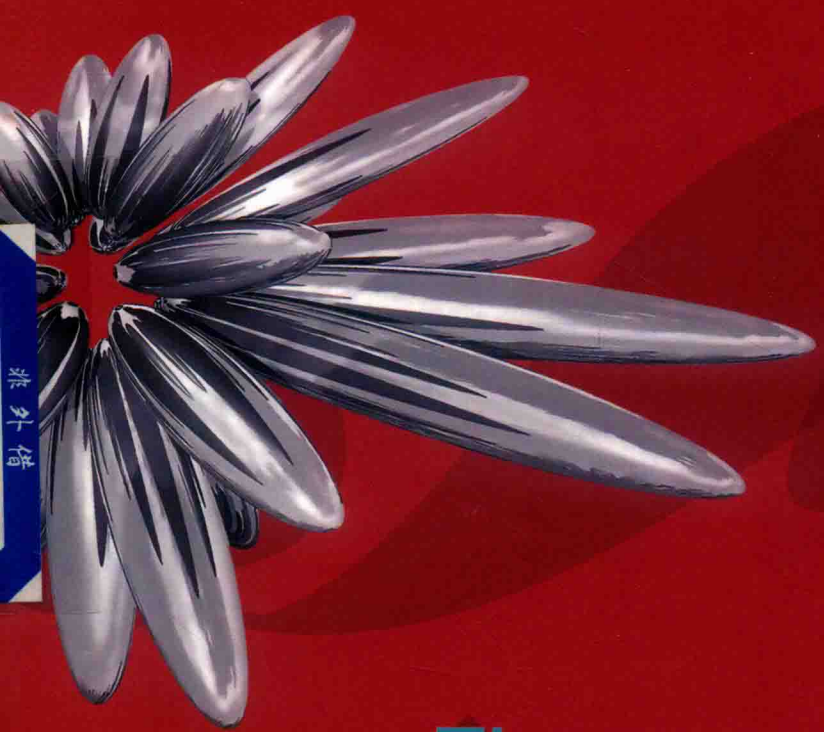
普通高等教育“十三五”规划教材

计算机类本科规划教材

Linux系统程序设计教程

◆王　凯　主编



中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

计算机类本科规划教材

Linux 系统程序设计教程

主　编　王　凯

副主编　杨　光　窦　乔　杨俊伟
余新桥　孙　斌

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书基于 Linux 内核，以 RedHat Linux 平台为例，介绍 Linux 系统程序设计的基础知识，为准备学习 Linux 系统程序开发的初学者提供方便快捷的途径。

全书共 12 章。第 1 章介绍 Linux 操作系统的基本使用方法；第 2 章介绍 Linux 平台下进行 C 语言开发需要的各种工具；第 3 章介绍 Linux 平台下常用的编程基础知识；第 4～6 章介绍文件、文件属性、目录文件相关的编程理论和方法；第 7、8 章介绍进程和线程的编程方法；第 9～11 章介绍信号、管道、信号量、共享内存、消息队列、套接字 6 种进程间通信方式；第 12 章介绍两个贯穿本书大多数知识点的综合案例。

本书可作为高等院校计算机科学与技术、软件工程、物联网工程等相关专业“Linux 程序设计”相关课程的教材，同时可供本科高年级学生自学使用，也可以作为相关工程技术人员和计算机爱好者的参考书。

图书在版编目（CIP）数据

Linux系统程序设计教程/王凯主编. —北京：电子工业出版社，2019.1
计算机类本科规划教材
ISBN 978-7-121-35855-5

Ⅰ. ①L… Ⅱ. ①王… Ⅲ. ①Linux操作系统－高等学校－教材 Ⅳ. ①TP316.85

中国版本图书馆 CIP 数据核字(2018)第 292039 号

责任编辑：凌　毅
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×1 092　1/16　印张：18　字数：496 千字
版　　次：2019 年 1 月第 1 版
印　　次：2019 年 1 月第 1 次印刷
定　　价：45.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系及邮购电话：(010)88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：(010)88254528，lingyi@phei.com.cn。

