

## 文件与文件夹操作

### ► 检查路径

命令	作用
<code>pwd</code>	返回当前所在文件夹的绝对路径
<code>realpath filename</code>	返回 <code>filename</code> 文件的绝对路径
<code>ls [path]</code>	显示 <code>path</code> 下所有文件和文件夹. 默认 <code>path</code> 为 <code>.</code>

### ► 新建

命令行下新建文件可以使用 `touch` 命令

```
touch new-file
```

`touch` 命令本身只是更新文件和文件夹的时间戳, 如文件不存在时则会创建.

用 `mkdir` 命令新建文件夹, 例如

```
mkdir new-dir
```

在当前目录下新建名为 `new-dir` 的文件夹. 当 `new-dir` 存在时, 会报错

```
mkdir: cannot create directory 'new-dir': File exists
```

并返回1. 使用 `-p` 可以消除这一错误.

使用 `-p` 的另一个情景是递归新建文件夹, 例如

```
mkdir -p a/b/c/new-dir
```

父文件夹 `a`, `b`, `c` 中任意一个不存在时, 都会在对路径新建.

### ► 复制

用 `cp` 复制文件

```
cp filename1 filename2
```

复制多个文件到某一文件夹下

```
cp filename1 filename2 filename3 dir
```

当 `cp` 后变量大于两个时, 最后一个必须是文件夹名.

复制文件夹时需要加上 `-r` 选项以递归

```
cp -r dirname1 dirname2
```

## ► 移动与重命名

用 `mv` 命令对文件夹和文件进行移动和重命名操作. 基本逻辑和 `cp` 类似, 不同的是文件夹操作不需要进行递归.

本质上, 重命名和移动是相同的操作, 即修改文件的路径. 命令

```
mv path1 path2
```

中, 如果 `path2` 是存在的文件夹, 那么这一命令就是将 `path1` 对象移动到 `path2` 下. 如果 `path2` 不存在, 那么就是将 `path1` 重命名为 `path2`. `path1` 所指的对象, 既可以是文件, 也可以是文件夹. 注意这一操作会覆盖 `path2` 下的内容. 除非有意为之, 在移动前最好检查路径 `path2` 是否存在.

批量操作只适用于移动文件

```
mv file1 file2 file3 dir
```

最后一个参数必须为已存在的文件夹.

## ► 软链接

用 `ln -s` 创建文件和文件夹的软链接. 使用软链接的主要情境包括

1. 将其他路径的文件或文件夹放入某特定路径, 但又不希望移动或者复制它. 例如在一个文件夹下统一管理不同程序的配置文件.
2. 使文件同时以不同的名称的副本存在, 所有副本随原本同步变化. 常见于创建可执行程序或库的副本, 文件名涉及大版本号与小版本号.

在 `path2` 处产生 `path1` 文件的软链接

```
ln -s path1 path2
```

`path1` 和 `path2` 最好都使用绝对路径.