

## 05【实操篇-文件目录类命令】

### 1.pwd 显示当前工作目录的绝对路径

pwd:print working directory 打印工作目录

到现在为止，我们还不知道自己在系统的什么地方。在浏览器上，我们能够通过导航栏上的url，了解到自己在互联网上的具体坐标。相似的功能，是由 pwd 命令提供的，它能够输出当前的工作目录。

pwd 命令是非常非常常用的命令，尤其是在一些 命令提示符 设置不太友好的机器上。另外，它也经常用在shell脚本中，用来判断当前的运行目录是否符合需求。

有很多线上事故，都是由于没有确认当前目录所引起的。比如 `rm -rf *` 这种危险的命令。在执行一些高危命令时，随时确认当前目录，是个好的习惯。

```
[root@www ~]# pwd [-P]
```

选项与参数：

- **-P**：显示出确实的路径，而非使用链接 (link) 路径。

实例：单纯显示出目前的工作目录：

```
[root@www ~]# pwd
/root    <== 显示出目录啦～
```

我们使用root用户默认登陆后，就停留在 `/root` 目录中。Linux中的目录层次，是通过 `/` 进行划分的。

实例显示出实际的工作目录，而非链接档本身的目录名而已。

```
[root@www ~]# cd /var/mail    <==注意，/var/mail是一个链接档
[root@www mail]# pwd
/var/mail    <==列出目前的工作目录
[root@www mail]# pwd -P
/var/spool/mail    <==怎么回事？有没有加 -P 差很多～
[root@www mail]# ls -ld /var/mail
lrwxrwxrwx 1 root root 10 Sep  4 17:54 /var/mail -> spool/mail
# 看到这里应该知道为啥了吧？因为 /var/mail 是链接档，链接到 /var/spool/mail
# 所以，加上 pwd -P 的选项后，会不以链接档的数据显示，而是显示正确的完整路径啊！
```

### 2.ls 列出目录的内容

ls:list 列出目录内容

ls 命令，能够列出相关目录的文件信息。可以被评为linux下最勤劳的命令标兵。

语法：

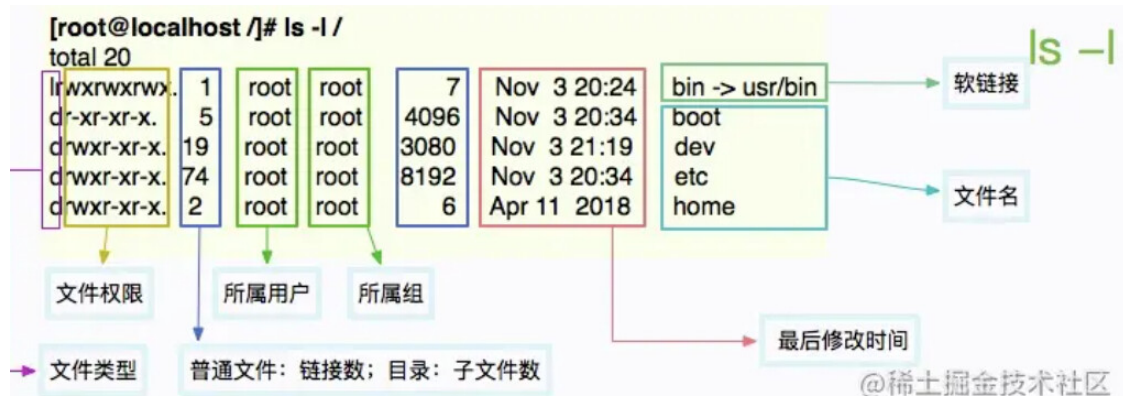
```
[root@www ~]# ls [-aAdFfHiLnrRSt] 目录名称
[root@www ~]# ls [--color={never,auto,always}] 目录名称
[root@www ~]# ls [--full-time] 目录名称
```

选项与参数:

