GNU/Linux
用户权限管理

用户权限管理

Linux 下可以对文件 / 目录等进行权限，属主，属组进行管理。来确保文件，目录的安全性

对于权限的赋予 , 应遵循以下原则 :

1. 可给可不给的不给
2. 赋权仅给予最恰当的权限
3. 对于不确定的权限应不赋权
4. 如确需赋权可给予最小权限

用户权限管理

Linux 权限识别

权限：

权限	说明	对应数值	说明
r	读权限	4	赋予文件 / 目录可读权限
w	写权限	2	赋予文件 / 目录可修改、写、删除权限
x	执行权限	1	赋予文件可以执行权限，目录则为进入权限
-	拒绝权限	0	取消相关权限

用户权限管理

Linux 权限识别

通过权限可以看出：

1. Linux 最大权限为 777
2. 文件最大权限应为 666
3. 目录最大权限应为 777
4. 对于所需要的权限可以进行累加
5. 应注意 Linux 的权限实际为 8 进制

用户权限管理

Linux 的属主、数组、其他人解释

1. 属主 : 即文件 / 目录的主人 (u)

2. 属组 : 即文件 / 目录的主人所在组对其资源权限 (g)

3. 其他人 : 即非属主、数组对其资源的权限 (o)

4. 所有人 (全部用户): 即属主、属组、其他人 (a)

用户权限管理 -chmod

命令 :chmod

功能 : 改变用户 / 目录的权限

语法格式 :chmod [选项] < 目标 >

选项 :
-R: 递归改变目标权限
-v: 改变权限时候显示详细动作
-f: 忽略在修改文件时的错误信息

用户权限管理 -chmod

操作运算符 :
+: 增加某个权限
-: 去除某个权限
=: 只拥有某个权限

示例 :
1. 对 test.txt 文件的权限，属主增加 rw, 属组增加 rw, 其他人增加 r 权限

```
#chmod  u+rw,g+rw,o+r  test.txt
```

用户权限管理 -chmod

示例 :

2. 对 test.txt 文件的权限，属主减去 r, 属组只有 x, 其他人减去 r 权限

```
#chmod  u-w,g=x,o-r  test.txt
```

3. 对 test.txt 文件权限设置属主有 rwx, 属组 rx, 其他人没有任何权限 , 并显示详细信息。

```
#chmod  -v  750  test.txt
```

