**1. bash脚本之前加上:**

set -euxo pipefail

**2. .linux添加用户和分组**

(1):sudo adduser test1（交互式添加用户）

(2):sudo useradd test2(创建用户时可跟更多选项)

(3):sudo groupadd

(4):sudo passwd test2（更改密码）

(5):sudo groupmod（连按三下tab，查看已存在的分组）

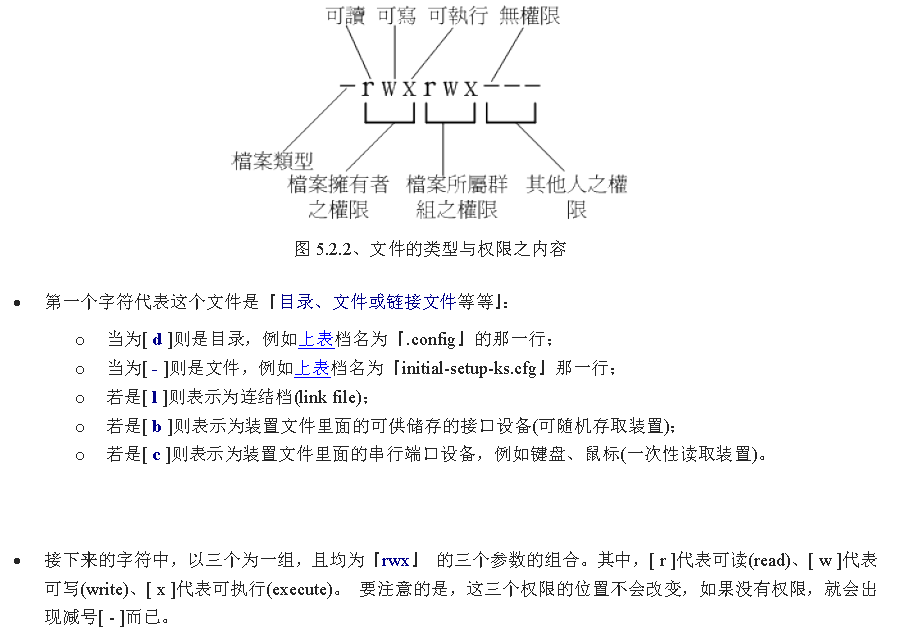
(6):sudo usermod -g root test2（将test2归为root组）

(7):id test2（查看test2所属组）

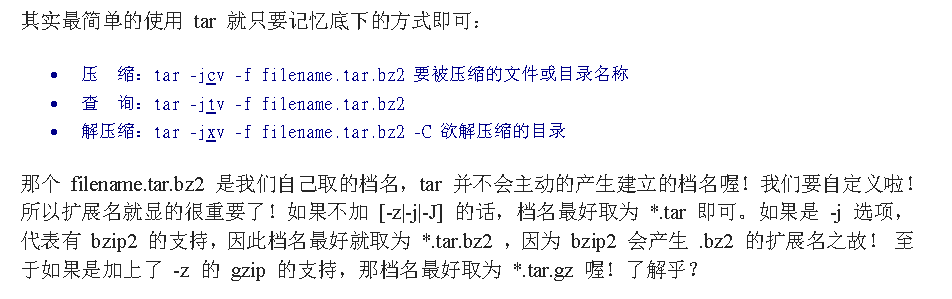
(8): chown 用户名:组名 filename（更改filename文件的拥有着和组名，常用于root将文件从一个用户目录cp到另一个用户目录，再更改该文件所属用户和组,可能还要再执行chmod来更改文件的权限）

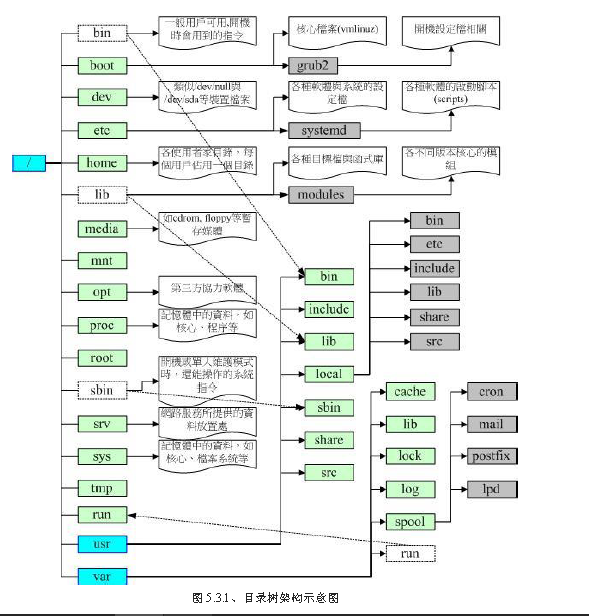
(9):sudo userdel(或groupdel) test2（删除用户或分组）

