实验一 进程创建与终止

姓名：张枭　　班级：计科1903　　学号：201906062629

一、实验目的

利用Windows提供的API函数，编写程序，实现进程的创建和终止（如创建写字板进程及终止该进程），加深对操作系统进程概念的理解，观察操作系统进程运行的动态性能，获得包含多进程的应用程序编程经验。

二、实验内容与步骤

1. 进程的创建和终止。编写一段程序，可以创建一个进程，并终止当前创建的进程。试观察记录程序执行的结果，并分析原因。

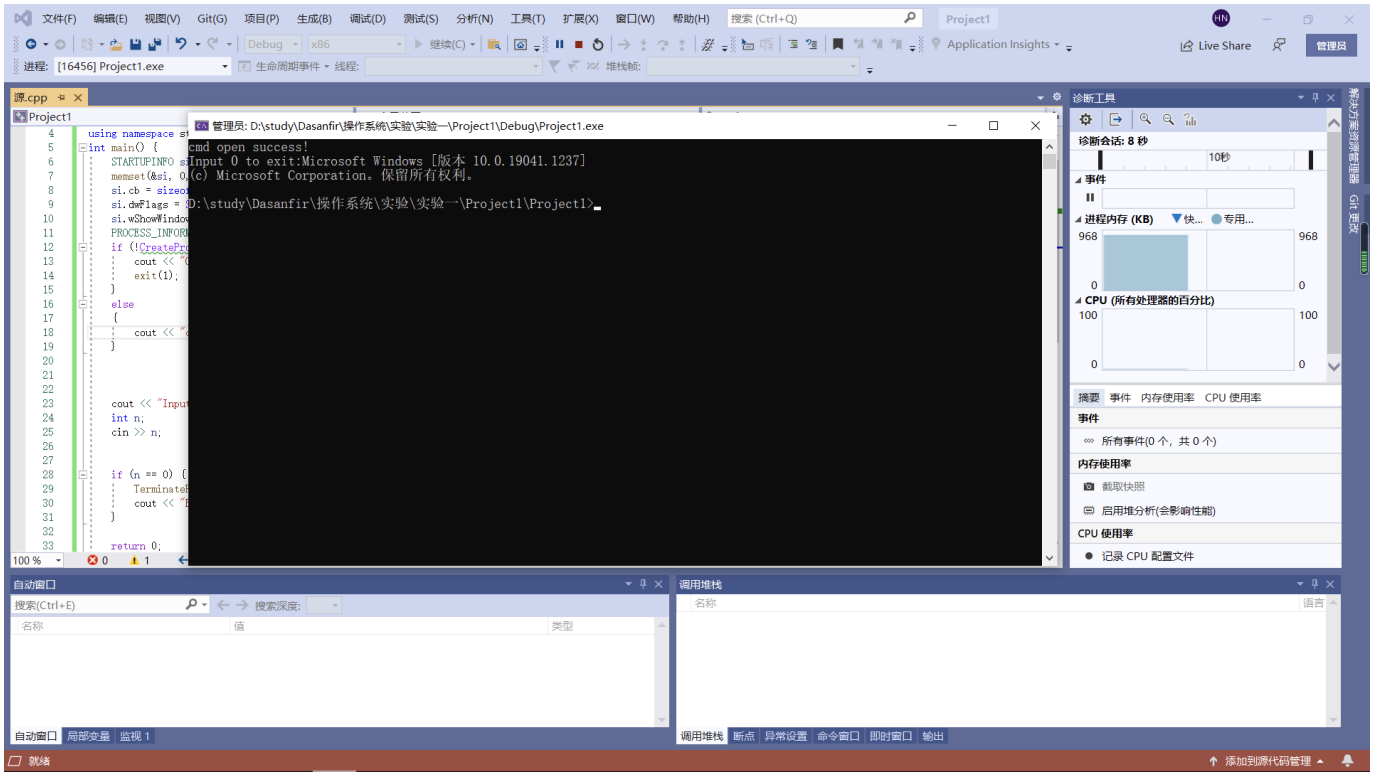
2. 利用VC++6.0实现上述程序设计和调试操作，对于进程创建的成功与否、终止进程操作的成功与否提供一定的提示框。

