|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 数学与信息科学学院 | | | | | | |  |
|  |  | | | | | | |  |
|  |  | | | | | | |  |
|  | 实验报告 | | | | | | |  |
|  |  | | | | | | |  |
|  |  | | | | | | |  |
|  | 课程名称： | | 计算机操作系统 | | | | |  |
|  | 姓 名： | | 任希恒 | | | | |  |
|  | 学 号： | | 541910010217 | | | | |  |
|  | 专业班级： | | 信息与计算科学专业19-02班 | | | | |  |
|  | 指导教师： | | 时海亮 | | | | |  |
|  |  | |  | | | | |  |
|  |  | |  | | | | |  |
|  |  | 2021-2022 | | 学年第 | 2 | 学期 |  |  |

实验4 系统内存使用统计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验日期： | 2022年05月13日 | 实验类型： | 设计型 | 实验成绩： |  |

# 一、实验目的

通过本实验了解Windows内存管理机制，理解也是存储管理技术，熟悉Windows内容管理基本数据结构，掌握Windows内存管理基本API的使用。

# 二、实验内容

1．实现系统内存使用统计功能。

# 三、实验代码及结果

1.代码如下：

#include<windows.h>

#include<iostream.h>

void main()

{

MEMORYSTATUS ms;

LPVOID pVir;

LPVOID pMal;

GlobalMemoryStatus(&ms);

cout<<"Memory Status before allocate memory: "<<endl;

cout<<"Memory Status Length: "<<ms.dwLength<<"Bytes"<<endl;

cout<<"Memory Load: "<<ms.dwMemoryLoad<<"%"<<endl;

cout<<"Total Physical Memory: "<<ms.dwTotalPhys/(1024\*1024)<<"MB"<<endl;

cout<<"Available Physical Memory: "<<ms.dwAvailPhys/(1024\*1024)<<"MB"<<endl;

cout<<"Total Page File: "<<ms.dwTotalPageFile/(1024\*1024)<<"MB"<<endl;

cout<<"Available Page File: "<<ms.dwAvailPageFile/(1024\*1024)<<"MB"<<endl;

cout<<"Total Virtual Memory: "<<ms.dwTotalVirtual/(1024\*1024)<<"MB"<<endl;

cout<<"Available Virtual Memory: "<<ms.dwAvailVirtual/(1024\*1024)<<"MB"<<endl;

cout<<endl<<endl;

cout<<"Now allocat 32M virtual memory and 16M physical memory..."<<endl;

pVir=VirtualAlloc(NULL,1024\*1024\*32,MEM\_COMMIT | MEM\_RESERVE,PAGE\_READWRITE);

if(!pVir)

{

cout<<"Virtual memory allocate fail!"<<endl;

return;

}

pMal=malloc(1024\*1024\*16);

GlobalMemoryStatus(&ms);

cout<<"Memory Status before allocate memory: "<<endl;

cout<<"Memory Status Length: "<<ms.dwLength<<"Bytes"<<endl;

cout<<"Memory Load: "<<ms.dwMemoryLoad<<"%"<<endl;

cout<<"Total Physical Memory: "<<ms.dwTotalPhys/(1024\*1024)<<"MB"<<endl;

cout<<"Available Physical Memory: "<<ms.dwAvailPhys/(1024\*1024)<<"MB"<<endl;

cout<<"Total Page File: "<<ms.dwTotalPageFile/(1024\*1024)<<"MB"<<endl;

cout<<"Available Page File: "<<ms.dwAvailPageFile/(1024\*1024)<<"MB"<<endl;

cout<<"Total Virtual Memory: "<<ms.dwTotalVirtual/(1024\*1024)<<"MB"<<endl;

cout<<"Available Virtual Memory: "<<ms.dwAvailVirtual/(1024\*1024)<<"MB"<<endl;

cout<<endl<<endl;

cout<<"Now release 32M virtual memory and 16M physical memory..."<<endl;

if(!VirtualFree(pVir,0,MEM\_RELEASE))

{

cout<<"Free the virtual memory fail!"<<endl;

return;

}

free(pMal);

GlobalMemoryStatus(&ms);

cout<<"Memory Status before allocate memory: "<<endl;

cout<<"Memory Status Length: "<<ms.dwLength<<"Bytes"<<endl;

cout<<"Memory Load: "<<ms.dwMemoryLoad<<"%"<<endl;

cout<<"Total Physical Memory: "<<ms.dwTotalPhys/(1024\*1024)<<"MB"<<endl;

cout<<"Available Physical Memory: "<<ms.dwAvailPhys/(1024\*1024)<<"MB"<<endl;

cout<<"Total Page File: "<<ms.dwTotalPageFile/(1024\*1024)<<"MB"<<endl;

cout<<"Available Page File: "<<ms.dwAvailPageFile/(1024\*1024)<<"MB"<<endl;

cout<<"Total Virtual Memory: "<<ms.dwTotalVirtual/(1024\*1024)<<"MB"<<endl;

cout<<"Available Virtual Memory: "<<ms.dwAvailVirtual/(1024\*1024)<<"MB"<<endl;

cout<<endl<<endl;

}

运行结果如图1所示

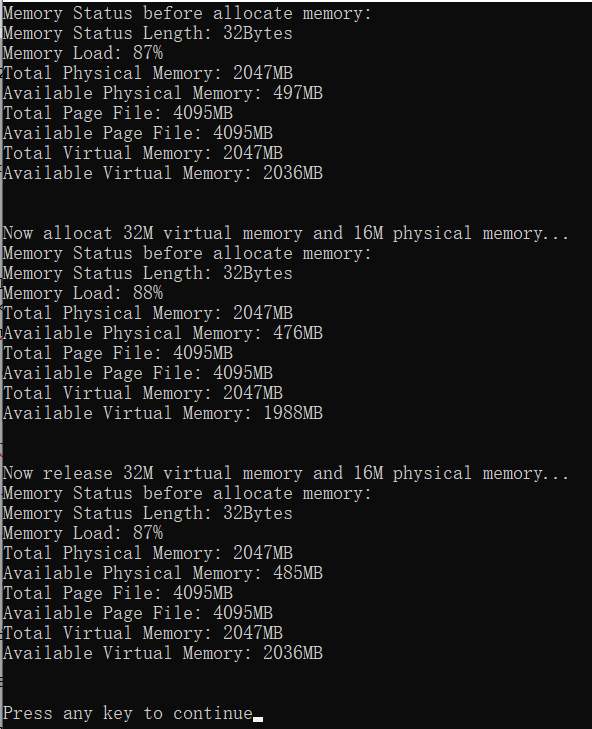


图1

# 四、实验总结

系统结构MEMORYSTATUS中包含当前物理内存和虚拟内存和虚拟内存信息，使用函数GlobaiMemoryStatus（）可以将这些信息存储在结构MEMORYSTATUS中。函数VirtualAlloc()可以在调用进程的虚拟地址空间中保留或提交若干页面。保留意味着这段虚拟地址地址不能被使用，当提交时，这段虚拟地址才真正被分配给进程。如果函数调用成功，则返回值为已分配虚拟地址空间的起始地址。如果函数调用失败，则返回值为NULL。若要得到更多的错误信息，可调用GetLast6Error()函数。