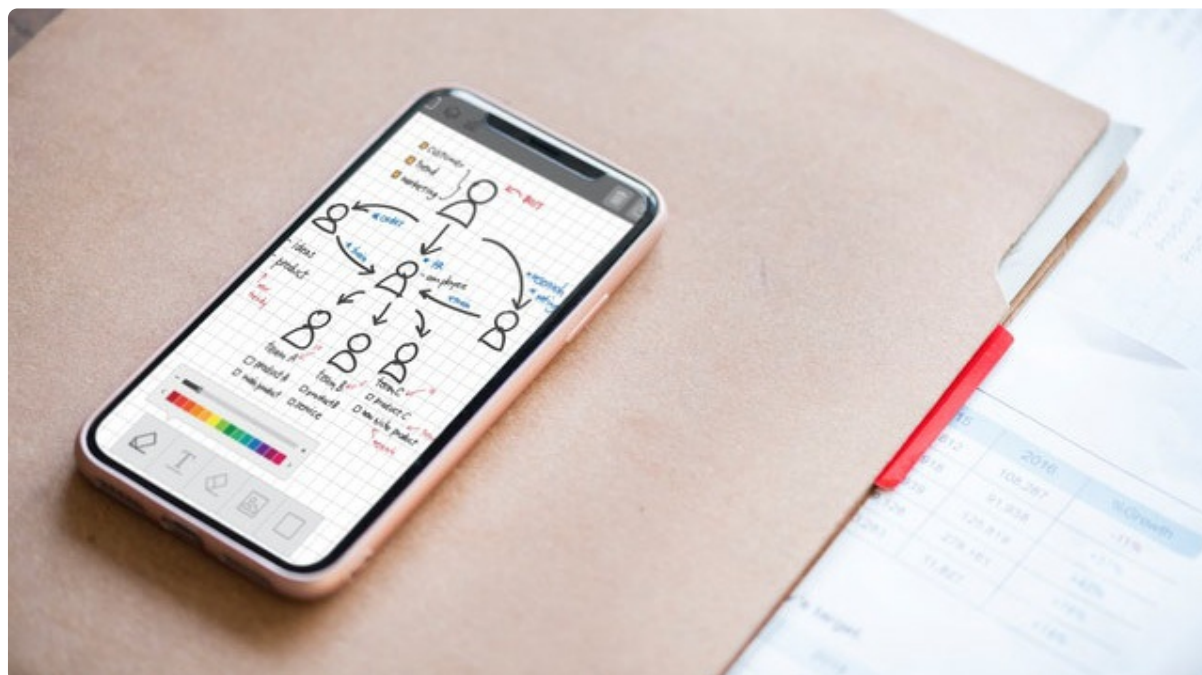


## 12 文件操纵，鼓掌之中。浏览和创建文件

更新时间：2019-07-01 14:02:47



受苦的人，没有悲观的权利。

更多一手资源+V : Andyqc1  
qq:3118617541

——尼采

aa: 31 18617541

## 内容简介

1. 前言
2. cat 命令和 less 命令：显示文件内容
3. head 命令和 tail 命令：显示文件的开头和结尾
4. touch 命令和 mkdir 命令：创建文件和目录
5. 总结

## 1. 前言

既然上两课文件组织、pwd和which命令。组织不会亏待你和目录相关的命令 ls、cd 和 du中，我们学习了 Linux 中文件的组织方式，那么现在就该是玩弄，啊不，是操纵它们的时候了。

文件操作有哪些呢？一般就是显示文件内容，还有拷贝文件、移动文件、删除文件等等。

下一课中，我们还将学习链接的知识，认识硬链接和软链接的区别和原理。

当然了，虽说是文件操作，但是目录操作我们也会讲。毕竟在 Linux 中，一切皆文件，目录也是文件。

这一课和下一课有很重要的基础概念，而且要学不少命令，都是很常用的。

希望大家认真学习，但也不要怕。我们的课程会秉持浅显易懂的原则，带大家领略 Linux 的无穷魅力。

## 2. cat 命令和 less 命令：显示文件内容

首先，我们来学习一下如何显示文件的内容。在 Linux 中，有好几个命令都可以显示文件内容，不过最常用的还是：

- cat
- less

这两个命令只能显示文件内容，并不能对文件内容进行修改。之后的课程我们会学习如何编辑文件的。

也许你会问：“为什么要学习两个命令呢？既然这两个命令都能显示文件内容，我学一个不就好了？”

