

第2章 目录树简介

本章将讨论标准Linux目录树的重要部件，它们是基于FSSTND文件系统标准的（“文件系统分层结构标准” [fhs]<http://www.pathname.com/fhs/>的延续。另外，还可参见“Linux标准基础知识”。其地址是<http://www.linuxbase.org>）。该标准描述了划分目录树的普通方式并指出这种划分方式的动机。出于不同的目的，我们通常需要把目录树分成若干个独立文件系统。此外，还将描述划分目录树的几种方式。

2.1 背景知识

本章没有严格参照Linux文件系统标准——FSSTND 1.2版（参见参考书目），这个版本尽力为如何组合Linux系统内的目录树建立一个标准。这样的标准有它的优点，即便于为Linux系统编写或移植软件，便于管理配置Linux的机器，因为Linux需要的一切东西都将放在“固定”的某个地方。这个标准的背后没有机构要求每个人都必须遵从它，但它正越来越多地得到许多Linux版本的认可。没有特别充足的理由，就拒绝采纳FSSTND不是件好事情。FSSTND尽力遵循Unix的习惯和当前的发展趋势，使Linux系统更加接近其他的Unix系统，Unix系统也更加接近Linux系统。

本章的重点不是讨论FSSTND。为了对其有全面的认识，系统管理员还应该好好看看FSSTND。

本章也不详细讨论全部文件。其目的不是在于细细剖析每个文件，而是从文件系统的角度出发，总览整个系统。有关各个文件的详情，可参考本书其他章节或手册页。

完整的目录树将被分为若干部分，各部分在其自己的磁盘或分区内，更适应于磁盘空间，更易于进行备份和其他系统管理。目录树中，主要的几个部分是root、/usr、/var和/home（如图2-1所示）。各部分用途不同。经过对目录树的设计，它已经能够正常运行于Linux机器组成的网络环境中，这些机器通过只读设备（比如光盘）或采用NFS的网络共享某些文件系统。

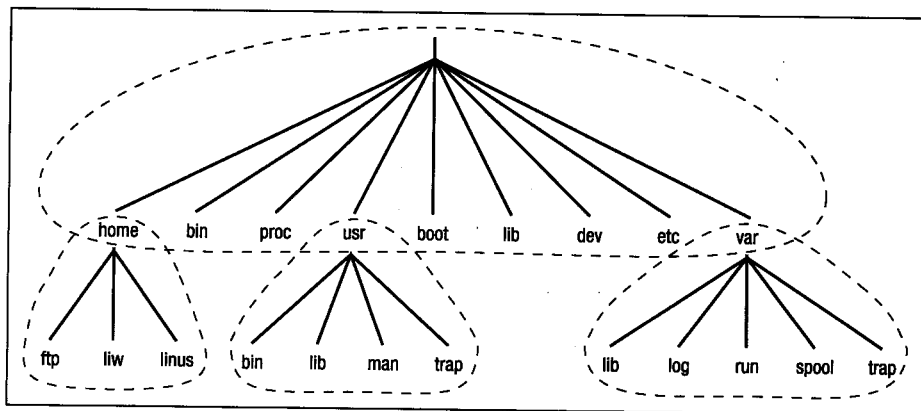


图2-1 Unix目录树中的各部分。虚线表示分区限制

目录树中，各部分扮演的角色是不同的。具体描述如下：

root文件系统是每台机器特有的（它一般保存在本地磁盘上，但也可保存在 RAM磁盘或网络驱动器上），其中包含启动系统和引导装入文件系统所需的文件。因此，对单用户模式来说，root文件系统的内容足之够矣。另外，该文件系统内还包含一些工具，用于修复损坏的系统和从备份中恢复丢失的文件。

/usr文件系统内包含所有的命令、库、手册页和其他没有变动过的文件（这些文件是普通操作期间所需要的）。/usr文件系统内没有专门针对哪台机器的文件，也没有普通操作期间应该修改的文件。这样一来，便可以通过网络实现文件共享，从而有效地节约成本，因为这样可节省磁盘空间（要知道 /usr文件系统起码也有几百个 MB），使管理工作更容易（在更新应用程序时，只有 master/usr需要改动，而不是逐一在每台机器上改动）。即使文件系统是在本地磁盘上，也可以采用只读方式装入它，减小系统崩溃时文件系统受损的可能性。

/var文件系统中包含有变动的文件，比如假脱机目录（用于邮件、新闻、打印机等）、日志文件、格式化的手册页和临时文件。/var内原来的所有东西都已经转移到 /usr下，但其结果是不能采取只读装入 /usr。

