**实验四 进程监视**

班级：07121502 学号：1320151098 姓名：赵璐

**1.实验目的**

通过本实验，熟悉Windows的存储器管理提供的各种机制，了解Windows的内存结构和虚拟内存的管理，学习如何在应用程序中管理内存体会Windows应用程序使用内存的简单性。

**2.实验内容**

在Windows中设计一个内存监视器，需要实现以下功能：

1. 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等。
2. 能通过进程的PID查询该程序的虚拟地址空间布局和工作集信息等。

**3.实验环境**

处理器：Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz

RAM：4.00 GB

系统类型：64位

实验环境：Visual Studio 2017

**4.程序设计及实现**

**4.1实验基本思路**

(1)先使用系统API函数GetSystemInfo()和GlobalMemoryStatusEx()获得系统的基本信息，包括内存的使用情况、系统地址空间布局和物理内存的使用情况等。

(2)使用API函数CreatToolhelp32Snapshot()获取系统快照，从而获得系统中正在运行的进程，并打印进程的名字和PID。

(3)由OpenProcess()函数通过进程PID得到该进程的句柄。

(4)使用VirtualQueryEx()遍历用户查询的进程的地址空间

**4.2Windows调用系统API及数据结构说明**

(1)GetSystemInfo()

功能：返回关于当前系统的信息。

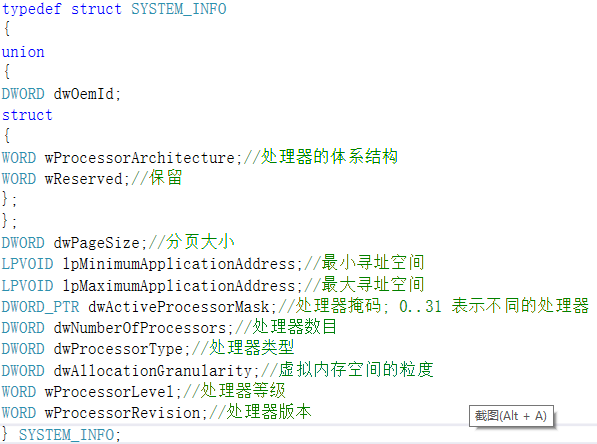
原型：void WINAPI GetSystemInfo

(

\_Out\_ LPSYSTEM\_INFO IpSystemInfo

);

数据结构：



(2)GlobalMemoryStatusEx()

功能：用于获取系统内存信息

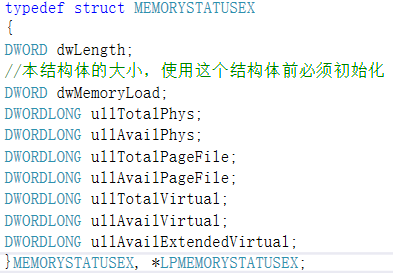
原型：bool WINAPI GlobalMemoryStatusEx

(

\_inout LPMEMORYSTATUSEX IpBuffer

)

数据结构：



(3)CreateToolhelp32Snapsh()

功能：建立系统快照，获取系统中正在运行的进程信息

原型：HANDLE WINAPI CreateToolhelp32Snapshot

(

//指定“快照”中需要返回的对象，可以是TH32CS\_SNAPPROCESS等

DWORD dwFlags,

//一个进程ID号，当获取系统进程列表时可以设为0

DWORD th32ProcessID

)

(4)Process32Fist()/Process32Next()

功能：获得快照后，用来得到第一个/下一个进程的句柄

原型：bool WINAPI Process32First

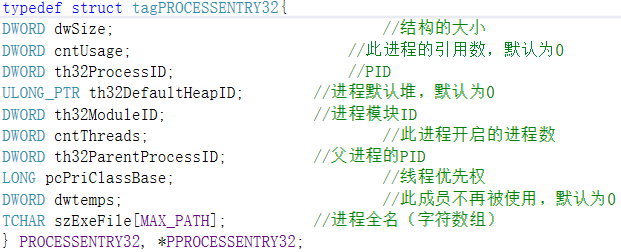
(

HANDLE hSnapshot,

LPPROCESSENTRY32 Ippe

)

数据结构：



(5)VirtualQueryEx()

功能：查询地址空间中内存地址的信息

原型：DWOED VirtualQueryEx

(

HANDLE hProcess, //进程句柄

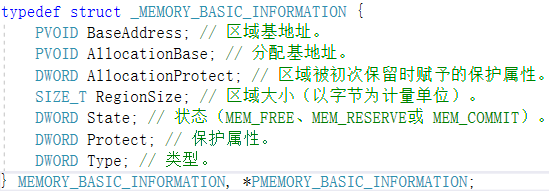
LPCVOID IpAddress, //查询内存的地址

PMEMORY\_BASIC\_INFORMATION IpBuffer, //指针，用于接收内存信息

DWORD deLength //结构的大小

)

数据结构：



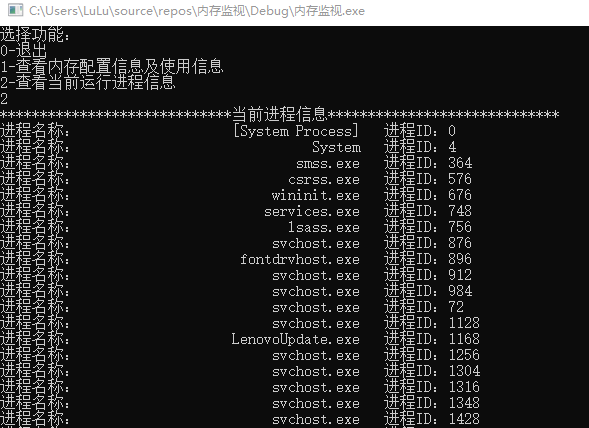
**5.实验结果**

功能一：选择功能1，查看内存配置信息及使用信息



功能二：选择功能2，首先显示系统正在运行的进程名称及其PID

由用户输入想要查看进程的PID，显示该进程信息





**6.心得体会**

实现实验四的主要方法就是掌握并正确使用需要用到的各个WINAPI函数，了解函数中用到的各种结构体。因为函数及结构体是系统预定义的，所以我也不必自定义结构体，更不必掌握函数的实现过程，所以这对于编程人员来说是非常简单快捷的。

通过本实验，我切实了解了Windows中进程的地址空间，知道了如何利用操作系统的虚拟内存机制来增强程序对内存的管理能力。