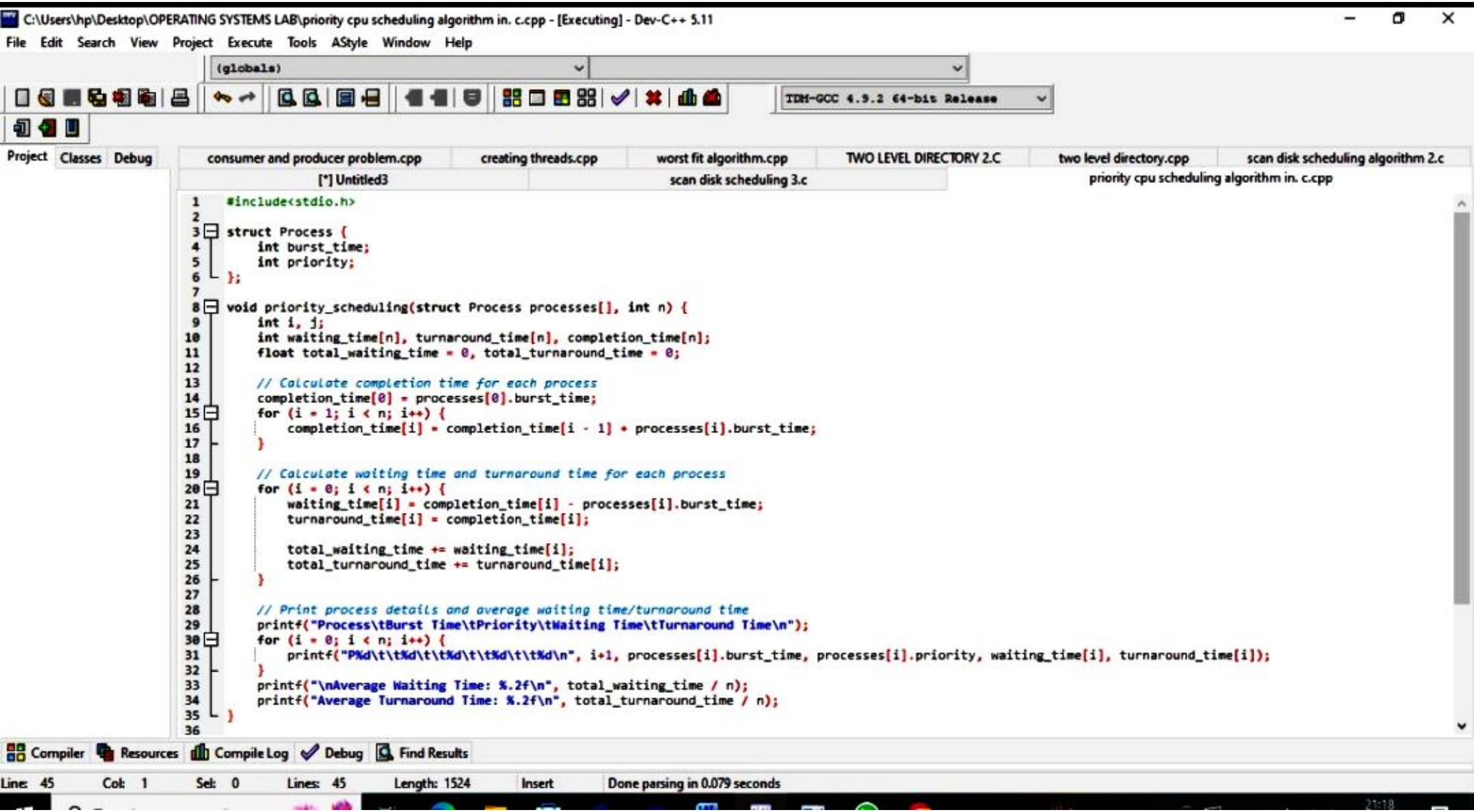
26. Write a program to compute the average waiting time and average turnaround time based on Priority scheduling for the following process with the given CPU burst times (and the assumption that all jobs arrive at the same time.)

| Process | Burst Time | Priority |
|---|---|---|
| P1 | 30 | 2 |
| P2 | 5 | 1 |
| P3 | 123 | |


