

<<程序员的自我修养>>

图书基本信息

书名：<<程序员的自我修养>>

13位ISBN编号：9787121085116

10位ISBN编号：7121085119

出版时间：2009-4

出版时间：电子工业出版社

作者：俞甲子,石凡,潘爱民

页数：459

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<程序员的自我修养>>

前言

作者序1两年前，甲子跟我提起，他在考虑写一本讲述计算机程序基本工作原理的书，由于代码背后的许多细节现在难以找到完整而又实用的资料，因此，系统性地讲述这些技术要素一定非常有意义。这是我非常感兴趣的话题，因为最近几年来，我每次给学生讲课或作技术报告时，经常会提到程序背后的一些细节知识，而当有人请我推荐一些参考资料时，我很难想得出有什么恰当的参考书可供学习。

我自己也曾想过要写一点这方面的书，只是一直下不了决心做这件事情。

甲子的提议让我意识到，写这样一本书的机会来了。

于是，我们认真规划了书的选题。

按我的建议，这应该是三卷本的书，每卷独立，合起来成一体系。

第一卷是基础篇，介绍程序的基本运行过程，即是您现在看到的这本书。

其他两卷还需要时日和机缘。

在过去两年中，我曾经以“Inside Windows Programs”为题在多所高校作过报告，旨在介绍Windows程序背后的一些支撑技术。

对于正在学习计算机或软件专业的学生，或者正在从事软件开发的工程师们，我认为理解这些支撑技术是很有必要的。

试想，即使一个简单的“Hello World!”程序，也依赖于背后的输入输出库（或流库）及系统提供的模块，这种依赖性已经成为现代软件在操作系统环境下运行的一个必要条件。

然而，有关这些支撑技术的系统性资料却少而又少，虽然Internet上并不缺乏任何一方面的细节信息，但是，能将程序的编译和运行过程所涉及的各种技术全面地串连起来介绍的，却尚未有先例。

甲子曾经在2006年夏天跟我实习过两个月，他帮我搭建了一个在Windows已有体系结构下将交换空间重定向到远程机器物理内存的原型系统。

完成这一系统并非易事，而且甲子事前并无Windows内核编程经验，但是，他凭借扎实的计算机系统软件功底，成功地打通了从页面错误（page fault）异常例程到远程机器内存管理器之间的数据通路。

在这一段实习经历中，我不仅看到了他驾驭代码和系统的能力，也感受到他做事认真负责的态度。

因此，当他提出要写一本介绍程序基础的书时，我认为他是非常合适的人选。

考虑到写书的艰巨性，他推荐石凡同学加入进来，这才有了我们三个人的组合。

我原先担心写作的进度，毕竟写这样一本书需要大量的时间投入。

幸运的是，在甲子和石凡的不懈努力下，这本书终于面市了。

本书讲解的内容，涉及在Windows和Linux两个系统平台上，一个应用程序在编译、链接和运行时刻所发生的各种事项，包括：代码指令是如何保存的，库文件如何与应用程序代码静态链接，应用程序如何被装载到内存中并开始运行，动态链接如何实现，C/C++运行库如何工作，以及操作系统提供的系统服务是如何被调用的。

每个技术专题都配备了大量图示和代码实例，力求将复杂的机制以简洁的形式表达出来。

本书最后还提供了一个小巧且跨平台的C/C++运行库MiniCRT，综合展示了与运行库相关的各种技术。

关于写作这本书的功劳，我不敢掠美。

在创作之初，包括拟定提纲及甄选内容方面，我跟甲子有过认真而细致的讨论；在写作过程中，我对甲子和石凡的初稿提出过一些建议，尤其在表述方面，同时我也协助他们与编辑进行了沟通和交流。对于正文的内容，我并无实质性的贡献，但基于我对甲子和石凡两位年轻人的了解，我相信他们自身的技术实践功底，以及足够的技术阐释能力。