/home文件系统中包含用户的根目录，也就是系统上的所有真实数据。把根目录和用户的其他目录树或文件系统区分开的目的是为了便于备份；其他部分通常是不需要备份的，或者说至少不需要经常备份（因为它们几乎没有变动）。一个大型的 /home文件系统可能必须分为若干个小型的文件系统，这就需要在 /home下面加一个额外的命名级别，比如 /home/students和/home/staff。

虽然前面的各部分被称为文件系统，但不要求它们真的在一个独立的文件系统上。如果系统是一个小型的单用户系统，而且用户想一切简单明了的话，以上各部分统统可合成一体。根据磁盘空间的大小和为不同用途分配空间的多少，目录树也可按照不同的方式分成若干个文件系统。可是，目录树中的重要部分就是使所有标准目录名正常运作的部分；也就是说，即使 /var和 /usr真的同在一个分区，/usr/lib/libc.a和 /var/log/message这两个目录名也必须能用，具体做法是把 /var下面的文件移入 /usr/var，再令 /var成为 /usr/var的符号链接。

Unix文件系统结构根据文件的用途，把它们分为各个组，也就是所有的命令归入一个地方，所有的数据文件归入另一个地方，文档又另行存放等等。另外，就是根据文件所属的程序来分组，也就是所有的 Emacs文件归入一个目录，所有的 TeX文件归入另一个目录等等。后一种分组方法的不足就是很难实现文件的共享（程序目录中，通常既有静态和可共享的文件，又有动态和不能共享的文件），而且有时还不方便文件的查找（比如，手册页可能散布在许多地方，对用户或管理员来说，不得不找出自己需要的手册页，无疑是场恶梦）。

2.2 root文件系统

一般说来，root文件系统应该较小，因为其中包含非常关键的文件和一个小型的、非频繁变动的文件系统。受损的root文件系统一般意味着系统不能启动，除非借助于特殊的启动设备（比如说软盘），所以一般不要轻易更改它。

root目录一般不包含任何文件，系统的标准启动镜像除外，这个镜像通常称为 /vmlinuz。其他的所有文件都保存在root文件系统的子目录下：

1. /bin

启动期间，可供普通用户使用的命令（也可能在启动之后）。

2. /sbin

和/bin一样，但不是供普通用户使用的，虽然在必要或经过允许的情况下，普通用户也可使用它们。

3. /etc

某台机器专用的配置文件。

4. /root

用户\texttt{root}的根目录。

5. /lib

root文件系统上的程序所需的共享库。

6. /lib/modules

可装载的内核模块，特别是从灾难中恢复时，启动系统所需的那些模块（比如，网络和文件系统驱动程序）。

7. /dev

设备文件。

8. /tmp

临时文件。启动后的程序运行应该采用 /var/tmp，而不是/tmp，因为前者可能在空间较大的磁盘上。

9. boot

启动装载程序所用文件，比如 LILO。内核镜像通常保存在这里，而不是 root目录中。如果有多个内核镜像，这个目录就可能增长得很快，所以最好把它单独保存在一个文件系统中。这样做的另一个原因是确保内核镜像在 IDE磁盘的前 1024个磁道内。

10. /mnt

系统管理员临时装入的装入点。程序不会自行装入 /mnt。/mnt也可以分为若干个子目录（比如/mnt/dosa可能是使用MS-DOS文件系统的软驱，而/mnt/extra则可能和ext2文件系统如出一辙）。

11. /proc、/usr、/vsn和/home

其他文件系统的装入点。

2.3 /etc文件系统

/etc目录中包含许多文件。下文将讨论其中的一部分。另外的文件，则应该由你决定它们属于哪个程序，并参考那个程序的手册页。许多连网配置文件也被包含在 /etc内，详情参见本书第一部分。

1. /etc/rc或/etc/rc.d或/etc/rc?.d

启动时或运行级别发生变化时运行的脚本或脚本的目录。有关详情，参考 init一章。

2. /etc/passwd

用户数据库，其中有一些字段指定用户名、用户真名、根目录、加密密码以及该用户的其他信息。其格式已编入\man{passwd}手册页。

3. /etc/fdprm

软盘参数表。描述各种软盘的不同格式。供 setfdprm使用。更多详情参考 setfdprm手册页。

4. /etc/fstab

列出启动时由 `mount -a` 命令（在 `/etc/rc` 或等同的启动文件内）自动装入的文件系统。Linux 系统中，这个文件还包含一些信息，这些信息和 `swapon -a` 自动采用的交换区有关。更多详情参考第3章中的“装入和卸载”小节和 `mount` 手册页。