4. 对 files 目录及其子目录及文件设定属主 rwx, 属组 rx, 其他人没有任何权限

```
#chmod -R 750  ~/files
```

用户权限管理 -chmod

Set 位

suid: 让其他用户以属主身份操作文件 / 目录

sgid: 让其他用户以属组身份操作文件 / 目录

sticky: 黏贴位，确保用户仅可以删除自己的文件

用户权限管理 -chmod

示例：

1. 对 test.txt 设置 suid 位

   #chmod 4644 test.txt

2. 对 test.txt 设置 sgid 位

   #chmod 2644 test.txt

3. 对 linuxtmp 目录进行 sticky 设置

   #chmod 1777 /linuxtmp

用户权限管理 -chmod

系统默认情况下的权限为
特权账户的默认权限为
umask 0022
即
文件创建默认权限为 :644
创建目录的默认权限为 :755

用户权限管理 -chmod

系统默认情况下的权限为
普通账户的默认权限为
umask 0002
即
文件创建默认权限为 :664
创建目录的默认权限为 :775

两者默认内容规定详见 /etc/profile 及 /etc/bashrc

查看当前账户的 umask 权限
#umask

用户权限管理 -chmod

如需要改变 umask 的默认权限可以

1. 临时修改

   #umask　新的权限

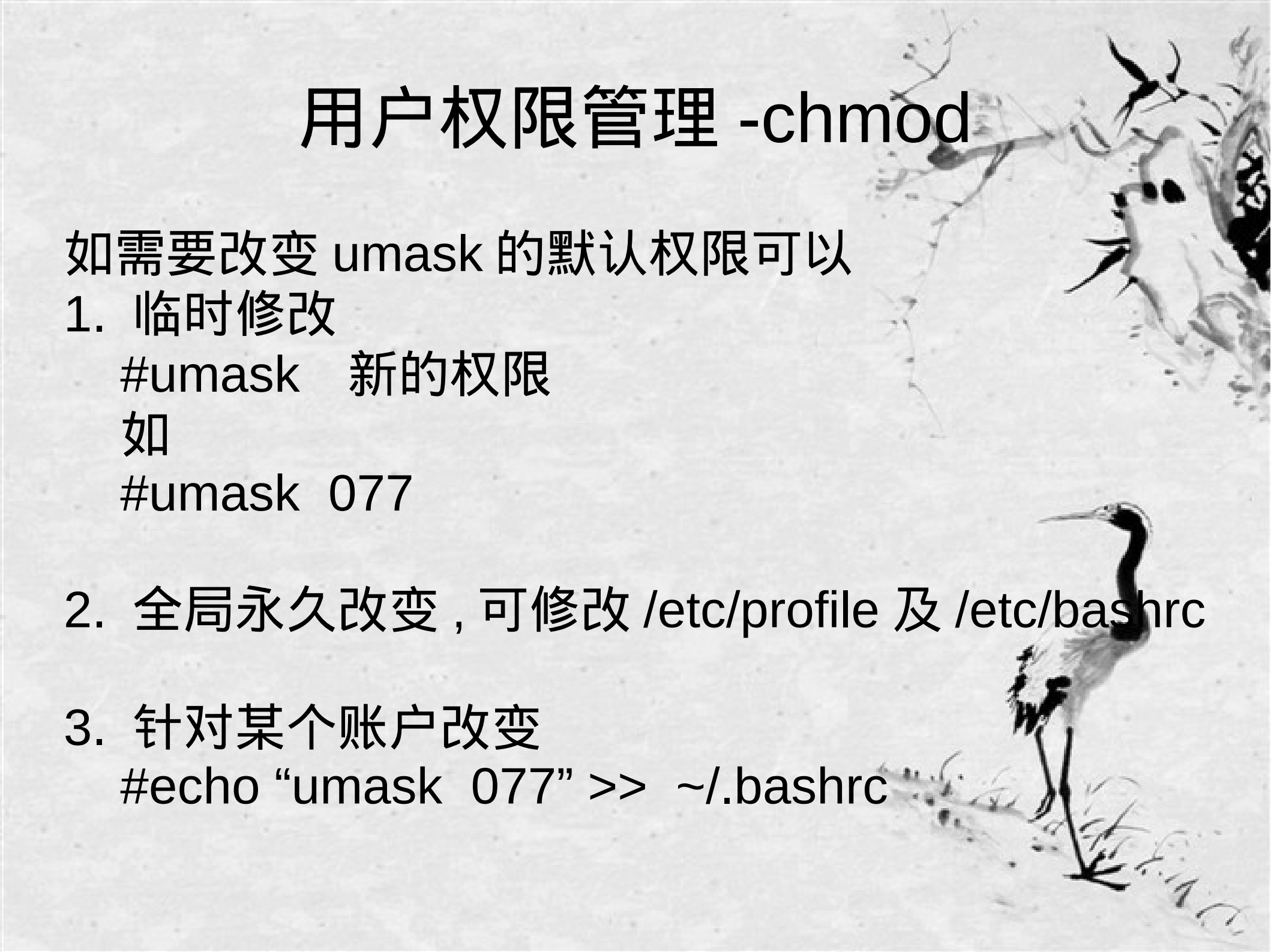
   如

   #umask 077

2. 全局永久改变，可修改 /etc/profile 及 /etc/bashrc

3. 针对某个账户改变

   #echo “umask 077” >> ~/.bashrc



用户权限管理 -chown

命令 :chown

功能 : 改变文件 / 目录的属主和 (或) 属组

语法格式 :chown [选项] [属主 . 属组] < 目标 >

选项 :
-R: 递归
-v: 执行时显示详细信息

用户权限管理 -chown

示例 :