事实上，这两个命令各有特色，我们分别来学习。

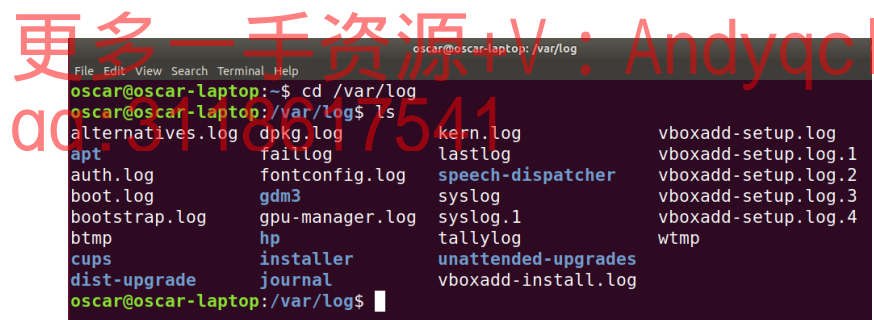
在开始学习今天的这些命令之前，我们来介绍一个我们将要操作的文件：syslog。

syslog 这个文件位于 /var/log 这个目录中。还记得之前我们讲的么，/var 这个目录通常包含程序的数据，比如一些 log（日志）文件，记录电脑中发生了什么事。

我们用 cd 命令来进入 /var/log，再用 ls 命令看看里面有什么：

```
cd /var/log
ls
```

如下图所示：



可以看到好多种不同颜色的文件，记得我们之前说的吗？Ubuntu 的终端会默认为不同类型的文件标记颜色：

- 蓝色→目录
- 绿色→可执行文件
- 红色→压缩文件
- 浅蓝色→链接文件
- 灰色→其他文件

我们看到 syslog 这个文件了，它是一个普通文件。不过这个文件很有用，因为它是当下你电脑里发生的几乎所有事情的日志。

其实从名字我们也可以窥见一二，syslog 由 sys 和 log 组成，sys 是 system 的缩写，是英语“系统”的意思；log 就是 log，是英语“日志”的意思。所以 syslog 就是记录系统日志的文件。

你可能注意到了，我们在 Linux 中看到的文件，很多都是没有后缀名的。如果习惯 Windows 的用户可能不太适应，因为 Windows 中一般这类文件会像这样命名：syslog.txt，syslog.log 等等。

即使 Linux 下很多文件没有后缀名，也不影响它们的显示和编辑，和其它文件是一样的。

## cat 命令：一次性显示文件的所有内容

cat 命令可以一次性在终端中显示文件的所有内容。cat 可不是英语“猫”的意思，它其实是 concatenate 的缩写，表示“链接，连接”。

cat 命令的描述是“concatenate files and print on the standard output”，意思是“把文件连接起来，一并打印在标准输出”。

cat 命令的用法也很简单，只需要在命令后加上想要显示的文件路径即可。例如：

```
cd /var/log
cat syslog
```

```
oscar@oscar-laptop:~$ cd /var/log
oscar@oscar-laptop:/var/log$ cat syslog
May  2 18:38:12 oscar-laptop rsyslogd: [origin software="rsyslogd" swVersion="8
.32.0" x-pid="589" x-info="http://www.rsyslog.com"] rsyslogd was HUPed
May  2 18:38:16 oscar-laptop anacron[446]: Job `cron.daily' terminated
May  2 18:38:16 oscar-laptop anacron[446]: Normal exit (1 job run)
May  2 18:48:14 oscar-laptop systemd[1]: Starting Cleanup of Temporary Directori
es...
May  2 18:48:14 oscar-laptop systemd[1]: Started Cleanup of Temporary Directorie
s.
May  2 18:50:07 oscar-laptop systemd[1]: Starting Message of the Day...
May  2 18:50:07 oscar-laptop systemd[1]: Started Message of the Day.
May  2 19:01:57 oscar-laptop systemd[1]: Started Run anacron jobs.
May  2 19:01:57 oscar-laptop anacron[2338]: Anacron 2.3 started on 2019-05-02
May  2 19:01:57 oscar-laptop anacron[2338]: Normal exit (0 jobs run)
```

syslog 文件的内容很多，一页根本显示不下，我们的截图只能看到一部分内容，其实下面还有很多。

不过，cat 命令的初衷就是如此：显示文件的所有内容。不过我们也意识到了，cat 命令更适合处理小文件。

cat 命令没有什么参数比较值得学习。因为这个命令比较简单，如果你想要在显示的文件内容上加上行号，那么你可以加上 -n 这个参数：

```
cat -n syslog
```

