**操作系统课程设计实验报告**

实验名称： 内存监视

姓名/学号： 张惟振/1120170117

1. **实验目的**

通过实验熟悉Windows存储器管理中提供的各类机制和实现的请求调页技术。

了解Windows内存结构和当前系统中内存的使用情况，包括系统地址空间的布局，物理内存的使用情况。

熟悉使用Windows存储器相关的API。

1. **实验内容**

设计一个内存监视器，能实时地显示当前系统中内存的使用情况，包括系统

地址空间的布局，物理内存的使用情况；能实时显示某个进程的虚拟地址空间 布局和工作集信息等。

1. **实验环境**

1、软件环境

Windows10操作系统，版本号：1909

2、硬件环境

Intel® Core™ i5-7200U CPU @ 2.50GHz×4

1. **程序设计与实现**

1、实验思路

（1）对于查看性能信息的功能，可以使用GetPerformanceInfo（）来实现，该函数得到一个PERFORMANCE\_INFORMATION结构体，保存性能信息。

（2）对于查看内存信息的功能，可以使用GlobalMemoryStatusEx（）来实现，该函数得到一个MEMORYSTATUSEX结构体，包含了物理内存和虚拟内存的现有信息。

（3）对于查看系统信息的功能，可以使用GetSystemInfo（）来实现，该函数得到一个SYSTEM\_INFO结构体，保存有当前计算机系统的信息。

（4）对于查看进程信息的功能，可以先使用CreateToolhelp32Snapshot（）获取当前运行进程的快照，然后使用Process32First（）获取第一个进程的句柄，进而使用Process32Next（）遍历进程快照中所有的进程，即可查看所有进程的信息。

（5）对于查看进程的虚拟地址空间布局功能，可以在对程序的地址空间进行遍历的过程中使用VirtualQueryEx（）检查虚拟内存的当前信息，该函数将查询到的信息存入MEMORY\_BASIC\_INFORMATION结构体中，借此可得到块的地址和大小以及状态、类型等信息。

2、Windows系统实现中的API和数据结构

（1）GetPerformanceInfo（）获取性能信息

PERFORMANCE\_INFORMATION结构体部分信息如下：

typedef struct \_PERFORMANCE\_INFORMATION {

DWORD cb;//大小

SIZE\_T CommitTotal;//系统当前提交的页面总数

SIZE\_T CommitLimit;//系统当前可提交的最大页面数

SIZE\_T CommitPeak;//系统历史提交页面峰值

SIZE\_T PhysicalTotal;//按页分配的总物理内存

SIZE\_T PhysicalAvailable;//当前可用的物理内存

SIZE\_T SystemCache;//系统缓存的容量

SIZE\_T KernelTotal;//内存总量

SIZE\_T KernelPaged;//分页池大小

SIZE\_T KernelNonpaged;//非分页池大小

SIZE\_T PageSize;//页的大小

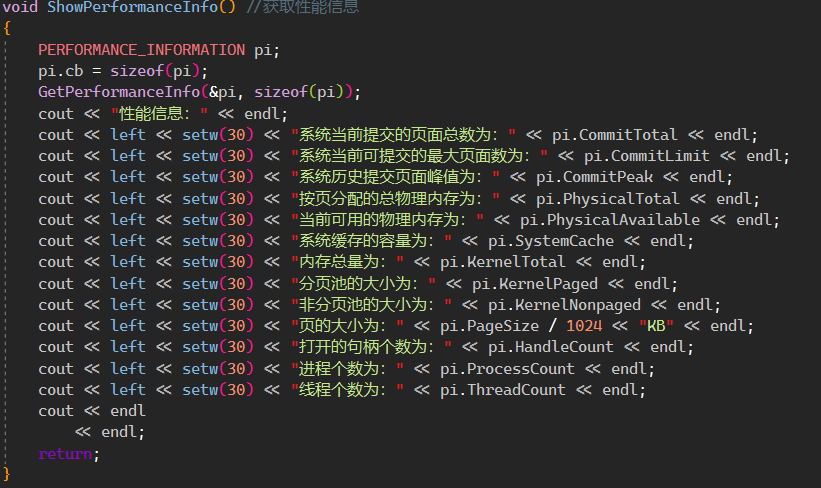
DWORD HandleCount;//打开的句柄个数

DWORD ProcessCount;//进程个数

DWORD ThreadCount;//线程个数

} PERFORMANCE\_INFORMATION, \*PPERFORMANCE\_INFORMATION;