(10).chgrp -v group1 filename(将filename文件改为group1群组，可用ll查看)，加上-R 修改整个目录

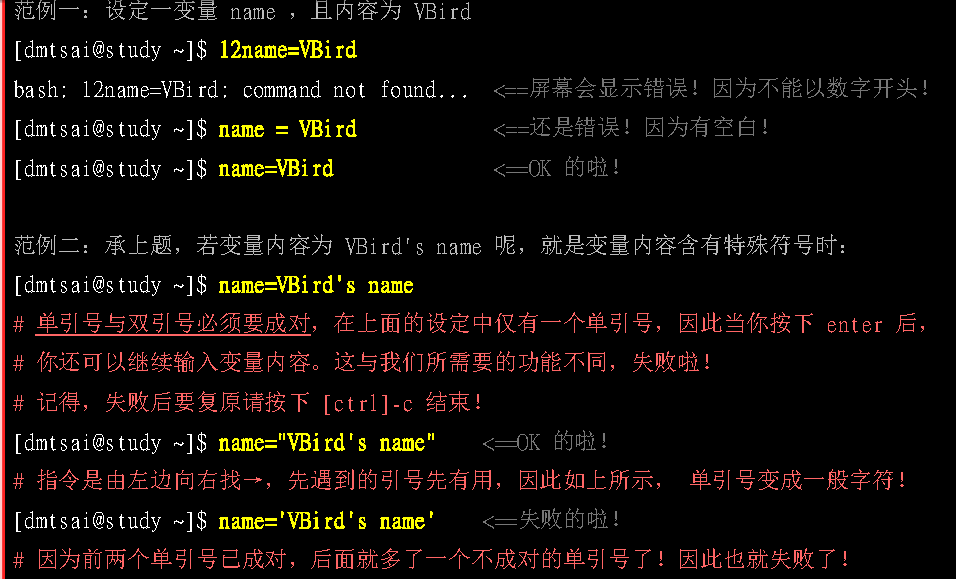


**3. 压缩**

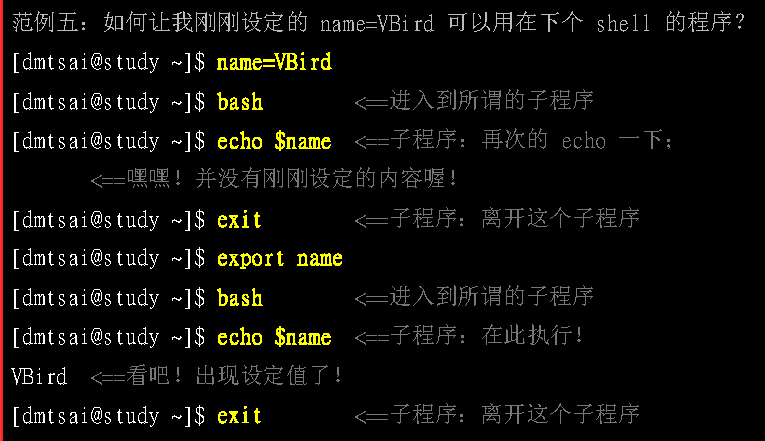




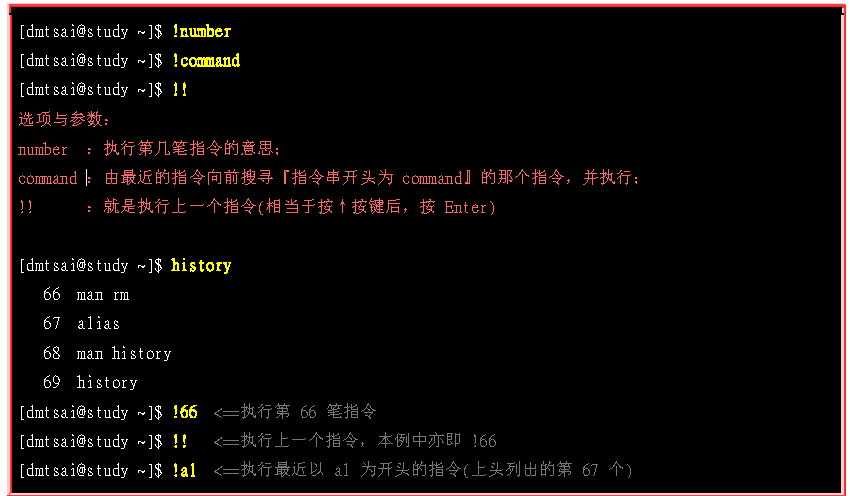
**4.bash shell**







**history和! 指令：**



**5.讯息撷取命令：cut、grep**

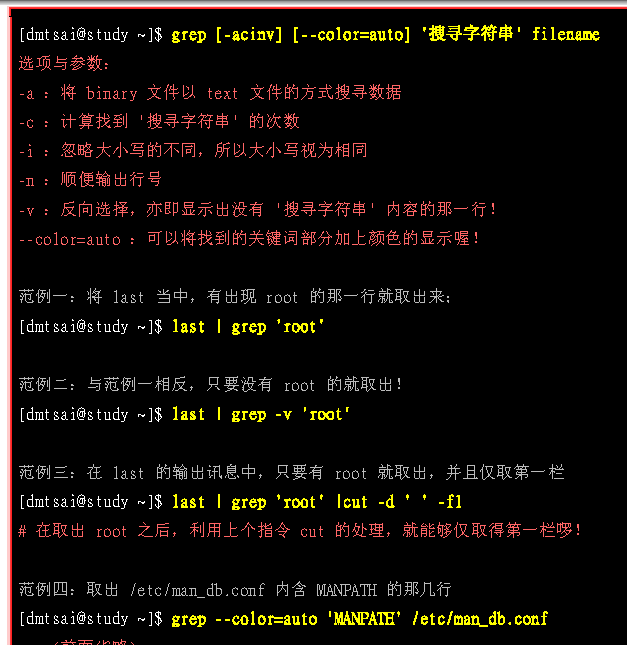
(1). cut -d ‘分割字符’ -f fields（表示取出第几段）

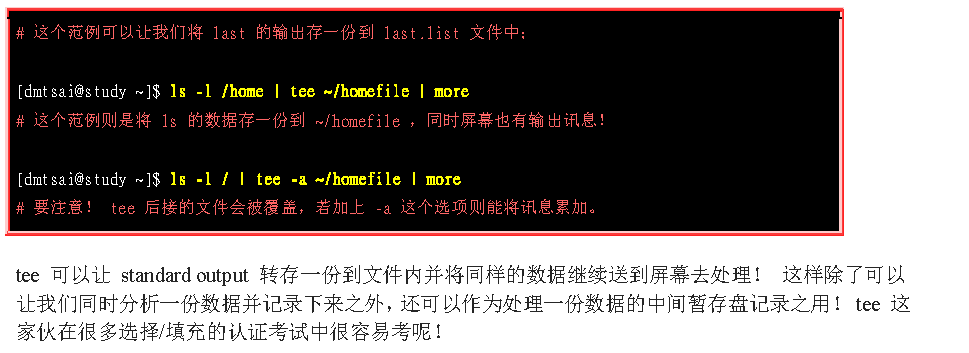
(2). cut -c 字符区间 （以字符为单位取出固定字符区间）



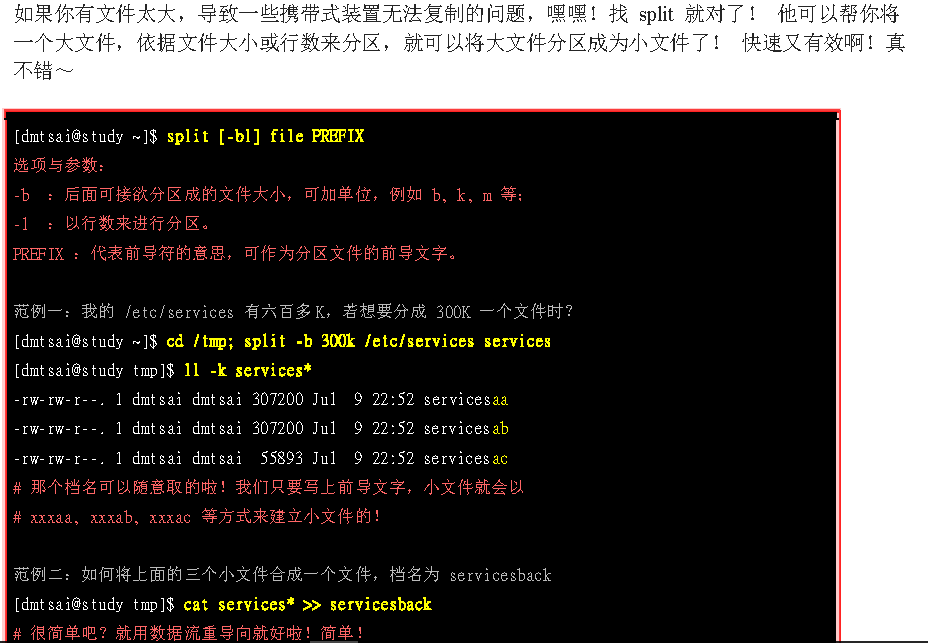
cut的主要用途在于将“同一行里面的数据进行分解！”最常用在分析一些文字数据如log日志的时候，不过cut在处理多空格相连的数据时不太方便，这是可以用awk来取代。

