

Linux攻略：让系统内存不再泄漏的好方法 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/245/2021\\_2022\\_Linux\\_E6\\_94\\_BB\\_E7\\_95\\_c103\\_245849.htm](https://www.100test.com/kao_ti2020/245/2021_2022_Linux_E6_94_BB_E7_95_c103_245849.htm) 内存泄漏 在此，谈论的是程序设计中内存泄漏和错误的问题，不过，并不是所有的程序都有这一问题。首先，泄漏等一些内存方面的问题在有的程序语言中是不容易发生的。这些程序语言一般都认为内存管理太重要了，所以不能由程序员来处理，最好还是由程序语言设计者来处理这些问题，这样的语言有Perl、Java等等。然而，在一些语言(最典型的就C和C++)中，程序语言的设计者也认为内存管理太重要，但必需由开发人员自己来处理。内存泄漏指的是程序员动态分配了内存，但是在使用完成后却忘了将其释放。除了内存泄漏以外，在开发人员自己管理内存的开发中，缓冲溢出、悬摆指针等其它一些内存的问题也时有发生。问题缘何产生 为了让程序能够处理在编译时无法预知的数据占用内存的大小，所以程序必需从操作系统实时地申请内存，这就是所谓的动态内存。这时候，就会出现程序申请到内存块并且使用完成后，没有将其归还给操作系统的错误。更糟的情况是所获取的内存块的地址丢失，从而系统无法继续识别、定位该内存块。还有其它的问题，比如试图访问已经释放的指针(悬摆指针)，再如访问已经被使用了的内存(内存溢出)的问题。后果不容忽视 对于那些不常驻内存的程序来说，由于执行过程很短，所以即使有漏洞可能也不会导致特别严重的后果。不过对于一些常驻内存的程序(比如Web服务器Apache)来说，如果出现这样的问题，后果将非常严重。因为有问题的程序会不断地向系统申请内存，并且

不释放内存，最终可能导致系统内存耗尽而导致系统崩溃。此外，存在内存泄漏问题的程序除了会占用更多的内存外，还会使程序的性能急剧下降。对于服务器而言，如果出现这种情况，即使系统不崩溃，也会严重影响使用。悬摆指针会导致一些潜在的隐患，并且这些隐患不容易暴发。它非常不明显，因此很难被发现。在这三种存在的问题形式中，缓冲溢出可能是最危险的。事实上，它可能会导致很多安全性方面的问题(一个安全的程序包含很多要素，但是最重要的莫过于小心使用内存)。正如上面所述，有时也会发生同一内存块被多次返还给系统的问题，这显然也是程序设计上的错误。一个程序员非常希望知道在程序运行的过程中，使用内存的情况，从而能够发现并且修正问题。如何处理现在已经有了一些实时监测内存问题的技术。内存泄漏问题可以通过定时地终止和重启有问题的程序来发现和解决。在比较新的Linux内核版本中，有一种名为OOM(Out Of Memory)杀手的算法，它可以在必要时选择执行Killed等程序。悬摆指针可以通过定期对所有已经返还给系统的内存置零来解决。解决内存溢出问题的方法则多种多样。事实上，在程序运行时来解决这些问题，显然要麻烦得多，所以我们希望能够在开发程序时就发现并解决这些问题。下面介绍一些可用的自由软件。工具一：垃圾回收器(GC)在GCC(下载)工具包中，有一个“垃圾回收器(GC)”，它可以轻松检测并且修正很多的内存问题。目前该项目由HP的Hans-J.Boehm负责。使用的技术GC使用的是名为Boehm-Demers-Weiser的可以持续跟踪内存定位的技术。它的算法通过使用标准的内存定位函数来实现。程序使用这些函数进行编译，然后执行，算法就会分析程序的操

作。该算法非常著名并且比较容易理解，不会导致问题或者对程序有任何干扰。性能 该工具有很好的性能，故可以有效提高程序效率。其代码非常少并且可以直接在GCC中使用。该工具没有界面，使用起来比较困难，所以要想掌握它还是要花一些工夫的。一些现有的程序很有可能无法使用这个编辑器进行配置。此外，为了让所有的调用能被捕获，所有的内存调用(比如malloc()和free())都必须使用由GC提供的相应函数来代替。我们也可以使用宏来完成这一工作，但还是觉得不够灵活。结论 如果你希望能够有跨平台(体系结构、操作系统)的解决方案，那么就是它了。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)