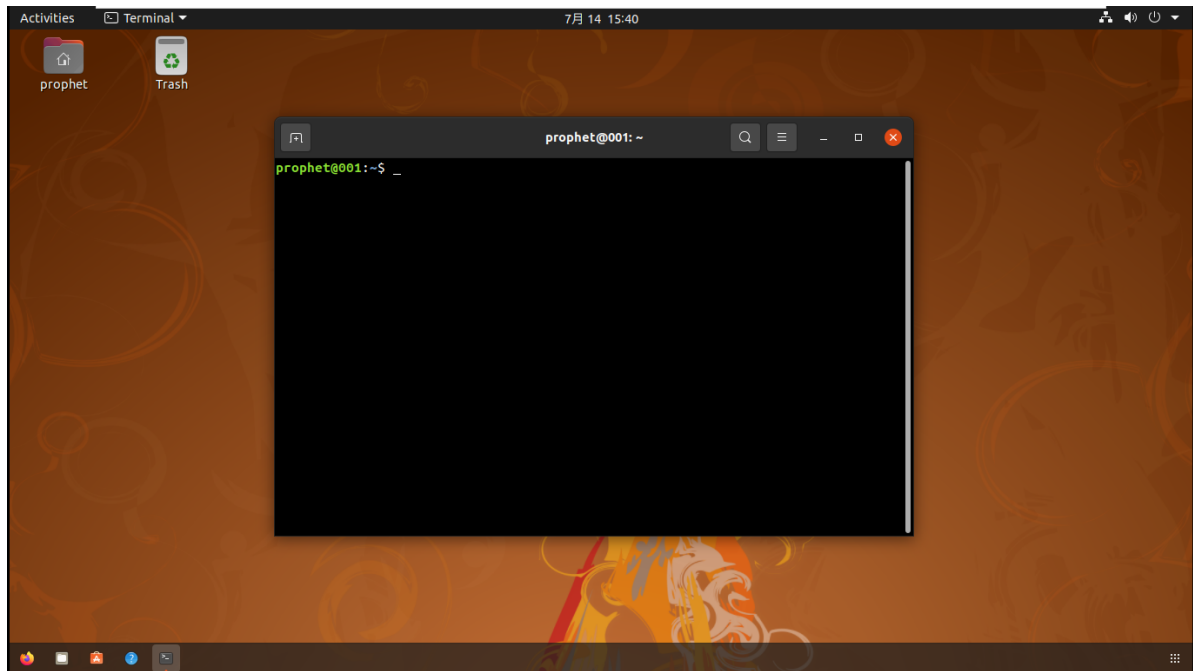


学习命令前准备

注意：在 Linux 命令中是区分字母大小写的切记勿忘！！

终端 (Terminal)

你要写的命令将会在终端机上写出，如图这是终端机。和 Windows cmd 类似。



呼出终端机方法

- 默认快捷键： `Ctrl` + `Shit` + `T`
- 也可以通过鼠标右键 open in terminal **注意这种打开方式的目录起始位置是桌面**
- 桌面右下角的 show all application 中找到 Terminal 打开。

终端机快捷键 Windows键盘

- `Ctrl` + `C` (kill foreground process) 发送SIGINT信号给前台进程组中的所有进程，强制终止程序的执行
- `Ctrl` + `L` 清屏
- `↑` / `↓` 或 `Ctrl` + `P` / `N` 可以翻找之前写过的命令
- `Tab` 补全命令
- `Ctrl` + `←` / `→` 跳跃查阅目录单词（适用于下划线命名法）
- `Ctrl` + `A` 输入的光标跳到这个命令的开头
- `Ctrl` + `E` 输入的光标跳到这个命令的结尾
- `Ctrl` + `B` 向命令的开头移动光标一次一字符
- `Ctrl` + `F` 向命令的末尾移动光标一次一字符

- `Ctrl` + `H` 删除命令中一个字符 相当于退格键
- `Ctrl` + `T` 光标左侧的字符向右侧移动 一次移动一个字符
- `Ctrl` + `K` 删除光标前不包括光标的所有字符
- `Ctrl` + `U` 删除光标后包括光标的所有字符
- `Ctrl` + `R` 搜索之前使用过的命令（需要手动输入命令字符筛选）
- `Ctrl` + `J` 相当于回车
- `Ctrl` + `Y` 写入上一条命令

打开终端第一条代码解析

```
prophet@001:~$
```

用户名 @ 机器名：当前所在目录 \$ 表示等待用户输入

- `~` 表示用户 `home` 目录
- `home` 是存放所有用户文件的根目录
- `Settings` 中 `About` 可以查看机器名与相关信息

Linux 的目录，它们和Windows有区别

- Windows 盘符，文件目录 `\` 反斜线
- Linux一切皆：文件夹+文件，文件目录 `/` 斜线

简单介绍 Linux 的根目录

- `/linux` 根目录
- `/bin` 系统的二进制目录 GUN工具 命令
- `/cdrom` 光驱
- `/etc` 系统配置文件
- `/home` 用户主目录
- `/lib` 库目录 存放库文件
- `/lost+found` 当系统发生错误时，将一些遗失的片段放置在这个目录下
- `/mnt` 手动挂载目录（由个人手动建立或删除载点，并非系统建立或删除）
- `/proc` 伪文件系统
- `/run` 运行目录 表示保存在内存（或基于磁盘的交换空间）中的数据
- `/snap` 包管理，软件包安装管理方式
- `/tmp` 临时目录
- `/var` 可变目录
- `/boot` 启动目录

- /dev 设备目录 (设备分区)
- /media 存放自动挂载的硬件(载点都是由系统自动建立和删除的)
- /opt 可选目录
- /root root 用户的主目录 管理员
- /sbin 系统二进制目录, GNU高级管理员使用的命令或工具
- /srv 服务目录
- /usr 用户二进制目录, GNU工具

详见: **FHS文件系统层级标准**。

基础命令展示

Linux 的命令有很多, 未来有更多, 这里仅介绍种类, 和使用率高的命令。完整命令可参考:

<https://github.com/jaywcjlove/linux-command>

或使用 `man + 想要查询的指令`

例:

```
prophet@001:~$ man
What manual page do you want?
For example, try 'man man'.
prophet@001:~$ man man
```

```
MAN(1)                                Manual pager utils                                MAN(1)

NAME
    man - an interface to the system reference manuals

SYNOPSIS
    man [man options] [[section] page ...] ...
    man -k [apropos options] regexp ...
    man -K [man options] [section] term ...
    man -f [whatis options] page ...
    man -l [man options] file ...
    man -w|-W [man options] page ...

DESCRIPTION
    man is the system's manual pager. Each page argument given to man is normally the name of a
    program, utility or function. The manual page associated with each of these arguments is
    then found and displayed. A section, if provided, will direct man to look only in that sec-
    tion of the manual. The default action is to search in all of the available sections follow-
    ing a pre-defined order (see DEFAULTS), and to show only the first page found, even if page
    exists in several sections.

    The table below shows the section numbers of the manual followed by the types of pages they
    contain.

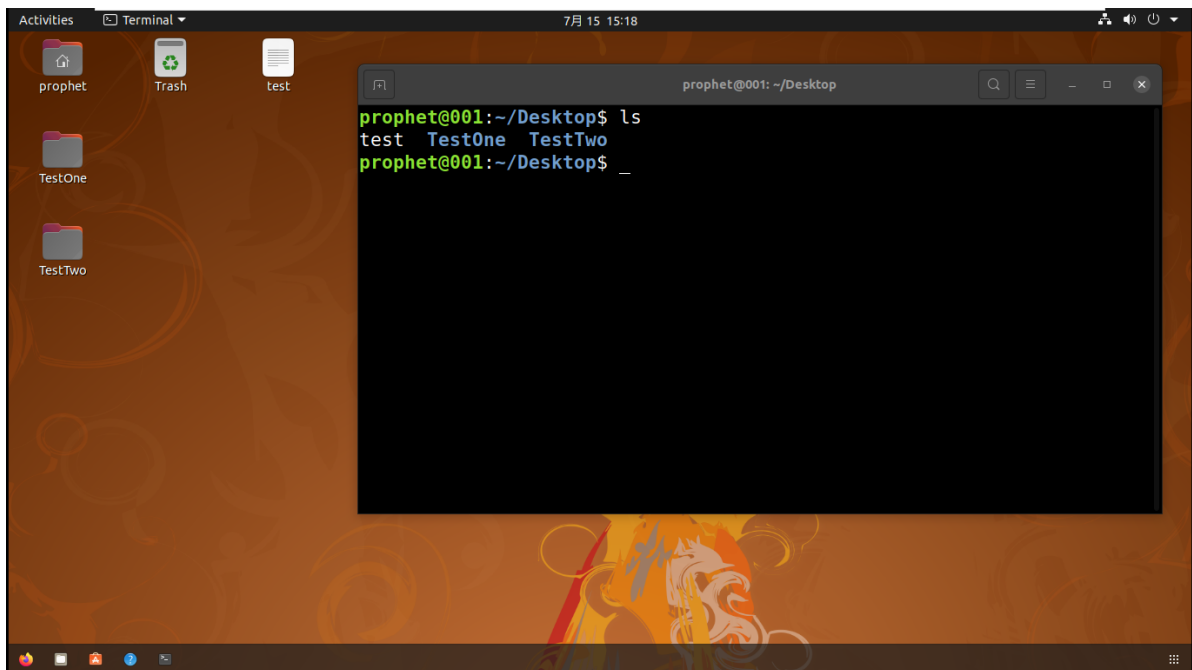
    1 Executable programs or shell commands
    2 System calls (functions provided by the kernel)
    3 Library calls (functions within program libraries)
    4 Special files (usually found in /dev)
Manual page man(1) line 1 (press h for help or q to quit)
```

man 命令是 Linux 下的帮助指令，通过 man 指令可以查看 Linux 中的指令帮助、配置文件帮助和编程帮助等信息

ls 显示目录内容列表

ls 命令就是list的缩写，用来显示目标列表，在Linux中是使用率较高的命令。ls命令的输出信息可以进行彩色加亮显示，以区分不同类型的文件。

例：



ls 延申命令

语法

```
ls [选项] [路径或文件名...]  
    [-labcdfgiklmnopqrstuxABCDGFLNQRSUX] [-w cols] [-T cols] [-I pattern] [--full-time]  
    [--format={long,verbose,commas,across,vertical,single-column}]  
    [--sort={none,time,size,extension}] [--time={atime,access,use,ctime,status}]  
    [--color[={none,auto,always}]] [--help] [--version] [--]
```

选项是可以组合的

可给指定文件夹名的路径查指定文件夹（文件也可以），在此基础上也可进行模糊查询。

模糊查询：

名字有相同字符的经筛选。

通配符为模糊查询服务：

- * 号替代之后的多个字符
- ? 号替代之后的一个字符
- [list] 匹配list中的任意一个字符，如 a[a, b, c]b
- [!list] 不匹配list中的任意一个字符，如 a[!a, b, c]b
- [c1-c2] 匹配c1到c2中的任意字符，其中c1到c2必须是连续的字符，如a[1-5]b,也可使用[!c1-c2]不匹配
- !(文件名) 排除括号里的文件名（仅限一个文件）

[] 可写多个，一个[]对应名字中一个字符的筛方案

常用命令

命令 ll

列出所有文件（包括隐藏）的详细信息

相当于：命令 `ls -l` 组合命令 `ls -a` 为命令 `ls -al` 的简写

命令 `ls -l`：列出当前目录可见文件详细信息

命令 `ls -a`：列出所有文件，包括以 "." 开头的隐含文件

命令 `ls -al`：列出所有文件（包括隐藏）的详细信息

例:

```
prophet@001:~/Desktop$ ls -a
.  ..  test  TestOne  TestTwo
prophet@001:~/Desktop$ ls -l
total 12
-rw-rw-r-- 1 prophet prophet  27 7月  15 15:08 test
drwxrwxr-x 2 prophet prophet 4096 7月  15 15:11 TestOne
drwxrwxr-x 2 prophet prophet 4096 7月  15 15:11 TestTwo
prophet@001:~/Desktop$ ls -al
total 20
drwxr-xr-x  4 prophet prophet 4096 7月  15 15:11 .
drwxr-xr-x 15 prophet prophet 4096 7月  10 12:43 ..
-rw-rw-r--  1 prophet prophet  27 7月  15 15:08 test
drwxrwxr-x  2 prophet prophet 4096 7月  15 15:11 TestOne
drwxrwxr-x  2 prophet prophet 4096 7月  15 15:11 TestTwo
prophet@001:~/Desktop$ ll
total 20
drwxr-xr-x  4 prophet prophet 4096 7月  15 15:11 ./
drwxr-xr-x 15 prophet prophet 4096 7月  10 12:43 ../
-rw-rw-r--  1 prophet prophet  27 7月  15 15:08 test
drwxrwxr-x  2 prophet prophet 4096 7月  15 15:11 TestOne/
drwxrwxr-x  2 prophet prophet 4096 7月  15 15:11 TestTwo/
prophet@001:~/Desktop$ _
```