**Grep:** 刚刚的cut是将一行讯息当中取出某部分我们想要的，而grep则是分析一行讯息，若当中由我们所需要的信息，则将该行拿出来。

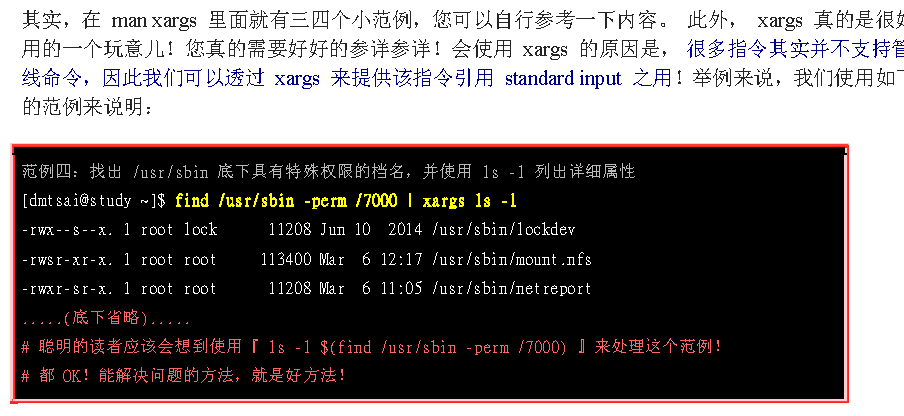


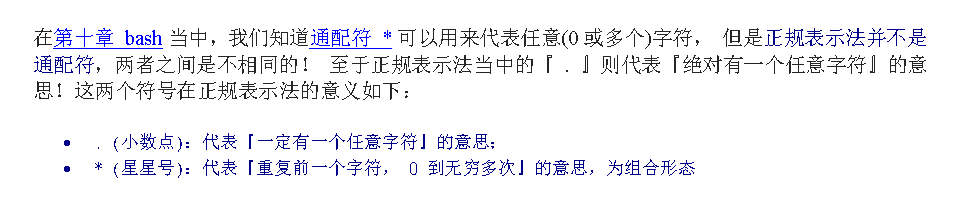


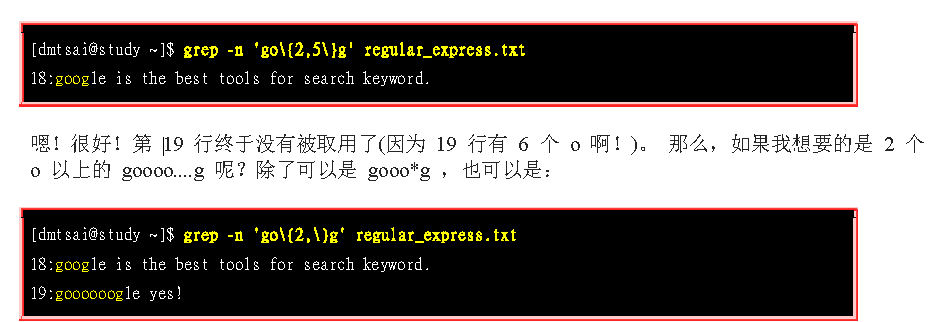
**Split:**



**Xargs:参数代换**







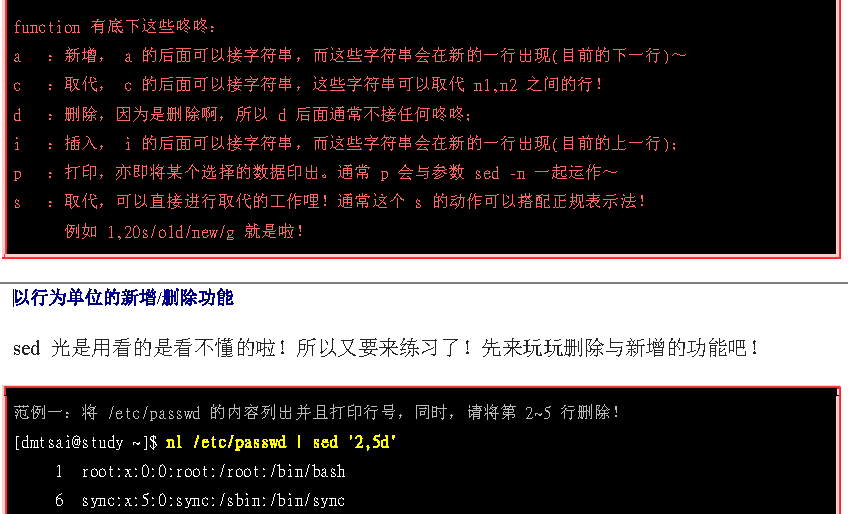
其中{ }用来限定\*能匹配的连续字符范围，并且需要用\来消除花括号的特殊含义。

grep ‘^[a-z]’：表示以小写字母开头

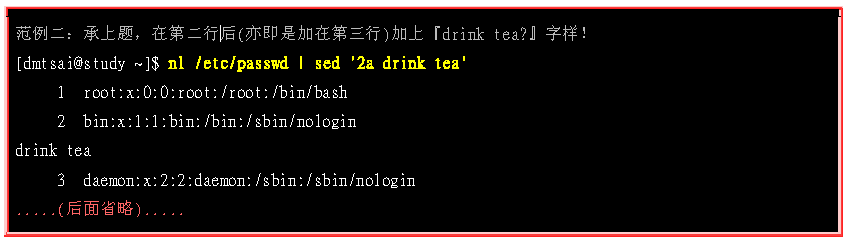
grep ‘[^a-z]’ : ^在集合里面则表示反向选择，即不以小写字母开头

grep ‘^$’：找出空白行,其中$为行尾字符

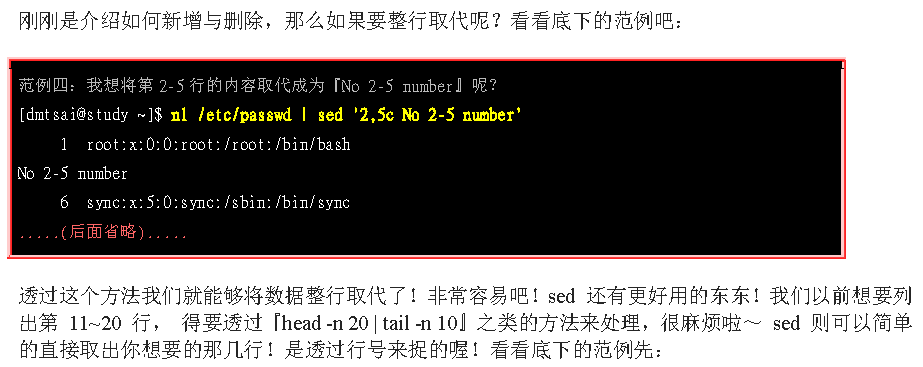
**Sed：function表示跟在数字后面，还可以带参数，比如-i 表示直接修改文件而非屏幕输出**



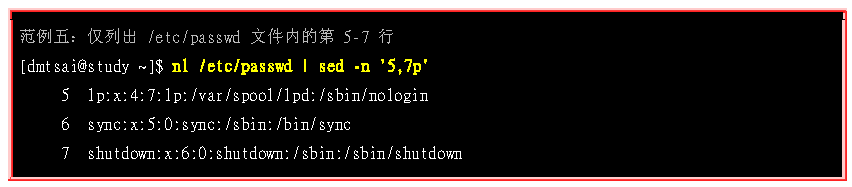
若要删除2至最后一列则将5换成$即可。 下面是添加多行，将a换成i则是在行前添加，而 sed -i **’$a 字符串’** filename则表示在文件行尾新增字符串