我期待这本书能够真正地提升程序员的自我修养，让程序员总是生活在“知其然，更知其所以然”的代码曼妙中。

最后，我要感谢这本书的四位编辑，他们是何艳、方舟、刘铁锋和陈元玉，谢谢他们为这本书付出的努力。

还要感谢博文视点团队的负责人周筠女士，谢谢她给予两位年轻作者的扶持和关爱。

<<程序员的自我修养>>

潘爱民2009年2月于北京微软七、作者序2两年前，我在浙江大学的一著名BBS的C++板块上担任版主，而俞甲子则是板上的资深版友（以及前版主）。

那时候我对链接装载、运行库等内容比较感兴趣，自己摸索着在博客上写了一篇关于链接的入门文章，而这就是一切的开始。

我猜想俞甲子可能对写这么一本书早有想法，看到我的文章正好找到了同路人。

他找到了我和潘爱民老师，我们一拍即合，就开始了这长达两年的写作历程。

考虑到当时俞甲子已经在链接部分有了相当的积累，因此我不得不放弃我最有兴趣的一部分转而在运行环境上做文章。

我把glibc和msvcrt的源代码翻了个底朝天，了解到了许多平时不可能接触到的内幕和技术细节。

事实上，这基本是一个现学现卖的过程，我一边学习着新的知识，一边把新知识组织整理写成文字。

读者在看某些章节的时候，会发现这些章节的讲解过程就是一个源代码的挖掘过程，这实际上也就是我的学习过程。

学习研究他人的代码是枯燥而耗时的，我很高兴能够做这样一个先行者，将我的经验写进书里，让读者能够避免重复劳动，直接获得其中的经验和关键技术。

本书所讲的内容不是活跃在当今IT舞台上的高新技术，也不是雄踞计算机某个领域的王牌霸主，而是默默服务于所有计算机应用的扫地僧。

也许阅读本书不能够直接在平时学习工作中的生产力上得到体现，但了解计算机的台前幕后会对读者产生潜移默化的影响。

当你的程序无法启动的时候，你可能会在脑海里多设想一种可能性；当你的代码链接失败的时候，你可能会更快地意识到问题的所在；当你的程序发生非法操作的时候，你可能不至于面对微软的错误报告毫无头绪。

有人总爱用“时效性”评价当今的IT技术。

仿佛一项技术的生存期就只有几年。

我不能说这样的想法是错误的，如今的技术的确在飞速地更替和发展。

但是本书所讲的技术，大多是成型在十年前，乃至二十年前，它们是整个计算机行业技术的根本，也几乎是现在所有计算机应用的基础。

在当今的计算机技术发生根本性变革之前，这些技术还将继续存在并保持活力。

我很荣幸能够有机会和读者分享这些技术，但写作水平有限（我在语文课上历来不是个好学生），最终在文字和结构上颇有缺憾，只能在这里说一声抱歉。

在这里要感谢我小学、初中和高中的语文老师，谢谢你们当初对我的教导，尽管最终可能辜负了你们的希望。

感谢潘老师、博文视点的编辑及所有支持我们的朋友们，谢谢你们对我们的帮助。

最后要感谢我的父母，没有你们，我永远不可能走到今天这一步。

石凡2009年2月于杭州八、作者序3CPU体系结构、汇编、C语言（包括C++）和操作系统，永远都是编程大师们的护身法宝，就如同少林寺的《易筋经》，是最为上乘的武功；学会了《易筋经》，你将无所不能，任你创造武功；学会了编程“易筋经”，大师们可以任意开发操作系统、编译器，甚至是开发一种新的程序设计语言！

——佚名念书的时候，作为标准的爱好技术的宅男，每天扫一遍各大高校BBS的技术版面，基本好比一日三餐一样平常。

我对计算机技术方面的口味很杂，从汇编版到C++到Linux内核开发、Linux应用开发、游戏开发、网络、编程语言、体系结构、移动开发、开源闭源我都会参上一脚。

我始终认为技术优劣取决于需求，与很多持有“编程语言血统论”的程序开发者不同，我不认为C++或Java本身有什么直接可比性，或者OOP与函数式编程谁优谁劣，我始终坚持认为作为开发者，MOP（Market/Money Oriented Programming）才是唯一不变的编程范式。

于是我往往不参与那些技术、平台、语言教派之间的宗教战争，这种论战基本上每周都会有，我很佩服论战各方见多识广、旁征博引、高屋建瓴的论断，但我往往只是灌灌水调节一下思绪。

