**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 | 05 | 题目 | 动态链接库的建立与调用 | | |

一、实验目的和要求

（1）理解动态链接库的实现原理，

（2）掌握Windows系统动态链接库的建立方法。

（3）掌握Windows环境下动态链接库的调用方法。

二、实验环境

VC++6.0

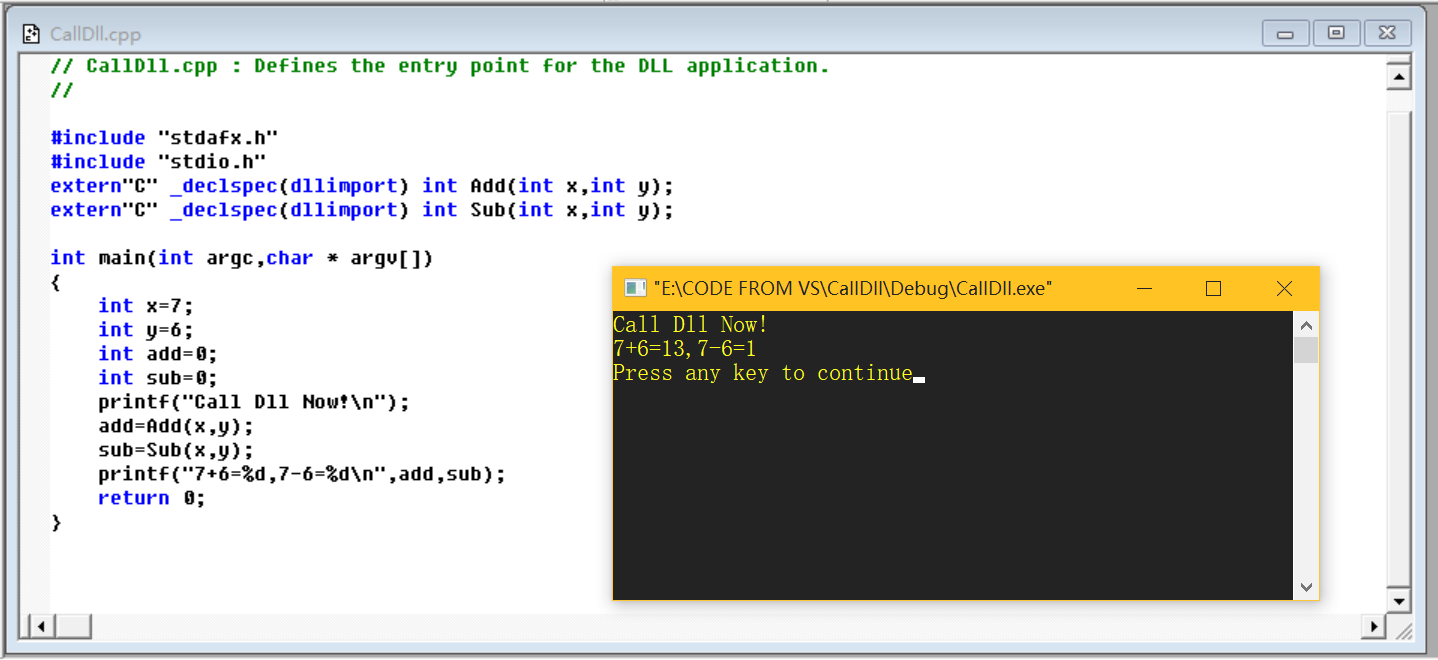
Windows 10

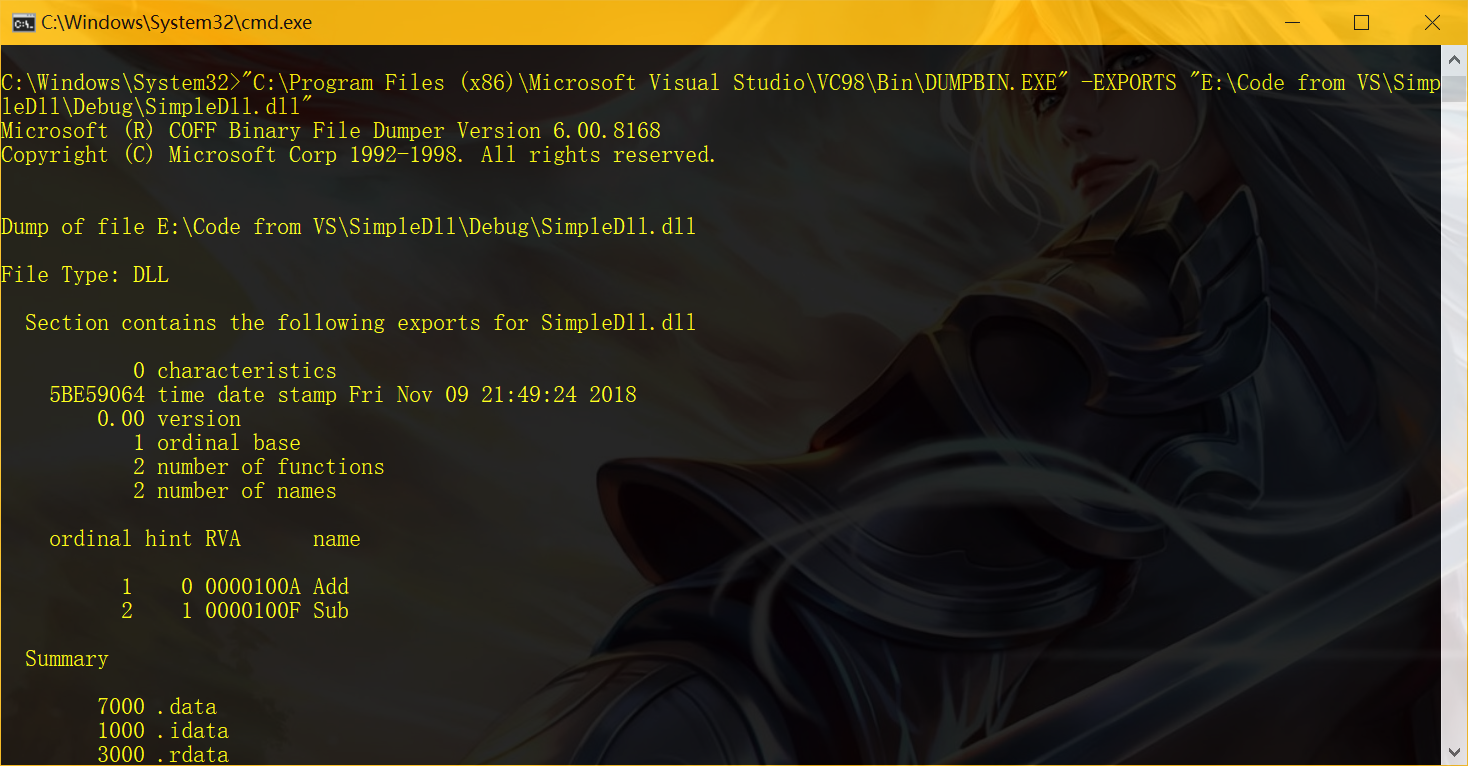
三、实验内容及实施

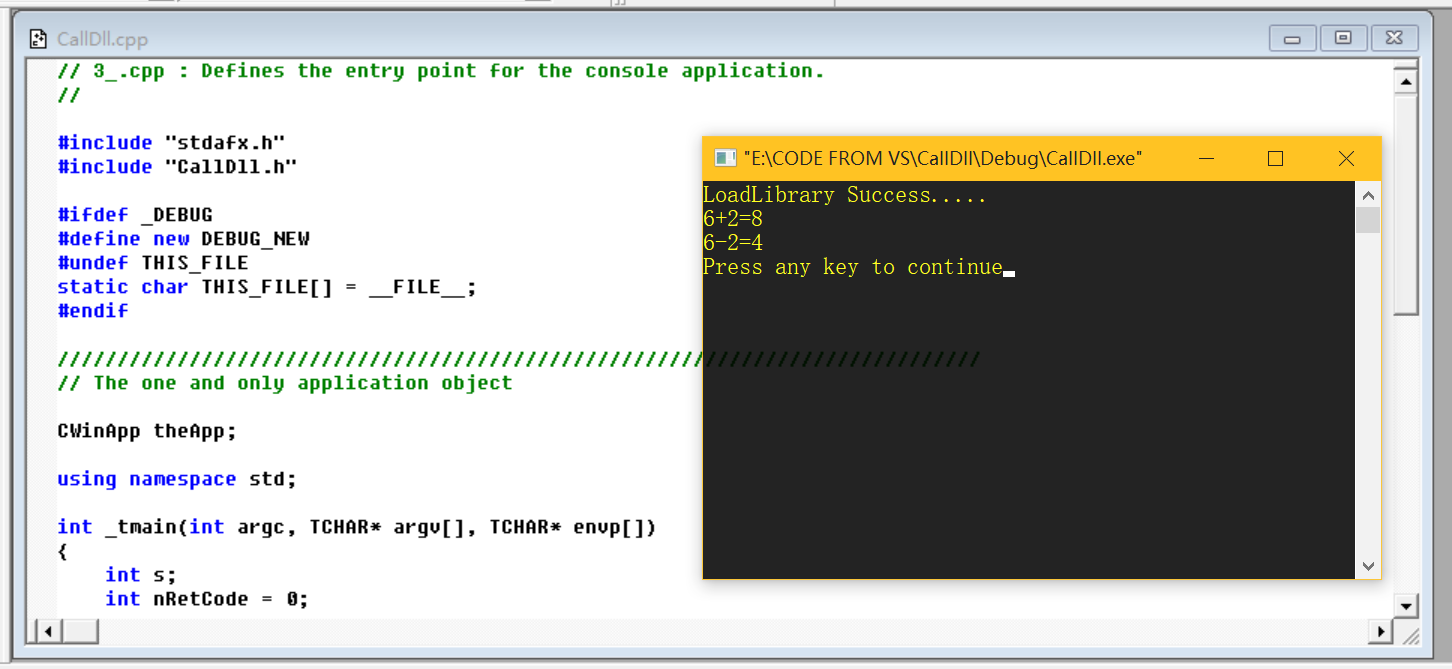
动态链接库知识，动态链接库入口函数，动态链接库导入/导出函数，动态链接库的连接方式

1. 在Windows环境下建立一个动态链接库
2. 使用隐形调用法调用动态链接库
3. 使用显式调用法调用动态链接库

四、实验结果







五、实验分析





六、实验讨论（可选）

一开始建的CallDll工程用的Dll类型，运行时总是让指定可执行文件，点了web之后一运行就弹出浏览器，心态就崩了，debug根本无从下手，从机房当了一节课咸鱼。回来之后仔细阅读指导书发现CallDll项目是控制台类型，改正后终于得到结果。

在仔细阅读实验指导书的过程中，学会了Dumpbin的使用方法，但是也不明白是干啥的。

七、附录 （可选）

// SimpleDll.cpp : Defines the entry point for the DLL application.

//

#include "stdafx.h"

extern "C" \_declspec(dllexport) int Add(int x, int y);

