# FTP客户端内存泄漏分析报告

【摘要】

本报告采用VMMap内存分析工具、VLD内存泄露分析工具、UMDH内存占用分析工具以及VS内置诊断工具，对FTP客户端项目的内存占用情况、内存泄露问题以及内存优化方案进行说明。报告中将对各个分析工具的使用进行说明，分析以降低特定功能的内存使用量、排查内存占用异常上涨为目的，通过中断分析调用栈、记录内存分配比较差异的方法来进行。

【关键词】

内存泄漏、内存优化

**一、项目内存占比分析**

VMMap是对进程所用虚拟、物理内存进行分析的工具，可以统计进程所使用的虚拟、物理内存并以条形图与文字的形式进行呈现。VMMap的过滤和刷新功能可以及时观察进程内存使用的来源以及占用情况。

运行FTP客户端项目，并使用VMMap工具进行检测，可首先观察到如下界面，图1.1。

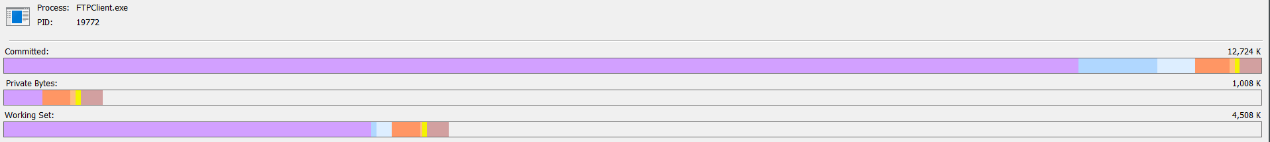
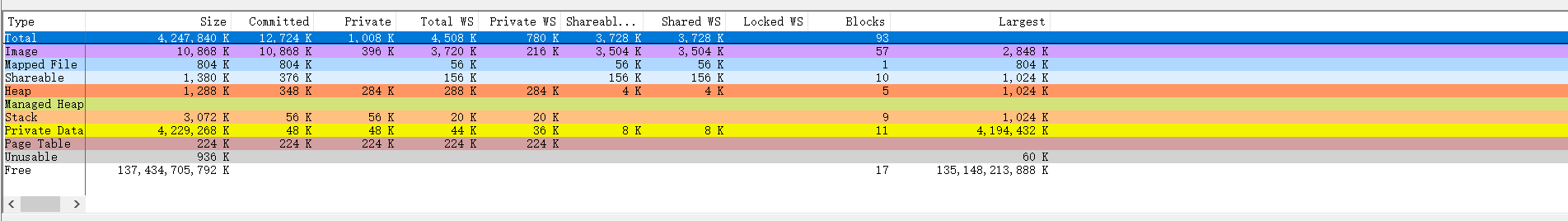
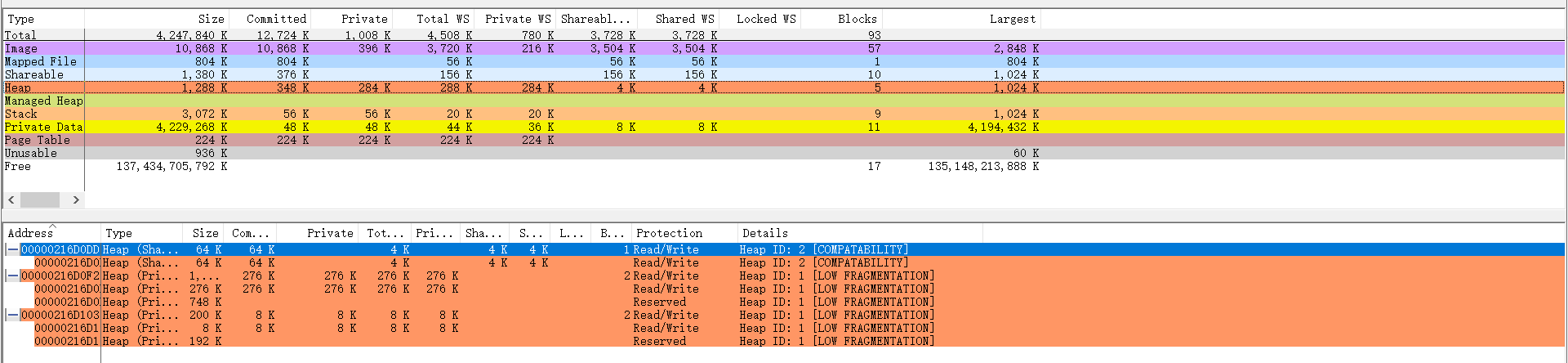


