



航天科技图书出版基金资助出版

航天型号 高可靠软件系统 调试原理与技术

蔡 铭 程 胜 王 瑞 著



中国宇航出版社

本书针对复杂航天型号对软件系统的高可靠性要求，在深入分析软件故障特征的基础上，阐述了当前软件调试手段与工具的技术分类体系，介绍了一批最新的、具有代表性的软件调试技术，包括程序规则分析、用户行为分析、虚拟化调试支持、故障重现与逆向调试、统计调试、不变式调试等内容，以提高软件故障诊断与分析的自动化、智能化程度，提高软件调试效率，缩短软件交付周期，提高航天型号软件系统的可靠性。

本书主要读者对象是航天型号软件系统的设计人员、开发人员、测试人员及管理人员，也可作为其他科技人员了解和掌握高可靠软件系统质量保证与调试技术的参考书。

ISBN 978-7-80218-424-4



9 787802 184244 >

定价：65.00元

航天科技图书出版基金资助出版

航天型号高可靠软件系统 调试原理与技术

蔡 铭 程 胜 王 瑞 著



中国宇航出版社

· 北 京 ·

版权所有 侵权必究

图书在版编目(CIP)数据

航天型号高可靠软件系统调试原理与技术/蔡铭,程胜,王瑞著.
—北京:中国宇航出版社,2008.8
ISBN 978-7-80218-424-4

I.航... II.①蔡...②程...③王... III.航天器—软件可靠性—调试 IV.V47-39

中国版本图书馆 CIP 数据核字(2008)第 119593 号

责任编辑 张艳艳 封面设计 03 工舍 责任校对 祝延萍

出版 中国宇航出版社

社址 北京市阜成路 8 号 邮编 100830
(010)68768548

网址 www.caphbook.com / www.caphbook.com.cn

经销 新华书店

发行部 (010)68371900 (010)88530478(传真)
(010)68768541 (010)68767294(传真)

零售店 读者服务部 北京宇航文苑
(010)68371105 (010)62529336

承印 北京画中画印刷有限公司

版次 2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷

规格 880×1230 开本 1/32

印张 12.25 字数 339 千字

书号 ISBN 978-7-80218-424-4

定价 65.00 元

本书如有印装质量问题,可与发行部联系调换

序

随着航天型号功能的日趋复杂,软件在型号中的应用越来越多,其规模和复杂度也日趋上升。从近年来对型号问题的统计数据来看,由于软件所导致的问题已占据一定比例。软件已经成为航天型号产品质量和可靠性的关键因素之一。

调试和故障诊断是软件研制过程中的重要环节,其主要的工作是针对故障现象完成故障的分析和定位,是保障和提高软件质量和可靠性的重要手段。目前,在型号软件研制过程中,软件调试工作绝大部分都依靠人工完成,尚未形成理论化、系统化、精确化、自动化的调试手段。

本书作者结合他们在开发高可靠复杂软件方面的实际经验和研究成果,在深入分析软件故障特征的基础上,阐述当前软件调试手段与工具的技术分类体系,并系统地介绍了当前国内外最新的、具有代表性的相关研究成果和技术。通过这些手段的应用,可以提高软件故障诊断与分析的自动化和智能化,快速、准确地定位故障,极大地提高软件调试效率。希望本书的出版能对推动航天型号软件可靠性工作起到积极作用。

总装备部载人航天工程软件专家组组长
中国航天科技集团公司软件专家组组长



前 言

软件系统广泛应用于当前各类航天型号系统中,为了能够适应各种复杂的空间环境和完成繁杂的空间任务,软件系统的应用规模、复杂度、重要性程度以及所承担的功能比重,近年来均呈急剧上升趋势。

软件是设计师塑造的、功能强大的工具,但寄生着各种 bug 的软件,顷刻间又可能转变为破坏力强大的敌人。随着计算机体系结构、程序设计语言的不断丰富,软件规模不断膨胀,任务复杂度不断提高,系统并发性不断加强,型号软件系统中潜在的 bug 已越来越难以捕捉、定位与控制。

不少事实证明,软件故障是导致型号系统失败的重要因素,软件系统的质量与可靠性已经成为型号产品质量与可靠性的关键因素之一,甚至可能直接影响发射任务的成败。

但令人遗憾的是,软件的质量和可靠性仍远不如人意。与硬件系统的可靠性相比,软件的可靠性一般要低一个数量级,在避错技术、防错技术、容错技术,以及故障检测和诊断等方面都明显落后。

