实验3 内存监视

班级： 07112002 学号： 1120201316 姓名： 金雅各

一、实验目的

独立设计并实现一个内存监视器，以加深对内存管理的理解。

1. 实验内容

1. 在Windows系统下设计实现一个内存监视器。

2. 使用该内存监视器：

能够实时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；

能够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. 可使用的相关系统调用包括：

GetSystemInfo,

VirtualQueryEx,

VirtualAlloc,

GetPerformanceInfo,

GlobalMemoryStatusEx等。

三、实验步骤

① 学习使用一些与Windows内存监视实验相关的数据结构

SYSTEM\_INFO (保存计算机系统信息)

MEMORYSTATUSEX (保存内存信息)

PERFORMANCE\_INFORMATION (保存性能信息)

PROCESSENTRY32 (保留拍摄快照时驻留在系统中的进程列表)

PROCESS\_MEMORY\_COUNTERS (保存进程内存统计信息)

MEMORY\_BASIC\_INFORMATION (保存进程虚拟内存空间信息)

② 学习使用与进程以及内存管理相关的Windows API

GetSystemInfo (LPSYSTEM\_INFO lpSystemInfo)；

(获取当前系统信息)

lpSystemInfo为指向SYSTEM\_INFO结构体的指针

GlobalMemoryStatusEx (LPMEMORYSTATUSEX lpBuffer);

(获得当前可用的物理和虚拟内存信息)

lpBuffer 为指向MEMORYSTATUSEX 结构的指针

GetPerformanceInfo (

PPERFORMANCE\_INFORMATION pPerformanceInformation,

DWORD cb

);

(获取性能信息)

pPerformanceInformation 为指向PERFORMANCE\_INFORMATION结构的指针

cb 为结构体的大小

CreateToolhelp32Snapshot(

DWORD dwFlags,

DWORD th32ProcessID

);

获得指定进程的快照

dwFlags 包含在快照中的系统部分，TH32CS\_SNAPPROCESS表示包含快照中的所有进程

th32ProcessID 要包含在快照中的进程的标识符

Process32First (

HANDLE hSnapshot,

LPPROCESSENTRY32 lppe

);

获取有关系统快照中遇到的第一额进程的信息

hSnapshot 使用从CreateToolhelp32Snapshot返回的快照句柄

lppe 指向PROCESSENTRY32结构的指针

Process32Next (

HANDLE hSnapshot,

LPPROCESSENTRY32 lppe

);

获取有关系统快照中记录的下一个进程的信息

hSnapshot 使用从CreateToolhelp32Snapshot返回的快照句柄

lppe 指向PROCESSENTRY32结构的指针

OpenProcess (

DWORD dwDesiredAccess,

BOOL bInheritHandle,

DWORD dwProcessId

);

打开现有的本地进程对象

dwDesiredAccess 对进程对象的访问权限

binheritHandle 若此值为TRUE，则该进程创建的进程将继承该句柄。否则，子进程不会继承该句柄

dwProcessId 要打开的本地进程的标识符，在实验中取PROCESSENTRY32结构中的th32ProcessID

GetProcessMemoryInfo (

HANDLE Process,

PPROCESS\_MEMORY\_COUNTERS ppsmemCounters,

DWORD cb

);

获取指定进程的内存使用情况的信息

Process 进程的句柄

ppsmemCounters 指向 PROCESS\_MEMORY\_COUNTERS 结构的指针

cb 为 ppsmemCounters 结构体的大小

VirtualQueryEx (

HANDLE hProcess,

LPCVOID lpAddress,

PMEMORY\_BASIC\_INFORMATION lpBuffer,

DWORD dwLength

);

查询地址空间中内存地址的信息

hProcess 进程句柄。

lpAddress 查询内存的地址。

lpBuffer 指向MEMORY\_BASIC\_INFORMATION结构的指针，用于接收内存信息。

dwLength MEMORY\_BASIC\_INFORMATION结构的大小。

VirtualAlloc (

LPVOID lpAddress, // 要分配的内存区域的地址

DWORD dwSize, // 分配的大小

DWORD flAllocationType, // 分配的类型

DWORD flProtect // 该内存的初始保护属性

);

