**山东师范大学**

**实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 院系部所 | 信息科学与工程学院 | | 课程名称 | | 计算机操作系统 |
| 所在班级 | 计工本1702 | 学号 | 201711010202 | 姓名 | 王汝芸 |
| 实验编号 | 01 | 题目 | 线程的创建与撤销 | | |

一、实验目的和要求

实验目的：

（1）熟悉Windows系统提供的线程创建与撤销系统调用

（2）掌握Windows系统环境下线程的创建与撤销方法

实验要求：

能正确使用CreateThread()、ExitThread()及Sleep()等系统调用函数，进一步理解进程与线程理论。

二、实验环境

Windows 10

VC++6.0

三、实验内容及实施

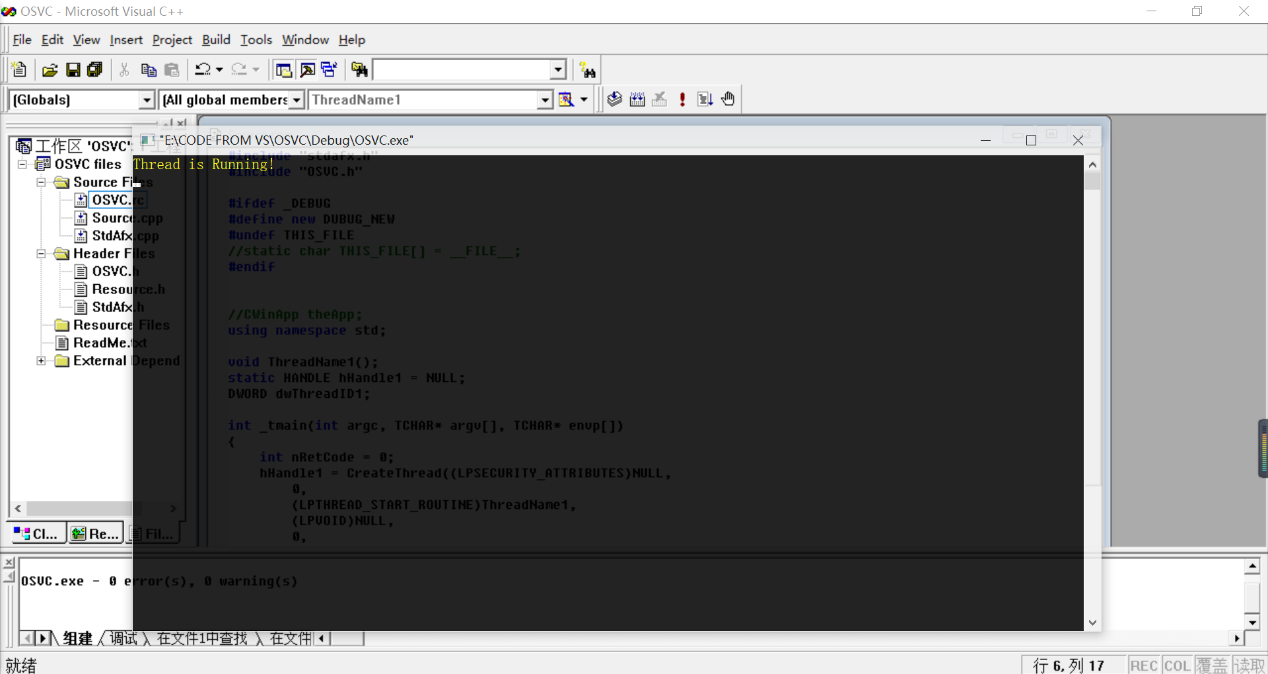
实验内容：

使用系统调用函数CreateThread()创建一个子线程，并在子线程序中显示：Thread is Running！。为了能让用户清楚地看到线程的运行情况，使用函数Sleep()使线程挂起5s，之后使用函数ExitThread(0)撤销线程。

实验流程：

1. 使用VC++6.0创建一个MFC项目；
2. 使用CreateThread()、ExitThread()及Sleep()等系统调用函数，实现线程的创建与撤销；
3. 编译运行。

