

操作系统课程设计实验报告

实验名称	内存监视		
学号	1120191600	姓名	张驰

一、实验目的

了解 Windows 系统中的相关的内存监视工作，了解相关的 API，了解在 Windows 系统上显示进程信息的方法。

二、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。Windows 把每个进程的虚拟内存地址映射为物理内存地址。

三、实验环境及配置方法

Windows10

VMware workstation pro

Ubuntu18

四、实验方法和实验步骤（程序设计与实现）

1. 相关 API 及其调用：

调用 windows 中的各种 API 函数实现要求的功能：展示内存信息，展示系统信息，展示性能信息，展示进程信息，展示相应进程的虚拟内存信息。相关的 API 描述如下所示：

(1) 显示当前系统内存使用情况的函数：

```
BOOL GlobalMemoryStatusEx(  
    LPMEMORYSTATUSEX lpBuffer  
);
```

该函数检索系统物理和虚拟内存中当前使用情况的信息，需要接受一个 MEMORYSTATUSEX 结构的指针，将内存信息存放在该结构体中。

```
MEMORYSTATUSEX statex;  
statex.dwLength = sizeof(statex);  
GlobalMemoryStatusEx(&statex);
```

(2) 显示当前系统信息情况的函数：

```
void GetSystemInfo(  
    LPSYSTEM_INFO lpSystemInfo
```

操作系统课程设计实验报告

```
);
```

获取当前系统的信息，接受一个 SYSTEM_INFO 结构体指针。

```
SYSTEM_INFO si;  
ZeroMemory(&si, sizeof(si)); //用 0 来填充该内存区域  
GetSystemInfo(&si);
```

(3) 显示当前性能信息情况的函数：

```
BOOL GetPerformanceInfo(  
    PPERFORMANCE_INFORMATION pPerformanceInformation,  
    DWORD cb  
);
```

获取当前性能信息情况的函数，需要一个 PERFORMANCE_INFORMATION 结构的指针和其结构体的大小。

```
PERFORMANCE_INFORMATION pi;  
pi.cb = sizeof(pi);  
GetPerformanceInfo(&pi, sizeof(pi));
```

(4) 显示所有进程的相关函数：

使用 CreateToolhelp32Snapshot 函数获取指定进程的快照，以及这些进程使用的堆、模块和线程；

使用 Process32First 函数获取有关系统快照中遇到的第一额进程的信息；

使用 Process32Next 函数获取有关系统快照中记录的下一个进程的信息；

使用 OpenProcess 函数打开现有的本地进程对象；

使用 GetProcessMemoryInfo 函数获取指定进程的内存使用情况的信息；

相关调用如下所示：

```
PROCESSENTRY32 pe;  
pe.dwSize = sizeof(pe);  
//获取指定进程的快照，以及这些进程使用的堆、模块和线程  
HANDLE hProcessSnap = ::CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);  
//获取有关系统快照中遇到的首个进程的信息  
BOOL bMore = ::Process32First(hProcessSnap, &pe);
```

下面得到打开进程信息并逐个获取相关的信息：

```
while (bMore)  
{  
    HANDLE hP = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pe.th32ProcessID);  
    PROCESS_MEMORY_COUNTERS pmc;  
    ZeroMemory(&pmc, sizeof(pmc));  
    if (GetProcessMemoryInfo(hP, &pmc, sizeof(pmc)))  
    {  
        //相关的输出  
    }  
}
```

操作系统课程设计实验报告

```
}  
//获取下个进程  
bMore = ::Process32Next(hProcessSnap, &pe);  
}
```

(5) 根据 PID 获得进程虚拟地址空间的信息:

```
int pid = 0;  
printf("Please search the infomation by PID:\n");  
SYSTEM_INFO si;  
ZeroMemory(&si, sizeof(si));  
GetSystemInfo(&si);  
HANDLE hProcess;  
.....  
hProcess = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid);
```

之后得到相关信息, 并进行输出

```
MEMORY_BASIC_INFORMATION mbi;  
ZeroMemory(&mbi, sizeof(mbi));  
LPCVOID pBlock = (LPVOID)si.lpMinimumApplicationAddress;  
printf("Address\t\t\t | Size\t\t | State\t | Protect\t | Type\t\t | Module\n\n");  
  
if (VirtualQueryEx(hProcess, pBlock, &mbi, sizeof(mbi)) == sizeof(mbi))  
{  
    LPCVOID pEnd = (PBYTE)pBlock + mbi.RegionSize;  
    TCHAR szSize[MAX_PATH];  
    StrFormatByteSize(mbi.RegionSize, szSize, MAX_PATH);  
}
```