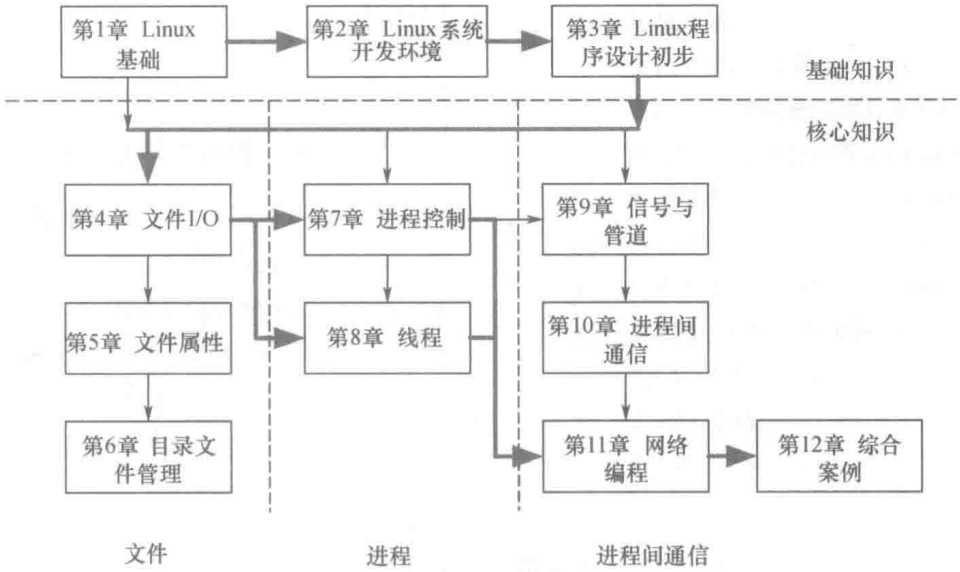
前　言

Linux 内核最初是由芬兰人 Linus Benedict Torvalds 在赫尔辛基大学上学时编写的。1991 年，Linus Benedict Torvalds 第一次发布了 Linux 内核，随后采用 GPL 协议，自此以后越来越多的程序员参与了 Linux 内核代码的编写和修改工作。目前 Linux 系统在服务器和超级计算机领域占据绝对主导地位，在手机系统领域（Android）也是占据了近三分之二的市场，另外在车载终端、智能电视等智能设备方面也占据了很大的市场份额。因此，对于 Linux 系统的使用和编程，是计算机爱好者和嵌入式领域工程技术人员非常重要的一项技能。

本书主要介绍基于 Linux 平台的文件、进程、进程间通信相关的编程理论和方法，共 12 章。第 1 章介绍 Linux 操作系统的基本使用方法；第 2 章介绍 Linux 平台下进行 C 语言开发需要的各种工具；第 3 章介绍 Linux 平台下常用的编程基础知识；第 4～6 章介绍文件、文件属性、目录文件相关的编程理论和方法；第 7、8 章介绍进程和线程的编程方法；第 9～11 章介绍信号、管道、信号量、共享内存、消息队列、Socket 6 种进程间通信方式；第 12 章介绍两个贯穿本书大多数知识点的综合案例。

本书的内容包括知识讲解和技能训练，并以案例为核心，将知识与技能有机地结合在一起。本书以典型的 Linux 系统综合案例为主线贯穿全书展开各部分的知识讲解。在每一章中除介绍相关知识外，还辅以若干个小案例的训练，从而将知识转化为解决问题的技能。

本书各章节关系图如下：



*粗线走向为综合案例贯穿本书的各个章节顺序。

*细线走向为本书中相对独立的三大知识体系：文件（第 4、5、6 章），进程（第 7、8 章），进程间通信（第 9、10、11 章）；第 1、2、3 章是这三大知识体系的支撑。

本书主要针对具有一定的 C 语言编程基础的读者，可作为高等院校计算机科学与技术、软件工程、物联网工程等相关专业“Linux 程序设计”相关课程的教材，同时可供本科高年级

