**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 | 06 | 题目 | 系统内存使用统计 | | |

一、实验目的和要求

实验目的：

（1）了解Windows内存管理机制，理解页式存储管理技术

（2）熟悉Windows内存管理基本数据结构

（3）掌握Windows内存管理基本API的使用

实验要求：

使用Windows系统提供的函数和数据结构显示系统存储空间的使用情况，当内存和虚拟存储空间发生变化是，观察系统显示变化情况。

二、实验环境

Windows 10

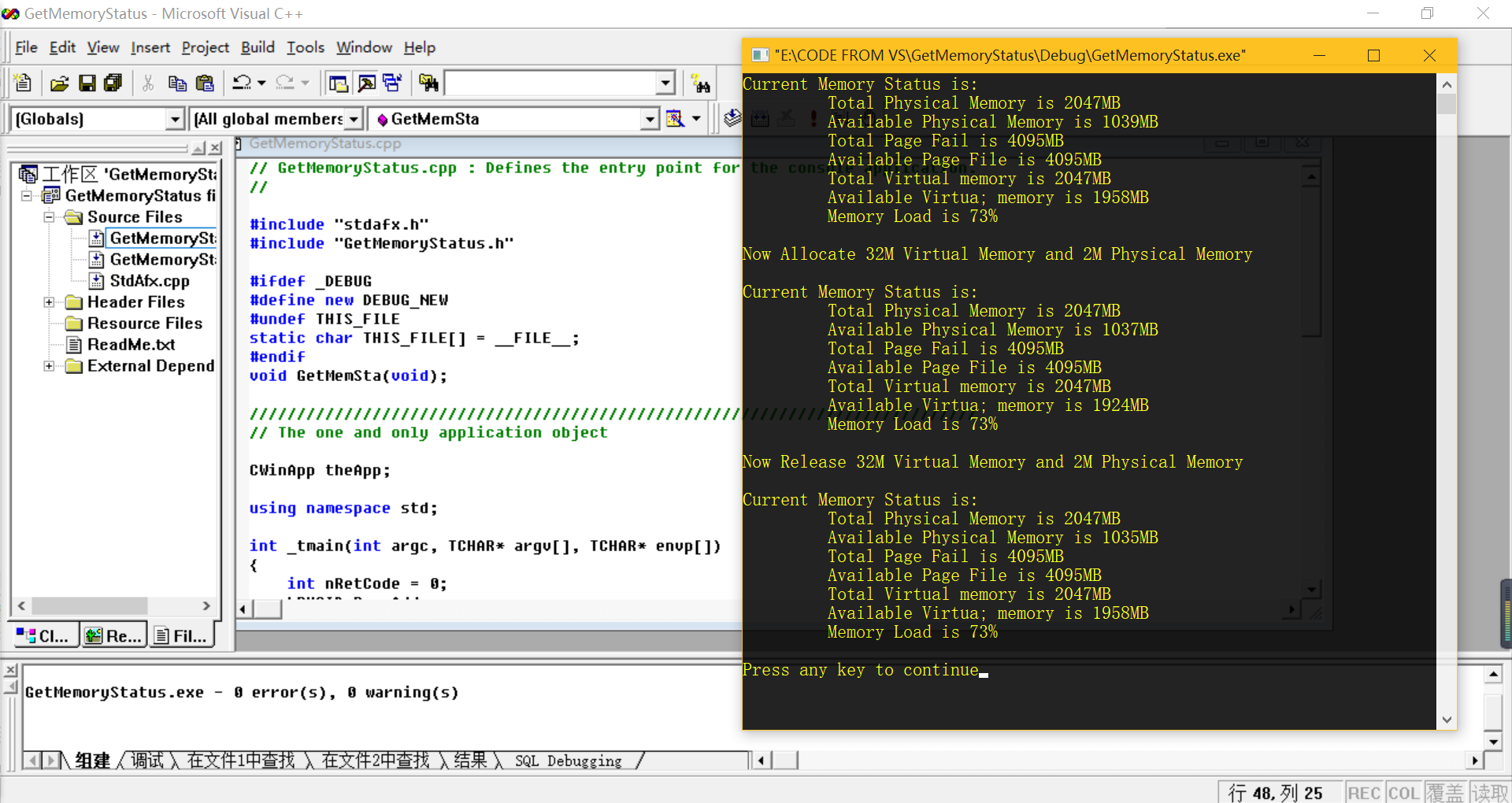
VC++6.0

三、实验内容及实施

实验步骤：建立MFC工程，运行源代码得到结果

实验内容：使用Windows系统提供的函数和数据结构显示系统存储空间的使用情况，当内存和虚拟存储空间发生变化是，观察系统显示变化情况。

四、实验结果



五、实验分析

（1）程序开始运行时，显示的可用内存空间为2047MB，可用页文件大小为4095MB，虚拟内存为1958MB；

（2）当分别使用函数VirtualAlloc()和malloc()分配了32MB虚拟内存和2MB物理内存后，系统显示可用物理内存为1037MB，可用页文件大小为4095MB，可用虚拟内存为1924MB；

（3）当分别使用函数VirtualFree()和free()释放32MB虚拟内存和2MB物理内存后，系统的显示情况又恢复了（1）的情况。

六、实验讨论（可选）

无。

七、附录 （可选）

1 *// GetMemoryStatus.cpp : Defines the entry point for the console application.*

2 *//*

3

4 #include "stdafx.h"

5 #include "GetMemoryStatus.h"

6

7 #ifdef \_DEBUG

8 #define new DEBUG\_NEW

9 #undef THIS\_FILE

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

11 #endif

12 **void** **GetMemSta**(**void**);

13

14 */////////////////////////////////////////////////////////////////////////////*

15 *// The one and only application object*

16

17 CWinApp theApp;

18

19 **using namespace** std;

20

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

22 {

23 **int** nRetCode = 0;

24 LPVOID BaseAddr;

25 **char** \* str;