相反，我很关注一些与语言、平台等相对独立的基本的系统概念方面的问题，这些问题比较具体，也

<<程序员的自我修养>>

比较实用，比如：为什么程序是从main开始执行？

“ malloc分配的空间是连续的吗？

” “ PE/ELF文件里面存的是什么？

” “ 我想写一个不需要操作系统可以直接在硬件上跑的程序该怎么做？

” “ 目标文件是什么？

链接又是什么？

” “ 为什么这段程序链接时报错？

” “ 句柄到底是什么东西？

” 这些问题看似很简单但实际上有很多值得深入挖掘的地方，比如第一个问题围绕着main函数执行前后可以延伸出一大堆问题：程序入口、运行库初始化、全局/静态对象构造析构、静态和动态链接时程序的初始化和装载等。

我们把这些问题归结起来，发现主要是三个很大的而且连贯的主题，那就是“链接、装载和库”。

事实上，现在市面上和网络上能找到的计算机技术方面的书籍和资料中，什么都很齐全，唯独关于这三个主题的讨论十分稀缺，即使能找到一些也是犹如残缺的典籍，不仅不完整而且很多已经过时了。

关于现在通用的Windows和Linux平台的链接、装载及PE/ELF文件的详细分析，实在很少见。

这个领域中，最为完整、也最为权威的莫过于John R. Levine的《Linkers & Loaders》，这本书我也前前后后通读了好几遍，虽然它对链接和装载方面的描述较为完整，但是过于理论化，对于实际的系统机制描述则过于简略。

我始终认为对于一个问题比较好的描述方式，是由一个很小很简单的问题或示例入手，层层剥开深入挖掘，不仅探究每个机制“怎么做”，而且要理解它们“为什么这样做”，力求深入浅出、图文并茂，尽力把每一步细节都呈现给读者。

这是我一贯的想法，也是我们在本书中努力试图达到的效果。

第一次有想写这样一本书的念头是在2006年底，当时我正在念研一，想起未来还有一年多漫长而又相对空闲的研究生生涯，觉得写一本这样的书大概是比较好的“消遣活动”。

于是我第一时间想到了在微软研究院实习时的导师潘爱民老师，潘老师在写作技术书籍方面有很深的功底和丰富的经验。

我把想法告诉潘老师以后，他十分支持，于是我又找到了当时刚好保送研究生、时间上也相对充裕的石凡，我们三个都对这个选题十分感兴趣，可谓一拍即合。

当时也没多想，以为写书大概也就跟BBS发帖连载差不多吧。

一旦写起来才发现自己完全轻视了写书的工作量。

书中的每一个章节、每一个小段、每一个例子甚至每一个用词有时候都要斟酌很久，生怕用得不恰当误导了读者。

“误人子弟”这四个字罪名可不轻，大有推出午门斩首五遍以儆效尤之过。

写书的时间的确很仓促，虽然我们都是在读研时写的，按理说相对于已经工作的作者来讲，已经是有很多闲余的时间了，但还是经常手忙脚乱。

想到以前看书看到作者写的序里，经常使用“时间仓促，水平有限”的话，推想作者不过是出于谦虚不免要客套一下。

现在轮到自己写序了，终于感觉到了这八个字的分量。

即使到现在已近完稿，我们还是心里十分忐忑，因为还有不少地方的确写得不够完善。

也听到了很多第一批读者的反馈意见，很多建议都正中这本书的软肋，我们也根据大家的意见又一次进行了修改，这已经是反反复复的第N次修订了。

这本书前前后后花了两年多的时间一直没有完稿，由于截稿时间快到了，我们才终于定稿，因为实在没有办法做到完美，只能向无限接近完美努力。

最后，我们在“著”和“编著”之间犹豫了很久，想到本书凝聚了我们很多的心血，还是诚惶诚恐地写上了“著”字，权当给自己壮胆了。

我们也相信，本书虽然没做到完美，但是它一定会给你带来一些你以前想看、想了解而又找不到的东西。

<<程序员的自我修养>>

或者以前在编程过程中困惑了你很久，但始终没有找到解释的问题，当在本书中终于找到答案且大呼“原来如此！”

