

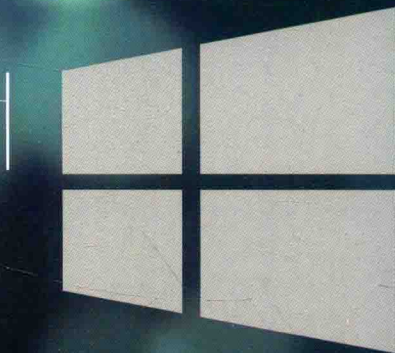
微软技术开发者丛书

微软全球执行副总裁沈向洋作序

微软公司全球资深副总裁潘正磊作序

.NET Core 2.0 **应用程序高级调试**

完全掌握 Linux、macOS 和 Windows
跨平台调试技术



Be an Expert in .NET Core Debugging on Linux,
macOS and Windows

李争 编著
Li Zheng

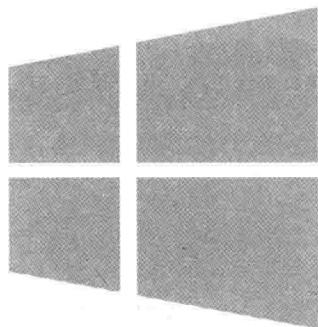
清华大学出版社



微软技术开发者丛书

.NET Core 2.0 应用程序高级调试

完全掌握 Linux、macOS 和 Windows
跨平台调试技术



Be an Expert in .NET Core Debugging on Linux,
macOS and Windows

李争 编著
Li Zheng

清华大学出版社
北京

内 容 简 介

随着.NET Core 开源和跨平台的特性逐渐被广大开发者熟知和接受,有越来越多的.NET 应用从 Windows 平台向 Linux 平台进行迁移,有越来越多的开发者在 Linux 或者 macOS 操作系统上开发.NET 应用。同时,这也给之前只熟悉在 Windows 平台上开发.NET 应用的开发者带来了一系列挑战。怎样在 Linux 和 macOS 操作系统上有效地使用工具对.NET Core 应用程序进行调试,找出程序中隐藏的代码错误和内存中的问题成为保障应用程序在 Linux 和 macOS 上平稳运行的重要课题。本书从.NET Core 概念、.NET Core 相关工具、调试器选择、调试命令介绍和多线程、内存调试实践等多个环节对.NET Core 在 Linux、macOS 和 Windows 三个操作系统上如何进行调试做了详尽的介绍。内容包括.NET Core 基础知识、.NET Core 的编译、.NET Core 命令行工具、调试环境的配置、调试器的基本命令、.NET 基本调试命令、多线程、async 和 await、内存和垃圾收集等,分9章全面地阐述了.NET Core 跨平台调试技术。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

.NET Core 2.0 应用程序高级调试:完全掌握 Linux、macOS 和 Windows 跨平台调试技术/李争编著. —北京:清华大学出版社,2018(2018.11重印)

(微软技术开发者丛书)

ISBN 978-7-302-50533-4

I. ①N… II. ①李… III. ①网页制作工具—程序设计 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2018)第 139397 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:李建庄

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:186mm×240mm 印 张:11.5

字 数:260 千字

版 次:2018 年 9 月第 1 版

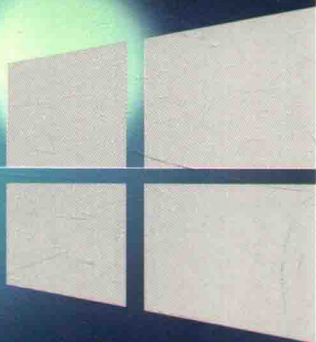
印 次:2018 年11月第2次印刷

定 价:59.00 元

产品编号:079444-01

李争

微软(中国)有限公司全渠道事业部资深技术顾问。专注于微软公有云平台 Azure 的解决方案和架构设计、Azure 应用的实施,以及 Azure 上的开源技术等工作。曾在微软企业服务部作为原厂技术支持工程师工作多年,承担企业开发者代码调试和技术支持、Web 应用前端后端代码调优,以及 IIS 的问题诊断、调优、培训等工作。具有丰富的企业客户临场解决严重系统问题的经验。拥有四十多门微软认证证书,涵盖了几乎全部微软开发相关技术,同时也是一位具有十多年授课经验的微软认证讲师(MCT)。著有《微软互联网信息服务(IIS)最佳实践》《微软开源跨平台移动开发实践》畅销图书。





