

Linux 设备文件简介

作者：[金步国](#)

版权声明

本文作者是一位开源理念的坚定支持者，所以本文虽然不是软件，但是遵照开源的精神发布。

- 无担保：本文作者不保证作品内容准确无误，亦不承担任何由于使用此文档所导致的损失。
- 自由使用：任何人都可以自由的[阅读/链接/打印](#)此文档，无需任何附加条件。
- 名誉权：任何人都可以自由的[转载/引用/再创作](#)此文档，但必须保留作者署名并注明出处。

其他作品

本文作者十分愿意与他人分享劳动成果，如果你对我的其他翻译作品或者技术文章有兴趣，可以在如下位置查看现有的作品集：

- [金步国作品集](http://www.jinbuguo.com/) [<http://www.jinbuguo.com/>]

联系方式

由于作者水平有限，因此不能保证作品内容准确无误。如果你发现了作品中的错误(哪怕是错别字也好)，请来信指出，任何提高作品质量的建议我都将虚心接纳。

- Email(QQ)：70171448在QQ邮箱

概述

设备管理是 Linux 中比较基础的知识，与内核的关系也比较密切。随着 Udev 的广泛使用，Linux 发行版的智能程度越来越高，许多 Linux 新用户对于 /dev 目录下的东西变得不再熟悉，有时候遇到问题就会抓狂。

Linux 中的设备按照存取方式的不同，可以分为两种：

字符设备

无缓冲且只能顺序存取

块设备

有缓冲且可以随机(乱序)存取

而按照是否对应物理实体，也可以分为两种：

物理设备

对实际存在的物理硬件的抽象

虚拟设备

不依赖于特定的物理硬件，仅是内核自身提供的某种功能

无论是哪种设备，在 /dev 目录下都有一个对应的文件(节点)，并且每个设备文件都必须有主/次设备号，主设备号相同的设备是同类设备，使用同一个驱动程序(虽然目前的内核允许多个驱动共享一个主设备号，但绝大多数设备依然遵循一个驱动对应一个主设备号的原则)。可以通过 `cat /proc/devices` 命令查看当前已经加载的设备驱动程序的主设备号。

内核能够识别的所有设备都记录在源码树下的 [Documentation/devices.txt](#) 文件中。另外，在 <http://www.lanana.org/docs/device-list/> 也有一份，貌似已经失去维护。

注意：在 /dev 目录下除了各种设备节点之外还通常还会存在：FIFO管道、Socket、软/硬连接、目录。这些东西并不是设备文件，因此也就没有主/次设备号。

设备文件

了解这些设备的最基本要求就是对每个设备文件的含义了如指掌，下面列出了 Linux-3.13.2 内核中常见的已注册设备及其含义(省略了生僻与罕见的设备)。

主设备号	设备类型
次设备号=文件名	简要说明

0 未命名设备(例如NFS之类非设备的挂载)
0 = 为空设备号保留

参见主设备号为144,145,146的块设备, 以了解"扩展区域"(expansion area)

1 char 内存设备

1 = /dev/mem 物理内存的全镜像。可以用来直接存取物理内存。
2 = /dev/kmem 内核看到的虚拟内存的全镜像。可以用来访问内核中的内容(查看内核变量或用作rootkit之类)。
3 = /dev/null 空设备。任何写入都将被直接丢弃(但返回"成功");任何读取都将得到EOF(文件结束标志)。
4 = /dev/port 存取I/O端口
5 = /dev/zero 零流源。任何写入都将被直接丢弃(但返回"成功");任何读取都将得到无限多的二进制零流。
7 = /dev/full 满设备。任何写入都将失败,并把errno设为ENOSPC(没有剩余空间);任何读取都将得到无限多的二进制零流。
这个设备通常被用来测试程序在遇到磁盘无剩余空间错误时的行为。
8 = /dev/random 真随机数发生器。以背景噪声数据或硬件随机数发生器作为熵池,读取时会返回小于熵池噪声总数的随机字节。
若熵池空了,读操作将会被阻塞,直到收集到了足够的环境噪声为止。建议用于需要生成高强度密钥的场合。
[注意]虽然允许写入,但企图通过写入此文件来"预存"随机数是徒劳的,因为写入的数据对输出并无影响。
9 = /dev/urandom 伪随机数发生器。更快,但是不够安全。仅用于对安全性要求不高的场合。
即使熵池空了,读操作也不会被阻塞,而是把已经产生的随机数做为种子来产生新的随机数。
[注意]虽然允许写入,但企图通过写入此文件来"预存"随机数是徒劳的,因为写入的数据对输出并无影响。
10 = /dev/aio 异步I/O通知接口
11 = /dev/kmsg 任何对该文件的写入都将作为printk的输出;而读取则得到printk的输出缓冲区内容。

1 block RAM disk [已过时,请用TMPFS]

0 = /dev/ram0 第1个 RAM disk
1 = /dev/ram1 第2个 RAM disk
...