实验使用如下：

 （2）GlobalMemoryStatusEx（）获取内存状态

MEMORYSTATUSEX结构体的部分信息如下：

typedef struct \_MEMORYSTATUSEX {

DWORD dwLength;//MEMORYSTATUSEX结构的大小

DWORD dwMemoryLoad;//内存的使用率

DWORDLONG ullTotalPhys;//内存的总容量

DWORDLONG ullAvailPhys;//可用内存大小

DWORDLONG ullTotalPageFile;//页文件的总容量

DWORDLONG ullAvailPageFile;//可用的页文件大小

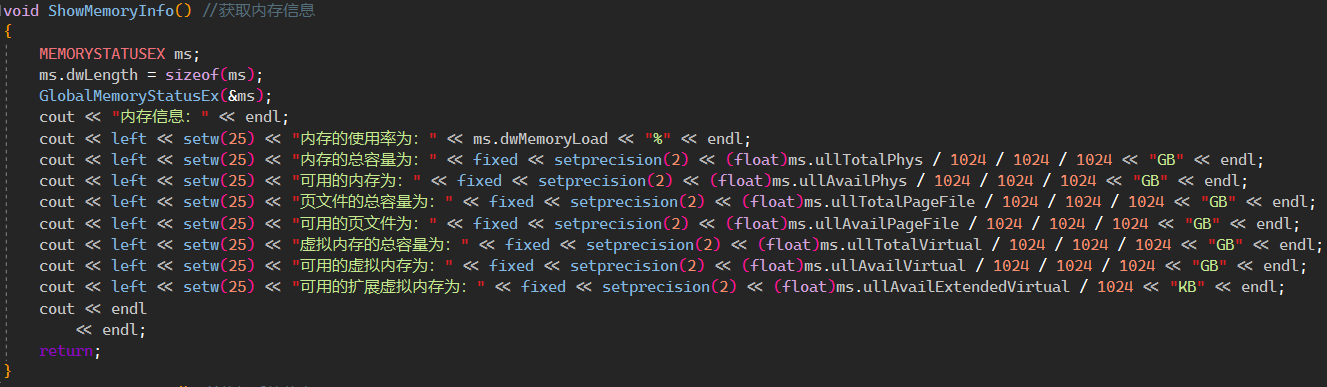
DWORDLONG ullTotalVirtual;//虚拟内存的总容量

DWORDLONG ullAvailVirtual;//可用的虚拟内存大小

DWORDLONG ullAvailExtendedVirtual;//可用的扩展虚拟内存大小

} MEMORYSTATUSEX, \*LPMEMORYSTATUSEX;

实验使用如下：

（3）GetSystemInfo（）获取系统信息

SYSTEM\_INFO结构体的部分信息如下：

typedef struct \_SYSTEM\_INFO {

union {

DWORD dwOemId;

struct {

WORD wProcessorArchitecture;//处理器的体系结构

WORD wReserved;

} DUMMYSTRUCTNAME;

} DUMMYUNIONNAME;

DWORD dwPageSize;//内存页大小

LPVOID lpMinimumApplicationAddress;//每个进程可用地址空间的最小内存地址

LPVOID lpMaximumApplicationAddress;//每个进程可用地址空间的最大内存地址

DWORD\_PTR dwActiveProcessorMask;//处理器掩码

DWORD dwNumberOfProcessors;//处理器数量

DWORD dwProcessorType;//处理器类型

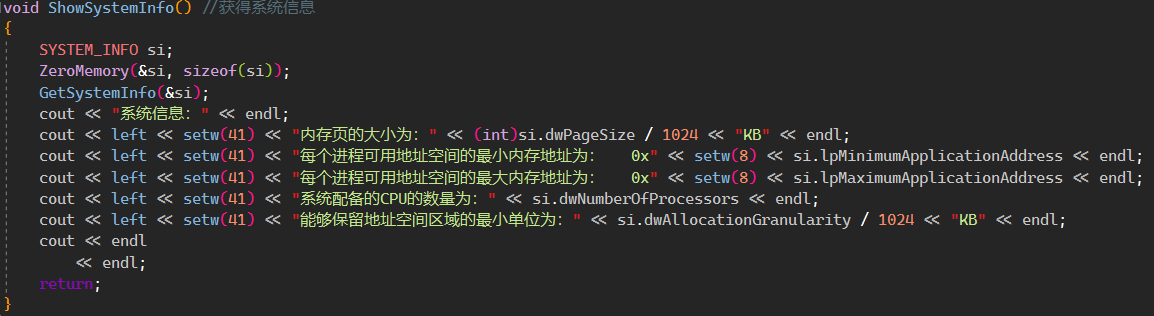
DWORD dwAllocationGranularity;//能够保留地址空间区域的最小单位