1. 改变 test.txt 的属主为 lisa 账户

   #chown lisa test.txt

2. 改变 test.txt 的属组为 thomas 组

   #chown .thomas test.txt

3. 改变 files 目录及其子目录，所有文件的属主为 snow, 属组为 nilu, 并显示更改动作

   #chown -Rv snow.niliu files

用户权限管理 -chgrp

命令 :chgrp

功能 : 改变文件 / 目标的属组

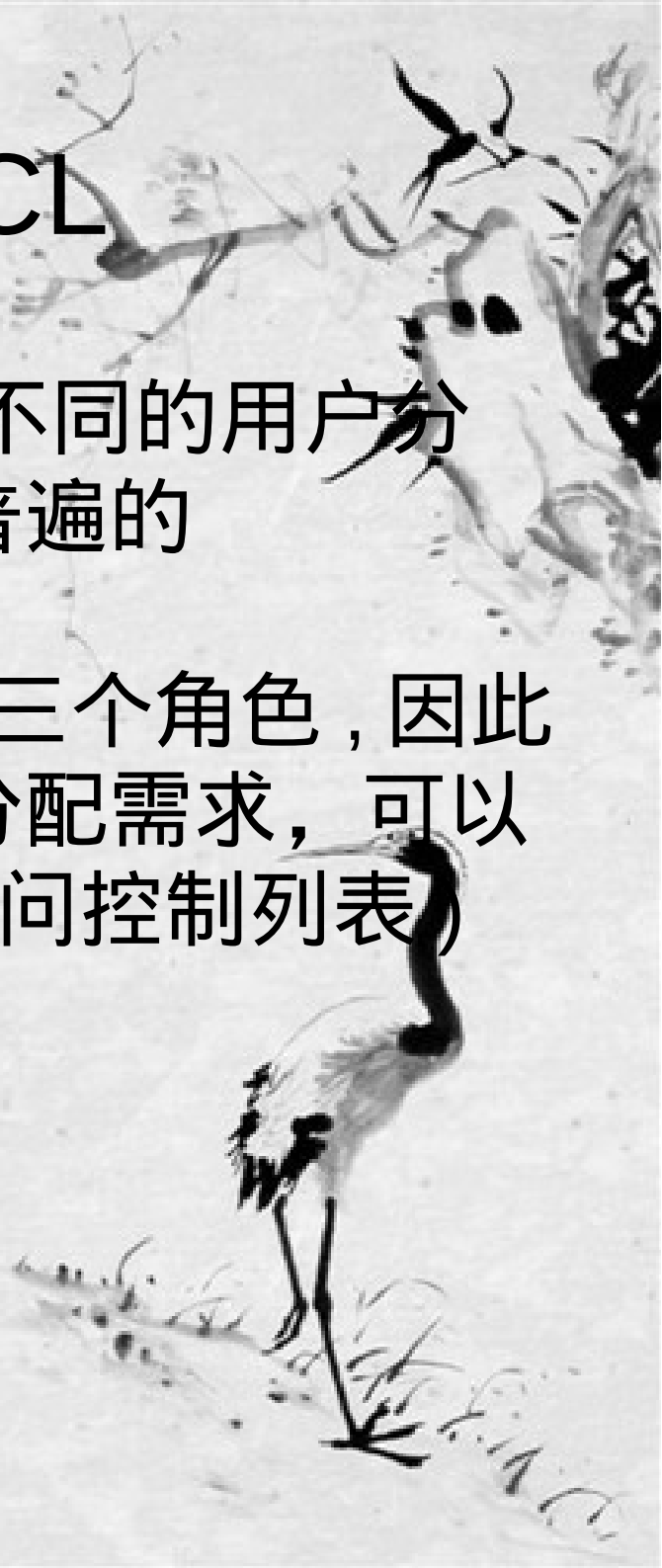
语法格式 :chgrp [选项] [属组] < 目标 >

选项 :
-R: 递归
-v: 执行时显示详细信息

用户权限管理 -ACL

对现今的操作系统而言，对于不同的用户分配不同的权限或有一定差别是非常普遍的

而 Linux 的 chmod 却只能分配三个角色，因此为了能够完成多个角色的不同权限分配需求，可以使用 ACL(Access Control List---- 访问控制列表)



