Linux 文件属性

教师: 李军辉

1. Linux最优秀的地方之一，就在于他的多人多任务环境。

2. 而为了让各个使用者具有较保密的文件数据，因此文件的权限管理就变的很重要了。

3. Linux一般将文件可存取的身份分为三个类别，分别是owner/group/others，且三种身份各有 read/write/execute 等权限。

使用者与群组

- 文件拥有者
- 群组概念
- 其他人的概念

用户 (User)

- 使用Linux系统时, 需要以一个用户的身份登录.
- 运行一个程序时, 需要以一个用户的身份运行.
- .....

用户 (User)

- 使用Linux系统时, 需要以一个用户的身份登录.
- 运行一个程序时, 需要以一个用户的身份运行.
- .....

用户限制使用者或进程可以使用、不可以使用哪些资源.

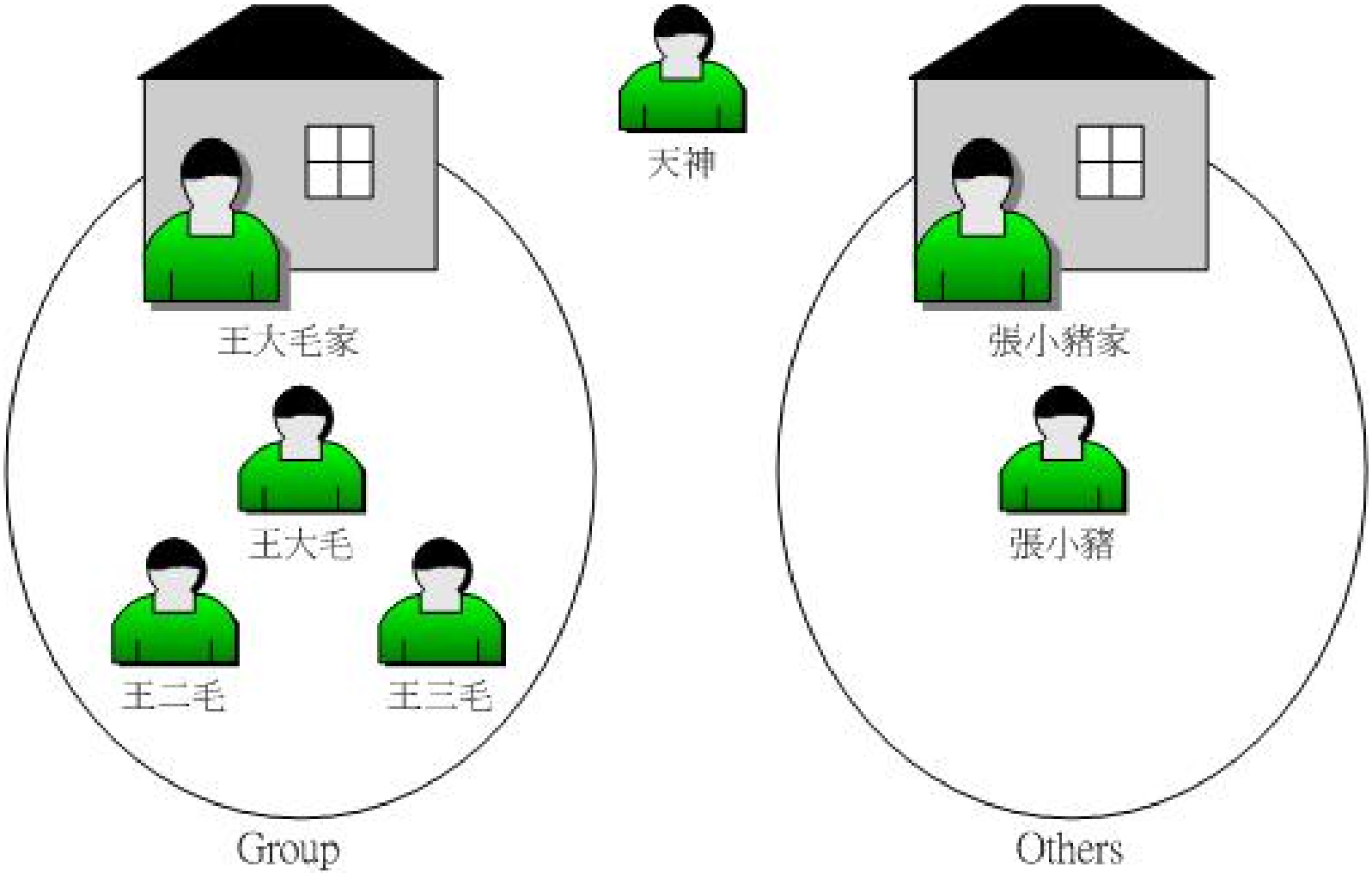
用户 (User)

