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__27___日

实验八　Windows　内存管理

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

1. 读者写者问题

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“8-1”，并且新建项“8-1.cpp”。

步骤 4：单按“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。

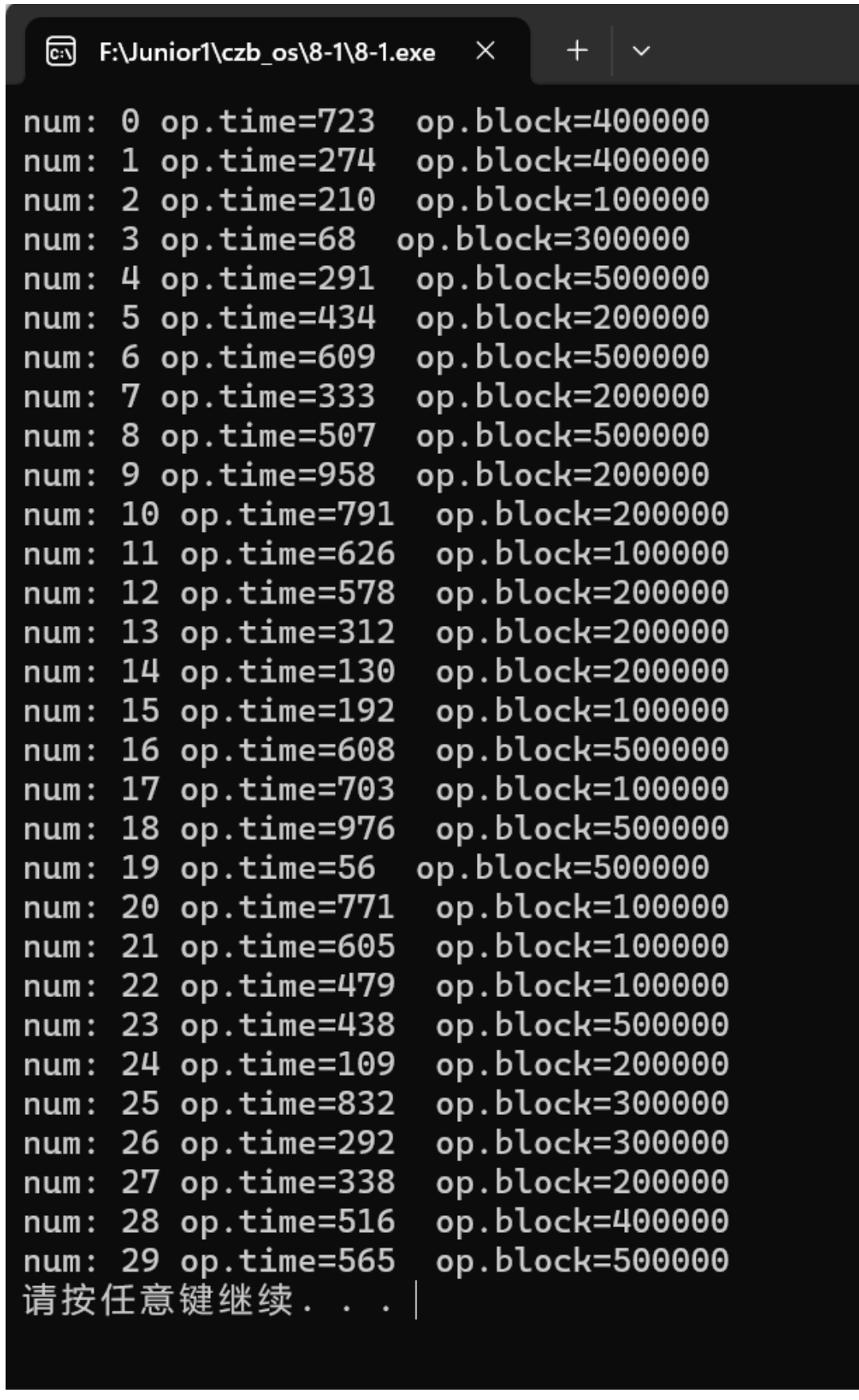
操作能否正常进行？如果不行，则可能的原因是什么？

操作能够正常进行，如果不行，可能是因为文件中包含中文字符，或文件路径包含中文。

运行结果是：

```c
int main()
{
    FILE * file;
    file = fopen("opfile", "wb");   //"opfile"为二进制用确定内存操作
    operation op;

    srand((unsigned)time(NULL));
    int num = 0;
    for (int j = 0; j < 6; j++)
    {
        for (int i = 0; i <5; i++)
        {
            //0-PAGE_READONLY;
            //1-PAGE_READWRITE;
            //2-PAGE_EXECUTE;
            //3-PAGE_ECECUTE_READ;
            //4-PAGE_EXECUTE_READWRITE;

            op.time = rand() % 1000;        //随即生成等待时间

            printf("num: %d op.time=%d  ",num++, op.time);
            op.block = (rand() % 5 + 1) *100;       //随即生成块大小
            printf("op.block=%d\n", op.block);
            op.oper = j;

            op.protection = i;
            fwrite(&op, sizeof(operation), 1, file);    //将生成的结构写入文件
        }
    }
    fclose(file);
    system("pause");
    return 0;
}
```

图片 1 修改块大小，并且加上 fclose(file)


F:\Junior1\czb_os\8-1\8-1.exe
num: 0 op.time=723 op.block=400000
num: 1 op.time=274 op.block=400000
num: 2 op.time=210 op.block=100000
num: 3 op.time=68 op.block=300000
num: 4 op.time=291 op.block=500000
num: 5 op.time=434 op.block=200000
num: 6 op.time=609 op.block=500000
num: 7 op.time=333 op.block=200000
num: 8 op.time=507 op.block=500000
num: 9 op.time=958 op.block=200000
num: 10 op.time=791 op.block=200000
num: 11 op.time=626 op.block=100000
num: 12 op.time=578 op.block=200000
num: 13 op.time=312 op.block=200000
num: 14 op.time=130 op.block=200000
num: 15 op.time=192 op.block=100000
num: 16 op.time=608 op.block=500000
num: 17 op.time=703 op.block=100000
num: 18 op.time=976 op.block=500000
num: 19 op.time=56 op.block=500000
num: 20 op.time=771 op.block=100000
num: 21 op.time=605 op.block=100000
num: 22 op.time=479 op.block=100000
num: 23 op.time=438 op.block=500000
num: 24 op.time=109 op.block=200000
num: 25 op.time=832 op.block=300000
num: 26 op.time=292 op.block=300000
num: 27 op.time=338 op.block=200000
num: 28 op.time=516 op.block=400000
num: 29 op.time=565 op.block=500000
请按任意键继续. . .

