

导 读

你将学到什么

本书将详细描述现在流行的 Windows 和 Linux 操作系统下各自的可执行文件、目标文件格式；普通 C/C++ 程序代码如何被编译成目标文件及程序在目标文件中如何存储；目标文件如何被链接器链接到一起，并且形成可执行文件；目标文件在链接时符号处理、重定位和地址分配如何进行；可执行文件如何被装载并且执行；可执行文件与进程的虚拟空间之间如何映射；什么是动态链接，为什么要进行动态链接；Windows 和 Linux 如何进行动态链接及动态链接时的相关问题；什么是堆，什么是栈；函数调用惯例；运行库，Glibc 和 MSVC CRT 的实现分析；系统调用与 API；最后我们自己还实现了一个 Mini CRT。

应当具备的基础知识

在本书中，我们尽量避免要求读者有很多的基础知识，但难免有些要求。其中包括对 C/C++ 编程语言的基本了解、x86 汇编语言基础、操作系统基本概念及基本编程技巧和计算机系统结构的基本概念。

本书的组织

本书分为 4 大部分，分别如下。

第 1 部分 简介

第 1 章 温故而知新

介绍基本的背景知识，包括硬件、操作系统、线程等。

第 2 部分 静态链接

第 2 章 编译和链接

介绍编译和链接的基本概念和步骤。

第 3 章 目标文件里有什么

介绍 COFF 目标文件格式和源代码编译后如何在目标文件中存储。

第 4 章 静态链接

介绍静态链接与静态库链接的过程和步骤。

第 5 章 Windows PE/COFF

介绍 Windows 平台的目标文件和可执行文件格式。

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

介绍进程的概念、进程地址空间的分布和可执行文件映射装载过程。

第 7 章 动态链接

以 Linux 下的 .so 共享库为基础详细分析了动态链接的过程。

第 8 章 Linux 共享库的组织

介绍 Linux 下共享库文件的分布和组织。

第 9 章 Windows 下的动态链接

介绍 Windows 系统下 DLL 动态链接机制。

第 4 部分 库与运行库

第 10 章 内存

主要介绍堆与栈、堆的分配算法、函数调用栈分布。

第 11 章 运行库

主要介绍运行库的概念、C/C++ 运行库、Glibc 和 MSVC CRT、运行库如何实现 C++ 全局构造和析构及以 fread() 库函数为例对运行库进行剖析。

第 12 章 系统调用与 API

主要介绍 Linux 和 Windows 的系统调用及 Windows 的 API。

第 13 章 运行库实现

本章主要实现了一个支持堆、基本文件操作、格式化字符串、基本输入输出、C++ new/delete、C++ string、C++ 全局构造和析构的 Mini CRT。

编译本书的程序

编译本书中所有的示例代码，在 Windows 平台下可使用 Microsoft Visual C++ 2005 或 2008，操作系统为 Windows XP sp3。读者可以去微软的官方网站免费下载 Visual C++ 2008 Express 版：

<http://www.microsoft.com/express/vc/>

Linux 下使用的 GCC 4.1.2，ld 版本为 2.18，Glibc 和 ld-linux.so 的版本为 2.6.1，操作系统为 Ubuntu 7.04。

联系博文视点

您可以通过如下方式与本书的出版方取得联系。

读者信箱: reader@broadview.com.cn

投稿邮箱: bvtougao@gmail.com

北京博文视点资讯有限公司 (武汉分部)

湖北省 武汉市 洪山区 吴家湾 邮科院路特 1 号 湖北信息产业科技大厦 1402 室

邮政编码: 430074

电 话: 027-87690813

传 真: 027-87690595

若您希望参加博文视点的有奖读者调查, 或对写作和翻译感兴趣, 欢迎您访问: <http://bv.csdn.net>

关于本书的勘误、资源下载及博文视点的最新书讯, 欢迎您访问博文视点官方博客:
<http://blog.csdn.net/bvbook>