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。

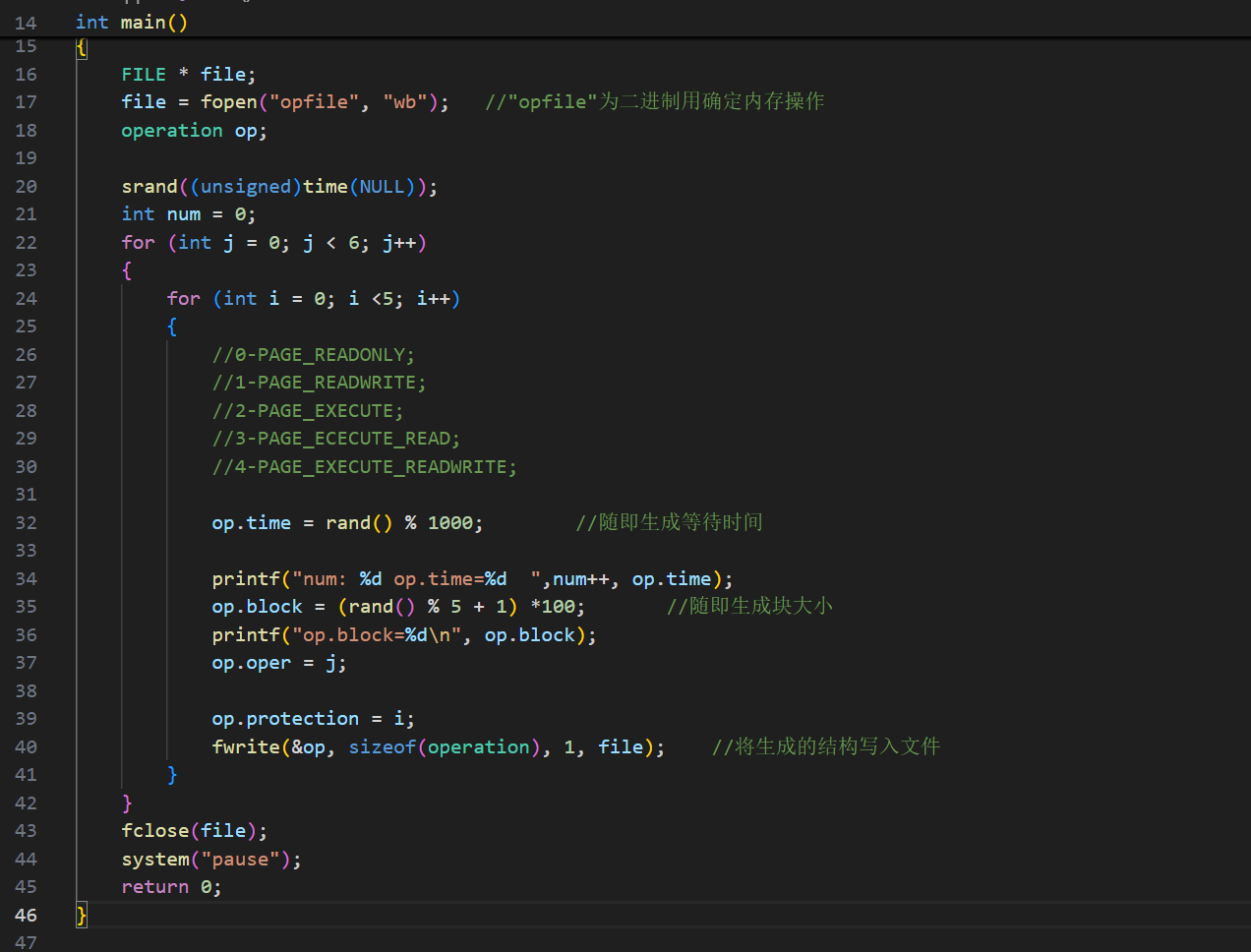
**步骤5：**按暂停按钮可暂停程序的执行，按终止按钮可终止程序的执行。



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：



图片 1 修改块大小，并且加上fclose(file)

文本

低可信度描述已自动生成

图片 2 控制台输出结果如上图所示

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

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：

文本

描述已自动生成

图片 4 运行结果1

文本

描述已自动生成

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。

表格

描述已自动生成

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

表格

描述已自动生成

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。

表格

描述已自动生成

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，

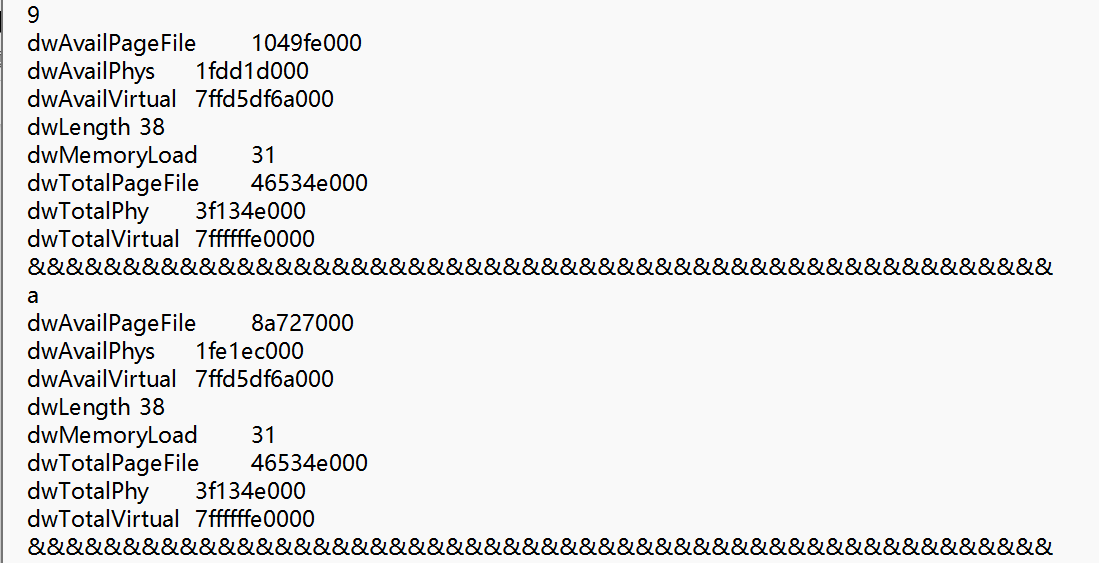
5号dwAvailPageFile– 6号dwAvailPageFile== 1,642,594,304，