2. 主函数相关控制机制:

通过输入操作码 opnum, 来控制进入显示界面的哪一部分, 如下所示。

```
while (TRUE)  
{  
    printf("-----\n\n");  
    printf("Welcome to Memory Information Sysyem Of 1120191600.\n");  
    printf("1.Memory Information press: '1'\n");  
    printf("2.Performance Information press '2'\n");  
    printf("3.System Information press '3'\n");  
    printf("4.Process Infomation press '4'\n");  
    printf("5.Process Virtual Memory Information press '5'\n");  
    printf("6.Quit System press '0'\n");  
    printf("-----\n\n");  
}
```

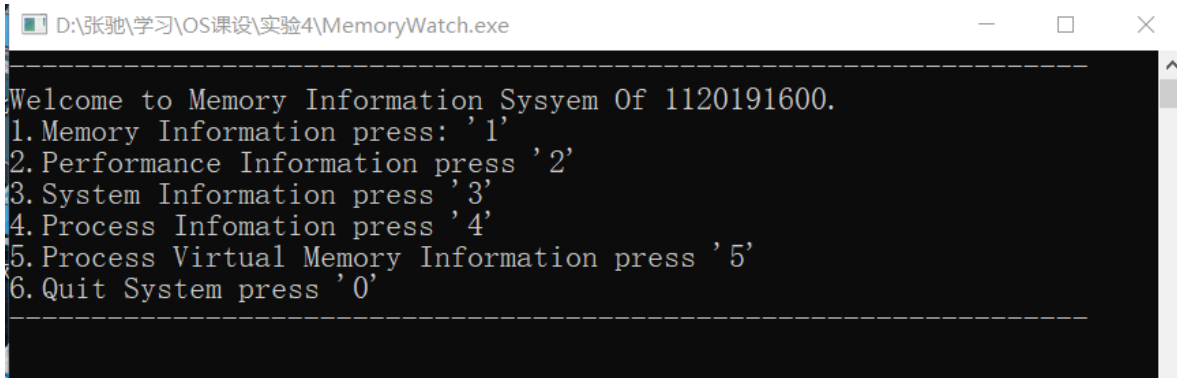
操作系统课程设计实验报告

```
---\n\n");  
  
    int opnum = 0;  
    scanf("%d",&opnum);
```

