




# COSC 3360 - 24967 - Fundamentals of Operating Systems


[Dashboard](#) / [My courses](#) / [COSC3360SP2023-01](#) / [EXAM 1](#) / [Programming Question 1](#)

 Description

 [Submission view](#)

 **Available from:** Thursday, 16 February 2023, 2:30 PM

 **Due date:** Thursday, 16 February 2023, 3:50 PM

 **Requested files:** main.cpp ( [Download](#))

**Type of work:**  Individual work

**The exam will close at 3:50 pm. You must save your work before 3:50 PM.**

Complete the C++ program below to generate the following process tree:

```
501 2962 2957 0 1:25PM ttys002 0:00.00 mainP
501 2963 2962 0 1:25PM ttys002 0:00.00 mainP
501 2964 2962 0 1:25PM ttys002 0:00.00 mainP
501 2965 2962 0 1:25PM ttys002 0:00.00 mainP
501 2966 2963 0 1:25PM ttys002 0:00.00 mainP
501 2967 2964 0 1:25PM ttys002 0:00.00 mainP
501 2968 2963 0 1:25PM ttys002 0:00.00 mainP
501 2969 2965 0 1:25PM ttys002 0:00.00 mainP
501 2970 2964 0 1:25PM ttys002 0:00.00 mainP
501 2971 2963 0 1:25PM ttys002 0:00.00 mainP
```

## Notes:

1. mainP is the name of the process.
2. The PID of the parent process is 2962.
3. Use wait and \_exit when needed to guarantee the following output:

```
I am the parent process
I am child process 0
I am a grandchild process from child process 0
I am a grandchild process from child process 0
I am a grandchild process from child process 0
I am child process 1
I am a grandchild process from child process 1
I am a grandchild process from child process 1
I am child process 2
I am a grandchild process from child process 2
```

## Requested files

main.cpp

```
1  #include <iostream>
2  #include <unistd.h>
3  #include <sys/types.h>
4  #include <sys/wait.h>
5  #include <cmath>
6
7
8  int main()
9  {
10 // std::cout << "I am the parent process" << std::endl;
11 // std::cout << "I am child process " << /*Variable Identifier*/ << std::endl;
12 // std::cout << "I am a grandchild process from child process " << /*Variable Identifier*/ << std::endl;
13     return 0;
14 }
15
16 /*
17      FORK EXAMPLE CODE WE WROTE DURING THE PREVIOUS LECTURE
18      YOU CAN USE THIS CODE TO WRITE YOUR SOLUTION FOR THIS QUESTION
19
20 #include <iostream>
21 #include <unistd.h>
22 #include <sys/wait.h>
23
24 int main()
25 {
26     int pid;
27     std::cout << "I am the parent process " << std::endl;
28     for (int i = 0; i < 3; i++)
29     {
30         pid = fork();
31         if (pid == 0)
32         {
33             std::cout << "I am child process " << i << std::endl;
34             if (i == 1)
35             {
36                 pid = fork();
37                 if (pid == 0)
38                 {
39                     std::cout << "I am a grandchild process from child process " << i << std::endl;
40                     _exit(0);
41                 }
42                 wait(nullptr);
43             }
44             _exit(0);
45         }
46         wait(nullptr);
47     }
48     // for (int i = 0; i < 3; i++)
49     //     wait(nullptr);
50     return 0;
51 }
52
53
54
55 */
```

[VPL](#)

You are logged in as Dosbol Aliev (Log out)

COSC3360SP2023-01

[Data retention summary](#)[Get the mobile app](#)