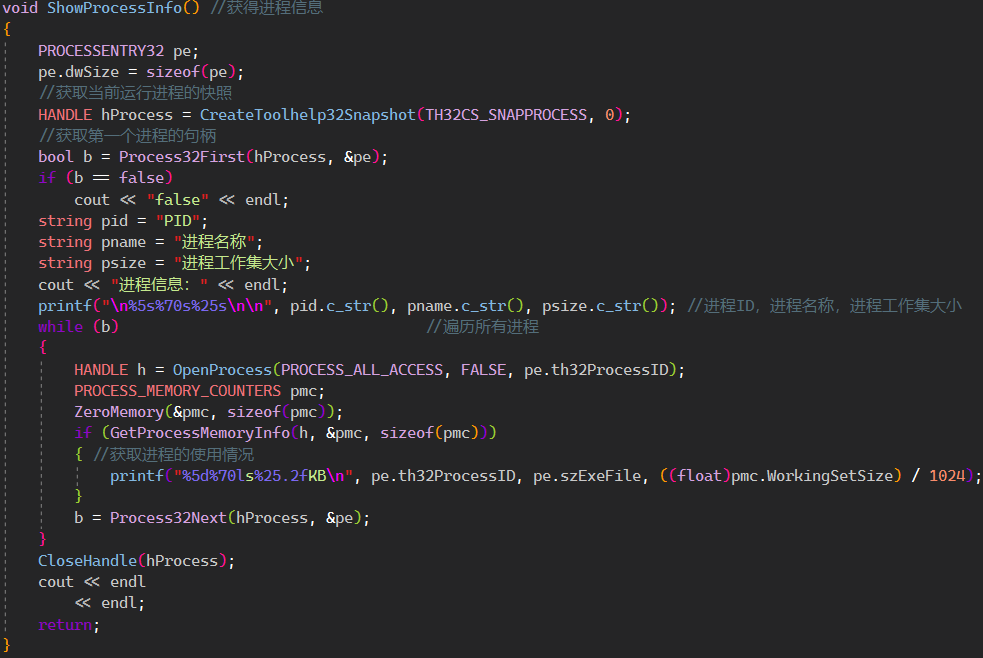
WORD wProcessorLevel;//处理器等级

WORD wProcessorRevision;//处理器版本

} SYSTEM\_INFO, \*LPSYSTEM\_INFO;

实验使用如下：

 （4）CreateToolhelp32Snapshot（）获取进程快照，当获取快照之后可以利用Process32First（）获取其中第一个进程的句柄，再结合Process32Next（）遍历所有进程。

 实验使用如下：

（5）VirtualQueryEx（）检查进程虚拟内存的当前信息，会将查询到的信息存入MEMORY\_BASIC\_INFORMATION结构体中，该结构体的部分信息如下：

typedef struct \_MEMORY\_BASIC\_INFORMATION {

PVOID BaseAddress;

PVOID AllocationBase;

DWORD AllocationProtect;

