

04 【实操篇-帮助命令 文件目录管理基础知识】

1.帮助命令

通常linux命令都十分简单，但是有些还是有些复杂度的。比如 `find`，`ps` 这种命令，如果要照顾到所有的场合，可能需要非常巨大的篇幅。但是，万一用到这种偏门的场合怎么办？

全面了解一下是非常有必要的，以便在使用的时候能够唤起记忆中最浅显的印象。然后剩下的，就可以交给类似于 `man` 的这种命令了。Linux上的每一个命令，都会有配套的帮助文件，这远比网络上那些转来转去的信息，正确的多。

正式介绍一下下面的两个命令：

- `man` 用来显示某个命令的文档信息。比如：`man ls`
- `info` 你可以认为和`man`是一样的，虽然有一些能够互补的内容。它们会在内容中进行提示的
- `--help` 很多命令通过参数 `--help` 提供非常简短的帮助信息。这通常是最有用最快捷的用例展示。

如果你根本就记不住一个非常拗口的单词，那就找找这些地方吧

注意：这些帮助信息，仅集中在命令的作用域本身。对于它的组合使用场景，并没有过多信息。也就是说，它教会了你怎么用，但并没有告诉你用它能够来做什么。

TAB补全

现在，在终端里，输入 `ca`，然后快速按2次 `<TAB>` 键盘，命令行会进入补全模式，显示以`ca`打头的所有命令。

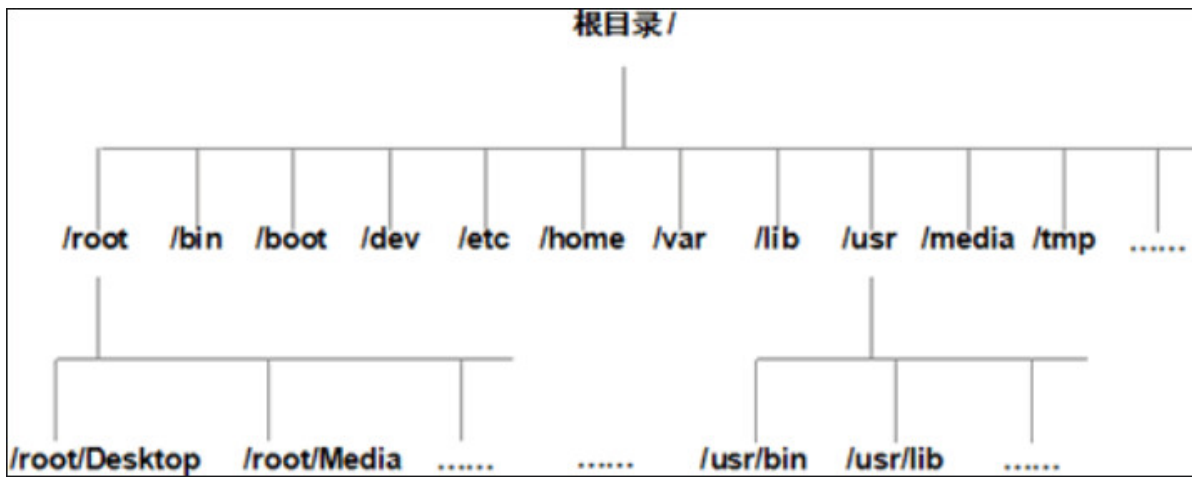
```
[root@localhost ~]# ca
cacertdir_rehash      cache_dump             cache_repair
cache_writeback
ca-legacy             capsh                  case                   catchsegv
cache_check           cache_metadata_size    cache_restore         cal
                      caller                 captainfo             cat                   catman
```

如果你对某个命令，只有模糊的印象，只记得前面的几个字母，这个功能是极好的，命令范围会一步步缩减。

2.文件和目录管理基础知识

2.1 文件系统的层次结构

平时打交道的都是文件，那么，应该如何找到它们呢？很简单，在 Linux 操作系统中，所有的文件和目录都被组织成以一个根节点“/”开始的倒置的树状结构。



其中，目录就相当于 Windows 中的文件夹，目录中存放的既可以是文件，也可以是其他的子目录，而文件中存储的是真正的信息。

文件系统的最顶层是由根目录开始的，系统使用“/”来表示根目录，在根目录之下的既可以是目录，也可以是文件，而每一个目录中又可以包含（子）目录或文件。如此反复就可以构成一个庞大的文件系统。

