



机器人 操作系统 测试技术

黄 明 郭书杰 梁 旭 魏洪兴 著

3



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

机器人操作系统 测试技术

黄明 郭书杰 梁旭

魏洪兴 著



電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

目前, 机器人操作系统测试技术的研究成果比较分散, 还没有专著出版。本书是在国家“863 计划”重点课题“机器人操作系统、软件仓库及模块化硬件平台开发”的支持下, 以 RGMP-ROS 操作系统的测试需求为牵引, 以解决测试中出现的难点为目标, 展开相关的研究工作。根据 RGMP-ROS 操作系统的特点, 分析了对 RGMP-ROS 进行测试时面临的难点问题。在对前人研究成果进行深入学习和总结的基础上, 结合 RGMP-ROS 机器人操作系统的特点, 提出了一系列解决这些难点问题的方案, 并对方案的先进性及可行性进行了实验验证。

本书可作为计算机、自动控制、人工智能、机器人等专业的研究生及高年级本科生教材, 也可作为从事机器人设计与开发、机器人模块化、机器人应用等相关领域研究人员和工程技术人员的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

机器人操作系统测试技术/黄明等著. —北京: 电子工业出版社, 2015.10

ISBN 978-7-121-27307-0

I. ①机… II. ①黄… III. ①机器人—操作系统—测试技术 IV. ①TP24

中国版本图书馆 CIP 数据核字 (2015) 第 231527 号

策划编辑: 赵 平

责任编辑: 周宏敏

印 刷: 北京季峰印刷有限公司

装 订: 北京季峰印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 720×1 000 1/16 印张: 8.25 字数: 185 千字

版 次: 2015 年 10 月第 1 版

印 次: 2015 年 10 月第 1 次印刷

定 价: 68.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

为了解决制约我国机器人产业发展中面临的技术难题，多家高校和科研单位共同开展了机器人模块化、网络化、标准化开放平台关键技术的研究工作，提出了一套解决机器人模块化开发的技术方案。研制了一套基于 ROS 的具有高度实时性、确定性、可裁剪和可配置的混合实时操作系统 RGMP-ROS 机器人操作系统。

作为一种通用的机器人操作系统，RGMP-ROS 的研制目的是解决机器人模块化问题，从而提高机器人系统的设计和开发效率，缩短机器人产品的设计与开发周期，降低开发成本，有效地促进我国机器人产业的发展。RGMP-ROS 能否得到有效推广和广泛使用是评判其成功与否的重要标准，由于人们对机器人安全性和可靠性的关注度越来越高，因而安全性和可靠性是决定其能否被广泛使用的主要因素。对该操作系统进行充分而有效的测试，是保证其可靠性的重要手段。因而，研究适合 RGMP-ROS 操作系统使用的测试技术并找到有效的测试方案，具有重要的理论价值和现实意义。

本书由 8 章组成。第 1 章为绪论，主要介绍了研究背景及意义、主要研究内容、操作系统测试中相关问题的研究现状、主要工作以及本书的结构等。研究背景及意义部分介绍了 RGMP-ROS 机器人操作系统的由来以及对其进行测试的重要性。主要研究内容部分介绍了对 RGMP-ROS 进行测试时面临的四方面的难点：如何高效准确地检测其中的资源问题；在进行回归测试时，如何能够选出对改动模块覆盖较好的测试用例；如何生成更小规模的组合测试用例集，以便既能对操作系统进行充分测试又不必花费太长时间；该如何对 RGMP-ROS 进行实时性测试。相关问题的研究现状部分介绍了资源问题检测、回归测试用例选择、组合测试用例生产、操作系统实时性能测试等几个方面的研究现状。主要工作部分介绍了对 RGMP-ROS 进行测试的四个难点解决方案：基于完全路径分析的资源问题检测方案；基于改进和声搜索算法的测试用例选择方案；基于 one-test-at-a-time 策略的差分进化算法的组合测试用例集生产方案；基于混合负载的操作系统实时性能测试方案。这四个解决方案分别在第 3、4、5、6 章中给