四、实验结果

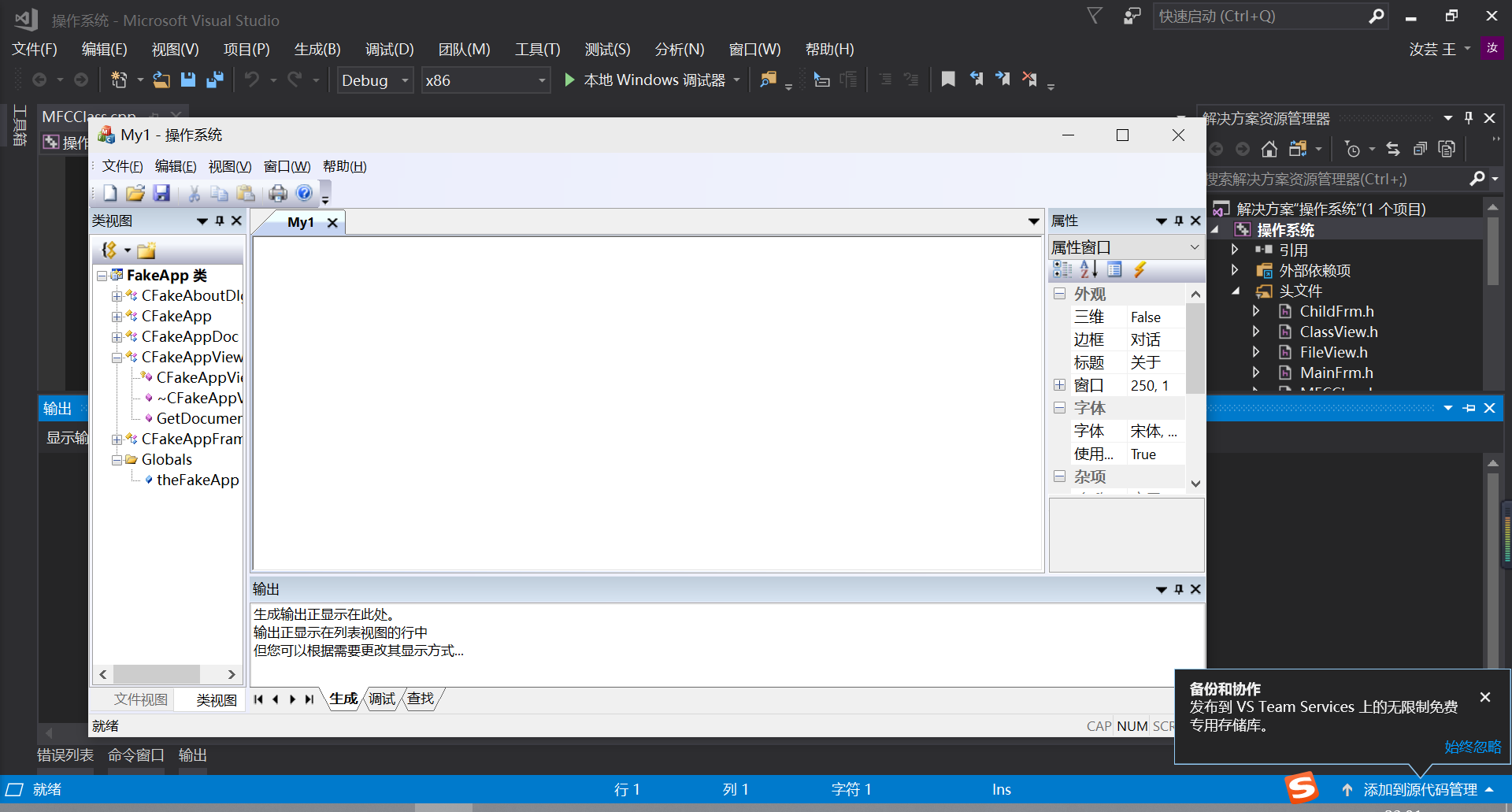


五、实验分析

在Windows系统中进程是资源的拥有者，线程是系统调度的单位。进程创建后，其主线程也随即被创建。在该实验中，又创建了一个名为ThreadName1的子线程，该子线程与主线程并发地被系统调度。为了能看到子线程的运行情况，在主线程创建了子线程后，将主线程挂起5s以确保子线程能够运行完毕，之后调用ExitThread(0)将所有线程（包括主、子线程序）撤销。

六、实验讨论（可选）

一开始尝试使用VS编译，反正啥也不懂，编译完也能运行，但是不是想要的结果，出了个框框，也不知道该干啥



又回头用VC++6.0编译，注释了两个重定义：

//CWinApp theApp;

//static char THIS\_FILE[] = \_\_FILE\_\_;

就能运行出来了，结果如贴图所示

七、附录 （可选）

#include "stdafx.h"

#include "OSVC.h"

#ifdef \_DEBUG

#define new DUBUG\_NEW

#undef THIS\_FILE

//static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

//CWinApp theApp;

using namespace std;

void ThreadName1();

static HANDLE hHandle1 = NULL;

DWORD dwThreadID1;

int \_tmain(int argc, TCHAR\* argv[], TCHAR\* envp[])

{

int nRetCode = 0;

hHandle1 = CreateThread((LPSECURITY\_ATTRIBUTES)NULL,

0,

(LPTHREAD\_START\_ROUTINE)ThreadName1,

(LPVOID)NULL,

0,

&dwThreadID1);

Sleep(5000);

CloseHandle(hHandle1);

ExitThread(0);

return nRetCode;

}

void ThreadName1()

{

printf("Thread is Running!\n");

}