



Chapter 01 Assignment

- 修改README.md并创建Pull request

请小范老师给大家做讲解



第二章

Linux文件系统

徐刚 工程师

华东师范大学软件工程学院

国家可信嵌入式软件工程技术研究中心

Outline

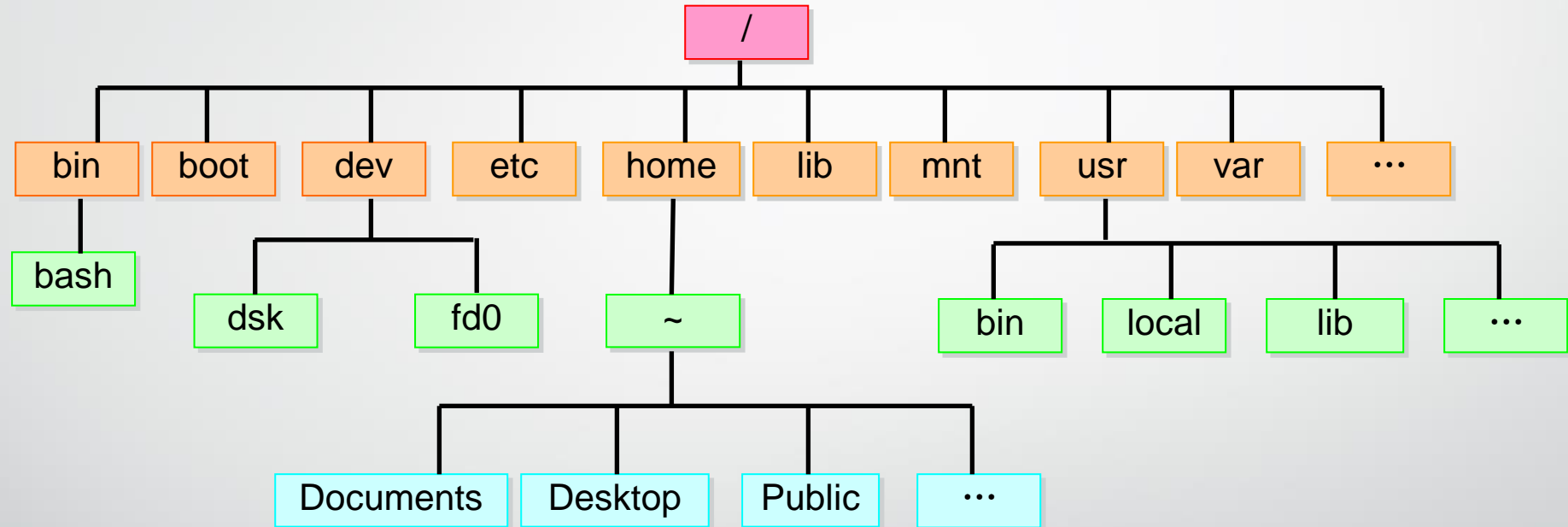
- Linux的文件
- 文件系统结构
- 文件操作命令

Linux 文件

- 在Linux中，一切都是文件
 - 内核、shell都是文件
- 文件分类
 - 普通文件：仅包含字符流的数据文件（文本文件、二进制文件）
 - 目录文件：文件夹和子目录的详细信息（文件名和inode号）
 - 设备文件：代表设备，完成对设备的读取和写入操作
- 文件命名规则
 - 字母和数字
 - 符号：. , - , _
 - 可以有扩展名，也可以没有
 - 区分大小写

Linux文件系统层次结构

- 层次结构



系统目录

- /: 根目录（之后的/都是目录分隔符）
- /home: 用户目录
- /bin: Unix常用命令，如bash, date, cat, tar等
- /sbin: 管理员命令，如fdisk, mkfs等
- /etc: 系统配置文件目录，如passwd, shadow（登录名和密码）等
- /dev: 设备文件目录，如硬盘等
- /lib: 库文件目录
- /var: 可变文件目录，如打印和邮件等
- /tmp: 临时文件目录

路径

- 绝对路径名：从根目录开始
- 相对路径名：从当前目录开始
- 根目录：/
- 当前用户主目录：~或者环境变量\$HOME
- 当前目录：.
- 父目录：..