当然了，cat 命令后面可以接不止一个文件，比如你可以连接两个文件的内容，将其一并输出：

```
cat syslog dpkg.log
```

## less 命令：分页显示文件内容

cat 命令会把文件所有内容一下子显示出来，如果文件比较大，是会令人“猝不及防”的。

这种时候，另外的一个命令 less 就显得必不可少。less 命令和 cat 命令之间最大的区别就是：less 命令会分页一页一页地显示文件内容，使我们可以方便地在终端里阅读。

当然了，其实还有个命令是 more。不过这个 more 命令和 less 命令的功能是类似的，而且 more 的缺陷就是它没有 less 那么强大，比如 more 命令虽然也是一页一页地显示文件，但是我们不能往后翻页，只能往前，“一路向北”。

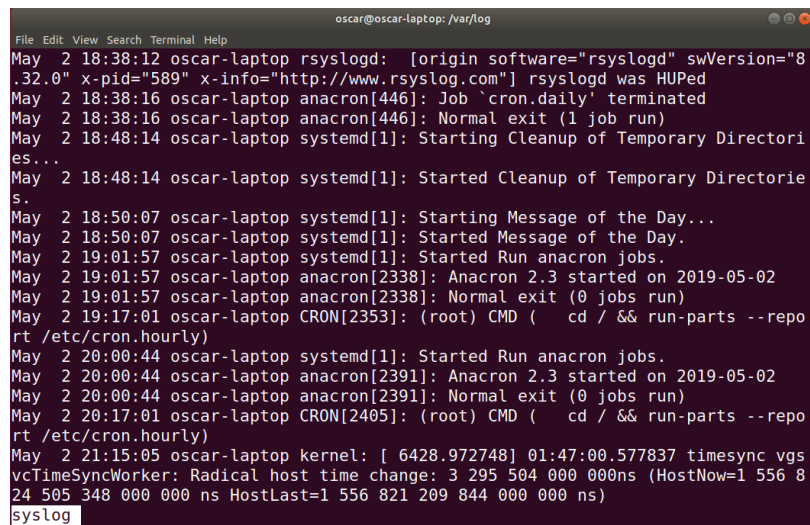
而且 less 命令还有更多其它功能是 more 没有的，究其原因呢就是 more 是比较老的命令了。这年头毕竟是吃“青春饭”的，more 命令只能哀叹：“怪我生得太早啊”...

其实 more 是英语“更多”的意思，less 是英语“更少”的意思。这一对，果然是“或多或少”有点缘分。

less 命令的用法与 cat 命令是一样的，也是在命令之后直接加文件的路径：

```
less syslog
```

显示如下：



```
oscar@oscar-laptop: /var/log
File Edit View Search Terminal Help
May  2 18:38:12 oscar-laptop rsyslogd: [origin software="rsyslogd" swVersion="8
.32.0" x-pid="589" x-info="http://www.rsyslog.com"] rsyslogd was HUPed
May  2 18:38:16 oscar-laptop anacron[446]: Job `cron.daily' terminated
May  2 18:38:16 oscar-laptop anacron[446]: Normal exit (1 job run)
May  2 18:48:14 oscar-laptop systemd[1]: Starting Cleanup of Temporary Directori
es...
May  2 18:48:14 oscar-laptop systemd[1]: Started Cleanup of Temporary Directorie
s.
May  2 18:50:07 oscar-laptop systemd[1]: Starting Message of the Day...
May  2 18:50:07 oscar-laptop systemd[1]: Started Message of the Day.
May  2 19:01:57 oscar-laptop systemd[1]: Started Run anacron jobs.
May  2 19:01:57 oscar-laptop anacron[2338]: Anacron 2.3 started on 2019-05-02
May  2 19:01:57 oscar-laptop anacron[2338]: Normal exit (0 jobs run)
May  2 19:17:01 oscar-laptop CRON[2353]: (root) CMD ( cd / && run-parts --repo
rt /etc/cron.hourly)
May  2 20:00:44 oscar-laptop systemd[1]: Started Run anacron jobs.
May  2 20:00:44 oscar-laptop anacron[2391]: Anacron 2.3 started on 2019-05-02
May  2 20:00:44 oscar-laptop anacron[2391]: Normal exit (0 jobs run)
May  2 20:17:01 oscar-laptop CRON[2405]: (root) CMD ( cd / && run-parts --repo
rt /etc/cron.hourly)
May  2 21:15:05 oscar-laptop kernel: [ 6428.972748] 01:47:00.577837 timesync vgs
vcTimeSyncWorker: Radical host time change: 3 295 504 000 ns (HostNow=1 556 8
24 505 348 000 ns HostLast=1 556 821 209 844 000 000 ns)
syslog
```