- -a: 全部的文件, 连同隐藏文件(开头为 . 的文件) 一起列出来(常用)
- -d: 仅列出目录本身, 而不是列出目录内的文件数据(常用)
- -l: 长数据串列出, 包含文件的属性与权限等等数据; (常用)

```
[root@localhost /]# ls /
# 注意: ls可以接受路径参数, 你不用先跳转, 就可以输出相关信息
bin boot dev etc home lib lib64 media mnt opt proc root run sbin
srv sys tmp usr var
[root@localhost /]# ls -l /
# 带上 -l参数, 能够看到文件的一些权限信息已经更新日期等。
total 20
lrwxrwxrwx. 1 root root 7 Nov 3 20:24 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 Nov 3 20:34 boot
drwxr-xr-x. 19 root root 3080 Nov 3 21:19 dev
drwxr-xr-x. 74 root root 8192 Nov 3 20:34 etc
drwxr-xr-x. 2 root root 6 Apr 11 2018 home
lrwxrwxrwx. 1 root root 7 Nov 3 20:24 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Nov 3 20:24 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 Apr 11 2018 media
drwxr-xr-x. 2 root root 6 Apr 11 2018 mnt
drwxr-xr-x. 2 root root 6 Apr 11 2018 opt
dr-xr-xr-x. 108 root root 0 Nov 3 21:19 proc
dr-xr-x---. 2 root root 135 Nov 4 07:53 root
drwxr-xr-x. 24 root root 740 Nov 3 21:20 run
lrwxrwxrwx. 1 root root 8 Nov 3 20:24 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 Apr 11 2018 srv
dr-xr-xr-x. 13 root root 0 Nov 3 21:19 sys
drwxrwxrwt. 9 root root 4096 Nov 4 03:40 tmp
drwxr-xr-x. 13 root root 155 Nov 3 20:24 usr
drwxr-xr-x. 19 root root 267 Nov 3 20:34 var
```

每行列出的信息依次是: 文件类型与权限 链接数 文件属主 文件属组 文件大小用byte 来表示 建立或最近修改的时间 名字



直接在你的/root目录里, 执行 `ls -al`, 你会看到更多东西。这些额外的隐藏文件, 都是以 . 开头, 以配置文件居多。这就是参数 a 的作用。

```
[root@localhost ~]# ls -al
total 28
dr-xr-x---.  2 root root  135 Nov  4 07:53 .
dr-xr-xr-x. 17 root root  224 Nov  3 20:28 ..
-rw-----.  1 root root 1273 Nov  3 20:28 anaconda-ks.cfg
-rw-----.  1 root root  246 Nov  4 11:41 .bash_history
-rw-r--r--.  1 root root   18 Dec 28 2013 .bash_logout
-rw-r--r--.  1 root root  176 Dec 28 2013 .bash_profile
-rw-r--r--.  1 root root  176 Dec 28 2013 .bashrc
-rw-r--r--.  1 root root  100 Dec 28 2013 .cshrc
-rw-r--r--.  1 root root  129 Dec 28 2013 .tcshrc
```

ls最常用的，就是加参数 `l` 或者参数 `a`。

细心的同学，应该会注意到两个特殊的目录。`.` 和 `..`。前者表示的是当前目录，而后者表示的是上层目录。

使用 `cd` 命令，将在这些目录中，自由穿梭。

小技巧：如果你对英文日期阅读困难，可以使用 `ls -al --full-time` 查看可读的日期。

## 3.cd 切换目录

cd:Change Directory 切换路径

执行cd命令，可以将工作目录切换到目标文件夹。为了展示cd命令的效果。请在root用户下，执行下面的命令，这将创建一个7层的目录。

```
cd
mkdir -p a1/b2/c3/d4/e5/f6/{g7,g8,g9,g10}
```

我们使用cd命令，切换到最后一层。然后，我们使用 `..` 切换到上层目录。

```
[root@localhost ~]# cd a1/b2/c3/d4/e5/f6/g7
[root@localhost g7]# pwd
/root/a1/b2/c3/d4/e5/f6/g7

[root@localhost g7]# cd ..
[root@localhost f6]# pwd
/root/a1/b2/c3/d4/e5/f6
```

所以，切换到上面n层目录，只需使用多层级的 `../` 即可。有几个特殊的变量，需要说明一下。

- `../` 指的是上层目录
- `../..` 指的是上两层目录
- `./` 指的是当前目录
- `~` 指的是当前的用户目录，这是一个缩写符号
- `-` 使用它，可以在最近两次的目录中来回切换

