

RUN AND DEBUG

RUN

Run and Debug

To customize Run and Debug create a launch.json file.

Show all automatic debug configurations.

To learn more about launch.json, see Configuring C/C++ debugging.

C PARENTID.c

C PARENTID.c > main()

```
1 #include<stdio.h>
2 #include<unistd.h>
3 int main()
4 {
5     printf("Process ID: %d\n", getpid() );
6     printf("Parent Process ID: %d\n", getpid() );
7     return 0;
8 }
9
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

SQL CONSOLE

Filter (e.g. text, exclude)

Loaded '/usr/lib/libobjc.A.dylib'. Symbols loaded.

Loaded '/usr/lib/libc++.dylib'. Symbols loaded.

Loaded '/usr/lib/libc++abi.dylib'. Symbols loaded.

=thread-selected,id="1"

Process ID: 1748

Parent Process ID: 1748

The program '/Users/tharishkumar/Desktop/CURRENT SUB FILES/OPERATING SYSTEM/PARENTID' has exited with code 0 (0x00000000).

Please start a debug session to evaluate expressions

BREAKPOINTS

0 0 0

0 0

Connect

Java: Ready

Ln 6, Col 46

Spaces: 4

UTF-8

() C

Go Live

Mac

EXPLORER

OPEN EDITORS

OPERATING SYSTEM

OUTLINE

TIMELINE

C PARNENTID.c

C FILE2.c

C FILE.c

C CPUSCHEDULING.c

C SMALLESTEXCEUTION.c

C PARNENTID.dSYM

C CPUSCHEDULING.dSYM

C FILE.dSYM

C FILE2.dSYM

C PARNENTID.dSYM

C QUESTION

C SMALLESTEXCEUTION.dSYM

C CPUSCHEDULING

C CPUSCHEDULING.c

C FILE

C FILE.c

C FILE2

C FILE2.c

C PARNENTID

C PARNENTID.c

C SMALLESTEXCEUTION

C SMALLESTEXCEUTION.c

C PARNENTID.c

C FILE2.c

C FILE.c

C CPUSCHEDULING.c

C SMALLESTEXCEUTION.c

C SMALLESTEXCEUTION.c

1 #include<stdio.h>

2

3 struct Process {

4 int id;

5 int arrival\_time;

6 int burst\_time;

7 int waiting\_time;

8 int turnaround\_time;

9 int completed;

10 };

11

12 void calculateTimes(struct Process processes[], int n) {

13 int total\_waiting\_time = 0;

14 int total\_turnaround\_time = 0;

15

16 processes[0].waiting\_time = 0;

17 processes[0].turnaround\_time = processes[0].burst\_time;

18 processes[0].completed = 1;

19

20 for (int i = 1; i < n; i++) {

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

SQL CONSOLE

cd "/Users/tharishkumar/Desktop/CURRENT SUB FILES/OPERATING SYSTEM/" && gcc SMALLESTEXCEUTION.c -o S

tharishkumar@tharishs-MacBook-Air OPERATING SYSTEM % cd "/Users/tharishkumar/Desktop/CURRENT SUB FILES/OPERATING SYSTEM/" && gcc SMALLESTEXCEUTION.c -o SMALLESTEXCEUTION && "/Users/tharishkumar/Desktop/CURRENT SUB FILES/OPERATING SYSTEM/"SMALLESTEXCEUTION

Enter the number of processes: 3

Enter arrival time and burst time for each process:

Process 1: 5

4

Process 2: 5

6

Process 3: 8

3

Process Arrival Time Burst Time Waiting Time Turnaround Time

1 5 4 0 4

2 5 6 15883216 5

3 8 3 2 5

Average Waiting Time: 5294406.00

Average Turnaround Time: 5294409.00

tharishkumar@tharishs-MacBook-Air OPERATING SYSTEM %

Ln 71, Col 1

Spaces: 4

UTF-8

LF

() C

Go Live

Mac

Visual Studio Code interface showing a C program for file operations.

**EXPLORER**

- OPEN EDITORS
  - PARNENTID.c
  - FILE2.c
  - FILE.c
  - CPUSHEDULING.c
  - SMALLESTEXCEUTION.c
- OPERATING SYSTEM
  - .vscode
  - CPUSHEDULING.dSYM
  - FILE.dSYM
  - FILE2.dSYM
  - PARNENTID.dSYM
  - QUESTION
  - SMALLESTEXCEUTION.dSYM
  - CPUSHEDULING
  - CPUSHEDULING.c
  - FILE
  - FILE.c
  - FILE2
  - FILE2.c
  - PARNENTID
  - PARNENTID.c
  - SMALLESTEXCEUTION
  - SMALLESTEXCEUTION.c

**FILE2.c**

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 void trimNewline(char* string) {
5     size_t length = strlen(string);
6     if (length > 0 && string[length - 1] == '\n') {
7         string[length - 1] = '\0';
8     }
9 }
10 int main() {
11     FILE *fptr1, *fptr2;
12     char readPath[1000], writePath[1000], readFilename[100], writeFilename[100];
13     char readFullPath[1100], writeFullPath[1100];
14     // For reading
15     printf("Enter the full directory path to open for reading (e.g., C:\\Users\\YourUsername\\D
16     fgets(readPath, sizeof(readPath), stdin);
17     trimNewline(readPath); // Remove the newline character
18     printf("Enter the filename to open for reading:\n");
19     fgets(readFilename, sizeof(readFilename), stdin);
20     trimNewline(readFilename); // Remove the newline character
21     snprintf(readFullPath, sizeof(readFullPath), "%s\\%s", readPath, readFilename);
22     fptr1 = fopen(readFullPath, "r");
23     if (fptr1 == NULL) {
24         printf("Cannot open file %s\n", readFullPath);
25         exit(0);
26     }
27     // For writing
28     printf("Enter the full directory path to open for writing (e.g., C:\\Users\\YourUsername\\D
29     fgets(writePath, sizeof(writePath), stdin);
```