学生自学使用，也可以作为相关工程技术人员和计算机爱好者的参考书。

本书配有电子课件、源程序代码、习题解答等教学资源，读者可以登录华信教育资源网（www.hxedu.com.cn）注册后免费下载。

参与编写本书的都是具有丰富一线教学经验的老师，在操作系统、Linux 管理与应用、C/C++编程开发、嵌入式软件开发、移动应用软件开发等领域具有多年的教学和实践经验。本书由王凯担任主编，杨光、窦乔、杨俊伟、余新桥、孙斌担任副主编，其中余新桥负责编写第 1 章，王凯负责编写第 2、3、11 章，孙斌负责编写第 4、5、6 章，杨俊伟负责编写第 8 章，杨光负责编写第 7、9、10 章，窦乔负责编写第 12 章。全书最后由王凯负责统稿和定稿。此外，李瑛达、陈艳秋、张福艳、李宁宁、高志君、郑纯军、贾宁等为本书做了大量的工作，在此表示感谢。

由于时间和作者的水平有限，书中难免有错误和不妥之处，请各位读者，特别是同行专家批评指正（E-mail：wk3113@163.com）。

编者

2018 年 12 月

目　录

第1章 Linux基础

操作系统是程序运行的平台和基础，因此读者首先要对 Linux 系统有一定的了解，才能进一步掌握在 Linux 系统上进行 C 语言程序设计的技能，完成 Linux 网络传输系统的开发。

本章设置了一个 SSH（Secure Shell）虚拟终端及登录的案例作为学习本章知识点后要完成的任务。该案例主要涉及远程登录访问基于 Linux 系统的服务器端，并使用常用命令对其操作。该案例要求在 Windows 中远程访问虚拟机的 Linux 系统，使用命令创建本书中所需的各章节目录，然后将这些目录下载到 Windows 的某个文件夹中。该案例涉及 Linux 命令及 SSH 客户端软件。

本章内容主要包括 UNIX/Linux 的发展历史和特点、库函数、系统调用及 Linux 常用命令。

1.1 UNIX/Linux 简介

UNIX 与 Linux 是当今主流的操作系统，通过它们的发展历史和特点，可以了解到 UNIX/Linux 系统从诞生到发展壮大的全过程，并可以理解它们成为主流操作系统的原因。

1.1.1 UNIX 简介

UNIX 是一种多用户、多任务、功能强大的操作系统，这个强大的操作系统的形成是逐步发展起来的。

1965 年，麻省理工学院、AT&T 贝尔实验室和通用电气合作进行了一个操作系统项目 Multics（Multiplexed Information and Computing System），Multics 被设计运行在 GE-645 大型主机上，但是由于整个目标过于庞大，1969 年该项目以失败告终。

AT&T 贝尔实验室的 Ken Thompson 为 Multics 系统编写了一个“Space Travel”游戏，运行速度很慢且耗费昂贵。项目失败后，为使游戏能继续运行，Ken 和 Dennis Ritchie 在贝尔实验室的一台 PDP-7 上用汇编语言开发了一个操作系统原型，用来运行这个游戏。

1971 年，Ken 申请了一台 PDP-11/24，并在其上开发了 UNIX 的第 1 版。1973 年，Ken 与 Dennis Ritchie 认为汇编语言编写的系统不容易移植，于是他们用 C 语言重新编写了 UNIX 的第 3 版内核。

UNIX 的产生引起了学术界的广泛兴趣，所以 UNIX 的第 5 版源代码被提供给各大学作为教学之用，成为当时操作系统课程中的范例。各大学及公司开始通过 UNIX 源代码对 UNIX 进行了各种各样的改进和扩展。于是，UNIX 开始广泛流行。

1978 年，加州大学伯克利分校推出了以第 6 版为基础并加上一些改进和新功能的 UNIX 版本。这就是著名的“1 BSD（1st Berkeley Software Distribution）”，从而开创了 UNIX 的另一个分支：BSD 系列。同时期，AT&T 成立 USG（UNIX Support Group），将 UNIX 变成商业化的产品。从此，伯克利分校的 UNIX 便和 AT&T 的 UNIX 分庭抗礼，UNIX 就分为 System 和 BSD 这两大主流，各自蓬勃发展。

1982 年，AT&T 基于第 7 版开发了 UNIX System Ⅲ的第一个版本，这是一个商业版本，

仅供出售。为了解决 UNIX 版本混乱的情况，AT&T 综合了其他大学和公司开发的各种 UNIX，开发了 UNIX System V Release 1，这个新的 UNIX 商业发布版本不再包含源代码。

