

天津商业大学信息工程学院专业实验室学生实验报告

实验日期：2014 年 11 月 26 日

实验时间：10:05——11:40

课程名称：操作系统

任课教师：苗序娟

实验成绩：

专业班级：软件工程 1201 班

学生姓名：王靖伟

实验项目名称：实验二 进程控制实践

实验设备：PC 一台，Windows 2000/XP 及以上操作系统, VC++ 6.0

实验目的、要求：

- 1、使用 MFC 提供的 CreateProcess 函数创建进程，并给出系统的进程的基本信息，如 ID 号，使用 Terminate 函数中止所创建的进程。
- 2、熟悉对进程的控制与描述的理解并熟悉 VC 的使用。
- 3、认真完成实验，正确记录各项数据，记录实验过程中出现的问题及其解决过程。
- 4、实验理论基础主要对应于教材中第 3-7 章的内容。

实验描述：

1、实验内容：进程的创建和终止。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。

2、实验步骤：

进程的创建、进程信息的列出、以及进程的删除实验，请回答如下问题：

(1) 请将进程创建代码粘贴如下：(10 分)

行数	void CProcessDlg::OnCreatproButton()
1	void CProcessDlg::OnCreatproButton()
2	{
3	// TODO: Add your control notification handler code here
4	PROCESS_INFORMATION pi;
5	STARTUPINFO si;
6	memset(&si,0,sizeof(si));
7	si.cb = sizeof(si);
8	bool fRet = CreateProcess(NULL, //程序名称
9	"cmd.exe", //程序执行命令行参数
10	NULL, //设定进程对象的安全属性
11	NULL, //设定线程对象的安全属性
12	false, //继承父进程属性开关
13	NORMAL_PRIORITY_CLASS CREATE_NEW_CONSOLE,
14	//标识标志，用来表示如何来创建进程
15	NULL, //指向包含新进程将要使用的环境字符串的内存块
16	NULL, //设置子进程的当前驱动器和目录
17	&si, //进程的初始化信息
18	&pi); //指向必须置顶的 PROCESS_INFORMATION 结构,

19	CreateProcess 在返回之前要对该结构进行初始化
20	
21	}

(2) 请将进程信息列出的代码粘贴如下：(10 分)

行数	void CProcessDlg::OnGetprocess()
1	void CProcessDlg::OnGetprocess()
2	{
3	// TODO: Add your control notification handler code here
4	//设置 ListBox, 用来显示当前进程
5	m_list.DeleteAllItems();
6	while(m_list.DeleteColumn(0));
7	m_list.ModifyStyle(0,LVS_REPORT);
8	m_list.InsertColumn(0,"进程 ID",LVCFMT_LEFT,80);
9	m_list.InsertColumn(1,"进程名称",LVCFMT_LEFT,150);
10	
11	//获得当前进程的一个快照的句柄
12	HANDLE handle = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS,0);
13	PROCESSENTRY32 *info = new PROCESSENTRY32;
14	info->dwSize = sizeof(PROCESSENTRY32);
15	int i = 0;
16	if (Process32First(handle,info))
17	/*process32First 是一个进程获取函数,
18	当我们利用函数 CreateToolhelp32Snapshot() 获得当前运行进程
19	的快照后,我们可以利用 process32First 函数来获得第一个进程的
20	句柄.其原型为(用的是 vfp)
21	DECLARE INTEGER Process32First IN WIN32API ;
22	INTEGER hSnapshot,STRING @ lppe
23	在 c 语言中如下
24	BOOL WINAPI Process32First(HANDLE hSnapshot,LPROCESSENTRY32 lppe);
25	*/
26	{
27	if (GetLastError() == ERROR_NO_MORE_FILES)
28	{
29	AfxMessageBox("No more process");
30	}
31	else
32	{
33	CString id;
34	id.Format("%d",info->th32ProcessID);
35	m_list.InsertItem(i,id);
36	m_list.SetItemData(i,info->th32ProcessID);
37	id.Format("%s",info->szExeFile);
38	m_list.SetItemText(i,1,id);