出详细介绍。

第2章为机器人模块化开发平台，是背景知识介绍，首先介绍了机器人模块化开发平台使用的技术方案：通过研制机器人专用操作系统实现机器人开发过程中的软硬件分离；建立机器人软件仓库，将典型的机器人软件模块存入软件仓库中供开发人员使用，以便降低机器人软件开发难度，提高代码复用率和软件开发效率；将典型机器人常用的硬件模块化，并使之支持 RGMP-ROS。然后以 RGMP-ROS 为重点，逐一介绍了机器人模块化开发平台的各个主要组成部分。

第3章为资源问题检测方案，介绍了本书的第一项主要工作，即机器人操作系统中资源问题检测方法。首先介绍了常见的资源问题及其发生原因。然后介绍了常用的资源问题检测方法：动态分析方法和静态分析方法，分析了选择静态分析方法来进行资源问题检测的原因。建立了C语言程序模型，通过该模型可以对待测代码进行大幅压缩，从而提高检测效率。基于C语言程序模型，建立了资源问题错误模型和检测模型，将源代码压缩后转换为资源路径图，进而转换为资源路径树和资源路径表，通过对资源路径表的遍历，依据资源问题检测模型就可以得到检测结果。

第4章为回归测试用例优化选择方案，介绍了本书的第二项主要工作，即机器人操作系统回归测试时有效测试用例选择方法。首先介绍了回归测试用例选择的相关概念以及选出的测试套件优劣的评判标准。建立了测试用例选择问题的数学模型，通过该模型，将回归测试用例选择问题转换为一个等式约束的最优化问题。接下来介绍了经典和声搜索算法，分析了算法的弱点。针对经典和声搜索算法的不足，提出了附带优秀和声库的和声搜索算法，介绍了算法的基本步骤以及在测试用例选择问题中的应用。通过实验验证了改进算法的可行性。

第5章组合测试用例集生成方案，介绍了本书的第三项主要工作，即一种规模更小的组合测试用例集生成方法。首先介绍了组合测试和差分进化算法的相关概念。然后介绍了如何将差分进化算法应用于求解组合测试用例集的生成，介绍了基于差分进化算法的组合测试用例集生产方案，并通过实验考查了算法参数对生成结果的影响，给出了相对较好的参数取值方案。

第6章为操作系统实时性能测试方法，介绍了本书的第四项主要工作，即一种通用的实时性能测试方案。首先分析了影响操作系统实时性能的因素：时间管理、任务管理、中断管理、同步与通信机制以及内存管理，从中选出任务切换时间、任务抢占时间、中断响应时间、信号量混洗时间等4个主要因素为测试对象。

制定了测试时的计时方案，并通过校准程序的引入来屏蔽硬件性能对测试结果的影响。最后给出了测试指标及其测试方法和测试评分标准。

第7章为 RGMP-ROS 操作系统测试，介绍了对 RGMP-ROS 操作系统的测试方案、测试方法及相关测试结果。

第8章为结论与展望，对主要工作进行了总结，并对下一步工作做出了展望。

本书可作为计算机、自动控制、人工智能、机器人等专业的研究生及高年级本科生教材，也可作为从事机器人设计与开发、机器人模块化、机器人应用等相关领域研究人员和工程技术人员的参考书。

本书由黄明、郭书杰、梁旭、魏洪兴著，陶宁也参加了部分编写工作。

由于作者水平有限，编写时间仓促，书中错误和不妥之处在所难免，恳请读者和专家批评指正。

作 者

2015 年 6 月

1.2 研究内容	3
1.2.1 RGMP-ROS 的特点	3
1.2.2 RGMP-ROS 的测试特点	3
1.2.3 研究内容	5
1.3 相关领域的研究现状	5
1.3.1 操作系统资源调度策略研究现状	5
1.3.2 回归测试用例选择研究现状	6
1.3.3 组合测试用例生成研究现状	10
1.3.4 操作系统实时性能测试研究现状	10
1.4 主要工作	11
1.5 本章小结	13
第2章 机器人模块化开发平台	14
2.1 技术方案	14
2.2 机器人实时操作系统 RGMP-ROS	15
2.2.1 ROS	15
2.2.2 混合操作系统框架 RGMP	16
2.2.3 RGMP-ROS 机器人操作系统	17
2.2.4 基于 RGMP-ROS 实时应用	18
2.3 机器人软件仓库	19
2.4 机器人模块化控制硬件平台	20
2.4.1 基于 X86 机器人控制器平台	20
2.4.2 基于 ARM 的硬件平台	20
2.4.3 一体化机器人模块化关节	21