[说明]将/dev/ram0用作[initrd](#)的做法已过时(因为它仅针对image-initrd格式),当下的主流是cpio-initrd格式。

4 char TTY(终端)设备

0 = /dev/tty0 当前虚拟控制台
1 = /dev/tty1 第1个虚拟控制台
...
63 = /dev/tty63 第63个虚拟控制台
64 = /dev/ttyS0 第1个UART串口
...
255 = /dev/ttyS191 第192个UART串口

[说明]"UART串口"是指 8250/16450/16550 UART串行控制芯片

4 block 如果根文件系统是以只读方式挂载的,那么就不可能创建真正的设备节点,此时就使用该设备作为动态分配的主设备的别名,并挂载为根文件系统。
0 = /dev/root

5 char 辅助 TTY 设备

0 = /dev/tty 当前 TTY 设备
1 = /dev/console 系统控制台(一般是/dev/tty0)
2 = /dev/ptmx 所有 Unix98 PTY master 的复用器
3 = /dev/ttyprintk 内核通过此设备使用printk发送内嵌的用户消息(依赖于CONFIG_TTY_PRINTK)
64 = /dev/cua0 对应于 ttyS0 的呼出(Callout)设备
...
255 = /dev/cua191 对应于 ttyS191 的呼出(Callout)设备

7 char 虚拟控制台捕捉设备(这些设备既允许读也允许写)

0 = /dev/vcs 当前虚拟控制台(vc)的文本内容
1 = /dev/vcs1 tty1 的文本内容
...
63 = /dev/vcs63 tty63 的文本内容
128 = /dev/vcsa 当前虚拟控制台(vc)的文本/属性内容
129 = /dev/vcsa1 tty1 的文本/属性内容
...
191 = /dev/vcsa63 tty63 的文本/属性内容

7 block 回环设备(用一个普通的文件来模拟一个块设备)

0 = /dev/loop0 第1个回环设备
1 = /dev/loop1 第2个回环设备
...

[提示]对回环设备的绑定由 mount(8) 或 [losetup](#)(8) 处理

8 block SCSI 磁盘(0-15)

0 = /dev/sda 第1个 SCSI 磁盘(整个磁盘)
16 = /dev/sdb 第2个 SCSI 磁盘(整个磁盘)
32 = /dev/sdc 第3个 SCSI 磁盘(整个磁盘)
...
240 = /dev/sdp 第16个 SCSI 磁盘(整个磁盘)

分区表示方法如下(以第3个 SCSI 磁盘为例)

33 = /dev/sdc1 第1个分区
34 = /dev/sdc2 第2个分区
...
47 = /dev/sdc15 第15个分区

[提示]对于DOS分区来说, 1-4是主分区, 5-15是逻辑分区。但对于GPT分区来说, 全是主分区, 没有逻辑分区的概念。

[提示]目前的内核将SATA/PATA/IED硬盘统一使用 /dev/sd* 来表示, 已经不再使用 /dev/hd* 这种过时的设备文件了。

9 block Metadisk(RAID)设备

0 = /dev/md0 第1组 metadisk
1 = /dev/md1 第2组 metadisk
...