- 用户UserID为32位, 从0开始. 但为了和老式系统兼容, 用户ID限制在60,000以下.
- 用户分为以下三种:
  - root 用户 (ID为 0的用户为root用户, 根用户, 超级用户)
  - 系统用户 (1 - 499)
  - 普通用户 (500以上)
- 系统中的文件都有一个所属用户及所属组.
- 使用id命令可以显示当前用户的信息.
- 使用passwd命令可以修改当前用户密码.

组 (Group)

组用来方便组织管理用户

- 每个用户拥用一个UserID, 操作系统实际使用的是用户UserID, 而非用户名.
- 每个用户属于一个主组, 属于一个或多个附属组
- 每个组拥有一个GroupID
- 每个进程以一个用户身份运行, 并受该用户可访问的资源限制
- 每个可登录的用户拥有一个指定的shell


天神
王大毛家
王大毛
王二毛
王三毛
Group
張小豬家
張小豬
Others

Linux 用户身份与群组记录的文件

1. /etc/passwd (所有账号及root用户信息)
2. /etc/shadow (对应的密码)
3. /etc/group (所有群组信息)

Linux 文件权限概念

Permission deny

```
~/Downloads$ cd android-sdk-linux/
-linux/: Permission denied
~/Downloads$
```

肯定是权限设定错误!!

Linux文件属性

```
lijunhui@lijunhui-OptiPlex-9020:~/Documents/Course/Unix-Linux$ ls -alh
total 125M
drwxrwxr-x  3 lijunhui lijunhui 4.0K Sep  4 13:08 .
drwxrwxr-x 12 lijunhui lijunhui 4.0K Sep  4 15:52 ..
drwxrwxr-x  3 lijunhui lijunhui 4.0K Sep  4 14:30 c1
-rw-r--r--  1 lijunhui lijunhui  224 Nov 29  2016 chkping.sh
-rw-r--r--  1 lijunhui lijunhui  123 Nov 29  2016 file.txt
-rw-rw-r--  1 lijunhui lijunhui  122 Nov 29  2016 uad.sh
-rw-rw-r--  1 lijunhui lijunhui 111M Sep  2  2016 UNIX-LINUX.pdf
-rw-rw-r--  1 lijunhui lijunhui   30 Nov 29  2016 users.txt
-rw-rw-r--  1 lijunhui lijunhui  14M Sep  3 16:09 vbird-linux-basic-4e.pdf
lijunhui@lijunhui-OptiPlex-9020:~/Documents/Course/Unix-Linux$
```

-rw-rw-r- - 1 lijunhui lijunhui 14M Sep 3 16:09 vbird-linux-basic-4e.pdf

Linux文件属性

```
lijunhui@lijunhui-OptiPlex-9020:~/Documents/Course/Unix-Linux$ ls -alh
total 125M
drwxrwxr-x  3 lijunhui lijunhui 4.0K Sep  4 13:08 .
drwxrwxr-x 12 lijunhui lijunhui 4.0K Sep  4 15:52 ..
drwxrwxr-x  3 lijunhui lijunhui 4.0K Sep  4 14:30 c1
-rw-r--r--  1 lijunhui lijunhui  224 Nov 29  2016 chkping.sh
-rw-r--r--  1 lijunhui lijunhui  123 Nov 29  2016 file.txt
-rw-rw-r--  1 lijunhui lijunhui  122 Nov 29  2016 uad.sh
-rw-rw-r--  1 lijunhui lijunhui 111M Sep  2  2016 UNIX-LINUX.pdf
-rw-rw-r--  1 lijunhui lijunhui   30 Nov 29  2016 users.txt
-rw-rw-r--  1 lijunhui lijunhui  14M Sep  3 16:09 vbird-linux-basic-4e.pdf
lijunhui@lijunhui-OptiPlex-9020:~/Documents/Course/Unix-Linux$
```

-rw-rw-r- - 1 lijunhui lijunhui 14M Sep 3 16:09 vbird-linux-basic-4e.pdf

[1] [2] [3] [4] [5] [6] [7]

