

第11章 Linux内核源代码

本章讲述在Linux内核源码中，应该从何处开始查找特定的内核函数。

本书并不要求读者具有C语言编程能力，也不要求读者有一份可参阅的Linux内核源码，事实上，通过查看内核源码可以在一定深度上理解Linux操作系统，同时这也是一个很好的实践机会。本章给出了对内核源码的概览：它们是如何编排的以及从何处开始查找特定代码。

11.1 怎样得到Linux内核源码

所有主要的Linux系统(Craftworks、Debian、Slackware、Red Hat等等)都包含有内核源码，通常所安装的Linux系统都是通过这源码创建的。由于Linux总是不断更新，因此用户所安装的Linux可能已过时，不过从附录A所列的站点上可得到最新的源码，所有这些站点地址都可在<ftp://ftp.cs.helsinki.fi>上查到。

Linux内核源码的版本号表示方法非常简单：所有偶数版(如2.0.30)都是已发行的稳定版；所有奇数版(如2.1.42)都是测试版，本书是基于2.0.30版撰写的。测试版包含所有的新特征，并支持所有的新设备，虽然测试版并不稳定，并且可能提供了一些用户不想要的东西，但对于Linux与用户沟通而言，测试新的内核是很重要的。不过请注意，在尝试非产品型的测试版之前，最好先完全备份系统。

对内核源码的修改是作为patch文件出现的，patch工具提供了一组对源码文件的编辑。例如，若想把2.0.29源码升级为2.0.30版，则使用patch文件来完成对源码的编辑，操作如下：

```
$ cd /usr/src/linux
$ patch -p1 < patch-2.0.30
```

这样做可以避免对所有源码文件的拷贝。在<http://www.linuxhq.com>站点上可找到很好的内核源码的patch。

11.2 内核源码的编排

在源码目录树的最顶端(/usr/src/linux)可看到如下一些目录：

- arch arch子目录包含所有的特定体系结构的内核源码，它的子目录分别对应着一种Linux所支持的体系结构，例如i386和alpha。
- include include子目录包含大部分的编译内核源码所需文件。
- init 此目录下包含了内核的初始化代码，由此可以很好地开始了解内核是如何工作的。
- mm 此目录下包含了所有内存管理代码，特定体系结构的内存管理代码在arch/*/mm目录下。
- drivers 此目录下包含了系统所有的设备驱动程序，其下子目录各针对不同的设备驱动程序类。
- ipc 此目录下包含了内核的内部进程通信代码。
- modules 此目录只是用来保存创建的模块。

- fs 所有文件系统代码，其下子目录各针对不同的系统所支持的文件系统。
- kernel 内核主代码，特定体系结构内核代码保存在 arch/*/kernel中。
- net 内核的网络代码。
- lib 此目录包含内核库代码，特定体系结构的库代码保存在 arch/*/lib目录下。
- scripts 此目录包含了内核设置时用到的脚本。

11.3 从何处看起

像Linux这样复杂的大程序，探究起来使人迷茫，这就像一个找不出头绪的大线团。要查看内核的某一部分通常会被引向许多其他的相关文件，最后甚至忘记了最初的动机。下面给出了一些提示，根据这些提示，对于给定的内容即可找到最好的开始阅读代码部分。

1. 系统启动和初使化

在基于Intel的系统中，通常先运行loadlin.exe或LILO，由这两个程序将内核载入内存并启动内核，之后便由内核控制系统。在 arch/i386/kernel/head.s中可找到这一部分，head.s先进行一些特定体系结构的安装，然后跳转到 init/main.c中的main()例程。

2. 内存管理

有关内存管理的代码大部分都在 mm中，但与特定体系结构相关的部分则保存在 arch/*/mm中，内存缺页处理代码在 mm/memory.c中，内存映射和页缓冲代码在 mm/filemap.c中，实现缓冲区缓存部分代码在 mm/buffer.c中，页交换代码在 mm/swap_state.c和 mm/swapfile.c中。

3. 内核

大部分通用内核代码在 kernel中，与特定体系结构相关的代码在 arch/*/kernel中，调度进程代码在 kernel/sched.c中，创建子进程代码在 kernel/fork.c中，Bottom Half控制程序代码在 include/linux/interrupt.h中，task_struct数据结构的定义在 include/linux/sched.h中。

4. PCI

PCI伪驱动程序在 drivers/pci/pci.c中，系统全局定义在 include/linux/pci.h中。每种体系结构都有其特定的PCI BIOS代码，如Alpha Axp的代码在 arch/alpha/kernel/bios32.c中。

5. 内部进程间通信

所有相关代码都在 ipc中，所有的 System V IPC对象都有一个 ipc_perm数据结构，在 include/linux/ipc.h中有该数据结构的定义。System V的消息机制代码在 ipc/msg.c中，共享内存代码在 ipc/shm.c中，信号量代码在 ipc/sem.c中，管道代码在 ipc/pipe.c中。

6. 中断处理

内核的中断处理代码几乎都与特定微处理器相关。Intel的中断处理代码在 arch/i386/kernel/irq.c中，并且定义在 include/asm-i386/irq.h中。

7. 设备驱动程序

大部分的Linux内核源码行在设备驱动程序中，所有设备驱动程序代码在 drivers中，并分为如下几类：

- /block 块设备驱动程序(如ide.c)。若要了解其初始化过程，参看 drivers/block/genhd.c中的device_setup()函数，该函数不仅能初始化硬盘，也可以初始化网络。块设备包括 IDE和SCSI设备。

- /char 字符设备驱动程序，例如 ttys、串口和鼠标。
- /cdrom Linux的所有CDROM代码。在此可找到特定的CDROM设备(如Soundblaster CDROM)，请注意，ide CD驱动程序在drivers/block下的ide-cd.c中，而SCSI CD驱动程序在drivers/scsi中的scsi.c中。
- /pci PCI伪驱动程序代码，由此可了解PCI子系统是如何映射和初始化的，arch/alpha/kernel/bios32.c中的Alpha PCI固化代码也值得一看。
- /scsi 所有的SCSI代码，以及Linux所支持的所有scsi设备驱动程序代码。
- /net 所有网络设备驱动程序代码。
- /sound 所有声卡驱动程序代码。

8. 文件系统

EXT2文件系统的代码都在 fs/ext2/目录下，其数据结构定义在 include/linux/ex2_fs.h、ext2_fs_i.h和ext2_fs_sb.h中，虚文件系统(Virtual File System)数据结构在include/linux/fs.h中，代码在fs/*中，缓冲区缓存代码在fs/buffer.c中。

9. 网络

网络部分代码在 net中，其大部分的 include文件在include/net中，BSD套接字代码在net/socket.c中，IP版本4 INET套接字代码在net/ipv4/af_inet.c中，常用的协议支持代码(包括sk_buff控制例程)在net/core中，TCP/IP网络代码在net/ipv4中，而网络设备驱动程序在drivers/net中。

10. 模块

内核模块代码一部分在kernel中，另一部分在模块包中，内核代码都在kernel/mcdules.c中，其数据结构和内核守护程序 kerneld消息分别在include/linux/module.h和include/linux/kerneld.h中，ELF对象文件的结构定义在include/linux/elf.h中。