

Linux 设备文件简介

作者：[金步国](#)

版权声明

本文作者是一位自由软件爱好者，所以本文虽然不是软件，但是本着 GPL 的精神发布。任何人都可以自由使用、转载、复制和再分发，但必须保留作者署名，亦不得对声明中的任何条款作任何形式的修改，也不得附加任何其它条件。您可以自由链接、下载、传播此文档，但前提是必须保证全文完整转载，包括完整的版权信息和作译者声明。

其他作品

本文作者十分愿意与他人共享劳动成果，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作品的列表：

- [金步国作品列表](#)

BUG 报告，切磋与探讨

由于作者水平有限，因此不能保证作品内容准确无误，请在阅读中自行鉴别。如果你发现了作品中的错误，请您来信指出，哪怕是错别字也好，任何提高作品 质量的建议我都将虚心接纳。如果你愿意就作品中的相关内容与我进行进一步切磋与探讨，也欢迎你与我联系。联系方式：Email: csfrank@citiz.net ； QQ : 70171448 ； MSN : csfrank122@hotmail.com

概述

设备管理是 linux 中比较基础的东西，但是由于 Linux 智能程度的越来越高，Udev 的使用越来越广泛，使得越来越多的 Linux 新用户对于 /dev 目录下的东西变得不再熟悉。有时候遇见问题就会变得抓狂。

Linux 中的设备有 2 种类型：字符设备(无缓冲且只能顺序存取)、块设备(有缓冲且可以随机存取)。每个字符设备和块设备都必须有主、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)。这些设备中，有些设备是对实际存在的物理硬件的抽象，而有些设备则是内核自身提供的功能(不依赖于特定的物理硬件， 又称为“虚拟设备”)。每个设备在 /dev 目录下都有一个对应的文件(节点)。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原码树下的 Documentation/devices.txt 文件中。在 /dev 目录下除了字符设备和块设备节点之外还通常还会存在：FIFO 管道、Socket、软/硬连接、目录。这些东西没有主/次设备号。

设备文件

Linux 内核所能识别的所有设备都记录在 <http://www.lanana.org/docs/device-list/> 而内核原码树中的 Documentation/devices.txt 可能不是最新版本。

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，下面列出常见的设备文件以及相应的含义(比较偏僻的就省略了)。

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：挂载的非设备) 0 = 为空设备号保留	
1	char内存设备	
1	1 = /dev/mem	直接存取物理内存
2	2 = /dev/kmem	存取经过内核虚拟之后的内存
3	3 = /dev/null	空设备。任何写入都将被直接丢弃，任何读取都将得到 EOF。
4	4 = /dev/port	存取 I/O 端口
5	5 = /dev/zero	零字节源，只能读取到无限多的零字节。
7	7 = /dev/full	满设备。任何写入都将失败，并把 errno 设为 ENOSPC 以表示没有剩余空间。
8	8 = /dev/random	随机数发生器。完全由用户的输入来产生随机数。 如果用户停止所有动作，则停止产生新的随机数。
9	9 = /dev/urandom	更快，但是不够安全的随机数发生器。尽可能由用户的输入来产生随机数， 如果用户停止所有动作，则把已经产生的随机数做为种子来产生新的随机数。
10	10 = /dev/aio	异步 I/O 通知接口
11	11 = /dev/kmsg	任何对该文件的写入都将作为 printk 的输出
1	block RAM disk	
0	0 = /dev/ram0	第 1 个 RAM disk (initrd 只能使用 ram0)
1	1 = /dev/ram1	第 2 个 RAM disk

...
200 = /dev/ram200 第 200 个 RAM disk

4 charTTY(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

4 block 如果根文件系统以只读方式挂载的，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(major)设备的别名
0 = /dev/root

5 char其他 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 PTY master 的复用器

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(用一个普通的磁盘文件来模拟一个块设备)
对回环设备的绑定由 mount(8) 或 losetup(8) 处理
0 = /dev/loop0 第 1 个回环设备
1 = /dev/loop1 第 2 个回环设备
...

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
对于 Linux/i386 来说，分区 1-4 是主分区，5-15 是逻辑分区。

9 block Metadisk(RAID)设备
0 = /dev/md0 第 1 组 metadisk
1 = /dev/md1 第 2 组 metadisk
...
metadisk 驱动用于将同一个文件系统分割到多个物理磁盘上。

10 char 非串口鼠标，各种杂项设备和特性
1 = /dev/psaux PS/2 鼠标
131 = /dev/temperature 机器内部温度
134 = /dev/apm_bios APM(高级电源管理) BIOS
135 = /dev/rtc 实时时钟(Real Time Clock)
144 = /dev/nvram 非易失配置 RAM
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
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/hwrmg 通用硬件随机数发生器
184 = /dev/cpu/microcode CPU 微代码更新接口
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
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
220 = /dev/mptctl Message passing technology (MPT) control
223 = /dev/input/uinput 用户层输入设备驱动支持
227 = /dev/mcelog X86_64 Machine Check Exception driver
228 = /dev/hpet HPET driver

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 基于内核的虚构机(基于 AMD SVM 和 Intel VT 硬件虚拟技术)

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
 32 = /dev/input/mouse0 第 1 个鼠标
 33 = /dev/input/mouse1 第 2 个鼠标
 ...
 62 = /dev/input/mouse30 第 31 个鼠标
 63 = /dev/input/mice 所有鼠标的统一
 64 = /dev/input/event0 第 1 个事件队列
 65 = /dev/input/event1 第 2 个事件队列
 ...
 95 = /dev/input/event1 第 32 个事件队列

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
 ...