其实，使用这种树状、具有层次的文件结构主要目的是方便文件系统的管理和维护，想象一下，如果所有的文件都放在一个目录下，其文件系统的管理和维护将变成一场噩梦。

现实中也有许多类似的例子，例如在整个行政管理体制中，村民就相当于文件，他们住在一个村庄中，村庄就是存储村民的目录。许多村又组成了个乡，这个乡就相当于存储村的目录，依此类推，最终就构建出了一个庞大的行政区域管理结构图。

注意，目录名或文件名都是区分大小写的，如 dog、DOG 和 Dog 为 3 个不同的目录或文件。完整的目录或文件路径是由一连串的目录名所组成的，其中每一个目录由“/”来分隔。如 cat 的完整路径是 /home/cat。

在文件系统中，有两个特殊的目录，一个是用户所在的工作目录，即当前目录，可用一个点“.”表示；另一个是当前目录的上一层目录，也叫父目录，用两个点“..”表示。

如果一个目录或文件名是以一个点开始，就表示这个目录或文件是一个隐藏目录或文件。即以默认方式查找（后续会讲查找命令）时，不显示该目录或文件。

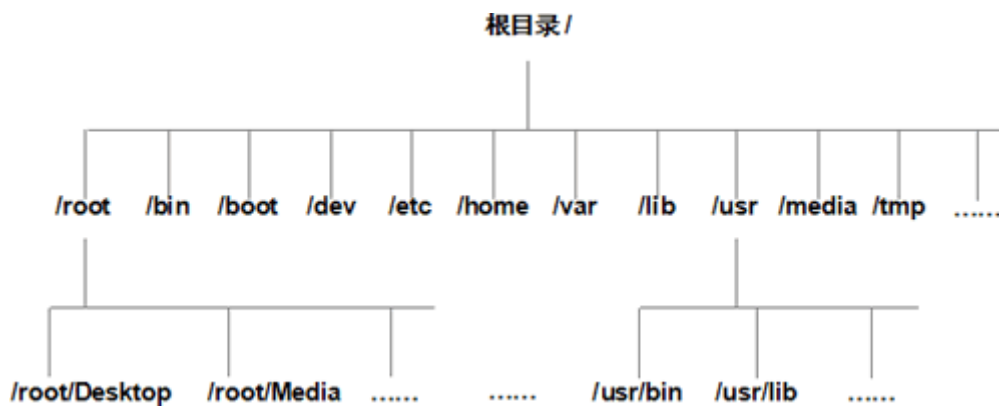
为了方便管理和维护，Linux 系统采用了文件系统层次标准，也称为 FHS 标准，它规定了根目录下各个目录应该存在哪些类型的文件（或子目录），比如说，在 /bin 和 /sbin 目录中存放的应该是可执行文件，有关各个目录存放文件的类型。

2.2 绝对路径和相对路径详解

在 Linux 中，简单的理解一个文件的路径，指的就是该文件存放的位置，例如， /home/cat 就表示的是 cat 文件所存放的位置。只要我们告诉 Linux 系统某个文件存放的准确位置，那么它就可以找到这个文件。

指明一个文件存放的位置，有 2 种方法，分别是使用绝对路径和相对路径。

我们知道，Linux 系统中所有的文件（目录）都被组织成以根目录“/”开始的倒置的树状结构。



绝对路径一定是由根目录 / 开始写起。例如，使用绝对路径的表示方式指明 bin 文件所在的位置，该路径应写为 /usr/bin，测试代码如下：

```
[root@localhost ~]# bin
bash: bin: command not found <-- 没有找到
[root@localhost ~]# /usr/bin
bash: /usr/bin: is a directory <-- 是一个文件
```

可以看到，如果仅传递给 Linux 系统一个文件名，它无法找到指定文件；而当将 bin 文件的绝对路径传递 Linux 系统时，它就可以成功找到。

和绝对路径不同，相对路径不是从根目录 / 开始写起，而是从当前所在的工作目录开始写起。使用相对路径表明某文件的存储位置时，经常会用到前面讲到的 2 个特殊目录，即当前目录（用 . 表示）和父目录（用 .. 表示）。

举个例子，当我们使用 root 身份登录 Linux 系统时，当前工作目录默认为 /root，如果此时需要将当前工作目录调整到 root 的子目录 Desktop 中，当然可以使用绝对路径，示例代码如下：

```
[root@localhost ~]# pwd <-- 显示当前所在的工作路径
/root
[root@localhost ~]# cd /root/Desktop
[root@localhost Desktop]# pwd
/root/Desktop
```