less 命令的好处是它会先读入文件开始的若干行，然后就停在那里，而这若干行的行数取决于终端屏幕的大小。这样的好处是我们可以有时间去渐进读文件的内容。

所以，暂时说来，我们只读取了文件的头几行。

那怎么读取文件后面的内容呢？

让我们来学习一些有用的键盘快捷键吧。

首先来看 less 命令中最基本最常用的快捷键：

- 空格键：文件内容读取下一个终端屏幕的行数，相当于前进一个屏幕（页）。很常用的快捷键。与键盘上的 PageDown（下一页）效果一样；
- 回车键：文件内容读取下一行，也就是前进一行，与键盘上的向下键效果是一样的；
- d 键：前进半页（半个屏幕）；
- b 键：后退一页，与键盘上的 PageUp（上一页）效果一样；
- y 键：后退一行，与键盘上的向上键效果是一样的；
- u 键：后退半页（半个屏幕）；
- q 键：停止读取文件，中止 less 命令。

这里所说的快捷键当中提到字母时，大小写是需要区分的。

比如我们说到 d 键，那么就是小写字母 d，而不是大写字母 D（shift + d）。

在 Linux 中，通常来说都区分大小写。

再来看看一些略显高级的快捷键，虽然不像上面的几个那么常用，但学习一下也真真是极好的：

- = 号：显示你在文件中的什么位置（会显示当前页面的内容是文件中第几行到第几行，整个文件所含行数，所含字符数，整个文件所含字符）；
- h 键：显示帮助文档，按 q 键退出帮助文档；
- /（斜杠）：进入搜索模式，只要在斜杠后面输入你要搜索的文字，按下回车键，就会把所有符合的结果都标识出来。要在搜索所得结果中跳转，可以按 n 键（跳到下一个符合项目），N 键（shift 键 + n。跳到上一个符合项

目)。当然了，正则表达式（Regular Expression）也是可以用在搜索内容中的。这里我们就不细说什么是正则表达式了，有兴趣可以 百度 / Google 看看；

- n 键：跳到下一个符合的搜索结果；
- N 键：跳到上一个符合的搜索结果。

如你所见，less 命令很强大，可以帮助我们更好地浏览文件内容。

虽然一开始要记着这些快捷键有点累人，但是熟练之后绝对让你事半功倍。

### 3. head 命令和 tail 命令：显示文件的开头和结尾

额，除了 cat 和 less，读取文件内容还有其它命令要学啊...“臣妾做不到啊...”。

但是这两个新命令可是很有用的，特别是 tail。所以请放下“皇后”的架子，来学一下吧。

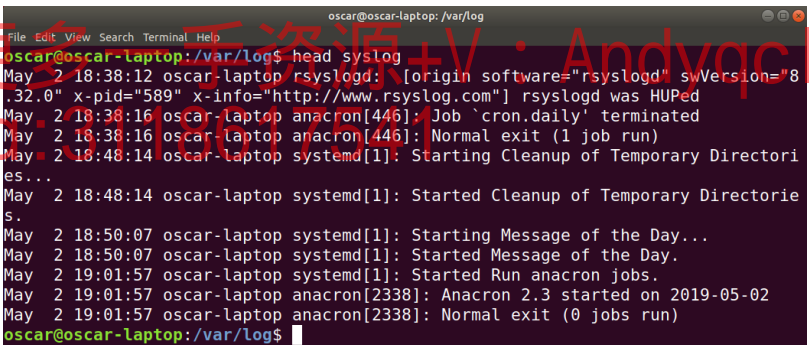
这两个命令有点对立的关系，因为 head 命令用于显示文件的开头几行，tail 命令用于显示文件结尾的几行。