同时，其他一些公司也开始为自己的小型机或工作站开发商业版本的 UNIX 系统，有些选择 System V 作为基础版本，有些则选择了 BSD。BSD 的一名主要开发者 Bill Joy 在 BSD 基础上开发了 SunOS，并最终创办了 Sun Microsystems 公司。

BSD UNIX 不断增强的影响力引起了 AT&T 的关注，1992 年，USL（UNIX Systems Laboratories）正式对 BSD（Berkeley Software Design，Inc.）提起诉讼，声称 BSD 剽窃了他们的源代码。后来 AT&T 卖掉了 UNIX 系统实验室，接手的 Novell 公司允许伯克利分校自由发布自己的 BSD，但前提是必须将来自 AT&T 的源代码完全删除，于是诞生了 4.4 BSD Lite 版。这个版本不存在法律问题，于是 4.4 BSD Lite 版成为了现代 BSD 系统的基础版本。

此后的几十年中，UNIX 版权所有者不断变更，授权者的数量也在增加。很多大公司在取得了 UNIX 的授权之后，开发了自己的 UNIX 产品，比如 IBM 的 AIX、HP 的 HPUX、Sun 的 Solaris 和 SGI 的 IRIX。

UNIX 因其安全可靠、高效强大的特点在服务器领域得到了广泛的应用。直到 GNU/Linux 开始流行前，UNIX 一直是科学计算、大型机、超级计算机等所用的主流操作系统。

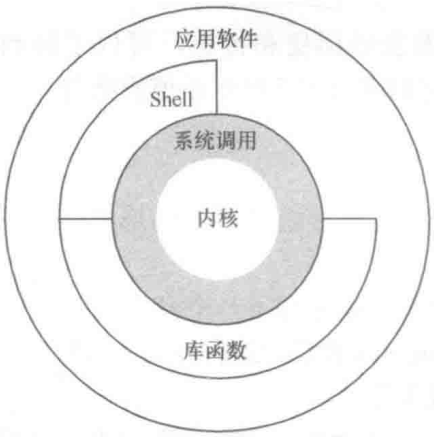


图 1.1　UNIX 体系结构

1.1.2　UNIX 体系结构

UNIX 系统的体系结构如图 1.1 所示。

内核：一组软件的集合，用来控制计算机硬件资源，提供程序运行环境。

系统调用：获取内核服务的接口。

Shell：一个特殊的应用程序，是用户和内核之间交互的界面。

库函数：构建在系统调用之上，获取一些功能的接口。

应用软件：用户使用的应用程序，基于 Shell、系统调用或库函数实现。

1.1.3　Linux 简介

UNIX 系统因其可靠和功能强大等特点广泛应用于服务器领域，同时又由于其价格昂贵，因此对于广大的 PC 用户，软件行业的大型供应商始终没有给出有效的解决方案。正在此时，出现了 MINIX 系统。

MINIX 系统是由荷兰人 Andrew S.Tanenbaum 于 1987 年开发的，主要用于学生学习操作系统原理，并且是免费使用的。

Linux 系统的创始者 Linus Benedict Torvalds 在芬兰大学学习的时候使用的就是 MINIX 系统。1991 年初，Linus 开始在一台 386 兼容微机上学习 MINIX 系统。通过学习，他逐渐不能满足于 MINIX 系统的性能，并开始酝酿开发一个新的免费操作系统。1991 年 10 月 5 日，Linus 在 comp.os.minix 新闻组上发布消息，正式向外宣布 Linux 内核系统的诞生。

Linux 系统刚开始时并不叫作 Linux，Linus 给他的操作系统取名为 FREAX，其英文含义

是怪诞的、怪物、异想天开等。在他将新的操作系统上传到 ftp.funet.fi 服务器时，管理员 Ari Lemke 很不喜欢这个名称，认为既然是 Linus 开发的操作系统，就取其谐音 Linux 作为该操作系统的名称，于是 Linux 这个名称就开始流传开来。