”时，我们也就很欣慰了！

关于本书的书名笔者们也讨论了很久，征询过很多意见，最终还是决定用“程序员的自我修养”作为书名，将“链接、装载与库”作为副标题。

书名源自于俄罗斯的演员斯坦尼斯拉夫斯基创作的《演员的自我修养》，作者为了写这本书前前后后修改了三十年之久，临终前才同意不再修改，拿去出版。

使用这个书名一方面是本书的内容的确不是介绍一门新的编程语言或展示一些实用的编程技术，而是介绍程序运行背后的机制和由来，可以看作是程序员的一种“修养”；另一方面是向斯坦尼斯拉夫斯基致敬，向他对作品精益求精的精神致敬。

在本书的创作过程中，很多人对我们的支持和帮助难以言表。

这里我要感谢博文视点的几位编辑何艳、方舟、刘铁锋和陈元玉等，他们为本书付出了很多心血；特别要感谢博文视点的周筠老师，这本书能够面世离不开她的支持和努力。

另外也要感谢浙江大学的张晓龙博士，他为本书提出了很多建议，并且贡献了“DLL HELL”一节。

俞甲子2009年2月于杭州

<<程序员的自我修养>>

内容概要

本书主要介绍系统软件的运行机制和原理，涉及在Windows和Linux两个系统平台上，一个应用程序在编译、链接和运行时刻所发生的各种事项，包括：代码指令是如何保存的，库文件如何与应用程序代码静态链接，应用程序如何被装载到内存中并开始运行，动态链接如何实现，C/C++运行库的工作原理，以及操作系统提供的系统服务是如何被调用的。

每个技术专题都配备了大量图、表和代码实例，力求将复杂的机制以简洁的形式表达出来。

本书最后还提供了一个小巧且跨平台的C/C++运行库MiniCRT，综合展示了与运行库相关的各种技术

。

本书对装载、链接和库进行了深入浅出的剖析，并且辅以大量的例子和图表，可以作为计算机软件专业和其他相关专业大学本科高年级学生深入学习系统软件的参考书。

同时，还可作为各行业从事软件开发的工程师、研究人员以及其他对系统软件实现机制和技术感兴趣者的自学教材。

<<程序员的自我修养>>

书籍目录

第1部分 简介	第1章 温故而知新	1.1 从HELLO WORLD说起	1.2 万变不离其宗	1.3 站得高, 望得远	1.4 操作系统做什么	1.5 内存不够怎么办	1.6 众人拾柴火焰高	1.7 本章小结
第2部分 静态链接	第2章 编译和链接	2.1 被隐藏了的过程	2.2 编译器做了什么	2.3 链接器年龄比编译器长	2.4 模块拼装——静态链接	2.5 本章小结	第3章 目标文件里有什么	3.1 目标文件的格式
	3.2 目标文件是什么样的	3.3 挖掘SIMPLESECTION.O	3.4 ELF文件结构描述	3.5 链接的接口——符号	3.6 调试信息	3.7 本章小结	第4章 静态链接	4.1 空间与地址分配
	4.2 符号解析与重定位	4.3 COMMON块	4.4 C++相关问题	4.5 静态库链接	4.6 链接过程控制	4.7 BFD库	4.8 本章小结	第5章 WINDOWS PE/COFF
	5.1 WINDOWS的 二进制文件格式PE/COFF	134	5.2 PE的前身——COFF	5.3 链接指示信息	5.4 调试信息	5.5 大家都有符号表	5.6 WINDOWS下的ELF——PE	5.7 本章小结
第3部分 装载与动态链接	第6章 可执行文件的装载与进程	6.1 进程虚拟地址空间	6.2 装载的方式	6.3 从操作系统角度看可执行文件的装载	6.4 进程虚存空间分布	6.5 LINUX内核装载ELF过程简介	6.6 WINDOWS PE的装载	6.7 本章小结
	第7章 动态链接	7.1 为什么要动态链接	7.2 简单的动态链接例子	7.3 地址无关代码	7.4 延迟绑定(PLT)	7.5 动态链接相关结构	7.6 动态链接的步骤和实现	7.7 显式运行时链接
	7.8 本章小结	第8章 LINUX共享库的组织	8.1 共享库版本	8.2 符号版本	8.3 共享库系统路径	8.4 共享库查找过程	8.5 环境变量	8.6 共享库的创建和安装
	8.7 本章小结	第9章 WINDOWS下的动态链接	9.1 DLL简介	9.2 符号导出导入表	9.3 DLL优化	9.4 C++与动态链接	9.5 DLL HELL	9.6 本章小结
第4部分 库与运行库	第10章 内存	10.1 程序的内存布局	10.2 栈与调用惯例	10.3 堆与内存管理	10.4 本章小结	第11章 运行库	11.1 入口函数和程序初始化	11.2 C/C++运行库
	11.3 运行库与多线程	11.4 C++全局构造与析构	11.5 FREAD实现	11.6 本章小结	第12章 系统调用与API	12.1 系统调用介绍	12.2 系统调用原理	12.3 WINDOWS API
	12.4 本章小结	第13章 运行库实现	13.1 C语言运行库	13.2 如何使用MINI CRT	13.3 C++运行库实现	13.4 如何使用MINI CRT++	13.5 本章小结	附录A
	A.1 字节序(BYTE ORDER)	A.2 ELF常见段	A.3 常用开发工具命令行参考	索引				

