<<Linux高级程序设计>>

图书基本信息

书名：<<Linux高级程序设计>>

13位ISBN编号：9787115179104

10位ISBN编号：7115179107

出版时间：2008-7

出版时间：人民邮电出版社

作者：Jon Masters,Richard Blum

页数：390

字数：660000

译者：陈健

<<Linux高级程序设计>>

内容概要

本书是Linux 程序设计领域的一部力作，讲解了大量程序员需要掌握的关键知识点，包括Linux 开发中的基本工具、Linux 系统编程、Linux 桌面开发以及Linux 与Web 开发。
书中包括大量有益的经验之谈和富于启发的示例。

本书主要针对已有一定Linux 开发经验或者从其他平台转到Linux 平台的专业程序员，同样也适合想更多了解系统以解决实际问题的Linux 使用者。

作者简介

Jon Masters著名Linux内核工程师。
目前效力于Red Hat公司。
13岁取得计算机科学学士学位，创造了英国记录。
他精通Linux内核引擎、Unix系统管理、基于Linux的嵌入式系统开发，而且在网络、安全等领域也颇有造诣。
目前正在负责维护Module—init—tools—Linux官方的一个工具包，

<<Linux高级程序设计>>

书籍目录

章节摘录

第1章 Linux简介为Linux编写软件的一个最大障碍是弄明I刍Linux是什么和它不是什么。
不同的人对Linux有着不同的理解。
虽然如今大多数用户都随意地将整个基于Linux的系统称为Linux，但从技术角度来说，Linux本身是由芬兰人LinusTorvalds编写的一个操作系统内核。
在短短几年时间里，Linux迅速发展并被全球一些最大的企业和最强大的计算机用户所广泛接受。
Linux现在已成为一个提供高收益和企业质量的操作系统。
它既用于一些最大型的超级计算机，也用在许多你根本不会想到的底层由Linux支持的最小型装置中。
然而，这样一个在现代计算机领域中流行的大品牌却并不属于任何一家公司。
Linux之所以会这么成功，是因为有数以千计来自世界各地的开发者在坚持不懈地努力来完善它。
这些开发者和你一样，对编写高质量的软件有浓厚的兴趣，并从Linux社区中获取他人的经验。
不管Linux对你意味着什么，你选择本书是因为你想了解更多的有关如何成为一位专业Linux程序员的知识。
当你准备开始这次学习之旅时，你将发现如果你对不同版本的Linux系统有所了解，知道如何开始对它们进行开发，并且清楚Linux开发和目前市场上其他流行平台的开发有何不同，将会对你的学习有很大的帮助。
如果你已是一位Linux专家，那么只需略读本章即可。
如果你想成为一位Linux专家，本章将会为你提供一些有用的指导。
在本章中，你将站在专业程序员的角度来学习什么是Linux以及Linux发行版的各个组件是如何组合在一起的。
你将学习Linux系统上所用的大多数自由／开放源码软件（FLoSS）的开发过程，并找到大量提供开放源码革命动力的在线社区。
最后，你还将了解至JJLinux与你之前遇到的其他操作系统的一些不同之处——我们将在本书的其余部分介绍更多这方面的内容。
1.1 Linux发展简史Linux有一个非常多样而有趣的历史，它的历史可能比你最初想象的要早得多。
事实上，Linux的继承历史跨越了30年，可以从20世纪70年代最早的UNIX系统算起。
这一事实不只是与执着的Linux狂热者有关，对读者来说也很重要，因为它至少让你对目前接触到的现代Linux系统的独特历史有一个大致的了解。
通过介绍这些历史能让你更好地理解将Linux和目前市场上的其他操作系统区分开来的细节特征，并有助于使Linux的开发更加有趣。
Linux本身的工作最早始于1991年夏天，但早在Linux存在之前，就有了GNU项目。
这个项目已花费了lof年的时间来创建很多必要的自由软件组件，其目的就是为了能创建一个完全自由的操作系统，如Linux。
如果没有GNU项目，就不会诞生Linux；同样，如果没有Linux，你可能也不会立刻去阅读GNU项目。
这两个项目彼此之间互相受益，正如你将要在本书所要讨论的主题中发现的那样。
1.1.1 GNU项目1983年，那时的RichardStallman（也称为RMS）还在麻省理工学院的人工智能实验室工作。
直到那时为止，许多软件应用程序还是以源代码的形式提供，或有源代码可用，以便用户在必要的时候针对自己的系统进行修改。
但从那时开始，已存在一种日益增长的趋势，即软件厂商只发行二进制版本的软件应用程序。
软件的源代码很快变成了公司的“商业机密”，并受到高度保护——开放源码的开发者通常将这些源代码称为“秘笈”。
GNU项目最初的目标是通过使用必要的工具从源代码开始创建一个自由的类UNIX操作系统。
该项目花了10多年的时间创建了所需的大多数工具，包括GCC编译器、GNUemacs文本编辑器和数十个其他的工具和文档。
其中许多工具都以它们的高品质和丰富的功能而闻名，例P口Gcc和GNU调试器。