[说明]MD驱动(CONFIG_BLK_DEV_MD)的作用是将同一个文件系统分割到多个物理磁盘上。

10 char 各种杂项设备(含非串口鼠标)

1 = /dev/psaux PS/2鼠标
128 = /dev/beep 能够让主板的蜂鸣器发出不同频率声音的设备([Fancy Beeper Daemon](#))
130 = /dev/watchdog 看门狗(CONFIG_WATCHDOG)
131 = /dev/temperature 机器内部温度
135 = /dev/rtc 实时时钟(Real Time Clock)
143 = /dev/pciconf PCI配置空间
144 = /dev/nvram 非易失配置RAM
151 = /dev/led 发光二极管(LED)灯
152 = /dev/kpoll 内核轮询(Poll)驱动
156 = /dev/lcd 液晶(LCD)显示屏
161 = /dev/userdma 用户空间DMA访问
162 = /dev/smbus 系统管理总线(System Management Bus)
164 = /dev/ipmo Intel的智能平台管理(Intelligent Platform Management)接口
165 = /dev/vmmon VMware虚拟机监视器
170 = /dev/thinkpad/thinkpad Thinkpad设备
173 = /dev/ipmikcs 智能平台管理(Intelligent Platform Management)接口
175 = /dev/agpgart AGP图形地址重映射表(Graphics Address Remapping Table)
182 = /dev/perfctr 性能监视计数器
183 = /dev/hwrng 通用硬件随机数发生器
184 = /dev/cpu/microcode CPU微代码更新接口(依赖于CONFIG_MICROCODE)
186 = /dev/atomicps 进程状态数据的原子快照
188 = /dev/smbusbios SMBus(系统管理总线) BIOS
189 = /dev/ussp_ctl 用户空间串口控制器
200 = /dev/net/tun TAP/TUN 网络设备(TAP/TUN以软件的方式实现了网络设备)
TAP模拟了以太网帧(第二层), TUN模拟了IP包(第三层)。
202 = /dev/emd/ctl 增强型 Metadisk RAID (EMD) 控制器
203 = /dev/cuse 用户空间的字符设备(Character device in user-space)
212 = /dev/watchdogs/0 第一只看门狗
213 = /dev/watchdogs/1 第二只看门狗
214 = /dev/watchdogs/2 第三只看门狗
215 = /dev/watchdogs/3 第四只看门狗
220 = /dev/mptctl Message passing technology (MPT) control

```

223 = /dev/input/uinput      用户层输入设备
224 = /dev/tpm              TCPA TPM driver
227 = /dev/mcelog           X86_64 Machine Check Exception driver
228 = /dev/hpet             高精度事件定时器(HPET)
229 = /dev/fuse             Fuse(用户空间的虚拟文件系统)
231 = /dev/snapshot         系统内存快照
232 = /dev/kvm              内核虚拟机(基于AMD SVM和Intel VT硬件虚拟技术)
234 = /dev/btrfs-control     Btrfs文件系统控制设备
235 = /dev/autofs           Autofs控制设备
236 = /dev/mapper/control    设备映射(Device-Mapper)控制器
237 = /dev/loop-control      回环设备控制器
238 = /dev/vhost-net        用于 virtio net 的宿主内核加速器

```

11 block SCSI CD-ROM 设备

```

0 = /dev/scd0      第1个 SCSI CD-ROM
1 = /dev/scd1      第2个 SCSI CD-ROM
...

```

13 char 核心输入设备

```

0 = /dev/input/js0  第一个游戏杆(joystick)
1 = /dev/input/js1  第二个游戏杆(joystick)
...
32 = /dev/input/mouse0      第1个鼠标
33 = /dev/input/mouse1      第2个鼠标
...
63 = /dev/input/mice  所有鼠标的合体
64 = /dev/input/event0      第1个事件队列
65 = /dev/input/event1      第2个事件队列
...

```

14 char OSS(Open Sound System)兼容设备[已过时]

```

0 = /dev/mixer      OSS混音控制器(CONFIG_SND_MIXER_OSS)
1 = /dev/sequencer  OSS音序器(CONFIG_SND_SEQUENCER_OSS)
3 = /dev/dsp        OSS数字音频波形输出设备(CONFIG_SND_PCM_OSS)
8 = /dev/sequencer2 第二OSS音序器(CONFIG_SND_SEQUENCER_OSS)
16 = /dev/mixer1     第二OSS混音控制器(CONFIG_SND_MIXER_OSS)
19 = /dev/dsp1       第二OSS数字音频波形输出设备(CONFIG_SND_PCM_OSS)

```

21 char 通用 SCSI 设备(通常是SCSI光驱)

```

0 = /dev/sg0      第1个通用 SCSI 设备
1 = /dev/sg1      第2个通用 SCSI 设备
...

```

[说明]许多发行版还将这些设备命名为 /dev/sga, /dev/sgb, /dev/sgc ...