我们来使用命令把上面这些特殊变量验证一下。

```
# 跳转到用户根目录
[root@localhost tmp]# cd ~
[root@localhost ~]# pwd
```

```
/root

# 进入到第三层目录
[root@localhost ~]# cd a1/b2/c3/
[root@localhost c3]# pwd
/root/a1/b2/c3

# 跳回到前三层目录
[root@localhost c3]# cd ../../..
[root@localhost ~]# pwd
/root

# 跳到上次访问的目录
[root@localhost ~]# cd -
/root/a1/b2/c3
[root@localhost c3]# pwd
/root/a1/b2/c3

# 进入当前目录：等于什么都没干
[root@localhost c3]# cd ./
[root@localhost c3]# pwd
/root/a1/b2/c3
```

## 4.mkdir 创建一个新的目录

mkdir: Make directory 建立目录

语法:

```
mkdir [-mp] 目录名称
```

选项与参数:

- m: 配置文件的权限喔! 直接配置, 不需要看默认权限 (umask) 的脸色~
- p: 帮助你直接将所需要的目录(包含上一级目录)递归创建起来!

实例: 请到/tmp底下尝试创建数个新目录看看:

```
[root@www ~]# cd /tmp
[root@www tmp]# mkdir test      <==创建一名为 test 的新目录
[root@www tmp]# mkdir test1/test2/test3/test4
mkdir: cannot create directory `test1/test2/test3/test4':
No such file or directory      <== 没办法直接创建此目录啊!
[root@www tmp]# mkdir -p test1/test2/test3/test4
```

加了这个 -p 的选项, 可以自行帮你创建多层目录!

实例: 创建权限为 **rwX--X--X** 的目录。

```
[root@www tmp]# mkdir -m 711 test2
[root@www tmp]# ls -l
drwxr-xr-x  3 root  root 4096 Jul 18 12:50 test
drwxr-xr-x  3 root  root 4096 Jul 18 12:53 test1
drwx--x--x  2 root  root 4096 Jul 18 12:54 test2
```

上面的权限部分，如果没有加上 -m 来强制配置属性，系统会使用默认属性。

如果我们使用 -m，如上例我们给予 -m 711 来给予新的目录 drwx--x--x 的权限。

## 5.rmdir 删除空的目录

rmdir:Remove directory 移除目录

语法：

```
rmdir [-p] 目录名称
```

选项与参数：

- **-p**：从该目录起，一次删除多级空目录

删除 runoob 目录

```
[root@www tmp]# rmdir runoob/
```

将 mkdir 实例中创建的目录(/tmp 底下)删除掉！

```
[root@www tmp]# ls -l    <==看看有多少目录存在？
drwxr-xr-x  3 root  root 4096 Jul 18 12:50 test
drwxr-xr-x  3 root  root 4096 Jul 18 12:53 test1
drwx--x--x  2 root  root 4096 Jul 18 12:54 test2
[root@www tmp]# rmdir test    <==可直接删除掉，没问题
[root@www tmp]# rmdir test1   <==因为尚有内容，所以无法删除！
rmdir: `test1': Directory not empty
[root@www tmp]# rmdir -p test1/test2/test3/test4
[root@www tmp]# ls -l        <==您看看，底下的输出中test与test1不见了！
drwx--x--x  2 root  root 4096 Jul 18 12:54 test2
```

利用 -p 这个选项，立刻就可以将 test1/test2/test3/test4 一次删除。

删除完test4发现test3是空目录继续删除，以此类推。

不过要注意的是，这个 rmdir 仅能删除空的目录，你可以使用 rm 命令来删除非空目录。

## 6.touch 创建空文件

1) 基本语法

touch 文件名称

2) 案例实操

```
[root@hadoop101 ~]# touch xiyou/dssz/sunwukong.txt
```

## 7.cp 复制文件或目录

cp 即拷贝文件和目录。

语法：

```
[root@www ~]# cp [-adfilprsu] 来源档(source) 目标档(destination)
[root@www ~]# cp [options] source1 source2 source3 .... directory
```

选项与参数：

- **-i**：若目标档(destination)已经存在时，在覆盖时会先询问动作的进行(常用)
- **-p**：连同文件的属性一起复制过去，而非使用默认属性(备份常用)；
- **-r**：递归持续复制，用于目录的复制行为；(常用)
- **-f**：为强制(force)的意思，若目标文件已经存在且无法开启，则移除后再尝试一次；