`-i` # 输出文件前先输出文件系列号（即 `i` 节点号：`i-node number`）。

cd 切换用户当前工作目录

选项

- `-L`（默认值）如果要切换到的目标目录是一个符号连接，那么切换到符号连接的目录。
- `-P` 如果要切换到的目标目录是一个符号连接，那么切换到它指向的物理位置目录。
- `-` 当前工作目录将被切换到环境变量 `OLDPWD` 所表示的目录，**也就是前一个工作目录**。

主要用途

- 切换工作目录至 `dir`。其中 `dir` 的表示法可以是绝对路径或相对路径。
- 若参数 `dir` 省略，则默认为使用者的 shell 变量 `$HOME`。
- 如果 `dir` 指定为 `~` 时表示为使用者的 shell 变量 `$HOME`，`.` 表示当前目录，`..` 表示当前目录的上一级目录。
- 环境变量 `CDPATH` 是由冒号分割的一到多个目录，你可以将常去的目录的上一级加入到 `CDPATH` 以便方便访问它们；如果 `dir` 以 `/` 开头那么 `CDPATH` 不会被使用。
- 当 `shopt` 选项 `cdable_vars` 打开时，如果 `dir` 在 `CDPATH` 及当前目录下均不存在，那么会把它当作变量，读取它的值作为要进入的目录。

例子

```
cd    # 进入用户主目录；
cd /  # 进入根目录
cd ~  # 进入用户主目录；
cd .. # 返回上级目录（若当前目录为“/”，则执行完后还在“/”；“..”为上级目录的意思）；
cd ../.. # 返回上两级目录；
cd !$  # 把上个命令的参数作为cd参数使用。（参数：上一条命令的最后一条路径）
```

Linux上的路径

绝对路径与相对路径

绝对路径：最全的文件位置，从最起始的目录开始。

例：tool 文件夹的绝对路径

```
/home/prophet/Desktop/TestOne/tool
```

相对路径：以某一个文件或文件夹的路径为参照，判断另一个文件或文件夹的路径，相同的省略，不同的写出。

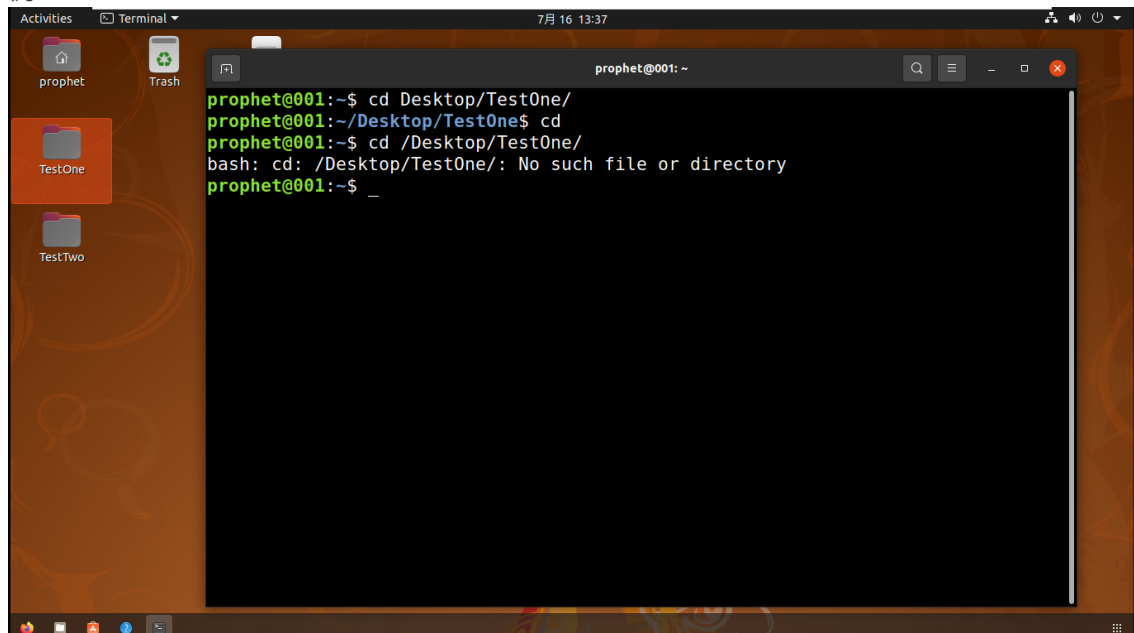
例：以 Desktop 文件夹参照，tool文件夹的 相对路径

```
TestOne/tool
```

注意事项

- Linux 的路径符文件夹层级符号是：/ Windows 是：\
- 相对路径前不能有 / 符号，该符号代表根目录，会找不到你想要的文件夹或某文件。

例：



- 如果相对路径就是想用 / 需要才前面加上 . 组成: ./。
- . 代表: 当前目录用 . 省略, 个人认为这样写失去相对路径的意义了。

touch 创建新的空文件

如果文件已经存在, 再次使用改文件名的 touch 命令, 不会覆盖文件的内容, 但会覆盖文件的创世时间。

可以使用 && 拼接命令

```
cd 1A && touch abc.txt
```

语法

```
touch(选项)(参数)
```

选项

```
-a: 或--time=atime或--time=access或--time=use 只更改存取时间;  
-c: 或--no-create 不建立任何文件;  
-d: <时间日期> 使用指定的日期时间, 而非现在的时间;  
-f: 此参数将忽略不予处理, 仅负责解决BSD版本touch指令的兼容性问题;  
-m: 或--time=mtime或--time=modify 只更改变动时间;  
-r: <参考文件或目录> 把指定文件或目录的日期时间, 统统设成和参考文件或目录的日期时间相同;  
-t: <日期时间> 使用指定的日期时间, 而非现在的时间; 书写格式为 YYMMDDhhmm  
--help: 在线帮助;  
--version: 显示版本信息。
```