图1.1 VMMap中进程内存总体情况

其中顶部为进程基本信息，下部为各类类型空间占用条形图。Committed表示进程向操作系统所申请的需要使用到的虚拟内存大小；Private Bytes表示进程所私用使用的虚拟内存大小；Working Set表示进程目前所使用的物理内存大小。

从图中可知，FTP客户端项目启动后，虚拟内存分配空间大小为12,724K，项目所私有使用的虚拟内存空间大小为1,008K，进程所实际使用的物理内存空间大小为4,508K。

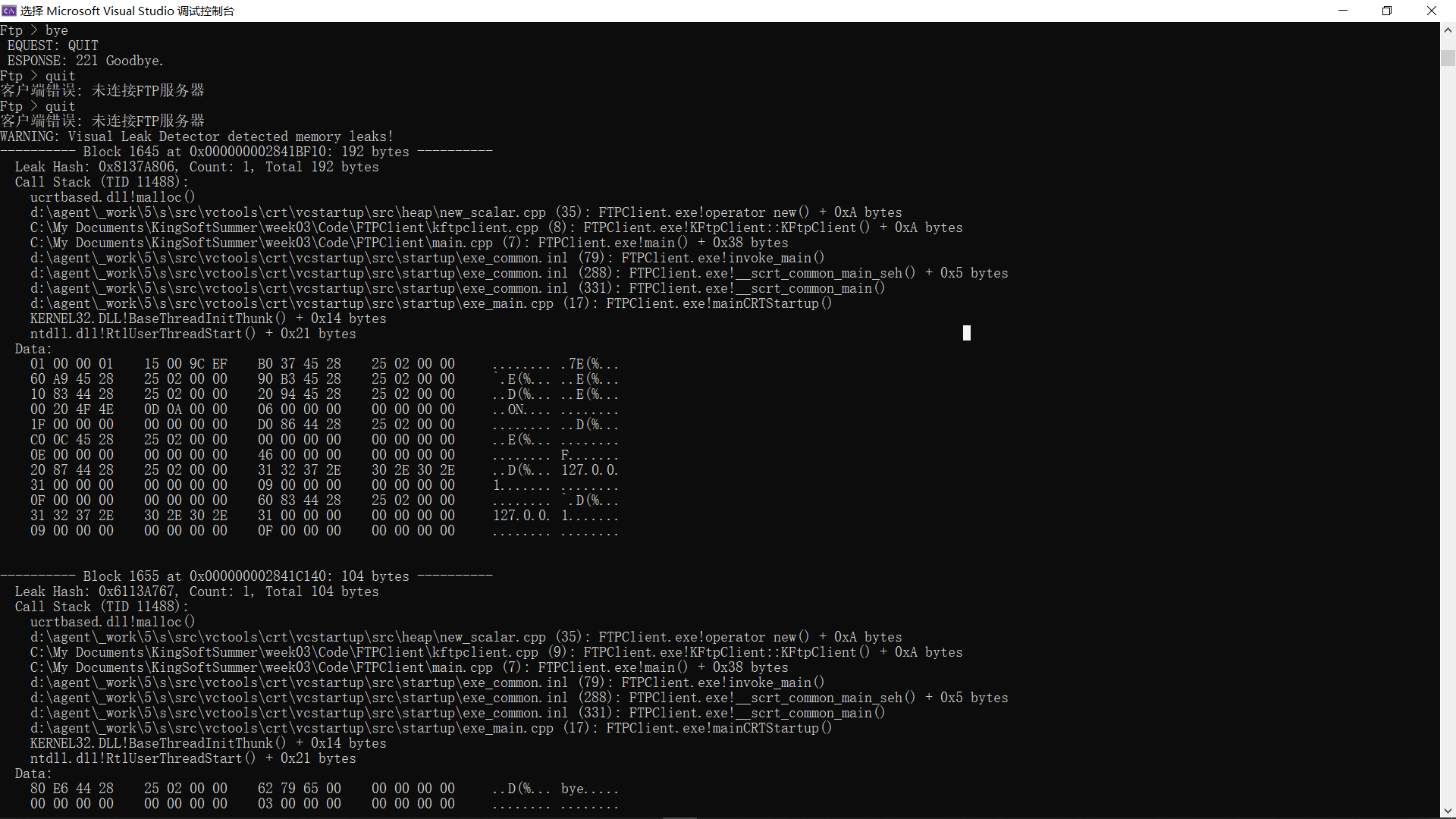
在工具的中部区域，对进程内存总体信息进行了进一步分析

横向表头分别代表总体内存大小、申请内存大小、私有内存大小，总体物理内存带下、私有物理内存大小、共享物理内存大小等类型。纵向表头代表总体、进程自身可执行文件以及包含的所有库、内存映射文件、共享内存、堆区空间、栈区空间等类型。其中new与malloc等分配的空间为堆区域，栈区域空间会主动释放，因此进一步对堆区空间进行分析。