丛书序

FOREWORD

四十不惑创新不止

从飞鸽传书到指尖沟通,从钻木取火到核能发电,从日行千里到探索太空……曾经遥不可及的梦想如今已经变为现实,有些甚至超出了人们的想象,而所有这一切都离不开科技创新的力量。

对于微软而言,创新是我们的灵魂,是我们矢志不渝的信仰。不断变革的操作系统,日益完善的办公软件,预见未来的领先科技……40年来,在创新精神的指引下,我们取得了辉煌的成绩,引领了高科技领域的突破性发展。

IT行业不墨守成规,只尊重创新。过往的成就不能代表未来的成功,我们将继续砥砺前行。如果说,以往诸如个人计算机、平板电脑、手机和可穿戴设备的发明大都是可见的,那么,在我看来,未来的创新和突破将会是无形的。“隐形计算”就是微软的下一个大事件。让计算归于“无形”,让技术服务于生活,是微软现在及未来的重要研发方向之一。

当计算来到云端后,便隐于无形,能力却变得更加强大;当机器学习足够先进,人们在尽享科技带来的便利的同时却觉察不到计算过程的存在;当我们只需通过声音、手势就可以与周边环境进行交互,计算机也将从人们的视线中消失。正如著名科幻作家亚瑟·查尔斯·克拉克所说:“真正先进的技术,看上去都与魔法无异。”

技术是通往未来的钥匙,要实现“隐形计算”,人工智能技术在这其中起着关键作用。近几年,得益于大数据、云计算、精准算法、深度学习等技术取得的进展,人工智能研究已经发展到现在的感知,甚至认知阶段。未来,要实现真正的人机互动、个性化的情感沟通,计算机视觉、语音识别、自然语言都将是人工智能领域进一步发展的突破口及热门的研究方向。2015年7月发布的Windows 10是微软在创新路上写下的完美注脚。作为史上第一个真正意义上跨设备的统一平台,Windows 10为用户带来了无缝衔接的使用体验,而智能人工助理Cortana、Windows Hello生物识别技术的加入,让人机交互进入了一个新层次。Windows 10也是历史上最好的Windows,最有中国印记的Windows,不但有针对中国本土的大量优化,还会有海量的中国应用。Windows 10是一个具有里程碑意义的跨时代产品,更是微软崇尚创新的具体体现,这种精神渗透在每一个微软员工的血液之中,激励着我们

“予力全球每一人、每一组织成就不凡”。

四十不惑的微软对前方的创新之路看得更加清晰,走得也更加坚定。希望这套丛书不仅成为新时代之下微软前行的见证,也能够助中国的开发者一臂之力,共同繁荣我们的生态系统,绽放更多精彩的应用,成就属于自己的不凡。

沈向洋

微软全球执行副总裁

推荐序

FOREWORD

.NET“刷新时刻”

.NET 自从 2002 年诞生以来,已经成为世界领先的桌面和 Web 开发框架。目前世界上有超过 10 万个的 Web 网站使用 .NET 作为网站开发框架。从全球范围来看,使用 ASP.NET 技术开发的 Web 网站远远大于用 Java、Python、Ruby、Node.js 和 Go 等语言开发的 Web 网站数量。

在 2014 年 11 月,我们让 .NET 开源并运行在所有如 macOS、Linux 等主流操作系统上。这是我们对 .NET 生态系统进行“刷新”的时刻。我们已经创建了针对企业级应用程序进行云优的 .NET Core 模块。我们在 Github 上为 .NET 创建了一个充满活力的社区。在中国,我们已经看到腾讯和网易等公司将 .NET Core 应用于支付和游戏等关键工作任务之中。

