

<<链接器和加载器>>

图书基本信息

书名：<<链接器和加载器>>

13位ISBN编号：9787811245714

10位ISBN编号：781124571X

出版时间：2009-9

出版时间：北京航空航天大学出版社

作者：John R.Levine

页数：184

译者：李勇

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>

<<链接器和加载器>>

前言

几乎从有计算机以来，链接器和加载器就是软件开发工具包中的一部分，因为它们允许使用模块（而不是一个单独的大文件）来构建程序的关键工具。

早在1947年，程序员们就开始使用原始的加载器：将程序的例程存储在多个不同的磁带上，并将它们合并、重定位为一个程序。

在20世纪60年代早期，这些加载器就已经发展得相当完善了。

由于那时内存很贵且容量有限，计算机的速度很慢（以今天的标准），为了创建复杂的内存覆盖策略（以将大容量的程序加载到小容量内存中），以及重新编辑先前链接过的文件（以节省重新创建程序的时间），这些链接器都包含了很多复杂的特性。

20世纪七八十年代，链接技术几乎没有什么进展。

链接器趋向于更加简单，虚拟内存技术将应用程序和覆盖机制中的大多数存储管理工作都转移给了操作系统，越来越快的计算机和越来越大的磁盘也使得重新链接一个程序或替换个别模块比仅仅链接改变过的地方更加容易了。

从20世纪90年代起，链接器又开始变得复杂起来，增加了诸多现代特性，包括对动态链接共享库的支持和对C++独特要求的支持。

同时，像IA64那样具有宽指令字和编译时访存调度特性的先进处理器架构，也需要将一些新的特性加入到链接器中，以确保在被链接的程序中可以满足代码的这些复杂需求。

<<链接器和加载器>>

内容概要

本书讲述构建程序的关键工具——链接器和加载器，内容包括链接和加载、体系结构、目标文件、存储分配、符号管理、库、重定位、加载和覆盖、共享库、动态链接和加载、动态链接的共享库，以及着眼于成熟的现代链接器所做的一些变化；并介绍一个持续的实践项目，即使用Perl语言开发一个可用的小链接器。

本书适合高校计算机相关专业的学生、实习程序员、语言设计者和开发人员阅读参考。

作者简介：莱文（John R. Levine）是很多书籍的作者或合作者，包括Lex & Yacc（O'Reilly），Programming for Graphics Files in C and C++（Wiley），以及7-helntemetforDummies（IDG）。他还是Journal of C Language Translation的荣誉退休发行人、comp.compilers新闻组的长期仲裁人员，以及某个最早的商用Fortran 77编译器的创建者。他在耶鲁大学获得了计算机科学的博士学位。

<<链接器和加载器>>

书籍目录

第1章 链接和加载 1.1 链接器和加载器做什么？

1.2 地址绑定：从历史的角度 1.3 链接与加载 1.4 编译器驱动 1.5 链接：一个真实的例子
练习第2章 体系结构的问题 2.1 应用程序二进制接口 2.2 内存地址 2.3 地址构成 2.4 指令格式 2.5 过程调用和寻址能力 2.6 数据和指令引用 2.7 分页和虚拟内存 2.8 Intel 386分段
2.9 嵌入式体系结构 练习第3章 目标文件 3.1 目标文件中都有什么？
3.2 空目标文件格式：MS-DOS的COM文件 3.3 代码区段：UNIX的a.out文件 3.4 重定位：MS
—DOS的EXE文件 3.5 符号和重定位 3.6 可重定位的a.out格式 3.7 UNIX的ELF格式 3.8 IBM 360
目标格式 3.9 微软可移植、可执行体格式 3.10 Intel/Microsoft的OMF文件格式 3.11 不同目标格式
的比较 练习 项目第4章 存储空间分配 4.1 段和地址 4.2 简单的存储布局 4.3 多种段类型
4.4 段与页面的对齐 4.5 公共块和其他特殊段 4.6 链接器控制脚本 4.7 实际中的存储分配
练习 项目第5章 符号管理 5.1 绑定和名字解析 5.2 符号表格式 5.3 名称修改 5.4 弱外
部符号和其他类型符号 5.5 维护调试信息 练习 项目第6章 库第7章 重定位第8章 加载
和覆盖第9章 共享库第10章 动态链接和加载第11章 高级技术参考文献

<<链接器和加载器>>

章节摘录

第1章 链接和加载 1.1 链接器和加载器做什么？

任何一个链接器和加载器的基本工作都非常简单：将更抽象的名字与更底层的名字绑定起来，好让程序员使用更抽象的名字编写代码。

也就是说，它可以将程序员编写的一个诸如 `get_line` 的名字绑定到“iosys模块内可执行代码的612字节处”，或者可以采用诸如“这个模块的静态数据开始的第450字节处”这样更抽象的数字地址，并将其绑定到数字地址上。

1.2 地址绑定：从历史的角度 链接器和加载器是做什么的？

深入理解的方法就是看看它们在计算机编程系统的发展中充当了什么角色。

最早的计算机完全是用机器语言进行编程的。

程序员需要在纸质表格上写下符号化的程序，然后手工将其汇编为机器码，通过开关、纸带或卡片将其输入到计算机中（真正的高手可以在开关上直接编码）。

如果程序员使用符号化的地址，那他就得手工完成符号到地址的绑定。

如果后来发现需要添加或删除一条指令，那么整个程序都必须手工检查一遍，并将所有被添加或删除指令影响的地址都进行修改。

这个问题就在于名字和地址绑定得过早了。

汇编器通过让程序员使用符号化名字编写程序，然后由程序将名字绑定到机器地址的方法解决了这个问题。

如果程序被改变了，那么程序员必须重新汇编它，但是地址分配的工作已经从程序员转给计算机了。

<<链接器和加载器>>

媒体关注与评论

“我很享受阅读这本对实现链接器和加载器的众多技术和挑战进行有效概述的书。虽然书中的多数例子都集中在今天被广泛使用的三种计算机体系结构上，但这本书也包含了很多描述过去的一些有趣和古怪的计算机体系结构的注解。通过这些真实的战例，我断定作者本人真正经历了这些事情并存活了下来给我们讲述这个故事。”
——Guy Steele

<<链接器和加载器>>

编辑推荐

不管你的编程语言是什么，不管你的平台是什么，你很可能总是会涉及链接器和加载器的功能。但是你知道如何最大限度地利用它们吗？

只有现在，随着《链接器和加载器》的出版，总算有一本深入完整地彻底揭示编译时和运行时过程的权威著作了。

《链接器和加载器》首先通过实例深入浅出地阐述了在不同的编译器和操作系统中链接和加载过程的差异。

在这个基础上，作者提出了清晰实用的忠告，来帮助你创建更快、更清晰的代码。

你将会学习如何规避和Windows DLL相关的陷阱，充分利用UNIX ELF库模式等。

如果你对程序设计抱有非常认真的态度，那么你可以通过这本书充分地理解这个领域内最难懂的主题之一。

《链接器和加载器》对于编译器和操作系统课程同样也是一本理想的补充读物。

《链接器和加载器》特性 覆盖了Windows，UNIX，Linux，BeOS和其它操作系统的动态链接过程。

解释了Java链接模式，以及它是如何应用在网络小应用程序和可扩展Java代码中的。

帮助你编写更优雅、更高效的代码，以及构建能够被更加高效地编译、加载和运行的应用程序

。

包含了一个用Perl构建链接器的练习项目，项目文件可以从网络下载得到。

<<链接器和加载器>>

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>