STAT243 Lecture 3.2 Bash Shell-File Management

Logic

文件通常具有以下属性：

- Name 姓名
- Type 类型
- Location 位置
- Size 大小
- Protection (i.e., permissions on what can be done with the file)
  保护（即对文件可执行的操作权限）
- Time, date, and user identification
  时间、日期和用户标识

1 查找文件和浏览文件系统

1.1 查找文件

可以通过文件名, 修改时间和类型查找文件:

Shell

```
$ find . -name '*.txt'  # find files named *.txt
$ find . -mtime -2      # find files modified less than 2 days ago
$ find . -type l        # find links
```

这里的 `.` 参数表示在当前工作目录及其子目录中查找文件

Remark

可以使用以下命令获取关于 `find` 命令的更多信息:

Shell

```
$ man find
$ find --help
```

1.2 保留多个先前工作目录的堆栈

除了使用 `cd -` 返回到前一个使用的工作目录外，如果你希望保留多个先前工作目录的堆栈而不是仅保留最后一个，还可以使用 `pushd` (push directory), `popd` (pop directory) 和 `dirs` (directories) 命令

1.3 特殊目录

在每个目录中，有两个特殊目录， `.` 和 `..` ，它们分别指向当前目录和当前目录的父目录。只有当我们使用 `-a` 标志来显示隐藏文件时，才能看到这些目录

Shell

```
$ ls -al
total 1489
drwxr-sr-x  7 paciorek scfstaff     31 Apr 21 16:39  ./
drwxr-sr-x 19 paciorek scfstaff     30 Feb 28 15:07  ../
```

2 文件名匹配 (globbing 通配符)

Wildcard 通配符	Function 功能
`*`	Match zero or more characters. 匹配零个或多个字符。
`?`	Match exactly one character. 匹配恰好一个字符。
`[characters]`	Match any single character from among *characters* listed between brackets. 匹配括号中列出的任意单个字符。
`[!characters]`	Match any single character other than *characters* listed between brackets. 匹配括号中未列出的任意单个字符。
`[a-z]`	Match any single character from among the range of characters listed between brackets. 匹配括号中列出的字符范围内的任意单个字符。
`[!a-z]`	Match any single character from among the characters not in the range listed between brackets 匹配括号中未列出的字符范围内的任意单个字符
`{frag1,frag2,…}`	Brace expansion: create strings frag1, frag2, etc. 括号扩展：创建字符串 frag1、frag2 等

Example

列出所有以数字结尾的文件:

Shell

```
$ ls *[0-9]
```

Example

将 `filename` 复制为 `filename.old` :

Shell

```
$ cp filename{,.old}
```

Example

删除所有以 a 或 z 开头的文件:

Shell

```
$ rm [az]*
```

Example

列出所有具有各种后缀的 R 代码文件:

Shell

```
$ ls *.{r,R}
```

Example

`echo` 命令可用于验证通配符扩展是否按预期工作：

Shell

```
$ echo cp filename{,.old}
cp filename filename.old
```

Remark

1. 可以使用反斜杠前缀（ `\` ）抑制通配符转义

   Shell

   ```
   $ touch \*test    # create a file called *test
   $ ls \**
   *test
   ```

2. 要了解更多关于 standard globbing patterns 的信息，可以使用:

