# 实验一 进程的创建和撤销

# 专业 软件工程 姓名 毛济洲 学号 20182344050

## 一．实验目的

（1）       加深对进程概念的理解和进程创建与撤消算法；

（2）       进一步认识并发执行的实质。

## 二．实验内容

（1）进程管理认识：通过Windows “任务管理器”观察进程的状态，进行进程的创建、切换和撤销。

（2）进程创建与撤销：通过系统调用编程实现进程的创建与撤销功能，并在屏幕上观察进程活动的结果。

（3）通过任务管理器，对比查看进程创建撤销前后的变化情况。

### 背景知识

**1.** 创建进程：**CreateProcess()** 调用的核心参数是可执行文件运行时的文件名及其命令行.下表详细地列出了每个参数的类型和名称.

|  |  |
| --- | --- |
| 参数名称 | 使用目的 |
| LPCTSTR lpApplivationName | 全部或部分地指明包括可执行代码的EXE文件的文件名 |
| LPCTSTR lpCommandLine | 向可执行文件发送的参数 |
| LPSECURIITY\_ATTRIBUTES lpProcessAttributes | 返回进程句柄的安全属性.主要指明这一句柄是否应该由其他子进程所继承 |
| LPSECURIITY\_ATTRIBUTES lpThreadAttributes | 返回进程的主线程的句柄的安全属性 |
| BOOL bInheritHandle | 一种标志，告诉系统允许新进程继承创建者进程的句柄 |
| DWORD dwCreationFlage | 特殊的创建标志 (如CREATE\_SUSPENDED) 的位标记 |
| LPVOID lpEnvironment | 向新进程发送的一套环境变量；如为null值则发送调用者环境 |
| LPCTSTR lpCurrentDirectory | 新进程的启动目录 |
| STARTUPINFO lpStartupInfo | STARTUPINFO结构，包括新进程的输入和输出配置的详情 |
| LPPROCESS\_INFORMATION lpProcessInformation | 调用的结果块；发送新应用程序的进程和主线程的句柄和ID |

**2.** 终止进程：所有进程都是以调用**ExitProcess()** 或者**TerminateProcess()** 函数结束的.但最好使用前者而不要使用后者，因为进程是在完成了它的所有的关闭“职责”之后以正常的终止方式来调用前者的.而外部进程通常调用后者即突然终止进程的进行，由于关闭时的途径不太正常，有可能引起错误的行为.

TerminateProcess() API函数只要打开带有PROCESS\_TERMINATE访问权的进程对象，就可以终止进程，并向系统返回指定的代码.这是一种“野蛮”的终止进程的方式，但是有时却是需要的.

如果开发人员确实有机会来设计“谋杀”(终止别的进程的进程) 和“受害”进程 (被终止的进程) 时，应该创建一个进程间通讯的内核对象——如一个互斥程序——这样一来，“受害”进程只在等待或周期性地测试它是否应该终止.

## 三．实验步骤

**（1）       任务分析：**

   以下是创建进程的关键知识：

   进程创建和撤销需要调用CreateProcess（）和TerminateProcess()。CreateProcess()实际上是运行一个新的可执行程序，这里，需要设置其必备的参数。而终止进程则首先是要找到运行进程的句柄，然后终止该进程。

**（2）       程序设计：**

a.总体设计：

通过main()函数调用两个函数void startprocess()和函数void stop\_process()。主要是在main()函数中用while循环和开关语句switch选择序号对其进行控制，可以先创建进程并观察然后在选择序号终止该进程。

b.具体实现：  首先我们必须知道函数CreateProcess()的参数定义（类型和名称），共10个参数，其函数原型为

BOOL CreateProcess

　　(

　　LPCTSTR lpApplicationName,

　　LPTSTR lpCommandLine,

　　LPSECURITY\_ATTRIBUTES lpProcessAttributes。

　　LPSECURITY\_ATTRIBUTES lpThreadAttributes,

　　BOOL bInheritHandles,

　　DWORD dwCreationFlags,

　　LPVOID lpEnvironment,

　　LPCTSTR lpCurrentDirectory,

　　LPSTARTUPINFO lpStartupInfo,

　　LPPROCESS\_INFORMATION lpProcessInformation

　　);