用户权限管理 -ACL

ACL 的种类

1) 存取 ACL(Access ACL), 针对文件 / 目录设置访问控制列表

2) 默认 ACL(Default ACL), 只针对目录设置。如果目录中没有设置 ACL 将自动使用默认 ACL

对于 ACL 而言就是设定某个特定的账户或组对某个文件 / 目录的操作权限

用户权限管理 -ACL

命令 :getfacl

功能 : 查看本地文件 / 目录的 ACL 权限

语法格式 :getfacl [选项] < 目标 >

示例 :

1. 查看本地的文件 test.txt 的 ACL 权限

```
#getfacl  test.txt
```

用户权限管理 -ACL

命令 :setfacl

功能 : 修改本地文件 / 目录的 ACL 权限

语法格式 :setfacl [选项] < 目标 >

用户权限管理 -ACL

选项：

常用参数	参数说明
-m	设定文件 ACL 规则
-M	从文件或标准输入读取 ACL 规则并设定
-x	删除文件 ACL 规则
-X	从文件或标准输入读取 ACL 规则并删除
-b	删除所有扩展的 ACL 规则，基本的 ACL 规则将被保留
-k	删除缺省的 ACL 规则，如没有缺省规则将不提示
-d	设定默认的 ACL 规则
-test	测试并列出 ACL 规则
-R	递归对所有文件及目录进行操作
-L	跟踪符号链接，直指目标目录
-P	跳过所有符号链接，包括符号链接文件
--help	帮助

用户权限管理 -ACL

示例 :

1. 确认本地分区允许进行 ACL 设定

```
#tune2fs -l /dev/sdax  | grep option
```

2. 让 lisa 对 test.txt 拥有 rw 权限

```
#setfacl  -m  u:lisa:rw-  test.txt
#getfacl  test.txt
```

3. 为 test.txt 增加 thomas 账户 ,niliu 组添加 rw 权限

```
#setfacl -m u:thomas:rw-,g:niliu:rw-  test.txt
```

用户权限管理 -ACL

示例 :

4. 对 files 目录及子目录与文件增加账户 snow 的 rwx 权限

```
#setfacl -R  -m u:snow:rwx  files
```

5. 去掉 lisa 对 test.txt 的 ACL 权限

```
#setfacl -x u:lisa  test.txt
```

6. 去掉所有的 test.txt 中的 ACL 权限

```
#setfacl -b test.txt
```

用户权限管理 -ACL

示例：

7. 对 files 目录设定继承权限，未来有新建的子目录与文件将实现自动继承 ACL

```
#setfacl  -d  --set  u:lisa:rwx,g:niliu:rwx files
```

或

```
#setfacl -m d:u:lisa:rw files
#getfacl
```

8. 将 a.txt 的权限，属主，属组等信息作为 b.txt 的模板

```
getfacl file-A | setfacl --set-file=- file-B
```

用户权限管理 - 文件属性

文件有其自己的属性，根据属性的设定也可以实现限制用户的部分行为 .

命令 :lsattr

功能 : 显示文件 / 目录的权限

语法格式 :lsattr [选项] < 目标 >

用户权限管理 - 文件属性

选项 :
-a: 显示指定目录下的文件及目录的属性

-R: 递归显示

示例 :
1. 查看 /root 下的文件及目录权限
    #lsattr -aR /root

用户权限管理 - 文件属性

命令 :chattr

功能 : 更改文件 / 目录的属性

语法格式 :charrt ＜选项＞ ＜属性＞ ＜目标＞

选项：
-R: 递归
-V: 执行时显示详细动作

用户权限管理 - 文件属性

操作方法 :

