

1. ls 列出当前目录
2. pwd 打印当前工作目录的完整路径名
3. touch 将每个文件的访问和修改时间更新为当前时间
4. cat 将FILE或标准输入连接到标准输出

### 使用 cat, tac 和 nl 命令查看文件

前两个命令都是用来打印文件内容到标准输出（终端），其中 cat 为正序显示，tac 为倒序显示。

标准输入输出：当我们执行一个 shell 命令行时通常会自动打开三个标准文件，即标准输入文件（stdin），默认对应终端的键盘、标准输出文件（stdout）和标准错误输出文件（stderr），后两个文件都对应被重定向到终端的屏幕，以便我们能直接看到输出内容。进程将从标准输入文件中得到输入数据，将正常输出数据输出到标准输出文件，而将错误信息送到标准错误文件中。

比如我们要查看之前从 /etc 目录下拷贝来的 passwd 文件：

```
$ cd /home/shiyanlou
$ cp /etc/passwd passwd
$ cat passwd
```

可以加上 -n 参数显示行号：

```
$ cat -n passwd
```

5. mkdir 如果目录不存在,则创建目录

### 新建目录

使用 mkdir (make directories) 命令可以创建一个空目录，也可同时指定创建目录的权限属性。

创建名为“mydir”的空目录：

```
$ mkdir mydir
```

使用 -p 参数，同时创建父目录（如果不存在该父目录），如下我们同时创建一个多级目录（这在安装软件、配置安装路径时非常有用）：

```
$ mkdir -p father/son/grandson
```

6. cd 切换当前目录至指定目录

## 删除文件

使用 `rm` (remove files or directories) 命令删除一个文件:

```
$ rm test
```

有时候你会遇到想要删除一些为只读权限的文件, 直接使用 `rm` 删除会显示一个提示, 如下:

```
shiyanolou:~/ $ touch test [17:19:21]
shiyanolou:~/ $ chmod 444 test [17:19:28]
shiyanolou:~/ $ ll test [17:19:34]
-r--r--r-- 1 shiyanolou shiyanolou 0 7月 16 17:19 test
shiyanolou:~/ $ rm test [17:19:37]
rm: 是否删除有写保护的普通空文件 'test'? n
shiyanolou:~/ $ rm -f test [17:19:44]
shiyanolou:~/ $ [17:19:53]
```

你如果想忽略这提示, 直接删除文件, 可以使用 `-f` 参数强制删除:

```
$ rm -f test
```

## 删除目录

跟复制目录一样, 要删除一个目录, 也需要加上 `-r` 或 `-R` 参数:

```
$ rm -r family
```

7. rm 删除

8. copy 将文件1中的内容复制到文件2中,文件本身的名称不会更改

9. which 在PATH变量指定的路径中,搜索某个系统命令的位置,并且返回第一个搜索结果.

## 复制文件

使用 `cp` (copy) 命令复制一个文件到指定目录。

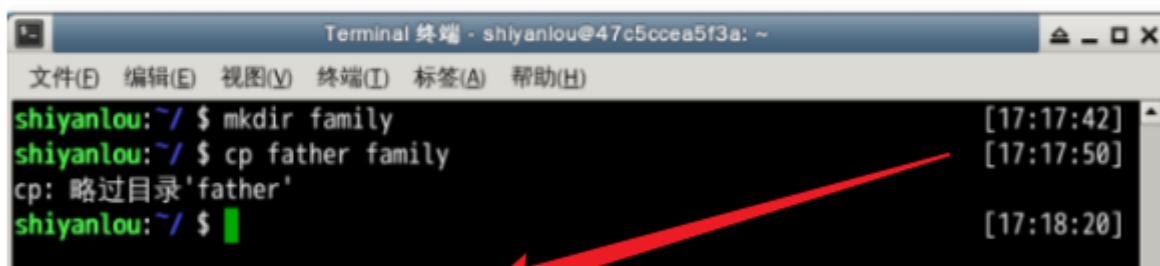
将之前创建的 "test" 文件复制到 "/home/shiyanlou/father/son/grandson" 目录中：

```
$ cp test father/son/grandson
```