注意，这里所使用的 pwd 和 cd 命令，目前只需知道它们的功能即可，具体用法会在后续文章中作详细讲解。

可以看到，通过使用绝对路径，我们成功地改变了当前工作路径。但除此之外，使用相对路径的方式会更简单。因为目前处于 /root 的位置，而 Desktop 就位于当前目录下，所以：

```
[root@localhost ~]# pwd <-- 显示当前所在的工作路径
/root
[root@localhost ~]# cd ./Desktop
[root@localhost Desktop]# pwd
/root/Desktop
```

此代码中，./Desktop 表示的就是 Desktop 文件相对于 /root 所在的路径。

再举一个例子，如果以 root 身份登录 Linux 系统，并实现将当前工作目录由 /root 转换为 /usr 目录，有以下 2 种方式：

```
#使用绝对路径
[root@localhost ~]# pwd <-- 显示当前所在的工作路径
/root
[root@localhost ~]# cd /usr
[root@localhost ~]# pwd
/usr
#使用相对路径
[root@localhost ~]# pwd <-- 显示当前所在的工作路径
/root
[root@localhost ~]# cd ../usr <-- 相对 root, usr 位于其父目录 /，因此这里要用到 ..
[root@localhost ~]# pwd
/usr
```

总之，绝对路径是相对于根路径 / 的，只要文件不移动位置，那么它的绝对路径是恒定不变的；而相对路径是相对于当前所在目录而言的，随着程序的执行，当前所在目录可能会改变，因此文件的相对路径不是固定不变的。

2.3 文件（目录）命名规则

介绍完 Linux 系统中目录结构之后，读者一定想知道如何为文件或目录命名。

我们知道，在 Linux 系统中，一切都是文件，既然是文件，就必须要有文件名。同其他系统相比，Linux 操作系统对文件或目录命名的要求相对比较宽松。

Linux 系统中，文件和目录的命名规则如下：

1. 除了字符“/”之外，所有的字符都可以使用，但是要注意，在目录名或文件名中，使用某些特殊字符并不是明智之举。例如，在命名时应避免使用 <、>、?、* 和非打印字符等。如果一个文件名中包含了特殊字符，例如空格，那么在访问这个文件时就需要使用引号将文件名括起来。
2. 目录名或文件名的长度不能超过 255 个字符。
3. 目录名或文件名是区分大小写的。如 DOG、dog、Dog 和 DOg，是互不相同的目录名或文件名，但使用字符大小写来区分不同的文件或目录，也是不明智的。
4. 与 Windows 操作系统不同，文件的扩展名对 Linux 操作系统没有特殊的含义，换句话说，Linux 系统并不以文件的扩展名来区分文件类型。例如，dog.exe 只是一个文件，其扩展名 .exe 并不代表此文件就一定是可执行文件。

需要注意的是，在 Linux 系统中，硬件设备也是文件，也有各自的文件名称。Linux 系统内核中的 udev 设备管理器会自动对硬件设备的名称进行规范，目的是让用户通过设备文件的名称，就可以大致推测出设备的属性以及相关信息。

udev 设备管理器会一直以进程的形式运行，并侦听系统内核发出的信号来管理位于 /dev 目录下的设备文件。

表 1 罗列出了 Linux 系统中常见硬件设备的文件名。

硬件设备	文件名称
IDE设备	/dev/hd[a-d]，现在的 IDE设备已经很少见了，因此一般的硬盘设备会以 /dev/sd 开头。
SCSI/SATA/U 盘	/dev/sd[a-p]，一台主机可以有多块硬盘，因此系统采用 a~p 代表 16 块不同的硬盘。
软驱	/dev/fd[0-1]
打印机	/dev/lp[0-15]
光驱	/dev/cdrom
鼠标	/dev/mouse
磁带机	/dev/st0 或 /dev/ht0

2.4 命令基本格式

本节开始，我们不会再见到图形界面了，因为对服务器来讲，图形界面会占用更多的系统资源，而且会安装更多的服务、开放更多的端口，这对服务器的稳定性和安全性都有负面影响。其实，服务器是一个连显示器都没有的家伙，要图形界面干什么？

说到这里，有很多人会很崩溃。笔者就经常听到抱怨 Linux 是落后于时代的老古董，就像笔者的白发一样！但是，大家要理解，对服务器来讲，稳定性、可靠性、安全性才是最主要的。而简单易用不是服务器需要考虑的事情，所以学习 Linux，这些枯燥的命令是必须学习和记忆的内容。