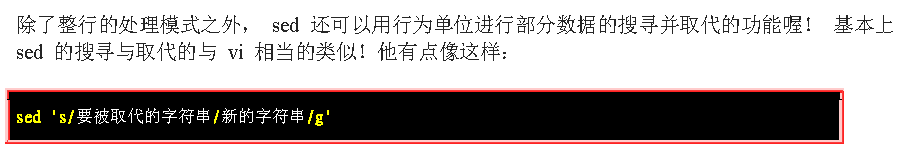


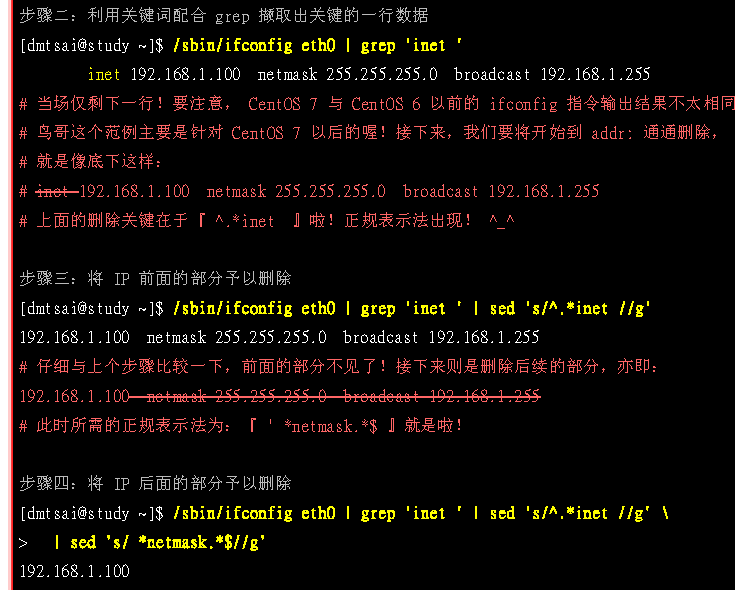
替换：



-n为安静模式，不加的话5到7行会重复输出

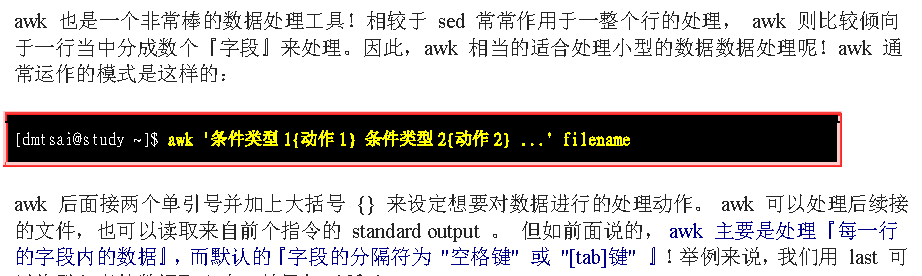


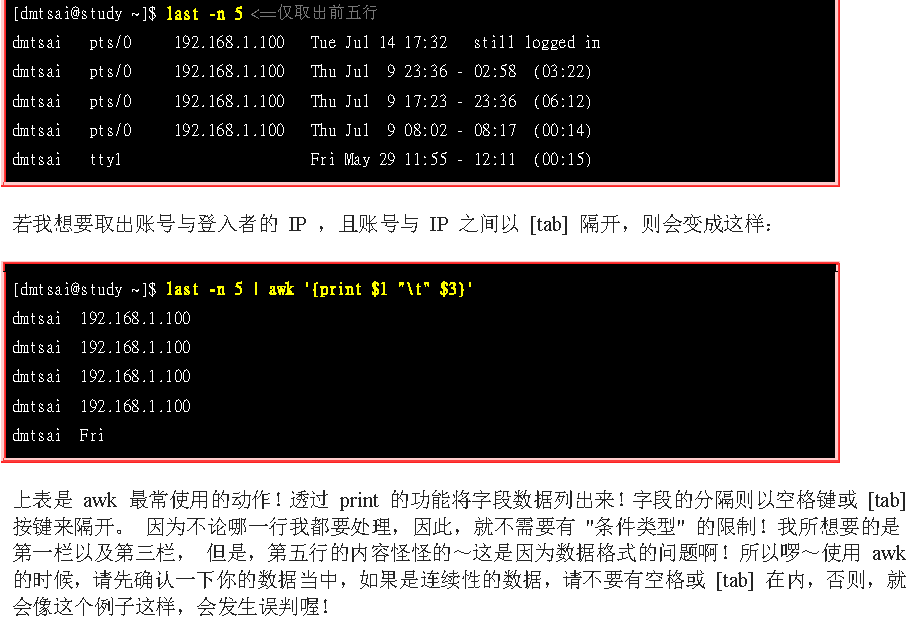


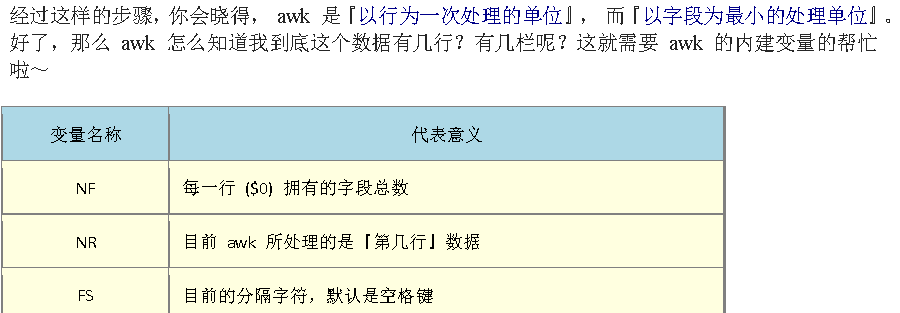


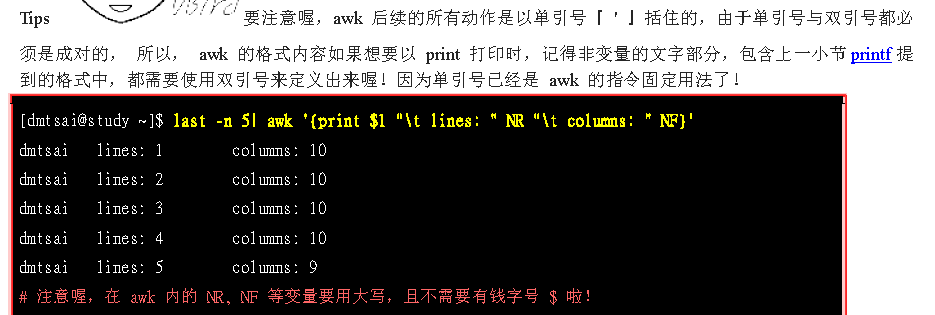
还可以用 sed ‘/^$/d’ 来删除空白行

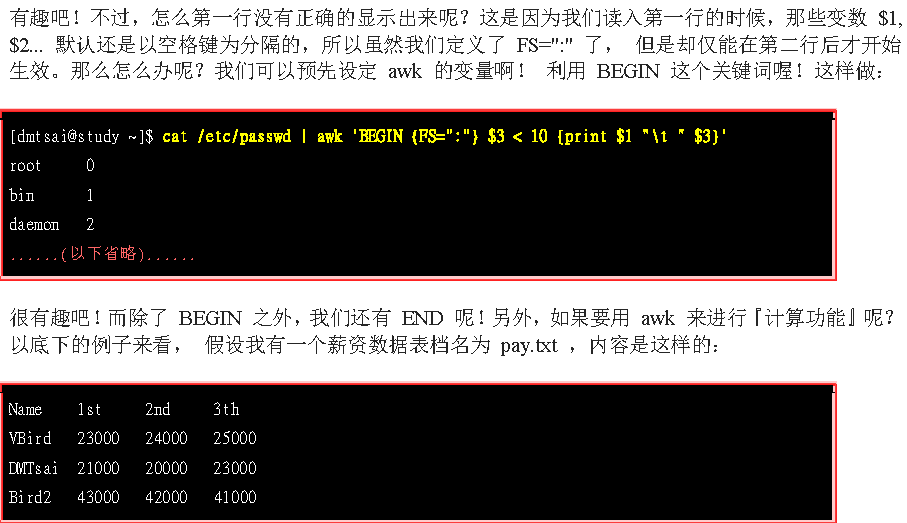
Awk:

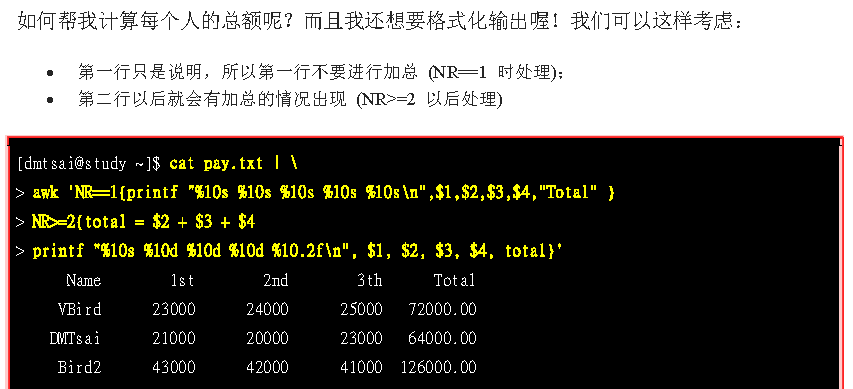




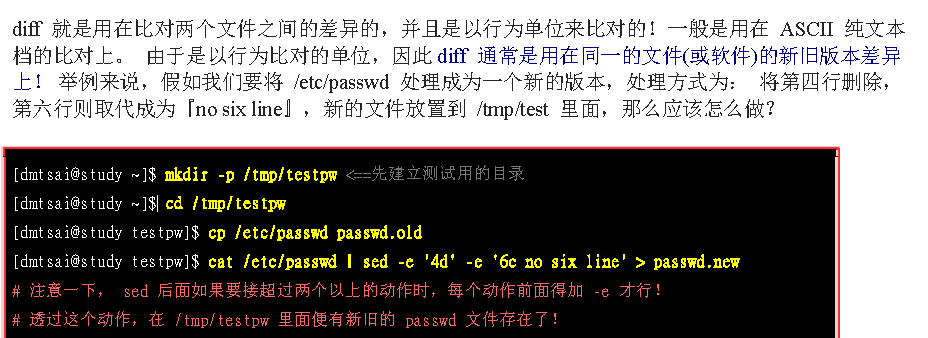




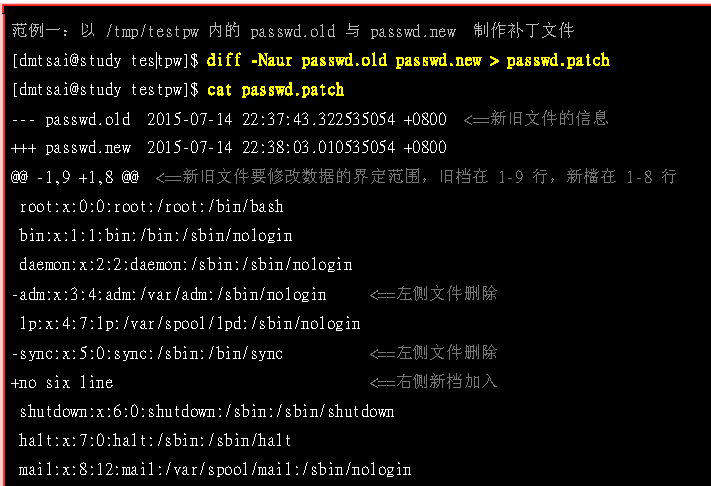
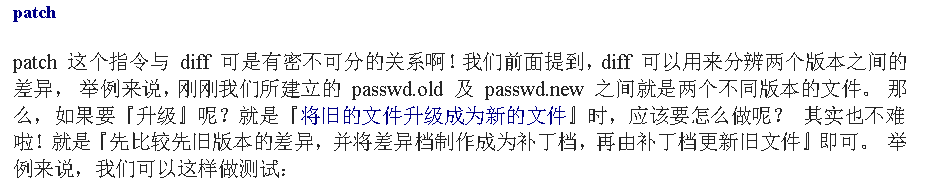


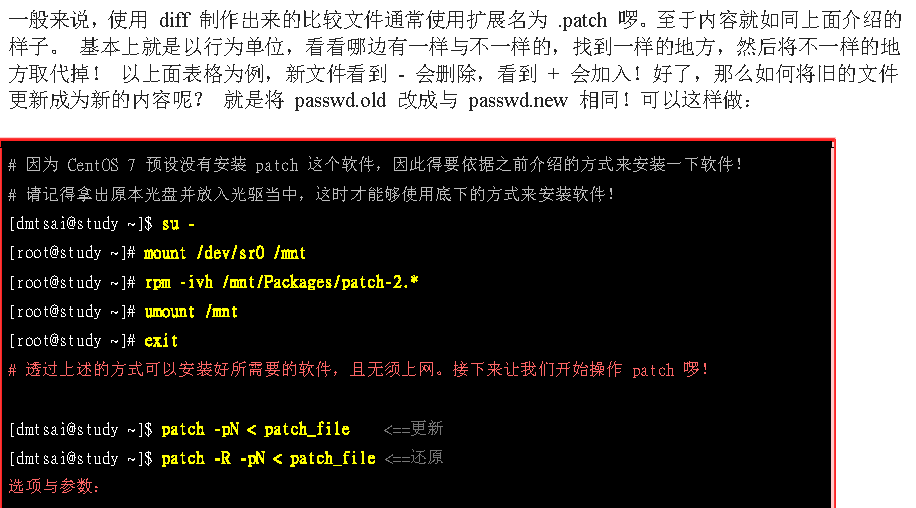


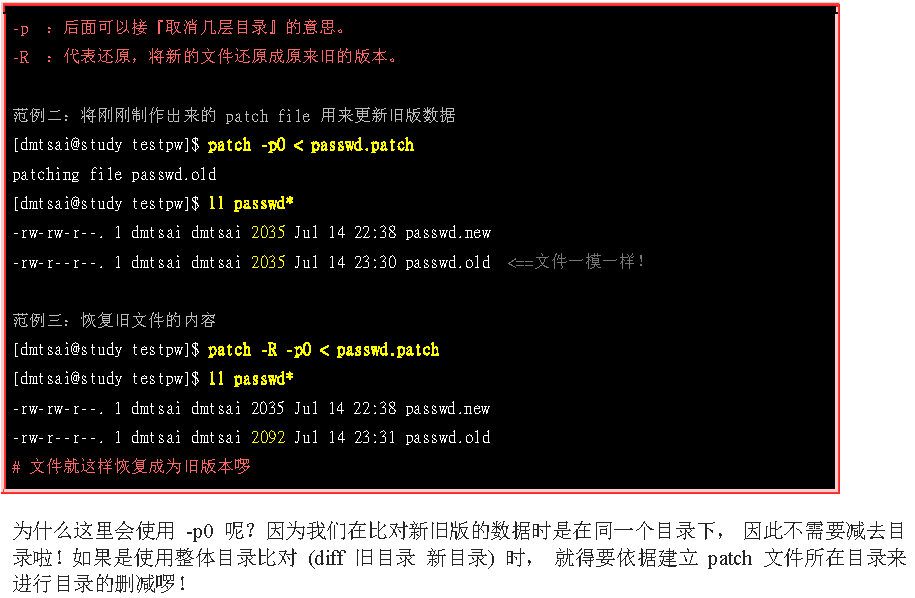
文件比对工具: diff，用在ASCII纯文本，以行为单位； cmp可用来比对非纯文档。











cat /proc/meminfo : 查看内存信息

/etc/passwd文件包含了所有系统用户账户列表以及每个用户的基本配置信息。如下条目：

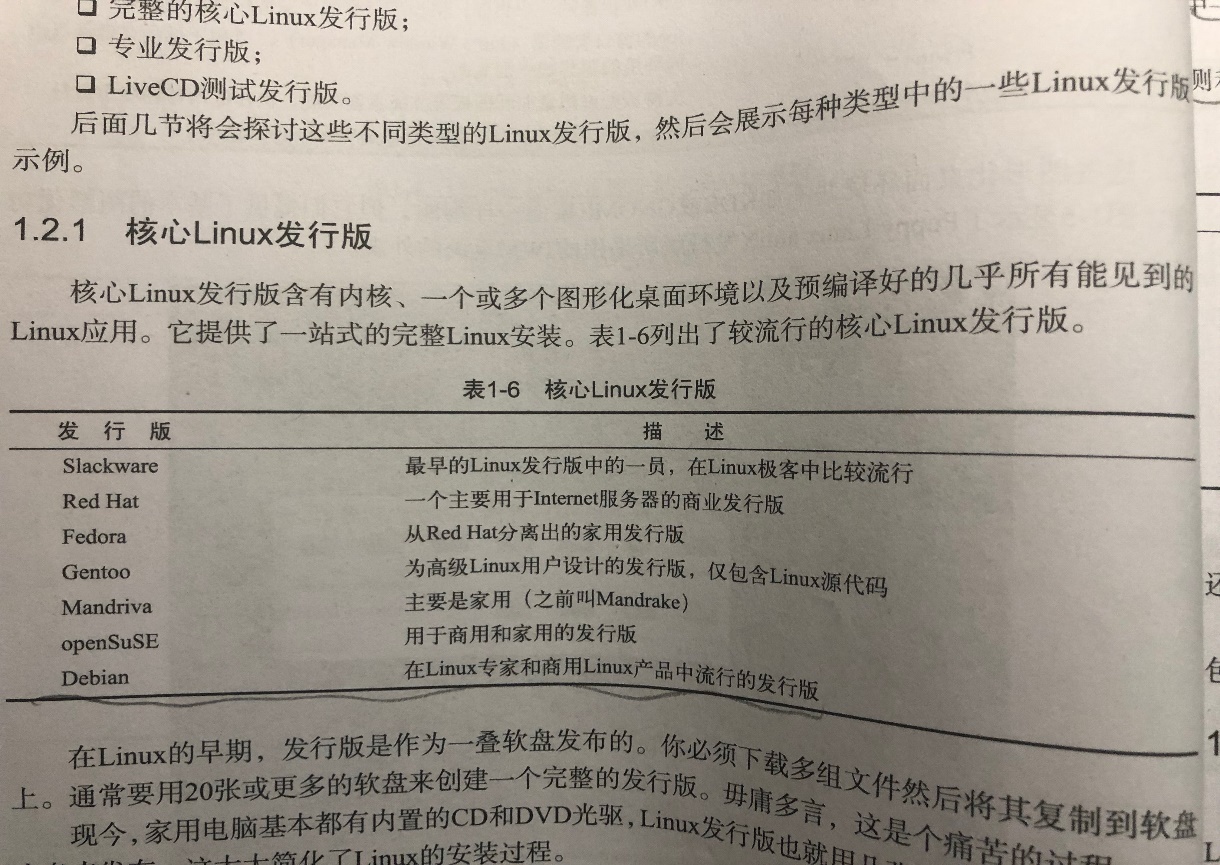
Rich:x:501:501:Rich Blum:/home/rich:/bin/bash