目 录

第 1 章 绪论	1
1.1 研究背景及意义	1
1.1.1 研究背景	1
1.1.2 研究意义	2
1.2 研究内容	3
1.2.1 RGMP-ROS 的特点	3
1.2.2 RGMP-ROS 的测试难点	3
1.2.3 研究内容	5
1.3 相关领域的研究现状	5
1.3.1 操作系统资源问题检测研究现状	5
1.3.2 回归测试用例选择研究现状	6
1.3.3 组合测试用例集生成研究现状	10
1.3.4 操作系统实时性能测试研究现状	10
1.4 主要工作	11
1.5 本章小结	13
第 2 章 机器人模块化开发平台	14
2.1 技术方案	14
2.2 机器人实时操作系统 RGMP-ROS	15
2.2.1 ROS	15
2.2.2 混合操作系统框架 RGMP	16
2.2.3 RGMP-ROS 机器人操作系统	17
2.2.4 基于 RGMP-ROS 实时应用	18
2.3 机器人软件仓库	19
2.4 机器人模块化控制器硬件平台	20
2.4.1 基于 X86 机器人控制器平台	20
2.4.2 基于 ARM 的硬件平台	20
2.4.3 一体化机器人模块化关节	21

2.5	全模块化机器人样机	21
2.6	本章小结	22
第 3 章	资源问题检测方案	24
3.1	研究背景	24
3.1.1	资源问题的种类	24
3.1.2	资源问题的检测方法	25
3.2	基于路径分析的资源问题检测方案	25
3.2.1	代码模型	25
3.2.2	错误模型	27
3.2.3	检测模型	28
3.2.4	检测方案	29
3.2.5	具体检测过程	31
3.3	方案可行性实验	35
3.3.1	实验设计	36
3.3.2	实验结果分析	37
3.4	本章小结	38
第 4 章	回归测试用例优化选择方案	39
4.1	测试用例选择	39
4.1.1	基本概念	39
4.1.2	测试套件评价标准	40
4.1.3	测试用例选择问题的数学模型	40
4.1.4	测试用例选择问题的实例	41
4.2	和声搜索算法	41
4.3	基于 EELHS 的测试用例选择	43
4.3.1	附带优秀和声库的和声搜索算法	43
4.3.2	基于 EELHS 的测试用例选择问题解决方案	46
4.4	算法参数取值实验	47
4.4.1	方案设计	48
4.4.2	实验结果	48
4.4.3	结果分析	53
4.5	方案可行性实验	53
4.5.1	方案设计	54
4.5.2	实验结果	54
4.5.3	结果分析	55

4.6	本章小结	55
第 5 章	组合测试用例集生成方案	57
5.1	组合测试	57
5.2	生成方案	59
5.2.1	方案概述	59
5.2.2	差分进化算法	59
5.2.3	数据存储结构	60
5.2.4	组合覆盖评价方案	61
5.2.5	测试用例集生成流程	62
5.3	相关实验	63
5.3.1	实验数据表述方式	63
5.3.2	DE/rand/1 与 DE/best/1 变异机制的性能比较	63
5.3.3	算法参数对性能的影响	64
5.3.4	与其他方法的比较	67
5.4	本章小结	68
第 6 章	操作系统实时性能测试方案	69
6.1	影响操作系统实时性能的因素	69
6.1.1	时间管理功能及其对实时性能的影响	69
6.1.2	任务管理功能及其对实时性能的影响	69
6.1.3	中断管理功能及其对实时性能的影响	70
6.1.4	同步与通信机制及其对实时性能的影响	71
6.1.5	内存管理及其对实时性能的影响	72
6.2	基于混合负载的操作系统实时性能测试方案	72
6.2.1	计时方法	72
6.2.2	屏蔽硬件性能对测试结果的影响	73
6.2.3	测试指标及其测试方法	74
6.2.4	测试评分标准	81
6.3	测试结果	82
6.3.1	响应性的测试结果	82
6.3.2	确定性的测试结果	82
6.4	本章小结	83
第 7 章	RGMP-ROS 操作系统测试	84
7.1	测试方案	84
7.1.1	静态测试	84