用 root 身份，将 root 目录下的 .bashrc 复制到 /tmp 下，并命名为 bashrc

```
[root@www ~]# cp ~/.bashrc /tmp/bashrc
[root@www ~]# cp -i ~/.bashrc /tmp/bashrc
cp: overwrite `/tmp/bashrc'? n <==n不覆盖，y为覆盖
```

## 8.rm 删除文件或目录

rm 是强大的删除命令，它可以永久性地删除文件系统中指定的文件或目录。在使用 rm 命令删除文件或目录时，系统不会产生任何提示信息。

语法：

```
rm [-fir] 文件或目录
```

选项与参数：

- **-f**：就是 force 的意思，忽略不存在的文件，不会出现警告信息；
- **-i**：互动模式，在删除前会询问使用者是否动作
- **-r**：递归删除啊！最常用在目录的删除了！这是非常危险的选项！！

注意，rm 命令是一个具有破坏性的命令，因为 rm 命令会永久性地删除文件或目录，这就意味着，如果没有对文件或目录进行备份，一旦使用 rm 命令将其删除，将无法恢复，因此，尤其在使用 rm 命令删除目录时，要慎之又慎。

【例 1】基本用法。

rm 命令如果任何选项都不加，则默认执行的是"rm -i 文件名"，也就是在删除一个文件之前会先询问是否删除。例如：

```
[root@localhost ~]# touch cangls
[root@localhost ~]# rm cangls
rm:是否删除普通空文件"cangls"?y
#删除前会询问是否删除
```

【例 2】删除目录。

如果需要删除目录，则需要使用"-r"选项。例如：

```
[root@localhost ~]# mkdir -p /test/lm/movie/jp
```

#递归建立测试目录

```
[root@localhost ~]# rm /test
```

rm:无法删除"/test/": 是一个目录

#如果不加"-r"选项,则会报错

```
[root@localhost ~]# rm -r /test
```

rm:是否进入目录"/test"?y

rm:是否进入目录"/test/lm/movie"?y

rm:是否删除目录"/test/lm/movie/jp"?y

rm:是否删除目录"/test/lm/movie"?y

rm:是否删除目录"/test/lm"?y

rm:是否删除目录"/test"?y

#会分别询问是否进入子目录、是否删除子目录

大家会发现,如果每级目录和每个文件都需要确认,那么在实际使用中简直是灾难!

【例3】强制删除。

如果要删除的目录中有1万个子目录或子文件,那么普通的rm删除最少需要确认1万次。所以,在真正删除文件的时候,我们会选择强制删除。例如:

```
[root@localhost ~]# mkdir -p /test/lm/movie/jp
```

#重新建立测试目录

```
[root@localhost ~]# rm -rf /test
```

#强制删除,一了百了

加入了强制功能之后,删除就会变得很简单,但是需要注意,数据强制删除之后无法恢复,除非依赖第三方的数据恢复工具,如extundelete等。但要注意,数据恢复很难恢复完整的数据,一般能恢复70%~80%就很难得了。所以,与其把宝压在数据恢复上,不如养成良好的操作习惯。

虽然"-rf"选项是用来删除目录的,但是删除文件也不会报错。所以,为了使用方便,一般不论是删除文件还是删除目录,都会直接使用"-rf"选项。

## 9.mv 移动文件与目录或重命名

语法:

```
[root@www ~]# mv [-fiu] source destination
```

```
[root@www ~]# mv [options] source1 source2 source3 .... directory
```

选项与参数:

- -f: force 强制的意思,如果目标文件已经存在,不会询问而直接覆盖;
- -i: 若目标文件(destination)已经存在时,就会询问是否覆盖!
- -u: 若目标文件已经存在,且source比较新,才会升级(update)

复制一文件,创建一目录,将文件移动到目录中

```
[root@www ~]# cd /tmp
```

```
[root@www tmp]# cp ~/.bashrc bashrc
```

```
[root@www tmp]# mkdir mvtest
```

```
[root@www tmp]# mv bashrc mvtest
```

将某个文件移动到某个目录去,就是这样做!