是不是很方便啊，如果在图形界面则需要先在源目录复制文件，再进到目的目录粘贴文件，而命令行操作步骤就一步到位了嘛。

## 复制目录

如果直接使用 `cp` 命令复制一个目录的话，会出现如下错误：

A terminal window titled "Terminal 终端 - shiyanlou@47c5ccea5f3a: ~" with a menu bar (文件(E), 编辑(E), 视图(V), 终端(T), 标签(A), 帮助(H)). The terminal shows the following commands and output:

```
shiyanlou:~/ $ mkdir family [17:17:42]
shiyanlou:~/ $ cp father family [17:17:50]
cp: 略过目录 'father'
shiyanlou:~/ $ [17:18:20]
```

A red arrow points from the text "会出现如下错误：" to the error message "cp: 略过目录 'father'".

要成功复制目录需要加上 `-r` 或者 `-R` 参数，表示递归复制，就是说有点“株连九族”的意思：

```
$ cd /home/shiyanlou
$ mkdir family
$ cp -r father family
```

10. cp

## 移动文件

使用 `mv` (move or rename files) 命令移动文件 (剪切) 。将文件“ file1 ”移动到 `Documents` 目录:

`mv` 源目录文件 目的目录:

```
$ mkdir Documents
$ touch file1
$ mv file1 Documents
```

```
shiyanolou:~ $ mkdir Documents [17:21:49]
shiyanolou:~ $ touch file1 [17:21:51]
shiyanolou:~ $ mv file1 Documents [17:22:01]
shiyanolou:~ $ cd Documents [17:22:08]
shiyanolou:Documents/ $ ls [17:22:12]
file1
shiyanolou:Documents/ $ [17:22:13]
```

## 重命名文件

将文件“ file1 ”重命名为“ myfile ”:

`mv` 旧的文件名 新的文件名:

```
$ mv file1 myfile
```

## 批量重命名

11. mv

### 批量重命名

要实现批量重命名, `mv` 命令就有点力不从心了, 我们可以使用一个看起来更专业的命令 `rename` 来实现。不过它要用 `perl` 正则表达式来作为参数, 关于正则表达式我们要在后面才会介绍到, 这里只做演示, 你只要记得这个 `rename` 命令可以批量重命名就好了, 以后再重新学习也不会有任何问题, 毕竟你已经掌握了一个更常用的 `mv` 命令。

```
$ cd /home/shiyanolou/

# 使用通配符批量创建 5 个文件:
$ touch file{1..5}.txt

# 批量将这 5 个后缀为 .txt 的文本文件重命名为以 .c 为后缀的文件:
$ rename 's/\.txt/\.c/' *.txt

# 批量将这 5 个文件, 文件名和后缀改为大写:
$ rename 'y/a-z/A-Z/' *.c
```

简单解释一下上面的命令，`rename` 是先使用第二个参数的通配符匹配所有后缀为 `.txt` 的文件，然后使用第一个参数提供的正则表达式将匹配的这些文件的 `.txt` 后缀替换为 `.c`，这一点在我们后面学习了 `sed` 命令后，相信你会更好地理解。

```
11.7.6.10 /etc/c
```

`nl` 命令，添加行号并打印，这是个比 `cat -n` 更专业的行号打印命令。

这里简单列举它的常用的几个参数：

```
-b : 指定添加行号的方式，主要有两种：
    -b a: 表示无论是否为空行，同样列出行号 ("cat -n" 就是这种方式)
    -b t: 只列出非空行的编号并列出 (默认为这种方式)
-n : 设置行号的样式，主要有三种：
    -n ln: 在行号字段最左端显示
    -n rn: 在行号字段最右边显示，且不加 0
    -n rz: 在行号字段最右边显示，且加 0
-w : 行号字段占用的位数 (默认为 6 位)
```