**二、项目内存泄露点**

 Visual Leak Detector，它是一款用于 Visual C++免费的内存检测工具。VLD可以得到内存泄漏点的调用堆栈、泄露内存的完整数据，还可以设置内存泄露报告的级别。在VS项目中，加上<vid.h>头文件后运行一次程序，即可得到内存泄露分析结果。可以检测出疑似内存泄露的具体行号，new生成的对象使用完毕后未及时释放，因此发生内存泄露。

如下图所示，在在进行内存检测时候，发现了大量的内存泄漏点



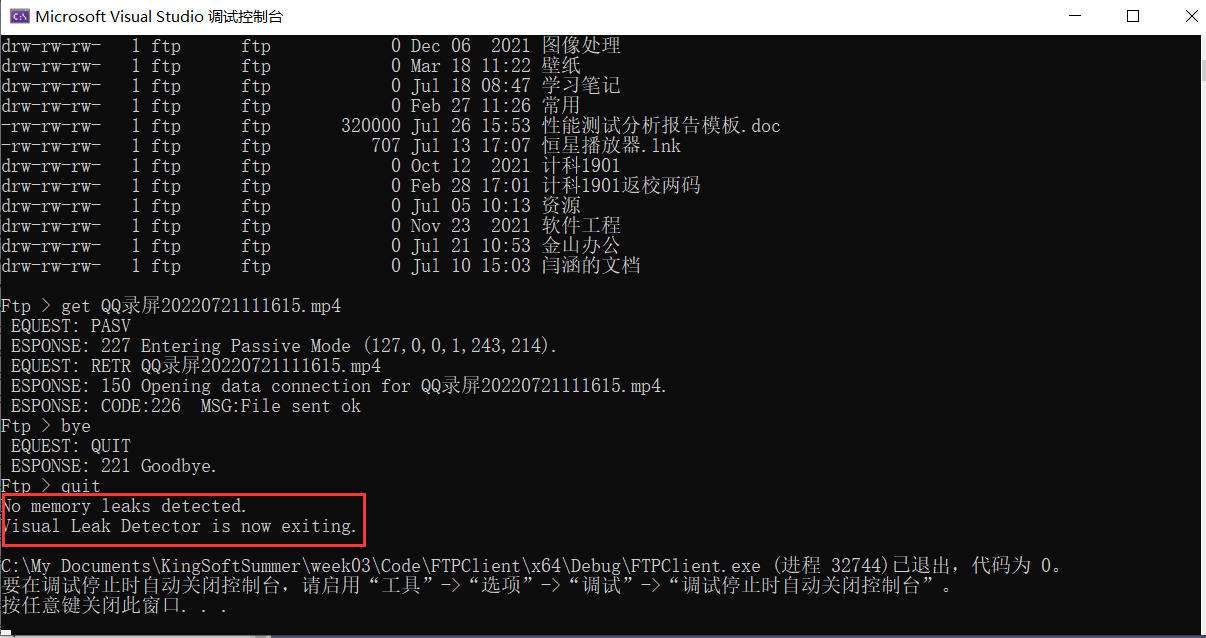
从第一个发现，是我的main函数里面，newl了一个客户端指针未释放，如下所示，导致我的对象树全都未释放代码如下：



于是直接将s\_c改为一个对象而不是指针，于是所有的内存泄漏点全部消失

如上还有三个socket关闭异常，是在释放Socket时候，重复关闭Socket导致的，于是在封装Socket的对象中加入一个flag变量，问题解决

如下所示



**三、UMDH的介绍**

UMDH（user Mode dump heap)是 Debugging Tools for Windows中附带的一个工具。该工具可以分析当前进程在堆上分配的内存，通过命令操作，可以对当前进程分配的每一块内存做日志记录。

