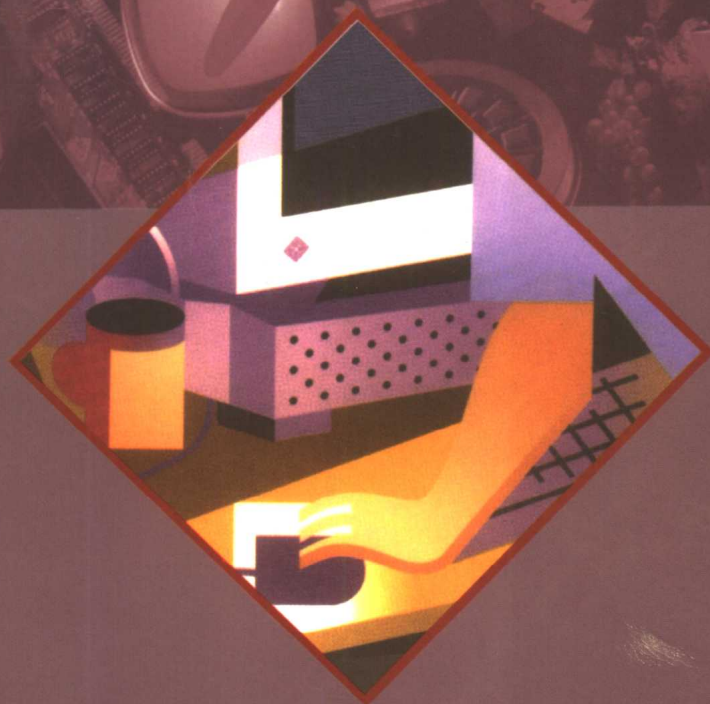


LINUX

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@poptph.com.cn
网址 <http://www.poptph.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
印数:1-5 000 册

2000 年 7 月第 1 版

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，gaobol128@sina.com。

编者

2000 年 4 月

目 录

第一章 Linux 内核及其引导	1
1.1 Linux 内核概述	1
1.2 系统引导	2
第二章 Linux 进程管理	8
2.1 概述	8
2.2 Linux 进程	8
2.3 进程系统调用	14
2.3.1 子进程的创建: fork() 系统调用	15
2.3.2 进程的并发	15
2.3.3 进程的终止: exit 系统调用	16
2.3.4 进程的同步: wait 系统调用	17
2.4 进程调度	17
2.4.1 调度原理	18
2.4.2 调度时机	18
2.4.3 调度标识的设置	18
2.4.4 调度策略与优先数的计算	18
2.4.5 调度的实现	19
2.4.6 task_struct 结构成员	19
2.4.7 调度管理器	20
2.4.8 部分源程序分析	21
2.4.9 多处理器系统中的调度	22
2.5 进程状态转换图	22
2.5.1 进程状态	22
2.5.2 进程的控制	24
2.6 软中断信号	25
2.6.1 概述	25
2.6.2 举例	25
2.6.3 软中断信号的处理步骤	26
2.7 进程虚空间描述	27
第三章 进程间通信 (IPC)	29
3.1 信号	29
3.2 管道	30

3.3 其它 IPC 机制	31
3.3.1 等待队列	31
3.3.2 文件加锁	32
3.4 UNIX 系统 V IPC 机制	33
3.4.1 消息队列	33
3.4.2 信号机	34
3.4.3 共享内存	36
3.4.4 UNIX 本地套接字	38
第四章 内存管理	39
4.1 概述	39
4.2 内存管理系统调用接口	39
4.3 虚拟内存技术	40
4.3.1 请求分页与交换	41
4.3.2 访问控制	43
4.3.3 Linux 分页表	44
4.3.4 请求换页	45
4.4 页面分配和解除分配	46
4.5 内存映射	50
4.6 高速缓存	50
4.6.1 缓冲区高速缓存	50
4.6.2 页面高速缓存	51
4.6.3 交换高速缓存	51
4.6.4 硬件高速缓存	51
第五章 文件系统	53
5.1 概述	53
5.2 虚拟文件系统(VFS)	55
5.2.1 VFS 内部工作机制	55
5.2.2 /proc 文件系统	61
5.3 EXT2 文件系统	61
5.3.1 EXT2 数据结构	63
5.3.2 EXT2 目录	67
5.3.3 数据块组描述子	67
5.3.4 EXT2 文件系统中的文件操作	68
5.4 缓存	69
5.4.1 VFS Inode 缓存	69
5.4.2 目录缓存	70
5.4.3 缓冲区缓存	70
5.5 控制台文件操作	71

