



链滴

如何操作 Unix / Linux 文件系统

作者: [superstonne](#)

原文链接: <https://ld246.com/article/1566042950156>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p> </p>

<p>作为一名电脑使用者，不懂图形化操作文件系统就不算会使用电脑。那么作为一名程序员，如果懂命令行操作 Linux 文件系统，就不敢说自己是一名合格的程序员。我们工作中使用 Unix / Linux 为应用程序服务器，无非就是把编写好的软件部署到远程服务器，其中少不了对文件的各种操作如果你不会操作文件那么将寸步难行。接下来我们来认识一下 Unix / Linux 文件系统。 </p>

<h5 id="1--本地文件系统">1. 本地文件系统</h5>

<p>在这里我们将讨论面向本地磁盘的文件系统（硬盘，CD-ROM，DVD-ROM，USB）。这类文件系统 UFS（Unix File System），FAT（File Allocation File 通常是 Windows 和 DOS 系统的文件系统），NTFS（New Technology File System，通常是 Windows NT，2000，XP 的文件系统），U F（Universal Disk Formt 通常是 DVD 文件系统），HFS+（Hierarchical File System 分级文件系统，例如 Mac OS X），ISO9660 通常为 CD-ROM 文件系统和 EXT2 扩展文件系统（它是 Linux 默的内核文件系统）等等。 </p>

<p>说了这么多，那我们使用的虚拟机中的 CentOS 7 使用的是什么类型的文件系统呢？使用 df -T 查看一下。 </p>

<p> </p>

<p>从 Type 一列我们看到它使用了 xfs 文件系统，这是一种高性能的日志文件系统，诞生于 1993，在 2000 年被移植到 Linux 系统上。上面介绍的都是面向磁盘的本地文件系统，其实还有面向网络和临时的虚拟文件系统，例如上图中的 tmpfs 就是一种面向内存的临时文件系统。 </p>

<h5 id=""></h5>

<h5 id="2--目录结构">2. 目录结构</h5>

<p>要想熟练操作文件就必须了解系统的目录结构，Unix / Linux 使用分层的文件系统结构，"/" 是文件系统的根基，其他的所有的目录都是从这里开始。我们使用 tree 命令来查看从根目录开始的目录构，只显示一层深度。如下： </p>

<p> </p>

<p>bin 目录主要存放二进制可执行文件，例如上一篇文章我们使用的 passwd，cat，vi 等程序。bo t 目录存放用于启动系统的文件。dev 目录存放设备文件。etc 目录存放系统配置文件，如 用户信息，osts 信息。lib 存放共享库的文件。media 存放挂在的媒体文件。mnt 存放挂在的移动设备文件。opt 额外安装的软件可以指定安装到该目录下，便于管理。proc 存在系统当前正在运行的进程信息。root 我的系统用户主目录。run 未查到相关信息。sbin ->usr/sbin 存放系统管理的必须程序。srv 存 用户主动产生的文件，对外服务。sys 存放全局设备文件，例如总线控制等。tmp 存放临时文件。usr 可以存放任意的文件，例如被很多用户使用的联机帮助页。var 存放长度可变的文件，例如日志文件 </p>

<h5 id="-"></h5>

<h5 id="3--常用的命令">3. 常用的命令</h5>

<p>要想操作文件，必须先学会文件系统的导航。接下来了解几个常用的导航命令吧！ </p>

<p>pwd 显示当前所在的目录，我们执行命令的时候需要确认当前目录是否是我们预想的目录，因 这个命令经常使用。如下显示我们现在位于 /root 目录

 </p>

<p>cd（Change Directory）去到任何我们想去的目录，如我们想去到 /etc 目录：

 </p>

<p>which 和 whereis，我们平常使用的一些工具，假如我们想知道他们真正位于什么目录，可以使

which。例如我们想知道 cd 工具位于什么目录，可以如下操作：

 </p>

<p>which 和 whereis 的不同之处在于 which 搜寻环境变量 PATH 里面指定的内容，whereis 将在 ATH 和 MANPATH 指定的目录中查找。 </p>