5. /etc/group

类似于 `/etc/passwd`，但它描述的不是用户，而是组。更多详情，请参考 `group` 手册页。

6. /etc/inittab

init 配置文件。

7. /etc/issue

登录提示出现之前的 `getty` 输出。通常包含对系统的简短说明或欢迎消息。其内容由系统管理员决定。

8. /etc/magic

文件的配置文件。其中包含不同文件格式的说明，以便根据格式猜测出文件的类型。更多详情，请参考 `magic` 和 `file` 手册页。

9. /etc/motd

日期消息，是在成功登录之后自动输出的。其内容由系统管理员决定。通常用来提示每个用户，比如既定的系统关闭警告等。

10. /etc/mtab

列出当前已装入的文件系统。最初是由启动脚本设置，由 `mount` 命令自动更新的。用于需要已装入文件系统列表时（比如说在运行 `df` 命令时）。

11. /etc/shadow

在已安装影子密码软件的系统上的影子密码文件。影子密码把已加密的密码从 `/etc/passwd` 移入 `/etc/shadow`；后者只有 `root` 才能读取。这样可进一步保证密码的安全性。

12. /etc/login.defs

login 命令的配置文件。

13. /etc/printcap

类似于 `/etc/termcap`，但对象是打印机。而且采用的语法也不同。

14. /etc/profile、/etc/csh.login 和 /etc/cshrc

登录或启动时，由 Bourne 或 C 外壳执行的文件。这些文件允许系统管理员为所有的用户设置全局默认设置。各外壳的详情，请参考手册页。

15. /etc/securetty

标识安全终端，也就是允许 `root` 通过哪些终端登录。一般说来，只列出了虚拟控制台，如此一来，恶意的用户不可能通过 modem 或网络攻击系统，从而获得超级用户特权（至少说很难）。

16. /etc/shells

列出受托（信得过的）外壳。`chsh` 命令允许用户把他们自己的登录外壳改成这个文件内列出的受托外壳。为计算机提供 FTP 服务的 `.ftpd` 服务器进程，将复查用户的外壳是否在 `/etc/shells` 内，如果在，将允许用户登录，如果不在，就不会让用户登录。

17. /etc/termcap

终端性能数据库。描述按照什么样的“转义序列”来控制不同的终端。编写程序时，不

是直接输出转义序列（只运行于特定品牌的终端），而是查找正确的序列，以执行自己打算在 `/etc/termcap` 内执行的操作。这样做的结果是，多数程序都可利用各种各样的终端。更多详情，请参考 `termcap`、`curs_termcap` 和 `terminfo` 手册页。

2.4 /dev 文件系统

`/dev` 目录下包含所有设备的特定设备文件。设备文件的命名有特殊的约定；对这些约定的描述包括在 `Device` 列表中。设备文件是在安装期间或后期，利用 `/dev/MAKEDEV` 脚本创建的。`/dev/MAKEDEV.local` 是由系统管理员编写的一个脚本，它创建只用于本地的设备文件或链接（也就是那些不属于标准 `MAKEDEV` 的设备文件，比如用于某些非标准设备驱动程序的设备文件）。

2.5 /usr 文件系统

`/usr` 文件系统通常较大，因为所有的程序都是保存在这个文件系统里的。`/usr` 内的文件通常来自 Linux 系统；本地安装的程序和其他东西都保存在 `/usr.local` 下面。这样一来，就能够通过该文件系统的新版本或全新版本来升级系统，根本不需要再次安装所有的程序。下面列出了部分 `/usr` 子目录（有些不重要的目录已被删除；更多详情请参考 `FSSTND`）。

1. `/usr/X11R6`

`X Window System` 包含所有的文件。为了简化 `X` 的开发和安装，`X` 文件尚未和系统的其他部分集成。`/usr/X11R6` 下面有一个目录树，它类似于 `/usr` 本身的目录树。

2. `/usr/X386`

类似于 `/usr/X11R6`，但针对的是 `X11 5` 版本。

3. `/usr/bin`

几乎包含所有的用户命令。有些命令在 `/bin` 或 `/usr/local/bin` 内。

4. `/usr/sbin`

`root` 文件系统上不需要的系统管理命令，例如，大多数服务器程序。

5. `/usr/man`、`/usr/info` 和 `/usr/doc`

分别包含手册页、GNU 信息文档和名目繁多的其他文档文件。

6. `/usr/include`

C 编程语言的头文件。实际上，为了保持数据的一致，这个文件应该保存在 `/usr/lib` 下面，但过去一直都采用这个名称。

7. `/usr/lib`

程序和子系统所用的未变动过的数据文件，其中包括一些和站点有关的配置文件。`lib` 这个名称源于库（`library`）；最初用来编写子例程的库都保存在 `/usr/lib` 内。