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。



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。**

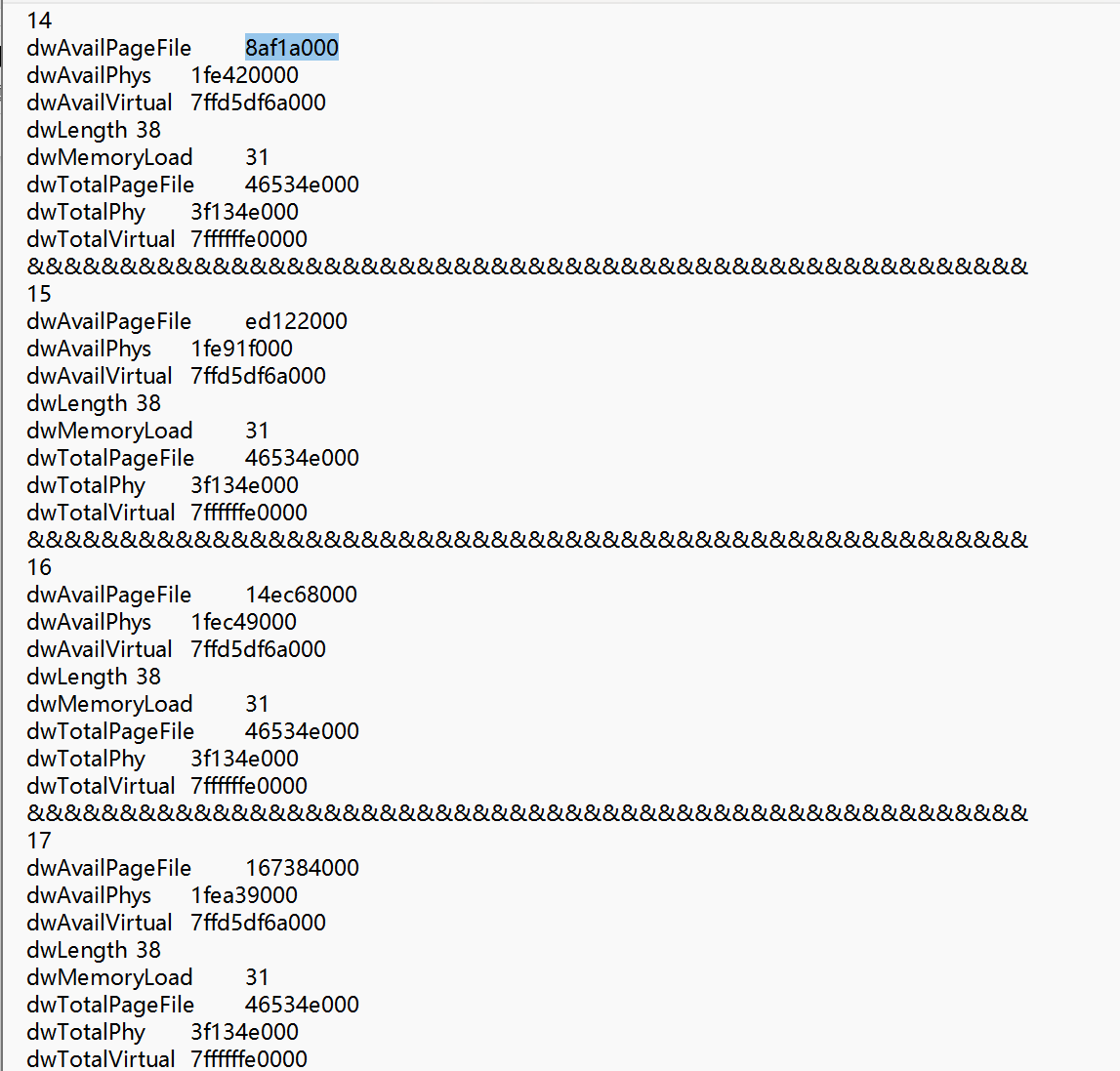
锁定内存会确保被锁定的内存始终驻留在物理内存中而不会被交换到页面文件：

* **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**。



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。

表格

低可信度描述已自动生成

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，井分析原因。

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：

文本

描述已自动生成

图片 12 修改代码将原本的flprotect参数更改为PAGE\_GUARD、PAGE\_NOACCESS、PAGE\_NOCACHE

电脑屏幕截图

描述已自动生成

图片 13 将commit更改为TOP DOWN形式

电脑萤幕的截图

描述已自动生成

图片 14 将release更改为reset

图片包含 文本

描述已自动生成

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。

文本

中度可信度描述已自动生成

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。

表格

描述已自动生成

图片 17 观察输出文件out.txt可以发现虚拟地址被正确保留。  
表格

描述已自动生成

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。

表格

描述已自动生成

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。