

文章编号: 1671-0576(2020)02-0057-04

基于桩函数动态赋值的软件覆盖率测试方法

万东燕, 罗睿梅, 李俊山, 陈 铨

(上海无线电设备研究所, 上海 201109)

摘 要: 针对软件单元覆盖率测试中需多次变更被测函数输入激励的需求, 以 Testbed 为测试平台, 结合变量的变化特性, 提出一种输入变量激励可动态更新的桩函数赋值方法。实验结果表明: 该方法有效解决了输入激励变化的被测函数测试覆盖率低的问题, 提高了用例设计的准确性和代码测试覆盖的充分性。

关键词: 单元测试; 测试用例; 桩函数; Testbed

中图分类号: TP311

文献标志码: A

DOI: 10.3969/j.issn.1671-0576.2020.02.010

Software Coverage Test Method Based on Dynamic Assignment of Pile Function

WAN Dong-yan, LUO Rui-mei, LI Jun-shan, CHEN Cheng

(Shanghai Radio Equipment Research Institute, Shanghai 201109, China)

Abstract: In view of the demand of changing the input excitation of the tested function for many times in software unit coverage test, taking Testbed as the test platform and combining the variable variation characteristics, a method of pile function assignment with dynamic updating of input variable excitation is proposed. The experimental results show that the method can effectively solve the problem of low test coverage of the function under test with variable input excitation. The accuracy of test case design and the sufficiency of code test coverage are improved.

Key words: unit test; test case; pile function; Testbed

0 引言

软件测试^[1]是软件研发过程中的一个重要环

节, 通过软件测试可以及时发现软件缺陷, 保障软件的安全性和可靠性。单元测试是软件开发过程中最基础的测试项, 是软件的独立单元在与程序的其他部分相隔离的情况下进行的测试。其中代码覆盖率能够体现代码的执行情况。测试人员可以通过覆盖率指标, 分析代码执行范围, 完善测试用例, 有效提高软件测试质量。

收稿日期: 2020-03-28

作者简介: 万东燕(1992—), 女, 硕士, 助理工程师, 主要从事嵌入式软件测试技术研究。E-mail: dongyan.jsu@gmail.com

在软件测试的过程中,执行单次的测试用例时,某些函数的输入变量激励会在代码执行过程中发生动态变化。若测试用例中的输入变量激励不能对应进行动态更新,则需激励更新才能进入的相关语句无法完成测试,代码覆盖率测试也无法完成。为了解决这一问题,测试人员需要在设计的测试用例中模拟输入变量激励的变化条件,来完成测试。本文结合专业的测试工具,研究了变量激励可动态更新的桩函数赋值法,并设计灵活的测试用例,以保障测试质量、提高测试效率。

1 软件的单元测试

1.1 单元测试方法

单元测试的主要目的是验证软件单元是否满足详细设计规格说明,发现需求和设计中的错误。测试用例是影响单元测试效率和效果的重要因素^[2]。本文采取黑盒与白盒结合的方法对软件进行测试。利用黑盒测试方法中的等价划分法和边界值分析法来设计测试用例。等价划分法^[3]基于输入条件的等价类评估,在每个等价类中选取有代表性的测试用例,从而达到等价类覆盖的要求。由于程序错误常在定义域和等价类的边缘被发现,所以需要每个测试变量进行边界值分析。边界值分析^[4]是等价类划分的精炼,即选择每个类的边界或终点进行测试。

覆盖率是软件开发过程中重要的度量指标,具体包括语句覆盖、分支覆盖、条件覆盖等。覆盖率测试^[5]作为一种白盒测试,其主要作用包括:
a) 评估测试质量,根据软件代码的覆盖率报告,发现漏洞场景,为测试人员及时补充新的测试用例提供指导;
b) 帮助识别冗余代码,对程序代码进行优化与重构。

1.2 单元测试工具

Testbed 是一款专业的软件测试工具,可用于生成和维护测试脚本,编译并运行测试可执行程序,查看测试结果和覆盖率数据。其中,Testbed/TBrun 模块可执行软件的单元测试^[6]。该模块可以在运行测试用例对软件功能进行测试的同时,分析得出软件单元相应代码的测试覆盖情况,评估测试状态,从而为设计新的测试用例提供指导。

2 测试用例的设计

2.1 测试用例生成基础

(1) 单元测试环境

为了方便对嵌入式软件进行单元测试,在测试前首先要去除代码中的硬件相关部分,将其移植到 Testbed 相应的测试环境中。移植的过程不会影响软件的功能,测试环境与现实环境差异不大,能够确保测试的有效性和可靠性。

