基于软件故障注入的嵌入式系统鲁棒性测试

王立荣
（中国船舶重工集团公司 江苏自动化研究所，江苏 连云港 222006）

摘 要：以研究对嵌入式系统鲁棒性进行评价和基于软件故障注入技术的嵌入式系统鲁棒性测试为目的。对嵌入式系统鲁棒性测试的相关概念以及软件故障注入技术原理进行了介绍，以 Linux 操作系统内核函数测试为例，通过对系统 API 参数的故障注入接口进行分析，提出基于 GDB 工具的软件故障注入方法来实现系统鲁棒性故障注入测试。完成了相应的 Linux 操作系统 API 接口故障注入测试实例并给出了测试结果。为嵌入式系统鲁棒性测试提供了更为直观、有效的方法。

关键词：鲁棒性；故障注入；软件测试；嵌入式系统
中图分类号：TP316.2 文献标识码：A 文章编号：1674-7720(2011)22-0014-03

Robustness testing of embedded software based on fault injection

Wang Lirong
（Jiangsu Automation Research Institute of CSIC, Lianyungang 222006, China）

Abstract: To study the robustness of the evaluation of embedded systems and software-based fault injection testing of embedded systems for the purpose of robustness. Test the robustness of the embedded system and software faults related concepts were introduced into the technical principles, The Linux operating system kernel function tests, for example, parameters of the system through the API interface fault injection analysis, based on GDB tools to implement software fault injection method robustness fault injection testing. Completed a corresponding API interface, Linux operating system fault injection test cases and test results are given. To test the robustness of embedded systems to provide a more intuitive and effective way.

Key words: robustness; fault injection; software testing; embedded system

嵌入式系统由于其自身的特点，在开发和测试方面的难度往往大于通用软件。传统的结构化测试方法在测试系统鲁棒性方面存在一些缺点[1]。故障注入方法作为传统测试方法的一种补充，重点测试系统在异常条件和无效参数下容错应对情况。通过注入故障的方法来测评容错机制、验证系统异常应对能力，从而提高系统鲁棒性。其最大的特点是高度灵活性，它既能通过特殊的硬件辅助设备进行硬件方法的测试，也能实现软件方法的故障注入测试。

在嵌入式系统软件测试方面，故障注入技术可以对系统的容错性、可靠性、安全性进行测试，既可以采用静态故障注入的方式，也可以采用动态故障注入的方式。本文针对嵌入式系统的结构特点和可靠性设计，利用嵌入式系统 API，在 OS 接口以及应用层实现软件故障注入，构建软件故障注入模型。通过对注入故障后系统反馈数据的分析，可实现测试和评估容错机制的目标，从而达到提高系统鲁棒性的目的。

1 嵌入式系统鲁棒性测试简介

系统鲁棒性是衡量系统在压力环境或异常输入下保持正常工作能力的一种度量[2]。嵌入式系统体系结构从底层到顶部的顺序依次是：内核（包含内核函数）、系统调用、内建程序(操作系统的命令)[3]。内核函数是内核代码的组成部分，其调用程序直接运行在内核空间。内核函数一旦出现异常，将立刻对整个操作系统产生影响。实施嵌入式系统鲁棒性测试的主要目的就是发现系统内核的薄弱环节，并予以消除或增强抵抗异常情况的能力，从而提高系统的鲁棒性。

传统的系统鲁棒性测试方法是通过观察系统的失效行为，通过分析错误记录来完成的。对一个高可靠系统而言，通过长时间实际观察来获取有关统计结果不切实际。嵌入式系统鲁棒性测试关心的是系统失效的情况，对于嵌入式系统鲁棒性的评价一般有基于测量的方法和基于故障注入的方法[4]。测试的重点在于通过在内核层面进行故障注入，人为地使系统出现故障，进而对

故障信息进行分析,最终根据测试结果生成相应的保护代码。

2 软件故障注入技术概述

故障注入技术就是按照选定的故障模型,用人工注入的方法有意识地产生故障并施加于运行的目标系统中,达到加速该系统错误和失效的产生,通过监控并采集系统对注入故障的反馈信息和现场数据,通过分析提供有关结果的测试过程[5]。一般故障注入流程如图 1 所示。