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

30 char iBCS-2 兼容设备
 0 = /dev/socksys 套接字访问接口
 1 = /dev/spx SVR3 本地 X 接口
 32 = /dev/inet/ip 网络访问接口
 33 = /dev/inet/icmp
 34 = /dev/inet/ggp
 35 = /dev/inet/ipip
 36 = /dev/inet/tcp
 37 = /dev/inet/egp
 38 = /dev/inet/pup
 39 = /dev/inet/udp
 40 = /dev/inet/idp
 41 = /dev/inet/rawip

此外, iBCS-2 还需要下面的连接必须存在
/dev/ip -> /dev/inet/ip
/dev/icmp -> /dev/inet/icmp
/dev/ggp -> /dev/inet/ggp
/dev/ipip -> /dev/inet/ipip
/dev/tcp -> /dev/inet/tcp
/dev/egp -> /dev/inet/egp
/dev/pup -> /dev/inet/pup
/dev/udp -> /dev/inet/udp
/dev/idp -> /dev/inet/idp
/dev/rawip -> /dev/inet/rawip
/dev/inet/arp -> /dev/inet/udp
/dev/inet/rip -> /dev/inet/udp
/dev/nfsd -> /dev/socksys
/dev/XOR -> /dev/null

36 char Netlink 支持
 0 = /dev/route 路由, 设备更新, kernel to user
 3 = /dev/fwmonitor Firewall packet 复制

59 char sf 防火墙模块
 0 = /dev/firewall 与 sf 内核模块通信

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)

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
...

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
...

103 block 审计(Audit)设备
0 = /dev/audit审计(Audit)设备

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

153 block Enhanced Metadisk RAID (EMD) 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

180 char USB 字符设备
96 = /dev/usb/hiddev0 第 1 个 USB 人机界面设备(鼠标/键盘/游戏杆/手写版等人操作计算机的设备)
...
111 = /dev/usb/hiddev15 第 16 个 USB 人机界面设备

180 block USB 块设备(U 盘之类)
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/nvidiactl Nvidia 卡控制设备

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

有没有感到很奇怪？为什么没有 /dev/hda 这样的设备，难道不常用么？原因在于从 2.6.19 版本开始，内核引入了新的 ATA 驱动，将 SATA/PATA 硬盘统一使用 /dev/sd? 来表示了，所以 /dev/hd? 就没有存在的必要了。具体说来也就是你在编译内核的时候不要在使用“ATA/ATAPI/MFM/RLL support”下面的驱动，而是使用更新的“SerialATA and ParallelATA drivers”驱动。

链接、套接字、管道、挂载点

这部分详细说明一些应该或可能存在于 /dev 目录之外的文件。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。

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	标准输入文件描述符
/dev/stdout	fd/1	symbolic	标准输出文件描述符
/dev/stderr	fd/2	symbolic	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
[注意] /dev/X0R 是 <字母 X>--<数字 0>--<字母 R>			

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	mouse port	symbolic	当前鼠标
/dev/tape	tape device	symbolic	当前磁带
/dev/cdrom	CD-ROM device	symbolic	当前 CD-ROM
/dev/cdwriter	CD-writer	symbolic	当前 CD-writer
/dev/scanner	scanner	symbolic	当前扫描仪
/dev/modem	modem ports	symbolic	当前调制解调器
/dev/root	root device	symbolic	当前根文件系统所在设备
/dev/swap	swap device	symbolic	当前 swap 所在设备

/dev/modem 不应当用于能够同时支持呼入和呼出的 modem，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 和 /dev/cdrom 应该分别指向“cooked”设备 /dev/st* 和 /dev/sr*；而 /dev/cdwriter 和 /dev/scanner 应当分别指向恰当的 /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/gpmctl socket (LFS-LiveCD 中出现)
/dev/initctl fifo pipeinit 监听它并从中获取信息(用户与 init 进程交互的通道)

挂载点

以下名称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界面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扮演控制终端角色的任何设备，包括：虚拟控制台、串行接口(已废弃)、伪终端(PTY)。

所有的终端设备共享一个通用的功能集合：line discipline，它既包含通用的终端 line discipline 也包含 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 (Virtual console)和控制台设备(console device)

虚拟控制台是在系统视频监视器上全屏显示的终端。虚拟控制台被命名为编号从 /dev/tty1 开始的 /dev/tty# 。/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*)的机器上存取系统视频卡，注意，不要将 /dev/console 用于此目的。/dev/console 由内核管理，系统消息将被发送到这里。单用户模式下必须允许 login 使用 /dev/console 。

串行接口

这里所说的“串行接口”是指 RS-232 串行接口和任何模拟这种接口的设备，不管是在硬件(例如调制解调器)还是在软件(例如 ISDN 驱动)中模拟。在 linux 中的每一个串行接口都有两个设备 名：主设备或呼入(callin)设备、交替设备或呼出(callout)设备。设备类型之间使用字母的大小写进行区分。比如，对于任意字母 X， “tty”设备名为 /dev/ttyX# ，而“cu”设备名则为 /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# 被保留为本地使用。

伪终端 (PTY)

伪终端用于创建登陆会话或提供其它功能，比如通过 TTY line discipline (包括 SLIP 或者 PPP 功能)来处理任意的数据生成。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/pmx 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)。