+: 增加某个属性

-: 去除某个属性

=: 只拥有某个属性

用户权限管理 - 文件属性

属性 :

i: 只读属性

A: 不更新文件的访问时间 . 可减少磁盘 IO 操作

a: 可追加数据，但无法修改、删除、重命名

c: 压缩 , 文件使用时自动解压 , 离开文件 / 目录时自动压缩，以节省使用空间

用户权限管理 - 文件属性

属性 :

d: 忽略 dump 备份 , 文件有此属性将不允许备份

S:sync 同步

s: 安全删除属性 , 文件不可恢复

u: 文件删除后，可恢复

用户权限管理 - 文件属性

示例 :

1. 将文件 test.txt 增加 i 属性 , 并测试

```
#chattr  +i  test.txt
```

用户权限管理 -su

命令 :su

功能 : 切换当前用户身份

特点 :

1. 管理员切换至普通用户 , 不需要普通用户密码
2. 普通用户切换至管理员 , 需要管理员密码
3. 普通用户切换至普通用户 , 需要对方的密码

语法格式 : su [-] [账户名]

用户权限管理 -su

示例 :

1. root 切换至 snow 用户 , 且继续使用 root 的 shell 环境

```
#su  snow
```

2. root 切换至 snow 用户 , 并使用 snow 的 shell 环境

```
#su  -  snow
```

用户权限管理 -sudo

命令 :sudo

功能 : 赋予普通用户特殊权限

特点 :

1. 普通用户可以通过特权账户赋予特殊权限而完成管理操作

2. 普通用户操作特权时，只需要知道自己的密码即可执行操作

用户权限管理 -sudo

配置文件及所在路径

sudo 的配置文件位于 /etc/sudoers, 打开配置文件的方法为 :

#visudo

配置文件说明

Host_Alias 为主机别名

User_Alias 为用户别名

用户权限管理 -sudo

配置文件说明

Cmnd_Alias 为命令别名

别名定义格式

关键语句 别名名称 (必须大写) = 值

如 :

Host_Alias NILIUHOSTS = localhost,niliu

User_Alias NILIUUSERS = lisa,snow,%niliu

用户权限管理 -sudo

如 :

Cmnd_Alias NILIUCMDS = /sbin/mkfs,/sbin/fdisk

组调用 :

%niliu <- 代表某个用户组

用户权限管理 -sudo

运算符

!: 可以对任何东西取反，除了上述的，包括别名本身。

*: 代替任意个任何字符。

?: 代替单个任何字符。

[...] : 匹配括号中的任何一个字符。

用户权限管理 -sudo

sudoer 配置格式

用户　主机　=　命令

用户权限管理 -sudo

示例 :

1. 将普通用户变为 root

   #visudo

   在配置中增加如下行

   snow　ALL=ALL

2. 赋予别名 NILIUUSER 可以完成 mkfs

   #visudo

   再配置中增加如下行

   NILIUUSER ALL=/sbin/mkfs

用户权限管理 -sudo

示例 :

3. 多别名使用

NILIUUSER NILIUHOSTS = NILIUCMDS

命令使用 :