39	i++;
40	while (Process32Next(handle,info) !=FALSE)
41	//与 Process32First 类似，获得下一个进程信息
42	{
43	//将进程信息显示到 ListBox
44	id.Format("%5d",info->th32ProcessID);
45	m_list.InsertItem(i,id);
46	m_list.SetItemData(i,info->th32ProcessID);
47	id.Format("%s",info->szExeFile);
48	m_list.SetItemText(i,1,id);
49	i ++;
50	}
51	}
52	}
53	CloseHandle(handle);
54	
55	}

(3) 请将进程删除的代码粘贴如下：(10 分)

行数	void CProcessDlg::OnKillprocess()
1	void CProcessDlg::OnKillprocess()
2	{
3	// TODO: Add your control notification handler code here
4	POSITION pos;
5	pos = m_list.GetFirstSelectedItemPosition();
6	int select = m_list.GetNextSelectedItem(pos);
7	
8	//获得选中进程的句柄
9	HANDLE h =
10	OpenProcess(PROCESS_ALL_ACCESS,TRUE,m_list.GetItemData(select));
11	if (h != NULL)
12	{
13	//结束进程
14	TerminateProcess(h,0);
15	}
16	OnGetprocess();
17	
18	}

(4) 详细描述实验过程，在编码过程中都遇见了什么问题，你是如何解决的？（30 分）

1. 打开 VC++6.0，创建一个名为 Process 的工程，如图 1 所示。

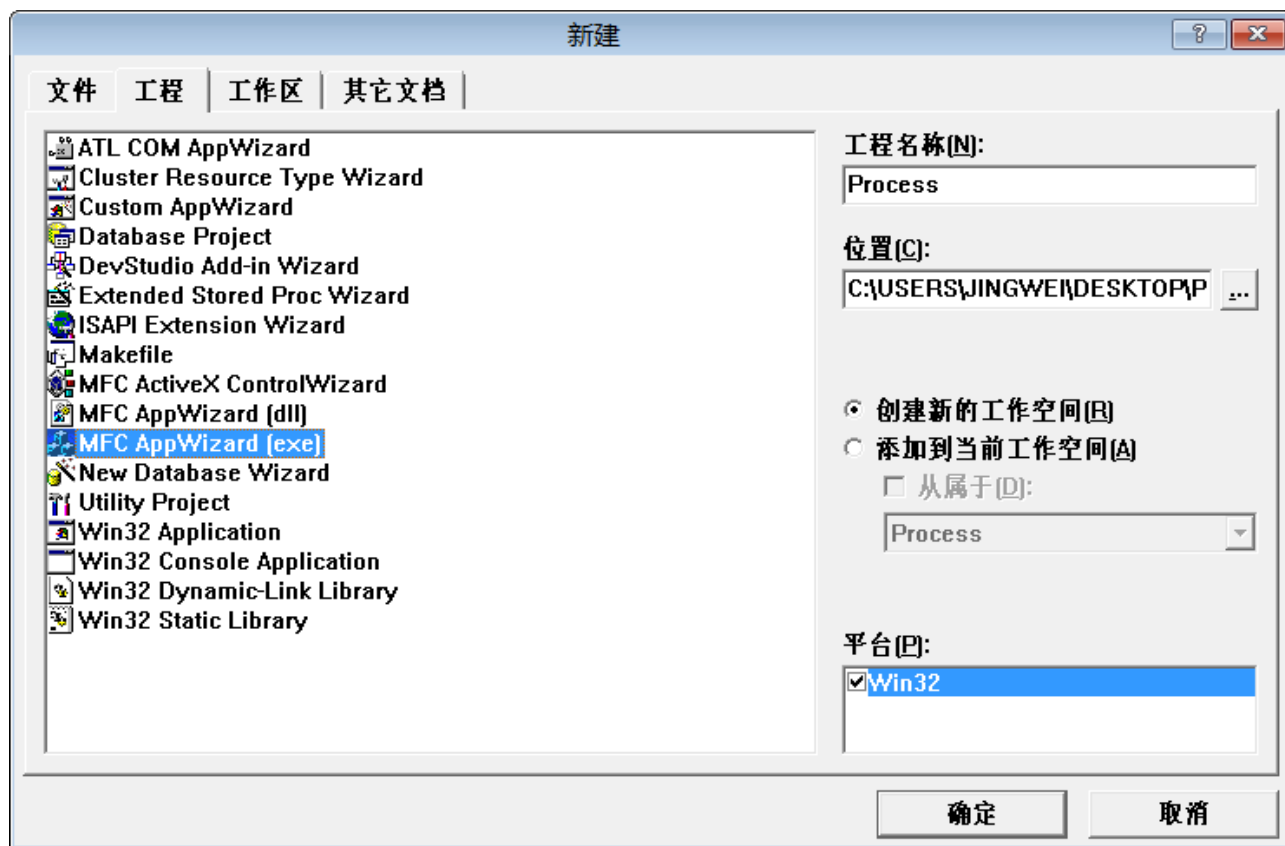


图 1. 创建名为 Process 的工程

2. MFC 应用向导对话框中选择**基本对话框(D)**直接点击完成，如图 2 所示。

3. 跳出新建工程对话框，直接点击确定。

4. 把控件拖到对话框上，如图 3 所示。

5. 修改各个控件的属性。四个 Button 控件的 ID 和标题分别为“IDC_CREATPRO_BUTTON”“创建进程”、“IDC_GETPROCESS”“获取进程”、“IDC_KILLPROCESS”“杀死进程”、“IDC_CANCEL”“退出”。举例“创建进程”、“获取进程”按钮如图 4、图 5 所示。

6. 右键点击 List Control 控件，点击**建立类向导**，为“IDC_LIST1”添加类型为“CListCtrl”的成员变量“m_list”，如图 6 所示。