该函数的功能是在调用进程的虚地址空间,预定或者提交一部分页

③ 编写showSystemInfo函数用以显示当前系统信息

数据结构: SYSTEM\_INFO

API: GetSystemInfo

④ 编写showMemoryStatus函数用以显示系统物理与虚拟内存信息

数据结构: MEMORYSTATUSEX

Api: GlobalMemoryStatusEx

⑤ 编写 showSystemPerformence 函数用以显示系统性能

数据结构: PERFORMANCE\_INFORMATION

API: GetPerformanceInfo

⑥ 编写 showProcessInfo 函数显示各个进程的信息

数据结构: PROCESSENTRY32、PROCESS\_MEMORY\_COUNTERS

API: CreateToolhelp32Snapshot、Process32First、OpenProcess、GetProcessMemoryInfo、Process32Next

⑦ 编写 showVirtualProcessMemory 函数显示某个进程的虚存使用信息

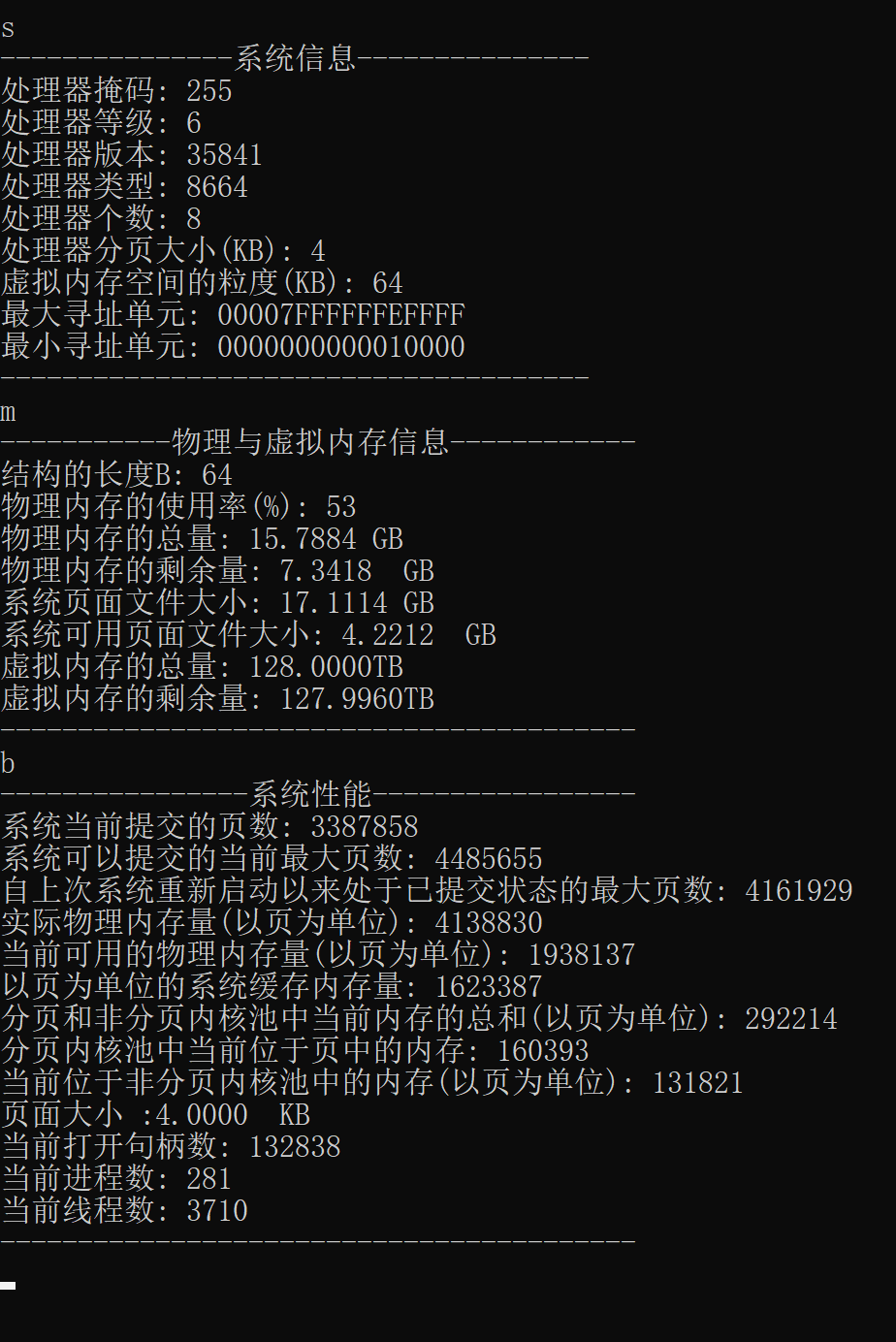
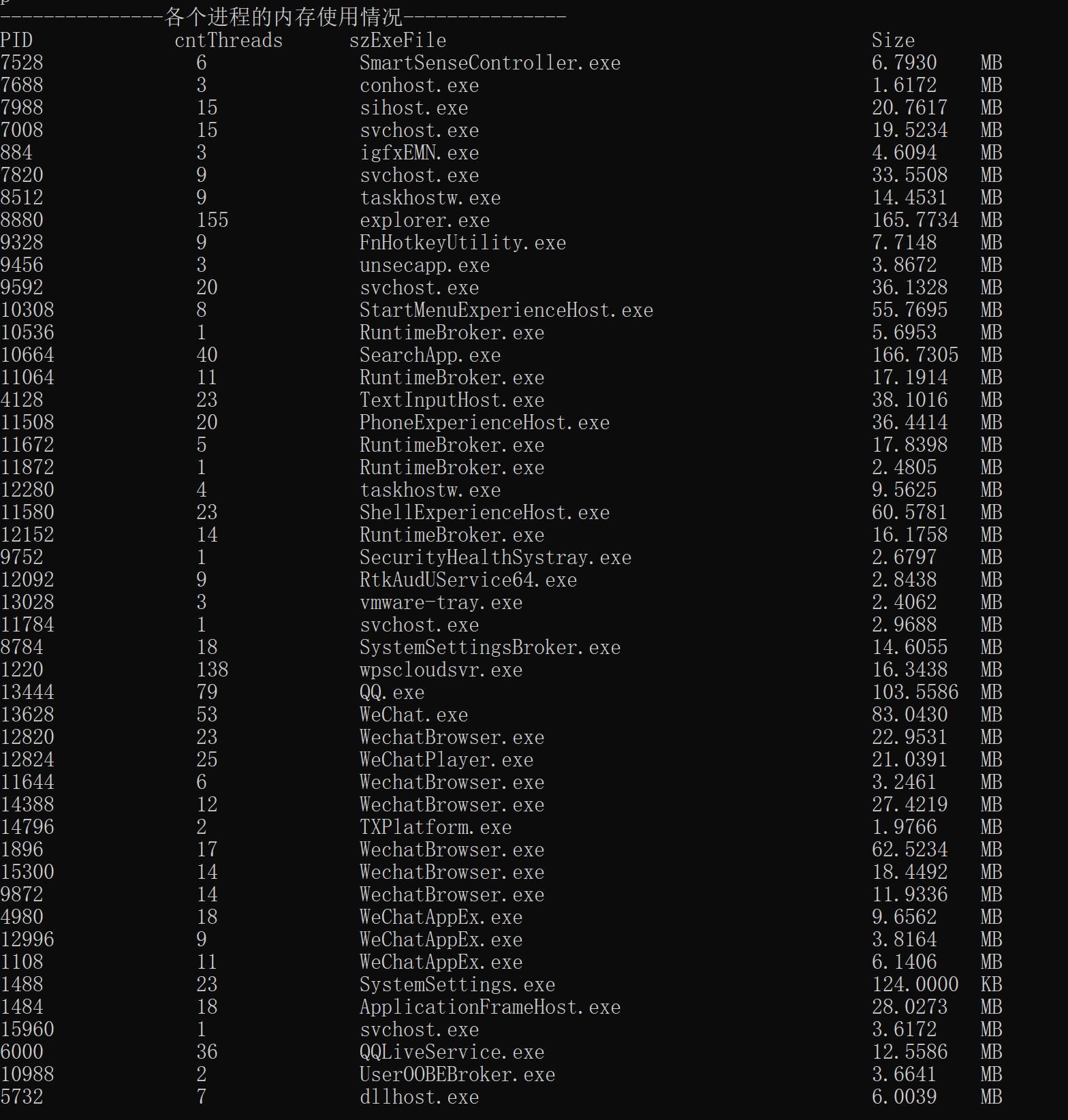
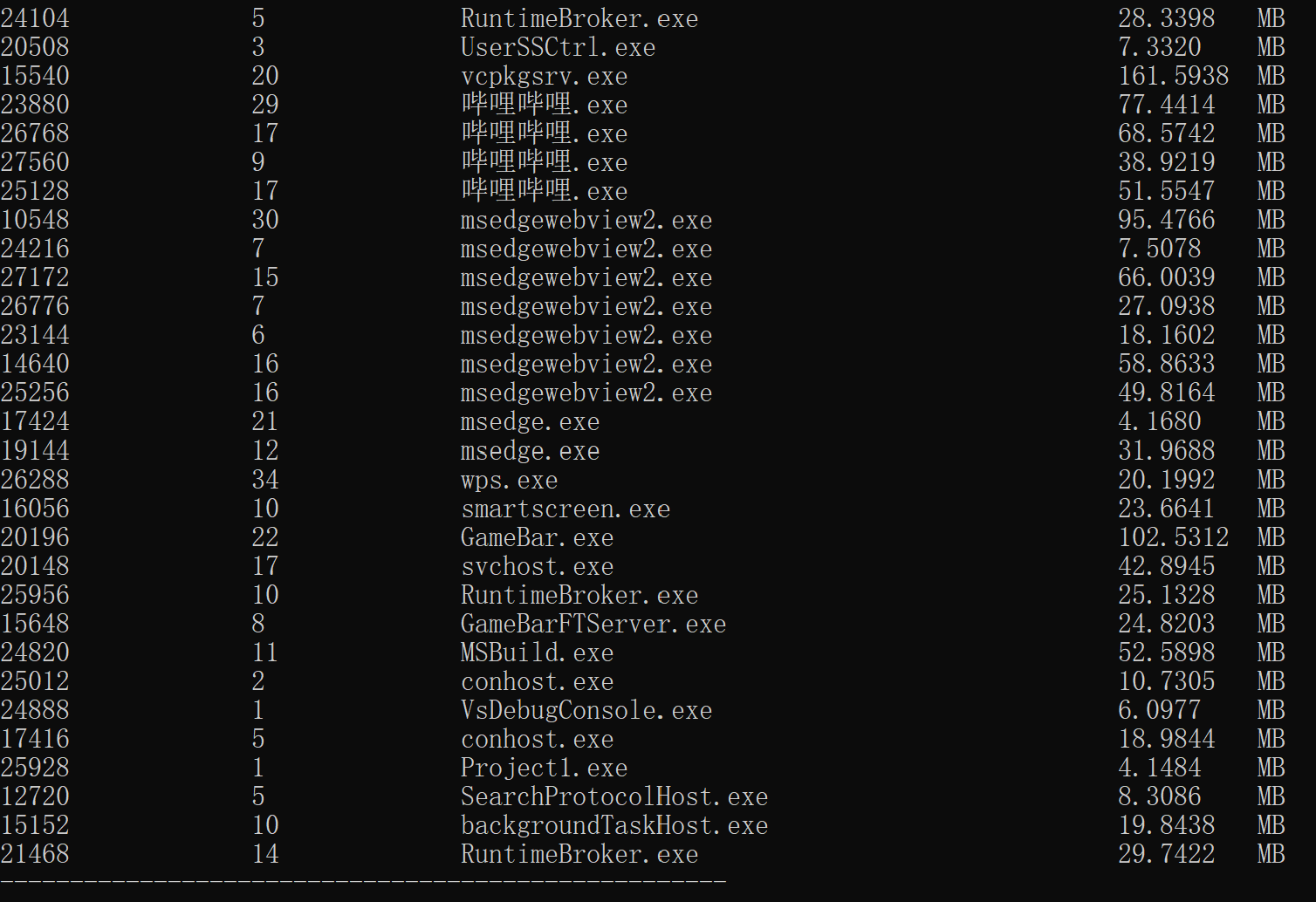
数据结构: MEMORY\_BASIC\_INFORMATION、SYSTEM\_INFO、