图片 2 控制台输出结果如上图所示

图片 3 文件夹中多出一个 opfile 文件

步骤 6：新建项目名为“8-2”，并且新建项“8-2.cpp”。

步骤 7：将“opfile”文件复制到该项目文件夹下。

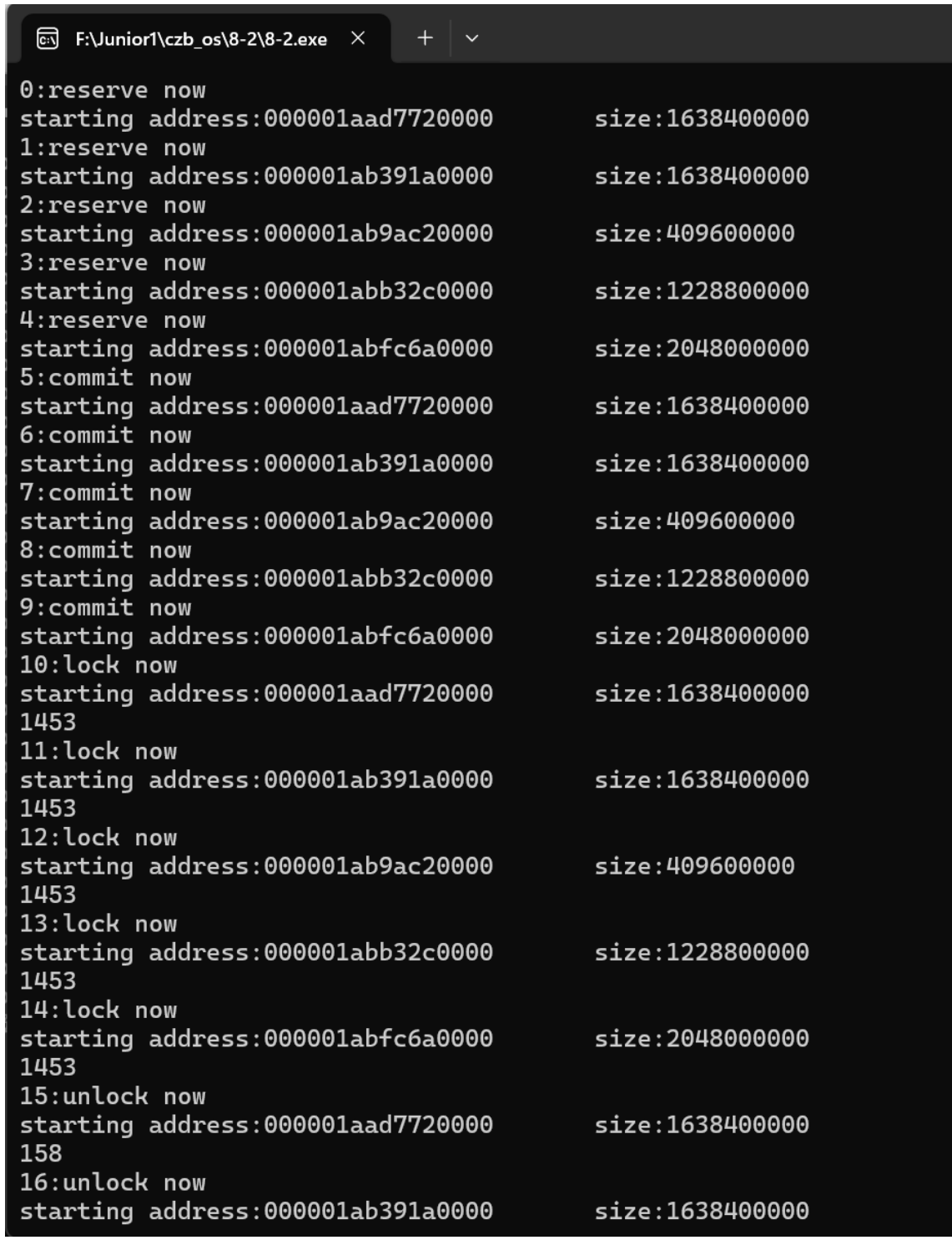
步骤 8：按“F5”开始调试，注意路径里不要含有中文。

步骤 9：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。

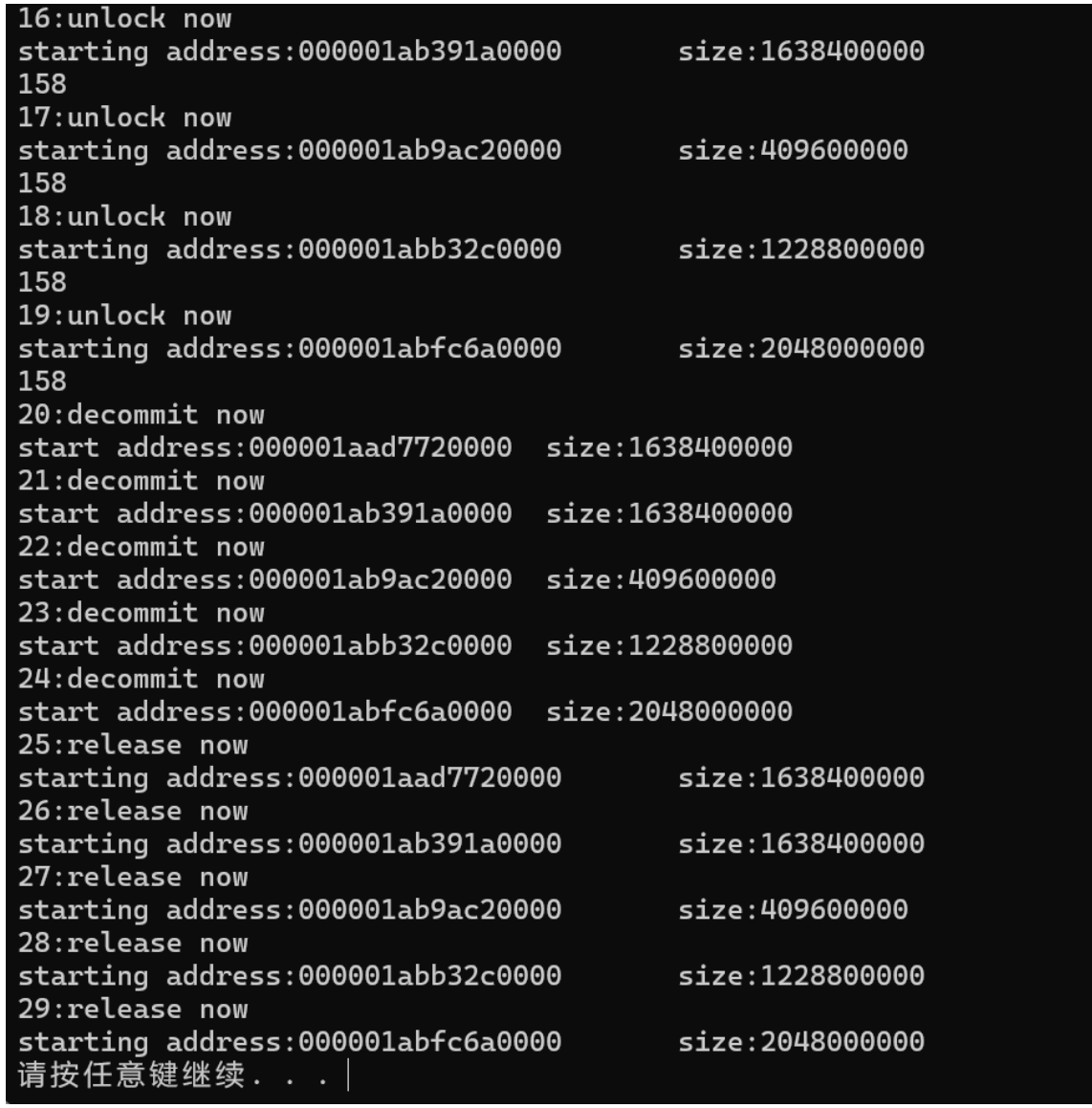
操作能否正常进行？如果不行，则可能的原因是什么？

操作能够正常进行，如果不行，可能是因为文件中包含中文字符，或文件路径包含中文。

运行结果是：


F:\Junior1\czb_os\8-2\8-2.exe
0:reserve now
starting address:000001aad7720000 size:1638400000
1:reserve now
starting address:000001ab391a0000 size:1638400000
2:reserve now
starting address:000001ab9ac20000 size:409600000
3:reserve now
starting address:000001abb32c0000 size:1228800000
4:reserve now
starting address:000001abfc6a0000 size:2048000000
5:commit now
starting address:000001aad7720000 size:1638400000
6:commit now
starting address:000001ab391a0000 size:1638400000
7:commit now
starting address:000001ab9ac20000 size:409600000
8:commit now
starting address:000001abb32c0000 size:1228800000
9:commit now
starting address:000001abfc6a0000 size:2048000000
10:lock now
starting address:000001aad7720000 size:1638400000
1453
11:lock now