7.1.2	单元测试	85
7.1.3	集成测试	88
7.1.4	系统测试	90
7.2	测试过程	91
7.2.1	功能测试	91
7.2.2	性能测试	99
7.2.3	回归测试	106
7.3	测试结果及分析	106
7.3.1	代码静态测试结果	106
7.3.2	动态测试结果	108
7.3.3	回归测试	108
7.3.4	性能测试结果	109
7.4	测试过程有效性评价	109
7.5	本章小结	110
第 8 章	结论与展望	111
8.1	结论	111
8.2	展望	112
参考文献		113

第1章 绪 论

1.1 研究背景及意义

1.1.1 研究背景

机器人技术是一种以自动化技术和计算机技术为主体、有机融合各种现代信息技术的系统集成和应用。当前，我国机器人产业的发展主要面临以下问题。

(1) 经过半个多世纪的发展，机器人技术已相对成熟，各类机器人被广泛应用于工业、军事、医疗、教育及家庭服务等多个领域。由于不同应用领域的独特需求，机器人产品正朝着多品种、多规格、小批量、复杂化的方向发展。而当前的机器人基本上都是根据特定任务来开发的，具有固定的结构和功能，并且使用环境也被限制在一定的范围内。这种方式使得机器人的设计和开发周期较长，研制和维护成本较高。

(2) 随着“人口红利”的到期，劳动力短缺和人力成本急剧上升，使得劳动力密集和低人力成本的制造模式在中国难以持续。在这种背景下，选择以机器人为主导的智能制造模式是实现中国制造业转型的重要路径。基于上述原因，我国对各类机器人的需求量急剧增加，近几年中国机器人市场的增长速度一直处于世界第一。然而，我国虽然在传统机器人技术研发及产业化方面取得了初步进展，但在信息技术与机器人技术结合方面仍然十分滞后，这使得我国机器人技术及产业发展与发达国家的差距进一步扩大。

为了解决上述制约我国机器人产业发展的技术难题，多家高校和科研单位根据我国机器人产业发展现状，在从技术体系、市场应用、产业发展等方面进行全面综合分析研究的基础上，共同展开了“机器人模块化、网络化、标准化开发平台关键技术与示范”项目的研究工作。该项目是“十二五”期间先进制造技术领域一项重要的“863”课题，其主要目标是面向服务机器人与工业机器人重点行业，

开展机器人模块化、网络化、标准化开发平台关键技术及示范研究,突破制约我国机器人产业化发展模块化/标准化设计技术,建立机器人模块化、网络化开发公共技术支持平台,从产品设计角度实现软硬件结构分离,系统软件与应用软件分离,实现机器人软件设计的模块化、网络化、分布式,为企业提供机器人软硬件产品设计服务,提高机器人系统的设计和开发效率。为了实现这一目标,需要从机器人体系结构角度实现机器人软硬件根本性分离,实现软硬件分离的核心是建立机器人操作系统,降低机器人相关软件的开发难度。ROS (Robot Operating System)是斯坦福大学人工智能实验室和机器人技术公司 Willow Garage 的个人机器人项目部共同开发的一套基于 Linux 的标准机器人开发环境,它具有分布式设计、支持多种编程语言、集成了丰富的机器人算法库、系统整体开源、适合大规模软件的开发等优点,已成为国际主流的机器人软件平台。然而,由于其采用分布式设计,加之高度依赖 Linux 操作系统,因而也存在着可移植性差、缺乏实时性等缺点,这些缺点严重制约了 ROS 在某些领域的机器人系统中的应用。为了解决机器人模块化的专用操作系统问题,“863”课题组研制了一种基于 ROS 的混合实时操作系统 RGMP-ROS (Real-time and General Operation System for Multi-core Processor-Robot Operating System),它具有高度的实时性、可扩展性、可配置性、可移植性和可复用性。为了充分验证 RGMP-ROS 的可靠性与安全性,考核其质量与性能,有必要针对 RGMP-ROS 的特点,研究相应的测试方案和测试技术。