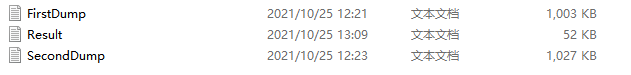


图3.1 UMDH中项目内存分析

**四、VS诊断工具的使用与项目内存分析**

在VS中，可启用自带的诊断工具对项目内存占用情况进行监控与分析，如图4.1。

执行了多次文件上传与下载命令，内存与CPU波动不大

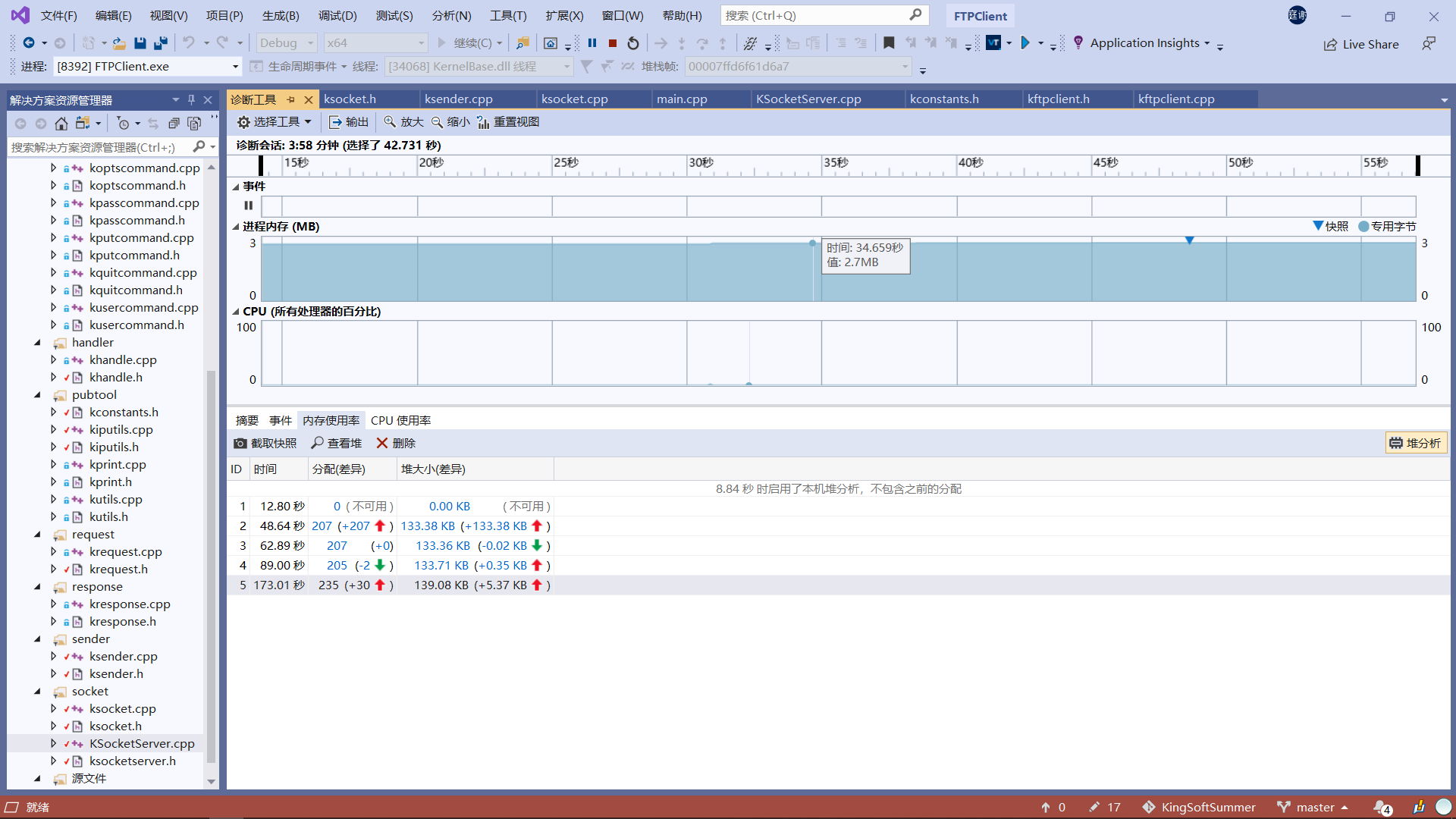
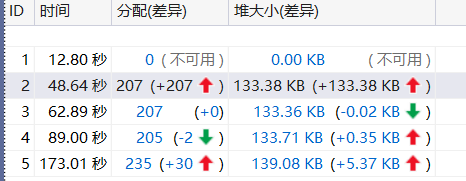


图4.1 VS诊断工具中项目内存分析

还可以通过内存快照，来截取当前时刻的内存使用情况，并对多个时刻的内存使用情况进行对比与分析，如图4.2。



**五、项目内存优化**

我们可以通过优化查找算法、使用数据压缩、使用内存池等方式对内存进行优化。要注意手动创建的对象内存使用完毕后是否释放，创建的Qt控件是否指定了父窗口，一些临时变量使用完毕后是否可以及时释放，在使用结构体等复杂的数据类型时要考虑字节对齐、字节序问题，通过调整变量声明顺序来优化内存结构。频繁的new和delete很容易导致内存碎片化，内存管理显得很重要。