将刚刚的目录名称更名为 mvtest2

```
[root@www tmp]# mv mvtest mvtest2
```

## 10.cat 查看文件内容

为了查看文件的生成效果，可以使用cat命令检测。cat命令将会把文件的内容，输出打印到终端上。如果加上参数n，甚至可以打印行号。效果如下：

```
[root@localhost ~]# cat spring
10
11
12
13
14
15
16
17
18
19
20
[root@localhost ~]# cat -n spring
1  10
2  11
3  12
4  13
5  14
6  15
7  16
8  17
9  18
10 19
11 20
```

除了查看文件内容，cat命令通常用在更多的地方。只有和其他命令联合起来，它才会觉得生活有意义。

```
# 合并a文件和b文件到c文件
cat a b>> c

# 把a文件的内容作为输入，使用管道处理。我们在后面介绍
cat a | cmd

# 写入内容到指定文件。在shell脚本中非常常用。我们在后面会多次用到这种写法
cat > index.html <<EOF
<html>
  <head><title></title></head>
  <body></body>
</html>
EOF
```

由于我们的文件不大，cat命令没有什么危害。但假如文件有几个GB，使用cat就危险的多，这只叫做猫的小命令，会在终端上疯狂的进行输出，你可以通过多次按ctrl+c来终止它。



# 11.less 分屏显示文件内容

既然cat命令不适合操作大文件，那一定有替换的方案。less和more就是。由于less的加载速度比more快一些，所以现在一般都使用 less。它最主要的用途，是用来分页浏览文件内容，并提供一些快速查找的方式。less是一个交互式的命令，你需要使用一些快捷键来控制它。

不仅如此，为了方面用户浏览文本内容，less 命令还提供了以下几个功能：

- 使用光标键可以在文本文件中前后（左后）滚屏；
- 用行号或百分比作为书签浏览文件；
- 提供更加友好的检索、高亮显示等操作；
- 兼容常用的字处理程序（如 Vim、Emacs）的键盘操作；
- 阅读到文件结束时，less 命令不会退出；
- 屏幕底部的信息提示更容易控制使用，而且提供了更多的信息。

less 命令的基本格式如下：

```
[root@localhost ~]# less [选项] 文件名
```

此命令可用的选项以及各自的含义如表所示。

选项	选项含义
-N	显示每行的行号。
-S	行过长时将超出部分舍弃。
-e	当文件显示结束后，自动离开。
-g	只标志最后搜索到的关键同。
-Q	不使用警告音。
-i	忽略搜索时的大小写。
-m	显示类似 more 命令的百分比。
-f	强迫打开特殊文件，比如外围设备代号、目录和二进制文件。
-s	显示连续空行为一行。
-b <缓冲区大小>	设置缓冲区的大小。
-o <文件名>	将 less 输出的内容保存到指定文件中。
-x <数字>	将【Tab】键显示为规定的数字空格。

在使用 less 命令查看文件内容的过程中，和 more 命令一样，也会进入交互界面，因此需要读者掌握一些常用的交互指令，如表所示。

- 空格 向下滚屏翻页
- b 向上滚屏翻页
- / 进入查找模式，比如 /1111 将查找1111字样
- q 退出less
- g 到开头
- G 去结尾
- j 向下滚动

- **k** 向上滚动，这两个按键和vim的作用非常像

【例 1】使用 less 命令查看 /boot/grub/grub.cfg 文件中的内容。

```
[root@localhost ~]# less /boot/grub/grub.cfg
#
#DO NOT EDIT THIS FILE
#
#It is automatically generated by grub-mkconfig using templates from /etc/grub.d
and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
if [ -s $prefix/grubenv ]; then
    set have_grubenv=true
    load_env
fi
set default="0"
if [ "${prev_saved_entry}" ]; then
    set saved_entry="${prev_saved_entry}"
    save_env saved_entry
    set prev_saved_entry= save_env prev_saved_entry
    set boot_once=true
fi

function savedefault {
    if [ -z "${boot_once}" ]; then
        :
```

可以看到，less 在屏幕底部显示一个冒号（:），等待用户输入命令，比如说，用户想向下翻一页，可以按空格键；如果想向上翻一页，可以按 b 键。

## 12.echo 输出内容到控制台

echo 输出内容到控制台

### 基本语法

echo [选项] [输出内容]

### 选项：

- -e：支持反斜线控制的字符转换

控制字符	作用
\	输出\本身
\n	换行符
\t	制表符，也就是 Tab

### 案例实操

```
[atguigu@hadoop101 ~]$ echo "hello\tworld"
hello\tworld
[atguigu@hadoop101 ~]$ echo -e "hello\tworld"
hello world
```

