


# C++内存泄露检查的5个方法

1 回复 463 查看




(<https://www.shiyanlou.com/user/8490>) 实验楼管理员  (<https://www.shiyanlou.com/vip>)

2015-10-24 14:41

技术分享 (<https://www.shiyanlou.com/questions/?tag=技术分享>)


文章整理常见定位内存泄漏的方法。

注意：我们的分析前提是Release版本。

 分享到微博

全部回答



实验楼管理员 (<https://www.shiyanlou.com/user/8490>)  (<https://www.shiyanlou.com/vip>)

(<https://www.shiyanlou.com/user/8490>)

## 一、前言

在Linux平台上有 `valgrind` 可以非常方便的帮助我们定位内存泄漏，因为Linux在开发领域的使用场景大多是跑服务器，再加上它的开源属性，相对而言，处理问题容易形成“统一”的标准。而在Windows平台，服务器和客户端开发人员惯用的调试方法有很大不同。下面结合我的实际经验，整理下常见定位内存泄漏的方法。

注意：我们的分析前提是 `Release` 版本，因为在Debug环境下，通过VLD这个库或者CRT库本身的内存泄漏检测函数能够分析出内存泄漏，相对而言比较简单。而服务器有很多问题需要在线上并发压力情况下才出现，因此讨论Debug版调试方法意义不大。

## 二、对象计数

**方法：**在对象构造时计数++，析构时--，每隔一段时间打印对象的数量

**优点：**没有性能开销，几乎不占用额外内存。定位结果精确。

**缺点：**侵入式方法，需修改现有代码，而且对于第三方库、STL容器、脚本泄漏等因无法修改代码而无法定位。

## 三、重载new和delete

**方法：**重载new/delete，记录分配点（甚至是调用堆栈），定期打印。

**优点：**没有看出

**缺点：**侵入式方法，需将头文件加入到大量源文件的头部，以确保重载的宏能够覆盖所有的new/delete。记录分配点需要加锁（如果你的程序是多线程），而且记录分配要占用大量内存（也是占用的程序内存）。

## 四、Hook Windows系统API

**方法：**使用微软的detours库，hook分配内存的系统 `Api`：`HeapAlloc/HeapRealloc/HeapFree`（`new/malloc`的底层调用），记录分配点，定期打印。

**优点：**非侵入式方法，无需修改现有文件（hook api后，分配和释放走到自己的钩子函数中），检查全面，对第三方库、脚本库等等都能统计到。

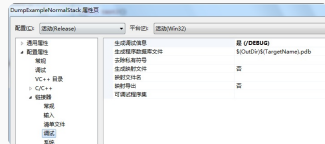
**缺点：**记录内存需要占用大量内存，而且多线程环境需要加锁。

## 五、使用DiagLeak检测

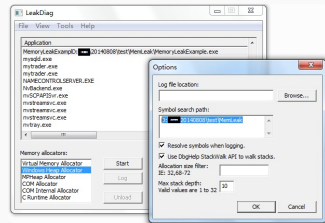
微软出品的内存泄漏分析工具，原理同hookapi方式。配合 LDGraph 可视化展示内存分配数据，更方便查找泄漏。

## 方法：

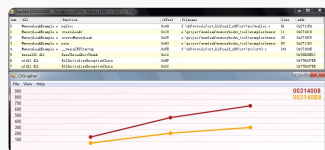
1.在IDE工程选项里面配置Release版本也生成调试信息，发布时，将pdb文件和exe文件一起发布。



2.程序运行后，打开LeakDiag，设置Symbol path



3.定期Log下目标进程的内存分配情况，通过 LDGraph 打印分配增长情况，来发现内存泄漏。



**优点：**同hookapi方法，非侵入式修改，无需做任何代码改动。跟踪全面。可视化分析堆栈一览无余！

**缺点：**对性能有影响，hook分配加锁，遍历堆栈。但是不会占用目标进程的自身内存。

## 六、总结

对于线上生产环境，建议大对象用计数来判断，定位快速准确，几乎无性能开销。在对外测试阶段，使用LeakDiag辅助分析，因为此时并发压力还不是太大，性能开销还是可以承受。在线上大规模应用阶段，通过HookApi的方法，结合GM指令控制部分时间段的检测，这样可以把对玩家的影响（服务器性能下降导致延迟）降到最低。