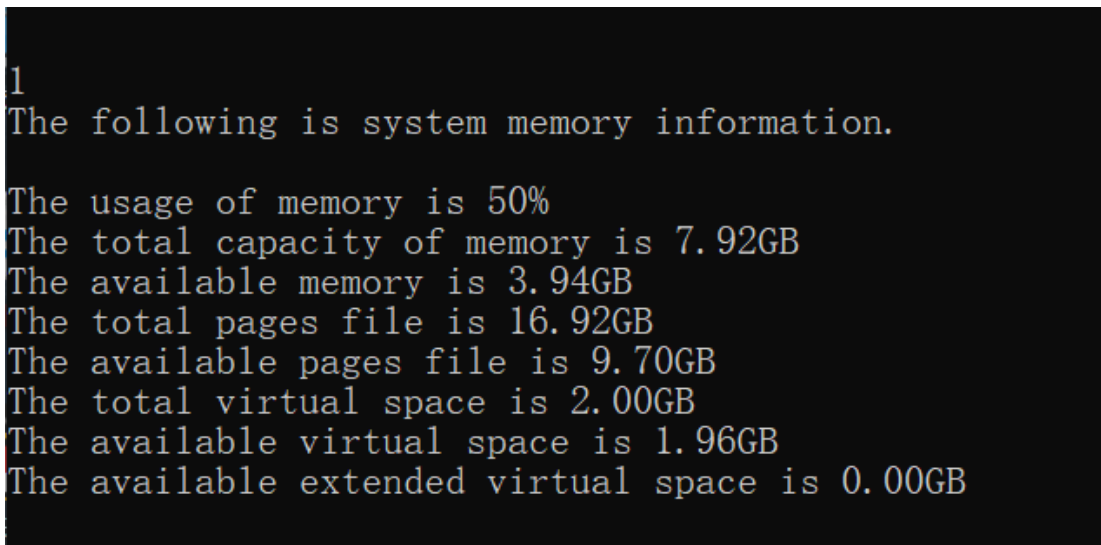
之后根据输入的操作码调用相关的函数。

五、实验结果和分析：

1. 打开的主界面：



2. 内存信息显示界面：



3. 性能信息显示界面：

操作系统课程设计实验报告

```
2
The following is performance information.

The number of pages currently committed by the system is 1884221
The current maximum number of pages that can be committed by the system without e
xtending the paging file(s) is 4436029
The maximum number of pages that were simultaneously in the committed state since
the last system reboot is 1979001
The amount of actual physical memory in pages is 2076733
The amount of physical memory currently available is 1057037
The amount of system cache memory is 968445
The sum of the memory currently in the paged and nonpaged kernel pools is 213071
The memory currently in the paged kernel pool is 124110
The memory currently in the nonpaged kernel pool is 88961
The size of a page is 4KBThe current number of open handles is 96238
The current number of processes is 236
The current number of threads is 2842
```

4. 系统信息显示界面：

```
3
The following is system information.

The page size and the granularity of page protection and commitment is 4KB
The pointer to the lowest memory address accessible to applications and dynamic-link libraries (DLLs) is 0x00010000
The pointer to the highest memory address accessible to applications and DLLs is 0x7ffeffff
The number of logical processors in the current group is 8
The granularity for the starting address at which virtual memory can be allocated is 64KB
```

5. 所有进程信息的显示：

```
4
The following is all process information.
```

PID	Execute File	Working Set Size(KB)
836	sihost.exe	25876.00
2344	svchost.exe	33028.00
9224	svchost.exe	34624.00
9368	taskhostw.exe	17692.00
10016	explorer.exe	167008.00
10368	SCMainFrame.exe	8500.00
10972	svchost.exe	36136.00
9280	StartMenuExperienceHost.exe	67960.00
11428	RuntimeBroker.exe	24336.00
11556	SearchApp.exe	181352.00
11876	RuntimeBroker.exe	30336.00
12140	YourPhone.exe	2840.00
12304	SettingSyncHost.exe	7748.00
12416	LockApp.exe	45548.00
12684	RuntimeBroker.exe	26284.00
13304	RuntimeBroker.exe	18704.00
11272	TextInputHost.exe	64052.00
13428	RuntimeBroker.exe	19232.00
13744	SecurityHealthSystray.exe	8380.00
13964	RtkAudUService64.exe	9464.00
12364	OneDrive.exe	62244.00

操作系统课程设计实验报告

11476	chrome.exe	64684.00
6272	chrome.exe	91872.00
6292	chrome.exe	33832.00
6880	chrome.exe	77372.00
8072	chrome.exe	91152.00
12368	chrome.exe	25636.00
13652	dllhost.exe	12500.00
1996	SearchProtocolHost.exe	8636.00
11496	smartscreen.exe	23864.00
14068	MemoryWatch.exe	6316.00
4904	conhost.exe	18488.00
14332	QQ.exe	224668.00
15288	TXPlatform.exe	2980.00
7368	QQExternal.exe	23608.00

6. 进程的虚拟地址空间信息的展示:

当存在此 PID 时, 直接输出该进程相关的信息

```
5
Please search the infomation by PID:
Enter A PID: 6272
```

Address	Size	State	Protect	Type	Module
00010000-7ffe0000	1.99 GB	Free	NOACCESS		
7ffe0000-7ffe1000	4.00 KB	Committed	READONLY	Private	
7ffe1000-7ffef000	56.0 KB	Free	NOACCESS		
7ffef000-7fff0000	4.00 KB	Committed	READONLY	Private	

当不存在此 PID 时, 提示重新输入

```
5
Please search the infomation by PID:
Enter A PID: 123
No process ID:123 , Please check your PID:
Enter A PID:
```

六、讨论、心得

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。

通过本实验, 掌握了许多在 Windows 下关于内存信息的系统调用函数, 了解系统内部内存的工作方式和工作情况。通过查阅相关资料和内容, 进一步了解了查询系统信息的 API 以及与系统信息有关的结构体, 了解了结构体中的内容, 在内存监控系统中对相应的信息进行输出, 从而对课堂上的内存分配方式有了进一步的了解和认识。