参数

文件: 指定要设置时间属性的文件列表。

cp 复制文件或文件夹到那个文件或文件夹

cp 命令复制到的文件夹或文件: 如果不存在会新建, 存在会被覆盖。

pwd (英文全拼: print working directory) 命令用于显示用户当前所在的工作目录 (以绝对路径显示)。

语法

```
cp 选项 参数
```


选项

- a: 此参数的效果和同时指定 "-dpr" 参数相同;
- d: 当复制符号连接时, 把目标文件或目录也建立为符号连接, 并指向与源文件或目录连接的原始文件或目录;
- f: 强行复制文件或目录, 不论目标文件或目录是否已存在;
- i: 覆盖既有文件之前先询问用户; 输入 y 同意当前操作; 反之n
- l: 对源文件建立硬连接, 而非复制文件;
- p: 保留源文件或目录的属性;
- R/r: 递归处理, 将指定目录下的所有文件与子目录一并处理;
- s: 对源文件建立符号连接, 而非复制文件;
- u: 使用这项参数后只会在源文件的更改时间较目标文件更新时或是名称相互对应的目标文件并不存在时, 才复制文件;
- S: 在备份文件时, 用指定的后缀 "SUFFIX" 代替文件的默认后缀;
- b: 覆盖已存在的文件目标前将目标文件备份;
- v: 详细显示命令执行的操作。

例: 不同路径 A文件夹的 MP3 后缀的文件复制到B文件夹

```
prophet@001:~$ ls -al Music/Song/
total 8
drwxrwxr-x 2 prophet prophet 4096 7月 18 16:00 .
drwxr-xr-x 3 prophet prophet 4096 7月 18 15:57 ..
-rw-rw-r-- 1 prophet prophet 0 7月 18 16:00 1.mp3
-rw-rw-r-- 1 prophet prophet 0 7月 18 16:00 2.mp3
prophet@001:~$ ls -al Project/Song/
total 8
drwxrwxr-x 2 prophet prophet 4096 7月 18 15:59 .
drwxrwxr-x 3 prophet prophet 4096 7月 18 15:55 ..
-rw-rw-r-- 1 prophet prophet 0 7月 18 15:59 1.mp3
-rw-rw-r-- 1 prophet prophet 0 7月 18 15:59 2.mp3
-rw-rw-r-- 1 prophet prophet 0 7月 18 15:59 3.mp3
-rw-rw-r-- 1 prophet prophet 0 7月 18 15:59 4.mp3
-rw-rw-r-- 1 prophet prophet 0 7月 18 15:59 5.java
prophet@001:~$ cp -ri Project/Song/*.mp3 Music/Song
cp: overwrite 'Music/Song/1.mp3'? y
cp: overwrite 'Music/Song/2.mp3'? y
prophet@001:~$ ls -al Music/Song/
total 8
drwxrwxr-x 2 prophet prophet 4096 7月 18 16:05 .
drwxr-xr-x 3 prophet prophet 4096 7月 18 15:57 ..
-rw-rw-r-- 1 prophet prophet 0 7月 18 16:05 1.mp3
-rw-rw-r-- 1 prophet prophet 0 7月 18 16:05 2.mp3
-rw-rw-r-- 1 prophet prophet 0 7月 18 16:05 3.mp3
-rw-rw-r-- 1 prophet prophet 0 7月 18 16:05 4.mp3
prophet@001:~$
```

In 链接文件

链接文件分为:

- 符号链接 (symbolic links) 国内又叫: 软连接 (soft links)
软链接的方式则是产生一个文件的档案, 该文件的内容是指向另一个目录或文件的位置的位置。
- 硬链接 (hard links)
硬链接的意思是一个文件可以有多个名称

符号链接与硬链接的特性

特性

符号链接

- 以路径的形式存在。类似于Windows操作系统中的快捷方式
- 可以跨文件系统。（跨硬盘）
- 可以对一个不存在的文件名进行链接
- 可以对目录进行链接
- 在目录长列表中，符号链接作为一种特殊的文件类型显示出来，其第一个字母是l。
- 符号链接的大小是其链接文件的路径名中的字节数。
- 当用 `ls -l` 命令列出文件时，可以看到符号链接名后有一个箭头指向源文件或目录，例如
`lrwxrwxrwx ... 14 jun 20 10:20 /etc/motd->/original_file` 其中，表示“文件大小”的数字“14”恰好说明源文件名 `original_file` 由14个字符构成。

硬链接

- 以文件副本的形式存在。但不占用实际空间。
- 不允许给目录创建硬链接
- 硬链接只有在同一个文件系统中才能创建。（同硬盘）

总结

无论是符号链接还是硬链接，文件都保持同步变化。

创建符号链接需要 `-s`，硬链接不需要。

复制符号链接是没有意义的，因为你就是复制的链接文件。如同复制快捷方式。

符号链接文件不是一个独立的文件，它的许多属性依赖于源文件，所以给符号链接文件设置存取权限是没有意义的。

语法

```
ln [选项]... [-T] 目标 链接名 （第一种格式）
或: ln [选项]... 目标 （第二种格式）
或: ln [选项]... 目标... 目录 （第三种格式）
或: ln [选项]... -t 目录 目标... （第四种格式）
```