Linux 系统主要版本的变迁见表 1.1。

表 1.1 Linux 系统主要版本的变迁

时间	版本	特　点
1994 年 3 月	Linux 1.0	仅支持单 CPU 系统
1995 年 3 月	Linux 1.2	第一个包含多平台（Alpha，Sparc，Mips 等）支持的官方版本
1996 年 6 月	Linux 2.0	包含很多新的平台支持，但最重要的是，它是第一个支持 SMP（对称多处理器）体系的内核版本
1999 年 1 月	Linux 2.2	SMP 系统上性能的极大提升，同时支持更多的硬件
2001 年 1 月	Linux 2.4	进一步提升了 SMP 系统的扩展性，同时集成了很多用于支持桌面系统的特性：USB，PC 卡（PCMCIA）的支持，内置的即插即用，等等
2003 年 12 月	Linux 2.6	模块子系统、统一设备模型和 PnP 支持模块子系统的改变；稳定性有所提高；NUMA、NPTL 的支持；新的调度器算法；支持更多的设备等
2011 年 7 月	Linux 3.0	改进了对虚拟化和文件系统的支持，对于内核开发并没有表现出里程碑似的特征
2015 年 4 月	Linux 4.0	实时内核补丁，可以实时修补内核，而无须重启；改进图形支持，支持显示端口的音频输出，改良风扇控制；开始对 Carrizo APU 进行开发；改进对 N 系显示方案的支持；存储系统方面的改进，包括 pNFS、Btrfs RAID 5/6 的相关支持；支持更多的硬件，包括 Intel Quark SoC 以及更多 ARM 设备、IBM z13，改进了对东芝系列笔记本电脑、罗技输入设备的支持

1.1.4 Linux 版本说明

关于 Linux 的版本有两种不同的叫法：一种是内核版本，一种是发行版本。

1. 内核版本

内核指的是一个提供硬件抽象层、磁盘及文件系统控制、多任务等功能的系统软件。一个内核不是一套完整的操作系统，只有基于 Linux 内核且增加了一些外围功能的软件才能叫操作系统。

Linux 内核使用 3 种不同的版本编号方式。

第一种方式用于 1.0 版本之前（包括 1.0）。第一个版本是 0.01，紧接着是 0.02、0.03、0.10、0.11、0.12、0.95、0.96、0.97、0.98、0.99 和之后的 1.0。

第二种方式用于 1.0 版本之后到 2.6 版本，数字由 3 部分“A.B.C”组成，A 代表主版本号，B 代表次主版本号，C 代表较小的末版本号。只有在内核发生很大变化时，A 才变化。可以通过数字 B 来判断 Linux 系统是否稳定，偶数的 B 代表稳定版，奇数的 B 代表开发版。C 代表一些 bug 修复、安全更新、新特性和驱动的次数。以版本 2.4.0 为例，2 代表主版本号，4 代表次版本号，0 代表改动较小的末版本号。在版本号中，序号的第二位为偶数的版本表明这是一个可以使用的稳定版本，如 2.2.5，而序号的第二位为奇数的版本一般有一些新的东西加入，是一个并不稳定的测试版本，如 2.3.1。稳定版本的版本号来源于上一个测试版本的升级版本号，而一个稳定版本发展到完全成熟后就不再发展。

第三种方式从 2004 年的 2.6.0 版本开始使用，即使用“time-based”的方式。在 Linux 2.6 的开发过程中，内核版本的编号方式发生了很大的变化。主要的变化在于第二个数字已经不再用于表示一个内核是稳定版本还是正在开发中的版本。因此，内核开发者在当时的 2.6 版本中

对内核进行了大幅改进。只有在内核开发者必须对内核的重大修改进行测试时，才会采用一个新的内核分支 2.7。这种 2.7 的分支要么产生新的版本，要么丢弃所修改的部分而回到 2.6 版本。这样一来，两个具有相同版本号、不同发布号的内核就可能在核心部件和基本算法上有很大的差别，于是具有新发布号的内核可能潜藏着不稳定性和各种错误。为了解决这个问题，内核开发者可能发布带有补丁程序的内核版本，并用第 4 位数字表示带有不同补丁的内核版本，如 2.6.35.12。因此，2.6 版本是一种“A.B.C.D”格式，前两个数字 A.B 即“2.6”保持不变，C 随着新版本的发布而增加，D 代表一些 bug 修复、安全更新、新特性和驱动的次数。3.0 版本之后采用“A.B.C”格式，B 随着新版本的发布而增加，C 代表一些 bug 修复、安全更新、新特性和驱动的次数。第三种方式中不再使用偶数代表稳定版本、奇数代表开发版本的命名方式，如 3.7.0 代表的不是开发版本，而是稳定版本。