其中第二个参数lpCommandLine参数允许调用者向新应用程序发送数据；STARTUPINFO所必需的，用于在必要时指明新应用程序的主窗口的外观. lpProcessInformation用于调用的结果块和发送新应用程序的进程和主线程的句柄和ID号。其他参数可以按缺省值为空。

判断创建进程是否成功，若不成功，输出创建不成功并返回；若成功则显示程序并获得句柄，以方便终止句柄。在终止进程过程中，若句柄不为空，调用TerminateProcess（）进一步判断。

其函数原型

BOOL   TerminateProcess(

      HANDLE   hProcess,   //   handle   to   the   process

      UINT   uExitCode

  );

（3）       程序结果：打印程序的输出结果（表明对应的数据）或主要界面图。

  开始时的界面：（如图所示）

输入序号1后回车 产生的结果：

接着输入序号2后该进程被撤销后的结果：

发现刚才创建的IE进程被杀死了

输入序号3退出程序：

（4）       调试与测试：

如果进入主界面时，直接执行序号2终止进程：

上述结果表明 ……。

## 四．实验总结

* 首先查看“任务管理器”进程的动态，并新建和关闭进程观察。接着是要创建进程和撤销进程，首要工作是弄清楚两个系统函数CreateProcess（）和TerminateProcess()入口参数的类型和函数，弄明白这些关键因素，然后就是如何将它们组织起来，用控制台下的switch语句来控制到底进行哪种操作。
* 通过这次试验，加深了平时学习的理论知识和实践的联系，并加深了印象。不过，我觉得这次试验如果能够……方法会更加直观，有些……弄的还是不太明白，还需要多多努力。

## 五．附录源代码

#include <iostream> #include <windows.h>

using namespace std; HANDLE m\_hPro; void startprocess()//创建进程

{

STARTUPINFO si;//必备的参数设置

int n=1;

memset(&si,0,sizeof(STARTUPINFO));

si.cb=sizeof(STARTUPINFO);

si.dwFlags=STARTF\_USESHOWWINDOW;

si.wShowWindow=SW\_SHOW;

PROCESS\_INFORMATION pi;

if(!CreateProcess(NULL,"C://Program Files//Internet Explorer//iexplore.exe",NULL,NULL,false,0,NULL,NULL,&si,&pi))

{

cout <<"Create Process Fail!" <<endl;

exit(1);

}

else

{

cout <<"专业+姓名+学号创建进程成功!" <<endl;

m\_hPro=pi.hProcess;

}

}

void stop\_process()

{

if(m\_hPro)

{

if(!TerminateProcess(m\_hPro,0))

{

//终止出现错误，显示错误信息

LPVOID lpMsgBuf;

FormatMessage(FORMAT\_MESSAGE\_ALLOCATE\_BUFFER |

FORMAT\_MESSAGE\_FROM\_SYSTEM |

FORMAT\_MESSAGE\_IGNORE\_INSERTS,

NULL,

GetLastError(),

MAKELANGID(LANG\_NEUTRAL, SUBLANG\_DEFAULT),

(LPTSTR) &lpMsgBuf,

0,

NULL );

}

else

{

cout<<"TerminateProcess sucess"<<endl;

}

m\_hPro=NULL;

}

else

{ cout<<"null m\_hPro"<<endl;

}

}

int main()

{

while(1){

cout<<"请输入序号"<<endl;

cout<<"1 创建进程 2 终止进程 3 退出"<<endl;

int i;

cin>>i;

switch(i)

{

case 1:startprocess();break;

case 2:stop\_process();break;

case 3:exit(1);

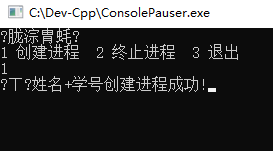
}

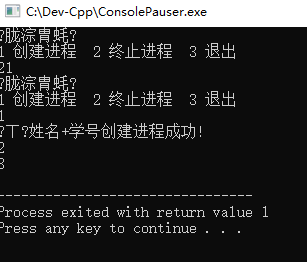
}

return(0);

}

## 六、调试分析





按1出啊昂见IE浏览器进程，按2结束进程，按3退出程序