文件系统相关命令

- ls 列出文件
- cd 变更目录
- mkdir 创建目录
- rmdir 删除目录
- cp 复制文件
- mv 移动文件，重命名
- rm 删除文件
- tar 文件打包压缩

ls (list)

- 命令功能：列出文件
- 命令格式：`ls [OPTION]... [FILE]...`
- 主要选项：
 - a 显示以.开头的文件
 - F 用*标记可执行文件，用/标记目录，用@标记符号链接
 - l 以列表形式显示文件属性
 - t 按最后修改时间排序（降序）
 - u 按最后访问时间排序
 - r 逆序排序
 - R 递归显示子目录和文件
- 示例：

cd (change directory)

- 命令功能：改变当前工作目录
- 命令格式：cd [dir]
- 示例：

- ```
draco@ubuntu:~$ cd /home/draco/Documents
draco@ubuntu:~/Documents$
```

- ```
draco@ubuntu:~$ cd Downloads  
draco@ubuntu:~/Downloads$
```

- ```
draco@ubuntu:~/Downloads$ cd ..
draco@ubuntu:~$
```

- ```
draco@ubuntu:/etc/emacs$ cd  
draco@ubuntu:~$
```

mkdir (make directory)

- 命令功能：创建目录（文件夹）

- 命令格式：`mkdir [OPTION]... DIRECTORY...`

- 示例：

- `draco@ubuntu:~$ mkdir test`

- `draco@ubuntu:~$ mkdir test1 test2 test/subtest`

- `draco@ubuntu:~$ mkdir test`
`mkdir: cannot create directory 'test': File exists`

- `draco@ubuntu:/bin$ mkdir test`
`mkdir: cannot create directory 'test': Permission denied`

rmdir (remove directory)

- 命令功能：删除空目录
- 命令格式：`rmdir [OPTION]... DIRECTORY...`
- 常用选项：-p 删除空目录树
- 示例：

- `draco@ubuntu:~/test$ rmdir subtest`

- `draco@ubuntu:~$ rmdir test/subtest test`

- `draco@ubuntu:~$ rmdir -p test/subtest`

- `draco@ubuntu:~$ rmdir Downloads`
`rmdir: failed to remove 'Downloads': Directory not empty`

cp (copy)

- 命令功能：复制文件
- 命令格式：`cp [OPTION]... SOURCE... DIRECTORY`
- 常用选项：
 - b 为已存在目标文件创建备份
 - n 不覆盖已有目标文件
 - i 覆盖时询问
 - R 递归复制子目录和文件
- 示例：

- ```
draco@ubuntu:~$ cp workspace/helloworld.c Documents
draco@ubuntu:~$ ls Documents
helloworld.c
```

- ```
draco@ubuntu:~$ cp -R workspace Documents/new
draco@ubuntu:~$ ls Documents/new
hello helloworld helloworld.c helloworld.c~ RemoteSystemsTempFiles test1
```

mv (move)

- 命令功能：移动文件，重命名文件
- 命令格式：`mv [OPTION]... SOURCE... DIRECTORY`
- 常用选项：
 - b 为已存在目标文件创建备份
 - n 不覆盖已有目标文件
 - i 覆盖时询问
- 示例：

```
draco@ubuntu:~/Documents$ ls
helloworld.c  new
draco@ubuntu:~/Documents$ mv helloworld.c helloworld.txt
draco@ubuntu:~/Documents$ ls
helloworld.txt  new
```

```
draco@ubuntu:~/Documents$ mv helloworld.txt new
draco@ubuntu:~/Documents$ ls new
hello helloworld helloworld.c~ helloworld.txt RemoteSystemsTempFiles test1
```

rm (remove)

- 命令功能：删除文件
- 命令格式：`rm [OPTION]... [FILE]...`
- 常用选项：
 - f 强制删除
 - i 删除时询问
 - R 递归删除子目录和文件
- 示例：

```
draco@ubuntu:~/Documents$ ls
hello helloworld helloworld.c~ helloworld.txt new RemoteSystemsTempFiles test1
draco@ubuntu:~/Documents$ rm helloworld.txt
draco@ubuntu:~/Documents$ ls
hello helloworld helloworld.c~ new RemoteSystemsTempFiles test1
```

```
draco@ubuntu:~/Documents$ ls
hello new RemoteSystemsTempFiles test1
draco@ubuntu:~/Documents$ rm -R ./*
```

cat (concatenation)