GNU享有许多早期的成就，但它在20世纪80年代缺少了一个最关键的组件。
它没有自己的内核，即操作系统的核心，而是需要用户在已有的商业操作系统，如专有的UNIX上安装GNU2L具。
虽然这并不会对许多在他们自己的专用系统上使用GNUl具的用户造成什么影响，但如果GNU项目没有自己的内核，它就不是一个完整的项目。
在Linux出现之前，针对是否开发这样一个内核（如发展中的GNUHURD）的激烈争论长期以来一直存在着。
Linux从来没有真正成为RichardStallman所设想的GNU操作系统的一部分。
事实上，尽管Linux已成为新一代用户和开发者的宠儿，并且是迄今为止更受欢迎的内核，GNU项目还是一直主张在其概念性的GNU系统中使用GNUHURD微内核。
尽管如此，仍然会偶尔看到在提及一个完整的Linux系统时使用术语“ GNU／Linux ”，这是对在构建和运行任何一个现代Linux系统中扮演重要角色的众多GNUI具的认可。
1.1.2 Linux，内核Linux内核的诞生远晚于GNU项目本身，在RichardStallman发表他的最初宣言之后的10多年后它才出现。
在此之前，已有一些可供选择的操作系统被开发出来，包括HURD微内核（在狂热的内核开发者社区之外只赢得了有限的大众关注）和FhAndrewTanenbaum编写的用于教学目的的Minix微内核。
但由于种种原因，在Linux初次登台之前没有一个系统在这一段最好的时机中获得一般计算机用户的广泛认可。
就在那时，一位在赫尔辛基大学读书的年轻的芬兰学生正受制于Minix操作系统中很多他认为不合理的地方。
于是，他开始专门为他的（当时很高级的）AT386微机设计自己的操作系统。
此人就是LinusTorvalds，他将继续领导这个已创造了整个Linux产业的项目并激励着新一代。
Linus在1991年夏天开发出Linux的最初版本后，就在Usenet新闻组comp.OS.minix中发表了如下的公告：
日期：25Au99120：57：08GMT组织：赫尔辛基大学所有使用minix的人们——我正在为386（486）AT微机开发一个（免费的）操作系统（只是个人爱好，它不会像9nu那样庞大和专业）。
我从4月份开始酝酿该系统，现在已进入准备阶段。
我需要任何喜欢或不喜欢minix的朋友的反馈意见，因为我的操作系统与它有些类似（同样的文件系统物理布局（由于某些实际原因）以及其他方面）。
我已将bash（1.08）和gcc（1.40）移植到该系统中，并且它们看起来可以正常工作。
这意味着在短短几个月内我就能够在该系统中做一些实际的事情，我很想知道大多数人都希望增加哪些功能。
欢迎大家提出任何建议，但我无法保证一定会实现它们。
尽管Linus一开始很谦逊，但人们对Linux内核的兴趣却在全球迅速扩大。
不久之后，Linux就推出了多个新版本，并且一个日益增长的用户群体（他们基本上都是开发人员，因为即使是简单地安装Linux也需要大量的专业技术）正在为Linux解决各种技术难题并将一些新的构思和想法付诸实现。
现在的许多大名鼎鼎的Linux开发者都是在当时加入这一行业的。
他们享受着能够在一个现代的完全免费的类UNIX系统上工作的乐趣，而不用忍受像其他系统那样的设计复杂性。
Linux的开发者利用许多已有的GNu工具来构建Linux内核并为它开发新的特性。
事实上，在Linux出现后不久，就有越来越多的人对它发生了兴趣，Minix用户也开始切换到Linux系统上来工作——这些事情最终导致在Minix创造者AndrewTANenbaum和LinusTorvalds之间发生了一系列著名的“ 口水战 ”。
Tanenbaum至今仍坚持认为Linux的设计根本不如Minix。
从理论上来说，这可能是事实，但对其他现代操作系统而言，可以说也存在着同样的问题。
读者可以从PeterH.Salus所著的AQuarterCenturyofUNIX（Addison-Wesley，1994）中了解到更多有关Linux和其他类UNIX操作系统历史继承的信息。