1.1.2 研究意义

随着机器人技术的不断发展和逐步成熟,投入使用的机器人种类不断增加,陆续出现了工业机器人、康复机器人、陪护机器人、娱乐机器人、医疗机器人、救援机器人等。这些机器人与人共处的机会也越来越多,一旦它们发生故障,不但会带来经济损失,还可能引发重大人身安全事故,因而对机器人的安全性和可靠性要求越来越高。操作系统是机器人的重要组成部分,操作系统的可靠性是影响机器人安全性和可靠性的核心因素之一。作为一种通用的机器人操作系统, RGMP-ROS 的研制目的是解决机器人模块化问题,从而提高机器人系统的设计和开发效率,缩短机器人产品的设计与开发周期,降低开发成本,促进我国机器人产业的发展。RGMP-ROS 能否得到有效推广和广泛应用是评判其成功与否的重要标准。由于人们对机器人安全性和可靠性的关注度越来越高,因而,安全性和可靠性是决定其能否被广泛应用的主要因素。对该操作系统进行充分而有效的测试,

是保证其可靠性的重要手段。因而,研究适合 RGMP-ROS 操作系统使用的测试技术并找到有效的测试方案,具有重要的理论价值和现实意义。

1.2 研究内容

1.2.1 RGMP-ROS 的特点

1. 对可靠性要求较高

RGMP-ROS 是一个机器人软件平台,是机器人系统资源的控制与管理者,也是硬件抽象描述、底层驱动程序管理、共用功能的执行、程序间消息传递、程序发行包管理等基础服务的提供者,是整个机器人系统的重要组成部分。机器人操作系统缺陷将给使用者的生命和财产安全带来较大隐患。因而, RGMP-ROS 要具有非常高的可靠性和安全性。

2. 升级修改频繁

RGMP-ROS 机器人操作系统要具备高实时性、较强的多任务处理能力、适应多核硬件平台的发展趋势、功能渐趋复杂等特性,因而机器人操作系统的代码量相对较大,其开发过程常常要经过多次迭代才能完成。同时,由于机器人操作系统比较庞大,在其生命周期中,需要不断地对发现的问题进行修改。另外,与计算机操作系统相比,机器人操作系统的个性化较强,软件系统和硬件的结合非常紧密,一般要针对硬件进行系统的更新和移植,即使在同一品牌、同一系列的产品中,也需要根据系统硬件的变化和增减进行修改。

3. 可配置项较多

为了满足易用性、通用性和可移植性的要求,实现高度可配置的目标, RGMP-ROS 中存在大量的可配置项目。

4. 具有良好的实时性能

RGMP-ROS 的研制主要是为了解决 ROS 操作系统的实时性不足的问题,由于机器人操作系统需要进行运动插补、状态反馈等实时控制,同时还要对关节过热、超极限位置、总线异常、电机异常等异常情况进行及时处理,这就要求 RGMP-ROS 具有较高的实时性。

1.2.2 RGMP-ROS 的测试难点

根据 RGMP-ROS 机器人操作系统的特点,对其进行测试时主要面临 4 个难点。

1. 资源问题的检测

资源是由操作系统内核管理的程序元素，如内存、内核锁、套接字等。资源的使用要遵循以下准则：不能使用未被实例化的资源；不再使用的资源应及时释放并且只能释放一次。不满足上述准则时，就会出现资源问题。资源问题具有以下特点。

- ◆ 隐蔽性强：编译器不能发现，而且没有明显的症状；
- ◆ 危害性强：可能导致系统崩溃，甚至发生重大事故，严重影响系统的用户体验；
- ◆ 随机性强：系统的某次运行中，可能发生，也可能不发生。

资源问题对软件系统的影响巨大，对操作系统来说更是如此。一个包含较多资源问题的操作系统给使用者带来的潜在危害可能是毁灭性的，因而有必要对机器人操作系统中的资源问题进行充分而有效的检测。然而，由于操作系统运行于整个系统的底层，较为精确的动态分析手段难以使用；同时，机器人操作系统代码量较大，测试周期又相对较短，因而，测试方案既要具有较低漏报率以便能够尽可能多地发掘资源问题，又要具有较低的误报率以便能够提高测试效率。