<ol start="4" >

find 查找目录或者文件，例如我们需要在 /root 目录下去查找一个叫做 test 的文件：

<p> </p>

>

<ol start="5" >

<p>file 查看文件的类型，例如查看 test 是什么类型的文件，显示结果为一个文本文件

 </p>

<p>ls 显示目录中的内容，假如我们想查看根路径的内容，则可以如下操作： </p>

<p> </p>

>

<p>-l 参数可以显示出文件的对应的详细内容，所有者，权限，大小，上次更改时间，符号链接。同我们也可以加上 -a 参数显示所有文件，包括 . 开头的隐藏文件。 </p>

<h5 id="--" ></h5>

<h5 id="4--文件的类型" >4. 文件的类型 </h5>

<p>也许你已经注意到了，用 ls 命令列出来的文件中的最左边有一些信息。例如 drwxr-xr-x，第一代表的就是文件的类型，后面的 9 位代表的文件的权限。- 代表这是一个文本文件，或者可执行文件 b 代表这是一个块文件，例如硬盘。c 字符设备文件，如硬盘。d 代表这是一个目录文件。l 代表这是个符号链接，可以理解为 Windows 系统中的快捷方式。p 代表这是一个管道，用于通讯。s 代表这套接字，用于通讯。由于我们平时操作会用到符号链接，因此我们简要介绍下符号链接。Unix / Linux 使用 inode 来引用文件而不是文件名，inode 在一个分区中是唯一的。符号链接有 2 种，硬链接和连接。首先我们来创建一个文件，然后分别为它创建一个硬链接和一个软连接。 </p>

<p> </p>

>

<p>接下来我们分别对源文件，硬链接，软链接进行写入操作，我们会发现，对它们中任何的一个的入操作在其他的上面都会反映出来。 </p>

<p> </p>

>

<p>既然效果都一样，那么软链接和硬链接究竟有什么区别呢？硬链接使用 inode 来实现，软链接用文件名来实现。因此，如果我们删除 file 文件，然后创建一个新的 file 文件。此时我们的软链接是然生效的，而硬链接已经失效了。如下实验数据： </p>

<p> </p>

>

<h5 id="---" ></h5>

<h5 id="5--文件的权限" >5. 文件的权限 </h5>

<p>我们使用 ls -l 命令查看文件的时候，最左边开始的字符串反映了文件的权限，如下例： </p>

<p></p>

<p>如上图我们可以清楚的明白各个用户对于文件的权限在哪里查看，那么对应的 r, w, x 分别就读，写，执行权限。我们可以使用 chmod (change mode) 来更改文件的权限，操作的时候有 2 中择，一种是使用符号模式，即 rwx，另外一种绝对模式用 8 个数字分别代表一种权限。使用 chmod 用户 (a/u/g/o) +/- 权限 (rwx) 文件名 命令格式来操作文件权限，实验结果：</p>

<p></p>

<p></p>

<p>同时我们也可以使用绝对模式来操作文件权限，例如 chmod 777 file，代表所有用户都用于对 file 文件的最大操作权限。绝对模式的数字代表的含义如下：</p>

<p>0 无权限，1 执行权限，2 写入权限，3 执行和写入权限，4 读取权限，5 读取和执行权限，6 读和写入权限，7 所有权限。此处留一道脑筋急转弯，为啥不用 3 代表 读取权限呢？</p>

<h5 id="----"></h5>

<h5 id="6--查看文件">6. 查看文件</h5>

<p>程序出错以后难免要去查看日志文件，此时查看文件的命令就尤为重要。我们可以使用 cat, more, less, head, tail 等命令查看文件内容。cat 将内容一股脑全部打印在屏幕上，不适合查看内容多文件。此时使用 more 就可以，查看完后按 enter 键，继续显示下一行，less 命令更加强大，可以使上下方向键来移动。head 查看文件头部的内容，tail 查看尾部的内容。在查看一些不断输出的日志文件时候，我们可以使用 tail -f 命令来不断刷新显示的内容。</p>

<h5 id="-----"></h5>

<h5 id="7--查看分区空间使用率">7. 查看分区空间使用率</h5>

<p>如果有个笨蛋程序员写了一个死循环的程序，那么避免不了产生一个超级大日志文件撑爆服务器那么我们如何查看分区空间使用率呢？df (disk free) 显示磁盘使用情况，例

</p>

<p>掌握以上命令以后，我们已经可以对 Unix / Linux 系统的文件一顿操作了，但是强大的系统远不止这点命令，就让我们在实战的过程中，边用边学习吧。</p>