软件调试技术是型号软件质量与可靠性保障体系中的重要环节之一,它是一个根据故障现象定位错误本质,并修正软件缺陷的循环迭代过程。与软件工程中的其他环节相比,软件调试技术无论在理论与方法、支撑技术与辅助工具、系统化支持程度,以及调试自动化、智能化方面仍然处于较原始的状态,软件调试过程低效、单调、乏味。

软件调试技术,尤其对于可靠性要求高的复杂型号软件而言,蕴含着非常丰富的实践经验和技巧,但是在当前计算机学科教学体系中(如 IEEE—CC2005),主要注重于软件的设计、开发方面的教学与训练,对于软件可靠性保障的后端技术,如软件测试、验证等则涉及极

少。其中,软件调试技术的教学部分完全为空白,软件调试的相关技术、方法和经验,需要程序员在实践工作中逐步摸索与积累,因而,难以在短期内被完全掌握。

随着型号软件系统越来越复杂化,对软件错误的跟踪、定位、分析与诊断困难重重,急需软件调试支撑技术、辅助工具的同步提升。当前软件调试技术在不断推陈出新,因此及时了解、熟悉与掌握最新的软件调试技术、方法与工具非常必要。

本书将软件调试技术作为一门学科进行研究和分析,以 bug 为中心,围绕软件调试技术这一主题进行组织,提供了大量的代码实例,对 bug 分类及其分布规律,以及各类调试技术原理、实现技术和支持工具等进行细致而深入的阐述,具有较强的可操作性与较广的技术覆盖面。

本书所涉及的工作得到了“十五”总装预研项目、国家“十五”863 重大软件专项、“十一五”总装预研项目、航天科技创新基金、航天支撑技术基金等多个项目的资助,课题组针对当前最新的各种软件调试方法、智能化故障定位技术等进行了全面的跟踪与探索,并与型号单位紧密配合进行了实践,面向高可靠型号软件开发应用的需求,研制了一系列软件调试、测试及验证等支持工具。本书是对上述成果的全面整理与总结。

全书共分为 13 章,从 3 个方面进行介绍:第一部分介绍软件调试技术概述、国内外型号软件中出现的典型 bug,bug 分类及分布规律分析;第二部分针对 bug 类型,详细介绍软件调试中涉及的内存类 bug 调试、静态分析、动态分片、Delta 调试、统计调试、不变式调试、难以重现类调试、体系结构扩展调试,以及数据挖掘调试技术;第三部分介绍对软件调试技术的评价与调试基准测试程序。

本书主要面向的读者群是型号软件系统的设计人员、开发人员、测试人员,以及管理人员。对于非型号软件系统的开发人员而言,本书也不失为一本有助于拓宽视野、了解和掌握高可靠软件质量保证与调试新技术的可选之书。

总装备部载人航天工程软件专家组组长、中国航天科技集团软件专家组组长汪玲研究员对本书提出了许多宝贵而重要的意见,并在百忙之中为本书作序,在此谨表示诚挚的谢意。

本书由蔡铭、程胜同志负责全书的总体策划、文字修改和最后统稿。参加本书编写工作的还有王瑞、富浩、杨子江、叶珂珂和马超等同志。因时间紧迫、水平有限,错误之处在所难免,敬请读者批评指正。

在本书的出版过程中,感谢航天科技图书出版基金的大力支持。

编 者

2008年6月

目 录

第 1 章 软件调试技术概述	1
1.1 软件系统的“双刃剑效应”	1
1.2 软件质量体系中的短板——调试技术	2
1.3 传统软件调试技术的局限性	4
1.4 软件调试技术的发展概况	8
1.5 本书的组织	11
第 2 章 型号软件中的 bug 分析	13
2.1 概述	13
2.2 国外型号软件中的 bug	13
2.2.1 金星探测器水手 1 号	13
2.2.2 阿里安 5	14
2.2.3 火星气候轨道器 MCO	17
2.2.4 火星极地着陆器	19
2.2.5 Titan/Centaur/Milstar 军事卫星	23
2.3 国内型号软件中的 bug	26
2.3.1 优先级运算问题	26
2.3.2 程序结构不合理问题	27
2.3.3 初始化不完备问题	29
2.3.4 原子性破坏问题	31
第 3 章 软件 bug 分类及分布规律	34
3.1 软件 bug 概述	34
3.1.1 关于 bug 的起源	34