starting address:000001ab391a0000 size:1638400000
1453
12:lock now
starting address:000001ab9ac20000 size:409600000
1453
13:lock now
starting address:000001abb32c0000 size:1228800000
1453
14:lock now
starting address:000001abfc6a0000 size:2048000000
1453
15:unlock now
starting address:000001aad7720000 size:1638400000
158
16:unlock now
starting address:000001ab391a0000 size:1638400000

图片 4 运行结果 1



图片 5 运行结果 2

结果分析：依次分析保留、提交、锁、解锁、回收、释放一个区域的操作引起的内存状态的变化。

保留内存主要影响虚拟内存空间。它会在地址空间中分配区域，但未实际分配物理内存或页面文件。

```
dwActiveProcessorMask    65535
dwAllocationGranularity    65536
dwNumberOfProcessors    16
dwOemId 9
dwPageSize      4096
dwProcessorType      8664
lpMaximumApplicationAddress      7ffffffeffff
lpminimemApplicationAddress            10000
wProcessorArchitecture    9
wProcessorLevel       6
wProcessorRevision   8d01
wReserved       0
**************************************************
AllocationBase   0000000000000000
AllocationProtect      0
BaseAddress      0000000000010000
Protect     1
RegionSize       7ffd0000
State 10000
Type 0
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0
dwAvailPageFile       22bd18000
dwAvailPhys      200903000
dwAvailVirtual   7ffefd00a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy       3f134e000
dwTotalVirtual   7fffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1
dwAvailPageFile       22a2e6000
dwAvailPhys      1feca5000
dwAvailVirtual   7ffe9b58a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy       3f134e000
dwTotalVirtual   7fffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

图片 6 如图 0 号 dwAvailVirtual – 1 号 dwAvailVirtual== 1,638,400,000，保留的虚拟空间大小和我们在前面 opfile 里要的空间块空间大小一样

```
1
dwAvailPageFile       22a2e6000
dwAvailPhys     1feca5000
dwAvailVirtual  7ffe9b58a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
2
dwAvailPageFile       22a212000
dwAvailPhys     1fe77f000
dwAvailVirtual  7ffe39b0a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
3
dwAvailPageFile       22a1a0000
dwAvailPhys     1fe790000
dwAvailVirtual  7ffe2146a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
4
dwAvailPageFile       22a1a7000
dwAvailPhys     1fe799000
dwAvailVirtual  7ffdd808a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

