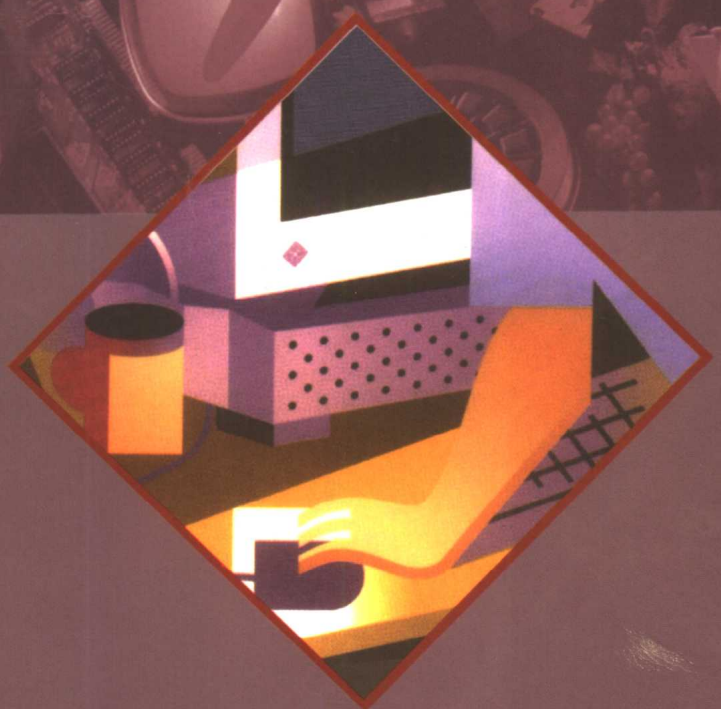
LINUX

Linux内核分析及常见问题解答

纪纯杰　贺晓能　编著　　高波　审校



人民邮电出版社
www.pptph.com.cn

Linux 内核分析及常见问题解答

纪纯杰　贺晓能　编著
高波　审校

人民邮电出版社

内 容 提 要

本书内容分为两部分。第一部分（包括第一至十一章）是 Linux 内核分析，详细地介绍了 Linux 系统的启动、进程管理、进程通信、内存管理、文件系统、设备驱动、内核监视调整以及与内核关系密切的网络系统，可以帮助读者在短时间内对 Linux 内核有一个整体上的了解；其中第七章对套接字的驱动程序 Socket.c 进行了详细的分析，这对于广大 Linux 编程人员也不无裨益。第二部分（包括第十二至二十一章）为 Linux 常见问题解答，主要分析了一些 Linux 问题的解决方法，着重于网络方面的问题，如多块网卡识别问题、设定 ppp 等。书中同时列举了大量实例，相信一定会引起广大读者的兴趣。

本书适用于 Linux 及 UNIX 的用户。

Linux 内核分析及常见问题解答

◆ 编　　著　纪纯杰　贺晓能
审　　校　高　波
责任编辑　梁　凝
◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@pptph.com.cn
网址　http://www.pptph.com.cn
北京汉魂图文设计有限公司制作
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销
◆ 开本：787 × 1092　1/16
印张：20.5
字数：502 千字　　2000 年 7 月第 1 版
印数：1 - 5 000 册　　2000 年 7 月北京第 1 次印刷
ISBN 7-115-08632-X/TP·1709

定价：31.00 元

前　言

Linux 是目前最为流行的操作系统之一，几乎每一个计算机爱好者，都能说出 Linux 是什么。Linux 的应用程序和软件包应有尽有，包括商业、统计、娱乐等诸多方面，然而对 Linux 是如何工作的，很多人都不清楚。他们可以随口说出几个术语：开放式的操作系统、公布源码等等，而实际上他们却并不清楚它到底是怎么一回事。

Linux 是什么，这是一个初学者经常错误理解的问题，其实如果直接把 Turbo Linux、Red Hat、Slackware 等说成是 Linux 是不正确的，正确地说，Linux 不是一个完整的操作系统，而是操作系统的内核，Redhat、Turbo Linux、Xteam Linux 等操作系统都是由 Linux 内核加上一些外围的应用程序组合起来的。