我们听到了成千上万希望学习 .NET 的新开发人员对更好的学习资料的强烈需求。我要感谢李争花时间写了一本好书,帮助中国的开发者利用 .NET Core 构建应用。

我们期待听到您对我们产品的反馈。我们迫不及待地想知道您使用 .NET Core 构建了怎样的应用。

潘正磊 (Julia Liuson)女士

微软公司全球资深副总裁

.NET “hit refresh”

Since its original inception in 2002, .NET has become one of the most popular framework for desktop and web worldwide. It's the platform of choice of the top one hundred thousand websites worldwide, you will find more websites developed with ASP.NET than Java, Python, Ruby, Node.js and Go combined.

In Nov 2014, we made .NET open source and run on all operating systems, eg Mac, Linux etc. This was the moment when we “hit refresh” on .NET ecosystem. We have created .NET core which is modular and cloud optimized for enterprises who are interested

in developing cloud native applications. We have created a vibrant community for .NET on Github. In China, we have seen adoption from companies like Tencent and NetEase targeting critical workload like payment and gaming.

We have heard strong demands about better learning materials from hundreds of thousands of new developers wanting to learn .NET. I want to thank Michael for taking the time and writing a great book that would help our developers in China building on .NET core.

We look forward to hear your feedback on our product. We can't wait to see what you build with .NET core.

Julia Liuson

前言

PREFACE

不知不觉,.NET Core 已经开源三年多了。在这三年多的时间里,我作为一个亲历者,经历了.NET Core 从 1.0 到 2.0 的涅槃。这几年,也是我个人转型为一名微软技术布道师(Evangelist)的重要时期。

作为一名 24 年前第一次接触计算机就使用微软产品的我来说,微软这三年带来的变化对我的影响真是太大了!开源和云计算除了让微软的股价翻了两番以外,也让我走上了学习和了解开源世界的道路。开源为我打开了世界的另一扇门,让我了解到传统企业软件以外的广阔世界。开源真的彻底改变了我的思维。以前,写一个客户端应用,我会直接打开 Visual Studio;现在,我会考虑清楚用哪种技术才能同时支持 Windows、Linux 和 macOS 三个操作系统平台,再去动手开发。你能想象吗?这本书的全部内容就是我在一台苹果笔记本上创作的,书稿的版本管理是通过 Git 和 Visual Studio Online 来实现的。

当然,作为一名 Windows 平台的开发者,向开源世界转型也并不是轻松的。为此,我专门买了一台 Macbook Pro。在工作和业余时间强迫自己去适应它,去熟悉开源世界的那些常用工具,在开发过程中体会 Visual Studio Code 的轻便快捷。在这个过程中也积累了一些经验,我的这本书就是我在开源世界工作经验的一部分总结。

.NET Core 作为 .NET Framework 的一个开源世界的变体,与 .NET Framework 既有千丝万缕的联系,又有很大的区别。一方面, .NET Core 的大部分代码都来自 .NET Framework,另一方面, .NET Core 还要处理好 .NET Framework 不曾面对的跨平台、自包含等新问题的挑战。在使用 .NET Core 开发的过程中,我发现有很多的待解决问题。于是,我决定用我的这本书将它们总结出来分享给广大 .NET 开发者,让他们在使用 .NET Core 开发应用程序时少走一些弯路。

这本书集成了我在微软作为开发方向原厂支持工程师(PFE)时的应用程序调试和调优的经验,同时也融合了我在 Linux 平台上的使用经验。通过本书,我将向大家介绍如何在 Linux 的各个发行版本以及 Windows 上利用调试器对 .NET Core 应用程序进行调试的技术和技巧。因为 .NET Core 要想在生产环境上大规模地使用,必须有强大的应用程序调试技术作为保证,以便快速定位问题和解决问题。

通过长达一年时间的写作和对 .NET Core 问题状态的追踪和分析,现在我真的认为是时候把应用程序迁移到 .NET Core 上了!