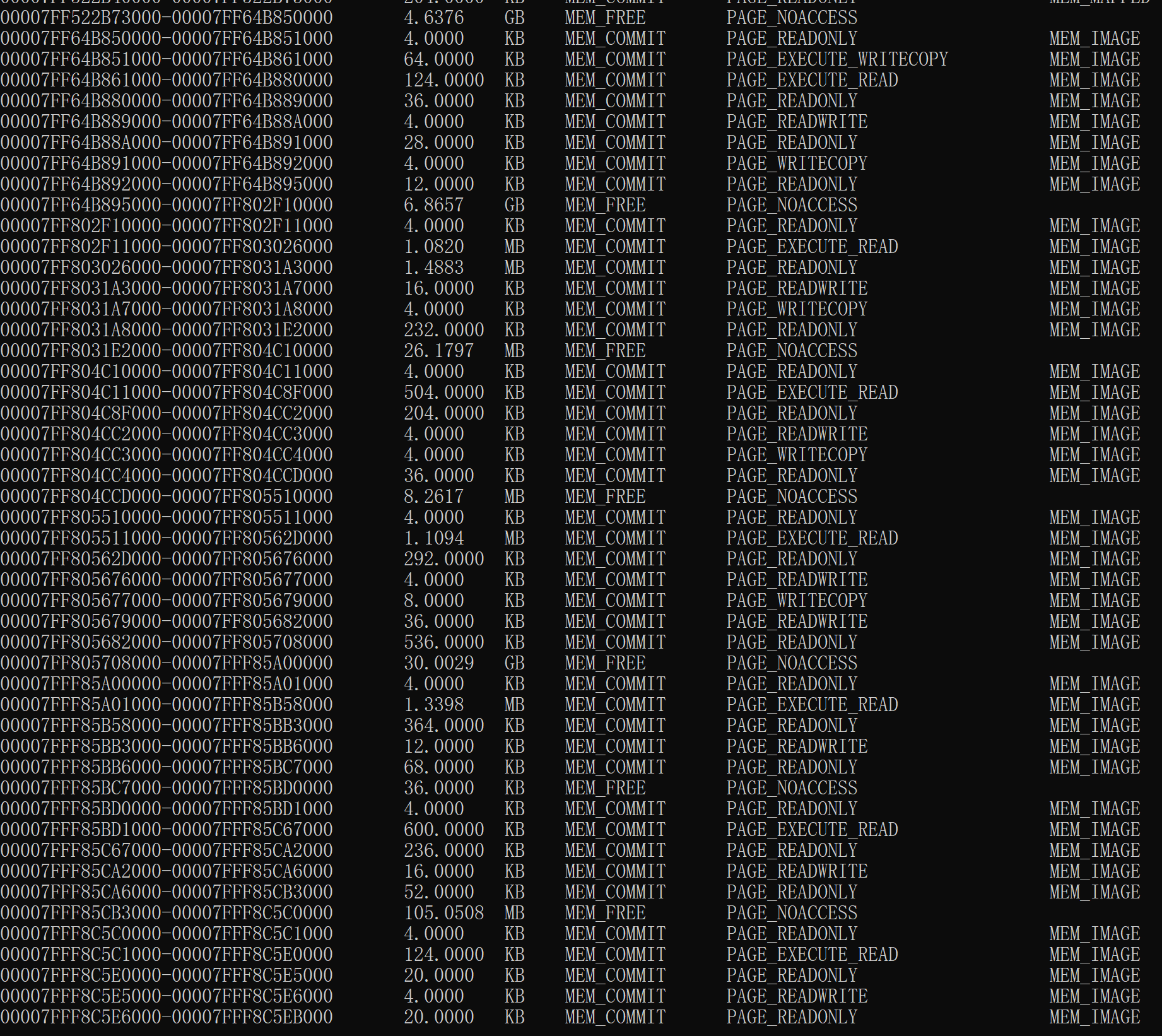
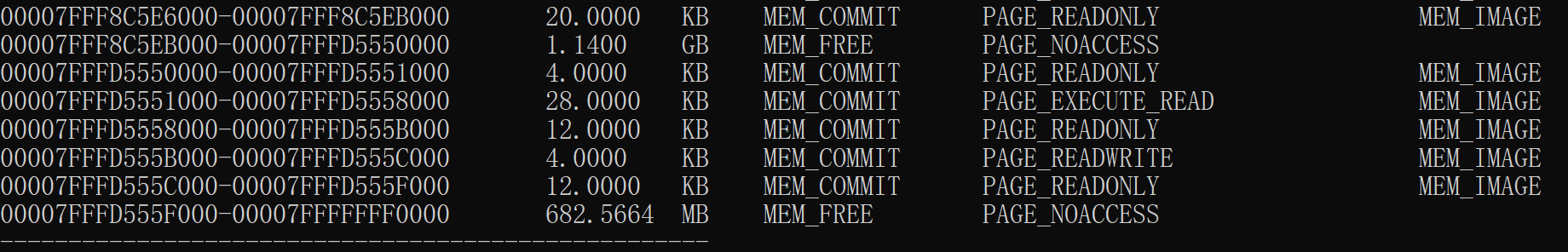