29 char 通用帧缓冲(frame buffer)设备

```

0 = /dev/fb0      第1个帧缓冲设备
1 = /dev/fb1      第2个帧缓冲设备
...
31 = /dev/fb31     第32个帧缓冲设备

```

43 block 网络块设备(Network block devices)

```

0 = /dev/nb0      第1个网络块设备
1 = /dev/nb1      第2个网络块设备
...

```

44 block 闪存转换层(Flash Translation Layer)文件系统

```

0 = /dev/ftla      第1个MTD(Memory Technology Device)上的FTL
16 = /dev/ftlb     第2个MTD(Memory Technology Device)上的FTL
32 = /dev/ftlc     第3个MTD(Memory Technology Device)上的FTL
...
240 = /dev/ftlp    第16个MTD(Memory Technology Device)上的FTL

```

[说明]分区的表示方法与SCSI磁盘相同(最大15个)

65 block SCSI 磁盘(16-31)

0 = /dev/sdq	第17个 SCSI 磁盘(整个磁盘)
16 = /dev/sdr	第18个 SCSI 磁盘(整个磁盘)
32 = /dev/sds	第19个 SCSI 磁盘(整个磁盘)
...	
240 = /dev/sdaf	第32个 SCSI 磁盘(整个磁盘)

66 block SCSI 磁盘(32-47)

0 = /dev/sdag	第33个 SCSI 磁盘(整个磁盘)
16 = /dev/sdah	第34个 SCSI 磁盘(整个磁盘)
32 = /dev/sdai	第35个 SCSI 磁盘(整个磁盘)
...	
240 = /dev/sdav	第48个 SCSI 磁盘(整个磁盘)

67 block SCSI 磁盘(48-63)

0 = /dev/sdaw	第49个 SCSI 磁盘(整个磁盘)
16 = /dev/sdax	第50个 SCSI 磁盘(整个磁盘)
32 = /dev/sday	第51个 SCSI 磁盘(整个磁盘)
...	
240 = /dev/sdbl	第64个 SCSI 磁盘(整个磁盘)

81 char video4linux

0 = /dev/video0	第1个视频采集设备(摄像头)
...	
63 = /dev/video63	第64个视频采集设备(摄像头)
64 = /dev/radio0	第1个无线电设备(收音机之类)
...	
127 = /dev/radio63	第64个无线电设备(收音机之类)
224 = /dev/vbi0	第1个垂直中断(vertical blank interrupt)
...	
255 = /dev/vbi31	第64个垂直中断(vertical blank interrupt)

89 char I2C 总线接口

0 = /dev/i2c-0	第1个 I2C 适配器
1 = /dev/i2c-1	第2个 I2C 适配器
...	

90 char 内存技术设备(Memory Technology Device) (RAM, ROM, Flash)

0 = /dev/mtd0	第1个 MTD (读写)
1 = /dev/mtdr0	第1个 MTD (只读)
...	
30 = /dev/mtd15	第16个 MTD (读写)
31 = /dev/mtdr15	第16个 MTD (只读)

93 block NAND闪存转换层(Flash Translation Layer)文件系统

0 = /dev/nftla	第1个NFTL层
16 = /dev/nftlb	第2个NFTL层
...	
240 = /dev/nftlp	第16个NFTL层

[说明]分区的表示方法与SCSI磁盘相同(最大15个)

98 block 用户模式下的虚拟块设备(分区处理方式与 SCSI 磁盘相同)

0 = /dev/ubda	第1个用户模式块设备
16 = /dev/udbb	第2个用户模式块设备
...	

108 char 独立于特定设备的 PPP 接口

0 = /dev/ppp	独立于特定设备的 PPP 接口(CONFIG_PPP)
--------------	-----------------------------

117 block 企业卷管理系统(Enterprise Volume Management System)

0 = /dev/evms/block_device EVMS块设备
 1 = /dev/evms/legacyname1 第1个EVMS传统设备
 2 = /dev/evms/legacyname2 第2个EVMS传统设备
 ...

两个范围都可以相互延伸(向上/向下)一直到它们相遇

...