- 命令功能：在shell中显示文件内容
- 命令格式： `cat [OPTION]... [FILE]...`
- 示例：

- ```
draco@ubuntu:~/workspace$ cat helloworld.c
/*helloworld.c*/
#include <stdio.h>
void main()
{
 printf("HelloWorld!\n");
}
```

- ```
draco@ubuntu:~/workspace$ cat helloworld.c helloworld.txt > double
draco@ubuntu:~/workspace$ cat double
/*helloworld.c*/
#include <stdio.h>
void main()
{
    printf("HelloWorld!\n");
}
/*helloworld.c*/
#include <stdio.h>
void main()
{
    printf("HelloWorld!\n");
}
```


tar (tape archiver)

- 命令功能：存档管理
- 命令格式：`tar [OPTION...] [FILE]...`
- 常用选项：
 - c 创建存档
 - x 提取存档
 - t 显示存档中的文件
 - f 指定存档名称
 - v 显示执行过程
 - z 同时使用gzip压缩文档

- 示例：

- ```
draco@ubuntu:~/workspace$ tar -cvf archive.tar helloworld helloworld.c
helloworld
helloworld.c
```

- ```
draco@ubuntu:~/Documents$ tar -xvf ../workspace/archive.tar
helloworld
helloworld.c
```

gzip

- 命令功能：压缩文件，解压缩文件
- 命令格式： `gzip [OPTION]... [FILE]...`
- 常用选项：
 - d 解压缩
 - k 保留原文件
 - l 显示压缩文件内容
 - r 递归压缩子目录和文件
- 示例：

- ```
draco@ubuntu:~/Documents$ gzip -k helloworld.c
draco@ubuntu:~/Documents$ ls
helloworld helloworld.c helloworld.c.gz
```

- ```
draco@ubuntu:~/Documents$ gzip -l helloworld.c.gz
      compressed      uncompressed  ratio uncompressed_name
            108              79   2.5% helloworld.c
```

用户管理相关命令

- sudo 使用root权限执行命令
- adduser 添加新用户
- passwd 修改用户密码
- usermod 修改用户信息
- deluser 删除用户
- su 切换登录用户

sudo

- 命令功能：暂时以root权限执行命令
- 命令格式：sudo 想要执行的命令
- 命令说明：使用时会要求输入当前用户的密码
- 示例：

- ```
draco@ubuntu:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

- ```
draco@ubuntu:~$ sudo cat /etc/shadow
[sudo] password for draco:
```

adduser

- 命令功能：添加新用户
- 命令格式：adduser 用户名
- 命令说明：
 - 以交互方式简单方便的添加用户
 - 为用户设定密码
 - 为用户创建同名用户组和主目录
- 示例：

passwd

- 命令功能：修改用户密码
- 命令格式：`passwd [options] [LOGIN]`
- 常用选项：
 - d 删除用户密码
 - l 锁定用户，阻止登录
 - u 解锁用户

示例：

usermod

- 命令功能：编辑用户信息
- 命令格式：**usermod [options] LOGIN**
- 常用选项：
 - c 账号说明
 - d 修改主目录
 - s 修改默认shell
 - g 修改主群组
 - G 修改附加群组
 - L 锁定用户，阻止登录
 - U 解锁用户
- 示例：

deluser

- 命令功能：删除用户
- 命令格式：`deluser [options] user`
- 常用选项：
 - group 删除用户组
 - remove-home 删除用户主目录
 - remove-all-files 删除系统中owner是该用户的所有文件
- 示例：

SU

- 命令功能：切换用户
- 命令格式：`su [options] [username]`
- 常见选项：
 - 复制当前环境变量
 - c 以该用户身份执行一条命令，不切换用户
- 示例：

文件属性

- 文件类型
- 文件权限
- 文件链接
- 改变文件权限

文件类型属性