本书包含哪些内容

本书系统论述了.NET Core 的相关概念、编译方法、命令行工具使用方法、调试环境搭建、调试器基本使用方法、.NET Core 调试扩展基本使用方法、.NET Core 多线程应用程序调试,以及.NET Core 内存管理垃圾收集等相关知识。本书全面详尽地阐述了.NET Core 源代码获取、编译、调试的全方位技术,你需要知道的.NET Core 技术、.NET Core 的编译、.NET Core 命令行工具的使用、调试环境的配置、调试器的基本命令、.NET 调试基本命令、多线程、async 和 await、内存和垃圾收集等内容。

如何高效地阅读本书

本书从读者角度出发,章节由浅入深,从.NET Core 常见问题讲起,直到最后综合运用各种工具对.NET Core 应用程序高级排错。因此,建议读者从头至尾顺序阅读。如果读者具有丰富的Linux使用经验,也可以忽略其中一些简单的章节。

本书适合哪些读者

本书适合使用.NET Core 技术进行应用程序开发的相关开发人员,也适合于希望深入了解和学习.NET Core 平台的读者。

致谢

首先,我要感谢我的家人和我可爱的女儿。因为我在写书稿时严重占用了和她们在一起休闲的时间。其次,要感谢我敬爱的老板崔宏禹老师,以及认真负责的责任编辑盛东亮,这是我们合作的第三本书了。最后我还要感谢我心里爱着的那个人,你是我创作的原动力。

由于作者水平有限,.NET Core 跨平台相关知识涉及广泛,书中难免存在疏漏和不妥之处,敬请广大读者批评指正。

书中样例代码

为了详尽描述调试的整个过程,突出一些要调试的现象,书中涉及了许多.NET Core 代码工程。这些代码都是使用 Visual Studio Code 针对.NET Core 2.0 进行编写,源代码下载地址:

<https://github.com/micli/netcoredebugging>

以上源代码都可以在.NET Core 2.0 环境的支持下,运行在 Windows、Linux 和 macOS 操作系统上。

书中特殊约定

为了直观,书中与操作系统相关的命令都通过操作系统对应的商标来标识,代表这些命令在对应的操作系统下测试通过。因Linux发行版本众多,不同发行版本的Linux部分命令可能略有差异。因作者精力有限,仅能覆盖CentOS/Red Hat 和 Debian/Ubuntu 等最流行的两组发行版本,望读者见谅。

(1)  代表 Debian 8.0 或者 Ubuntu 操作系统;

(2)  代表 CentOS 7.0 或者 Red Hat 操作系统;

(3)  代表 Windows 8.1、10、Server 2008/2012/2016 系列操作系统；

(4)  代表 macOS Sierra 系列操作系统。

在书中配有大量的 Shell 命令、C# 代码和调试器指令以及为了说明调试输出的图片。书中的命令是指用户与操作系统 Shell 之间交互时输入的命令；代码是指经过编译，可以运行在 .NET Core 2.0 上的 C# 语言源代码；调试是指调试者与 Windbg 或者 LLDB 调试器之间交互时输入的调试指令。请读者在阅读时加以区分。

编著者

2018 年 8 月

赞誉

微软开发工具解决方案技术专家 庄俊乾

很荣幸接到李争的邀请,给《.NET Core 2.0 应用程序高级调试——完全掌握 Linux、macOS 和 Windows 跨平台调试技术》一书写一个短评。在写之前,我想先说说李争。李争是微软公司老兵,从 MVP 到 MCS,然后是 DX,最后是 OCP。十几年,也是弹指一挥间,李争的工作角色变化很大,但他始终没有放下技术,每天用 Visual Studio 还是比 Office 多。在一个喧哗的时代,听从自己的内心,坚守自我,始终不断要求自己,持续进步,这很难得。平时上班,李争永远来得比我早,做实验,理头绪,写书。两年内,李争写了三本技术书籍。大家知道,.NET Core 含着微软公司金钥匙出生,却也必须经历一个新产品的不断实践、不断优化的过程。所以,在 .NET Core 的持续优化过程中,要出版一本 .NET Core 的核心书籍,也必须经过多次内容迭代、持续优化的过程。李争是几易其稿,其口头禅是“我又掉坑了”“我爬出来了”,其中之艰辛,其人之坚韧,可想而知。

.NET Core 2.0 以后,实现了大部分的 .NET Standard 接口,功能也愈加稳定和完善。“纸上得来终觉浅,绝知此事要躬行”,推荐大家参考这本新书,并不断实践、思考、充实自己!