选项

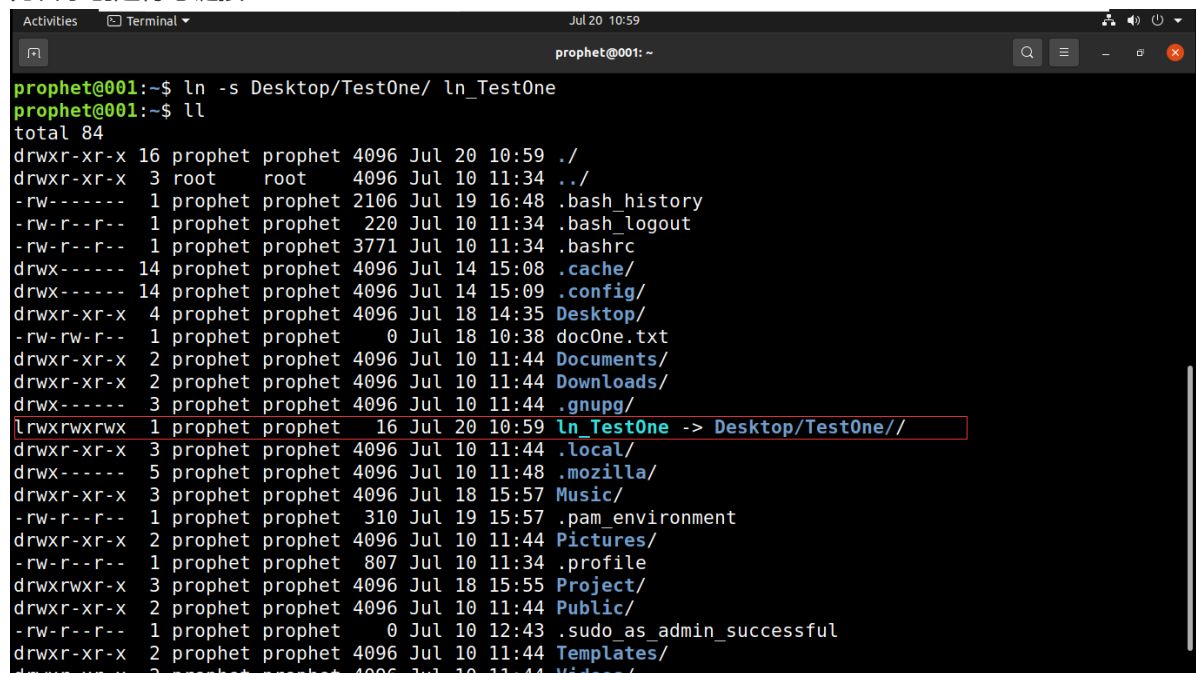
<code>--backup[=CONTROL]</code>	# 为每个已存在的目标文件创建备份文件
<code>-b</code>	# 类似 <code>--backup</code> ，但不接受任何参数
<code>-d, -F, --directory</code>	# 创建指向目录的硬链接(只适用于超级用户)
<code>-f, --force</code>	# 强行删除任何已存在的目标文件
<code>-i, --interactive</code>	# 覆盖既有文件之前先询问用户
<code>-L, --logical</code>	# 取消引用作为符号链接的目标
<code>-n, --no-dereference</code>	# 把符号链接的目的目录视为一般文件
<code>-P, --physical</code>	# 直接将硬链接到符号链接
<code>-r, --relative</code>	# 创建相对于链接位置的符号链接

```
-s, --symbolic          # 对源文件建立符号链接，而非硬链接
-S, --suffix=SUFFIX    # 用"-b"参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，预设的备份字符串是符号“~”，用户可通过“-S”参数来改变它
-t, --target-directory=DIRECTORY # 指定要在其中创建链接的DIRECTOR
-T, --no-target-directory # 将“LINK_NAME”视为常规文件
-v, --verbose          # 打印每个链接文件的名称
--help                # 显示此帮助信息并退出
--version              # 显示版本信息并退出

#最长用：
ln -s
```

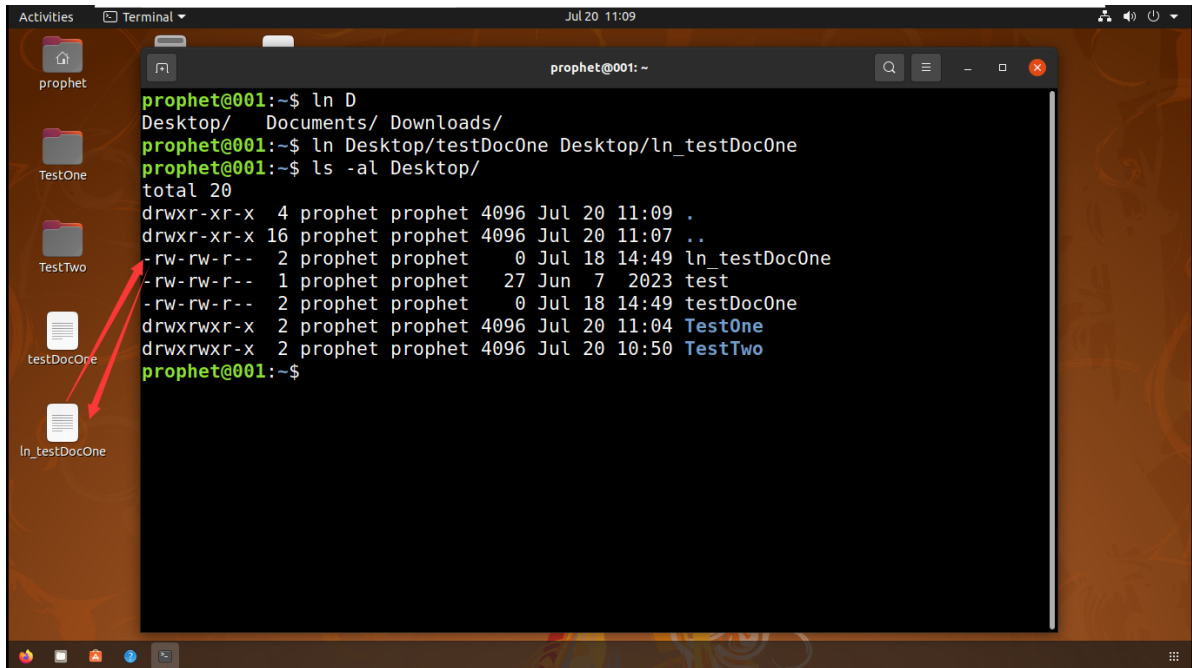
例：

为目录创建符号链接



```
Activities Terminal Jul 20 10:59 prophet@001: ~
prophet@001:~$ ln -s Desktop/TestOne/ ln_TestOne
prophet@001:~$ ll
total 84
drwxr-xr-x 16 prophet prophet 4096 Jul 20 10:59 ./
drwxr-xr-x  3 root     root     4096 Jul 10 11:34 ../
-rw-r--r--  1 prophet prophet 2106 Jul 19 16:48 .bash_history
-rw-r--r--  1 prophet prophet  220 Jul 10 11:34 .bash_logout
-rw-r--r--  1 prophet prophet 3771 Jul 10 11:34 .bashrc
drwx----- 14 prophet prophet 4096 Jul 14 15:08 .cache/
drwx----- 14 prophet prophet 4096 Jul 14 15:09 .config/
drwxr-xr-x  4 prophet prophet 4096 Jul 18 14:35 Desktop/
-rw-rw-r--  1 prophet prophet   0 Jul 18 10:38 docOne.txt
drwxr-xr-x  2 prophet prophet 4096 Jul 10 11:44 Documents/
drwxr-xr-x  2 prophet prophet 4096 Jul 10 11:44 Downloads/
drwx-----  3 prophet prophet 4096 Jul 10 11:44 .gnupg/
lrwxrwxrwx  1 prophet prophet  16 Jul 20 10:59 ln_TestOne -> Desktop/TestOne//
drwxr-xr-x  3 prophet prophet 4096 Jul 10 11:44 .local/
drwx-----  5 prophet prophet 4096 Jul 10 11:48 .mozilla/
drwxr-xr-x  3 prophet prophet 4096 Jul 18 15:57 Music/
-rw-r--r--  1 prophet prophet  310 Jul 19 15:57 .pam_environment
drwxr-xr-x  2 prophet prophet 4096 Jul 10 11:44 Pictures/
-rw-r--r--  1 prophet prophet  807 Jul 10 11:34 .profile
drwxrwxr-x  3 prophet prophet 4096 Jul 18 15:55 Project/
drwxr-xr-x  2 prophet prophet 4096 Jul 10 11:44 Public/
-rw-r--r--  1 prophet prophet   0 Jul 10 12:43 .sudo_as_admin_successful
drwxr-xr-x  2 prophet prophet 4096 Jul 10 11:44 Templates/
drwxr-xr-x  2 prophet prophet 4096 Jul 10 11:44 Videos/
```