7. 分别双击四个 Button 控件，添加代码。

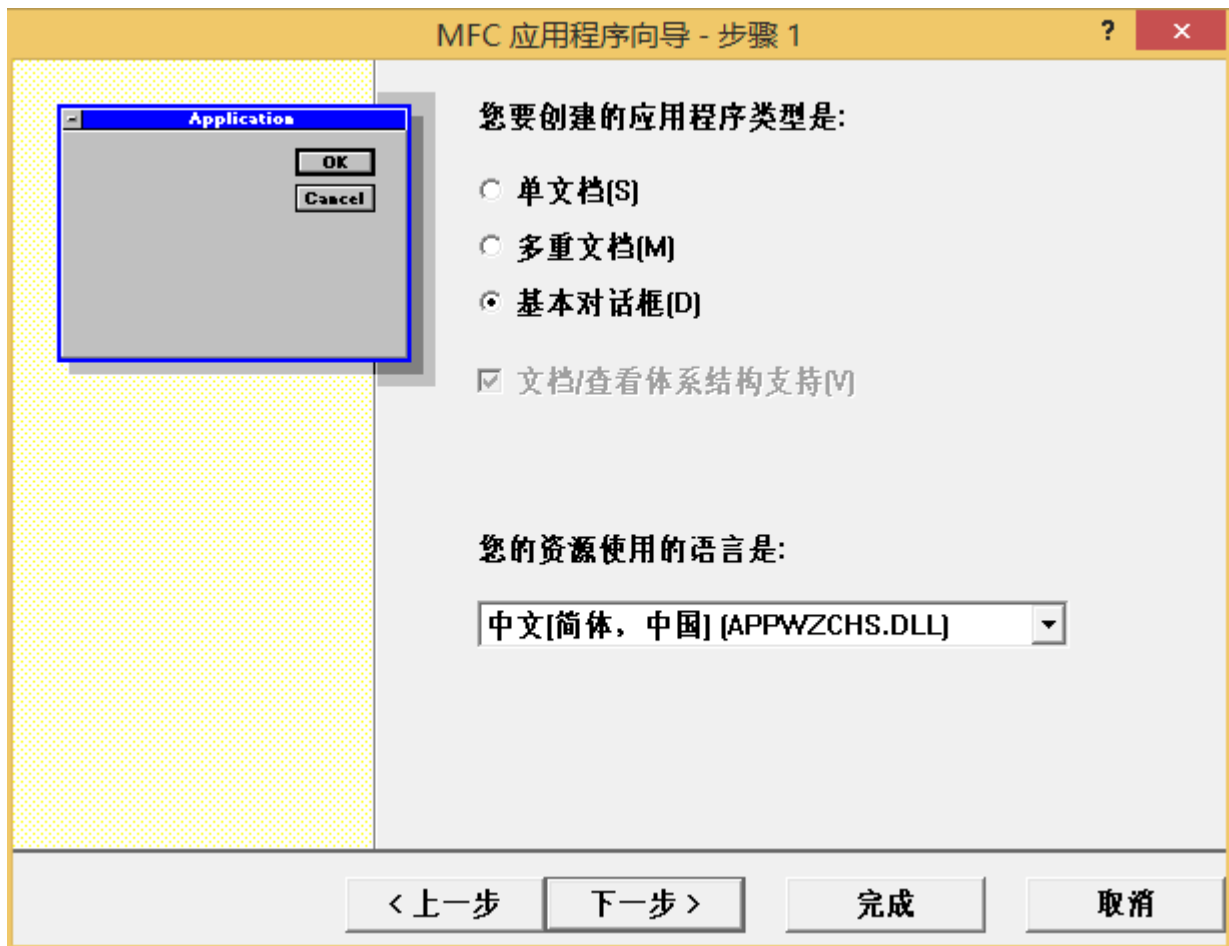


图 2. MFC 应用向导对话框

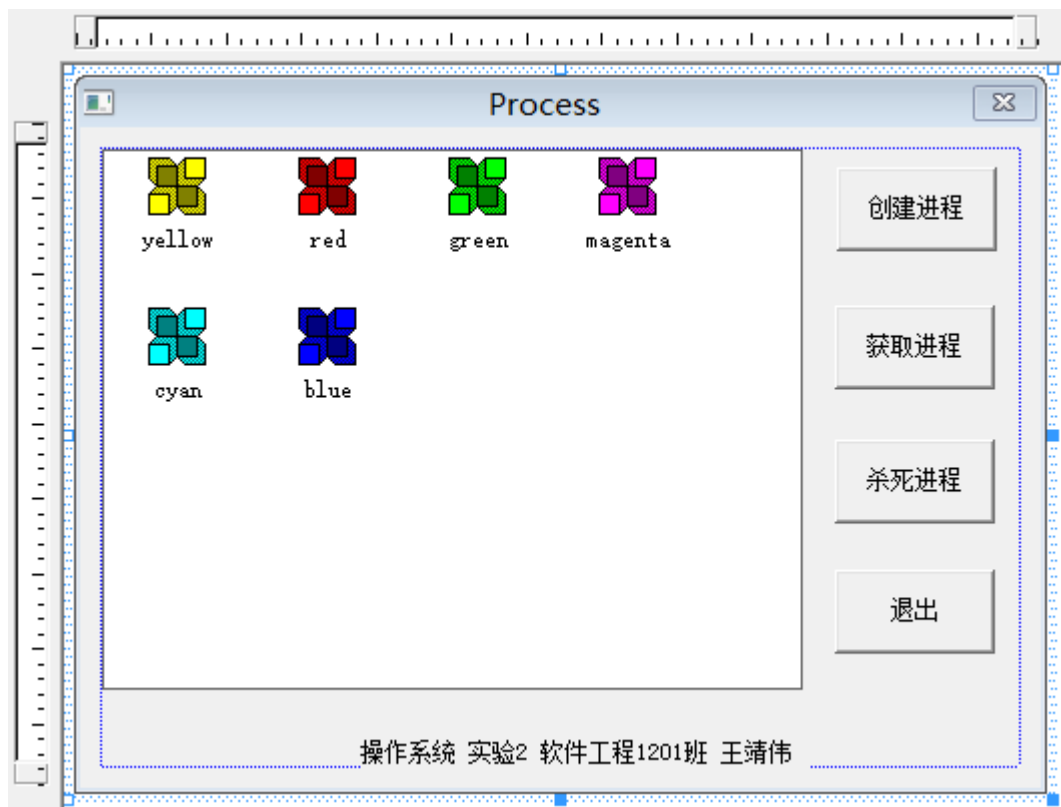


图 3. 把控件拖到对话框上



图 4. “创建进程”按钮属性

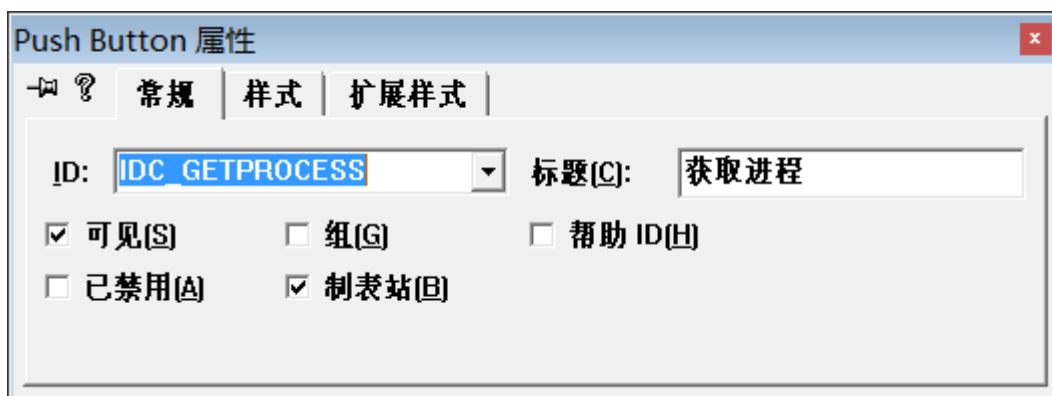


图 5. “获取进程”按钮属性

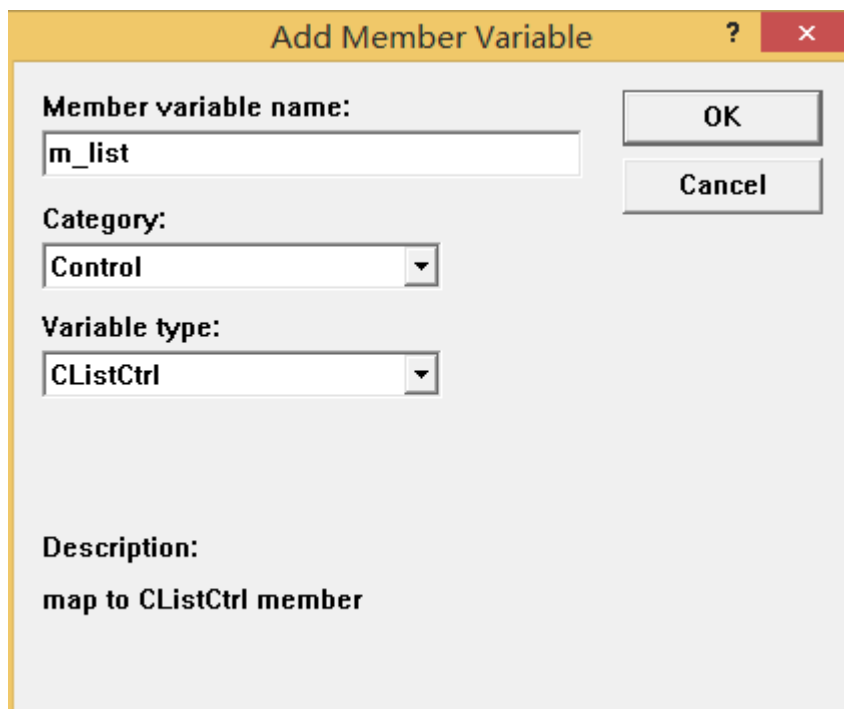


图 6. 为“IDC_LIST1”添加类型为“CListCtrl”的成员变量“m_list”

在编码过程中都遇见的问题：

在按照实验指导书为控件添加完代码后，编译时报错，报错信息如下所示。

```
-----Configuration: Process - Win32 Debug-----  
  
Compiling...  
ProcessDlg.cpp  
F:\★总表\课程\操作系统\操作系统实验\实验 2\Process\ProcessDlg.cpp(193) : warning  
C4800: 'int' : forcing value to bool 'true' or 'false' (performance warning)  
F:\★总表\课程\操作系统\操作系统实验\实验 2\Process\ProcessDlg.cpp(267) : error C2065:  
'OngetProcess' : undeclared identifier  
执行 cl.exe 时出错.  
  
ProcessDlg.obj - 1 error(s), 0 warning(s)
```