**TERMINAL**

```
Enter the full directory path to open for reading (e.g., C:\Users\YourUsername\Desktop):
all files
Enter the filename to open for reading:
certificate
```

Ln 51, Col 1 Spaces: 4 UTF-8 LF C Go Live Mac



EXPLORER

OPEN EDITORS

OPERATING SYSTEM

OUTLINE

TIMELINE

JAVA PROJECTS

PARNENTID.c

FILE.c

CPUSHEDULING.c

SMALLESTEXCEUTION.c

HIGHESTPRIORITY.C

PARNENTID

PARNENTID.c

SMALLESTEXCEUTION

SMALLESTEXCEUTION.c

OPERATING SYSTEM

HIGHESTPRIORITY.C > main()

```
37 temp_process = processes[i];
38 processes[i] = processes[position];
39 processes[position] = temp_process;
40 }
41 processes[0].waiting_time = 0;
42 // Calculate waiting time for each process
43 for (int i = 1; i < number_of_processes; i++) {
44     processes[i].waiting_time = 0;
45     for (int j = 0; j < i; j++) {
46         processes[i].waiting_time += processes[j].burst_time;
47     }
48     total_waiting_time += processes[i].waiting_time;
49 }
50 average_waiting_time = (float) total_waiting_time / (float) number_of_processes;
51 // Calculate turnaround time for each process and display process details
52 printf("\n\nProcess_name\tBurst Time\tWaiting Time\tTurnaround Time\n");
53 printf("-----\n");
54 int total_turnaround_time = 0;
55 for (int i = 0; i < number_of_processes; i++) {
56     processes[i].turn_around_time = processes[i].burst_time + processes[i].waiting_time;
57     total_turnaround_time += processes[i].turn_around_time;
58     printf("\t%c\t\t%d\t\t%d\t\t%d\n", processes[i].process_name, processes[i].burst_time, proc
59     printf("-----\n");
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

SQL CONSOLE

Process_name	Burst Time	Waiting Time	Turnaround Time
B	6	0	6
A	4	6	10
E	7	10	17
C	3	17	20
D	6	20	26

Average Waiting Time: 10.60  
Average Turnaround Time: 15.80

Ln 61, Col 49 (86 selected) Spaces: 4 UTF-8 LF C++ Go Live Mac