为文件创建硬链接



mv 移动或重命名文件夹或文件

用来对文件或目录重新命名，或者将文件从一个目录移到另一个目录中。source表示源文件或目录，target表示目标文件或目录。如果将一个文件移到一个已经存在的目标文件中，则目标文件的内容将被覆盖。

语法

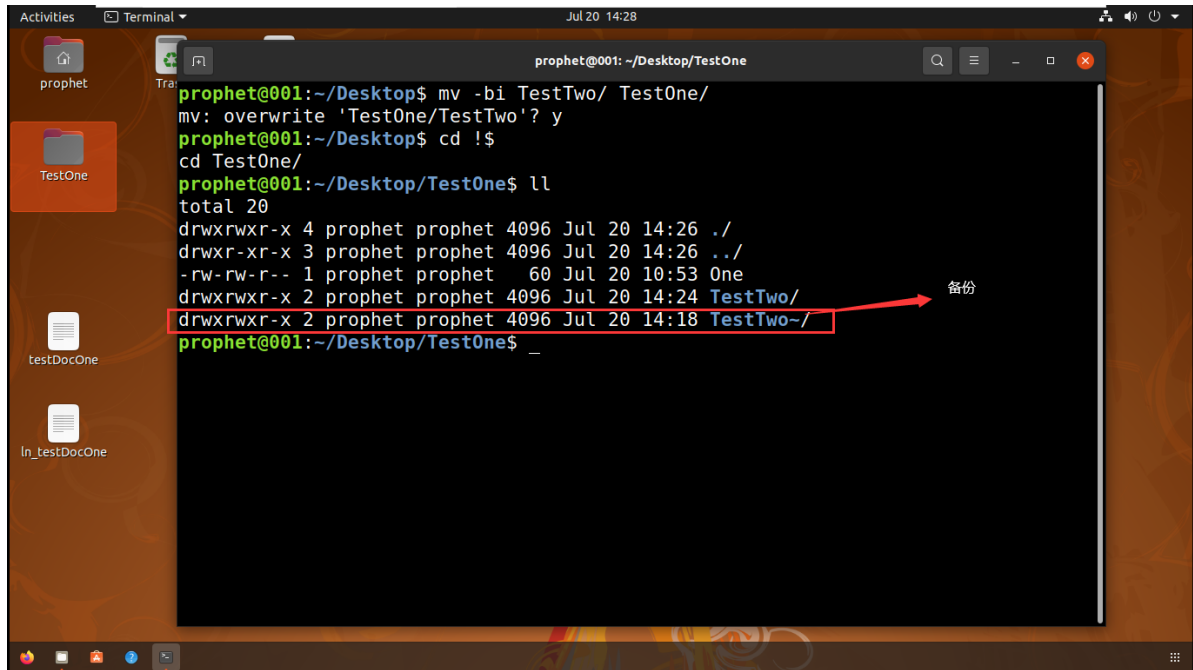
mv (选项) (参数)

选项

- backup=<备份模式>**: 若需覆盖文件，则覆盖前先行备份:
- b**: 当文件存在时，覆盖前，为其创建一个备份;
- f**: 若目标文件或目录与现有的文件或目录重复，则直接覆盖现有的文件或目录;
- i**: 交互式操作，覆盖前先行询问用户，如果源文件与目标文件或目标目录中的文件同名，则询问用户是否覆盖目标文件。用户输入“y”，表示将覆盖目标文件; 输入“n”，表示取消对源文件的移动。这样可以避免误将文件覆盖。
- strip-trailing-slashes**: 删除源文件中的斜杠“/”;
- S<后缀>**: 为备份文件指定后缀，而不使用默认的后缀;
- target-directory=<目录>**: 指定源文件要移动到目标目录;
- u**: 当源文件比目标文件新或者目标文件不存在时，才执行移动操作。

