实验名称	实验 3：内存监视		
学号	1120220576	姓名	宋宇翔

1. 实验目的

了解 Windows 的内存查看、分配机制以及物理内存使用情况，学习 Windows 系统的 API。

二、实验内容

Windows 设计一个内存监视器，要求：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；实时显示本进程控制的虚拟地址空间布局和工作集信息。

相关的系统调用：GetSystemInfo, VirtualQueryEx,VirtualAlloc, GetPerformanceInfo, GlobalMemoryStatusEx ...

三、实验环境及配置方法

Windows11 系统。

四、实验方法和实验步骤（程序设计与实现）

首先，想要获取系统的内存基本信息（地址空间，物理内存使用情况等），需要如下的代码块：

```
2.  SYSTEM_INFO system_info;
3.  GetSystemInfo(&system_info);
4.  _PERFORMANCE_INFORMATION performance_info;
5.  GetPerformanceInfo(&performance_info, sizeof(performance_info));
6.  cout << "Page Size:" << system_info.dwPageSize << "B    Maximun Address:" << syste
    m_info.lpMaximumApplicationAddress << "    Minimun Address" << system_info.lpMinim
    umApplicationAddress << endl;
7.
8.  cout << "Available Memory:" << (double)(performance_info.PhysicalAvailable * page_
    size) / (double)(1024 * 1024 * 1024) << "GB    Total Memory:" << (double)(performa
    nce_info.PhysicalTotal * page_size) / (double)(1024 * 1024 * 1024) << "GB    Memor
    y Usage:" << (1 - ((double)performance_info.PhysicalAvailable / (double)performanc
    e_info.PhysicalTotal)) * 100 << "%    System Usage:" << performance_info.SystemCac
    he << endl;
```

其中 **performance_info.PhysicalAvailable** 可获取系统可用的物理内存空间的大小，**performance_info.PhysicalTotal** 将返回系统的物理内存总大小，

system_info.lpMaximumApplicationAddress 负责返回物理内存空间的最大地址，最小地址同理，而系统对内存的使用率可通过可用内存大小和总内存大小计算得到，最后，系统划分的页面大小可过 system_info.dwPageSize 查看。

本进程所控制的内存的页面以及虚拟内存的相关信息，需要通过输入的进程名查询该进程的信息，具体代码如下。

```
do{
        //如果要找指定进程的话，使用 strncpy 比较一下可执行文件即可找到进程 id，再通过进程 id 打开进程对象，再通过进程对象访问对应进程的各种详细信息
        if (strncmp(target_processname, now_process.szExeFile, MAX_NAME_SIZE) == 0)
        {
            pid = now_process.th32ProcessID;
        }

} while (Process32Next(hProcessesShot, &now_process));
```

查询到该进程后，通过如下的代码输出进程占用内存的基本信息：

```
cout << "Base Address of The Page: " << memory_basic_information.BaseAddress << endl << "Starting Address of Allocated Memory: " << memory_basic_information.AllocationBase << endl;
```

通过 VirtualAlloc 分配后返回的一块地址，通常和 BaseAddress 一样，但是如果对这块内存中的某一段进行了修改（比如改变中间某一段的保护属性），则一个 VirtualAlloc 分配的地址会被拆成许多个 BaseAddress 小块。

五、实验结果和分析

在 Windows 系统终端用 g++编译器进行编译后运行：

```
C:\Users\syuxi\Desktop\OS\3\代码\memory.exe
Page Size:4096B
Maximun Address:0x7fffffffffff    Minimun Address0x10000
Process Number:312    Thread Number:5344    Sentence Number:142826
Available Memory:6.67277GB    Total Memory:15.6731GB    Memory Usage:57.4253%    System Usage:1619370
Type exit to exit the program, or input memory.exe to check the current status of the process:
```

输出了系统的基本信息，如地址空间，进程线程数等。

输入 memory.exe 查看本进程信息：

```
Base Address of The Page: 0x7ffc7b8a7000
Starting Address of Allocated Memory: 0
Page Status: Released
Page Size:15107133440 B
Protection Type: Undefined
Page Type: Undefined
Search Completed
Process memory.exe (pid 28244)    Mininum Work Set:200KB    Maximun Work Set:1380KB
```

得到了该进程占用的（多个）页面的起始与终止地址，以及占用虚拟地址空间和

工作集。

六、讨论、心得

通过此次实验，我了解到大量关于Windows系统的程序调用接口的相关知识，同时也对计算机操作系统的内存分配环节有了一定的了解，对其具体实现方法有了进一步认识。

过程中也多次出现程序输出乱码、地址格式不匹配等问题，通过转换为全英文输出，地址用DWORD表示并用cout输出的方式得以解决。