在对上面的报错分析之后发现添加控件时修改属性后自动生成的函数名称的大小写与实验指导书上的有出入，改为上文代码清单(3)的 `void CProcessDlg::OnKillprocess()` 中的第 16 行代码所示，如图 7 所示，再次编译后，代码无错误，可以生成可执行文件。

```
void CProcessDlg::OnKillprocess()  
{  
    // TODO: Add your control notification handler code here  
    POSITION pos;  
    pos = m_list.GetFirstSelectedItemPosition();  
    int select = m_list.GetNextSelectedItem(pos);  
  
    //获得选中进程的句柄  
    HANDLE h = OpenProcess(PROCESS_ALL_ACCESS, TRUE, m_list.GetItemData(select));  
    if (h != NULL)  
    {  
        //结束进程  
        TerminateProcess(h, 0);  
    }  
    OnGetprocess();  
}
```

图 7. 修改后的代码

(5) 进程运行结果图：(3 个图，其中分别为创建结果图，浏览信息图和删除结果图)。(25 分)

1. 运行 Process.exe 如图 8，点击“创建进程”后，自动开启 cmd.exe，cmd.exe 中路径为 Process.exe 所在文件夹，创建 cmd.exe 进程结果如图 9 所示。



图 8. 运行 Process.exe

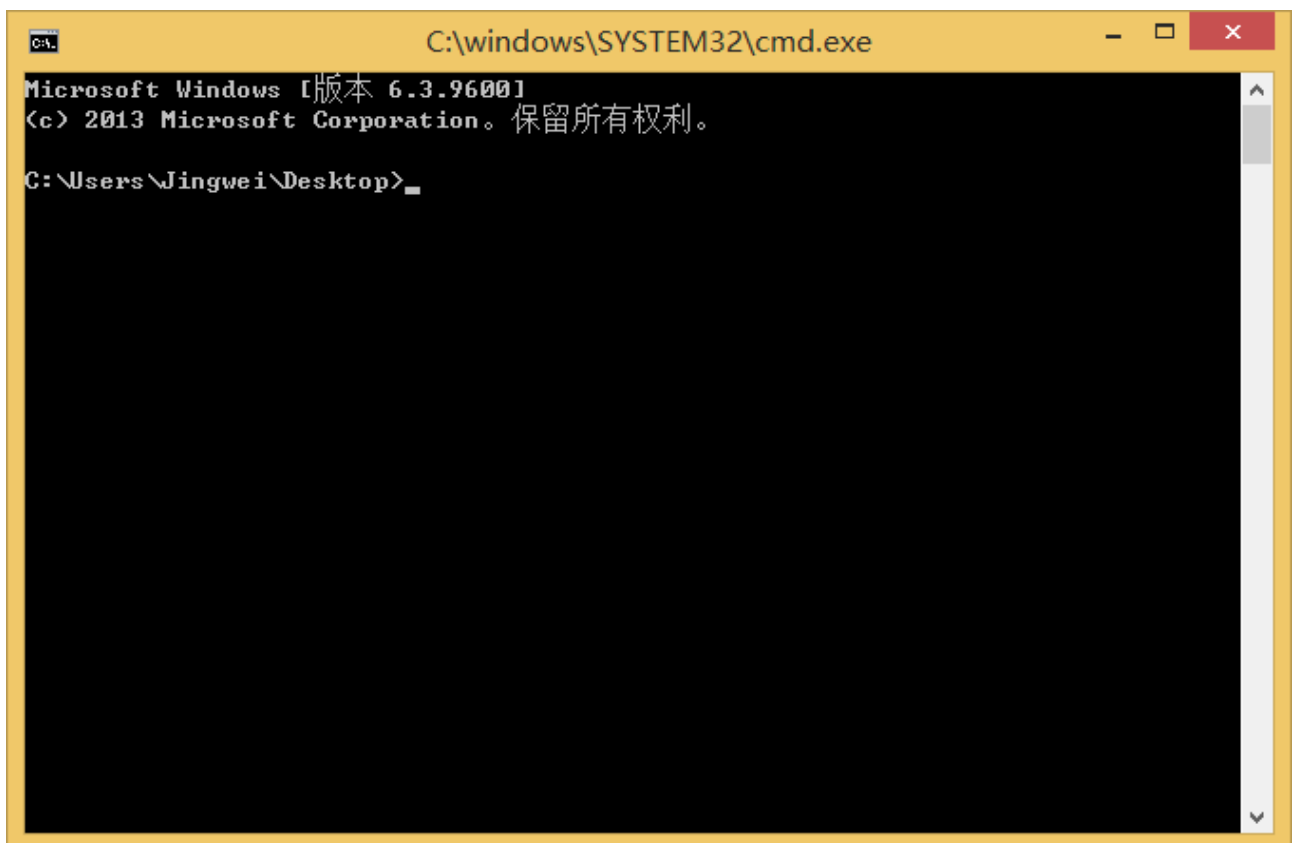


图 9. 点击“创建进程”后创建 cmd.exe 进程

2. 点击“获取进程”按钮后，获取本机当前运行的进程 ID 和进程名称，如图 10 所示。



图 10. 点击“获取进程”按钮后获取的信息

3. 点击列表中进程 ID 为 2720 的进程，为刚创建的 cmd.exe 进程，如图 11 所示。现在将其结束，点击“杀死进程”按钮后，cmd.exe 窗口消失，再次获取进程则不再有进程 ID 为 2720 的 cmd.exe 进程，如图 12 所示。

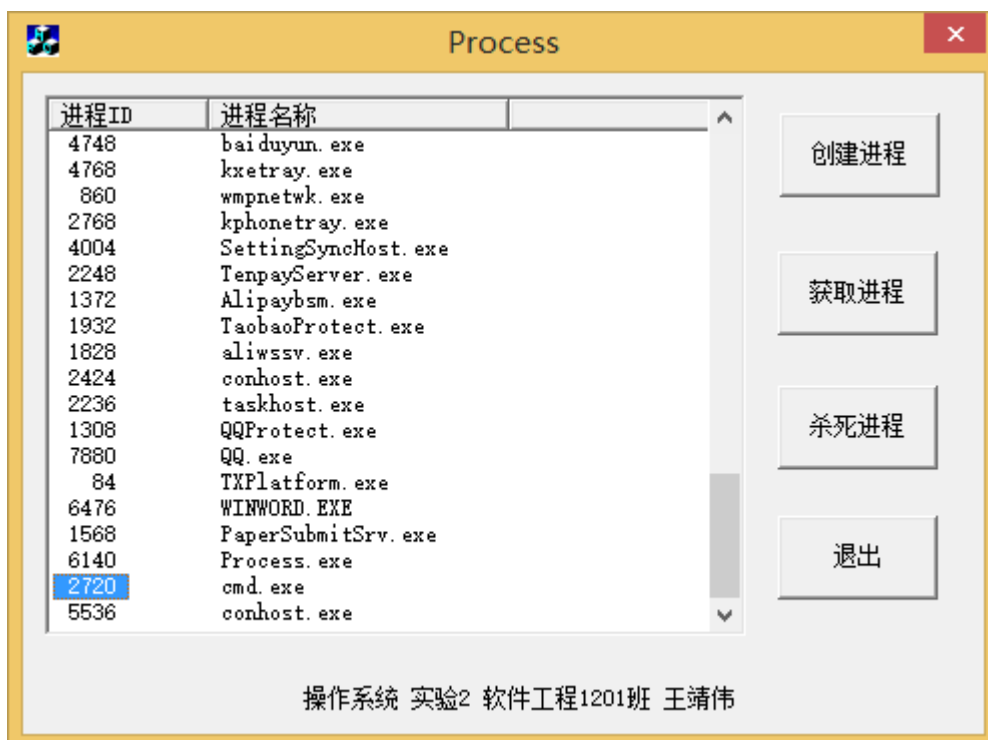


图 11. 选中进程 ID 为 2720 的进程



图 12. 点击“杀死进程”后的结果

- (6) 进程创建的 API 函数是： CreateProcess()函数。(5 分)
- (7) 获取后的进程信息存储在 __m_list__ 数据结构中。(5 分)
- (8) 删除进程的 API 函数是： TerminateProcess()函数。(5 分)