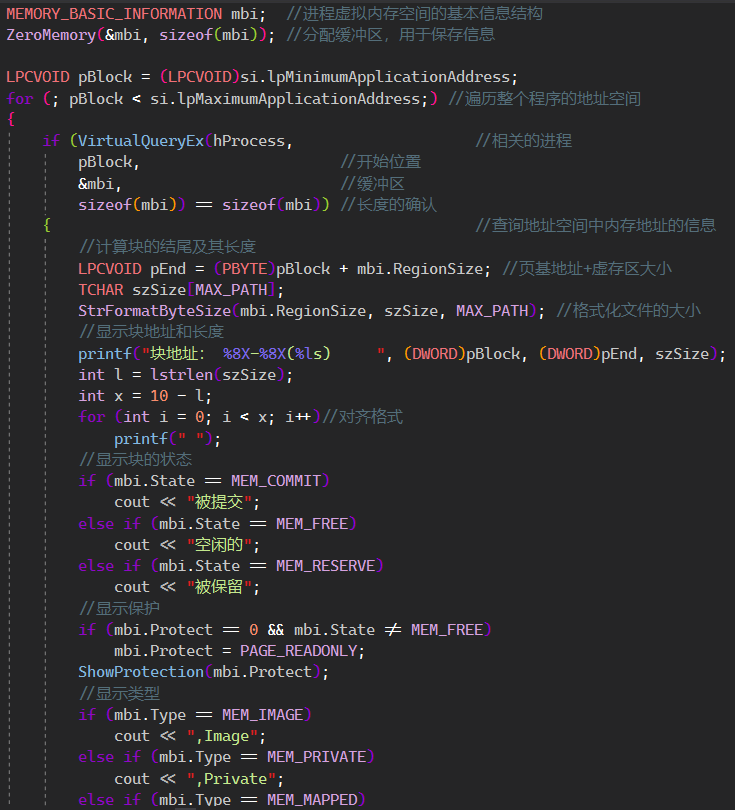
SIZE\_T RegionSize;//从基址开始的虚存区大小

DWORD State;//页面状态

DWORD Protect;//页面的访问保护

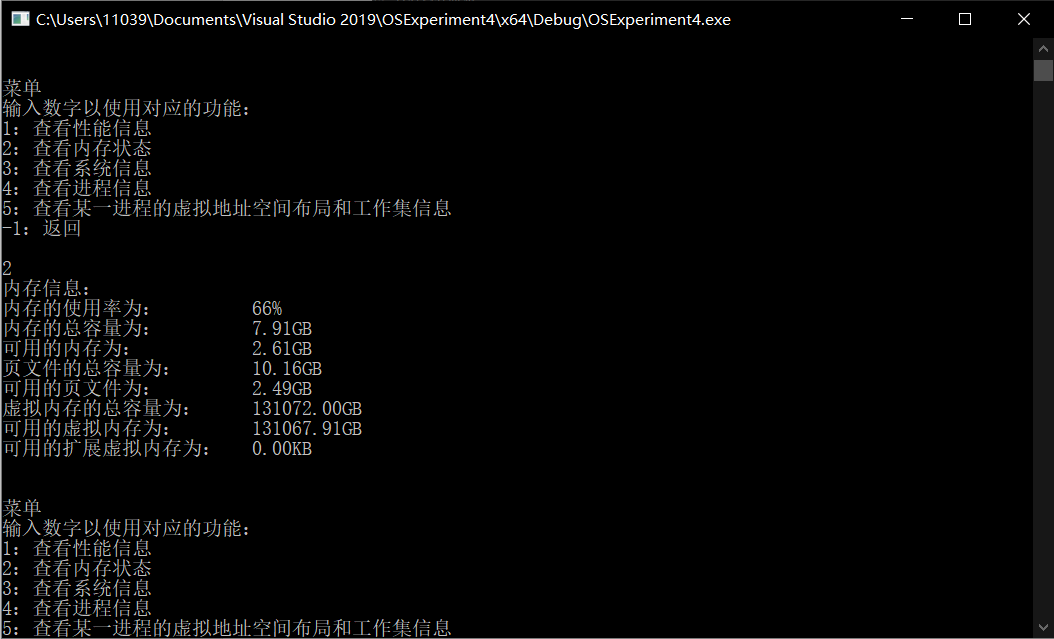
DWORD Type;//页面的类型

} MEMORY\_BASIC\_INFORMATION, \*PMEMORY\_BASIC\_INFORMATION;