extern "C" \_declspec(dllexport) int Sub(int x, int y);

BOOL APIENTRY DllMain(HANDLE hModule,

DWORD ul\_reason\_for\_call,

LPVOID lpReserved

)

{

return TRUE;

}

int Add(int x, int y)

{

int z;

z = x + y;

return z;

}

int Sub(int x, int y)

{

int z;

z = x - y;

return z;

}

//隐式调用动态链接库的程序  
// CallDll.cpp : Defines the entry point for the DLL application.  
//  
  
#include "stdafx.h"  
#include "stdio.h"  
extern"C"\_declspec(dllimport) int Add(int x,int y);  
extern"C"\_declspec(dllimport) int Sub(int x,int y);  
  
int main(int argc,char \* argv[])  
{  
int x=7;  
int y=6;  
int add=0;  
int sub=0;  
printf("Call Dll Now!\n");  
add=Add(x,y);  
sub=Sub(x,y);  
printf("7+6=%d,7-6=%d\n",add,sub);  
return 0;  
}

// CallDll\_.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "CallDll.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

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

#endif

/////////////////////////////////////////////////////////////////////////////

// The one and only application object

CWinApp theApp;

using namespace std;

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

{

int s;

int nRetCode = 0;

typedef int (\* pAdd)(int x,int y);

typedef int (\* pSub)(int x,int y);

HMODULE hDll;

pAdd add;

pSub sub;

hDll=LoadLibrary("SimpleDLL.dll");

if(hDll==NULL)

{

printf("LoadLibrary Error......\n");

return nRetCode;

}

else printf("LoadLibrary Success.....\n");

add=(pAdd)GetProcAddress(hDll,"Add");

s=add(6,2);

printf("6+2=%d\n",s);

sub=(pSub)GetProcAddress(hDll,"Sub");

s=sub(6,2);

printf("6-2=%d\n",s);

FreeLibrary(hDll);

return nRetCode;

}