   Shell

   ```
   $ man 7 glob
   ```

   Remark

   该指令在 mac 上无法运行, 可以考虑使用 `man 3 glob` 替代

3 文件权限

3.1 查看文件权限

可以在 `ls` 后添加 `-l` flag 来查看文件权限

Shell

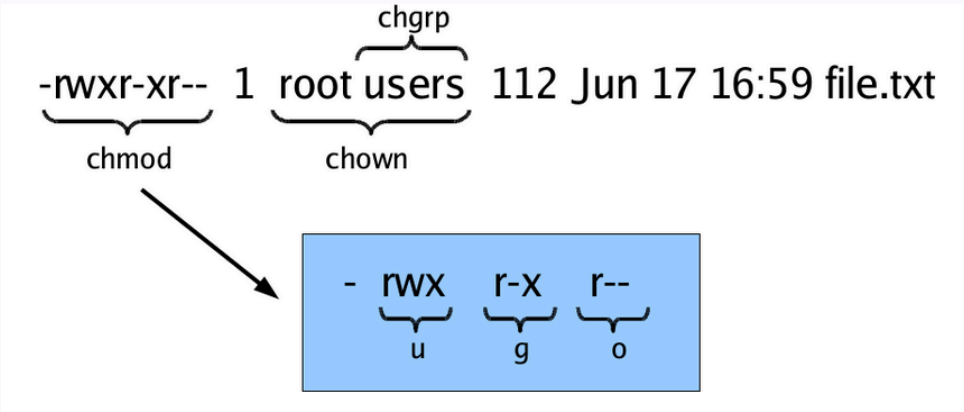
```
$ cd ~/stat243-fall-2020
$ ls -l
```

```
total 152
drwxrwxr-x 2 scflocal scflocal 4096 Dec 28 13:15 data
drwxrwxr-x 2 scflocal scflocal 4096 Dec 28 13:15 howtos
drwxrwxr-x 2 scflocal scflocal 4096 Dec 28 13:15 project
drwxrwxr-x 2 scflocal scflocal 4096 Dec 28 13:15 ps
-rw-rw-r-- 1 scflocal scflocal 11825 Dec 28 13:15 README.md
drwxrwxr-x 13 scflocal scflocal 4096 Dec 28 13:15 sections
-rw-rw-r-- 1 scflocal scflocal 37923 Dec 28 13:15 syllabus.lyx
-rw-rw-r-- 1 scflocal scflocal 77105 Dec 28 13:15 syllabus.pdf
drwxrwxr-x 2 scflocal scflocal 4096 Dec 28 13:37 units
```

其中所包含的重要信息分别是:

- (column 1) file permissions (more later)
  文件权限（稍后详述）
- (column 3) the owner of the file ('scflocal' here)
  文件的所有者（此处为'scflocal'）
- (column 4) the group of users that the file belongs too (also 'scflocal' here)
  文件所属的用户组（此处也为'scflocal'）
- (column 5) the size of the file in bytes
  文件大小（字节）
- (column 6-8) the last time the file was modified
  文件最后修改时间
- (column 9) name of the file
  文件名

Example



这是名为"file.txt"的文件的一个 graphical summary，其所有者是"root"，组是"users"。（图中还表明可以使用命令 `chmod` 、 `chown` 和 `chgrp` 来更改文件权限和所有者。）

3.2 文件权限详解

```
$ ls -l
total 156
drwxrwxr-x  2 scflocal scflocal  4096 Dec 28 13:15 data
drwxrwxr-x  2 scflocal scflocal  4096 Dec 28 13:15 howtos
drwxrwxr-x  2 scflocal scflocal  4096 Dec 28 13:15 project
drwxrwxr-x  2 scflocal scflocal  4096 Dec 28 13:15 ps
-rw-rw-r--  1 scflocal scflocal 11825 Dec 28 13:15 README.md
drwxrwxr-x 13 scflocal scflocal  4096 Dec 28 13:15 sections
-rw-rw-r--  1 scflocal scflocal 37923 Dec 28 13:15 syllabus.lyx
-rw-rw-r--  1 scflocal scflocal 77105 Dec 28 13:15 syllabus.pdf
drwxrwxr-x  2 scflocal scflocal  4096 Dec 28 13:37 units
```

