

Windows 操作系统

C/C++ 程序实验

姓名：_____陈展博_____

学号：_____1221001003_____

班级：_____计科 1 班_____

院系：_____信工_____

_____2024_____年__11__月__6__日

实验五 Windows 内存结构

一、背景知识

二、实验目的

三、工具/准备工作

四、实验内容与步骤

Windows 提供了一个 API 即 `GetSystemInfo()`，以使用户能检查系统中虚拟内存的一些特性。清单 5-1 显示了如何调用该函数以及显示系统中当前内存的参数。

步骤 1：登录进入 Windows。

步骤 2：在“开始”菜单中单击“程序-Microsoft Visual Studio Code”。

步骤 3：新建项目名为“5-1”，并且新建项“5-1.cpp”。

步骤 4：按“F5”开始调试，注意路径里不要含有中文。

步骤 5：按暂停按钮可暂停程序的执行，按终止按钮可终止程序的执行。

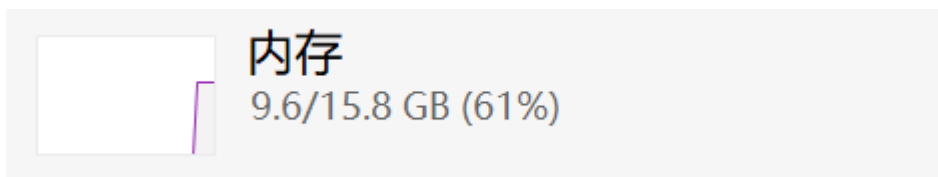


操作能否正常进行？如果不行，则可能的原因是什么？

操作能够正常进行，如果不行可能是文件路径包含中文字符，或者文件中有错误的字符格式。

运行结果（如果运行不成功，则可能的原因是什么？）：

- 1) 虚拟内存每页容量为：4.00KB
- 2) 最小应用地址：0x00010000
- 3) 最大应用地址：0xfffffff
- 4) 当前可供应用程序使用的内存空间为：3.99GB
- 5) 当前计算机的实际内存大小为：15.8GB



图片 1 任务管理器中查看物理内存大小

阅读和分析程序 5-1，请回答问题：

1) 在 32 位系统中，理论上每个 Windows 应用程序可以独占的最大存储空间是：理论上，在 32 位系统中可以独占的最大空间为 4GB。

2) 在清单 5-1 程序中，用于检查系统中虚拟内存特性的 API 函数是：

`si.lpMinimumApplicationAddress` 给出每个进程的最小应用地址 `0x00010000`，在其开头空出 64KB 的保护区。

`si.lpMaximumApplicationAddress` 给出每个进程的最大应用地址 `0xfffffff`，在其结尾余下 64KB 的保护区。

提示：可供应用程序使用的内存空间实际上已经减去了开头与结尾两个 64KB 的保护区。虚拟内存空间中的 64KB 保护区是防止编程错误的一种 Windows 方式。任何对内存中这一区域的访问（读、写、执行）都将引发一个错误陷阱，从而导致错误并终止程序的执行。也就是说，假如用户有一个 NULL 指针（地址为 0），但仍试图在此之前很近的地址处使用另一个指针，这将因为试图从更低的保留区域读写数据，从而产生意外错误并终止程序的执行。

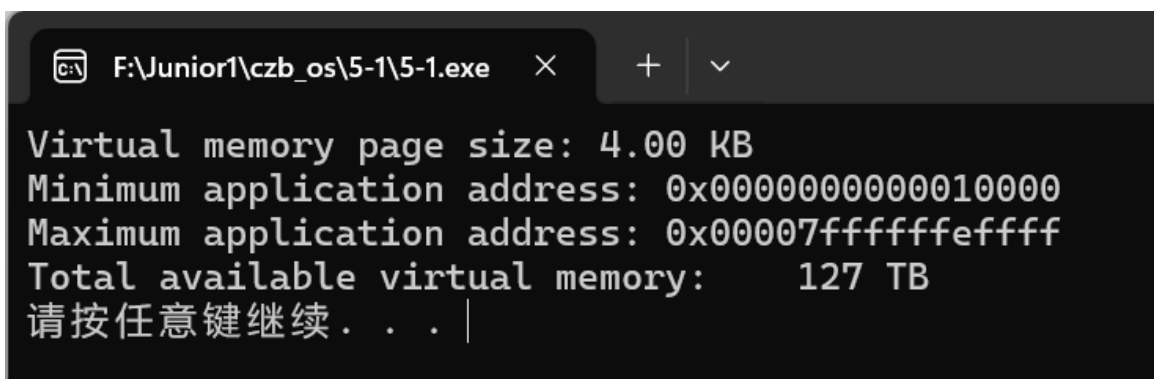
选作：对于 64 位系统，说明如何修改程序以便显示相关信息。

```

19 TCHAR szPageSize [MAX_PATH] ;
20 :: StrFormatByteSize(si.dwPageSize, szPageSize, MAX_PATH) ;
21 DWORD64 dwMemSize = (DWORD64) si.lpMaximumApplicationAddress -
22     (DWORD64) si.lpMinimumApplicationAddress;
23 TCHAR szMemSize [MAX_PATH] ;
24 :: StrFormatByteSize(dwMemSize, szMemSize, MAX_PATH) ;
25
26 MEMORYSTATUSEX statex;
27 statex.dwLength = sizeof (statex);
28 GlobalMemoryStatusEx (&statex);
29 // 将内存信息显示出来
30 std :: cout << "Virtual memory page size: " << szPageSize
31     << std :: endl;
32 std :: cout.fill ('0') ;
33 std :: cout << "Minimum application address: 0x"
34     << std :: hex << std :: setw(16)
35     << (DWORD64) si.lpMinimumApplicationAddress
36     << std :: endl;
37 std :: cout << "Maximum application address: 0x"
38     << std :: hex << std :: setw(16)
39     << (DWORD64) si.lpMaximumApplicationAddress
40     << std :: endl;
41
42 _tprintf (TEXT("Total available virtual memory:%*I64d TB\n"),
43     WIDTH, statex.ullTotalVirtual/DIV/DIV/DIV/DIV); //DIV 1027 WIDTH 7
44

```

图片 2 修改代码查看 64 位系统的虚拟内存



```

F:\Junior1\czb_os\5-1\5-1.exe
Virtual memory page size: 4.00 KB
Minimum application address: 0x00000000000010000
Maximum application address: 0x00007fffffffeffff
Total available virtual memory: 127 TB
请按任意键继续. . .

```

图片 3 代码输出结果

- 64 位 Windows 上的 64 位进程的虚拟地址空间在 128 TB 范围内，0x000'000000000 到 0x7FFF'FFFFFFFFF。

图片 4 Windows 官方学习文档提供资料

所以在 64 位系统中可用的最大虚拟内存为 128TB，进程可用的最小地址为 0x10000，进程可用的最大地址为 0x7fffffffeffff，留有 64KB 的两个保护区。