ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.14, No.28,October. 2018

E-mail:kfyj@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

内存泄漏检测方法研究综述

汪明晔

(同济大学,上海 201804)

摘要:内存泄漏是一个长期困扰软件开发人员的问题,学术界针对该问题进行了长期的研究和讨论。当前定位内存泄漏位置的方法主要有静态检测、动态检测、静态检测和动态检测相结合的三种方法。本文将介绍内存泄漏的原因,分别阐述上述三种内存泄漏检测的原理和方法,分析它们各自的优势和不足,最后指明内存泄漏检测技术未来的发展方向。

中图分类号:TP311 文献标识码:A 文章编号:1009-3044(2018)28-0276-01
DOI:10.14004/j.cnki.ckt.2018.3351

内存泄漏问题的产生一般是由于软件开发设计过程留下的代码缺陷,导致程序向系统申请的堆区内存没有被回收或者没有被及时回收。内存泄漏的不断积累通常会使程序的可用内存空间减少,执行速度减慢,甚至发生无法挽回的系统故障[1]。同时,内存泄漏缺陷的存在还可能被一些针对软件系统的攻击行为(比如拒绝服务 Denial of Service)所利用和放大,对软件安全性产生十分不良的影响[2]。

为了解决内存泄漏问题,静态检测、动态检测、静态检测和动态检测相结合等方法被提出。通过这些检测方法,软件开发人员可以找到造成内存泄漏的代码位置,修复造成内存泄漏的软件缺陷。

本文将重点介绍内存泄漏检测原理和方法,围绕它们各自的优势和不足,分析每种方法的优缺点,科学预测内存泄漏检测技术未来的发展方向。

1 内存泄漏检测原理和方法

静态检测方法是在目标程序运行前对程序源码或目标代码进行静态分析,发现潜在的内存泄漏代码缺陷;动态检测方法是当目标程序运行时,收集程序内存操作信息,判断是否发生内存泄漏;为了提高动态检测方法的效率,也为了减少对目标程序运行的干扰,可以先对程序的源码做静态分析或者目标机器代码反汇编并做静态分析,预测可能发生泄漏的代码段,当程序运行时,只在这些代码段被执行时跟踪收集程序的内存操作信息,这便是静态检测和动态检测相结合的方法。

1.1 内存泄漏静态检测

静态检测原理是通过对目标程序的源码或者目标代码进行词法语法分析或解析,发现潜在的代码缺陷,报告给软件开发人员,其一般流程如图1所示:

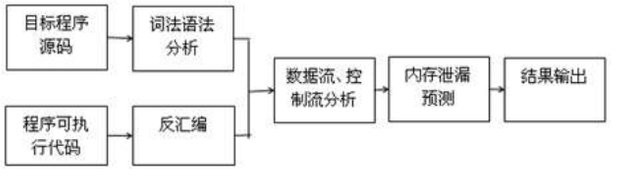

图1 内存泄漏静态检测流程

静态检测方法一般采用模式匹配预测可能发生内存泄漏的位置,具体使用哪种模式因程序的高级语言或者汇编指令而异,通过判断堆指针是否被正确释放检测是否发生内存泄漏。

1.2 内存泄漏动态检测

动态检测原理一般是通过截获目标程序的内存操作函数(如 malloc、free 等)获取所申请的内存的信息,通过判断动态申请的内存是否被释放或者在预定时间内被释放检测是否发生内存泄漏。动态检测方法的一般步骤如图2所示:

图2 内存泄漏动态检测流程

1.3 内存泄漏静态检测和动态检测相结合的方法

静态检测和动态检测相结合的方法综合了静态检测和动态检测两种方法,该方法首先使用静态检测方法确定程序可能发生内存泄漏的位置,当程序运行时,通过对这些潜在内存泄漏位置进行动态跟踪检测,该方法的一般步骤如图3所示:

图3 内存泄漏静态和动态相结合的检测方法流程

2 三种内存泄漏方法的比较结论

静态检测方法对程序的运行没有影响,但是仅通过对程序静态分析产生内存泄漏警报存在误报的情况;动态检测方法通过跟踪进程的内存操作确认内存泄漏的结果是否准确,但是会影响到程序的运行,而且存在漏报情况;静态和动态检测相结合的方法,通过静态检测减少了漏报的情况和对程序运行的干扰,通过动态检测使检测的结果更加可靠,是一种折中的方法,也是未来内存泄漏检测技术研究的发展趋势。

参考文献:

[1] 王喆. C/C++代码内存泄漏缺陷检测方法研究[D].大连理工大学, 2012.
[2] 李孟宸. 面向C语言程序内存泄漏的动态确认技术[D].南京大学, 2014.

【通联编辑:梁书】

收稿日期:2018-08-08