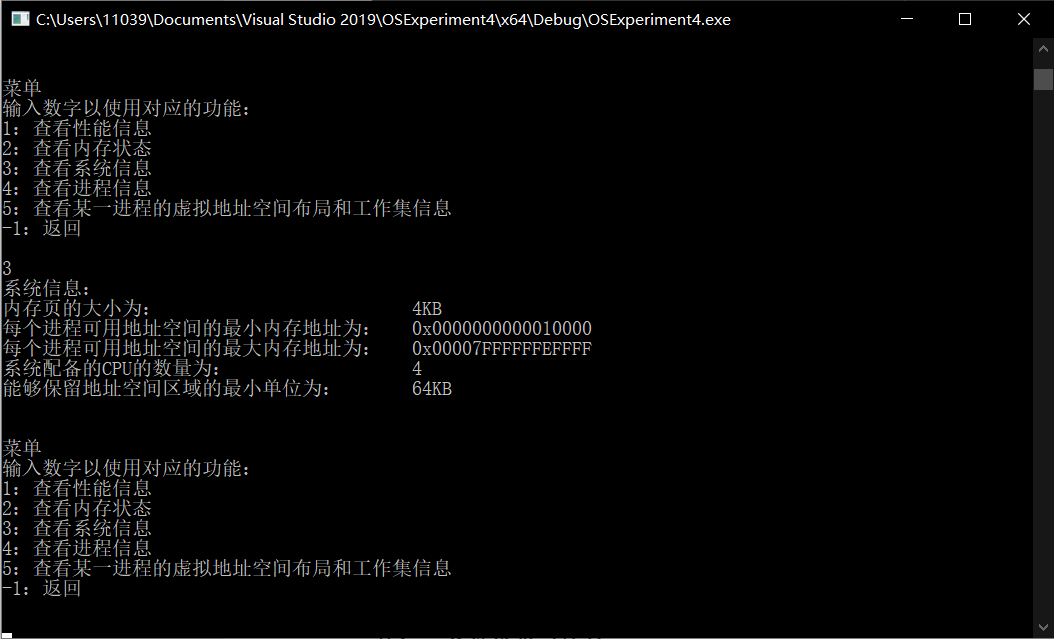
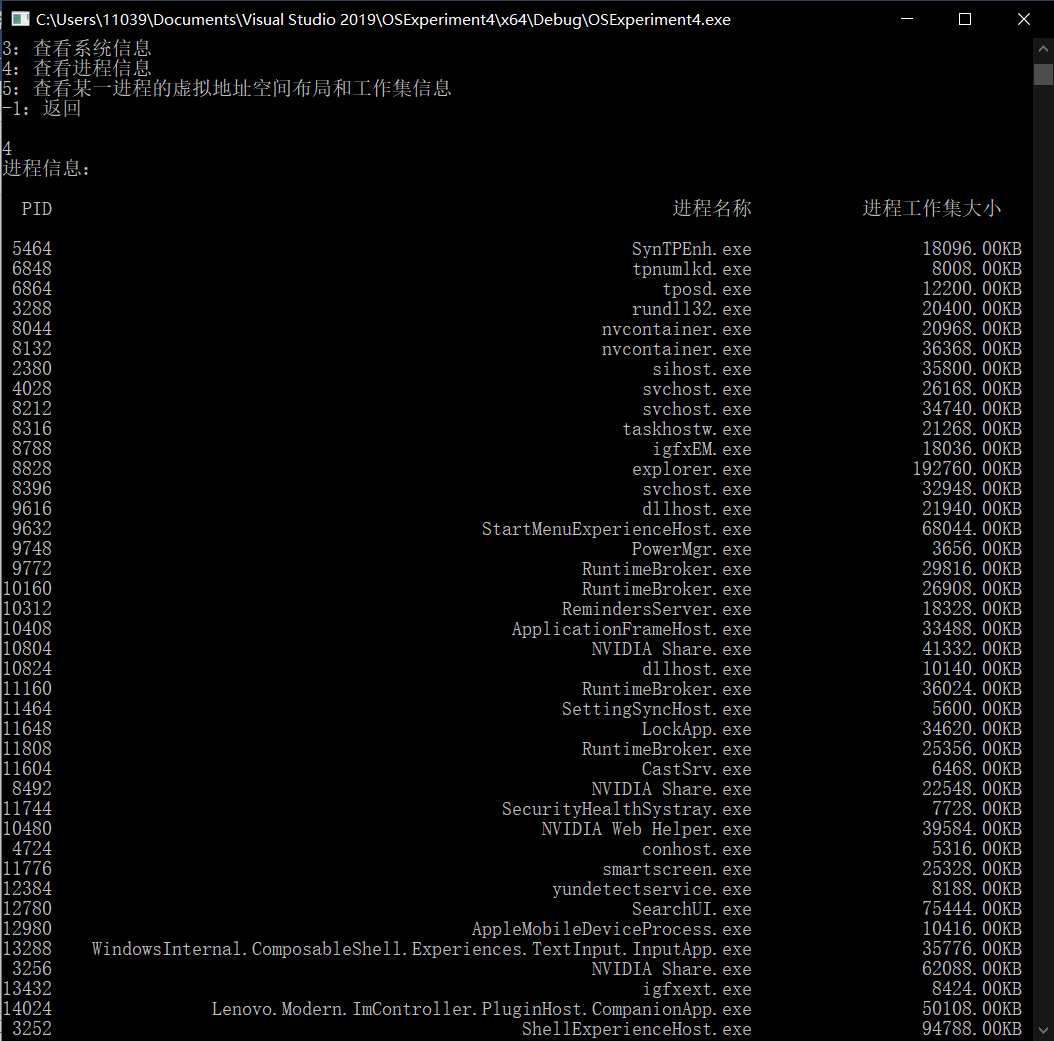
 实验使用如下：

