cpp

```c
#include<stdio.h>
int main()
{
    int i, NOP, sum=0, count=0, y, quant, wt=0, tat=0, at[10], bt[10], te
    float avg_wt, avg_tat;

    printf("Total number of processes in the system: ");
    scanf("%d", &NOP);
    y = NOP;
    for(i=0; i<NOP; i++)
    {
        printf("\nEnter the Arrival and Burst time of Process[%d]\n", i+
        printf("Arrival time: ");
        scanf("%d", &at[i]);
        printf("Burst time: ");
        scanf("%d", &bt[i]);
        temp[i] = bt[i];
    }
    printf("Enter the Time Quantum for the process: ");
    scanf("%d", &quant);
    for(i=0; i<NOP-1; i++) {
        for(int j=i+1; j<NOP; j++) {
```

Compile Log | Debug | Find Results | Close

Message
n Files (x86)\Dev-Cpp\MinGW64\x86_64-w64-...	cannot open output file C:\Users\Administrator\Document
dministrator\Documents\collect2.exe	[Error] ld returned 1 exit status

Sel: 0 Lines: 62 Length: 1908 Insert Done parsing in 0.031 seconds

C:\Users\Administrator

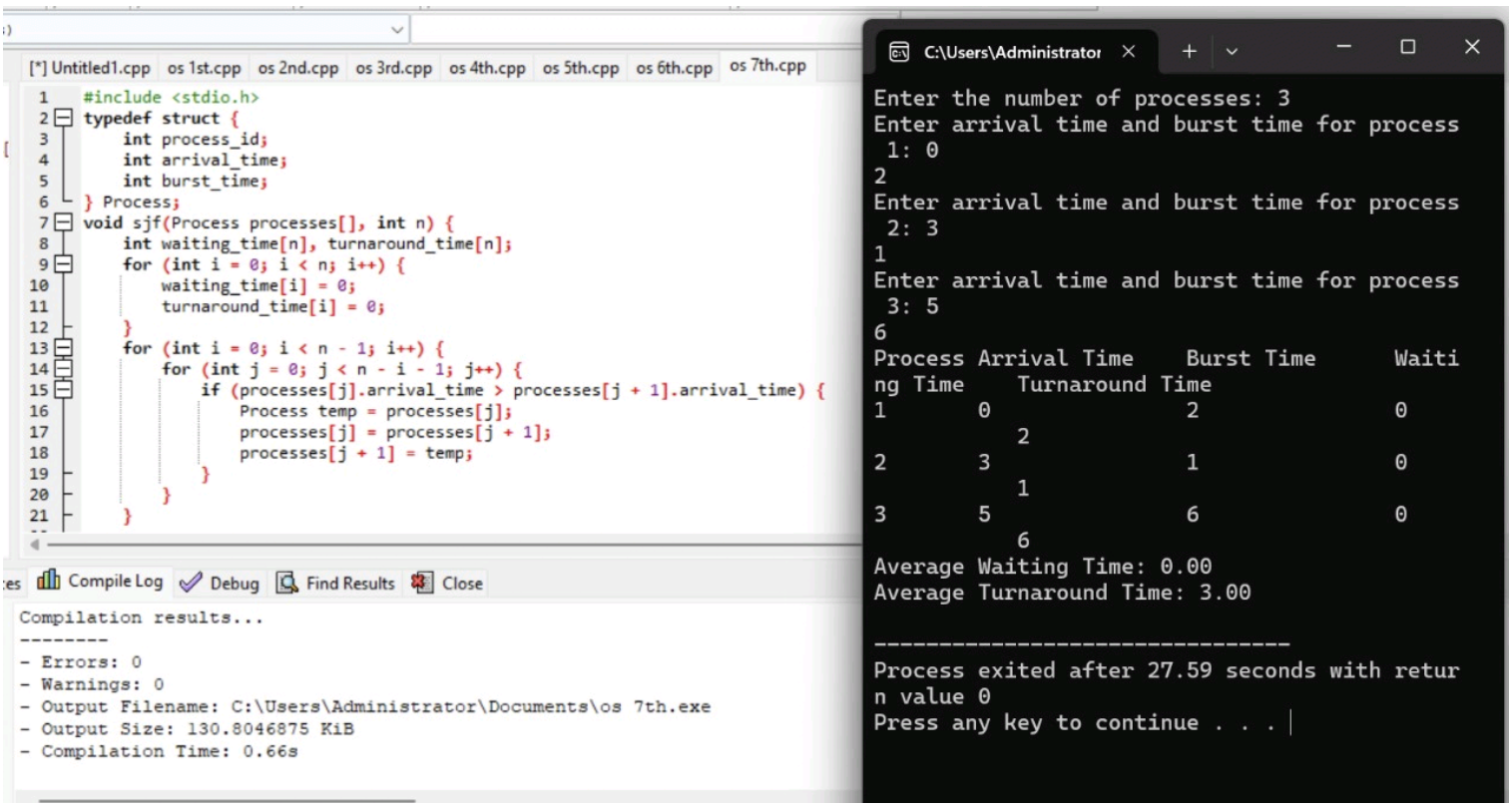
```
]
Arrival time: 3
Burst time: 2

Enter the Arrival and Burst time of Process[5
]
Arrival time: 4
Burst time: 3
Enter the Time Quantum for the process: 2

Process No      Burst Time      TAT     Waiti
ng Time
Process No[3]           1               3
          2
Process No[4]           2               4
          2
Process No[2]           3               11
          8
Process No[5]           3               9
          6
Process No[1]           5               14
          9

Average Turn Around Time: 8.20
Average Waiting Time: 5.40

--------------------------------
Process exited after 37.19 seconds with retur
n value 0
Press any key to continue . . .
```