3.1.2 软件 bug 的定义	35
3.2 典型软件 bug 分类体系简介	36
3.2.1 Boris Beizer 分类体系	36
3.2.2 IEEE 1044—1994 分类体系	37
3.2.3 QJ 3026—1998 分类体系	38
3.3 C 语言软件 bug 分类体系	39
3.3.1 内存相关错误	42
3.3.2 初始化错误	46
3.3.3 计算错误	49
3.3.4 输入输出错误	54
3.3.5 控制流错误	55
3.3.6 数据处理解释错误	65
3.3.7 竞争类错误	67
3.3.8 平台相关错误	72
3.3.9 其他错误	76
3.4 当前软件 bug 分布规律分析	78
3.5 软件 bug 分布发展趋势	82
3.6 对软件调试技术的需求	83
第 4 章 内存类 bug 调试	84
4.1 内存类 bug 产生原因	84
4.1.1 内存类 bug 现状	84
4.1.2 动态内存管理	85
4.2 内存类错误调试支持工具	87
4.2.1 Insure++	87
4.2.2 Purify	93
4.2.3 Valgrind	107
第 5 章 静态分析调试	113

5.1 静态分析概述	113
5.2 典型静态分析技术	113
5.2.1 基于规则的检查	113
5.2.2 符号执行	114
5.2.3 定理证明	115
5.2.4 类型推导	115
5.2.5 抽象解释	116
5.2.6 模型检测	116
5.3 静态分析工具	118
5.3.1 Testbed 简介	118
5.3.2 其他静态分析工具简介	125
5.4 静态分析局限性	127
第 6 章 动态分片调试	135
6.1 什么是程序分片	135
6.1.1 程序分片的发展历史	137
6.1.2 程序分片的分类	138
6.1.3 程序分片的应用	138
6.2 静态分片	139
6.2.1 静态分片	139
6.2.2 Weiser 的算法	140
6.2.3 Ottenstein 的算法	141
6.2.4 基于系统依赖图的算法	143
6.2.5 静态分片和动态分片	144
6.3 动态分片	144
6.3.1 分片标准	145
6.3.2 def-use 动态分片算法	146
6.3.3 Agrawal 和 Horgan 的算法	148
6.4 分片调试实例	154

6.4.1 采用可信度剪枝的动态程序分片	155
6.4.2 Delta 调试和动态分片相结合的软件调试方法	156
6.5 商品化的分片工具	163
第 7 章 Delta 调试	168
7.1 Delta 调试概述	168
7.2 Delta 调试分类	169
7.2.1 简化	170
7.2.2 分离	171
7.3 Delta 调试基本原理	172
7.3.1 简化算法	172
7.3.2 层次化 Delta 调试	175
7.3.3 分离故障起因	180
7.3.4 分离因果链	185
7.4 Delta 调试工具举例	192
7.4.1 ASKIGOR	192
7.4.2 DDchange 和 DDstate	194
7.5 问题和局限性	196
第 8 章 统计调试	198
8.1 统计调试概述	198
8.1.1 统计调试的定义	198
8.1.2 统计调试的特点	199
8.1.3 统计调试的发展历史	200
8.2 统计原理	201
8.2.1 常用分布	201
8.2.2 常用定理及统计推断	202
8.3 统计调试分类	204
8.3.1 在线和离线统计调试	204

8.3.2 单一 bug 和 多个 bug 定位	205
8.4 统计调试基本方法	206
8.4.1 矢量表示	206
8.4.2 特征选择	208
8.4.3 聚类	208
8.4.4 谓词排序	210
8.5 统计调试模型	210
8.5.1 T-Proximity 模型	210
8.5.2 Liblit05 模型	214
8.5.3 SOBER 模型	217
8.5.4 R-Proximity 模型	219
8.5.5 Vote 模型	223
8.5.6 ARGUS 模型	226
8.5.7 各模型比较	231
第 9 章 不变式调试	232
9.1 不变式相关概念	232
9.2 不变式的分类	234
9.2.1 循环不变式	234
9.2.2 函数的前置不变式和后置不变式	236
9.2.3 类不变式	237
9.3 不变式的应用	239
9.3.1 程序设计	239
9.3.2 辅助程序理解	240
9.3.3 程序版本验证	242
9.3.4 故障定位	243
9.3.5 软件重构	248
9.4 发现不变式的方法	250
9.4.1 不变式的静态发现技术	250