2．发行版本

一些组织或厂家将 Linux 系统的内核与外围实用程序和文档包装起来，并提供一些系统安装界面和系统配置、设定与管理工具，这样就构成了一种发行（distribution）版本，实际上就是 Linux 内核再加上外围实用程序组成的一个大软件包。

发行版本的版本号随发布者的不同而不同，如 SUSE、RedHat、Ubuntu、Slackware 等，RedHat Linux 9.0 或 Ubuntu 10.10 指的就是一种发行版本号。

1.1.5 Linux 特点

1．开放性

开放性是指系统遵循世界标准规范，特别是遵循开放系统互连（OSI）国际标准。凡遵循国际标准所开发的硬件和软件，都能彼此兼容，从而方便地实现互连。

2．多用户

多用户是指系统资源可以被不同用户各自使用，即每个用户对自己的资源（例如，文件、设备）有特定的权限，互不影响。Linux 和 UNIX 都具有多用户的特性。

3．多任务

多任务是现代计算机的最主要的一个特点。它是指计算机同时执行多个程序，而且各个程序的运行互相独立。Linux 系统调度每一个进程，平等地访问微处理器。由于 CPU 的处理速度非常快，因此各种应用程序好像在并行运行。事实上，从微处理器执行应用程序中的一组指令到 Linux 调度微处理器再次运行这个程序之间只有很短的时间延迟，用户是感觉不出来的。

4．良好的用户界面

Linux 为用户提供了两种系统界面：用户界面和系统调用。

Linux 的传统用户界面是基于文本的命令行界面，即 Shell，用户可以通过 Shell 程序与内核进行交互。除此之外，Shell 还有很强的程序设计能力，用户可以方便地用它编制程序，从而为用户扩充系统功能提供了更高级的手段。

Linux 还为用户提供了图形用户界面。通过鼠标、菜单、窗口、滚动条等工具，Linux 为用户呈现一个直观、易操作、交互性强的友好的图形化界面。

系统调用是包围在内核外层的界面，用户可以在编程时直接使用 Linux 系统提供的系统调用。系统通过这个界面为用户程序提供底层的、高效率的服务。

5．设备独立性

设备独立性是指操作系统把所有外部设备统一当成文件来看待，只要安装好设备驱动程

序，任何用户都可以像使用文件一样，操纵、使用这些设备，而不必知道它们的具体存在形式。

具有设备独立性的操作系统，通过把每一个外围设备看作一个独立文件来简化增加新设备的工作。当需要增加新设备时，系统管理员就在内核中增加必要的连接。这种连接（也称作设备驱动程序）保证每次调用设备提供服务时，内核以相同的方式来处理它们。当新的或更好的外部设备被开发并交付给用户时，在这些设备连接到内核后，用户就能不受限制地立即访问它们。

设备独立性的关键在于内核的适应能力。不具有设备独立性的操作系统只允许一定数量或一定种类的外部设备连接。而具有设备独立性的操作系统能够容纳任意种类及任意数量的外部设备，因为每一个外部设备都是通过其与内核的专用连接独立进行访问的。

Linux 是具有设备独立性的操作系统，其内核具有高度适应能力。随着更多的程序员加入 Linux 编程，会有更多外部设备加入各种 Linux 内核和发行版本中。另外，由于用户可以免费得到 Linux 的内核源代码，因此用户可以修改内核源代码，以便适应新增加的外部设备。

6．丰富的网络功能

完善的内置网络是 Linux 的一大特点。Linux 在通信和网络功能方面优于其他操作系统：其他操作系统不包含如此紧密地和内核结合在一起的连接网络的能力，也没有内置这些联网特性的灵活性。Linux 为用户提供了完善的、强大的网络功能。

首先，Linux 免费提供了大量支持 Internet 的软件。Internet 是在 UNIX 中建立并繁荣起来的，用户能用 Linux 与世界上的其他人通过 Internet 进行通信。