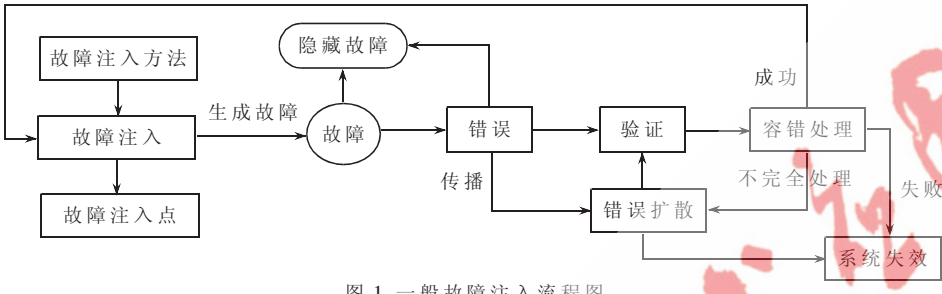

图 1 一般故障注入流程图

软件故障注入技术是通过特定的程序对系统软件、硬件错误状态进行仿真。这种方法容易测试出新的故障类型。故障注入原理就是通过修改程序执行语句,增加、修改、删除数据或直接修改寄存器或存储器的内容来模拟硬件或软件故障的发生。软件故障注入的目标可以是应用程序或操作系统。如果目标为应用程序,故障注入程序插入到应用程序中或作为应用程序和操作系统之间的一层。如果目标是操作系统,注入程序只能嵌入到操作系统中。

软件实现的故障注入技术有以下三种:

(1) 修改程序的源代码使其出现故障,通常用于变异测试;

(2) 在执行过程中修改程序某部分的数据状态,通常用于故障行为研究;

(3) 基于压力的故障注入。即通过一定的工作负载来触发系统容错行为,验证系统容错机制。

软件实现的故障注入投入低且易于控制,无需额外的硬件设备,可以在程序指令能够访问到的软硬件位置上自由选择故障注入点。作为近年来新兴实验技术手段,软件的故障注入方法有着成本低、灵活度高、可再现性等优点而被广泛采用。

3 嵌入式系统故障注入方法

嵌入式系统中用户层与内核的交互都需要通过系统接口 API 来实现,所以对系统内核的故障注入需要针对这些系统调用接口进行[6]。可采用嵌入式系统中的常用工具 GDB 进行。GDB 的全称是 GNU Debuger,是 GNU 开源组织发布的一个强大的 UNIX 下的程序调试工具。使用 GDB 工具可以让被调试程序停止在任意一个位置。在程序停止时,可查看或修改程序内存空间或寄存器的值。GDB 的功能可用于对被测试程序进行故障注入,设置断点进行故障触发,修改内存和寄存器的值来进行故障数据注入,查看内存和寄存器的值来回收故障数据等。

Ptrace()是 GDB 的基础调试工具,通过 Ptrace()可以在一个进程中观察和控制另一个进程的执行状态,主要用于执行断点调试和系统调用跟踪。在嵌入式系统下,可使用 Ptrace()系统调用来进行故障注入和故障数据采集。Ptrace()的系统调用格式如下:

```
#include <sys/ptrace.h>
Long ptrace (enum_ptrace_request request, pid_t pid, void *addr, void *data)
```

参数 pid 表示进程号,request 表示具体操作,data 表示要注入的数据,addr 指明地址。在进行具体的故障注入时,需要控制运行的进程,并对进程的上下文进行读写操作。通过 ptrace()调用可以对进程间通信进行跟踪,使得一个进程可动态地读写另一个进程的内存和寄存器值,包括代码段、数据段和堆栈段,以及访问和修改通用寄存器和专用寄存器[7]。其故障注入原理为利用 ptrace()调用追踪应用进程,对应用进程中的系统调用服务注入故障,并由 ptrace()监控应用进程故障注入结果[8]。