254 = /dev/evms/EVMSname2 第2个EVMS本地设备
 255 = /dev/evms/EVMSname1 第1个EVMS本地设备

[说明]"legacyname"来源于普通的块设备名,

例如 /dev/sda5 将会变成 /dev/evms/sda5

119 char VMware虚拟网路控制器

0 = /dev/vnet0 第1个虚拟网路
 1 = /dev/vnet1 第2个虚拟网路
 ...

128-135 char Unix98 PTY master

这些设备不应当存在设备节点, 而应当通过 /dev/ptmx 接口访问。

136-143 char Unix98 PTY slave

0 = /dev/pts/0 第1个 Unix98 PTY slave
 1 = /dev/pts/1 第2个 Unix98 PTY slave
 ...

这些设备节点是自动生成的(伴有适当的权限和模式), 不能手动创建。

方法是通过使用适当的 mount 选项(通常是: mode=0620,gid=<"tty"组的gid>)

将 devpts 文件系统挂载到 /dev/pts 目录即可。

144 block 用于更多非设备型挂载的扩展区域(Expansion Area)#1

0 = mounted device 256
 255 = mounted device 511

145 block 用于更多非设备型挂载的扩展区域(Expansion Area)#2

0 = mounted device 512
 255 = mounted device 767

146 block 用于更多非设备型挂载的扩展区域(Expansion Area)#3

0 = mounted device 768
 255 = mounted device 1023

147 block DRBD(Distributed Replicated Block Device)

0 = /dev/drbd0 第1个 DRBD 设备
 1 = /dev/drbd1 第2个 DRBD 设备
 ...

153 block EMD(Enhanced Metadisk RAID)存储单元(分区处理方式与 SCSI 磁盘相同)

0 = /dev/emd/0 第1个存储单元
 1 = /dev/emd/0p1 第1个存储单元的第1个分区
 2 = /dev/emd/0p2 第1个存储单元的第2个分区
 ...
 15 = /dev/emd/0p15 第1个存储单元的第15个分区
 16 = /dev/emd/1 第2个存储单元
 32 = /dev/emd/2 第3个存储单元
 ...
 240 = /dev/emd/15 第16个存储单元

179 block MMC(MultiMediaCard)块设备

0 = /dev/mmcblk0 第1块 SD/MMC 卡
 1 = /dev/mmcblk0p1 第1块 SD/MMC 卡的第1个分区

8 = /dev/mmcblk1 第2块 SD/MMC 卡
...

[说明]每个MMC块设备保留的次设备号数量由 CONFIG_MMC_BLOCK_MINORS 决定(默认=8)。
或者也可以通过内核引导参数 mmcblk.perdev_minors 进行修改。

180 char USB字符设备
0 = /dev/usb/lp0 第1个USB打印机
...
15 = /dev/usb/lp15 第16个USB打印机
48 = /dev/usb/scanner0 第1个USB扫描仪
...
63 = /dev/usb/scanner15 第16个USB扫描仪
96 = /dev/usb/lp0 第1个USB打印机
...
111 = /dev/usb/lp15 第16个USB打印机
132 = /dev/usb/lp0 第1个USB打印机

180 block USB块设备
0 = /dev/uba 第1个USB块设备
8 = /dev/ubb 第2个USB块设备
16 = /dev/ubc 第3个USB块设备
...

192 char 内核 profiling 接口
0 = /dev/profile Profiling 控制设备
1 = /dev/profile0 CPU 0 的 Profiling 设备
2 = /dev/profile1 CPU 1 的 Profiling 设备
...

193 char 内核事件跟踪接口
0 = /dev/trace 跟踪控制设备
1 = /dev/trace0 CPU 0 的跟踪设备
2 = /dev/trace1 CPU 1 的跟踪设备
...

195 char Nvidia 图形设备(比如显卡)
0 = /dev/nvidia0 第1个 Nvidia 卡
1 = /dev/nvidia1 第2个 Nvidia 卡
...
255 = /dev/nvidia0 第1个 Nvidia 卡

202 char 特定于CPU模式的寄存器(model-specific register,MSR)
0 = /dev/cpu/0/msr CPU 0 的 MSRs
1 = /dev/cpu/1/msr CPU 1 的 MSRs
...