例：

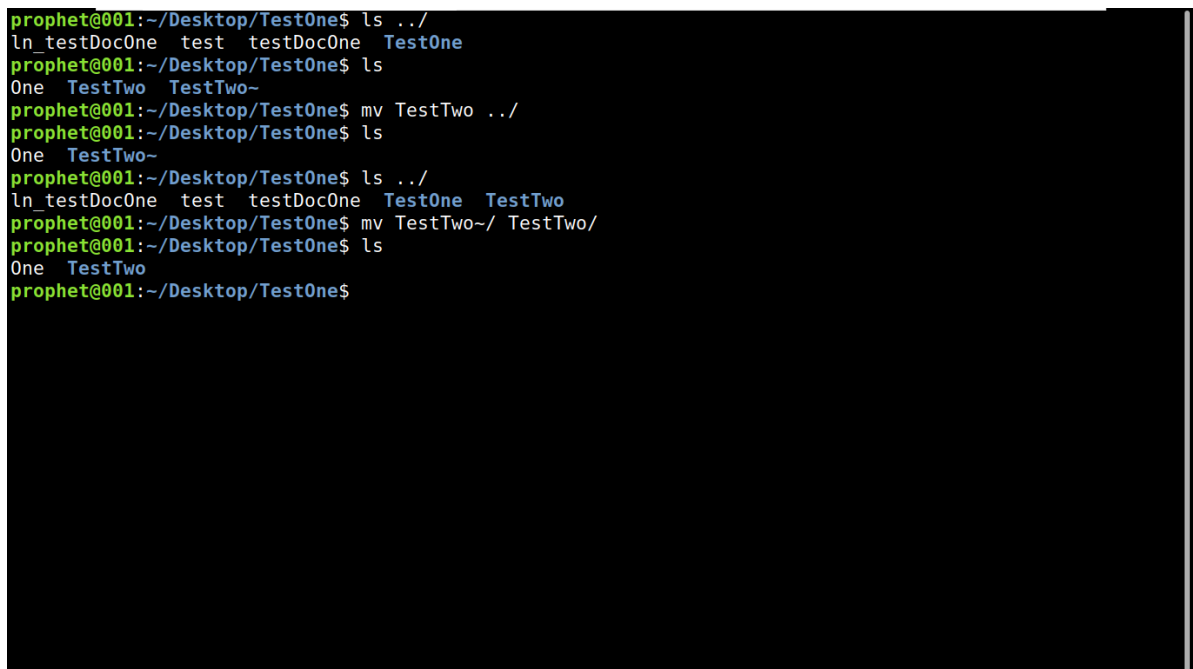
移动文件夹，名字与目标目录中的文件夹冲突，创建备份，并覆盖。



The screenshot shows a terminal window on a Linux desktop. The user is in the directory ~/Desktop/TestOne. They execute the command `mv -bi TestTwo/ TestOne/`. The terminal shows the prompt `mv: overwrite 'TestOne/TestTwo'? y`, followed by `cd !$` and `cd TestOne/`. Then, they run `ll` to list the contents of the current directory. The output shows a directory listing with permissions, owner, group, size, date, and file name. A red box highlights the entry `drwxrwxr-x 2 prophet prophet 4096 Jul 20 14:18 TestTwo~/`, with a red arrow pointing to it and the label "备份" (Backup) in Chinese. The terminal window title is `prophet@001: ~/Desktop/TestOne`.

```
prophet@001:~/Desktop$ mv -bi TestTwo/ TestOne/
mv: overwrite 'TestOne/TestTwo'? y
prophet@001:~/Desktop$ cd !$
cd TestOne/
prophet@001:~/Desktop/TestOne$ ll
total 20
drwxrwxr-x 4 prophet prophet 4096 Jul 20 14:26 ./
drwxr-xr-x 3 prophet prophet 4096 Jul 20 14:26 ../
-rw-rw-r-- 1 prophet prophet 60 Jul 20 10:53 One
drwxrwxr-x 2 prophet prophet 4096 Jul 20 14:24 TestTwo/
drwxrwxr-x 2 prophet prophet 4096 Jul 20 14:18 TestTwo~/
prophet@001:~/Desktop/TestOne$
```

移回去，并把备份文件夹改名



The screenshot shows a terminal window with the following commands and output:

```
prophet@001:~/Desktop/TestOne$ ls ../
ln_testDocOne test testDocOne TestOne
prophet@001:~/Desktop/TestOne$ ls
One TestTwo TestTwo~
prophet@001:~/Desktop/TestOne$ mv TestTwo ../
prophet@001:~/Desktop/TestOne$ ls
One TestTwo~
prophet@001:~/Desktop/TestOne$ ls ../
ln_testDocOne test testDocOne TestOne TestTwo
prophet@001:~/Desktop/TestOne$ mv TestTwo~/ TestTwo/
prophet@001:~/Desktop/TestOne$ ls
One TestTwo
prophet@001:~/Desktop/TestOne$
```

rm 用于删除给定的文件和目录

注意：使用rm命令要格外小心。因为一旦删除了一个文件，就无法再恢复它。所以，在删除文件之前，最好再看一下文件的内容，确定是否真要删除。rm命令可以用-i选项，这个选项在使用文件扩展名字符删除多个文件时特别有用。使用这个选项，系统会要求你逐一确定是否要删除。这时，必须输入y并按Enter键，才能删除文件。如果仅按Enter键或其他字符，文件不会被删除。

语法

```
rm (选项)(参数)
```

选项

- d**: 直接把欲删除的目录的硬连接数据删除成0，删除该目录；
- f**: 强制删除文件或目录；
- i**: 删除已有文件或目录之前先询问用户；
- r**或**-R**: 递归处理，将指定目录下的所有文件与子目录一并处理；
- preserve-root**: 不对根目录进行递归操作；
- v**: 显示指令的详细执行过程。

如果参数中含有目录，则必须加上 **-r** 或者 **-R** 选项。

删库跑路命令

```
sudo rm -rf /*
```

也可以使用通配符筛选后删除，尽量少用不安全啊。

mkdir 创建目录（文件夹）

该命令创建由dirname命名的目录。如果在目录名的前面没有加任何路径名，则在当前目录下创建由dirname指定的目录；如果给出了一个已经存在的路径，将会在该目录下创建一个指定的目录。

语法

```
mkdir (选项)(参数)
```

选项

- Z**: 设置安全上下文，当使用SELinux时有效；
- m**<目标属性>或**--mode**<目标属性>建立目录的同时设置目录的权限；
- p**或**--parents** 若所要建立目录的上层目录目前尚未建立，则会一并建立上层目录；
- version** 显示版本信息。

rmmdir 用来删除指定的空目录

利用rmmdir命令可以从一个目录中删除一个或多个空的子目录。该命令从一个目录中删除一个或多个子目录，其中dirname佬表示目录名。如果dirname中没有指定路径，则删除当前目录下由dirname指定的目录；如dirname中包含路径，则删除指定位置的目录。删除目录时，必须具有对其父目录的写权限。

注意：子目录被删除之前应该是空目录，就是说，该目录中的所有文件必须用rm命令全部，另外，当前工作目录必须在被删除目录之上，不能是被删除目录本身，也不能是被删除目录的子目录。

语法

```
rmmdir(选项)(参数)
```

选项

- p**或**--parents**: 删除指定目录后，若该目录的上层目录已变成空目录，则将其一并删除；
- ignore-fail-on-non-empty**: 此选项使rmmdir命令忽略由于删除非空目录时导致的错误信息；
- v**或**-verboes**: 显示命令的详细执行过程；
- help**: 显示命令的帮助信息；
- version**: 显示命令的版本信息。

file 用来探测给定文件的类型

file命令对文件的检查分为文件系统、魔法幻数检查和语言检查3个过程。

语法

```
file(选项)(参数)
```

文件：要确定类型的文件列表，多个文件之间使用空格分开，可以使用shell通配符匹配多个文件。

选项

- b: 列出辨识结果时，不显示文件名称；
- c: 详细显示指令执行过程，便于排错或分析程序执行的情形；
- f<名称文件>: 指定名称文件，其内容有一个或多个文件名称时，让file依序辨识这些文件，格式为每列一个文件名称；
- L: 直接显示符号连接所指向的文件类别；
- m<魔法数字文件>: 指定魔法数字文件；
- v: 显示版本信息；
- z: 尝试去解读压缩文件的内容。
- i: 显示MIME类别（媒体类型）。