图片 7 如图 1 号 dwAvailVirtual – 2 号 dwAvailVirtual== 1,638,400,000，保留的虚拟空间大小和我们在前面 opfile 里要的空间块空间大小一样，

图 2 号 dwAvailVirtual – 3 号 dwAvailVirtual== 409,600,000，保留的虚拟空间大小和我们在前面 opfile 里要的空间块空间大小一样，

图 3 号 dwAvailVirtual – 4 号 dwAvailVirtual== 1,228,800,000，保留的虚拟空间大小和我们在前面 opfile 里要的空间块空间大小一样。

```
5
dwAvailPageFile       22a1b9000
dwAvailPhys     1fe8e6000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
6
dwAvailPageFile       1c8339000
dwAvailPhys     1fe697000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7fffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
7
dwAvailPageFile       166698000
dwAvailPhys     1fe033000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7fffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
8
dwAvailPageFile       14ddde000
dwAvailPhys     1fda08000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7fffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

图片 8 4 号 dwAvailVirtual – 5 号 dwAvailVirtual== 2,048,000,000，保留的虚拟空间大小和我们在前面 opfile 里要的空间块空间大小一样。

0-4 号为止，对虚拟内存的 reserve 操作结束。5-9 号，开始 commit 操作，开始对物理内存提交分配申请。

5 号 dwAvailPhys – 6 号 dwAvailPhys == 2,420,736，
5 号 dwAvailPageFile– 6 号 dwAvailPageFile== 1,642,594,304，

6 号 dwAvailPhys – 7 号 dwAvailPhys == 6,701,056，

6 号 dwAvailPageFile– 7 号 dwAvailPageFile== 1,640,632,320，
7 号 dwAvailPhys – 8 号 dwAvailPhys == 6,467,584
7 号 dwAvailPageFile– 8 号 dwAvailPageFile== 411,803,648。

```
9
dwAvailPageFile       1049fe000
dwAvailPhys     1fdd1d000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
a
dwAvailPageFile       8a727000
dwAvailPhys     1fe1ec000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

图片 9　8 号 dwAvailPhys – 9 号 dwAvailPhys == -3,231,744，

8 号 dwAvailPageFile– 9 号 dwAvailPageFile== 1,228,800,000，

9 号 dwAvailPhys – a 号 dwAvailPhys == -5,042,176

9 号 dwAvailPageFile– a 号 dwAvailPageFile== 2,049,798,144。

提交内存将保留区域的内存分配给物理页面或页面文件。这一步使保留的内存可用，但会减少物理内存和页面文件的可用空间，这时

dwAvailPhys 减少：物理内存用于支持提交的页面。

dwAvailPageFile 减少：页面文件分配更多空间。

5-9 号为止，对虚拟内存的 commit 操作结束。a-e 号，开始 lock 操作，开始对物理内存锁定。

锁定内存会确保被锁定的内存始终驻留在物理内存中而不会被交换到页面文件：

- **dwAvailPhys 几乎不变**：物理内存被锁定。
- **dwAvailPageFile 不变**，因为锁定内存不会影响页面文件。
- **dwAvailVirtual 不变**，因为没有新增保留的虚拟内存。

a-e 号为止，对虚拟内存的 lock 操作结束。F-13 号，开始 unlock 操作，开始对物理内存解锁。

解锁内存会将内存的锁定状态解除，允许其被交换至页面文件：

- **dwAvailPhys 几乎不变**。
- **dwAvailPageFile 不变**。
- **dwAvailVirtual 不变**。

```
14
dwAvailPageFile      8af1a000
dwAvailPhys     1fe420000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile      46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
15
dwAvailPageFile      ed122000
dwAvailPhys     1fe91f000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile      46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
16
dwAvailPageFile      14ec68000
dwAvailPhys     1fec49000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile      46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
17
dwAvailPageFile      167384000
dwAvailPhys     1fea39000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile      46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
```

图片 10 15 号 dwAvailPhys – 14 号 dwAvailPhys == 5,238,784，
15 号 dwAvailPageFile– 14 号 dwAvailPageFile== 1,646,297,088，
16 号 dwAvailPhys – 15 号 dwAvailPhys == 3,317,760，
16 号 dwAvailPageFile– 15 号 dwAvailPageFile== 1,639,211,008。