Linux文件属性

```
lijunhui@lijunhui-OptiPlex-9020:~/Documents/Course/Unix-Linux$ ls -alh
total 125M
drwxrwxr-x  3 lijunhui lijunhui 4.0K Sep  4 13:08 .
drwxrwxr-x 12 lijunhui lijunhui 4.0K Sep  4 15:52 ..
drwxrwxr-x  3 lijunhui lijunhui 4.0K Sep  4 14:30 c1
-rw-r--r--  1 lijunhui lijunhui  224 Nov 29  2016 chkping.sh
-rw-r--r--  1 lijunhui lijunhui  123 Nov 29  2016 file.txt
-rw-rw-r--  1 lijunhui lijunhui  122 Nov 29  2016 uad.sh
-rw-rw-r--  1 lijunhui lijunhui 111M Sep  2  2016 UNIX-LINUX.pdf
-rw-rw-r--  1 lijunhui lijunhui   30 Nov 29  2016 users.txt
-rw-rw-r--  1 lijunhui lijunhui  14M Sep  3 16:09 vbird-linux-basic-4e.pdf
lijunhui@lijunhui-OptiPlex-9020:~/Documents/Course/Unix-Linux$
```

-rw-rw-r- - 1 lijunhui lijunhui 14M Sep 3 16:09 vbird-linux-basic-4e.pdf

[1] [2] [3] [4] [5] [6] [7]

[权限][连结][拥有者][群组][文件容量][修改日期][文件名]

-rw-r- -r- -

第一个字符代表这个文件是『目录、文件或链接文件等等』：

- 当为[-]则是文件；
- 当为[d]则是目录；
- 若是[l]则表示为连结档(link file)；
- 若是[b]则表示为装置文件里面的可供储存的接口设备(可随机存取装置)；
- 若是[c]则表示为装置文件里面的串行端口设备，例如键盘、鼠标(一次性读取装置)。

rwx

- r 代表可读(read)
- w 代表可写(write)
- x 代表可执行(execute)
- \- 代表没有权限

- 第一组为文件拥有者的权限
- 第二组为同群组的权限
- 第三组为其他非本群组的权限

问题1

若有一个文件的类型与权限数据为『-rwxr-xr–』，请说明其意义为何?

Linux文件属性

```
lijunhui@lijunhui-OptiPlex-9020:~/Documents/Course/Unix-Linux$ ls -alh
total 125M
drwxrwxr-x  3 lijunhui lijunhui 4.0K Sep  4 13:08 .
drwxrwxr-x 12 lijunhui lijunhui 4.0K Sep  4 15:52 ..
drwxrwxr-x  3 lijunhui lijunhui 4.0K Sep  4 14:30 c1
-rw-r--r--  1 lijunhui lijunhui  224 Nov 29  2016 chkping.sh
-rw-r--r--  1 lijunhui lijunhui  123 Nov 29  2016 file.txt
-rw-rw-r--  1 lijunhui lijunhui  122 Nov 29  2016 uad.sh
-rw-rw-r--  1 lijunhui lijunhui 111M Sep  2  2016 UNIX-LINUX.pdf
-rw-rw-r--  1 lijunhui lijunhui   30 Nov 29  2016 users.txt
-rw-rw-r--  1 lijunhui lijunhui  14M Sep  3 16:09 vbird-linux-basic-4e.pdf
lijunhui@lijunhui-OptiPlex-9020:~/Documents/Course/Unix-Linux$
```

-rw-rw-r- - 1 lijunhui lijunhui 14M Sep 3 16:09 vbird-linux-basic-4e.pdf

[1] [2] [3] [4] [5] [6] [7]

[权限][连结][拥有者][群组][文件容量][修改日期][文件名]

问题2

假设test1, test2, test3同属于testgroup这个群组，如果有下面的两个文件，请说明两个文件的拥有者与其相关的权限为何?

-rw-r- -r- - 1 root root 238 Jun 18 17:22 test.txt

-rwxr-xr- - 1 test1 testgroup 5238 Jun 19 10:25 ping_tsai

问题3

如果我的目录为底下的样式，请问testgroup这个群组的成员与其他人(others)是否可以进入本目录?

drwxr-xr- - 1 test1 testgroup 5238 Jun 19 10:25 groups/

Linux目录的权限

- r(Read，读取)：对文件而言，具有读取文件内容的权限；对目录来说，具有浏览目录的权限。
- w(Write,写入)：对文件而言，具有新增,修改,删除文件内容的权限；对目录来说，具有新建，删除，修改，移动目录内文件的权限。
- x(eXecute，执行)：对文件而言，具有执行文件的权限；对目录了来说该用户具有进入目录的权限。

Linux文件权限的重要性

与Windows系统不一样的是，在Linux系统当中，每一个文件都多加了很多的属性进来，尤其是群组的概念，这样有什么用途呢?