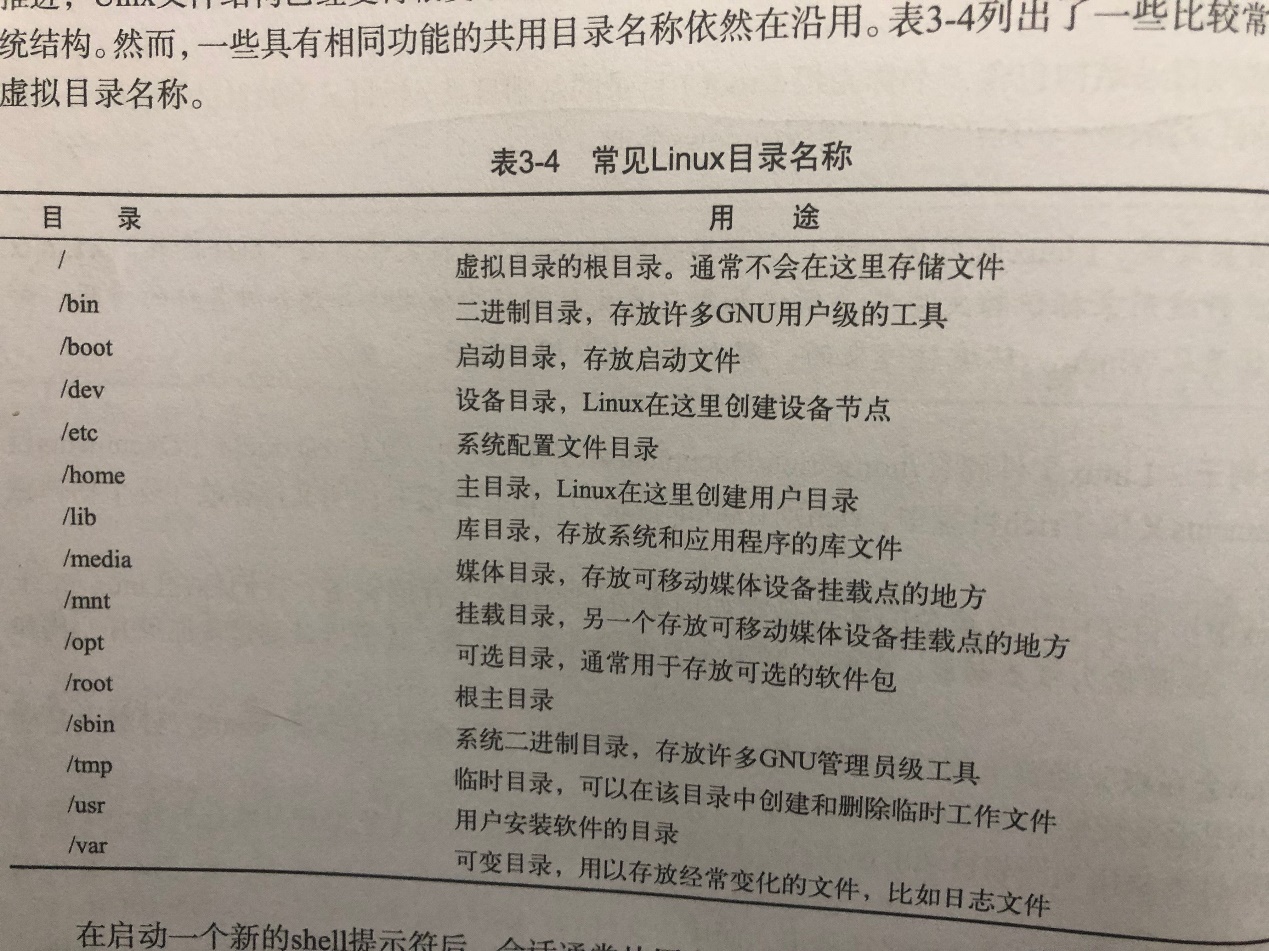
总共七个字段，分别为用户名，用户密码，用户的系统UID，用户的组ID，用户的全名，默认主目录，默认shell程序。bash启动时会自动处理用户主目录下.bashrc文件中的命令，许多Linux发行版在此文件中加载特殊的共用文件，在共用文件中保存着针对所有系统用户的命令和设置，通常该文件位于/etc/bashrc，它经常设置各种应用程序中用到的环境变量。

Linux划分为如下四个部分：Linux内核，GNU工具组件，图形化桌面环境，应用软件。其中内核负责以下四项主要功能：系统内存管理，软件程序管理，硬件设备管理，文件系统管理。 内核通过硬盘上的存储空间（swap space）来实现虚拟内存。内核创建了第一个进程(init进程)来启动系统上所有其他进程。当内核启动时，它会将init进程加载到虚拟内存中。

ps aux

STAT中的S表示睡眠，W为等待，R为运行中，进程名的方括号表示由于不活动而被从内存中换出到磁盘交换空间。





ln -l 硬链接， -s 符号链接（软链接）

不能在不同挂载点下的文件间创建硬链接，但可以使用软链接。

硬链接会创建一个独立文件，其中包含了源文件的信息以及位置，引用硬链接等同于用了源文件，其采用和源文件相同的索引节点号，会一直维持这个索引节点号来保留数据，直到删除了最后一个硬链接它的文件，而如果删除了源文件则所有的软链接文件指向的也就是一个无效文件了。

mount -t iso9660 -o loop name.iso mnt

挂载CD镜像iso文件，可cd mnt访问其文件系统。

当前在username主目录时： unmount /home/username/mnt

若无法卸载可通过 lsof /path/to/mount/point获得使用该设备的进程信息，再kill掉。（killall支持通过进程名而不是进程号来结束进程，也支持通配符。）最后可用ps或top命令看进程是否已停止运行（top是实时显示进程信息）

df -hl : 查看已挂载磁盘空间

du -csh : 显示特定目录下每个单独的文件和目录的大小。

sort -t ‘:’ -k 3 -n /etc/passwd : -t指定字段分隔符，-k指定排序的字段，-n将数字识别为数字而非字符。-r反序排序

du -sh \* | sort -nr : 查看目录下哪些文件占用空间最多。

printenv : 查看全局环境变量 set：查看全局和局部变量（只能在定义他们的进程中可见）

export : 设置全局变量

unset：删除环境变量

登录Linux系统时，shell依次处理以下文件：/etc/profile , $HOME/.bash\_profile , $HOME/.bash\_login , $HOME/.profile ，其中第一个是主启动文件，另外三个是用户专有的。