以上方法，我将后续的博客中逐一详细介绍，文中提到到 detours 库和 LeakDiag 工具，想了解的朋友，可以上 codeprojet（**小编注：**<http://www.codeproject.com/>）上搜索下相关介绍，如果要上google查找资料，也可以上这里（**小编注：**<http://www.gso8.com/>）。如果大家有新方法也可留言讨论，我们共同完善这一系列的介绍文章。

via:<http://www.cnblogs.com/ajie0112/p/4317697.html>

2015-10-24 14:45

登录后才能回答问题哟~

## 我要提问

### 标签

Linux (<https://www.shiyanlou.com/questions/?tag=Linux>)    Python (<https://www.shiyanlou.com/questions/?tag=Python>)

C/C++ (<https://www.shiyanlou.com/questions/?tag=C/C++>)    实验环境 (<https://www.shiyanlou.com/questions/?tag=实验环境>)

技术分享 (<https://www.shiyanlou.com/questions/?tag=技术分享>)    功能建议 (<https://www.shiyanlou.com/questions/?tag=功能建议>)

课程需求 (<https://www.shiyanlou.com/questions/?tag=课程需求>)      Java (<https://www.shiyanlou.com/questions/?tag=Java>)

其他 (<https://www.shiyanlou.com/questions/?tag=其他>)      SQL (<https://www.shiyanlou.com/questions/?tag=SQL>)

NodeJS (<https://www.shiyanlou.com/questions/?tag=NodeJS>)      Hadoop (<https://www.shiyanlou.com/questions/?tag=Hadoop>)

常见问题 (<https://www.shiyanlou.com/questions/?tag=常见问题>)      Web (<https://www.shiyanlou.com/questions/?tag=Web>)

Shell (<https://www.shiyanlou.com/questions/?tag=Shell>)      PHP (<https://www.shiyanlou.com/questions/?tag=PHP>)

Git (<https://www.shiyanlou.com/questions/?tag=Git>)      HTML (<https://www.shiyanlou.com/questions/?tag=HTML>)

HTML5 (<https://www.shiyanlou.com/questions/?tag=HTML5>)      信息安全 (<https://www.shiyanlou.com/questions/?tag=信息安全>)

网络 (<https://www.shiyanlou.com/questions/?tag=网络>)      GO (<https://www.shiyanlou.com/questions/?tag=GO>)

NoSQL (<https://www.shiyanlou.com/questions/?tag=NoSQL>)      训练营 (<https://www.shiyanlou.com/questions/?tag=训练营>)

Android (<https://www.shiyanlou.com/questions/?tag=Android>)      Ruby (<https://www.shiyanlou.com/questions/?tag=Ruby>)

Perl (<https://www.shiyanlou.com/questions/?tag=Perl>)

## 相关问题

谈Runtime机制和使用的整体化梳理 (<https://www.shiyanlou.com/questions/3010>)

JavaScript：彻底理解同步、异步和事件循环(Event Loop) (<https://www.shiyanlou.com/questions/3009>)

Github上的十大深度学习项目 (<https://www.shiyanlou.com/questions/3000>)

git基础知识整理 (<https://www.shiyanlou.com/questions/2999>)

Linux编程之内存映射 (<https://www.shiyanlou.com/questions/2992>)

## 动手做实验，轻松学IT。

实验楼-通过动手实践的方式学会IT技术。

公司简介 (<https://www.shiyanlou.com/aboutus>) 联系我们 (<https://www.shiyanlou.com/contact>) 常见问题 (<https://www.shiyanlou.com/faq#howtostart>)  
我要开课 (<https://www.shiyanlou.com/labs>) 隐私协议 (<https://www.shiyanlou.com/privacy>) 会员条款 (<https://www.shiyanlou.com/terms>)  
友情链接 (<https://www.shiyanlou.com/friends>)  
站长统计 ([http://www.cnzz.com/stat/website.php?web\\_id=5902315](http://www.cnzz.com/stat/website.php?web_id=5902315)) 蜀ICP备13019762号 (<http://www.miibeian.gov.cn/>)



QQ群



微信



微博  
(<http://weibo.com/shiyanlou2013>)