第一列中包含 10 个 individual single-character columns, 分别表示

- 第 1 个字母: 表示是否是 directory (`d` 表示是, `-` 表示不是)
- 第 2-4 个字母: 表示**文件所有者**是否可以读取 (`r`), 写入 (`w`) 和执行 (`x`),
- 第 5-7 个字母: 表示**文件所属 group 的任何成员**是否可以读取 (`r`), 写入 (`w`) 和执行 (`x`)
- 第 8-10 个字母: 表示**其他任何用户**是否可以读取 (`r`), 写入 (`w`) 和执行 (`x`)
- `-` 表示没有对应权限

Example

例如，对于 syllabus.pdf 文件，文件的所有者可以读取它，并且可以通过写入来修改文件（第一个 triplet 是 `'rw-'`），文件所属组的用户也可以这样做。但对于其他用户，他们只能读取它（第三个 triplet 是 `'r--'`）

3.3 更改文件权限

可以使用 `chmod` 指令来更改文件权限, 使用时需要指定:

- 用户类型:
  - `u`: 文件所有者
  - `g`: 文件所属组的用户
  - `o`: 其他任何用户
- 进行添加/移除操作:
  - `+`: 添加权限
  - `-`: 移除权限
- 权限类型:
  - `r`: 读取
  - `w`: 写入
  - `x`: 执行

3.4 例子: 阻止任何人读取 `tmp.txt` 文件

首先创建文件:

```
$ echo "first line" > tmp.txt  # create a test text file that contains "first line"
$ ls -l tmp.txt
-rw-rw-r--  1 scflocal scflocal    11 Dec 28 13:39 tmp.txt
```

移除读取权限:

```
$ chmod u-r tmp.txt # prevent owner from reading
$ chmod g-r tmp.txt # prevent users in the file's group from reading
$ chmod o-r tmp.txt # prevent others from reading
$ ls -l tmp.txt
--w--w---- 1 scflocal scflocal 11 Dec 28 13:39 tmp.txt
```

或者使用单行指令:

```
$ chmod ugo-r tmp.txt # prevent all three
$ ls -l tmp.txt
--w--w---- 1 scflocal scflocal 11 Dec 28 13:39 tmp.txt
```

此时我们无法读取文件:

```
$ cat tmp.txt
cat: tmp.txt: Permission denied
```

如果想同时移除读取和写入权限, 可以:

```
$ chmod ugo-rw tmp.txt # prevent all three
```

此时如果使用 `>>` 重定向运算符向文件添加一行，会被拒绝:

```
$ echo "added line" >> tmp.txt
-bash: tmp.txt: Permission denied
```

接下来我们把权限恢复给所有者:

```
$ chmod u+rw tmp.txt
$ echo "added line" >> tmp.txt
$ cat tmp.txt
first line
added line
```

3.5 令其他用户访问文件

在使文件对其他用户可访问时，还有许多重要的细节，包括:

- how to make files in a particular directory available to other users on the system,
  如何使特定目录中的文件对系统上的其他用户可用
- how to set up a directory for use by a UNIX group
  如何为 UNIX group 设置 directory，即使用所谓的“sticky bit”，以便将来在目录中创建的文件属于该组，使组成员可以默认轻松访问它们
- how to use *access control lists*
  如何使用访问控制列表来获得更多控制权

4 尽可能使用简单的 text files

- UNIX 命令是强大的 text file 操作工具, 因此尽可能地将信息存储在 text files 中是有帮助的
- UNIX 操作文件的基本命令是**逐行操作**的（例如， `grep` 、 `sed` 、 `cut` 等）。因此，使用每行包含不同信息集的格式（如 CSV）比其他存储相关信息在多行中的文本格式（如 XML 和 JSON）更有优势

> ⚠ **Remark**
>
> 对于大型数据集, 仍然建议存储在二进制文件中，特别是考虑
>
> - 数据访问速度是否快
> - 存储格式是否高效

5 文档格式和转换

Pandoc 是一个广泛使用的文档转换工具

要将一个 Markdown（ `report.md` ）格式的文件转换为 PDF（ `report.pdf` ），可以这样做:

Shell

```
$ pandoc -o report.pdf report.md
```