F-13 号为止，对虚拟内存的 unlock 操作结束。14-18 号，开始 decommit 操作，开始对物理内存撤销。

回收内存会释放提交的页面，保持保留状态。回收后：

- **dwAvailPhys 增加**：释放了提交的物理页面。
- **dwAvailPageFile 增加**：页面文件也释放相应空间。
- **dwAvailVirtual 不变**：保留的虚拟内存仍然存在。

```
19
dwAvailPageFile       22affb000
dwAvailPhys     1ff337000
dwAvailVirtual  7ffd5df6a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1a
dwAvailPageFile       22b09a000
dwAvailPhys     1ff2f2000
dwAvailVirtual  7ffdbf9ea000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1b
dwAvailPageFile       22aef0000
dwAvailPhys     1ff24c000
dwAvailVirtual  7ffe2146a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7fffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1c
dwAvailPageFile       22b27d000
dwAvailPhys     1ff7a0000
dwAvailVirtual  7ffe39b0a000
dwLength 38
dwMemoryLoad       31
dwTotalPageFile       46534e000
dwTotalPhy      3f134e000
dwTotalVirtual  7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

图片 11 1a 号 dwAvailVirtual – 19 号 dwAvailVirtual== 1,638,400,000，
1b 号 dwAvailVirtual – 1a 号 dwAvailVirtual== 1,638,400,000，
1c 号 dwAvailVirtual – 1b 号 dwAvailVirtual== 409,600,000，
1d 号 dwAvailVirtual – 1c 号 dwAvailVirtual== 1,228,800,000，
1e 号 dwAvailVirtual – 1d 号 dwAvailVirtual== 2,049,048,576。

14-18 号为止，对虚拟内存的 decommit 操作结束。19-1d 号，开始 release 操作，开始对虚拟保留内存释放。

释放内存会完全释放保留的虚拟地址空间，归还给系统：

- **dwAvailVirtual 增加**：虚拟地址空间恢复。
- **dwAvailPhys 和 dwAvailPageFile 几乎不变**，因为只影响虚拟地址。

选作：在以上源代码的基础上，编写一个程序。

创建两个线程，一个用于内存分配，另一个用于跟踪内存的分配情况并打印信息。将virtualalloc 函数的参数 ftallocationtype 分别改为 MEM_RESET 或 MEM_TOP_DOWN，将 flprotect 参数分别改为 PAGE_GUARD、PAGE_NOACCESS 或 PAGE_NOCACHE，再进行本练习的各项操作，再查看内存分配的各个结果，分析原因。尝试调换分配、回收、内存复位、加锁、解锁、提交、回收的次序，查看结果，并分析原因。

MEM_RESET、MEM_TOP_DOWN、PAGE_GUARD、PAGE_NOACCESS、PAGE_NOCACHE 属性的含义在 MSDN 中均有详细介绍，请读者自行查阅。

请描述你所做的工作：