最大的用途是在『数据安全性』上面的。

1. 系统保护的功能

举个简单的例子，在你的系统中，关于系统服务的文件通常只有root才能读写或者是执行，例如/etc/shadow这一个账号管理的文件，由于该文件记录了你系统中所有账号的数据，因此是很重要的一个配置文件，当然不能让任何人读取(否则密码会被窃取啊)，只有root才能够来读取！所以该文件的权限就会成为[-rw- - - - - - -]!

2. 团队开发软件或数据共享的功能

此外，如果你有一个软件开发团队，在你的团队中，你希望每个人都可以使用某一些目录下的文件，而非你的团队的其他人则不予以开放呢？以上面的例子来说，testgroup的团队共有三个人，分别是test1, test2, test3，那么我就可以将团队所需的文件权限订为[-rwxrwx- - -]来提供给testgroup的工作团队使用！

3. 未将权限设定妥当的危害

再举个例子来说，如果你的目录权限没有作好的话，可能造成其他人都可以在你的系统上面乱搞！ 例如本来只有root才能做的开关机、ADSL的拨接程序、新增或删除用户等等的指令，若被你改成任何人都可以执行的话， 那么如果使用者不小心给你重新启动啦！ 重新拨接啦！ 等等的！ 那么你的系统不就会常常莫名其妙的挂掉！ 而且万一你的用户的密码被其他不明人士取得的话，只要他登入你的系统就可以轻而易举的执行一些root的工作

如何改变文件属性与权限

几个常用于群组、拥有者、各种身份的权限之修改的指令，如下所示：

- chgrp ： 改变文件所属群组
- chown ： 改变文件拥有者
- chmod ： 改变文件的权限, SUID, SGID, SBIT等等的特性

改变所属群组, chgrp

```
[root@www ~]# chgrp [-R] dirname/filename ...
选项与参数：
-R：进行递归(recursive)的持续变更，亦即连同次目录下的所有文件、目录
   都更新成为这个群组之意。常常用在变更某一目录内所有的文件之情况。
范例：
[root@www ~]# chgrp users install.log
[root@www ~]# ls -l
-rw-r--r--  1 root users 68495 Jun 25 08:53 install.log
[root@www ~]# chgrp testing install.log
chgrp: invalid group name `testing' <== 发生错误讯息啰~找不到这个群组名~
```

改变的组名必须要在/etc/group 文件内存在才行

改变拥有者, chown

```
[root@www ~]# chown [-R] 账号名称 文件或目录
[root@www ~]# chown [-R] 账号名称:组名 文件或目录
选项与参数：
-R : 进行递归(recursive)的持续变更，亦即连同次目录下的所有文件都变更

范例：将install.log的拥有者改为bin这个账号：
[root@www ~]# chown bin install.log
[root@www ~]# ls -l
-rw-r--r--  1 bin  users 68495 Jun 25 08:53 install.log

范例：将install.log的拥有者与群组改回为root：
[root@www ~]# chown root:root install.log
[root@www ~]# ls -l
-rw-r--r--  1 root root 68495 Jun 25 08:53 install.log
```

改变的组名必须要在/etc/passwd 文件内存在才行

改变权限, chmod

文件权限的改变使用的是chmod这个指令

1. 数字类型改变文件权限
2. 符号类型改变文件权限

数字类型改变文件权限

使用数字来代表各个权限

1. r:4
2. w:2
3. x:1

每种身份(owner/group/others)各自的三个权限(r/w/x)分数是需要累加的，例如当权限为： [-rwxrwx- - -] 分数则是：

- owner = rwx = 4+2+1 = 7
- group = rwx = 4+2+1 = 7
- others= - - - = 0+0+0 = 0

数字类型改变文件权限

```
[root@www ~]# ls -al .bashrc
-rw-r--r--  1 root root 395 Jul  4 11:45 .bashrc
[root@www ~]# chmod 777 .bashrc
[root@www ~]# ls -al .bashrc
-rwxrwxrwx  1 root root 395 Jul  4 11:45 .bashrc
```