本书在结构上分为内核分析部分和问题解答部分。内核分析部分详细地介绍了 Linux 内核的组成，着重把 Linux 内核的原理和应用技术有机地结合起来。第二部分为常见问题解答，主要包含一些 Linux 的网络常见问题分析，还有一些 Linux 使用的技巧，都是有的放矢，针对性很强，相信能很好地帮助读者学习 Linux。

在本书的编写过程中，得到了很多朋友的热心帮助，他们是陈红、黄智、马红央、李荣阁等，他们都是 Linux 的爱好者，参加了本书的编写和大量程序测试工作，这里对他们表示深深的感谢。

由于编者水平有限，书中难免有错误和不妥之处，恳切希望读者予以指正，笔者的联系方式是 jichunjie@263.net，gaobo1128@sina.com。

编者

2000 年 4 月

目　录

第一章　Linux 内核及其引导

1.1　Linux 内核概述

Linux 内核不能孤立于系统独自运行,它必须参加整个系统的协调才能起到内核的作用。整个 Linux 操作系统的构成如图 1.1 所示。

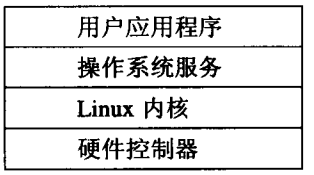


图 1.1　Linux 操作系统组成

由图 1.1 我们可以看出，整个 Linux 操作系统由四个主要的子系统组成：

(1) 用户应用程序

是直接同用户接口的程序组。基于不同的操作系统，也有一套不同的应用程序，如字处理程序、网络浏览器等等。

(2) 操作系统服务

通常被看作操作系统的一个部分，如视窗系统、Shell 等。此外，还包括同内核的程序接口，如编译工具和库等。

(3) Linux 内核

一般来讲，内核完成用户和硬件（如 CPU）的中介工作。这是本书重点讨论的对象。

(4) 硬件控制器

包括所有物理设备，如 CPU、内存、硬盘及网络设备等等。

每个子系统层只能同与它比邻的子系统层通信。另外，高层依赖于底层，而底层并不依赖于高层。

本书讨论重点在于 Linux 内核，所以，对用户应用程序、操作系统服务以及硬件控制并不做深入的探讨。有兴趣的读者可以参考有关文献。

Linux 内核在整个操作系统中，充当了一个用户进程虚拟机接口的作用。编程人员在编写应用程序时，并不需要了解系统硬件的安装情况。Linux 内核将所有的硬件抽象为一个一致的虚拟接口。此外，Linux 支持多进程。实际上，Linux 内核同时运行多个进程，并负责每个进程同硬件的接口工作。

Linux 内核主要由以下几个子系统组成：

- 进程调度（SCHED）

该子系统主要负责控制进程访问 CPU。调度管理器运行时有一套规则，以保证每个进程都有公平的机会访问 CPU，同时确保必要的实时硬件中断运行。

- 进程间通信（IPC）

IPC 子系统支持在单个 Linux 系统中多种进程间通信机制，如信号、管道和共享内存等等。

- 内存管理（MM）

允许多个进程安全地共享系统内存。此外，内存管理子系统支持虚拟内存技术，这样可以使进程使用远远大于物理内存的存储空间。

- 虚拟文件系统（VFS）

将不同的硬件设备抽象成普通的文件格式并加以管理。此外，VFS 支持多种文件格式以保证同其它操作系统的兼容。

- 网络接口（NET）

提供对多种网络标准及不同网络硬件的访问。

- 模块（Module）

Linux 模块是一种可在系统启动后的任何时候动态连入内核的代码块。当程序不再需要它时又可以将它从内核中卸载并删除。Linux 模块多指设备驱动和伪设备驱动，如网络设备和文件系统等等。

在所有 Linux 内核子系统中，最重要的是进程调度子系统。由于所有其它子系统工作的完成都需要建立进程、终止进程和恢复进程等操作，因此必须依靠进程调度子系统来予以协调。