#### head 命令：显示文件开头

head 在英语中是“头部”的意思，所以顾名思义，这个命令用于显示文件的开头。

默认情况下，head 会显示文件的头 10 行：

```
head syslog
```

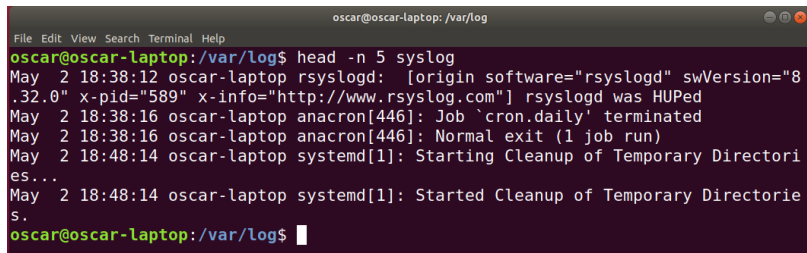


```
oscar@oscar-laptop: /var/log
File Edit View Search Terminal Help
oscar@oscar-laptop: /var/log$ head syslog
May  2 18:38:12 oscar-laptop rsyslogd: [origin software="rsyslogd" swVersion="8
.32.0" x-pid="589" x-info="http://www.rsyslog.com"] rsyslogd was HUPed
May  2 18:38:16 oscar-laptop anacron[446]: Job `cron.daily' terminated
May  2 18:38:16 oscar-laptop anacron[446]: Normal exit (1 job run)
May  2 18:48:14 oscar-laptop systemd[1]: Starting Cleanup of Temporary Directori
es...
May  2 18:48:14 oscar-laptop systemd[1]: Started Cleanup of Temporary Directorie
s.
May  2 18:50:07 oscar-laptop systemd[1]: Starting Message of the Day...
May  2 18:50:07 oscar-laptop systemd[1]: Started Message of the Day.
May  2 19:01:57 oscar-laptop systemd[1]: Started Run anacron jobs.
May  2 19:01:57 oscar-laptop anacron[2338]: Anacron 2.3 started on 2019-05-02
May  2 19:01:57 oscar-laptop anacron[2338]: Normal exit (0 jobs run)
oscar@oscar-laptop: /var/log$
```

我们也可以指定显示的行数，用 -n 这个参数：

```
head -n 5 syslog
```

就会显示文件的头 5 行。



```
oscar@oscar-laptop: /var/log
File Edit View Search Terminal Help
oscar@oscar-laptop: /var/log$ head -n 5 syslog
May  2 18:38:12 oscar-laptop rsyslogd: [origin software="rsyslogd" swVersion="8
.32.0" x-pid="589" x-info="http://www.rsyslog.com"] rsyslogd was HUPed
May  2 18:38:16 oscar-laptop anacron[446]: Job `cron.daily' terminated
May  2 18:38:16 oscar-laptop anacron[446]: Normal exit (1 job run)
May  2 18:48:14 oscar-laptop systemd[1]: Starting Cleanup of Temporary Directori
es...
May  2 18:48:14 oscar-laptop systemd[1]: Started Cleanup of Temporary Directorie
s.
oscar@oscar-laptop: /var/log$
```