202 block Xen 虚拟块设备
0 = /dev/xvda 第1个 Xen 虚拟磁盘(整块磁盘)
16 = /dev/xvdb 第2个 Xen 虚拟磁盘(整块磁盘)
32 = /dev/xvdc 第3个 Xen 虚拟磁盘(整块磁盘)
...
240 = /dev/xvdp 第16个 Xen 虚拟磁盘(整块磁盘)

[说明]分区的表示方法与SCSI磁盘相同(最大15个)

203 char CPU CPUID 信息
0 = /dev/cpu/0/cpuid CPU0的CPUID
1 = /dev/cpu/1/cpuid CPU1的CPUID
...

226 char DRI(Direct Rendering Infrastructure)

```
0 = /dev/dri/card0    第1个显卡
1 = /dev/dri/card1    第2个显卡
...
```

```
232 char    生物识别设备
0 = /dev/biometric/sensor0/fingerprint  第1个设备的第1个指纹传感器
1 = /dev/biometric/sensor0/iris         第1个设备的第1个虹膜传感器
2 = /dev/biometric/sensor0/retina       第1个设备的第1个视网膜传感器
3 = /dev/biometric/sensor0/voiceprint   第1个设备的第1个声波传感器
4 = /dev/biometric/sensor0/facial       第1个设备的第1个面部传感器
5 = /dev/biometric/sensor0/hand         第1个设备的第1个手掌传感器
...
10 = /dev/biometric/sensor1/fingerprint  第2个设备的第1个指纹传感器
...
20 = /dev/biometric/sensor2/fingerprint  第3个设备的第1个指纹传感器
...
```

/dev 目录中的其他内容

这部分详细说明一些应该或可能存在于 /dev 目录中的其他文件。链接最好使用与这里完全相同的格式(绝对路径或相对路径)。究竟是使用硬链接(hard)还是软连接(symbolic)取决于不同的设备，但最好与这里给出示范保持一致。

必须的链接

必须在所有的系统上都存在这些连接：

链接	目标	链接类型	简要说明
/dev/fd	/proc/self/fd	symbolic	文件描述符
/dev/stdin	fd/0	symbolic	stdin(标准输入)文件描述符
/dev/stdout	fd/1	symbolic	stdout(标准输出)文件描述符
/dev/stderr	fd/2	symbolic	stderr(标准错误)文件描述符
/dev/nfsd	socksys	symbolic	仅为 iBCS-2 所必须
/dev/X0R	null	symbolic	仅为 iBCS-2 所必须

推荐的链接

推荐在所有的系统上都存在这些连接：

链接	目标	链接类型	简要说明
/dev/core	/proc/kcore	symbolic	为了向后兼容
/dev/ramdisk	ram0	symbolic	为了向后兼容
/dev/ftape	qft0	symbolic	为了向后兼容
/dev/bttv0	video0	symbolic	为了向后兼容
/dev/radio	radio0	symbolic	为了向后兼容
/dev/i2o*	/dev/i2o/*	symbolic	为了向后兼容
/dev/scd?	sr?	hard	代替 SCSI CD-ROM 的名字

本地定义的链接

下面的链接很可能需要根据机器的实际硬件配置创建其中的一部分甚至全部。这些链接仅仅是为了迎合习惯用法，它们既非必须也非推荐。

链接	目标	链接类型	简要说明
/dev/mouse	鼠标设备	symbolic	当前鼠标设备
/dev/tape	磁带设备	symbolic	当前磁带设备
/dev/cdrom	光盘设备	symbolic	当前光盘设备
/dev/cdwriter	刻录机设备	symbolic	当前刻录机设备
/dev/scanner	扫描仪设备	symbolic	当前扫描仪设备
/dev/modem	调制解调器(猫)	symbolic	当前拨号设备
/dev/root	根文件系统设备	symbolic	当前根文件系统设备
/dev/swap	swap设备	symbolic	当前swap设备

/dev/modem 不应当用于能够同时支持接入(dialin)和呼出(dialout)的猫，因为往往会导致锁文件问题。如果存在 /dev/modem ，那么它应当指向一个恰当的主 TTY 设备。

对于SCSI设备, /dev/tape 应该指向 /dev/st* , 而 /dev/cdrom 应该指向 /dev/sr* ; 而 /dev/cdwriter 和 /dev/scanner 应当分别指向对应的 /dev/sg* 。