- ```
draco@ubuntu:~/Documents$ ls -l
total 24
-rwxrwxr-x 1 draco draco 8600 Jul 30 16:04 helloworld
-rw-rw-r-- 1 draco draco 79 Jul 30 16:04 helloworld.c
-rw-rw-r-- 1 draco draco 108 Jul 30 16:04 helloworld.c.gz
drwxrwxr-x 2 draco draco 4096 Sep 17 18:19 newdir
```
- - 表示普通文件
- d 表示目录
- l 表示符号链接

# 文件权限属性

- ```
draco@ubuntu:~/Documents$ ls -l
total 24
-rwxrwxr-x 1 draco draco 8600 Jul 30 16:04 helloworld
-rw-rw-r-- 1 draco draco  79 Jul 30 16:04 helloworld.c
-rw-rw-r-- 1 draco draco 108 Jul 30 16:04 helloworld.c.gz
drwxrwxr-x 2 draco draco 4096 Sep 17 18:19 newdir
```

- 权限
 - r 可读，w 可写，x 可执行，- 无权限
- 权限字符串：9位字符串
 - 前三位表示所有者权限
 - 中三位表示用户组权限
 - 后三位表示其他用户权限

文件所有权

- ```
draco@ubuntu:~/Documents$ ls -l
total 24
-rwxrwxr-x 1 draco draco 8600 Jul 30 16:04 helloworld
-rw-rw-r-- 1 draco draco 79 Jul 30 16:04 helloworld.c
-rw-rw-r-- 1 draco draco 108 Jul 30 16:04 helloworld.c.gz
drwxrwxr-x 2 draco draco 4096 Sep 17 18:19 newdir
```
- 所有权和用户组所有权
- 文件所有者可以修改文件权限
- 文件所有者对文件的操作权限显示在权限字符串的前三位
- 用户组所有者对文件的操作权限显示在权限字符串的中三位

# 文件大小

- ```
draco@ubuntu:~/Documents$ ls -l
total 24
-rwxrwxr-x 1 draco draco 8600 Jul 30 16:04 helloworld
-rw-rw-r-- 1 draco draco 79 Jul 30 16:04 helloworld.c
-rw-rw-r-- 1 draco draco 108 Jul 30 16:04 helloworld.c.gz
drwxrwxr-x 2 draco draco 4096 Sep 17 18:19 newdir
```
- 文件的字符数，与实际占用空间不同
- 目录的大小取决于所包含的文件名，显示的是预期值

文件最后修改时间

- ```
draco@ubuntu:~/Documents$ ls -l
```

| Permissions | Count | User  | Group | Size | Time         | File            |
|-------------|-------|-------|-------|------|--------------|-----------------|
| -rwxrwxr-x  | 1     | draco | draco | 8600 | Jul 30 16:04 | helloworld      |
| -rw-rw-r--  | 1     | draco | draco | 79   | Jul 30 16:04 | helloworld.c    |
| -rw-rw-r--  | 1     | draco | draco | 108  | Jul 30 16:04 | helloworld.c.gz |
| drwxrwxr-x  | 2     | draco | draco | 4096 | Sep 17 18:19 | newdir          |
- 文件最后修改的时间戳

# chmod

- 命令功能：改变文件权限
- 命令格式：`chmod [OPTION]... MODE[,MODE]... FILE...`
- 常用选项：-R 递推改变所有子目录和文件权限
- mode:

| 类别     | 操作       | 权限     |
|--------|----------|--------|
| u 用户   | + 增加权限   | r 读权限  |
| g 用户组  | - 删除权限   | w 写权限  |
| o 其他用户 | = 指定绝对权限 | x 执行权限 |
| a 全部   |          |        |

- 示例：



# 八进制权限表示

- 绝对权限指定
  - 读权限：4（100）
  - 写权限：2（010）
  - 执行权限：1（001）
- 默认普通文件权限：
  - 666 rw-rw-rw-
- 示例：

| 权限  | 二进制 | 八进制 |
|-----|-----|-----|
| --- | 000 | 0   |
| --x | 001 | 1   |
| -w- | 010 | 2   |
| -wx | 011 | 3   |
| r-- | 100 | 4   |
| r-x | 101 | 5   |
| rw- | 110 | 6   |
| rwX | 111 | 7   |