2.5 命令提示符

登录系统后，第一眼看到的内容是：

```
[root@localhost ~]#
```

这就是 Linux 系统的命令提示符。那么，这个提示符的含义是什么呢？

- []：这是提示符的分隔符号，没有特殊含义。
- root：显示的是当前的登录用户，笔者现在使用的是 root 用户登录。
- @：分隔符号，没有特殊含义。
- localhost：当前系统的简写主机名（完整主机名是 localhost.localdomain）。
- ~：代表用户当前所在的目录，此例中用户当前所在的目录是家目录。
- #：命令提示符，Linux 用这个符号标识登录的用户权限等级。如果是超级用户，提示符就是 #；如果是普通用户，提示符就是 \$。

家目录（又称主目录）是什么？Linux 系统是纯字符界面，用户登录后，要有一个初始登录的位置，这个初始登录位置就称为用户的家：

- 超级用户的家目录：/root。
- 普通用户的家目录：/home/用户名。

用户在自己的家目录中拥有完整权限，所以我们也建议操作实验可以放在家目录中进行。我们切换一下用户所在目录，看看有什么效果。

```
[root@localhost ~]# cd /usr/local
```

```
[root@localhost local]#
```

仔细看，如果切换用户所在目录，那么命令提示符中的会变成用户当前所在目录的最后一个目录（不显示完整的所在目录 /usr/ local，只显示最后一个目录 local）。

2.6 命令的基本格式

接下来看看 Linux 命令的基本格式：

```
[root@localhost ~]# 命令[选项][参数]
```

命令格式中的 `[]` 代表可选项，也就是有些命令可以不写选项或参数，也能执行。那么，我们就用 Linux 中最常见的 `ls` 命令来解释一下命令的格式（有关 `ls` 命令的具体用法，后续章节会详细介绍）。如果按照命令的分类，那么 `ls` 命令应该属于目录操作命令。

```
[root@localhost ~]# ls
anaconda-ks.cfg install.log install.log.syslog
```

2.6.1 选项的作用

`ls` 命令之后不加选项和参数也能执行，不过只能执行最基本的功能，即显示当前目录下的文件名。那么加入一个选项，会出现什么结果？

```
[root@localhost ~]# ls -l
总用量44
-rw-----.1 root root 1207 1 月 14 18:18 anaconda-ks.cfg
-rw-r--r--.1 root root 24772 1 月 14 18:17 install.log
-rw-r--r--.1 root root 7690 1 月 14 18:17 install.log.syslog
```

如果加一个 `-l` 选项，则可以看到显示的内容明显增多了。`-l` 是长格式（long list）的意思，也就是显示文件的详细信息。至于 `-l` 选项的具体含义，我们稍后再详细讲解。可以看到选项的作用是调整命令功能。如果没有选项，那么命令只能执行最基本的功能；而一旦有选项，则可以显示更加丰富的数据。

Linux 的选项又分为短格式选项（`-l`）和长格式选项（`--all`）。短格式选项是英文的简写，用一个减号调用，例如：

```
[root@localhost ~]# ls -l
```

而长格式选项是英文完整单词，一般用两个减号调用，例如：

```
[root@localhost ~]# ls --all
```

一般情况下，短格式选项是长格式选项的缩写，也就是一个短格式选项会有对应的长格式选项。当然也有例外，比如 `ls` 命令的短格式选项 `-l` 就没有对应的长格式选项。所以具体的命令选项可以通过后面我们要学习的帮助命令来进行查询。

2.6.2 参数的作用

参数是命令的操作对象，一般文件、目录、用户和进程等可以作为参数被命令操作。例如：

```
[root@localhost ~]# ls -l anaconda-ks.cfg
-rw-----.1 root root 1207 1 月 14 18:18 anaconda-ks.cfg
```

但是为什么一开始 `ls` 命令可以省略参数？那是因为默认参数。命令一般都需要加入参数，用于指定命令操作的对象是谁。如果可以省略参数，则一般都有默认参数。例如：

```
[root@localhost ~]# ls  
anaconda-ks.cfg install.log install.log.syslog
```

这个 ls 命令后面没有指定参数，默认参数是当前所在位置，所以会显示当前目录下的文件名。

总结一下：命令的选项用于调整命令功能，而命令的参数是这个命令的操作对象。