/dev/mouse 可以指向一个主串行 TTY 设备、一个硬件鼠标、或者一个对应鼠标驱动程序的套接字(例如 /dev/gpmdata)。

套接字和管道

持久套接字和命名管道可以存在于 /dev 中。常见的有：

/dev/printer	socket	lpd 本地套接字
/dev/log	socket	syslog 本地套接字
/dev/gpmdata	socket	gpm 鼠标多路复用器(multiplexer)
/dev/initctl	fifo pipe	init 监听它并从中获取信息(用户与 init 进程交互的通道)

挂载点

以下目录被保留用于挂载特殊的文件系统。这些特殊的文件系统只提供内核接口而不提供标准的设备节点。

/dev/pts	devpts	PTY slave 文件系统
/dev/shm	tmpfs	提供对 POSIX 共享内存的直接访问

终端设备

[终端\(或TTY\)](#)设备是一种特殊的字符设备。终端设备是可以在会话中用作控制终端的任何设备, 包括: 虚拟控制台、串行接口、伪终端(PTY)。

所有终端设备共享一个通用的功能集合(线路规则), 这包含常规的终端线路规程以及SLIP和PPP模式。所有的终端设备的命名都很相似。这部分内容将解释命名规则和各种类型的TTY(终端)的使用。需要注意的是这些命名习惯包含了几个历史遗留包袱。其中的一些是Linux所特有的, 另一些则是继承自其他系统, 还有一些反映了Linux在成长过程中抛弃了原来借用自其它系统的一些习惯。井号(#)在设备名里表示一个无前导零的十进制数。

虚拟控制台和控制台设备

虚拟控制台是在系统视频监视器上显示的全屏终端。虚拟控制台被命名为 /dev/tty# (编号从 /dev/tty1 开始)。
/dev/tty0 是当前虚拟控制台。
/dev/tty0 用于在不能使用帧缓冲设备(/dev/fb*)的机器上存取系统显卡, 但 /dev/console 并不用于此目的。
控制台设备(/dev/console)由内核直接管理, 用于接收和显示系统消息, 以及单用户模式登陆。

串行接口

这里所说的"串行接口"是指 RS-232 串口和任何模拟这种接口的设备, 无论是硬件(如调制解调器)还是软件(如ISDN驱动)。Linux中的每一个串口都有两个设备名: 主设备或呼入(callin)设备、辅设备或呼出(callout)设备。两者之间使用字母的大小写进行区分。比如, 对于任意字母"X", 设备名分别为 /dev/ttyX# 与 /dev/cux# 。由于历史原因, /dev/ttyS# 和 /dev/ttyC# 分别等价于 /dev/cua# 和 /dev/cub# 。名称 /dev/ttyQ# 和 /dev/cuq# 被保留为仅供本地使用。

串口的仲裁是通过锁文件(/var/lock/LCK..ttyX#)来提供的。锁文件的内容应该是以ASCII码表示的锁定进程的PID。常见的做法是安装一个诸如 /dev/modem 这样的链接来指向串口。为了确保能够正确的预先锁定这些链接, 软件应该追踪符号链接并锁定所有可能的名字。此外, 还建议为相应的辅设备安装对应的锁文件。为了避免死锁, 建议按以下顺序获取锁, 并按相反的顺序释放锁:

1. 符号链接名, 如果有(/var/lock/LCK..modem)
2. "tty"名(/var/lock/LCK..ttyS2)
3. 辅设备名(/var/lock/LCK..cua2)

在符号链接出现嵌套的情况下, 锁文件应按照符号链接的解析顺序来安装。

在任何情况下, 应用程序都应该等待另一个程序释放锁之后, 再持有这个锁。此外, 试图为辅设备创建锁文件的应用程序应考虑被用于非串口的TTY端口的可能性(此时不存在辅设备)。

伪终端(PTY)

伪终端既可以用于创建登陆会话, 也可以为其他需要通过TTY线路规则(包括SLIP或者PPP功能)来生成数据的进程提供帮助。每一个 PTY 都有一个master端和一个slave端。按照 System V/Unix98 的 PTY 命名方案, 所有master端共享同一个 /dev/ptmx 设备节点(打开它内核将自动给出一个未分配的PTY), 所有slave端都位于 /dev/pts/ 目录下, 名为 /dev/pts/# (内核会根据需要自动生成和删除它们)。

一旦master端被打开, 相应的slave设备就可以按照与 TTY 设备完全相同的方式使用。master设备与slave设备之间通过内核进行连接, 等价于拥有 TTY 功能的双向管道(pipe)。