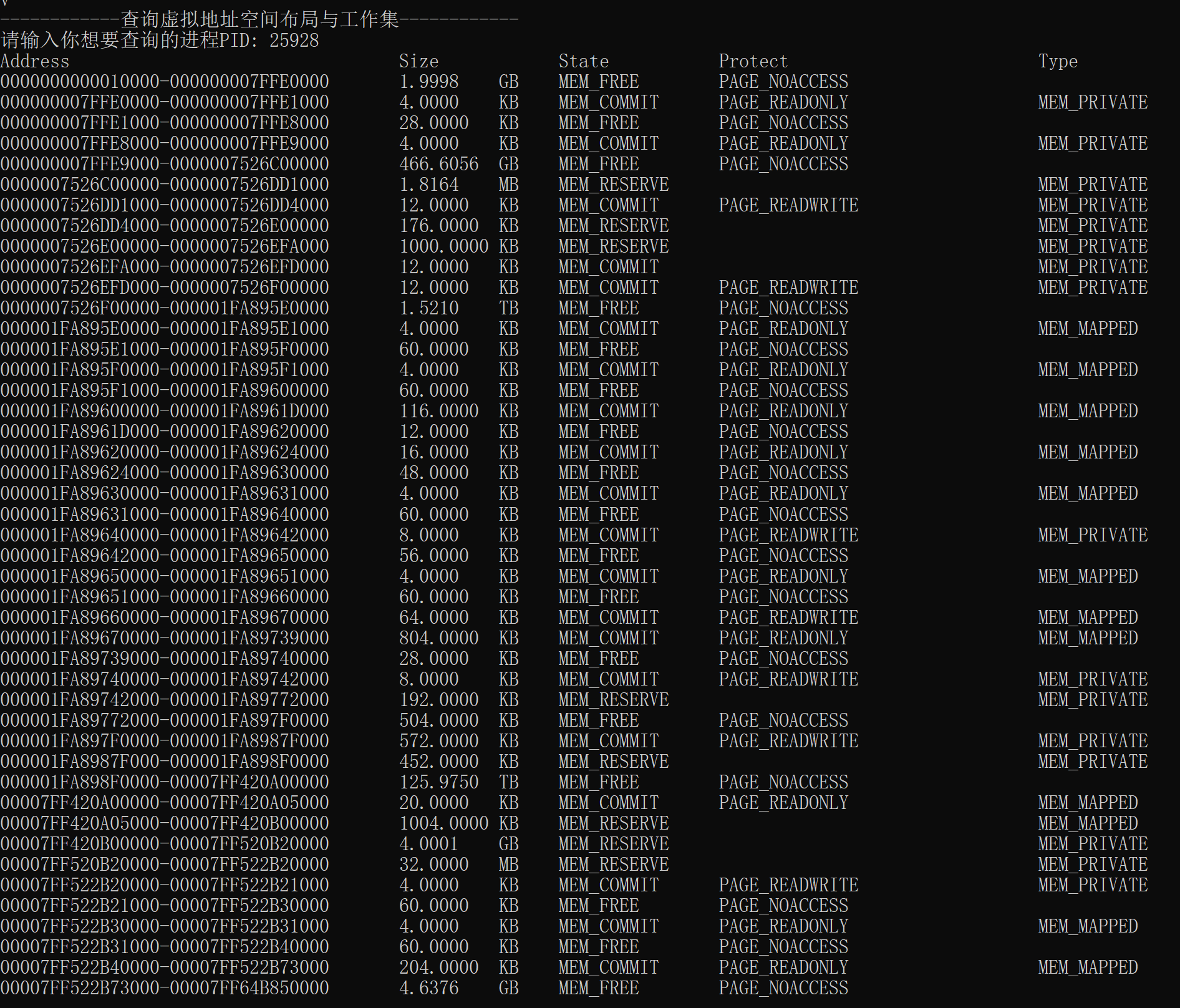
API: OpenProcess、GetSystemInfo、VirtualQueryEx

