COSC 3360 - 24967 - Fundamentals of Operating Systems

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

```cpp
#include <iostream>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <cmath>


int main()
{
//  std::cout << "I am the parent process" << std::endl;
//  std::cout << "I am child process " << /*Variable Identifier*/ << std::endl;
//  std::cout << "I am a grandchild process from child process " << /*Variable Identifier*/ << std::endl;
    return 0;
}

/*
            FORK EXAMPLE CODE WE WROTE DURING THE PREVIOUS LECTURE
        YOU CAN USE THIS CODE TO WRITE YOUR SOLUTION FOR THIS QUESTION

#include <iostream>
#include <unistd.h>
#include <sys/wait.h>

int main()
{
  int pid;
  std::cout << "I am the parent process " << std::endl;
  for (int i = 0; i < 3; i++)
  {
    pid = fork();
    if (pid == 0)
    {
      std::cout << "I am child process " << i << std::endl;
      if (i == 1)
      {

        pid = fork();
        if (pid == 0)
        {
          std::cout << "I am a grandchild process from child process " << i << std::endl;
          _exit(0);
        }
        wait(nullptr);
      }
      _exit(0);
    }
    wait(nullptr);
  }
  //  for (int i = 0; i < 3; i++)
  //      wait(nullptr);
  return 0;
}


*/
```

VPL