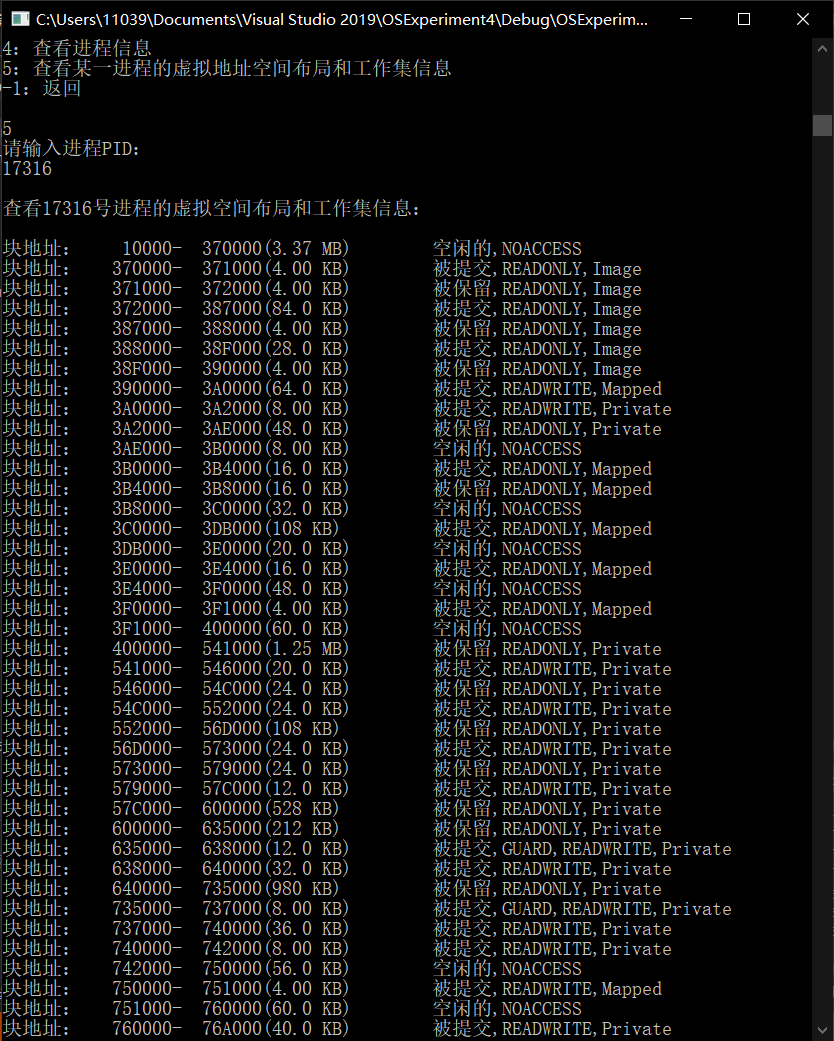
3、实验结果

查看性能信息

 查看内存状态

查看系统信息

 查看进程信息

 查看某一进程的虚拟地址空间布局和工作集信息

1. **实验收获与体会**

这次实验其实说难也不难，说简单也不简单。其中重点内容就是调用Windows系统中的API。在对API进行调用的时候就，需要了解各种参数的意义和使用，需要分别对其进行准确的对应起来，才可以保证API的正确调用,通过这次实验我对获取系统信息的API有了更多的了解。

此外，在这个实验中除了调用API，另外一个重点的内容是需要了解windows中的几个结构体，比如SYSTEM\_INFO、MEMORYSTATUS和PERFORMANCE\_INFORMATION等结构体。在这些结构体中，有很多我们需要的信息，这些信息就是在本实验中需要用到的信息，了解这些系统预定义的结构体时完成这次试验的重要部分。

最后，我对操作系统内存分配方式有了进一步的印象。Windows把每个进程的虚拟内存地址映射到物理内存地址，操作系统通过页式管理的方式对内存进行管理。