

实验二 进程控制

一、 实验名称

进程控制

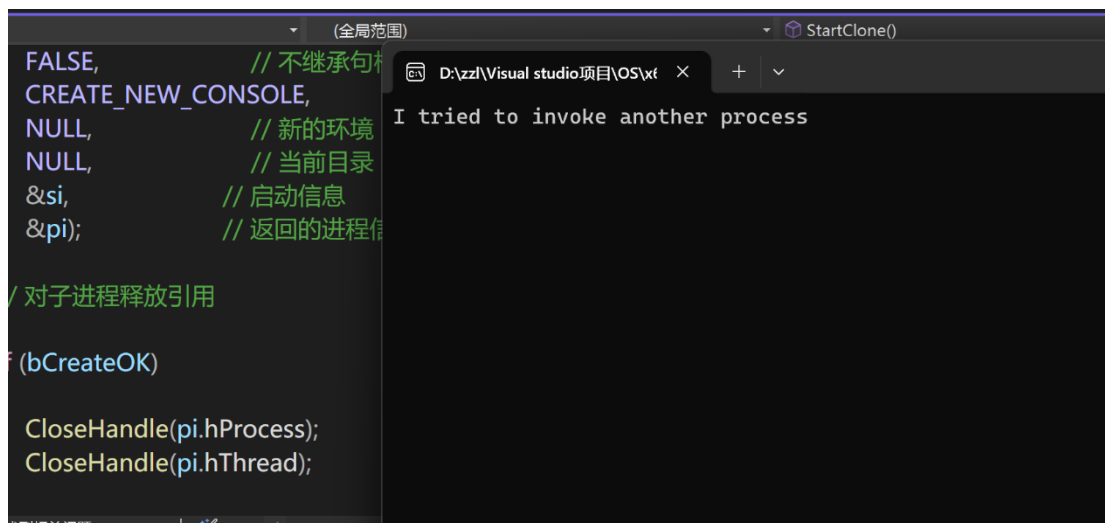
二、 实验目的

1. 掌握进程在操作系统中的作用，了解进程创建、终止、状态切换等基本操作。
2. 通过实际操作，学习与进程控制相关的系统调用，理解系统调用如何与操作系统内核交互。

三、 实验步骤

代码 2-1

结果：



```
FALSE,           // 不继承句柄
CREATE_NEW_CONSOLE,
NULL,            // 新的环境
NULL,            // 当前目录
&si,             // 启动信息
&pi);            // 返回的进程信息

// 对子进程释放引用

if (bCreateOK)
{
    CloseHandle(pi.hProcess);
    CloseHandle(pi.hThread);
}
```

I tried to invoke another process

代码 2-2

结果：



代码 2-3

结果

运行初始结果:

```
// procterm项目
#include <windows.h>
#include <stdio.h>
#include <conio.h>
#include <iostream>
using namespace std;

static LPCTSTR MutexName = "计算机 22级";

D:\zzl\Dev_C++\my_file\2-3.e Now, I'm the parent, begin...
```

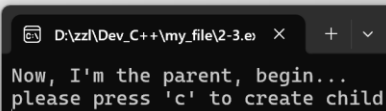
在父进程中如果输入的不是‘c’，会显示:

```

if (hMutexSuicide != NULL)
{
    // 创建子进程
    cout << "Now, I'm the parent, begin..." << endl;
    // 暂停
    // Sleep(2000);
    while (getch() != 'c')
        cout << "please press 'c' to create child" << endl;
    StartClone();

    // 指令子进程自己终止
    cout << "Telling the child process to quit. " << endl;
}

```



```

D:\zzl\Dev_C++\my_file\2-3.e
Now, I'm the parent, begin...
please press 'c' to create child

```

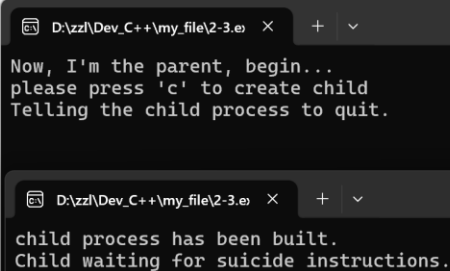
在父进程的窗口中输入‘c’会生成一个子进程：

```

// 创建子进程
cout << "Now, I'm the parent, begin..." << endl;
// 暂停
// Sleep(2000);
while (getch() != 'c')
    cout << "please press 'c' to create child" << endl;
StartClone();

// 指令子进程自己终止
cout << "Telling the child process to quit. " << endl;
}

```



```

D:\zzl\Dev_C++\my_file\2-3.e
Now, I'm the parent, begin...
please press 'c' to create child
Telling the child process to quit.

D:\zzl\Dev_C++\my_file\2-3.e
child process has been built.
Child waiting for suicide instructions.

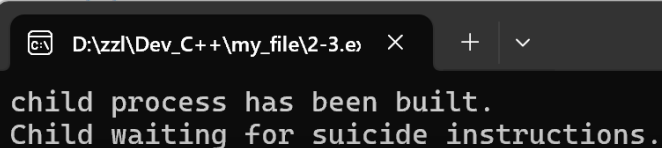
```

如果在父进程中再次输入的字符不是‘r’的话会显示：

```

// 指令子进程自己终止

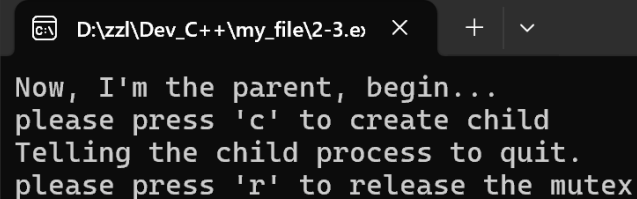
```



```

D:\zzl\Dev_C++\my_file\2-3.e
child process has been built.
Child waiting for suicide instructions.