查看简单的文件（test类型文件=Windows中的txt类型文件）

- cat 连接多个文件并打印到标准输出，适用于短文本
- more 显示文件内容，每次显示一，以全屏方式显示。b 上一页 space 下一页 q 退出
- less 分屏上下翻页浏览文件内容
- tail 在屏幕上显示指定文件的末尾若干行
- head 显示文件的开头部分

less 命令允许用户向前或向后浏览文件，而 more 命令只能向前浏览。

其实以上为基础命令，正经实操还是不常用的，或有更好替代方法，这里仅着重介绍。

cat 命令

语法

cat (选项) (文件名可多个)

选项

长选项与短选项等价

- | | |
|-----------------------|--------------------------|
| -A, --show-all | 等价于"-vET"组合选项。 |
| -b, --number-nonblank | 只对非空行编号，从1开始编号，覆盖"-n"选项。 |
| -e | 等价于"-vE"组合选项。 |
| -E, --show-ends | 在每行的结尾显示'\$'字符。 |
| -n, --number | 对所有行编号，从1开始编号。 |
| -s, --squeeze-blank | 压缩连续的空行到一行。 |
| -t | 等价于"-vT"组合选项。 |
| -T, --show-tabs | 使用"^I"表示TAB（制表符）。 |
| -u | POSIX兼容性选项，无意义。 |

`-v`, `--show-nonprinting` 使用"`^`"和"`M-`"符号显示控制字符，除了LFD（line feed，即换行符"`\n`"）和TAB（制表符）。

`--help` 显示帮助信息并退出。

`--version` 显示版本信息并退出。

less 命令

用less命令显示文件时，用PageUp键向上翻页，用PageDown键向下翻页。要退出less程序，应按Q键。

使用该命令时输入 `/` 可高亮查询字符

语法

`less(选项)(文件名)`

选项

- `-e`: 文件内容显示完毕后，自动退出；
- `-f`: 强制显示文件；
- `-g`: 不加亮显示搜索到的所有关键词，仅显示当前显示的关键词，以提高显示速度；
- `-l`: 搜索时忽略大小写的差异；
- `-N`: 每一行行首显示行号；
- `-s`: 将连续多个空行压缩成一行显示；
- `-S`: 在单行显示较长的内容，而不换行显示；
- `-x<数字>`: 将TAB字符显示为指定个数的空格字符。

tail 命令

- 默认在屏幕上显示指定文件的末尾10行。
- 处理多个文件时会在各个文件之前附加含有文件名的行。
- 如果没有指定文件或者文件名为`-`，则读取标准输入。
- 如果表示字节或行数的`NUM`值之前有一个`+`号，则从文件开头的第`NUM`项开始显示，而不是显示文件的最后`NUM`项。
- `NUM`

值后面可以有后缀：

- `b`: 512
- `kB`: 1000
- `k`: 1024
- `MB`: 1000 * 1000
- `M`: 1024 * 1024

- **GB** : 1000 * 1000 * 1000
- **G** : 1024 * 1024 * 1024
- **T**、**P**、**E**、**Z**、**Y** 等以此类推。

语法

tail (选项) (参数)

选项

-c, --bytes=NUM	输出文件尾部的NUM（NUM为整数）个字节内容。
-f, --follow[={name descript}]	显示文件最新追加的内容。“name”表示以文件名的方式监视文件的变化。
-F	与 “ --follow=name --retry ” 功能相同。
-n, --line=NUM	输出文件的尾部NUM（NUM位数字）行内容。
--pid=<进程号>	与“-f”选项连用，当指定的进程号的进程终止后，自动退出 tail 命令。
-q, --quiet, --silent	当有多个文件参数时，不输出各个文件名。
--retry	即是在 tail 命令启动时，文件不可访问或者文件稍后变得不可访问，都始终尝试打开文件。使用此选项时需要与选项“ --follow=name ”连用。
-s, --sleep-interval=<秒数>	与“-f”选项连用，指定监视文件变化时间间隔的秒数。
-v, --verbose	当有多个文件参数时，总是输出各个文件名。
--help	显示指令的帮助信息。
--version	显示指令的版本信息。

head 命令

- 在未指定行数时默认显示前10行。
- 处理多个文件时会在各个文件之前附加含有文件名的行。
- 当没有文件或文件为 - 时，读取标准输入。

-c, --bytes=[-]NUM	显示前NUM字节；如果NUM前有“-”，那么会打印除了文件末尾的NUM字节以外的其他内容。
-n, --lines=[-]NUM	显示前NUM行而不是默认的10行；如果NUM前有“-”，那么会打印除了文件末尾的NUM行以外的其他行。
-q, --quiet, --silent	不打印文件名行。
-v, --verbose	总是打印文件名行。
-z, --zero-terminated	行终止符为NUL而不是换行符。
--help	显示帮助信息并退出。
--version	显示版本信息并退出。

history 显示或操作历史列表。

选项

- c** 清空历史列表。
- d offset** 根据offset删除记录。如果是正数则表示offset位置的记录，如果为负数则表示从结尾向前offset位置的记录。
- a** 将当前终端的历史记录行添加到历史记录文件。
- n** 将尚未从历史文件中读取的历史行追加到当前历史列表中。
- r** 读取历史文件，并将其内容附加到历史列表中。
- w** 将当前历史记录列表附加到历史记录文件中并且附加它们到历史列表中。
- p** 在每个arg上执行历史记录扩展并在标准输出上显示结果，而不将结果存储在历史记录列表中。
- s** 将每个arg作为单个条目附加到历史记录列表。

配合 history 命令使用的符号命令

```
!! #上一次命令
!num #执行 history 命令中的第几条命令(num是数字)
!^ #上一次命令的第一个参数
!$ #上一次命令的最后一个参数
!* #上一次命令的全部参数
!: #命令本身
!:3 #指定第3个参数
!:2-5 #指定2到5个参数
!:-3 #指定0到3个参数
!:2* #指定2到最后一个参数
!:2- #指定2到最后二个参数
!:s/xx/yy #将上次命令中的xx替换为yy
```