1.1.3 Linux发行版随着Linux核越来越受欢迎，人们希望Linux系统也能为那些对其内部编程机理没有深入了解的用户提供很好的服务。
为了创建这样一个可用的Linux系统，需要的不仅仅只是一个Linux内核。
事实上，如今一个普通的Linux桌面系统为了能提供从系统加电到具备丰富功能的图形桌面环境（如GNOME），它利用了成千上万个独立的软件程序。
当Linux第一次发布时，它还没有那么多丰富的软件可用。
事实上，Linus开始时只有一个应用程序-GNUBorneAgainSHell（bash）。
那些曾经以受限的“单用户”模式（只运行一个bashshell）启动过Linux或UNIX系统的用户都知道这是一种什么体验。
Linus使用一个单一的bash命令行shell对早期的Linux做了大量的测试，但是，即便这样一个单一的bash也不能直接运行在Linux系统上。
它首先需要经过移植或修改才能从一个已有的系统（如Minix）转换到Linux系统上运行。
随着越来越多的人开始使用Linux并为Linux开发软件，那些有耐心编译和安装软件的用户已有大量的软件可用。
但随着时间的流逝，人们发现以一种从无到有的方式来构建每一个Linux系统显然是一种不能忍受、不可复加的梦魇，除了最有热情的用户以外，它将阻止所有其他用户体验Linux所提供的功能。
解决方法是使用Linux发行版的形式或将预先创建好的软件集以及Linux内核以软盘（或之后的光盘）形式提供给广泛的潜在用户群。
早期的Linux发行版只是简单地为那些不想自己从无到有构建整个系统的用户提供便利。
它并没有跟踪系统上已安装了哪些软件或对软件的安全删除以及新软件的添加做出处理。
直到包（package）管理软件RedHat的RPM和Debian的dpkg的出现，才使得不具备详尽专业知识的普通用户安装Linux系统成为可能。
当你在本书后续章节中学习如何为Linux发行版构建自己的Linux软件包时，将会了解到更多有关软件包管理的内容。
现代的Linux发行版的规模和大小各不相同，并且针对不同的市场。
有些版本针对的是常规的桌面型Linux用户；有些版本针对的是企业级用户，他们要求操作系统具备可扩展性和健壮的性能；有些版本甚至是专为嵌入式设备如PDA、手机和机顶盒设计的。
尽管各种Linux发行版都有各自不同的包装方式，但它们通常都有用户可以利用的共性。
例如，大多数发行版都力争在一定程度上与事实上的可兼容Linux环境标准“Linux标准化规范（LSB）”兼容。
1.1.4 自由软件与开放源码RichardStallman启动了GNU项目并成立了自由软件基金会作为一个非营利组织来负责监督该项目。
他还编写了第一版的通用公共许可证（GPL）——为Linux系统编写的大部分软件都使用GPL许可证。
GPL本身是一个很有意思的文档，因为它的目的不是限制你使用GPL授权的软件，而是保护用户和开发者获得源代码的权利。
GPL条款允许你修改Linux内核和其他GPL许可的自由软件，作为回报，你应该公布这些修改，以便其他用户使用它们（或将它们集成到指定软件的下一个正式版本中）。
例如，GPL允许你修复一个重要的应用程序如OpenOffice的bu9，或为GNOME桌面系统上的totem多媒体播放器添加定制音频文件支持。
GPL带给开发者很大的灵活性，你可以出于任何目的来使用Linux，只要你将自己的修改也同样地提供给其他用户即可。
这是关键的一点——即GPL试图保持开发过程的开放性。
对RichardStallman来说，遗憾的是，英语中还没有一个可以和法语单词libre（英语单词libertyqb自由的含义）完全相当的单词，所以很多人混淆了自由软件和免费软件的概念。
事实上，许多自由软件都是完全免费的，但也有一些公司通过销售GPL许可的软件（包括它的自由发布的源代码）来赚钱。
他们并不是通过软件本身来赚钱，而是当软件出现故障时，通过提供各种技术支持和附加的专业服务

来赚钱。
为了减少对“ 自由软件 ”一词理解上的混乱，人们提出了术语“ 开放源码 ”，这个术语成为了20世纪90年代的流行词汇。
与自由软件不同，开放源码并不特指GPL许可的软件，而是指一种对包括源代码的软件（使之可以被他人调整、调试和改进）的普遍需求，即使该源代码是在一个比GPL限制更严格的许可证下授权的。因此，有更多的软件虽然不是自由软件，但从技术上却满足开放源码的定义。
当你要修改现有的拥有GPL许可的软件时，理解GPL究竟对这项工作有哪些要求是非常重要的。
虽然你不一定在自己的程序中使用GPL许可，但你必须尊重已这么做的其他人的权利。
在因特网上有许多潜在侵害GPL许可的例子——通常是由于公司不清楚在对软件（如Linux内核）做出修改时需要将这些修改提供给用户。
当然，你并不想成为下一个这样的案例，所以请始终确保你和你的同事都了解GPL，并尽早决定你准备如何利用它开展工作。

编辑推荐

《Linux高级程序设计》主要针对已有一定Linux开发经验或者从其他平台转到Linux平台的专业程序员，同样也适合想更多了解系统以解决实际问题的Linux使用者。
“《Linux高级程序设计》出色地为其他平台的程序员揭示了Linux程序设计的复杂本质。
而且特别强调了内核开发。
为作者喝彩！
”——LinuxMagazine杂志读了《Linux程序设计（第3版）》之后还不过瘾？
《Linux高级程序设计》将为你献上一顿饕餮大餐《Linux高级程序设计》是Linux程序设计领域内的经典著作。
涵盖了各种常用的和最重要的Linux程序设计的技术和方法。
书中蕴含了作者的宝贵经验。
提供了大量的最佳实践。
无论你是有开发经验的Linux程序员。
还是从其他平台转至Linux上的专业开发者。
都能通过《Linux高级程序设计》学到最新的Linux平台开发技术。
迅速成为现代Linux程序员。

<<Linux高级程序设计>>

版权说明