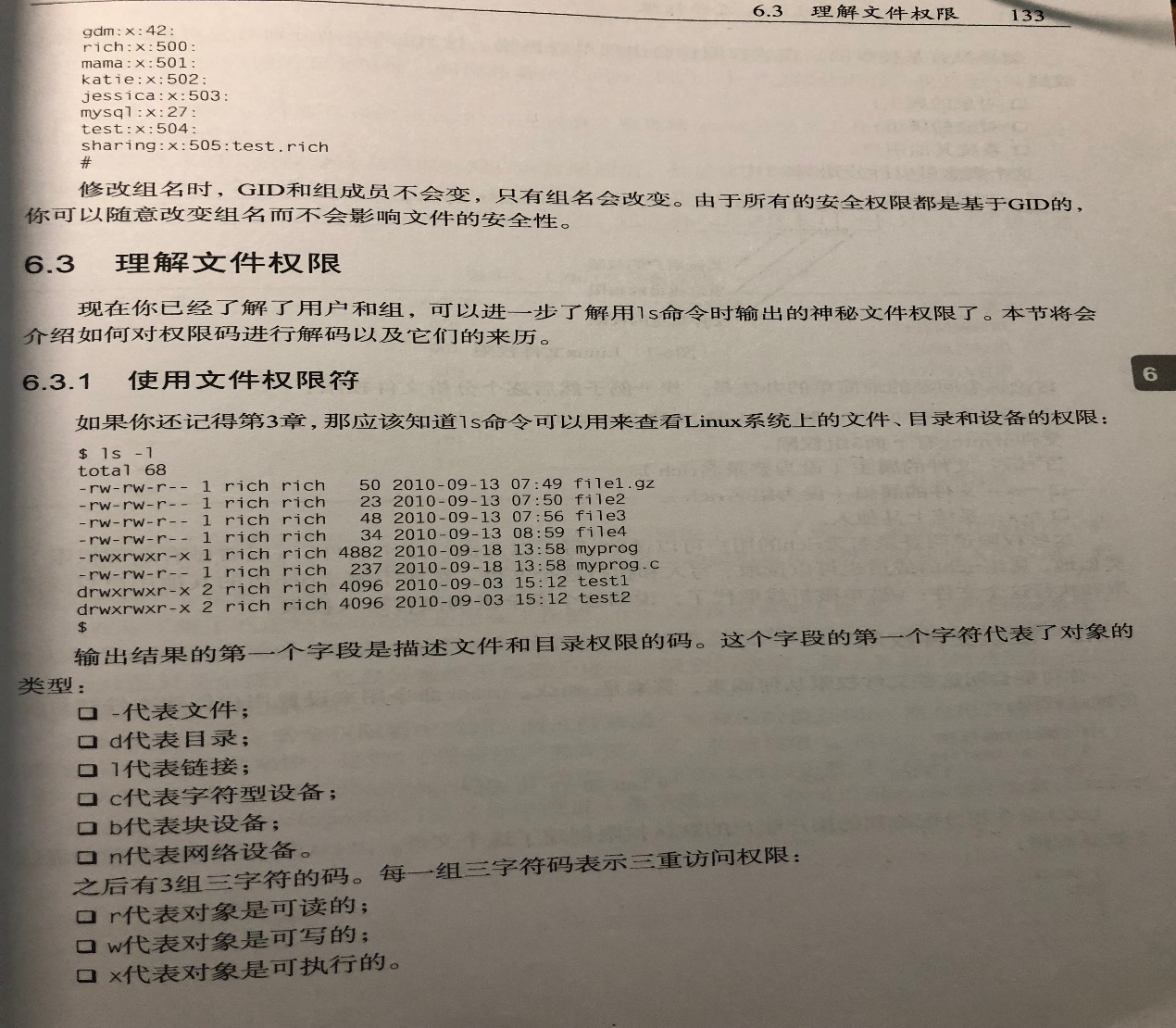
groupadd shared创建新组shared，usermod -G shared test：将test用户添加到shared组。

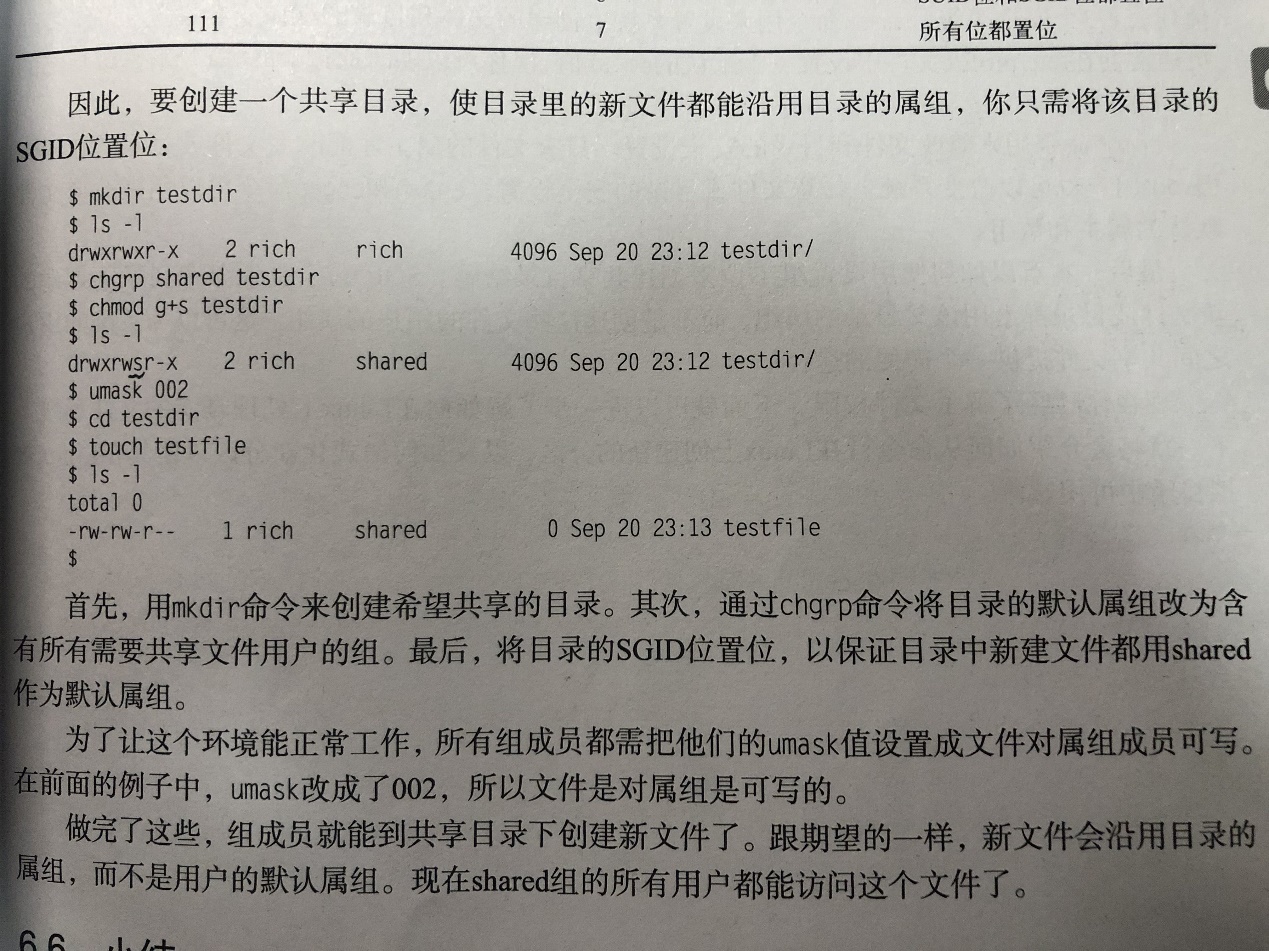
groupmod可以修改已有组的GID（-g）或组名(-n)：如，groupmod -n sharing shared