单元测试的对象是不可独立运行的函数单元。测试时需要建立两种辅助测试模块:驱动模块(driver)和桩模块(stub)。驱动模块用于模拟被测函数的上层模块,桩模块用于模拟被测函数在执行过程中所调用的其他函数,以保证被测单元能完整闭合地运行。在单元测试中,被测函数可能调用多个其他函数,故可能存在多个桩函数。

(2) 桩函数

桩函数是根据测试需要编写的,用于替代被测函数调用的函数。桩函数有两种情况:一是该桩函数在被测函数中不需要返回特定值,不需要作其他处理;二是该桩函数在被测函数中需要返回特定值,不同返回值会影响代码运行与输出结果,需要根据测试需求,在 Testbed/TBrun 模块中进行相应的返回值设置。

在桩函数中设置返回值时,通常只支持输入变量激励的一次性静态赋值,在代码的运行过程中不能进行修改。由于这一局限,导致无法直接在单次执行过程中需多次变更输入变量激励的函数进行测试。同时这一问题会在覆盖率测试中反映,由于对应的分支和语句无法被执行,所以覆盖率测试也无法完全完成。为了解决这一问题,需要分析进入不同分支的变量激励的更新要求,并设计可动态更新的桩函数返回值,来模拟进入相应分支的条件,完成测试。

以某数字信号处理软件为例,测试其误差处理函数 DJDMBJC 时,需要调用函数 FindMaxPoint 来获取并返回最大值变量 max_data。在测试软件运行过程中,函数 FindMaxPoint 中变量 max_data 的值会在执行过程中动态更新,软件通过判断变量值所在范围,进入相应程序分支,相关程序流程如图 1 所示。

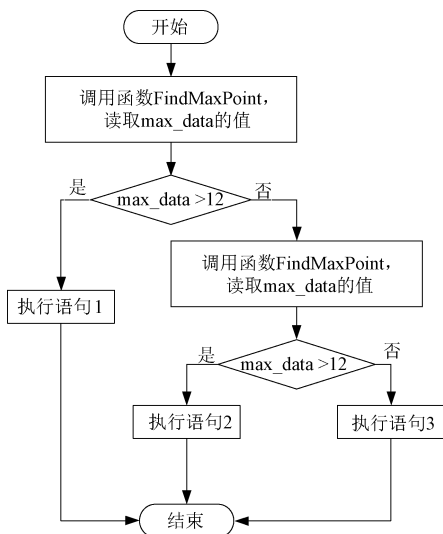


图1 误差处理函数测试程序流程图

由图1可知,该段代码调用了两次函数 Find-MaxPoint,并读取 max_data 的值,判断该值并进入对应的执行语句。对函数 DJDMBJC 进行测试时,若要到达第二个判断框,则需在第一个判断处进入“否”分支,即满足 $\max_data \leq 12$ 。在该前提下,若 max_data 的值无法更新,则只能进入第二个判断框的“否”分支,而无法进入与“执行语句2”对应的“是”分支,无法对该段代码完成测试覆盖。因此,在测试中,需要研究桩函数返回值的设置方法,实现输入变量激励的动态更新。

2.2 桩函数返回值赋值法

为了解决2.1节中由于被测单元函数所调用桩函数的输入变量激励值无法更新,导致覆盖率测试不能完成的问题,提出一种桩函数返回值动态赋值方法。根据 Testbed/TBrun 模块中桩函数的代码运行方式,在桩函数的后台代码段中构造并声明一个整型变量 count,用来记录不同赋值点的位置。位置 i 对应的返回值 a_i 即为该时刻的变量值,当函数运行到 count 值所对应的位置 i 时,将返回值 a_i 赋值给变量 max_data,从而实现被调用的桩函数输入变量激励的动态更新。具体实现过程如图2所示,其中 n 代表变量激励动态更新的次数。

将上述方法应用于2.1节函数 DJDMBJC 的测试用例设计。执行改进后的测试用例,将两次调用桩函数后的变量 max_data 激励值从 11 更新

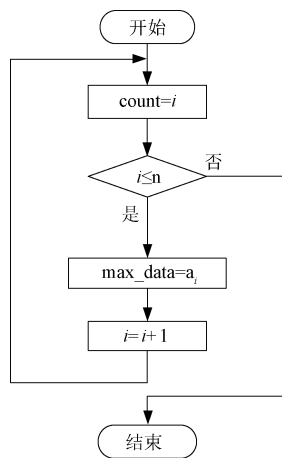


图2 桩函数返回值动态赋值程序流程图

为 13,即可先进入图1中的第一个“否”语句,再进入第二个“是”语句,实现不同分支的全部执行。

2.3 测试用例的设计实现

本文被测软件中,变量 max_data 的取值范围为 $0 \sim 25$ 。由判定条件 $\max_data > 12$ 可知,12 为判断边界值,即该变量的边界值为 0,12 和 25。对变量进行等价类与边界值分析,结果如表1所示。