9.4.2	不变式的动态发现技术	251
9.4.3	静态和动态方法的对比	253
9.5	不变式发现工具	255
9.5.1	Daikon	255
9.5.2	DIDUCE	260
9.5.3	Agitator	261
第 10 章	难以重现类 bug 调试	262
10.1	重现 bug 的必要性	262
10.2	bug 难以重现的原因	263
10.2.1	优先级逆转	264
10.2.2	中断导致 bug 难以重现	265
10.2.3	调试器对程序时序的影响	266
10.2.4	调试器对程序内容的影响	267
10.3	重放调试概述	268
10.3.1	重放调试的基本思想	268
10.3.2	重放调试的分类	269
10.4	基于虚拟机的全系统执行重放	271
10.4.1	bbreplayer 系统框架	271
10.4.2	bbreplayer 系统功能	272
10.4.3	bbreplayer 实现	273
10.5	分布式系统的重放调试	279
10.5.1	Liblog 的设计要求	279
10.5.2	Liblog 的设计	280
10.5.3	挑战与解决方案	281
10.6	VMware 新增功能介绍	283
10.6.1	简介	283
10.6.2	典型的应用场景	285
10.6.3	主要功能	286

10.6.4 主要应用	286
10.6.5 使用 VAssert	286
第 11 章 体系结构扩展调试	288
11.1 通用调试技术的局限性	288
11.1.1 现有体系结构对软件调试的支持	290
11.1.2 系统结构扩展支持调试的趋势	297
11.2 体系结构扩展技术分类	297
11.2.1 冯·诺依曼体系结构	297
11.2.2 体系结构扩展对应方法	299
11.2.3 体系结构扩展方法分类	299
11.3 软件扩展调试系统	305
11.3.1 iWatcher	305
11.3.2 FullDebugger	311
第 12 章 基于数据挖掘的调试方法	320
12.1 数据挖掘支持调试	320
12.1.1 数据挖掘调试概述	320
12.1.2 基于数据挖掘的调试方法分类	324
12.1.3 相关数据挖掘技术	325
12.2 基于源代码的关联挖掘调试技术	337
12.3 基于程序行为分类的挖掘调试技术	346
12.3.1 基于程序计数器(PC)的不变式	347
12.3.2 AccMon	348
12.4 基于程序运行轨迹的挖掘调试技术	350
第 13 章 软件调试技术评价	354
13.1 软件调试技术评价体系概况	354
13.1.1 调试技术评价体系定义	354

13.1.2	调试技术评价体系的历史	354
13.1.3	调试技术评价体系的要求	355
13.2	调试技术评价基本方法	356
13.2.1	寻找能力	357
13.2.2	定位能力	357
13.2.3	性能	359
13.2.4	对潜在错误的影响	359
13.3	常见的调试技术评价基准程序	360
13.3.1	西门子套件	360
13.3.2	SPEC 2000 性能套件	362
13.3.3	PEST 套件	365
13.3.4	BugBench 套件	365
13.3.5	IBM Haifa 套件	367
13.3.6	Nofib-buggy 套件	369
13.3.7	iBUGS 工具及其数据集	371
参考文献		373

第 1 章 软件调试技术概述

1.1 软件系统的“双刃剑效应”

软件系统作为计算机系统的神经中枢,已经延伸到现代武器型号设备、装置中的各个角落,为了能够适应各种复杂的空间环境和完成繁杂的空间任务,软件系统的应用规模、复杂度以及重要性程度,近年来均呈急剧上升趋势。例如:

- 一个国际太空站需要上百万行的软件系统,控制各种导航、通信及实验设备;

- 美国航空航天局(NASA)的太空飞船项目中,其船载软件代码量大于 50 万行,地面控制和处理软件代码量约 350 万行;

- 我国神舟五号载人飞船中,船载软件模块共 60 余个,软件指令达 70 万条,地面支持系统的软件规模则大于 140 万条指令;在神舟六号飞船的 7 大系统、13 个分系统中,软件模块规模进一步扩大到 82 个。

在型号设备的功能分布中,由软件系统承担的功能比重不断加大。例如,在美国第二代歼击机 F-111 中,由软件部分所实现的功能约占 20%,到了第四代机 F-22,这个比例已上升为 80%。与此相类似,在我国新研的军用飞机中,其飞控系统、火控系统及弹射救生系统等,均采用软件系统逐步替代原有的机械、光学设备,实现其控制,某些机种的机载代码量已超过了百万行量级。

由于软件错误直接造成系统失效的比例持续递增,据 1986 年的统计数据表明,系统失效事件中诱因是软件错误的比例约占 25%,而到 2000 年,该比率已超过 40%。软件错误所导致的经济损失也触目惊心,根据美国国家标准技术研究所 2002 年 6 月公布的调查表明,由