[*] Untitled1.cpp
os 1st.cpp
os 2nd.cpp
os 3rd.cpp
os 4th.cpp
os 5th.cpp
os 6th.cpp
os 7th.cpp
#include <stdio.h>
typedef struct {
int process_id;
int arrival_time;
int burst_time;
} Process;
void sjf(Process processes[], int n) {
int waiting_time[n], turnaround_time[n];
for (int i = 0; i < n; i++) {
waiting_time[i] = 0;
turnaround_time[i] = 0;
}
for (int i = 0; i < n - 1; i++) {
for (int j = 0; j < n - i - 1; j++) {
if (processes[j].arrival_time > processes[j + 1].arrival_time) {
Process temp = processes[j];
processes[j] = processes[j + 1];
processes[j + 1] = temp;
}
}
}
Compile Log
Debug
Find Results
Close
Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Documents\os 7th.exe
- Output Size: 130.8046875 KiB
- Compilation Time: 0.66s
C:\Users\Administrator
Enter the number of processes: 3
Enter arrival time and burst time for process 1: 0
2
Enter arrival time and burst time for process 2: 3
1
Enter arrival time and burst time for process 3: 5
6
Process Arrival Time Burst Time Waiting Time Turnaround Time
1 0 2 0 2
2 3 1 0 1
3 5 6 0 6
Average Waiting Time: 0.00
Average Turnaround Time: 3.00
Process exited after 27.59 seconds with return value 0
Press any key to continue . . .

[*] Untitled1.cpp | os 1st.cpp | os 2nd.cpp | os 3rd.cpp | os 4th.cpp | os 5th.cpp | os 6th.cpp | os 7th.cpp | os 8th.cpp

```c
#include<stdio.h>
#include<stdlib.h>
#define MAX_PROCESSES 10
struct Process {
    int id;
    int priority;
};
struct Process selectHighestPriority(struct Process processes[], int n) {
    struct Process highestPriorityProcess = processes[0];
    for (int i = 1; i < n; i++) {
        if (processes[i].priority > highestPriorityProcess.priority) {
            highestPriorityProcess = processes[i];
        }
    }
    return highestPriorityProcess;
}
int main() {
    struct Process processes[MAX_PROCESSES];
    int n;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
printf("Enter details of each process:\n");
```

Compile Log | Debug | Find Results | Close

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Documents\os 8th.exe
```

C:\Users\Administrator\Docu

```
Enter the number of processes: 5
Enter details of each process:
Process 1:
Priority: 3
Process 2:
Priority: 4
Process 3:
Priority: 4
Process 4:
Priority: 5
Process 5:
Priority: 2
Process with the highest priority:
ID: 4
Priority: 5

--------------------------------
Process exited after 92.06 seconds with return
 0
Press any key to continue . . .
```