2. 回归测试中的测试用例选择

在软件开发过程中，每一轮的迭代完成或者对软件进行升级或适应性改造后，都需要对其进行回归测试，以便考核新修改部分的正确性，并检查新修改部分是否影响了软件整体的正确性。随着测试的进行，测试用例库中的测试用例会不断增加。同时，由于某次修改往往仅对软件部分模块做了改动，当前基线测试用例库中的测试用例对本次改动模块的敏感度各不相同，有些测试用例对改动模块及其相关模块均有覆盖，而有些用例则根本不覆盖任何与本次修改相关的模块。因而，无论从测试时间、测试成本考虑，还是从测试需求考虑，选择测试用例库中的所有测试用例作为回归测试用例的做法均是不可取的。有必要找出一种高效的回归测试用例选择方法，以便从测试用例库中找出具有针对性的测试用例进行回归测试，提高回归测试效率。对复杂性相对较高的机器人操作系统软件来说更是如此，由于机器人操作系统开发过程必然要进行多次迭代，在其整个生命周期中，也必定要经过多次升级，因而回归测试的频率比较高，更需要寻找一种测试用例的优化选择方法，以便选用少量的测试用例准确地覆盖新版本中尽可能多的改动，降低回归测试的时间及成本。

3. 组合测试用例集的生成

组合测试（Combinatorial Test）是一种测试用例生成方法，它将被测试应用抽象为一个受到多个因素影响的系统，其中每个因素的取值是离散且有限的，通

过一组测试用例集来测试各个因素之间的相互作用给系统造成的影响。两因素 (Pairwise) 组合测试生成一组测试用例集, 可以覆盖任意两个因素的所有取值组合, 在理论上可以暴露所有由两个因素共同作用而引发的缺陷。多因素 (N -way, $N > 2$) 组合测试可以生成测试用例集, 以覆盖任意 N 个因素的所有取值组合, 在理论上可以发现由 N 个因素共同作用引发的缺陷。组合测试是提高软件测试效率的有效方法, 是保证软件质量的重要手段, 非常适用于机器人操作系统这类代码量大、功能繁杂、配置选项较多的软件系统的测试。组合测试用例集的生成问题是组合测试的关键, 由于随着组合维度的提高, 测试用例数呈爆炸式增长, 测试效率会急剧降低。因而考核一个组合测试用例集好坏的标准就是其所含测试用例数量的多少, 对于相同维度的组合测试来说, 测试用例数量越少越好。为了保证缺陷检测能力, 并尽量减少测试用例的规模, 以提高测试效率, 有必要研究适用于机器人操作系统的组合测试技术, 其难点和关键在于组合测试用例集的生成。

4. 实时性能测试方案的制定

与功能测试相比, 适用于操作系统实时性能测试的工具相对缺乏, 操作系统实时性能测试理论也不成熟, 尚不存在统一的评测标准。为此, 有必要研究具有较强通用性和较好可移植性的机器人操作系统实时性能测试方案。

1.2.3 研究内容

以 RGMP-ROS 操作系统的测试需求为牵引, 以机器人操作系统内核模块的测试技术的研究为重点, 以研制具有较高实用性和先进性的操作系统内核模块测试工具为目标, 紧紧围绕上述难点的解决开展相关研究工作。

1.3 相关领域的研究现状

1.3.1 操作系统资源问题检测研究现状

资源问题的严重性引发了不少学者的关注, 但大多数研究人员仅仅关注了内存资源的泄露问题, 没有考虑内存访问冲突问题, 也没有考虑其他资源的泄露或访问冲突问题。比较典型的内存泄露问题检测有以下几种方案。

1. 基于内存管理模型的检测

David 在其博士论文中提出了一种通过静态分析来半自动识别大型 C 语句编

制程序中的内存泄露问题^[1]; Xu 等提出了一种适用于精确路径敏感分析的静态内存管理模型, 用于解决内存泄露问题的检测^[2]。上述方法均采用内存管理模型来完成软件系统内存泄露问题的检测。这种方法在大型系统错误测试方面比较有效, 但也存在以下问题: 构建模型的难度较大; 检查是基于代码抽象出的模型而不是代码本身, 代码变化以后, 需要重新建立检查模型; 其模型本身的可信性有待商榷。