问题4

将刚刚你的.bashrc这个文件的权限修改回-rw-r- -r- -的情况.

chmod ??? .bashrc

符号类型改变文件权限

chmod	u g o a	+(加入) –(除去) =(设定)	r w x	文件或目录

符号类型改变文件权限

```
[root@www ~]# chmod u=rwx,go=rx .bashrc
# 注意喔！那个 u=rwx,go=rx 是连在一起的，中间并没有任何空格！
[root@www ~]# ls -al .bashrc
-rwxr-xr-x  1 root root 395 Jul  4 11:45 .bashrc
```

符号类型改变文件权限

```
[root@www ~]# chmod a-x .bashrc
[root@www ~]# ls -al .bashrc
-rw-rw-rw-   1 root root 395 Jul   4 11:45 .bashrc
```

目录与文件之权限意义

文件权限对于一般文件与目录文件有何不同呢?

文件可以执行（即可执行文件），目录也可以执行吗?

权限对文件的重要性

- r (read)：可读取此一文件的实际内容，如读取文本文件的文字内容等；
- w (write)：可以编辑、新增或者是修改该文件的内容(但不含删除该文件)；
- x (execute)：该文件具有可以被系统执行的权限。

什么是可执行(x)?

在Windows下, 一个文件是否具有执行的能力是藉由『扩展名』来判断的

- .exe
- .bat
- .com
- .....

在Linux底下，我们的文件是否能被执行，则是藉由是否具有『x』这个权限来决定的！跟文件名是没有绝对的关系的！

权限对目录的重要性

- r (read contents in directory)
  - 表示具有读取目录结构列表的权限，所以当你具有读取(r)一个目录的权限时，表示你可以查询该目录下的文件名数据。所以你就可以利用 ls 这个指令将该目录的内容列表显示出来！

权限对目录的重要性

- r (read contents in directory)
  - 表示具有读取目录结构列表的权限，所以当你具有读取(r)一个目录的权限时，表示你可以查询该目录下的文件名数据。所以你就可以利用 ls 这个指令将该目录的内容列表显示出来！
- w (modify contents of directory)
  - 具有改动该目录结构列表的权限
  - 建立新的文件与目录；
  - 删除已经存在的文件与目录(不论该文件的权限为何！)
  - 将已存在的文件或目录进行更名；
  - 搬移该目录内的文件、目录位置。

权限对目录的重要性

- r (read contents in directory)
  - 表示具有读取目录结构列表的权限，所以当你具有读取(r)一个目录的权限时，表示你可以查询该目录下的文件名数据。所以你就可以利用 ls 这个指令将该目录的内容列表显示出来！
- w (modify contents of directory)
  - 具有改动该目录结构列表的权限
  - 建立新的文件与目录；
  - 删除已经存在的文件与目录(不论该文件的权限为何！)
  - 将已存在的文件或目录进行更名；
  - 搬移该目录内的文件、目录位置。
- x (access directory)
  - 用户能否进入该目录成为工作目录

问题5

有个目录的权限如下所示：

drwxr- -r- - 3 root root 4096 Jun 25 08:35 .ssh

系统有个账号名称为vbird，这个账号并没有支持root群组，请问vbird对这个目录有何权限？是否可切换到此目录中？

问题6

假设有个账号名称为dmtsai，他的家目录在/home/dmtsai/，dmtsai对此目录具有[rwx]的权限。若在此目录下有个名为the_root.data的文件，该文件的权限如下：

-rwx- - - - - - 1 root root 4365 Sep 19 23:20 the_root.data

请问dmtsai对此文件的权限为何？可否删除此文件？

问题7

假设有个名为test.txt的文件，该文件的权限如下: -rwx- - - - - - 1 test

test 4365 Sep 19 23:20 test.txt

请问哪些用户可以对该文件执行chmod、chgrp、chown操作?

问题8

假设有个名为/home/test/test.txt的文件，该文件的权限如下:

-rwxrwxrwx 1 test test 4365 Sep 19 23:20 test.txt

现在jhli用户通过cp命令将test.txt文件拷贝至/home/jhli/test.txt:

cp /home/test/test.txt /home/jhli/test.txt

请问生成的/home/jhli/test.txt文件权限是什么、拥有者和群组是什么?