⑧ 编写一些辅助工具函数用以输出和提示输入

⑨ 对项目进行调试

四、实验结果及分析





结构显示：

分页大小为：4KB

寻址从0000000000010000到 00007FFFFFFEFFFF

系统进程数为：256

线程数为：3525

物理内存大小为：15.7887 GB

虚存大小为：128TB

选择查看正在运行的进程 SourceCode.exe （即本项目）查看进程的虚存信息

SourceCode.exe进程的虚拟地址从0000000000010000到00007FFFFFFF0000

占据了系统的整个虚拟地址空间，工作集的状态有

MEM\_COMMIT 指明已分配物理内存或者系统页文件

MEM\_FREE 空闲状态。该区域的虚拟地址不受任何内存的支持。

MEM\_RESERVE 指明页面被保留，但是没有分配任何物理内存。该状态下Protect成员未定（没有输出）。

类型有

MEM\_IMAGE 明该区域的虚拟地址原先受内存映射的映像文件（如.exe或DLL文件）的支持，但也许不再受映像文件的支持。

MEM\_MAPPED 该区域的虚拟地址原先是受内存映射的数据文件的支持，但也许不再受数据文件的支持。

MEM\_PRIVATE 指明该内存区域是私有的。不被其他进程共享。

Protect 属性 包括只读、读写、只允许执行、允许执行和读写等等

1. 实验收获与体会

① 通过Windows内存监视实验，掌握了许多在Windows下关于内存信息的系统调用函数，也熟悉了一些与内存信息相关的数据结构，同时也对系统内部内存的工作方式和工作情况有了一定的了解

②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。

附录：程序清单及说明

源代码在文件夹SourceCode，请用Visual打开文件夹并运行，可选择x64或x86环境运行；

输入命令时，如输出其他未定义的命令，程序会输出提示符，帮助你输入正确的命令