## 13.head 显示文件头部内容

---

取出文件前面几行

语法：

```
head [-n number] 文件
```

选项与参数：

- -n：后面接数字，代表显示几行的意思

```
[root@www ~]# head /etc/man.config
```

默认的情况中，显示前面 10 行！若要显示前 20 行，就得要这样：

```
[root@www ~]# head -n 20 /etc/man.config
```

## 14.tail 输出文件尾部内容

---

取出文件后面几行

语法：

```
tail [-n number] 文件
```

选项与参数：

- -n：后面接数字，代表显示几行的意思
- -f：表示持续侦测后面所接的档名，要等到按下[ctrl]-c才会结束tail的侦测

```
[root@www ~]# tail /etc/man.config
# 默认的情况中，显示最后的十行！若要显示最后的 20 行，就得要这样：
[root@www ~]# tail -n 20 /etc/man.config
```

对于部分程序员来说，`tail -f`或许是最常用的命令之一。它可以在控制终端，实时监控文件的变化，来看一些滚动日志。比如查看nginx或者tomcat日志等等。

```
# 滚动查看系统日志
[root@localhost ~]# tail -f anaconda-ks.cfg
@server-platform
@server-policy
pax
oddjob
sgpio
certmonger
pam_krb5
krb5-workstation
perl-DBD-SQLite
%end
#光标不会退出文件，而会一直监听在文件的结尾处
```

这条命令会显示文件的最后 10 行内容，而且光标不会退出命令，每隔一秒会检查一下文件是否增加新的内容，如果增加就追加到原来的输出结果后面并显示。因此，这时如果向文件中追加一些数据（需要开启一个新终端）：

```
[root@localhost ~]# echo 2222222222 >> anaconda-ks.cfg
[root@localhost ~]# echo 3333333333 >> anaconda-ks.cfg
#在新终端中通过echo命令向文件中追加数据
```

那么，在原始的正在监听的终端中，会看到如下信息：

```
[root@localhost ~]# tail -f anaconda-ks.cfg @server-platforin
@server-policy
pax
oddjob
sgpio
certmonger
pam_krb5
krb5-workstation
perl-DBD-SQLite
%end
2222222222
3333333333
#在文件的结尾处监听到了新增数据
```

如果想终止输出，按【Ctrl+c】键中断 tail 命令即可。

通常情况下，日志滚动的过快，依然会造成一些困扰，需要配合grep命令达到过滤效果。

```
# 滚动查看包含info字样的日志信息
tail -f /var/log/messages | grep info
```

对于tail命令来说，还有一个大写的参数F。这个参数，能够监控到重新创建的文件。比如像一些log4j等日志是按天滚动的，tail -f无法监控到这种变化。

## 15.> 输出重定向和 >> 追加

### 1) 基本语法

(1) ls -l > 文件 （功能描述：列表的内容写入文件 a.txt 中（覆盖写））

- (2) `ls -al >> 文件` (功能描述: 列表的内容追加到文件 aa.txt 的末尾)
- (3) `cat 文件1 > 文件2` (功能描述: 将文件1 的内容覆盖到文件2)  
`cat 文件1 文件2 > 文件3` (功能描述: 将文件1 和 2的内容合并后输出到文件3中。)
- (4) `echo "内容" >> 文件`

## 2) 案例实操

- (1) 将 `ls` 查看信息写入到文件中

```
[root@hadoop101 ~]# ls -l>houge.txt
```

- (2) 将 `ls` 查看信息追加到文件中

```
[root@hadoop101 ~]# ls -l>>houge.txt
```

- (3) 采用 `echo` 将 `hello` 单词追加到文件中

```
[root@hadoop101 ~]# echo hello>>houge.txt
```

- (4) 将文件 `file1.txt` 和 `file2.txt` 的内容合并后输出到文件 `file3.txt` 中。

```
[root@localhost base]# ls
file1.txt  file2.txt
[root@localhost base]# cat file1.txt
ds(file1.txt)
[root@localhost base]# cat file2.txt
is great(file2.txt)
[root@localhost base]# cat file1.txt file2.txt > file3.txt
[root@localhost base]# more file3.txt
#more 命令可查看文件中的内容
ds(file1.txt)
is great(file2.txt)
[root@localhost base]# ls
file1.txt  file2.txt  file3.txt
```