#### tail 命令：显示文件结尾

tail 在英语中是“尾部”的意思，所以顾名思义，这个命令用于显示文件的结尾。

默认情况下，tail 会显示文件的尾 10 行。



我们也可以指定显示的行数，用 `-n` 这个参数：

```
tail -n 5 syslog
```

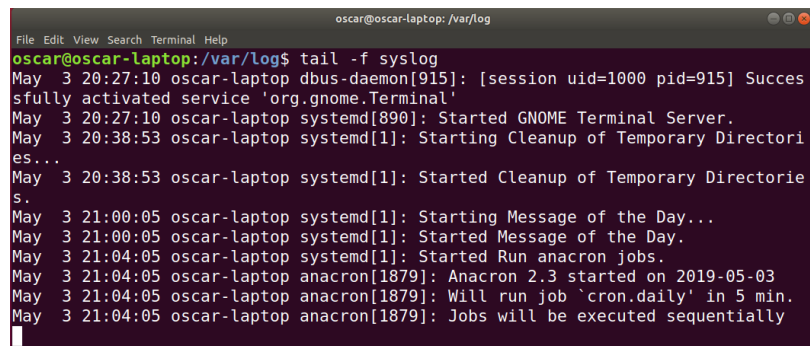
就会显示文件的尾 5 行。

如果你认为 `tail` 命令就这点本事，那您可就错啦...

`tail` 命令还可以配合 `-f` 参数来实时追踪文件的更新：

```
tail -f syslog
```

这样，就会检查文件是否有追加内容，如果有，就显示新增内容：

A terminal window titled 'oscar@oscar-laptop: /var/log' showing the output of the command 'tail -f syslog'. The output displays system logs in real-time, including messages from 'dbus-daemon[915]', 'systemd[890]', and 'anacron[1879]'. The logs show the successful activation of the GNOME Terminal service, the starting of the GNOME Terminal Server, the cleanup of temporary directories, the starting of the Message of the Day, and the starting of anacron jobs. The logs are timestamped with the date 'May 3' and the time '20:27:10' and '20:38:53'.

可以用快捷键 `Ctrl + c` 来终止 `tail -f` 命令。

事实上，`Ctrl + c` 这个组合快捷键（同时按下）可以终止大部分终端的命令和正在执行的程序，有点类似 Windows 中的 `Alt + F4`。

默认地，`tail -f` 会每过 1 秒检查一下文件是否有新内容。你也可以指定间隔检查的秒数，用 `-s` 参数：

```
tail -f -s 4 syslog #每隔 4 秒检查一次文件是否有更新
```

```
tail -f -s 2.5 syslog #每隔 2.5 秒检查一次文件是否有更新
```

## 4. touch 命令和 mkdir 命令：创建文件和目录

长舒一口气吗？终于结束了如何显示文件的学习。

我们再来看看其它文件操作。

### touch 命令：创建一个空白文件

事实上，Linux 没有一个命令是专门用来创建一个空白文件的。

平时我们习惯用文本编辑器（Emacs、Vim、Gedit、Sublime Text、Atom、Visual Studio Code、nano 等）来创建空白文件，并且编辑文件。

`touch` 命令其实一开始的设计初衷是修改文件的时间戳，就是可以修改文件的创建时间或修改时间，让电脑以为文件是在那个时候被修改或创建的。

touch 在英语中是“触摸，触碰”的意思，就是说“我碰一下那个文件，让电脑以为我刚修改过”。

但是，为什么把 touch 命令放在这里来学习呢？因为 touch 命令如果后面跟着的文件名是不存在的，那么它就会新建一个。

touch 的用法也比较简单，就是 touch 命令后面直接跟文件路径。

我们在演示 touch 命令的用法前，得先挪个地方，因为我们不希望在 /var/log 这个系统目录里“胡作非为”。

如果你认真学习了前几课，那么你应该知道如何快速地回到我们的用户家目录，是的，就是单独一个 cd 命令，什么参数也不加。

```
File Edit View Search Terminal Help
oscar@oscar-laptop: /var/log$ cd
oscar@oscar-laptop: ~$
```

回到我们的家目录之后，我们用 touch 命令来创建新文件：

```
touch new_file
```

```
oscar@oscar-laptop: ~
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ touch new_file
oscar@oscar-laptop:~$ ls
Desktop  Downloads  new_file  Public  Templates
Documents Music  Pictures  share  Videos
oscar@oscar-laptop:~$
```

看到了吗，我们用 touch new\_file 创建了一个名叫 new\_file 的新空白文件。

我们可以再来创建一个：

```
touch new_file_2
```

```
oscar@oscar-laptop: ~
File Edit View Search Terminal Help
oscar@oscar-laptop:~$ touch new_file
oscar@oscar-laptop:~$ ls
Desktop  Downloads  new_file  Public  Templates
Documents Music  Pictures  share  Videos
oscar@oscar-laptop:~$ touch new_file_2
oscar@oscar-laptop:~$ ls
Desktop  Downloads  new_file  Pictures  share  Videos
Documents Music  new_file_2  Public  Templates
oscar@oscar-laptop:~$
```

