

## 13.7 复制或者移动文件和目录

### 问题

你想要复制或移动文件和目录，但是又不想调用shell命令。

### 解决方案

`shutil` 模块有很多便捷的函数可以复制文件和目录。使用起来非常简单，比如：

```
import shutil
```

```
# Copy src to dst. (cp src dst)
shutil.copy(src, dst)
```

```
# Copy files, but preserve metadata (cp -p src dst)
shutil.copy2(src, dst)
```

```
# Copy directory tree (cp -R src dst)
shutil.copytree(src, dst)
```

```
# Move src to dst (mv src dst)
shutil.move(src, dst)
```

这些函数的参数都是字符串形式的文件或目录名。底层语义模拟了类似的Unix命令，如上面的注释部分。

默认情况下，对于符号链接而已这些命令处理的是它指向的东西。例如，如果源文件是一个符号链接，那么目标文件将会是符号链接指向的文件。如果你只想复制符号链接本身，那么需要指定关键字参数 `follow_symlinks`，如下：

如果你想保留被复制目录中的符号链接，像这样做：

```
shutil.copytree(src, dst, symlinks=True)
```

`copytree()` 可以让你在复制过程中选择性的忽略某些文件或目录。你可以提供一个忽略函数，接受一个目录名和文件名列表作为输入，返回一个忽略的名称列表。例如：

```
def ignore_pyc_files(dirname, filenames):
    return [name in filenames if name.endswith('.pyc')]
```

```
shutil.copytree(src, dst, ignore=ignore_pyc_files)
```

由于忽略某种模式的文件名是很常见的，因此一个便捷的函数 `ignore_patterns()` 已经包含在里面了。例如：

```
shutil.copytree(src, dst, ignore=shutil.ignore_patterns('*~', '*.pyc'))
```

### 讨论

使用 `shutil` 复制文件和目录也忒简单了点吧。不过，对于文件元数据信息，`copy2()` 这样的函数只能尽自己最大能力来保留它。访问时间、创建时间和权限这些基本信息会被保留，但是对于所有者、ACLs、资源fork和其他更深层次的文件元信息就说不准了，这个还得依赖于底层操作系统类型和用户所拥有的访问权限。你通常不会去使用 `shutil.copytree()` 函数来执行系统备份。当处理文件名的时候，最好使用 `os.path` 中的函数来确保最大的可移植性（特别是同时要适用于

Unix和Windows）。例如：

```
>>> filename = '/Users/guido/programs/spam.py'
>>> import os.path
>>> os.path.basename(filename)
'spam.py'
>>> os.path.dirname(filename)
'/Users/guido/programs'
>>> os.path.split(filename)
('/Users/guido/programs', 'spam.py')
>>> os.path.join('/new/dir', os.path.basename(filename))
'/new/dir/spam.py'
>>> os.path.expanduser('~'/guido/programs/spam.py')
'/Users/guido/programs/spam.py'
>>>
```

使用 `copytree()` 复制文件夹的一个棘手的问题是对于错误的处理。例如，在复制过程中，函数可能会碰到损坏的符号链接，因为权限无法访问文件的问题等等。为了解决这个问题，所有碰到的问题会被收集到一个列表中并打包为一个单独的异常，到了最后再抛出。下面是一个例子：

```
try:
    shutil.copytree(src, dst)
except shutil.Error as e:
    for src, dst, msg in e.args[0]:
        # src is source name
        # dst is destination name
        # msg is error message from exception
        print(dst, src, msg)
```

如果你提供关键字参数 `ignore_dangling_symlinks=True`，这时候 `copytree()` 会忽略掉无效符号链接。

本节演示的这些函数都是最常见的。不过，`shutil` 还有更多的和复制数据相关的操作。它的文档很值得一看，参考 [Python documentation](#)