# 16.history 查看已经执行过历史命令

## 1) 基本语法

`history` (功能描述: 查看已经执行过历史命令)

## 2) 案例实操

- (1) 查看已经执行过的历史命令

```
[root@hadoop101 test1]# history
```

- (2) 显示最近3条命令历史

```
histroy 3
```

- (3) 清除历史记录

```
history -c
```

# 17.In软链接

软链接也称为符号链接，类似于 windows 里的快捷方式，有自己的数据块，主要存放了链接其他文件的路径。

### 1) 基本语法

```
ln -s [原文件或目录] [软链接名] (功能描述：给原文件创建一个软链接)
```

### 2) 经验技巧

删除软链接：rm -rf 软链接名，而不是 rm -rf 软链接名/

**如果使用 rm -rf 软链接名/ 删除，会把软链接对应的真实目录下内容删掉**

查询：通过 ll 就可以查看，列表属性第 1 位是 l，尾部会有位置指向。

### 3) 案例实操

#### (1) 创建软连接

```
[root@hadoop101 ~]# mv hougex.txt xiyou/dssz/
[root@hadoop101 ~]# ln -s xiyou/dssz/hougex.txt ./houzi
[root@hadoop101 ~]# ll
lrwxrwxrwx. 1 root root 20 6月 17 12:56 houzi ->
xiyou/dssz/hougex.txt
```

#### (2) 删除软连接(注意不要写最后的/)

```
[root@hadoop101 ~]# rm -rf houzi
```

#### (3) 进入软连接实际物理路径

```
[root@hadoop101 ~]# ln -s xiyou/dssz/ ./dssz
[root@hadoop101 ~]# cd -P dssz/
```

## 18.总结

### 18.1 文件剪贴删除复制重名等

- pwd：Print Working Directory，显示当前工作目录的绝对路径。
- ls：-a：显示当前目录所有的文件和目录，包括隐藏的；
  - -l：以列表的方式显示信息。
- cd：cd ~：回到自己的家目录；cd ...：回到当前目录的上一级目录。
- mkdir：创建目录；-p：创建多级目录。
- rmdir：删除空目录。rmdir不能删除非空的目录。如果需要删除非空的目录，需要使用rm -rf。
- cp：拷贝文件到指定目录；
  - -r：递归复制整个文件夹。强制覆盖不提示的方法：
  - cp命令改为\cp
- rm：移除文件或目录；
  - -r：递归删除整个文件夹；
  - -f：强制删除不提示。
- mv：移动文件与目录或重命名，两种功能！
- touch：创建空文件。可以一次性创建多个文件

- ln 给文件创建一个软连接
  - 用法:ln -s [源文件或目录][软连接名]

## 18.2 文件查看

- cat: 查看文件内容。只能浏览文件，而不能修改文件。
  - -n: 显示行号。
  - 结尾加上 | more: 分页显示，不会全部一下显示完。
- more: 是一个基于VI编辑器的文本过滤器，它以全屏幕的方式按页显示文本文件的内容。more还内置了很多快捷键：
  - 空白键 (Space) : 向下翻一页
  - Enter: 向下翻一行
  - q: 立刻离开more, 不再显示该文件内容
  - Ctrl + F: 向下滚动一屏
  - Ctrl + B: 返回上一屏
  - = :输出当前行的行号
  - :f 输出文件名和当前行的行号
- less: 用来分屏查看文件内容，与more相似，但是更强大，支持各种显示终端。less指令在显示文件内容时，并不是一次将整个文件加载之后才显示，而是根据显示需要加载内容。对于显示大型文件具有较高的效率。
- head: 显示文件的开头部分。-n 5: 看前面5行内容。
- tail: 输出文件中尾部的内容。
  - -n 5: 看后面5行内容。
  - -f: 时事追踪该文档的所有更新
- >指令: 输出重定向。如果不存在会创建文件，否则会将原来的文件内容覆盖。
- >>指令: 追加。如果不存在会创建文件，否则不会覆盖原来的文件内容，而是追加到文件的尾部。
- echo: 输出内容到控制台。
- history: 查看历史指令