C:\Users\hp\Desktop\OPERATING SYSTEMS LAB\priority cpu scheduling algorithm in. c.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
TDM-GCC 4.9.2 64-bit Release
Project Classes Debug
consumer and producer problem.cpp
creating threads.cpp
worst fit algorithm.cpp
TWO LEVEL DIRECTORY 2.C
two level directory.cpp
scan disk scheduling algorithm 2.c
[*] Untitled3
scan disk scheduling 3.c
priority cpu scheduling algorithm in. c.cpp
#include<stdio.h>
struct Process {
int burst_time;
int priority;
};
void priority_scheduling(struct Process processes[], int n) {
int i, j;
int waiting_time[n], turnaround_time[n], completion_time[n];
float total_waiting_time = 0, total_turnaround_time = 0;
// Calculate completion time for each process
completion_time[0] = processes[0].burst_time;
for (i = 1; i < n; i++) {
completion_time[i] = completion_time[i - 1] + processes[i].burst_time;
}
// Calculate waiting time and turnaround time for each process
for (i = 0; i < n; i++) {
waiting_time[i] = completion_time[i] - processes[i].burst_time;
turnaround_time[i] = completion_time[i];
total_waiting_time += waiting_time[i];
total_turnaround_time += turnaround_time[i];
}
// Print process details and average waiting time/turnaround time
printf("Process\tBurst Time\tPriority\tWaiting Time\tTurnaround Time\n");
for (i = 0; i < n; i++) {
printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i+1, processes[i].burst_time, processes[i].priority, waiting_time[i], turnaround_time[i]);
}
printf("\nAverage Waiting Time: %.2f\n", total_waiting_time / n);
printf("Average Turnaround Time: %.2f\n", total_turnaround_time / n);
}
Compiler Resources Compile Log Debug Find Results
Line: 45 Col: 1 Sel: 0 Lines: 45 Length: 1524 Insert Done parsing in 0.079 seconds

C:\Users\hp\Desktop\OPERATING SYSTEMS LAB\priority cpu scheduling algorithm in. c.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

consumer and producer problem.cpp | creating threads.cpp | worst fit algorithm.cpp | TWO LEVEL DIRECTORY 2.C | two level directory.cpp | scan disk scheduling algorithm 2.c

[*] Untitled3 | scan disk scheduling 3.c | priority cpu scheduling algorithm in. c.cpp