5.6 模块	72
5.6.1 源代码简述	73
5.6.2 加载模块	73
5.6.3 卸载模块	75
第六章 设备驱动	76
6.1 概述	76
6.1.1 驱动程序和内核	77
6.1.2 功能及特点	78
6.2 设备驱动管理	78
6.3 设备驱动分类描述	80
6.3.1 字符设备	80
6.3.2 块设备	83
6.3.3 网络设备	86
6.4 设备驱动程序的相互调用	90
6.5 设备驱动程序的实例研究	91
6.5.1 设备假想	91
6.5.2 工作次序	92
6.5.3 实例 ramdisk.c	92
第七章 特殊设备 Socket 的设备驱动程序	99
7.1 概述	99
7.2 源程序分析	100
第八章 内核监视和系统调整	130
8.1 概述	130
8.2 监视系统状态	130
8.3 监视 CPU	131
8.4 监视内存	132
8.5 监视进程	134
8.6 监视磁盘和文件系统	136
8.7 监视网络	136
8.8 端口监视	139
8.8.1 端口监视器	139
8.8.2 配置端口监视器和服务	139
8.8.3 记账服务	140
8.8.4 进程调度	141

第九章 Shell 原理	142
9.1 Shell 原理	142
9.2 Shell 命令结构	142
9.3 Shell 控制结构	143
9.3.1 if 结构	143
9.3.2 for 结构	144
9.3.3 case 结构	144
9.4 Shell 运行环境	145
9.5 其他 Shell	146
第十章 再次讨论 Linux 的开机过程	147
10.1 开机过程	147
10.1.1 设定 LILO	147
10.1.2 加电过程	148
10.1.3 加载内核至内存	148
10.1.4 磁盘检查	148
10.1.5 单用户模式	149
10.1.6 多用户模式	149
10.2 文件配置	149
10.2.1 父进程 init	150
10.2.2 子进程的调度——inittab 文件	151
10.3 自动作业控制	155
10.3.1 系统启动时的作业控制	155
10.3.2 用户登录时的自动作业控制	160
10.4 三种作业自动控制的命令	161
10.4.1 定期重复运行作业命令 cron	161
10.4.2 特定日期运行一次的作业 (at)	163
10.4.3 系统低负荷时运行一次的作业 (batch)	163
第十一章 网络系统	164
11.1 内核源代码	164
11.2 Linux 与计算机网络	164
11.3 Linux 网络互联	166
11.3.1 TCP/IP	166
11.3.2 Socket	168
11.3.3 Socket 通信	168
11.4 IP 层	171
11.4.1 sk_buff	171
11.4.2 数据报文的传递	174

11.4.3 接收和发送 IP 包	175
11.4.4 网络地址到物理地址的映射 (ARP)	178
11.4.5 IP 路由	179
第十二章 常见问题解答——启动和用户	182
12.1 如何从主引导记录中删除 LILO 并且重建原先的 Windows MBR	182
12.2 如何在 MBR 中重建 LILO	182
12.3 用软盘来引导系统 (拷贝 LILO 到软盘)	183
12.4 如何设定安装前系统的硬件检测(CMOS)参数	184
12.5 如何设定大硬盘的 LILO	184
12.6 为什么要做 Rescue(急救)盘	185
12.7 如何在 Linux 下做一张类似 Windows 中的 dos 启动盘	185
12.8 如何制作一张 RedHat Linux 引导盘	186
12.9 如何在 login 之前执行预定进程	186
12.10 在 Linux 的非图形界面下如何增加一个用户	187
12.11 Linux 有哪些对用户操作的简单而又有效的指令	189
第十三章 常见问题和解答——文件和目录	191
13.1 如何识别文件的扩展名	191
13.2 如何用通配符指定文件	191
13.3 如何查找当前目录和改变目录	192
13.4 如何使用登录目录的缩写	193
13.5 如何建立和删除目录	193
13.6 快速进入某些目录	193
13.7 RedHat 下显示彩色目录列表	193
13.8 显示文件的类型	194
13.9 显示命令文件的路径	194
13.10 查找文件	194
13.11 删除无用的 core 文件	194
13.12 把 man 或 info 的信息存为文本文件	194
13.13 用当前路径作提示符	195
13.14 压缩可执行文件	195
13.15 查看 Linux 启动时的信息	195
13.16 处理文件名内含有特殊字符的文件	197
13.17 一次处理整个目录	197
13.18 如何防止 rm * 误删文件	197
13.19 一些特殊而实用的删除文件的方法	198
13.20 如何使用文件列表	198
13.21 如何查看文件属性	199
13.22 如何统计文件	199