故障注入程序运行时,父进程执行 fork()生成子进程,子进程执行系统调用 ptrace(),使该子进程处于被追踪的状态,程序同时开辟共享内存空间并定义一个全局的故障存储数据结构,父子进程间通过信号量互斥的方法使用共享内存传送数据。父进程将 request、addr 和 data 拷贝到该数据结构后唤醒子进程并使其进入就绪状态,然后进入睡眠直到子进程响应。当子进程继续运行时,执行适当的跟踪命令,把子进程的回答写到故障存储数据结构中去,然后唤醒父进程。根据故障注入程序设计的不同,可释放子进程,也可使子进程挂起并等待新的命令,以便再次进入被追踪状态执行故障测试项。当故障注入程序继续执行时,父进程调用 ptrace()追踪测试项,将子进程提供的返回值保存起来,对故障值与参考值进行比较,如果不符即表示存在容错机制未覆盖的故障,然后将故障存储数据结构有效值返回给用户。图 2 是利用 ptrace()实现的故障注入方法流程图。

通过 Ptrace 函数就可以实现 Linux 系统的故障注入,将可能引发故障的数据注入到被测系统中,通过分析注入后的结果数据来判定程序中是否存在特定故障。

4 测试实现

随着嵌入式系统在使用过程中出现的各种问题被不断地解决,系统的鲁棒性也随之不断提高。多数嵌入式系统的开源性质也决定了其鲁棒性提升得很快[9],

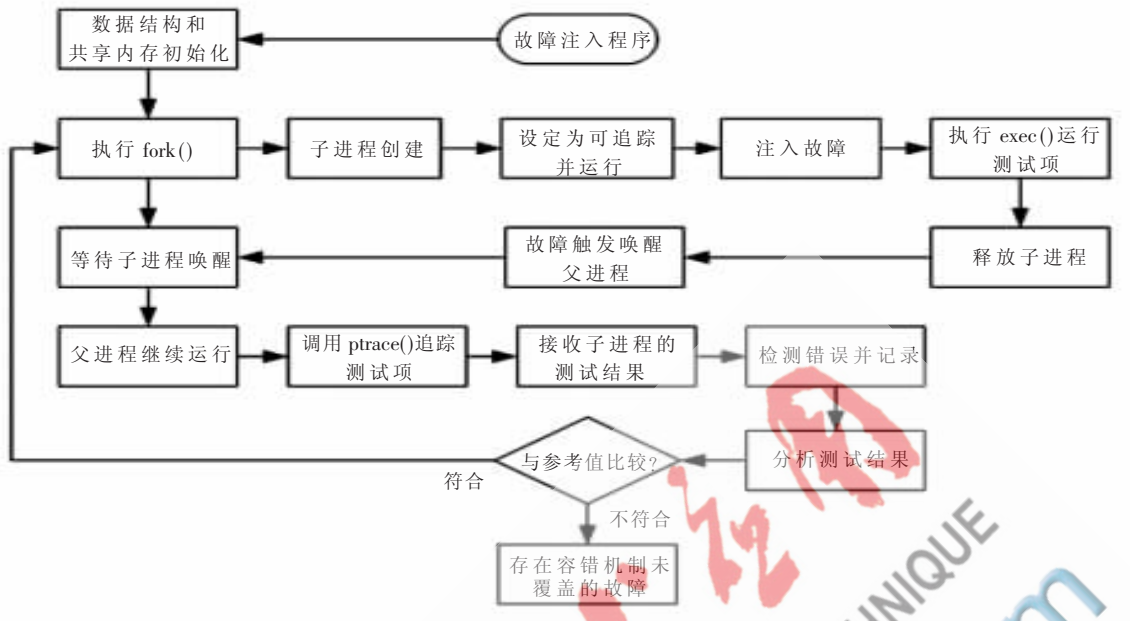

图 2 利用 ptrace()实现的故障注入方法流程图

Linux 系统便是一个很好的例子。

实验中利用故障注入方法对 Linux 内核进行 API 调用的鲁棒性测试,将其 API 划分为进程管理、进程通信、文件系统、内存管理、网络管理这五方面。测试环境:宿主机为 DELL 的 DIMENSION 4700, 操作系统为 Windows xp-sp3;目标机为友善之臂的 ARM 开发套件,系统为基于 ARM 移植的 Linux 系统,内核为 2.2.24。整个故障注入测试过程是:在宿主机上选择故障注入测试项,根据不同 API 调用对象生成所需参数,然后将上述测试信息传递至目标系统。在目标系统上,根据接口信息要求生成测试任务项。根据故障注入对象选择故障注入方法,根据接口信息中的参数进行故障注入,将收集到的测试结果返回至宿主机进行结果分析[10]。在实验中采用 GNU 的调试工具 GDB 观察目标机系统内存和寄存器的值,利用文件系统观察测试数据信息,使用串口打印信息进行调试。通过这些方法监视系统状态,并进行多次试验观察。表 1 为实验故障注入结果分析。

表 1 实验故障注入结果分析

故障类型	故障注入次数	系统失效次数	失效概率%
进程管理故障	22	2	9.1
文件系统故障	26	7	26.9
进程通信故障	18	1	5.6
内存管理故障	15	3	20
网络管理故障	24	4	16.7
合计	105	17	16.2