其次，用户能通过一些 Linux 命令完成内部信息或文件的传输。

最后，Linux 不仅允许进行文件和程序的传输，它还为系统管理员和技术人员提供了访问其他系统的窗口。通过这种远程访问的功能，系统管理员和技术人员能够有效地为多个系统服务，即使那些系统位于相距很远的地方。

7．可靠的系统安全

Linux 采取了许多安全技术措施，包括对读写进行权限控制、带保护的子系统、审计跟踪、核心授权等，为网络多用户环境中的用户提供了必要的安全保障。

8．良好的可移植性

可移植性是指将操作系统从一个平台转移到另一个平台时它仍然能按其自身的方式运行的能力。

Linux 是一种可移植的操作系统，能够在从微型计算机到大型计算机的任何环境中和任何平台上运行。可移植性为运行 Linux 的不同计算机平台与其他任何机器进行准确而有效的通信提供了手段，不需要另外增加特殊的和昂贵的通信接口。

1.2 库函数与系统调用

操作系统提供应用程序要求的服务，所有的操作系统都提供多种服务的入口点，程序由此向内核请求服务。各种版本的 UNIX 都提供定义明确、数量有限、可直接进入内核的入口点，这些入口点被称为系统调用。从图 1.2 中可以看出，系统调用是紧贴内核的一层，可以说系统调用是程序获取内核服务的接口，Linux 的不同版本提供了 240～260 个系统调用，见附录 A。

从用户角度观察，系统调用和库函数之间的区别不是特别重要。它们都以 C 函数形式出现，提供给用户一种功能实现的接口，需要用户输入指定的参数，调用结束得到指定的返回值。

从实现者角度观察，两者的区别就很大了。

● 库函数是在系统调用上层的函数，库函数一般指程序员可以使用的通用函数。虽然这些函数可能会调用一个或多个内核的系统调用，但是它们并不是内核的入口点。例如，printf 函数会调用 write 系统调用以输出一个字符串，但函数 strcpy（复制一字符串）和 atoi（将 ASCII 转换为整数）并不使用任何系统调用。

● 在 Linux 下，每个系统调用由两部分组成。

① 核心函数：是实现系统调用功能的代码，作为操作系统的核心驻留在内存中，是一种共享代码，运行在核心态。

② 接口函数：是提供给应用程序的 API，以库函数的形式存在于 Linux 的库 lib.a 中，该库中存放了所有系统调用的接口函数的目标代码，用汇编语言书写。其主要功能是把系统调用号、入口参数地址传给相应的核心函数，并使用户态下运行的应用程序陷入核心态。

用户编写程序时，有时可以调用库函数也可以调用系统调用来实现具体的功能。

总之，可以这样理解：系统调用与库函数从形式上来说是一样的，都以 C 语言函数形式存在，但系统调用比库函数更底层；或许从用户角度来看它们完成的功能相同，然而从实现者角度来看差别很大。

另外，Linux 系统中使用的外部命令本质上就是可执行文件，这些可执行文件是使用系统调用或库函数编写的程序经过编译生成的。

系统调用、库函数、外部命令之间的关系如图 1.2 所示。

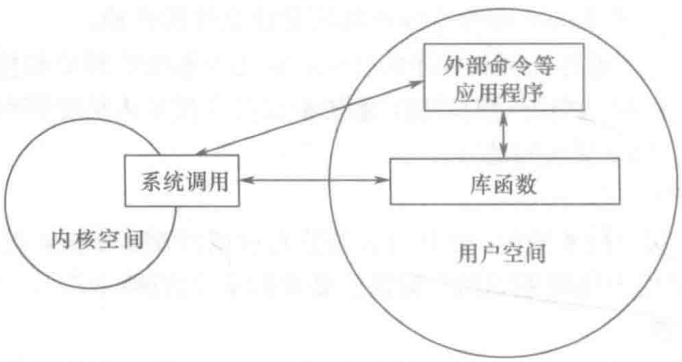


图 1.2　系统调用、库函数、外部命令之间的关系

1.3　Linux 常用命令

作为 Linux 的用户，掌握常用的 Shell 命令是必要的。尽管现在所提供的图形化界面操作起来更加直观、方便，但是 Shell 命令所提供的功能更加强大，执行效率更高。而且在很多环境下，只能进行 Shell 命令操作。下面介绍常用的 Linux 命令，供本章项目的完成和后续章节的学习参考。