```
draco@ubuntu:~/Documents$ ls -l helloworld.c
-r--rw-r-- 1 draco draco 79 Jul 30 16:04 helloworld.c
draco@ubuntu:~/Documents$ chmod 664 helloworld.c
draco@ubuntu:~/Documents$ ls -l helloworld.c
-rw-rw-r-- 1 draco draco 79 Jul 30 16:04 helloworld.c
```

# 目录权限

- r 读权限：是否可以访问文件名列表
- w 写权限：是否可以创建或删除文件
- x 执行权限：是否可以在路径中使用
- 默认目录权限：777 rwxrwxrwx
- 示例：

- ```
draco@ubuntu:~/Documents$ chmod u-r newdir
draco@ubuntu:~/Documents$ ls newdir
ls: cannot open directory 'newdir': Permission denied
```

- ```
draco@ubuntu:~/Documents$ chmod u-w newdir
draco@ubuntu:~/Documents$ cp helloworld.c newdir
cp: cannot create regular file 'newdir/helloworld.c': Permission denied
```

- ```
draco@ubuntu:~/Documents$ ls -ld newdir
drwxrwxr-x 2 draco draco 4096 Sep 17 21:18 newdir
draco@ubuntu:~/Documents$ ls -l newdir
total 12
----rwxr-x 1 draco draco 8600 Sep 17 21:18 helloworld
draco@ubuntu:~/Documents$ rm newdir/helloworld
rm: remove write-protected regular file 'newdir/helloworld'? y
```

umask

- 命令功能：设定默认权限
- 命令格式：umask [八进制权限掩码]
- 作用：最终权限 = 默认权限码 - 权限掩码
- 示例：

- ```
draco@ubuntu:~/Documents$ ls -l newfile
-rw-rw-r-- 1 draco draco 5 Sep 17 21:37 newfile
draco@ubuntu:~/Documents$ umask 022
draco@ubuntu:~/Documents$ cp newfile newfile1
draco@ubuntu:~/Documents$ ls -l newfile1
-rw-r--r-- 1 draco draco 5 Sep 17 21:39 newfile1
```

# chown (change owner)

- 命令功能：改变文件所有者
- 命令格式：`chown [OPTION]... [OWNER][:[GROUP]] FILE...`
- 需要superuser权限才能执行
- 示例：

```
draco@ubuntu:~/Documents$ sudo chown nancy helloworld.c.gz
draco@ubuntu:~/Documents$ ls -l
total 32
-rw-rw-r-- 4 draco draco 79 Jul 30 16:04 alias.c
-rwxrwxrwx 2 draco draco 8600 Jul 30 16:04 helloworld
-rw-rw-r-- 4 draco draco 79 Jul 30 16:04 helloworld.c
-rw-rw-r-- 2 nancy draco 108 Jul 30 16:04 helloworld.c.gz
drwxr-xr-x 2 draco draco 4096 Sep 17 22:17 linkdir
drwxrwxr-x 2 draco draco 4096 Sep 17 22:16 newdir
```

# chgrp (change group)

- 命令功能：改变文件用户组所有权
- 命令格式： `chgrp [OPTION]... GROUP FILE...`
- 需要super user权限
- 示例：

- ```
draco@ubuntu:~/Documents$ sudo chgrp nancy helloworld.c.gz
draco@ubuntu:~/Documents$ ls -l
total 32
-rw-rw-r-- 4 draco draco  79 Jul 30 16:04 alias.c
-rwxrwxrwx 2 draco draco 8600 Jul 30 16:04 helloworld
-rw-rw-r-- 4 draco draco  79 Jul 30 16:04 helloworld.c
-rw-rw-r-- 2 nancy nancy 108 Jul 30 16:04 helloworld.c.gz
```

文件与inode

- Linux支持多个文件系统
- 每个文件都使用inode编号与一个inode表格相关联
- inode表格包含了文件的属性：
 - 类型，权限，链接数，所有者，用户组，文件大小，修改时间
 - 一个指向文件存储磁盘区块的指针数组
- 文件在一个文件系统中唯一的inode编号
- 使用 `ls -li` 命令查看文件inode编号

```
draco@ubuntu:~/Documents$ ls -li
total 24
919078 -rwxrwxrwx 1 draco draco 8600 Jul 30 16:04 helloworld
920582 -rw-rw-r-- 1 draco draco  79 Jul 30 16:04 helloworld.c
920599 -rw-rw-r-- 1 draco draco 108 Jul 30 16:04 helloworld.c.gz
1447709 drwxrwxr-x 2 draco draco 4096 Sep 17 21:26 newdir
```

文件链接