chmod [ugoa…][+-=][rwx…] :改变权限

chown owner[.group] file : 如，chown candy.shared -R mydir 递归地改变mydir目录下文件的属主和属组。

chgrp shared newfile : 更改文件或目录的默认属组。创建组是Linux系统上共享文件访问权限的方法。





aptitude show pkg\_name ： 查看软件库中的相关包信息

aptitude search pkg\_name : 查找包，如果第一列是i表示已安装，是p则表示未安装，是c则表示软件已删除但配置文件尚未从系统中清除。

aptitude install pkg\_name

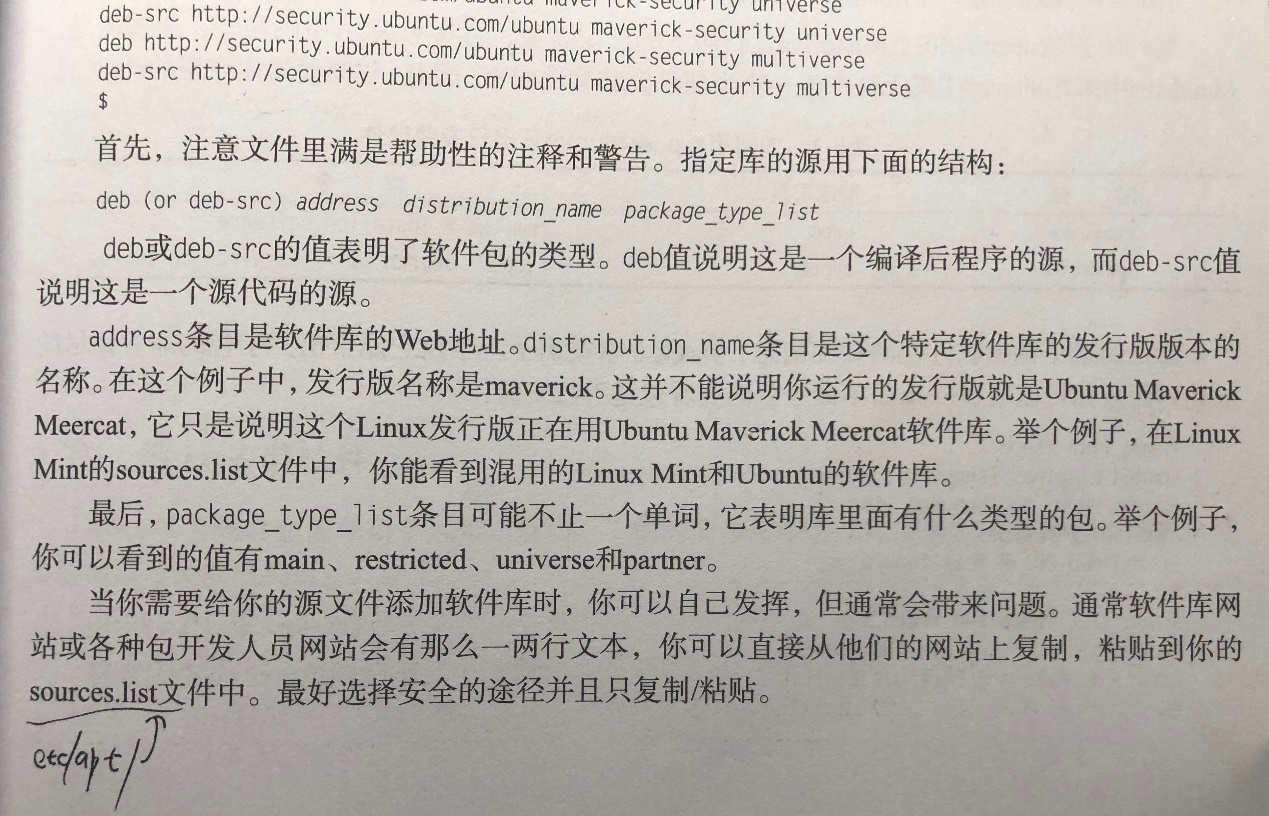
aptitude safe-upgrade

dpkg -L pkg\_name : 列出包中文件

dpkg –search absolute\_file\_name : 查找绝对路径下某个特定文件属于哪个软件包

aptitude remove pkg\_name : 只删除软件包但不删除数据和配置文件

aptitude purge pkg\_name : 同时删除软件包和相关数据



yum list installed

yum list pkg\_name

yum provides file\_name : 找出什么软件包提供了系统上的某个特定文件

yum install pkg\_name

yum localinstall pkg\_name.rpm

yum update [pkg\_name]

yum remove pkg\_name

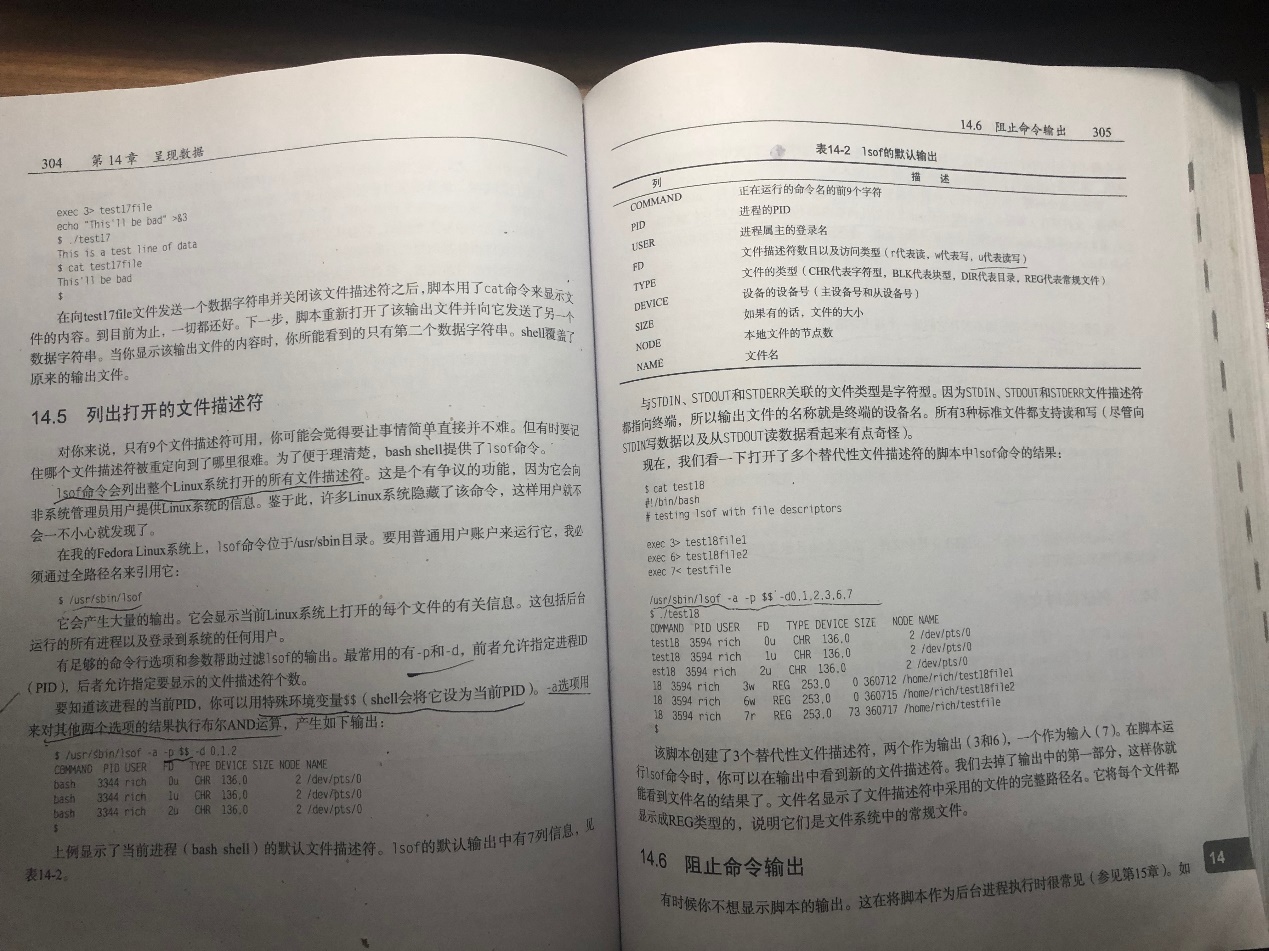
yum erase pkg\_name

处理损坏的包依赖关系

先yum clean all 再yum update

如果这还解决不了问题，则执行 yum deplist pkg\_name ,这个命令显示了所有包的库依赖关系以及什么软件可以提供这些库依赖关系，一旦知道了某个包需要的库，就能安装了他们了。

最后还可以试试： yum update –skip-broken ,这可能没法帮助损坏的包但可以更新其他包。



sed -n ‘{1!G ; h ; $p }’ file # 反转文本中的所有行

sed ‘=’ file | sed ‘N; s/\n/ /’ #输出行号