```
{
case 0:
{
    index = 0;
    temp = PAGE_GUARD;
    break;
}
case 1:
    temp = PAGE_NOACCESS;
    break;
case 2:
    temp = PAGE_NOCACHE;
    break;
case 3:
    temp = PAGE_EXECUTE_READ;
    break;
case 4:
    temp = PAGE_EXECUTE_READWRITE;
    break;
default:
    temp = PAGE_GUARD;
}
```

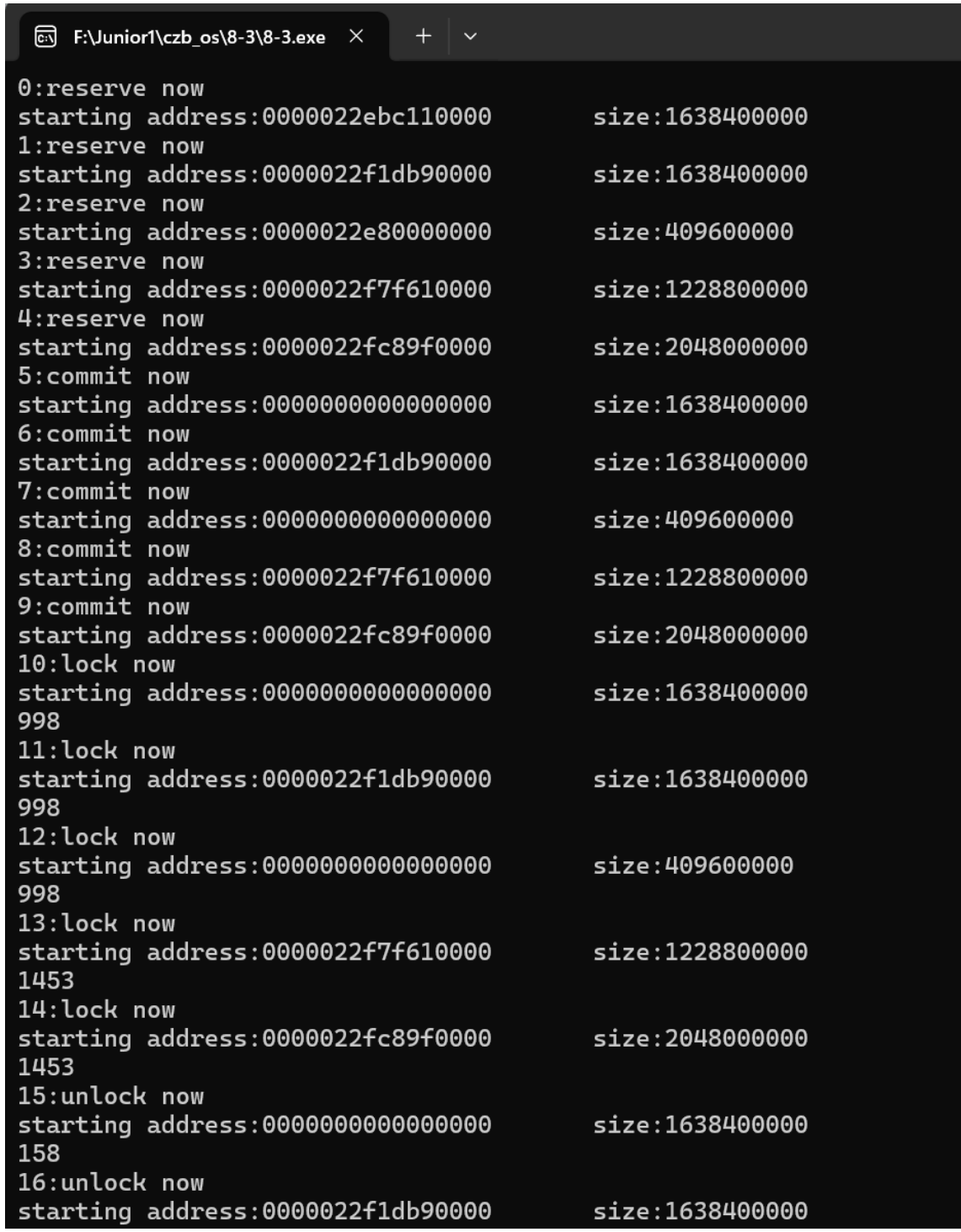
图片 12 修改代码将原本的 flprotect 参数更改为 PAGE_GUARD、PAGE_NOACCESS、

```
155         case 1:     //MEM_TOP_DOWN提交一个区域
156         {
157             cout << "commit now" << endl;
158 
159 traceArray[index].start=VirtualAlloc(traceArray[index].start, traceArray[index].size, MEM_COMMIT | MEM_TOP_DOWN, temp);
160 
161             index++;
162 cout << "starting address:" <<  std::setw(16) << std::setfill('0') << std::hex
163      << (DWORD64) traceArray[index - 1].start << '\t' << std::dec << "size:" << traceArray[index - 1].size << endl;
164 
165             break;
166         }
```

图片 13 将 commit 更改为 TOP DOWN 形式

```
200         case 5:     //RESET一个区域
201         {
202             cout << "reset now" << endl;
203      cout << "starting address:"<<  std::setw(16) << std::setfill('0') << std::hex
204       << (DWORD64) traceArray[index].start << '\t' << std::dec<< "size:" << traceArray[index].size << endl;
205 
206             if (!VirtualFree(traceArray[index++].start, 0, MEM_RESET))
207                 cout << GetLastError() << endl;
208             break;
209         }
210         default:
211             cout << "error" << endl;
212         }
213         ReleaseSemaphore(trac, 1, NULL);    //释放信号量通知tracker可以打印信息
```

图片 14 将 release 更改为 reset



图片 15 控制台输出 1

commit 操作时部分操作返回起始地址为 0000000000000000，说明 VirtualAlloc 在提交时失败了，因为 PAGE_GUARD 和 PAGE_NOCACHE 状态下无法 commit。

部分锁定失败，返回错误码 998 或 1453

错误 998 可能是因为提交失败导致锁定的起始地址为 0000000000000000。

错误 1453 表明锁定的内存超出了系统的工作集限制。