- ```
draco@ubuntu:~/Documents$ ls -l
total 24
-rwxrwxr-x 1 draco draco 8600 Jul 30 16:04 helloworld
-rw-rw-r-- 1 draco draco 79 Jul 30 16:04 helloworld.c
-rw-rw-r-- 1 draco draco 108 Jul 30 16:04 helloworld.c.gz
drwxrwxr-x 2 draco draco 4096 Sep 17 18:19 newdir
```
- 一个文件可以有多个文件名,
- 称为一个文件有多个链接
- 可以通过任意一个链接访问该文件

# ln

- 命令功能：创建一个链接
  - 硬链接：创建一个具有相同inode编号的文件名
  - 符号链接：创建一个拥有独立inode的文件指向链接的文件
- 命令格式： `ln [OPTION]... [-T] TARGET LINK_NAME`
- 常用选项： `-s` 创建符号链接  
`-f` 强制创建链接（当链接名已被使用时）
- 示例：

```
draco@ubuntu:~/Documents$ ln helloworld.c alias.c
draco@ubuntu:~/Documents$ ls -li
total 28
920582 -rw-rw-r-- 2 draco draco 79 Jul 30 16:04 alias.c
919078 -rwxrwxrwx 1 draco draco 8600 Jul 30 16:04 helloworld
920582 -rw-rw-r-- 2 draco draco 79 Jul 30 16:04 helloworld.c
```



# 硬链接的作用

- 可以在文件实际位置发生变化时使用链接访问文件
- 假设程序中引用了如下位置的文件：file/data.txt
- 之后该文件被移动到了新目录：newfile/data.txt
- 为了避免修改程序，可以在file目录中创建一个newfile/data.txt的硬链接

```
draco@ubuntu:~/Documents$ ln newfile/data.txt file
draco@ubuntu:~/Documents$ ls -li file
total 4
918959 -rw-r--r-- 2 draco draco 7 Sep 17 22:55 data.txt
draco@ubuntu:~/Documents$ ls -li newfile
total 4
918959 -rw-r--r-- 2 draco draco 7 Sep 17 22:55 data.txt
```

- 可以防止意外删除：使用rm命令只是删除了一个链接

# 符号链接的作用

- 符号链接类似于windows系统中的快捷方式
- 创建一个指向该文件的指针文件
- 支持跨文件系统建立链接

```
dracoking@ubuntu:~/Documents/aa$ ln -s $HOME/Documents/newaa/* .
dracoking@ubuntu:~/Documents/aa$ ls -li
total 4
1712512 -rw-rw-r-- 2 dracoking dracoking 6 Sep 18 08:28 aa.lnk
1708659 lrwxrwxrwx 1 dracoking dracoking 38 Sep 18 09:05 aa.txt -> /home/dracoking/Documents/newaa/aa.txt

dracoking@ubuntu:~/Documents$ cat aa/aa.txt
adata
```

# 对比和区别

- 在Linux系统中，硬链接和符号链接是两种不同类型的链接，它们的主要区别如下：
  1. 硬链接：硬链接是一个指向文件数据的直接指针，也就是说，硬链接和它所链接的文件共享同一个inode。因此，硬链接和原始文件在文件系统中是等价的，删除其中一个不会影响另一个的访问。但是，硬链接不能跨文件系统，也不能链接到目录。
  2. 符号链接：符号链接（也被称为软链接）是一个指向另一个文件路径的特殊文件。符号链接有自己的inode，和它所链接的文件的inode是不同的。如果删除了符号链接所指向的原始文件，符号链接将变成一个指向不存在的文件的无效链接。符号链接可以跨文件系统，也可以链接到目录。

# stat

- 命令功能：查看文件属性
- 命令格式： `stat [OPTION]... FILE...`
- 常用选项：-f 查看文件系统属性
- 示例：