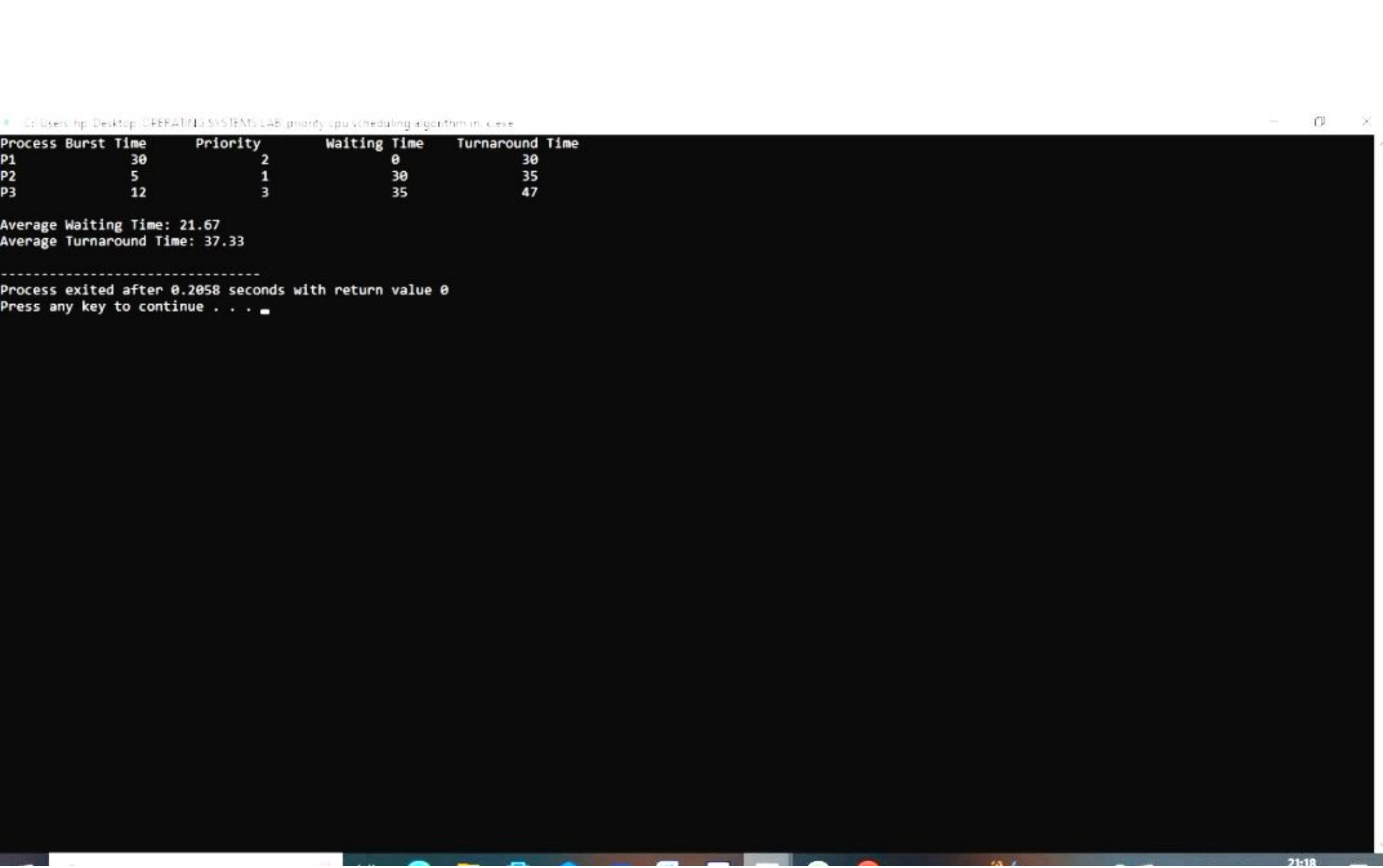
```
11      float total_waiting_time = 0, total_turnaround_time = 0;
12
13      // Calculate completion time for each process
14      completion_time[0] = processes[0].burst_time;
15      for (i = 1; i < n; i++) {
16          completion_time[i] = completion_time[i - 1] + processes[i].burst_time;
17      }
18
19      // Calculate waiting time and turnaround time for each process
20      for (i = 0; i < n; i++) {
21          waiting_time[i] = completion_time[i] - processes[i].burst_time;
22          turnaround_time[i] = completion_time[i];
23
24          total_waiting_time += waiting_time[i];
25          total_turnaround_time += turnaround_time[i];
26      }
27
28      // Print process details and average waiting time/turnaround time
29      printf("Process\tBurst Time\tPriority\tWaiting Time\tTurnaround Time\n");
30      for (i = 0; i < n; i++) {
31          printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i+1, processes[i].burst_time, processes[i].priority, waiting_time[i], turnaround_time[i]);
32      }
33      printf("\nAverage Waiting Time: %.2f\n", total_waiting_time / n);
34      printf("Average Turnaround Time: %.2f\n", total_turnaround_time / n);
35  }
36
37  int main() {
38      struct Process processes[] = {{30, 2}, {5, 1}, {12, 3}};
39      int n = sizeof(processes) / sizeof(processes[0]);
40
41      priority_scheduling(processes, n);
42
43      return 0;
44  }
45
```

Compiler | Resources | Compile Log | Debug | Find Results

Line: 45 Col: 1 Sel: 0 Lines: 45 Length: 1524 Insert Done parsing in 0.079 seconds

Type here to search

Match ENG 21:18 15-05-2023


C:\Users\hp\Desktop\OPERATING SYSTEMS LAB\priority cpu scheduling algorithm in c.exe
Process Burst Time Priority Waiting Time Turnaround Time
P1 30 2 0 30
P2 5 1 30 35
P3 12 3 35 47
Average Waiting Time: 21.67
Average Turnaround Time: 37.33
Process exited after 0.2058 seconds with return value 0
Press any key to continue . . .