```
16:unlock now
starting address:0000022f1db90000       size:1638400000
158
17:unlock now
starting address:0000000000000000       size:409600000
158
18:unlock now
starting address:0000022f7f610000       size:1228800000
158
19:unlock now
starting address:0000022fc89f0000       size:2048000000
158
20:decommit now
start address:0000000000000000  size:1638400000
487
21:decommit now
start address:0000022f1db90000  size:1638400000
22:decommit now
start address:0000000000000000  size:409600000
487
23:decommit now
start address:0000022f7f610000  size:1228800000
24:decommit now
start address:0000022fc89f0000  size:2048000000
25:reset now
starting address:0000000000000000       size:1638400000
87
26:reset now
starting address:0000022f1db90000       size:1638400000
87
27:reset now
starting address:0000000000000000       size:409600000
87
28:reset now
starting address:0000022f7f610000       size:1228800000
87
29:reset now
starting address:0000022fc89f0000       size:2048000000
87
请按任意键继续. . .
```

图片 16 控制台输出 2

unlock，操作大部分成功，但仍然对无效地址进行了操作。

decommit 操作，部分回收失败，返回错误码 487，地址为 0000000000000000 的区域在之前的提交阶段已经失败，因此回收自然会失败。

reset 操作返回错误码 87，出现参数无效。

```
0
dwAvailPageFile    14487a000
dwAvailPhys  1850cd000
dwAvailVirtual     7ffefd00a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1
dwAvailPageFile    141e8b000
dwAvailPhys  182252000
dwAvailVirtual     7ffe9b58a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
2
dwAvailPageFile    142e1f000
dwAvailPhys  182e40000
dwAvailVirtual     7ffe39b0a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
3
dwAvailPageFile    142af8000
dwAvailPhys  182c45000
dwAvailVirtual     7ffe2146a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
4
dwAvailPageFile    142b22000
dwAvailPhys  182bc0000
dwAvailVirtual     7ffdd808a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

图片 17 观察输出文件 out.txt 可以发现虚拟地址被正确保留。

```
5
dwAvailPageFile    142bae000
dwAvailPhys  182dac000
dwAvailVirtual     7ffd5df6a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
6
dwAvailPageFile    143272000
dwAvailPhys  18377e000
dwAvailVirtual     7ffd5df6a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
7
dwAvailPageFile    e16f9000
dwAvailPhys  18374b000
dwAvailVirtual     7ffd5df6a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
8
dwAvailPageFile    e2cbc000
dwAvailPhys  184b4a000
dwAvailVirtual     7ffd5df6a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
9
dwAvailPageFile    9953a000
dwAvailPhys  184d28000
dwAvailVirtual     7ffd5df6a000
dwLength     38
dwMemoryLoad    3d
dwTotalPageFile    46534e000
dwTotalPhy   3f134e000
dwTotalVirtual     7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

图片 18 commit **操作有成功和失败的，主要观察** PageFile **变换大小，**PAGE_GUARD **和** PAGE_NOCACHE **状态下无法** commit。

加锁也有因同样原因无法 lock 的，但 unlock 操作均正常。

```
1a
dwAvailPageFile    1c85bd000
dwAvailPhys  1866c1000
dwAvailVirtual      7ffd5df6a000
dwLength      38
dwMemoryLoad    3d
dwTotalPageFile    4e6e27000
dwTotalPhy    3f134e000
dwTotalVirtual      7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1b
dwAvailPageFile    1c84b0000
dwAvailPhys  1866c5000
dwAvailVirtual      7ffd5df6a000
dwLength      38
dwMemoryLoad    3d
dwTotalPageFile    4e6e27000
dwTotalPhy    3f134e000
dwTotalVirtual      7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1c
dwAvailPageFile    1c8732000
dwAvailPhys  18659b000
dwAvailVirtual      7ffd5df6a000
dwLength      38
dwMemoryLoad    3d
dwTotalPageFile    4e6e27000
dwTotalPhy    3f134e000
dwTotalVirtual      7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1d
dwAvailPageFile    1c8808000
dwAvailPhys  1867a5000
dwAvailVirtual      7ffd5df6a000
dwLength      38
dwMemoryLoad    3d
dwTotalPageFile    4e6e27000
dwTotalPhy    3f134e000
dwTotalVirtual      7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1e
dwAvailPageFile    1c7de8000
dwAvailPhys  185b92000
dwAvailVirtual      7ffd5df6a000
dwLength      38
dwMemoryLoad    3d
dwTotalPageFile    4e6e27000
dwTotalPhy    3f134e000
dwTotalVirtual      7ffffffe0000
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

图片 19 MEM_RESET 与 MEM_RELEASE 不同，在调用 MEM_RESET 时，最后虚拟内存不会被释放。