- ```
dracoking@ubuntu:~/Documents$ stat helloworld
File: 'helloworld'
Size: 8600          Blocks: 24          IO Block: 4096   regular file
Device: 801h/2049d  Inode: 1711882     Links: 1
Access: (0775/-rwxrwxr-x)  Uid: ( 1000/dracoking)   Gid: ( 1000/dracoking)
Access: 2016-08-04 21:19:43.029012180 +0800
Modify: 2016-08-02 20:21:24.182565370 +0800
Change: 2016-08-02 20:21:24.182565370 +0800
Birth: -
```
- ```
dracoking@ubuntu:~/Documents$ stat -f /
File: "/"
ID: 2c8b9ce5a445d1be Namelen: 255 Type: ext2/ext3
Block size: 4096 Fundamental block size: 4096
Blocks: Total: 9256080 Free: 7756248 Available: 7280306
Inodes: Total: 2359296 Free: 2091928
```

# touch

- 命令功能：修改文件时间戳，创建空文件
- 命令格式：**touch [OPTION]... FILE...**
- 常用选项：
  - a 只改变访问时间
  - c 不创建文件
  - m 只改变修改时间
  - t 使用YYMMDDhhmm更新时间
- 示例：

```
dracoking@ubuntu:~/Documents$ touch -a kk
dracoking@ubuntu:~/Documents$ stat kk
 File: 'kk'
 Size: 27 Blocks: 8 IO Block: 4096 regular file
Device: 801h/2049d Inode: 1708665 Links: 1
Access: (0664/-rw-rw-r--) Uid: (1000/dracoking) Gid: (1000/dracoking)
Access: 2016-09-20 20:55:10.713628948 +0800
Modify: 2016-09-20 20:35:28.803681615 +0800
Change: 2016-09-20 20:55:10.713628948 +0800
 Birth: -
```

# wc (word count)

- 命令功能：计算行数、字数和字符数
- 命令格式： `wc [OPTION]... [FILE]...`
- 常用选项：
  - c 计算字符数
  - l 计算行数
  - w 计算字数

- 示例：

- ```
draco@ubuntu:~/Documents$ cat see
change at seeme
draco@ubuntu:~/Documents$ wc see
1  3 16 see
```

- ```
draco@ubuntu:~/Documents$ wc -l helloworld.c
6 helloworld.c
```

# find

- 命令功能：查找文件
- 命令格式：find path -option expression
- 常用选项：
  - name 按文件名查找
  - perm XXX 按权限查找
  - type f, d, l 按类型查找
  - user 按所有者查找
  - group 按用户组所有者查找
  - ctime -x +x 按创建时间查找
- 示例：

- ```
dracoking@ubuntu:~$ find ./Documents/ -name aa.txt
./Documents/aa/aa.txt
./Documents/newaa/aa.txt
```

- ```
dracoking@ubuntu:~$ find ./Documents/ -type d
./Documents/
./Documents/aa
./Documents/newaa
```

# 课后练习 1

- 在git仓库目录下创建目录{学号}\_Linux2024
- 在其中建立文件foo并将其内容变为自己的学号+空行+文件/etc/passwd的内容
- 修改foo的权限，使得当前用户只能读但不能写和执行该文件，而其他用户对其没有任何操作权限
- 设置foo的修改时间为2024年09月20日（其他不变）
- 建立一个secret目录，并在其中创建文件seeme,写入一些信息，使得除创建用户外任何人都不能看到这个文件（尝试用另外一个用户登录Linux系统，并查看seeme文件）



## 课后练习 2

- 在{学号}\_Linux2024目录下为seeme文件创建一个硬链接see-hlink
- 在see中输入内容 “change at see”，之后查看seeme的内容
- 使用另一个账户登录系统，分别查看see和seeme文件的内容
- 回到主用户登录系统，在{学号}\_Linux2024目录下为seeme创建一个符号链接see-slink，并修改seeme的内容为 “change at see-slink”
- 使用另一个账户登录系统，分别查看see和see-slink
- 回到主用户登录系统，将seeme文件删除，查看see-slink、see-hlink，哪个依然能够显示原先seeme的内容？
- 将{学号}\_Linux2024目录打包成一个压缩文档{学号}\_Linux2024.tar.gz并将其复制到 {学号}\_Linux2024目录中

# 作业提交

- ~~• Commit并push到自己的gitee仓库（如果没有先Fork）~~
- ~~• Pull request到课程git仓库的CH02分支~~
- 压缩包上传到gitee班级作业中（作业2）