26 **GetMemSta**();

27 **printf**("Now Allocate 32M Virtual Memory and 2M Physical Memory\n\n");

28 BaseAddr = **VirtualAlloc**(NULL,

29 1024\*1024\*32,

30 MEM\_RESERVE|MEM\_COMMIT,

31 PAGE\_READWRITE);

32 **if**(BaseAddr == NULL)**printf**("Virtual Allocate Fail\n");

33 str = (**char**\*)**malloc**(1024\*1024\*2);

34 **GetMemSta**();

35 **printf**("Now Release 32M Virtual Memory and 2M Physical Memory\n\n");

36 **if**(**VirtualFree**(BaseAddr,0,MEM\_RELEASE)==0)

37 **printf**("Release Allocate Fail\n");

38 **free**(str);

39 **GetMemSta**();

40

41 **return** nRetCode;

42 }

43

44 **void** **GetMemSta**(**void**)

45 {

46 MEMORYSTATUS MemInfo;

47 **GlobalMemoryStatus**(&MemInfo);

48 **printf**("Current Memory Status is:\n");

49 **printf**("\t Total Physical Memory is %dMB\n",MemInfo.dwTotalPhys/(1024\*1024));

50 **printf**("\t Available Physical Memory is %dMB\n",MemInfo.dwAvailPhys/(1024\*1024));

51 **printf**("\t Total Page Fail is %dMB\n",MemInfo.dwTotalPageFile/(1024\*1024));

52 **printf**("\t Available Page File is %dMB\n",MemInfo.dwAvailPageFile/(1024\*1024));

53 **printf**("\t Total Virtual memory is %dMB\n",MemInfo.dwTotalVirtual/(1024\*1024));

54 **printf**("\t Available Virtua; memory is %dMB\n",MemInfo.dwAvailVirtual/(1024\*1024));

55 **printf**("\t Memory Load is %d%% \n\n",MemInfo.dwMemoryLoad);

56 }