8. `/usr/local`

本地安装的软件和其他文件的地方。

2.6 /var 文件系统

`/var` 文件系统内包含系统正常运行时所改动的数据。它是各个系统专有的，也就是说，不能通过网络和其他计算机共享。

1. /var/catman

按需格式化的手册页之缓冲区。手册页的原件通常保存在 /usr/man/man* 内；有的手册页还可能有一个预先格式化版本，这个版本保存在 /usr/man/cat* 内。至于其他手册页，则需要在初次查看时，进行格式化；格式化过后的版本保存在 /var/man 内，以便下一个查看该手册页的用户无须等待其格式化（/var/catman 的清空方式和临时目录的清空方式一样）。

2. /var/lib

系统正常运行期间发生变化的文件。

3. /var/local

针对安装在 /usr/local 内的程序的变化数据（也就是已经由系统管理员安装的程序）。注意，即使是本地安装的程序，如果适当的话，也应该使用其他的 /var 目录（比如 /var/lock）。

4. /var/lock

锁文件。许多程序都习惯在 /var/lock 内建立一个锁文件，借以表明它们正在使用某个特定的设备或文件。其他程序将注意到这个锁文件，并不再尝试使用这个特定的设备或文件。

5. /var/log

日志文件，它源于各个程序，特别是 login（/var/log/wtmp，记录所有的系统登录和注销活动）和 syslog（/var/log/messages，通常保存所有的内核和系统程序消息）。/var/log 内的文件通常增长较快，需要定期清空。

6. /var/run

文件，其中包含系统相关信息，在系统下一次启动之前，都是有效的。例如，/var/run/utmp 内包含和当前登录用户有关的信息。

7. /var/spool

用于邮件、新闻、打印机队列和其他队列作业的目录。对每个不同的假脱机来说，在 /var/spool 下面都有其自己的子目录，比如用户信箱就在 /var/spool/mail 内。

8. /var/tmp

临时文件，通常较大，或需要保存的时间比 /tmp 长的文件（虽然系统管理员可能不允许 /var/tmp 内也保存一些非常老的文件）。

2.7 /proc 文件系统

/proc 文件系统内包含一个有误的文件系统。它不存在于磁盘上。相反地，内核是在内存中创建它的。它用于提供和系统相关的信息（最初是进程相关信息，并由此得名）。下面将对有些比较重要的文件和目录进行解释。/proc 文件系统的更多详情，请参考 proc 手册页。

1. /proc/1

目录，其中有 1 号进程的相关信息。每个进程在 /proc 下面都有一个子目录，这个子目录名就是该进程的编号。

2. /proc/cpuinfo

其中保存关于中央处理器的信息，比如型号、制造商、模型和性能等。

3. /proc/devices

其中列出了已经配置到当前正在运行的内核之中的设备驱动程序。

4. /proc/dma

展示当前正在使用的DMA通道。

5. `/proc/filesystems`

已配置到内核中的文件系统。

6. `/proc/interrupts`

展示哪些中断号正在使用中，以及各中断号使用了多少次。

7. `/proc/ioports`

展示此时哪些I/O端口正在使用中。

8. `/proc/kcore`

系统物理内存的镜像。其大小完全和你的物理内存一样，但事实上占不了多少内存；它是在程序访问它时，即时生成的（记住，除非你把它复制到别的地方，否则，`/proc`根本就不占用任何磁盘空间）。

9. `/proc/kmsg`

内核输出的消息。同时，也被路由到 `syslog`。

10. `/proc/ksyms`

内核的符号表。

11. `/proc/loadavg`

系统的“装载平衡”；无意义的三个识别符，表示此时系统应该做多少操作。

12. `/proc/meminfo`

包含和内存使用相关的信息，其中既包括物理内存，又有交换空间。

13. `/proc/modules`

表明此时正在装载哪些内核模块。

14. `/proc/net`

和网络协议相关的状态信息。

15. `/proc/self`

指向一个程序进程目录的符号链接，这个程序此刻正在查看 `/proc`。如果有两个程序都在查看 `/proc`，它们就会得到两个不同的符号链接。这主要是为了方便程序更容易得到自己的进程目录。

16. `/proc/stat`

关于系统的各种统计数据，比如自系统启动以来出现的页故障次数统计。

17. `/proc/uptime`

表明系统已启用多久。

18. `/proc/version`

内核版本号。

注意，上面的文件越来越发展成为易于理解的文本化文件，但有时，它们采用的格式却是难以理解的。所以，目前有许多命令将上面的文件转换为更便于理解的格式。比如，有个免费程序读取 `/proc/meminfo`，并将指定的字节转换为千字节（同时，还增加了少许信息）。