2. 基于动态分析技术的检测

Cherem 等人提出了一种通过跟踪分配点到释放点之间的流量来检测内存泄露的过程分析算法^[3]; Clause 等人提出了一种动态检测内存非法访问的方法, 用于检测与内存访问相关的软件故障^[4]; Clause 等人提出了一种通过跟踪动态分配内存区域的指针, 来检测并定位内存泄露点^[5]; HuuHai 等人提出了一种通过监视动态内存分配来检测和消除程序中的内存泄露问题的方案^[6]。上述方法均采用动态检测技术来完成内存问题的检测, 用于检测内存泄露的动态检测技术相对成熟, 精确度较高; 然而由于这种技术需要使用动态二进制插桩 (Dynamic Binary Instrumentation, DBI) 手段完成问题检测, 由于 DBI 自身的局限性, 很难应用于操作系统的检测; 而且, 对检测出来的内存泄露问题进行代码定位也是一项很有难度的工作。

当前, 研究内存泄露以外的资源问题的人员极少, 文献检索结果显示, 仅 Saha 研究了其他资源问题。他通过对程序中错误处理代码及资源使用上下文环境的关注, 提出了一种用于 Linux 操作系统的资源释放问题检测方法^[7]。该方法能够检测到内存泄露以外的其他资源泄露问题, 但由于该方法的关注重点是异常处理代码, 因此会丢失部分潜在资源泄露问题, 而且也不能检测出资源访问冲突问题。

1.3.2 回归测试用例选择研究现状

通过对比软件系统新旧版本的不同之处, 依据测试用例对代码的覆盖信息, 选出对变更部分代码覆盖较好的测试用例来完成回归测试, 从而提高回归测试的效率, 这种方法称为回归测试用例选择。回归测试用例选择是一种重要的回归测试优化技术。为了解决回归测试用例的选择问题, 人们提出了多种解决方案。包括整数规划法^[8~12]、数据流分析法^[13~16]、符号执行法^[17]、动态切片法^[18]、图遍历法 (Graph Walk based Approach)^[19~24]、代码分析法^[25~26]、SDG 切片法^[27]、路径分析法^[28]、修改检测法^[29]、防火墙法^[30~34]、CFG 集群识别法^[35]以及其他方法^[36~54]。

1. 整数规划法

为了解决 FORTRAN 程序的回归测试用例选择问题, Fischer 首次提出将整数规划技术应用于回归测试用例选择问题。假设一个程序有 m 个程序块, n 个测试用例, 则回归测试用例选择问题的数学模型为:

$$\begin{aligned} \min \quad & Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq b_m \end{aligned}$$

决策变量向量 $\langle x_1, x_2, \dots, x_n \rangle$ 用于描述测试用例的选择情况, 如果第 i 个测试用例被选中, 则 x_i 取 1, 否则 x_i 取 0。系数 $c_1, c_2, \dots, c_n$ 代表每个测试用例的执行成本。依赖矩阵 $a_{11}, \dots, a_{mn}$ 用于描述程序块与测试用例间的覆盖信息。如果第 i 个程序段被第 j 个测试用例覆盖, 则 a_{ij} 取 1, 否则 a_{ij} 取 0。 b_i 用于标识多少个程序模块受本次修改的影响, 其值可由用于描述各程序段间可达性的可达矩阵求得。

Fischer 方案的缺点是无法应对新版本程序的控制流的变化。依赖矩阵 $a_{11}, \dots, a_{mn}$ 由程序的控制流得到, 当控制流发生改变时需要重新执行全部的测试用例来得到依赖矩阵, 而这有悖于回归测试用例选择的初衷。

2. 数据流分析法

基于数据流分析, 研究人员提出了许多测试用例选择技术。Harold 和 Soffa 将数据流分析作为软件维护阶段单元测试的增量法检测标准。Taha 等人提出了一种基于增量数据流分析算法的测试用例选择框架。Gupta 等人使用程序切片技术来识别受代码修改影响的程序段。

数据流分析方法的缺点是无法检测不引起数据流变化的程序修改。

3. 符号执行法

符号执行法是通过符号表达式来执行程序路径的一种静态分析技术。其中, 程序的执行用符号来模拟。例如, 使用变量名而不是实际值, 程序的输出被表示成包含这些符号的逻辑或数学表达式。Yau 等人提出了一种基于符号执行法的测试用例选择方案。该方案首先通过对代码的静态分析来确定输入部分; 接下来生成一些测试用例, 并确保每个输入至少被执行一次; 在给定程序修改信息的基础上, 方案会在控制流图中标识出触发修改代码的边; 最后选出能够覆盖新增或修