12. nl

## 使用 `more` 和 `less` 命令分页查看文件

如果说上面的 `cat` 是用来快速查看一个文件的内容的，那么这个 `more` 和 `less` 就是天生用来“阅读”一个文件的内容的，比如说 `man` 手册内部就是使用的 `less` 来显示内容。其中 `more` 命令比较简单，只能向一个方向滚动，而 `less` 为基于 `more` 和 `vi`（一个强大的编辑器，我们有单独的课程来让你学习）开发，功能更强大。`less` 的使用基本和 `more` 一致，具体使用请查看 `man` 手册，这里只介绍 `more` 命令的使用。

使用 `more` 命令打开 `passwd` 文件：

```
$ more passwd
```

### 13. more less

使用 `more` 命令打开 `passwd` 文件：

```
$ more passwd
```

```
shiyanolou:~/ $ more passwd [17]
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbi
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:
se
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/net
false
--更多--(56%)
```

打开后默认只显示一屏内容，终端底部显示当前阅读的进度。可以使用 `Enter` 键向下滚动一行，使用 `Space` 键向下滚动一屏，按下 `h` 显示帮助，`q` 退出。

## 14. head tail

**使用 head 和 tail 命令查看文件**

这两个命令，那些性子比较急的人应该会喜欢，因为它们一个是只查看文件的头几行（默认为 10 行，不足 10 行则显示全部）和尾几行。还是拿 passwd 文件举例，比如当我们想要查看最近新增加的用户，那么我们可以查看这个 `/etc/passwd` 文件，不过我们前面也看到了，这个文件里面一大堆乱糟糟的东西，看起来实在费神啊。因为系统新增加一个用户，会将用户的信息添加到 passwd 文件的最后，那么这时候我们就可以使用 `tail` 命令了：

```
$ tail /etc/passwd
```

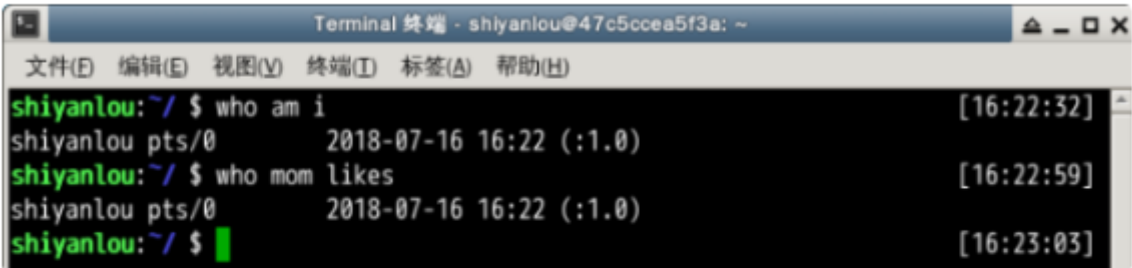
甚至更直接的只看一行，加上 `-n` 参数，后面紧跟行数：

```
$ tail -n 1 /etc/passwd
```

```
$ who am i

或者

$ who mom likes
```



输出的第一列表示打开当前伪终端的用户的用户名（要查看当前登录用户的用户名，去掉空格直接使用 `whoami` 即可），第二列的 `pts/0` 中 `pts` 表示伪终端，所谓伪是相对于 `/dev/tty` 设备而言的，还记得上一节讲终端时的那七个使用 `[Ctrl]+[Alt]+[F1]~[F7]` 进行切换的 `/dev/tty` 设备么,这是“真终端”，伪终端就是当你在图形用户界面使用 `/dev/tty7` 时每打开一个终端就会产生一个伪终端，`pts/0` 后面那个数字就表示打开的伪终端序号，你可以尝试再打开一个终端，然后在里面输入 `who am i`，看第二列是不是就变成 `pts/1` 了，第三列则表示当前伪终端的启动时间。

who 命令其它常用参数

参数	说明
-a	打印能打印的全部
-d	打印死掉的进程
-m	同 <code>am i,mom likes</code>
-q	打印当前登录用户数及用户名
-u	打印当前登录用户登录信息
-r	打印运行等级

15. who