但是，各个子系统间也是相互依赖和相互协调的。进程间通信子系统依靠内存管理子系统支持共享内存通信机制。这种机制允许两个进程访问同一块内存以完成相互的通信；虚拟文件系统使用网络接口来支持网络文件系统（NFS），同时也使用内存管理来提供 ramdisk 服务；内存管理子系统使用虚拟文件系统支持交换（swapping）技术等等。

此外，内核中所有的子系统都依赖一些共有的系统资源，包括分配和释放内存的共有例程，打印告警和错误信息的例程以及系统调试例程等等。

本书的目的是通过分析 Linux 的内核源代码使读者深入了解 Linux 操作系统的工作原理以及整个系统的体系结构，最终让读者能够自觉地分析源代码并能开发自己的源代码。所以，在以下的章节中，将尽可能地列出 Linux 的关键源代码并做简要分析，帮助读者快速地掌握 Linux 内核。

1.2 系统引导

当 PC 机加电后，80x86 处理器执行从地址 0xFFFF0 开始的程序代码，而该地址即是 ROM BIOS 的起始地址。BIOS 运行一些自检工作后初始化中断向量。之后，处理器将引导盘的第一个扇区载入地址 0x7C00 并开始执行。

对于 Linux 内核而言，最开始的部分由 8086 汇编语言写成。在源码文件中为 /arch/i386/boot/bootsect.S。当运行时，它将自己移至地址 0x90000 处，并装载后续的 2KB 代码，内核其余部分放在地址 0x10000 后。系统装载时，显示“loading...”。

以下列出 bootsect.S 的部分源代码，并做详细分析以帮助读者了解 Linux 的引导过程。内核版本为 2.1.19 版。

bootsect.S

```
.
.
.
SETUPSECS = 4
BOOTSEG    = 0x07C0
INITSEG    = DEF_INITSEG
SETUPSEG   = DEF_SETUPSEG
SYSSEG     = DEF_SYSSEG
SYSSIZE    = DEF_SYSSIZE
```

这里解释以下各个变量:

SETUPSECS

setup 扇区的默认 nr。

BOOTSEG

Boot 扇区的原始地址。

INITSEG

将 boot 移动到这里。

SETUPSEG

Setup 从这里开始。

SYSSEG

系统从这里装载。

SYSSIZE

系统大小。

```
.
.
.
mov     ax,#BOOTSEG
mov     ds,ax
mov     ax,#INITSEG
mov     es,ax
mov     cx,#256
sub     si,si
sub     di,di
cld
rep
movsw
jmpi    go,INITSEG
.
```

```
.
go:     mov     di,#0x4000-12
```

注意此处的 0x4000 是一个任意值，其值大于 bootsect+setup+stack 的总长度。

```
.
.
.
```

以下将系统装于地址 0x10000 处：

```
mov     ax,#SYSSEG
mov     es,ax               ! segment of 0x010000
call    read_it
call    kill_motor
call    print_nl
.
.
.
```

之后，控制权交给另一个汇编代码/arch/i386/boot/Setup.S。setup 识别主机系统的某些特征和 VGA。如果需要的话，它请求用户选择 video 模式。之后，它又将整个系统从地址 0x10000 移至地址 0x1000，并进入保护模式工作。

setup.S

```
INITSEG  = DEF_INITSEG
SYSSEG   = DEF_SYSSEG
SETUPSEG = DEF_SETUPSEG
DELTA_INITSEG = SETUPSEG - INITSEG
```

其中, DEF_INITSEG=0x9000; DEF_SYSSEG=0x1000; DEF_SETUPSEG=0x9020; SETUPSEG-INITSEG=0x0020。定义同上。

```
.
.
.
```

以下移动剩余的 setup 源代码及数据，由 LILO 装载四个扇区。

```
mov     di,#2048
sub     si,si
mov     ax,cs
mov     es,ax
mov     ax,#SYSSEG
mov     ds,ax
rep
movsw
.
.
```