问题9

假设有个名为/home/test/test.txt的文件，该文件的权限如下:

-rwxr–r– 1 test test 4365 Sep 19 23:20 test.txt

当前登录用户为jhli, 请问cat /home/test/test.txt是否总是能够运行成功?

问题10

当前登录用户为jhli, 在/home/jhli目录下执行 (假设该目录下不存在文件abc.txt和目录dir1):

```
touch abc.txt
mkdir dir1
```

请问abc.txt和dir1的权限分别是什么?

新增一个文件或目录时，默认的权限是什么？

文件默认权限: umask

- umask
- umask -S

文件默认权限: umask

- umask
- umask -S

举例

umask
0022　　与一般权限有关的是后面三个数字！第一个数字0表示8进制

umask -S
u=rwx,g=rx,o=rx

举例

umask
0022

umask -S
u=rwx,g=rx,o=rx umask 0002 (设置当前用户的umask值为0002)

在默认权限的属性上，目录与文件是不一样的。

- 我们知道 x 权限对於目录是非常重要的！
- 但是一般文件的创建则不应该有运行的权限

举例

umask
0022

umask -S
u=rwx,g=rx,o=rx umask 0002 (设置当前用户的umask值为0002)

在默认权限的属性上，目录与文件是不一样的。

- 我们知道 x 权限对於目录是非常重要的！
- 但是一般文件的创建则不应该有运行的权限

- 创建文件时：-rw-r- -r- -
- 创建目录时：drwxr-xr-x

思考题

假设你的 umask 为 003，请问该 umask 情况下，创建的文件与目录权限为？

思考题

假设你的 umask 为 003，请问该 umask 情况下，创建的文件与目录权限为？

文件: -rw-rw-r–
目录: drwxrwxr- -

文件的特殊属性(隐藏属性)

能否创建这样的文件:

- 只能往某文件追加数据, 不能删除内容
- 只能往某文件夹添加新的文件或子文件夹, 不能删除已有文件或子文件夹
- 不能删除、重命名文件名; 不能不能写入、新增数据

chattr命令

chattr [+-=][Asaci] [文件或者目录]

- A: 表示文件或目录的atime将不可修改;
- s: 会将数据同步写入磁盘中;
- a: 只能追加不能删除, 非root用户不能设定该属性;
- c: 自动压缩该文件, 读取时会自动解压;
- i: 文件不能删除、重命名、设定链接、写入以及新增数据, 非root用户不能设定该属性.

lsattr命令

lsattr [-aR] [文件/目录名]

- -a: 类似于ls的-a选项, 即连同隐藏文件一同列出.
- -R: 连同子目录的数据一同列出

特殊权限

除普通权限外, 还有三个特殊权限:

特殊权限

除普通权限外, 还有三个特殊权限:

- suid (文件) 以文件的所属用户身份执行, 而非执行文件的用户

特殊权限

除普通权限外, 还有三个特殊权限:

- suid (文件) 以文件的所属用户身份执行, 而非执行文件的用户
- sgid (文件) 以文件所属组身份执行;
  (目录) 在该目录中创建的任意新文件的所属组与该目录的所属组相同.

特殊权限

除普通权限外, 还有三个特殊权限:

- suid (文件) 以文件的所属用户身份执行, 而非执行文件的用户
- sgid (文件) 以文件所属组身份执行;
  (目录) 在该目录中创建的任意新文件的所属组与该目录的所属组相同.
- sticky (目录) 对目录拥有写入权限的用户仅可以删除其拥有的文件, 无法删除其他用户所拥有的文件.

设置特殊权限

- 设置 suid:
  - chmod u+s exp.pl
- 设置 sgid:
  - chmod g+s exp
- 设置 sticky:
  - chmod o+t exp

与普通权限一样, 特殊权限也可以使用数字表示方式:

- SUID = 4
- SGID = 2
- Sticky = 1

chmod 4755 exp

设置uid举例

passwd这个命令具有uid权限. 当普通用户执行passwd命令时, 可以临时获得root权限, 从而可以更改密码.

设置gid举例

任务：我的使用者 pro1, pro2, pro3 是同一个项目计划的开发人员，我想要让这三个用户在同一个目录底下工作，但这三个用户还是拥有自己的家目录与基本的私有群组。假设我要让这个项目计划在 /srv/projecta 目录下开发，可以如何进行?