事实上，我们可以用 touch 命令同时创建多个文件：

```
touch new_file new_file_2
```

如果我所要创建的文件的名字里有空格怎么办呢？

很简单，加上双引号：

```
touch "new file"
```

不过，在 Linux 中，尽量不要让文件或目录名字中有空格，有时会给编程造成麻烦。

## mkdir: 创建一个目录

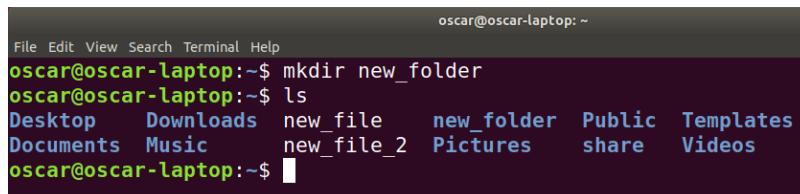
如果说 touch 命令可以用来创建一个文件，那么 mkdir 命令就是用于创建一个目录的。

mkdir 其实是 mk 和 dir 的缩合，mk 是 make 的缩写，表示“创建”；dir 是 directory 的缩写，表示“目录”。

所以，学编程的过程中我们也可以提高英语，不是吗？

mkdir 的用法也和 touch 一样，就是后面接需要创建的目录路径。例如：

```
mkdir new_folder
```

A terminal window titled 'oscar@oscar-laptop: ~' showing the execution of 'mkdir new\_folder' and 'ls'. The 'ls' output shows 'new\_folder' in blue, indicating it is a directory. A large red watermark is overlaid on the image.

```
oscar@oscar-laptop: ~  
File Edit View Search Terminal Help  
oscar@oscar-laptop:~$ mkdir new_folder  
oscar@oscar-laptop:~$ ls  
Desktop  Downloads  new_file    new_folder  Public  Templates  
Documents Music      new_file_2  Pictures    share   Videos  
oscar@oscar-laptop:~$
```

可以看到，我们用 mkdir new\_folder 创建了一个名叫 new\_folder 的新目录，这个目录里当然还没有什么东西。

可以看到用 ls 命令后，显示的 new\_folder 是深蓝色的，表示它是一个目录。

mkdir 也可以同时创建好几个目录：

```
mkdir new_folder new_folder_2
```

如果我所要创建的目录的名字里有空格怎么办呢？

很简单，加上双引号：

```
mkdir "new folder"
```

不过，再提一句：在 Linux 中，尽量不要让文件或目录名字中有空格，有时会给编程造成麻烦。

我们还可以用 -p 参数来递归地创建目录结构。

比如：

```
mkdir -p one/two/three
```

就会先创建 one 这个目录，在 one 目录里面创建 two 这个目录，在 two 里面再创建 three 这个目录：



```
File Edit View Search Terminal Help
oscar@oscar-laptop: ~/one/two
oscar@oscar-laptop:~$ mkdir -p one/two/three
oscar@oscar-laptop:~$ ls
Desktop  Downloads  new_file  new_folder  Pictures  share  Videos
Documents Music      new_file_2 one         Public    Templates
oscar@oscar-laptop:~$ cd one
oscar@oscar-laptop:~/one$ ls
two
oscar@oscar-laptop:~/one$ cd two
oscar@oscar-laptop:~/one/two$ ls
three
oscar@oscar-laptop:~/one/two$
```

## 小结

1. cat 命令用于显示文件的所有内容。如果文件内容太多，最好用 less 命令，可以一页一页地显示文件；
2. 我们可以用 head 和 tail 命令分别显示文件开头和结尾的若干行。特别地，使用 tail -f，我们可以实时地显示文件的变化，对于随时间一直在改变的文件，例如日志（log）文件特别有用；
3. mkdir 命令用于创建目录。touch 命令用于创建空文件。使用 mkdir -p，我们可以递归地创建目录结构。

今天的课就到这里，一起加油吧！

更多一手资源+V：AndyqcI  
aa:3118617541