3. 通过阅读和分析实验程序，学习创建进程、观察进程和终止进程的程序设计方法。

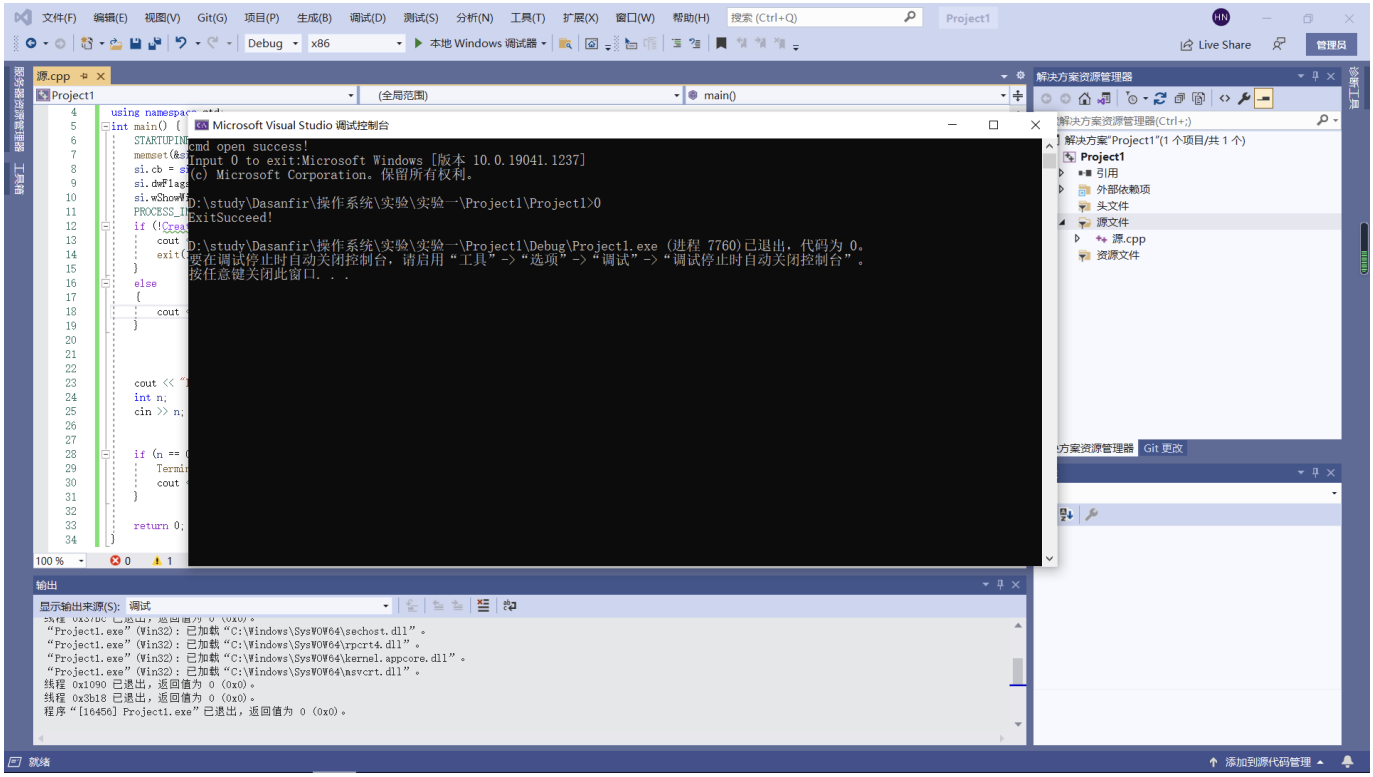
三、实验结果

由于本身visual Studio C++就是对VC++的升级，因此我直接在VS下编写程序并得以实现。

创建一个cmd进程结果如下：



输入0后终止该进程：



四、实验中遇到的问题及解决方法

1. 因为之前未接触过调用api的程序，因此需要对CreateProgcess()等进程操作的函数多做了解。在使用过程中忘记使用NULL作为空指针，以及后面lpStartupInfo和

lpProcessInfomation两个参数没有引用其地址，导致程序报错，经过引用其参数地址以后得以解决。

2．进程的句柄与退出代码

使用退出函数TerminateProcess()时不了解其参数含义，通过查阅资料了解句柄含义如下：

从数据类型上来看它只是一个32位(或64位)的无符号整数

在Windows环境中，句柄是用来标识项目的。WINDOWS程序中并不是用物理地址来标识一个内存块，文件，任务或动态装入模块的。相反，WINDOWS API给这些项目分配确定的句柄，并将句柄返回给应用程序，然后通过句柄来进行操作。

在程序设计中，句柄是一种特殊的智能指针 。当一个应用程序要引用其他系统（如数据库、操作系统）所管理的内存块或对象时，就要使用句柄。

句柄与普通指针的区别在于，指针包含的是引用对象的内存地址，而句柄则是由系统所管理的引用标识，该标识可以被系统重新定位到一个内存地址上。这种间接访问对象的模式增强了系统对引用对象的控制。

在本实验中，Create Process()中创建了进程pi ,可直接用pi.hProcess获取。

五、思考题

1.对可执行文件分配内存，执行。

2. 编译程序读取源程序（字符流），对之进行词法和语法的分析，将高级语言指令转换为效的汇编代码，再由汇编程序转换为机器语言，并且按照操作系统对可执行文件格式的要求链接生成可执行程序。

3. 系统为所有用户进程维护了一个task table，在这里面存放着指向每个进程的进程控制块（PCB）的指针。在某次时钟中断中，轮到一个进程被唤醒（新建进程也得乖乖进入任务队列排队），CPU读取该进程PCB结构中那个指向TTS的指针。各进程的TTS构成一个表，表的段描述符存放于GDT（全局段描述符表）或者LDT（局部段描述符表）。然后CPU读取TTS中的各项数据并且根据它的各项值来设置寄存器，包括CS（代码段选择子）和IP（指令偏移地址），而这就是进程的入口。接下来，CPU开始执行进程的指令。