

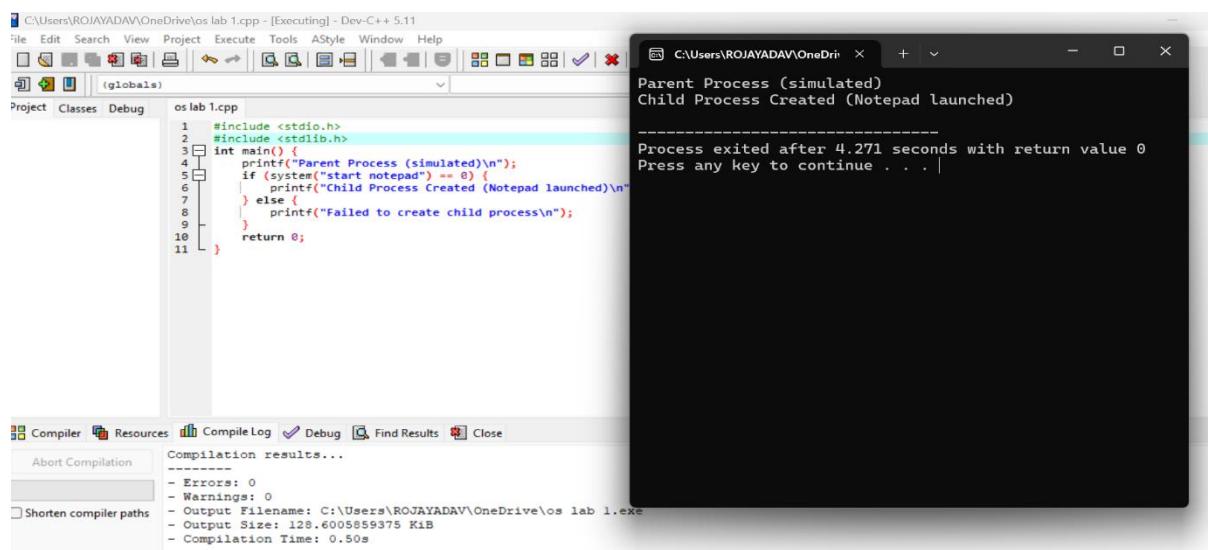
Create a new process by invoking the appropriate system call. Get the process identifier of the currently running process and its respective parent using system calls and display the same using a C program.

Program:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    printf("Parent Process (simulated)\n");
    if (system("start notepad") == 0) {
        printf("Child Process Created (Notepad launched)\n");
    } else {
        printf("Failed to create child process\n");
    }
    return 0;
}
```

Output:



The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays the file 'oslab 1.cpp' with the provided C code. On the right, the terminal window shows the execution results. The terminal output is as follows:

```
C:\Users\ROJAYADAV\OneDrive\os lab 1.cpp - [Executing] - Dev-C++ 5.11
Parent Process (simulated)
Child Process Created (Notepad launched)

Process exited after 4.271 seconds with return value 0
Press any key to continue . . .
```

Below the terminal, the compiler log window shows the compilation results:

```
Compiler Resources Compile Log Debug Find Results Close
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\ROJAYADAV\OneDrive\os lab 1.exe
- Output Size: 128.6005859375 Kib
- Compilation Time: 0.50s
```