腾讯高级工程师 张善友

.NET Core 已经经历了从 1.0 到 2.0 的涅槃,然而还是有很多人怀疑 .NET Core 的成熟性。李争这本《.NET Core 2.0 应用程序高级调试——完全掌握 Linux、macOS 和 Windows 跨平台调试技术》向各位传递了一个非常重要的信息:.NET Core 已经成熟,是时候把应用程序迁移到 .NET Core 上了,.NET Core 2.0 已经具备强大的应用程序调试技术,保障我们在生产环境上出现问题时快速定位和解决问题。

特来电首席科学家 鞠强

很多问题不会随着时间的推移而自动消失。

以前用 C++ 写就的代码,会出现莫名的崩溃(crash)、停止响应(hang)、内存暴涨,让人欲哭无泪。没有 cdb/Windbg 以及后来的 DebugDiag 等神器,大部分应用还在黑暗中踟蹰,直到无法再用,只能重启。.NET 时代,有了 SOS,对于托管代码的调试、托管堆的分析,增加了极大的便利性。到了新时代,微软公司拥抱了 Linux,.NET Core 横空出世。.NET Core 2.0 已经发布了,整个框架日趋成熟和稳定。在 .NET Core 时代,对于 .NET 调试新手,内存问题、CPU 问题依然存在,怎么去分析,如何去解决?对于调试“老鸟”,需要重新学习哪些基础知识,掌握哪些调试技能?这么多文章,那么多技巧,该怎么把它们变成自己的?

李争的这本书,从 CLR 基本原理开始讲起,通过翔实的文字图表、经典的案例分析,完整地讲述了 .NET Core 的调试过程。系统地阅读与学习本书后,读者能掌握 .NET Core 基本原理及相关的调试思路和技能。当读者了解了底层运行框架的本质,就能在以后面对各种生产环境中的问题时保持清醒,不再迷茫,按照正确的逻辑去寻找关键点,最终找到问题的解决方案。

智捷课堂 CEO/著名畅销书专家 关东升

随着移动互联网的发展,跨平台开发的需求越来越多,微软公司推出的 .NET Core 顺应了这个需求。经过从 .NET Core 1.0 到 .NET Core 2.0 的发展,.NET Core 越来越成熟,用户也越来越多,很多开发人员又重新回到 .NET 平台开发产品。因此开发人员急需介绍 .NET Core 2.0 技术的图书。李争所编著的《.NET Core 2.0 应用程序高级调试——完全掌握 Linux、macOS 和 Windows 跨平台调试技术》一书的出版,恰逢时机。本书从理论到实践,由浅入深系统地介绍了 .NET Core 编译、命令行工具、调试环境、async、await,以及内存和垃圾收集等内容,是 .NET Core 开发的必备图书。

微软云计算事业部资深项目经理/《Windows 用户态程序高效排错》作者 熊力

本书可以让读者“体验快速定位问题的快感,享受精确解决问题的愉悦!”同时,参考本书介绍的工作方法,有助于提高工作效率,让领导满意,让公司盈利,让同事敬佩!可以说,《.NET Core 2.0 应用程序高级调试——完全掌握 Linux、macOS 和 Windows 跨平台调试技术》是一本涵盖 Windows 和 Linux 运维实战技巧的必备宝典!