第1步. 假设这三个账号都尚未创建，可先创建一个名为 projecta 的群组，

再让这三个用户加入其次要群组的支持即可：

```
groupadd projecta
useradd -G projecta -c “projecta user” pro1
useradd -G projecta -c “projecta user” pro2
useradd -G projecta -c “projecta user” pro3
echo “password” — passwd - -stdin pro1
echo “password” — passwd - -stdin pro2
echo “password” — passwd - -stdin pro3
```

第2步. 开始创建此项目的开发目录：

```
mkdir /srv/projecta

chgrp projecta /srv/projecta

chmod 2770 /srv/projecta

ll -d /srv/projecta
```

ACL 的使用

ACL 是 Access Control List 的缩写，主要的目的是在提供传统的 owner,group,others 的 read,write,execute 权限之外的细部权限配置。

ACL 可以针对单一使用者，单一文件或目录来进行 r,w,x 的权限规范，对于需要特殊权限的使用状况非常有帮助。

ACL可以:

- 使用者 (user)：可以针对使用者来配置权限；
- 群组 (group)：针对群组为对象来配置其权限；
- 默认属性 (mask)：还可以针对在该目录下在创建新文件/目录时，规范新数据的默认权限；

- getfacl: 取得某个文件/目录的 ACL 配置项目；
- setfacl：配置某个目录/文件的 ACL 规范。

针对特定用户的方式

创建一个文件, 并单独设置vbird1用户可以访问和修改这个文件。

```
touch acl_test1
ll acl_test1
setfacl -m u:vbird1:rx acl_test1
ll acl_test1

getfacl acl_test1
```

针对特定群组的方式

配置规范：『g:[群组列表]:[rwx]』，例如针对 mygroup1 的权限规范 rx ：

```
setfacl -m g:mygroup1:rx acl_test1
getfacl acl_test1
```

任务: 将前一小节任务二中 /srv/projecta 这个目录，让 myuser1 可以进入查阅，但 myuser1 不具有修改的权力。

任务: 将前一小节任务二中 /srv/projecta 这个目录，让 myuser1 可以进入查阅，但 myuser1 不具有修改的权力。

#1. 先测试myuser1用户对文件夹的/srv/projecta的权限
cd /srv/projecta

#2. 用管理员权限修改/srv/projecta的权限 setfacl -m u:myuser1:rx /srv/projecta
getfacl /srv/projecta

#3. 再要使用 myuser1 去测试看看结
cd /srv/projecta
ll -a
touch testing

权限的继承, 上面的做法myuser1可以进入/srv/projecta这个目录, 但在/srv/projecta目录中再建立新文件或新文件夹，新文件或新文件夹的acl权限并没有被继承。

```
setfacl -m d:u:myuser1:rx /srv/projecta
```