1. 将 snow 账户通过 su 到 root

$sudo su

2. 使用 snow 账户进行格式化

$sudo /sbin/mkfs -t ext4 /dev/sdb1

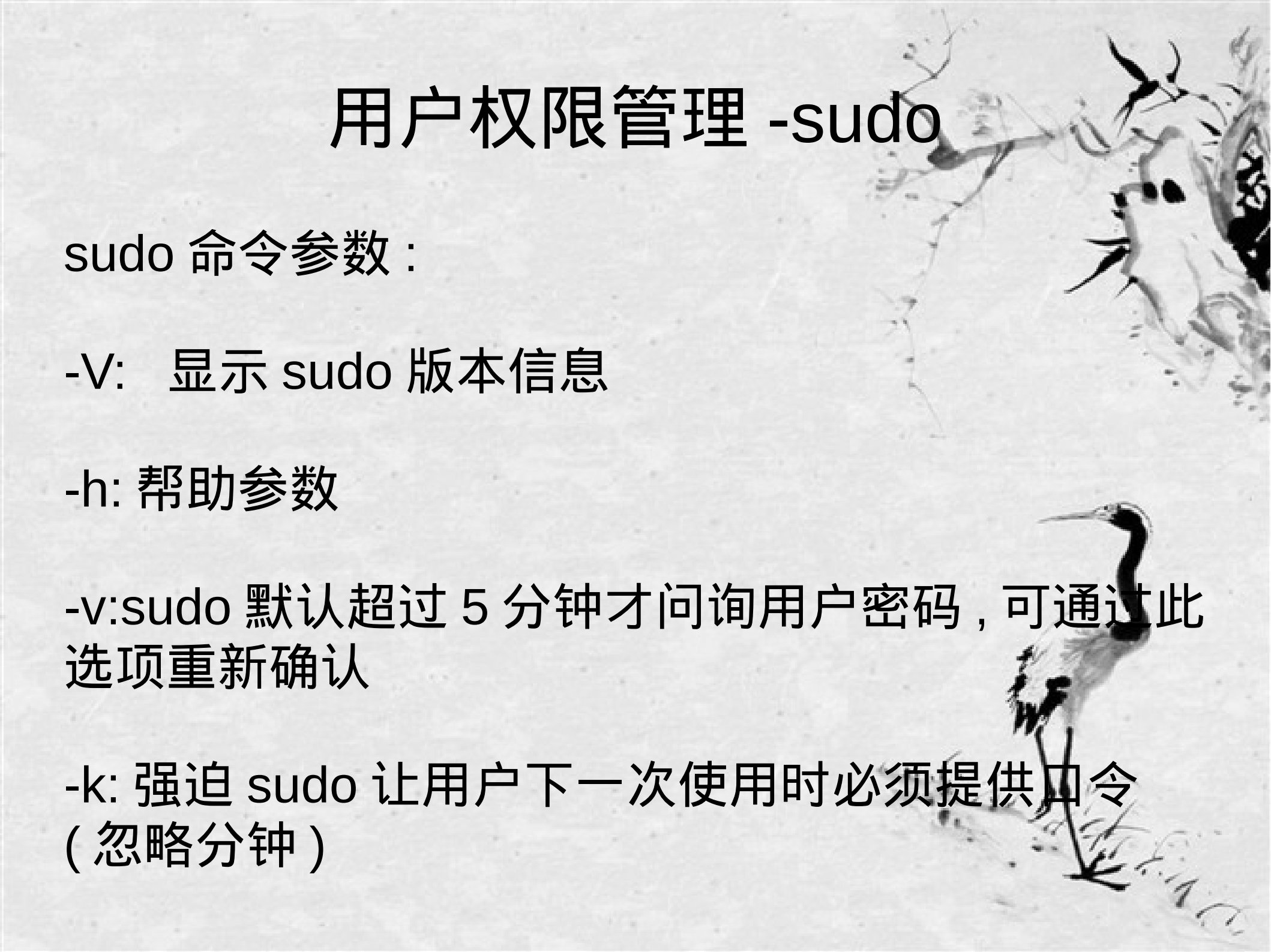
用户权限管理 -sudo

示例：

4. 多别名使用，并不需要用户输入密码

NILIUUSER NILIUHOSTS = NILIUCMDS NOPASSWD:ALL

用户权限管理 -sudo



sudo 命令参数 :

-V:　显示 sudo 版本信息

-h: 帮助参数

-v:sudo 默认超过 5 分钟才问询用户密码 , 可通过此选项重新确认

-k: 强迫 sudo 让用户下一次使用时必须提供口令 (忽略分钟)

用户权限管理 -sudo

sudo 命令参数 :

-b: 将指令放入后台执行

-u: 指定以某个 UID 执行此程序

-g: 指定以某个 GID 执行此程序

-s: 指定使用的 shell

-H: 将当前主目录变更为变更身份的主目录