目 录

CONTENTS

丛书序	I
推荐序	III
前言	V
赞誉	IX
第 1 章 .NET Core 基础知识	1
1.1 .NET Core 到底是什么	1
1.1.1 从软件许可协议说开源	1
1.1.2 构成 .NET Core 的重要组件	2
1.2 .NET Standard 又是什么	5
1.3 .NET Core 的一些重要工具	6
1.4 常见问题解答	9
第 2 章 .NET Core 的编译	10
2.1 .NET Core 源代码在 Linux 操作系统上的编译	10
2.1.1 获取 .NET Core 源代码	10
2.1.2 安装编译源代码必要的工具	11
2.1.3 在 CentOS 上手工编译 LLVM、Clang 和 LLDB	13
2.1.4 在 Linux 上编译 .NET Core 源代码	16
2.2 .NET Core 源代码在 Windows 操作系统上的编译	18
2.2.1 下载和安装 Visual Studio	18
2.2.2 安装其他必备软件	19
2.2.3 在 Windows 系统上执行 .NET Core 编译	20
2.3 .NET Core 源代码在 macOS 操作系统上的编译	20
第 3 章 .NET Core 命令行工具	23
3.1 .NET Core CLI 的安装	23
3.2 创建 .NET Core 项目	23

3.3	.NET Core 项目的迁移	25
3.4	.NET Core 项目的构建	26
3.5	.NET Core 项目的发布	29
3.6	对.NET Core 项目进行管理	30
3.6.1	dotnet sln 命令介绍	30
3.6.2	项目之间的引用管理	31
3.6.3	项目的包管理	32
3.6.4	项目引用 NuGet 包的恢复	32
3.7	.NET Core 应用的执行	33
3.8	将.NET Core 项目发布成 NuGet 包	33
3.8.1	dotnet pack 命令介绍	34
3.8.2	dotnet nuget push 命令介绍	34
3.8.3	dotnet nuget locals 命令介绍	35
3.8.4	dotnet nuget delete 命令介绍	36
3.9	dotnet 相关命令的使用	36
3.9.1	创建解决方案和项目	36
3.9.2	设置项目的引用	38
3.9.3	添加测试工程	40
第 4 章	调试环境的配置	42
4.1	调试环境设置概述	42
4.2	Linux 操作系统调试环境设置	43
4.2.1	在 Linux 上设置 ulimit	44
4.2.2	在 Linux 操作系统上部署调试器	44
4.2.3	在 Linux 操作系统上抓取内存转储文件	46
4.3	在 macOS 操作系统上部署调试器	47
4.4	在 Windows 操作系统上部署调试器	49
4.4.1	Windows 上安装 Windbg	50
4.4.2	在 Windows 上抓取内存转储	51
第 5 章	调试器的基本命令	54
5.1	使用 LLDB 进行调试	54
5.1.1	LLDB 调试器简介	54
5.1.2	命令行参数	55
5.1.3	一段用于演示的代码	57
5.1.4	LLDB 的启动和退出	58

5.1.5	设置断点	59
5.1.6	单步调试指令	60
5.1.7	查看调用堆栈	61
5.1.8	线程切换	63
5.1.9	寄存器调试指令	63
5.1.10	查看内存数据	64
5.2	Windbg 调试器和基本指令	65
5.2.1	Windbg 简介	65
5.2.2	Windbg 的启动和退出	66
5.2.3	Windbg 设置断点	68
5.2.4	Windbg 查看堆栈调用	69
5.2.5	Windbg 线程相关指令	69
5.2.6	Windbg 寄存器相关指令	71
5.2.7	Windbg 查看内存数据	72
第 6 章	.NET 基本调试命令	73
6.1	.NET 调试扩展概览	73
6.2	.NET 数据结构的基本知识	74
6.2.1	对象在内存中的形态	75
6.2.2	MethodTable 和 EEClass	76
6.2.3	MethodDesc	77
6.3	.NET 调试扩展命令	77
6.3.1	代码和堆栈调试命令	77
6.3.2	CLR 数据结构相关调试命令	83
6.3.3	内存对象分析相关命令	89
6.4	那些所谓的调试套路	95
第 7 章	多线程	98
7.1	多线程基础	98
7.1.1	线程的基本概念	98
7.1.2	.NET Core 多线程同步对象	98
7.2	一个简单的多线程程序调试	100
7.2.1	MassiveThreads 程序	100
7.2.2	LLDB 调试 MassiveThreads	101
7.2.3	Windbg 调试 MassiveThreads	108
7.2.4	MassiveThreads 调试总结	113