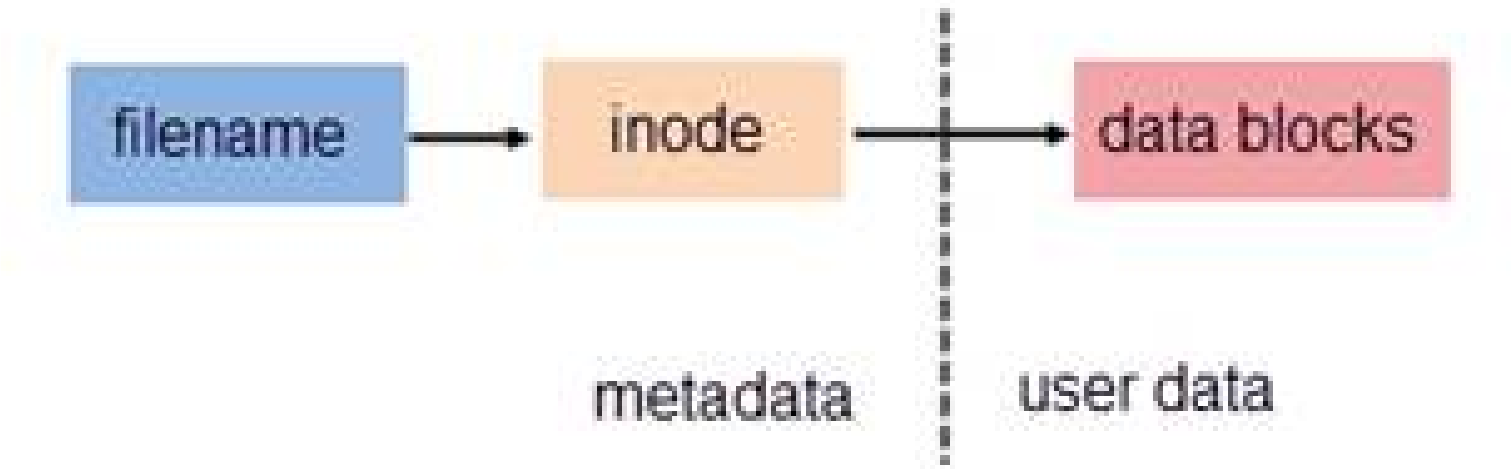
用户数据与元数据

文件都有文件名与数据

- 用户数据 (user data): 即文件数据块 (data block)，数据块是记录文件真实内容的地方
- 元数据 (metadata): 文件的附加属性，如文件大小、创建时间、所有者等信息

元数据中的 inode 号

- 文件元数据的一部分
- inode 号即索引节点号
- 才是文件的唯一标识而非文件名



stat 或 ls -i 查看文件inode

stat file1.tar.gz

mv file1.tar.gz file2.tar.gz

stat file2.tar.gz

实体链接与符号链接： ln

Hard Link (实体链接, 硬链接或实际链接)

Symbolic link (符号链接，亦即是快捷方式)

```
touch 1.txt 2.txt
ln 1.txt 1.txt.ln
ln 2.txt 2.txt.ln
ls -alh 1.txt 1.txt.ln 2.txt 2.txt.ln
rm 1.txt 2.txt
ls -alh 1.txt.ln 2.txt.ln
```

硬链接

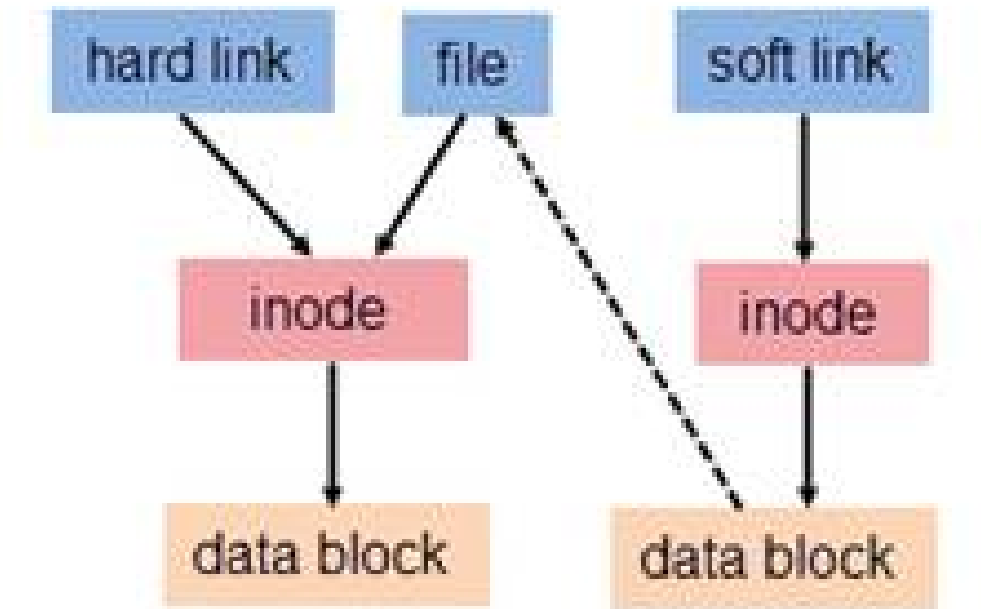
- 若一个 inode 号对应多个文件名，则称这些文件为硬链接。
- 硬链接就是同一个文件使用了多个别名

硬链接的好处:

- 文件有相同的 inode 及 data block；
- 只能对已存在的文件进行创建；
- 不能对目录进行创建，只可对文件创建；
- 删除一个硬链接文件并不影响其他有相同 inode 号的文件。

软链接

- 若文件用户数据块中存放的内容是另一文件的路径名的指向，则该文件就是软连接。
- 软链接有着自己的 inode 号以及用户数据块。



- 软链接有自己的文件属性及权限等；
- 可对不存在的文件或目录创建软链接；
- 软链接可交叉文件系统；
- 软链接可对文件或目录创建；
- 创建软链接时，链接计数 i_nlink 不会增加；
- 删除软链接并不影响被指向的文件，但若被指向的原文件被删除，则相关软连接被称为死链接。