<<程序员的自我修养>>

章节摘录

第1部分 简介 第1章 温故而知新 1.2 万变不离其宗 计算机是个非常广泛的概念，大到占用数层楼的用于科学计算的超级计算机，小到手机上的嵌入式芯片都可以被称为计算机。

虽然它们的外形、结构和性能都千差万别，但至少它们都有“计算”这个概念。

在本书里面，我们将计算机的范围限定在最为流行、使用最广泛的PC机，更具体地讲是采用兼容x86指令集的32位CPU的个人计算机。

原因很简单：因为笔者手上目前只有这种类型的计算机可供操作和实验，不过相信90%以上的读者也是，所以在这一点上我们很快能达成共识。

其实选择具体哪种平台并不是最关键的，虽然各种平台的软硬件差别很多，但是本质上它们的基本概念和工作原理都是一样的，只要我们能够掌握一种平台上的技术，那么其他的平台都是大同小异的，很轻松地可以举一反三。

所以我们相信，只有你能够深刻地理解x86平台下的系统软件背后的机理，当有一天你需要在MIPS指令集的嵌入式平台上做开发，或者需要为64位的Windows或linux开发应用程序的时候，你很快就能找到它们之间的相通之处。

撇开计算机硬件中纷繁复杂的各种设备、芯片及外围接口等，站在软件开发者的角度看，我们只须抓住硬件的几个关键部件。

对于系统程序开发者来说，计算机多如牛毛的硬件设备中，有三个部件最为关键，它们分别是中央处理器CPU、内存和I/O控制芯片，这三个部件几乎就是计算机的核心了；对于普通应用程序开发者来说，他们似乎除了要关心CPU以外，其他的硬件细节基本不用关心，对于一些高级平台的开发者来说（如Java、.NET或脚本语言开发者），连CPU都不需要关心，因为这些平台为它们提供了一个通用的抽象的计算机，他们只要关心这个象的计算机就可以了。

.....

<<程序员的自我修养>>

媒体关注与评论

这是一本深入阐述链接、装载和库等问题的优秀图书，读来让人愉悦，你从中可以清晰地了解程序的前世今生，彻底理解敲人的代码如何变成程序任系统中运行。

通读本书不管对于开发还是trouble shootin9都会很有帮助。

建议每一位希望从事系统开发、或希望更实务地理解操作系统和编译器、或不满足于只写代码的优秀程序员都拥有这样一本书。

——邹飞，趋势科技（中国）研发中心资深软件要程师本书从大处着眼，小处着手，以通俗易懂的语言，深入浅出地对系统软件的底层形成机制进行条分缕析，正合药山禅师所谓“高高山顶立，深深海底行”。

循着作者的思绪一路走来，有如醍醐灌顶。

畅快淋漓。

非常高兴有预览此书初稿的宝贵机会，我在浏览书稿和查核相关资料的过程中，学到了很多以前未知或知之不深的内容。

——冯亮，阿里巴巴（中国）网络技术有限公司运维部系统架构师

<<程序员的自我修养>>

编辑推荐

《程序员的自我修养：链接、装载与库》深入浅出地对系统软件的底层形成机制进行条分缕析真正提升程序员的自我修养

<<程序员的自我修养>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>