操作系统实验报告

实验四 内存监视

2019-3-31

北京理工大学 计算机学院

谭超 1120161874

操作系统课程设计实验报告

实验名称：　　内存监视

姓名/学号：　　谭超　1120161874

一、　实验目的

熟悉 Windows 的存储器管理提供的各种机制和实现的请求调页和群集技术。了解 Windows 系统内存结构和虚拟内存的管理。

二、　实验内容

设计一个内存监视器，能实时的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。相关的系统调用有：GetSystemInfo,VirtualQueryEx,GetPerformenceInfo,GlobalMemoryStatusEx 等。

三、　实验环境

Windows10 1803

处理器 Intel Core i5-6300HQ 2.3GHz ,RAM 8G

四、　程序设计与实现

首先通过 GetSystemInfo()获得系统的相关信息，包括页面的大小、每进程可用地址空间的最小内存地址、每个进程可用的私有地址空间的最大内存地址以及用于预定地址空间区域的分配粒度

```
SYSTEM_INFO si;
ZeroMemory(&si, sizeof(si));
GetSystemInfo(&si);
printf("--------------------系统信息-----------------------\n");
printf("内存页的大小为：%dKB.\n", (int)si.dwPageSize / 1024);
cout << "每个进程可用地址空间的最小内存地址为：0x" << si.lpMinimumApplicationAddress << endl;
cout << "每个进程可用的私有地址空间的最大内存地址为：0x" << si.lpMaximumApplicationAddress << endl;
cout << "用于预定地址空间区域的分配粒度为：" << si.dwAllocationGranularity / 1024 << "KB" << endl;
printf("-----------------------------------------------\n");
```

```
----------------------系统信息-----------------------------
内存页的大小为：4KB.
每个进程可用地址空间的最小内存地址为：0x00010000
每个进程可用的私有地址空间的最大内存地址为：0x7FFEFFFF
用于预定地址空间区域的分配粒度为：64KB
-----------------------------------------------------------
```

然后通过 GetPerformanceInfo()获取系统的性能信息，包括总的物理内存大小、可用的物理内存大小、缓存的大小、页的大小、当前打开的句柄的数量、当前进程的数量与线程的数量

```
//获取系统的性能信息
PERFORMANCE_INFORMATION pi;
pi.cb = sizeof(pi);
GetPerformanceInfo(&pi, sizeof(pi));
printf("----------------系统的存储器使用情况------------------\n");
cout << "总物理内存大小: " << pi.PhysicalTotal<<"页" << endl;
cout << "可用物理内存大小: " << pi.PhysicalAvailable <<"页"<< endl;
cout << "系统Cache的容量为: " << pi.SystemCache<<"页" << endl;
cout << "页的大小为: " << pi.PageSize <<"字节"<< endl;
cout << "当前打开的句柄个数为: " << pi.HandleCount << endl;
cout << "当前进程个数为: " << pi.ProcessCount << endl;
cout << "当前线程个数为: " << pi.ThreadCount << endl;
printf("-----------------------------------------------------\n");
```

```
----------------系统的存储器使用情况------------------
总物理内存大小: 2044667页
可用物理内存大小: 349470页
系统Cache的容量为: 358691页
页的大小为: 4096字节
当前打开的句柄个数为: 170812
当前进程个数为: 266
当前线程个数为: 4397
-----------------------------------------------------
```

再通过 GlobalMemoryStatusEx()获取系统的内存信息，包括物理内存的使用率、物理内存的总容量、可用的物理内存、总的页面大小、当前进程可用的最大内存、当前进程最大内存寻址地址、当前进程可用最大内存

```
//获得系统内存信息
MEMORYSTATUSEX statex;
statex.dwLength = sizeof(statex);
GlobalMemoryStatusEx(&statex);
printf("--------------内存信息----------------\n");
printf("物理内存的使用率为:%1d%%\n", statex.dwMemoryLoad);
printf("物理内存的总容量为: %.2fGB.\n", (float)statex.ullTotalPhys / 1024 / 1024 / 1024);
printf("可用的物理内存为: %.2fGB.\n", (float)statex.ullAvailPhys / 1024 / 1024 / 1024);
printf("总的页面大小为:%.2fGB.\n", (float)statex.ullTotalPageFile / 1024 / 1024 / 1024);
printf("当前进程可用的最大内存数为: %.2fGB.\n", (float)statex.ullAvailPageFile / 1024 / 1024 / 1024);
printf("当前进程最大内存寻址地址: %.2fGB.\n", (float)statex.ullTotalVirtual / 1024 / 1024 / 1024);
printf("当前进程可用最大内存: %.2fGB.\n", (float)statex.ullAvailVirtual / 1024 / 1024 / 1024);
printf("-----------------------------------------------------\n");
printf("-------进程虚拟地址空间布局和工作集信息查询---------\n");
```

```
----------------内存信息-----------------
物理内存的使用率为:82%
物理内存的总容量为: 7.80GB.
可用的物理内存为: 1.33GB.
总的页面大小为:13.30GB.
当前进程可用的最大内存数为: 1.79GB.
当前进程最大内存寻址地址: 2.00GB.
当前进程可用最大内存: 1.95GB.
-----------------------------------------------
```

最后再利用 VirtualQueryEx()获得某个进程的虚拟地址空间布局和工作集信息。输入需要查询的进程 ID，通过 OpenProcess()函数获得该进程的句柄，然后通过 VirtualQueryEx()循环查询该进程的所有虚拟内存块。当输入的进程 ID 小于 0 时，程序结束，当输入的进程 ID 大于 0 时，输出该进程的虚拟地址空间布局。

```
printf("-------进程虚拟地址空间布局和工作集信息查询---------\n");
int pid;
while (true)
{
    printf("输入要查询的进程ID:");
    cin >> pid;
    if (pid <0) break;
    HANDLE hP = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid);
    WalkVM(hP);

}
```

```
-------进程虚拟地址空间布局和工作集信息查询---------
输入要查询的进程ID:988
00010000-7ffe0000(1.99 GB)空闲,NOACCESS
7ffe0000-7ffe1000(4.00 KB)已提交,READONLY, Private
7ffe1000-ffff0000(2.00 GB)空闲,NOACCESS
输入要查询的进程ID:
```

五、 实验收获与体会

本次实验主要是通过调用 Windows 存储器管理相关的 API 对 Windows 的内存机制以及虚拟内存的管理有一个深入的了解。通过本次实验，我理解了页式存储管理的大致结构、物理内存和虚拟内存之间的联系与区别以及这种机制的作用。另外，一个进程并不是所有的页在物理内存中的存储都是连续的，而是分散的。