```

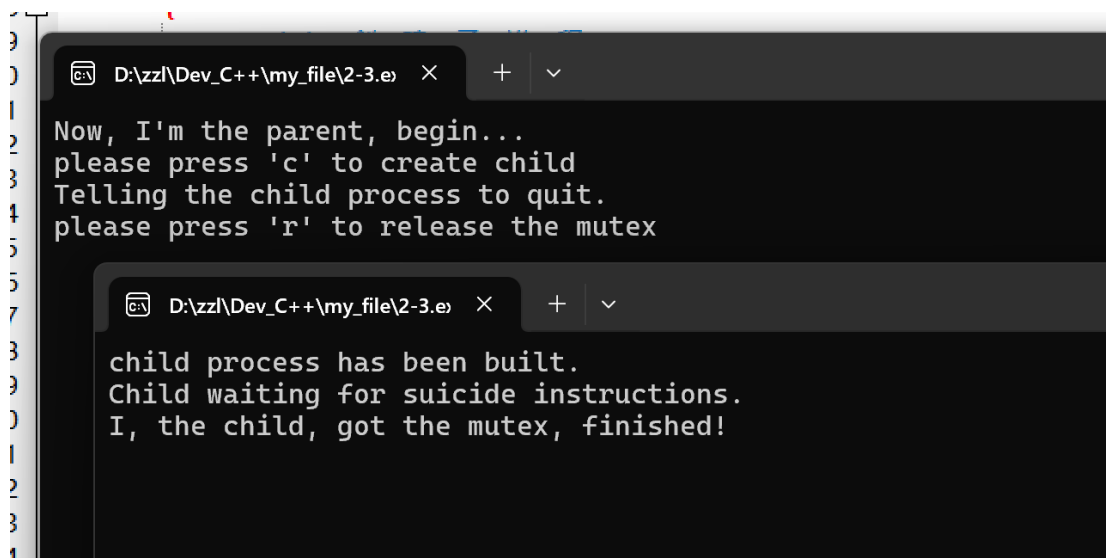


```

D:\zzl\Dev_C++\my_file\2-3.e
Now, I'm the parent, begin...
please press 'c' to create child
Telling the child process to quit.
} please press 'r' to release the mutex

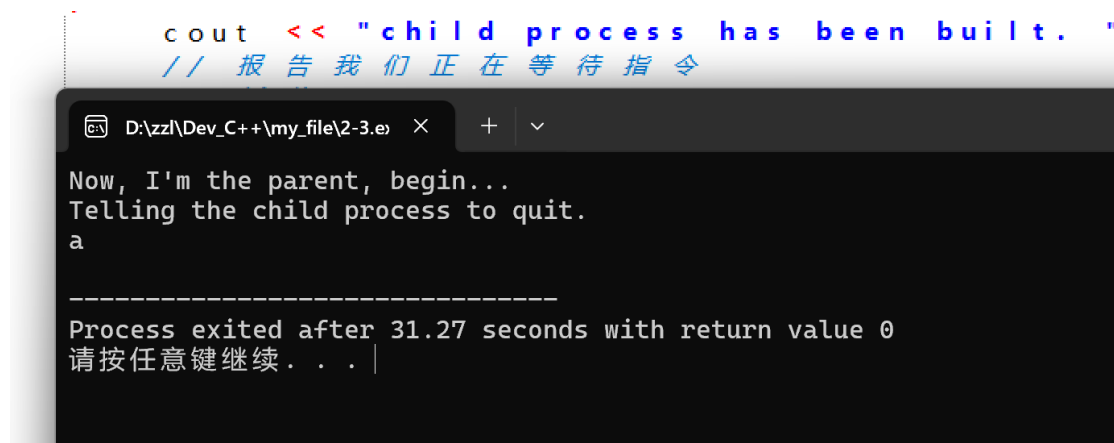
```

在父进程中输入字符‘r’会将子进程中止，此时在子进程中输入一个字符，子进程的窗口便会被关闭。



The image shows two overlapping terminal windows. The top window, titled 'D:\zzl\Dev_C++\my_file\2-3.e', displays the following text: 'Now, I'm the parent, begin...', 'please press 'c' to create child', 'Telling the child process to quit.', and 'please press 'r' to release the mutex'. The bottom window, also titled 'D:\zzl\Dev_C++\my_file\2-3.e', displays: 'child process has been built.', 'Child waiting for suicide instructions.', and 'I, the child, got the mutex, finished!'.

之后在父进程的窗口中输入一个字符，父进程也会中止执行：



The image shows a terminal window titled 'D:\zzl\Dev_C++\my_file\2-3.e'. At the top, there is a line of C++ code: 'cout << "child process has been built. "' followed by a comment '// 报告我们正在等待指令'. Below the code, the terminal displays: 'Now, I'm the parent, begin...', 'Telling the child process to quit.', and a character 'a'. After a series of dashes, it shows 'Process exited after 31.27 seconds with return value 0' and '请按任意键继续. . . |'.

四、 实验小结

通过此次实验，深入理解了操作系统中进程的基本概念及其管理机制。通过编写和运行进程控制程序，掌握了系统调用的具体用法，并了解了父子进程间的关系与进程间通信的实现方式。此外，还通过信号量等机制，理解了并发环境下进程同步的重要性，解决了多个进程争夺共享资源的问题。

通过本次实验，对进程的生命周期及其状态转换有了更加清晰的认识，并能通过实际操作去验证操作系统的进程管理原理，为后续学习和实际开发提供了扎实的理论与实践基础。