13.23	如何加密文件	199
13.24	如何移动、拷贝文件和目录	200
13.25	如何比较文件和目录	201
13.26	如何在文件中实现自由查找	201
13.27	如何设定文件的权限	203
13.28	如何使用 vi 剪切、删除、粘贴文件内容	204
13.29	如何使用 cut 剪切、粘贴文件	205
13.30	如何使用 tar 和 cpio 进行文件备份和恢复	206
13.31	如何查找一个用户信息	206
13.32	如何监测是否有人在查询自己	206
13.33	如何为某些用户设定严格的 Shell	207
13.34	如何记录不成功的登录企图	207
13.35	如何为一个账户设定生存期限	207
13.36	如何实现系统账务和进行系统检查活动	208
13.37	如何终止某些用户的进程	208
13.38	如何在软盘上创建 msdos/ext2 文件系统等	209
13.39	如何使用压缩和解压缩命令	209
第十四章 常见问题解答——X Window		212
14.1	如何配置 XFree86	212
14.2	如何确定显示卡信息	218
14.3	如何定制 X Window 管理器	220
14.4	如何使 X Window 支持 AGP 显卡	222
14.5	如何启动后直接进入 X Window	222
14.6	如何后台运行 X Window 程序	222
14.7	如何强行退出 X Window	222
第十五章 常见问题解答——硬件要求及疑难解析		223
15.1	硬件要求	223
15.1.1	主板和 CPU 要求	223
15.1.2	内存要求	223
15.1.3	硬盘驱动控制的要求	223
15.1.4	硬盘空间要求	224
15.1.5	显示器以及视频适配器的要求	224
15.1.6	其它硬件	224
15.1.7	以太网卡	225
15.2	疑难解析	225
15.2.1	启动安装介质所遇到的问题	225
15.2.2	硬件问题	226
15.2.3	安装软件时遇到的问题	228

15.2.4 Linux 安装后出现的问题	229
15.3 编译内核	230
第十六章 常见问题解答——网卡	239
16.1 如何手动设置网卡	239
16.2 如何在一个 Linux 系统中安装两块网卡及如何实现网卡的自动检测	240
第十七章 常见问题解答——SLIP 和 PPP	243
17.1 串行协议和 SLIP	243
17.1.1 dip	243
17.1.2 slattach	243
17.1.3 dip 和 slip 的选择	243
17.1.4 拨号	244
17.1.5 配制	244
17.2 PPP	247
17.2.1 PPP 简介	247
17.2.2 PPP 功能	248
17.2.3 利用 Linux 系统配置 PPP	248
17.2.4 配置 MODEM 和串口	249
17.2.5 使用 root 权限设置 PPP 连接文件	250
17.2.6 PPP 服务器认证	252
17.2.7 建立 PPP 连接	253
17.2.8 常见问题及解答	253
17.2.9 使用 PPP 连接两个局域网	254
附录 17A 配置两块 NE2000 网卡心得	254
附录 17B 如何在一台 Linux 单机上拨号上网	255
第十八章 常见问题解答——TCP/IP	258
18.1 使用 TCP/IP	258
18.1.1 Internet 协议族	258
18.1.2 TCP/IP 基础	259
18.1.3 TCP/IP Internet 程序包	259
18.2 网络配置	266
18.2.1 常用命令	266
18.2.2 域名服务	268
18.3 IP Alias 技术	270
第十九章 常见问题解答——UUCP 系统	271
19.1 引言	271

19.2 UUCP 系统概述	271
19.3 UUCP 网络	272
19.3.1 网络的结构	272
19.3.2 uuname 命令的使用	272
19.4 命令	273
19.4.1 uucp 命令	273
19.4.2 cu 命令	274
19.4.3 ct 命令	277
19.4.4 UUTO 命令	277
19.4.5 uupick 命令	278
19.4.6 unstat 命令	279
19.4.7 UUX 远程执行命令	280
第二十章 常见问题解答——邮件系统	281
20.1 如何使用邮件系统	281
20.2 邮件的地址	281
20.3 阅读电子邮件	282
20.4 发送电子邮件	284
20.5 有效地使用电子邮件	285
20.6 邮件系统的管理	286
第二十一章 常见问题解答——NFS,DFS,RFS	289
21.1 NFS	289
21.1.1 引言	289
21.1.2 安装 NFS	289
21.1.3 启动 NFS	290
21.1.4 配置 NFS	291
21.1.5 NFS 服务	293
21.1.6 NFS 的安全性	293
21.1.7 如何访问远程文件	294
21.1.8 NFS 的故障检修与系统崩溃	294
21.1.9 NFS 不能做什么	295
21.2 DFS 管理	295
21.3 RFS	296
21.3.1 RFS 基础	296
21.3.2 RFS 管理	297
第二十二章 Linux 在网管系统和 MIS 系统集成方面的应用	301
22.1 概述	301
22.2 网络管理系统	301

22.2.1 基本组成	301
22.2.2 基于 SNMP 的网络管理模型	302
22.2.3 SNMP 协议结构	302
22.3 Linux 下数据库的安装	303
22.3.1 Informix 的安装	303
22.3.2 安装 oracle	311

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。

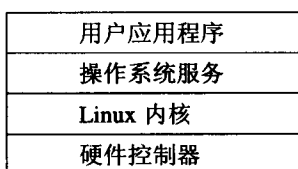


图 1.1 Linux 操作系统组成

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

jmp 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
```