1.3.1　用户和用户组命令

1. useradd 命令

【格式】useradd　[选项]　用户名

【功能】新建用户账号。只有超级用户才能使用此命令。

【选项】

-c　　全称　　指定用户的全称，即用户的注释信息。

-d　　主目录　　指定用户的主目录。
-g　　组 ID　　指定用户所属的主要组。
-G　　组 ID　　指定用户所属的附加组。
-s　　登录 Shell　　指定用户登录后启动的 Shell 类型。
-u　　用户 ID　　指定用户的 ID。

【举例】

```
# useradd  NewUser
```

【说明】

当不使用任何选项时，Linux 系统将按照默认值新建用户。系统将在/home 目录新建与用户同名的子目录作为该用户的主目录，并且还将新建一个与用户同名的组作为该用户的主要组。该用户的登录 Shell 为 Bash，UID 由系统决定。

使用 useradd 命令新建用户账号，将在/etc/passwd 文件和/etc/shadow 文件中增加新用户的记录。如果同时还新建了同名的组，那么将在/etc/group 文件和/etc/gshadow 文件中增加记录。

2．passwd 命令

【格式】passwd [-dkls][-u <-f>][用户名]

【功能】设置密码。

【选项】

-d　　用户名　　删除密码。本参数仅有超级用户才能使用。
-k　　用户名　　只能在密码过期失效后设置。
-l　　用户名　　锁住密码。
-s　　用户名　　列出密码的相关信息。本参数仅有超级用户才能使用。
-u　　用户名　　解开已上锁的账号。
-f　　用户名　　强制执行。

【举例】

```
#passwd  NewUser
```

屏幕将出现如下提示信息：

```
Changing password for user NewUser.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

注意：在输入密码时屏幕上是不会出现任何提示的。

【说明】

passwd 命令让用户可以更改自己的密码，而超级用户则能用它管理系统用户的密码。只有超级用户可以指定用户名，普通用户只能变更自己的密码。

3．userdel 命令

【格式】userdel　[-r]　用户名

【功能】删除指定的用户账号，只有超级用户才能使用此命令。

【选项】

-r　　用户名　　系统不仅将删除此用户账号，并且还将用户的主目录也一并删除。如果不使用“-r”选项，则仅删除此用户账号。

【举例】

```
# userdel  -r  NewUser
```

【说明】

如果在新建该用户时创建了同名的组，而该组当前没有其他用户，那么在删除用户的同时也将一并删除这个组。正在使用系统的用户不能被删除，必须在终止该用户所有的进程后才能删除该用户。

4. groupadd 命令

【格式】groupadd [选项]　组名

【功能】新建组，只有超级用户才能使用此命令。

【选项】

-g　　组 ID　　指定新建组的 ID。

【举例】

```
# groupadd  NewUsers
```

【说明】

利用 groupadd 命令新建组时，如果不指定 GID，则其 GID 由系统指定。groupadd 命令的执行结果将在/etc/group 文件和/etc/gshadow 文件中增加一行记录。

5. groupdel 命令

【格式】groupdel　组名

【功能】删除指定的组，只有超级用户才能使用此命令。在删除指定组之前，必须保证该组群不是任何用户的主要组，否则需要首先删除那些以此组作为主要组的用户，才能删除该组。

【举例】

```
# groupdel  NewUsers
```

1.3.2　文件和目录命令

1. mkdir 命令

【格式】mkdir　[选项]　目录

【功能】创建目录。

【选项】

-m　访问权限　　设置目录的权限。

-p　目录树　　一次创建多级目录。

【举例】

```
# mkdir  -p  directory/list
```

2. mv 命令

【格式】mv　[选项]　源文件或目录　目的文件或目录

【功能】移动或重命名文件或目录。

【选项】

-b　若存在同名文件，则覆盖之前先备份原来的文件。

-f　强制覆盖同名文件。

【举例】

```
# mv  neu  neusoft
# mv  neusoft  ../
```

3. cp 命令

【格式】cp　[选项]　源文件或目录　目的文件或目录