测试结果表明,系统 API 中进程管理和进程通信等调用失效率较低, 文件系统类型调用是故障高发部位,经过测试分析发现其较大的失效概率主要是由于高级 I/O、字符串处理和字符处理类函数的健壮性失效而引起[11]。实验通过故障注入导致系统产生失效状态,通过分析失效状态下的故障注入位置和参数信息,对系统容错机制的漏洞进行了完善,使系统内核函数的鲁棒性得到有效提升。

鲁棒性是衡量系统性能的重要指标,尽管已经有了一些测试方法和工具,嵌入式系统的鲁棒性测试仍然是一个需要完善的领域[12]。本文在分析软件故障注入技术的基础上,提出了嵌入式系统鲁棒性测试的故障注入方法,完成了基于 Linux 系统 API 接口的故障注入实验,并对实验获取的故障注入结果进行了分析。测试结果表明,使用这种测试方法降低了系统失效概率,根据测试结果生成相应的保护代码后使该嵌入式系统的鲁棒性得到进一步保证。

参考文献

[1] 刘利枚,汪文勇,唐科.嵌入式软件测试方法与技术[J].计算机与现代化,2005,116(4):124-126.

[2] 王乐春,龚正虎,陈建荣.基于错误注入技术的协议实现鲁棒性测试体系结构[J].计算机工程与应用, 2003,47(22):139-141,184.

[3] 朱鸿宇,谢余强,刘瑰.基于故障注入发现缓冲区溢出漏洞的研究[J].微计算机应用,2005,26(6):676-679.

[4] SOSNOWSKI J GAWKOWSKI P: Enhancing fault injection testbench[A].In:DepCos-RELCOMEX'06[C].2006:76-83.

[5] TANG D. Engineering oriented dependability Evaluation: MEADEP and its application[A].Fault-Tolerant Systems Proceeding IEEE[C].1997:85-90.

[6] 任献彬.测控软件的软件测试方法研究[J].计算机测量与控制,2002,10(8):547-549.

[7] 周章慧,王同洋,吴俊军,等.基于有限状态机的健壮性测试研究[J].计算机工程与科学, 2009, 31(5):93-97.

[8] HSUEH M C,TSAI T K,IYER R K. Fault injection techniques and Tools[J].IEEE Computer,1997,30(4):75-82.
[9] 王轶辰,徐萍.嵌入式软件机内测试的设计与测试[J].计算机工程,2009,35(17):34-36,39.
[10] 石君友,李郑,骆明珠,等.故障注入控制软件的设计与实现[J].测控技术,2008,27(04):65-67,70.
[11] 任献彬.测控软件的软件测试方法研究[J].计算机测量与控制,2002,10(8):547-549.
[12] 郑人杰.计算机软件测试技术[M].北京:清华大学出版社,1990.

(收稿日期:2011-04-23)

作者简介:
王立荣,女,1978年生,高级工程师,主要研究方向:软件测试。