表1 等价类与边界值分析

输入等价类	max_data 取值
有效等价	$[0, 12] \& (12, 25]$
无效等价	$(-\infty, 0) \& (25, +\infty)$
边界值	$-1, 0, 1; 11, 12, 13; 24, 25, 26$

由图1可知,当 $\max_data \leq 12$ 时,程序会进入第一个“否”分支,然后对已更新的变量 max_data 值进行第二次判断。因此,在进行边界值测试时,要考虑两次函数调用时变量的判断边界。结合变量 max_data 的判断边界,对测试用例进行覆盖有效等价类和边界值测试与覆盖无效等价类和边界值测试分类,实现程序边界的完整分析与覆盖,测试用例设计如表2所示。其中 max_data_1 和 max_data_2 分别代表第一和第二个判断框对应的 max_data 的值。通过对两处的 max_data 进行分类赋值,即可达到函数功能以及语句与分支的全部覆盖。

执行改进的测试用例前后,函数 DJDMBJC 的测试覆盖率如表3所示。在函数 DJDMBJC 的测试过程中,执行改进的测试用例,语句与分支判

表 2 测试用例表

分类	用例号	输入数据	期望结果
覆盖有效等价类和边界值	1	max_data_1=0 max_data_2=0,1,11,12,13,24,25	成功执行
	2	max_data_1=1 max_data_2=0,1,11,12,13,24,25	成功执行
	3	max_data_1=11 max_data_2=0,1,11,12,13,24,25	成功执行
	4	max_data_1=12 max_data_2=0,1,11,12,13,24,25	成功执行
	5	max_data_1=max_data_2=13	成功执行
	6	max_data_1=max_data_2=24	成功执行
	7	max_data_1=max_data_2=25	成功执行
覆盖无效等价类和边界值	8	max_data_1=max_data_2=-1	不执行
	9	max_data_1=0,1,11,12 max_data_2=-1	不执行
	10	max_data_1=0,1,11,12 max_data_2=26	不执行
	11	max_data_1=max_data_2=26	不执行

定的覆盖率分别由 90%、93% 提高到 100%，解决了函数测试分支无法到达的问题，实现了该函数的覆盖率测试，提升了测试质量。

表 3 函数 DJDMBJC 测试覆盖率 %

测试内容	语句	分支
改进前	93	90
改进后	100	100

3 结论

本文以 Testbed 软件为测试工具，提出了变量激励可动态更新的桩函数返回值设计方法，解决了单元测试中语句和分支覆盖率测试不全的问题。结合等价类划分与边界值测试，设计全面且高效的测试用例，有效地精简了测试用例个数，提高了测试用例设计的准确性和测试效率，对提升

测试质量具有重要的积极意义。

参考文献

- [1] 单锦辉, 姜瑛, 孙萍. 软件测试研究进展[J]. 北京大学学报, 2005, 41(1): 134-145.
- [2] 孙康. 基于国家军用标准的软件测试研究[J]. 航空标准化与质量, 2018(6): 23-26.
- [3] 赵浩强. 基于等价类划分的测试用例设计在软件评测中的应用[J]. 软件, 2017, 38(6): 91-94.
- [4] 刘高军, 钱程. 基于边界值分析法的 CNONIX 标准测试用例生成方法研究[J]. 工业技术创新, 2015, 2(2): 228-233.
- [5] 姜文, 刘立康. 基于持续集成的 C/C++ 软件覆盖率测试[J]. 计算机技术与发展, 2018, 28(3): 37-41, 46.
- [6] 张新玉, 李寅龙. TBrun 在航天嵌入式汇编软件测试中的应用[J]. 测控技术, 2016, 35(11): 103-107.
- [7] 韩晓东, 梁志恒, 颜漳. 多散射点雷达目标模型的射频仿真模拟[C]// 全国第五届信号和智能信息处理与应用学术会议专刊: 第一册. [出版地不详]: 中国高科技产业化研究会信号处理专家委员会, 2011: 249-254.
- [8] 李荣环. 数字阵列雷达扩展目标回波模拟技术研究[实现][D]. 南京: 南京理工大学, 2017.
- [9] 王锐, 翟刚毅, 许树军, 等. 一种基于 VPX 标准的侦测雷达信号处理系统设计方法[J]. 雷达与对抗, 2016, 36(2): 22-25, 29.
- [10] 张锡熊, 陈方林. 雷达抗干扰原理[M]. 北京: 科学出版社, 1982: 20-40.

(上接第 56 页)