大数据开发之linux必会

谭唐华

课程大纲

- Linux环境搭建
- Linux常用命令
- Linux系统管理
- Shell命令编程

Linux介绍、虚拟机介绍

- 常见linux系统
- 常见虚拟机

Vmware的使用

- vmware中安装Centos6.5
- 虚拟机导入，配置虚拟机网络ip

win下连接linux工具介绍

- 远程命令行：secureCRT，putty
- 远程FTP：File Transfer Protocol, FileZilla， WinSCP
- 远程编辑工具：notpad++，UltraEdit
- 远程可视化工具：Xmanager

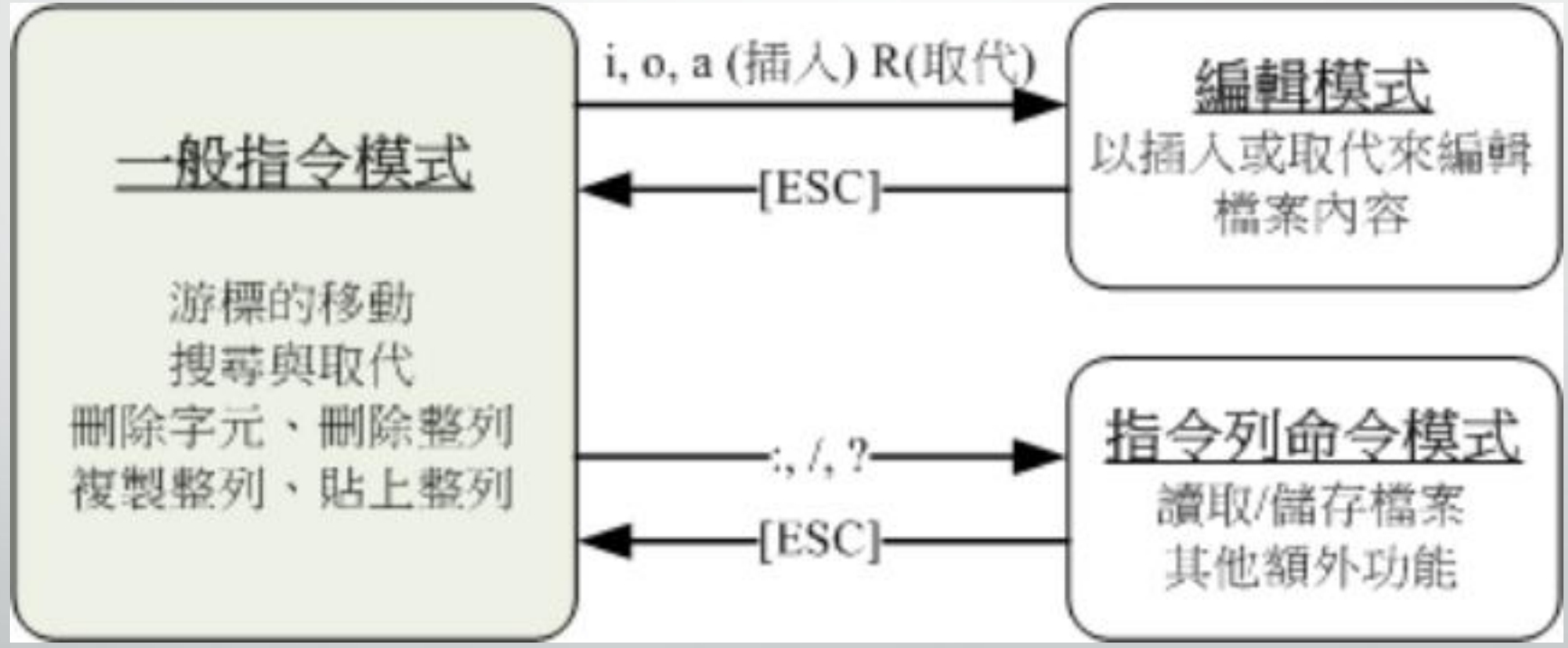
课程大纲

- Linux环境搭建
- Linux常用命令
- Linux系统管理
- Shell命令编程

常用命令

- 关机
- 重启
- 用户添加，删除
- 超级用户启用
- 用户切换

vim使用


一般指令模式
游標的移動
搜尋與取代
刪除字元、刪除整列
複製整列、貼上整列
i, o, a (插入) R(取代)
[ESC]
:, /, ?
[ESC]
編輯模式
以插入或取代來編輯
檔案內容
指令列命令模式
讀取/儲存檔案
其他額外功能

主机名与ip映射

- 主机名设置
- 主机名与ip的映射配置

课程大纲

- Linux环境搭建
- Linux常用命令
- Linux系统管理
- Shell命令编程

Linux文件权限概念

- Linux文件属性
- 如何改变文件属性与权限： chgrp, chown, chmod
- 目录与文件之权限意义
- Linux文件种类与扩展名

使用者与群组

- 文件拥有者
- 群组概念
- 其他人的概念

Linux文件属性


[root@www ~]# ls -al
total 156
drwxr-x--- 4 root root 4096 Sep 8 14:06 .
drwxr-xr-x 23 root root 4096 Sep 8 14:21 ..
-rw------- 1 root root 1474 Sep 4 18:27 anaconda-ks.cfg
-rw------- 1 root root   199 Sep 8 17:14 .bash_history
-rw-r--r-- 1 root root   24 Jan 6 2007 .bash_logout
-rw-r--r-- 1 root root   191 Jan 6 2007 .bash_profile
-rw-r--r-- 1 root root   176 Jan 6 2007 .bashrc
-rw-r--r-- 1 root root   100 Jan 6 2007 .cshrc
drwx------ 3 root root 4096 Sep 5 10:37 .gconf
<=范例说明处
drwx------ 2 root root 4096 Sep 5 14:09 .gconfd
-rw-r--r-- 1 root root 42304 Sep 4 18:26 install.log
<=范例说明处
-rw-r--r-- 1 root root 5661 Sep 4 18:25 install.log.syslog
[1][2][3][4][5][6][7]
[权限][连结][拥有者][群组][文件容量][修改日期][檔名]

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~
```

改变文件拥有者, 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
```

改变权限, chmod

```
[root@www ~]# chmod [-R] xyz 文件或目录
选项与参数:
xyz : 就是刚刚提到的数字类型的权限属性，为 rwx 属性数值的相加。
-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会变更
```

举例来说，如果要将.bashrc这个文件所有的权限都设定启用，那么就下达:

```
[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
```

目录与文件之权限意义

- 权限对文件的重要性
- 文件是实际含有数据的地方，包括一般文本文件、数据库内容文件、二进制可执行文件(binary program)等等。 因此，权限对于文件来说，他的意义是这样的：

- 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。

目录与文件之权限意义

- 权限对目录的重要性
- r: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:这个可写入的权限对目录来说，是很了不起的！因为他表示你具有异动该目录结构列表的权限，也就是底下这些权限:
- 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:目录的x代表的是用户能否进入该目录成为工作目录的用途！所谓的工作目录(work directory)就是你目前所在的目录啦！举例来说，当你登入Linux时，你所在的家目录就是你当下的工作目录。而变换目录的指令是『cd』(change directory)啰!

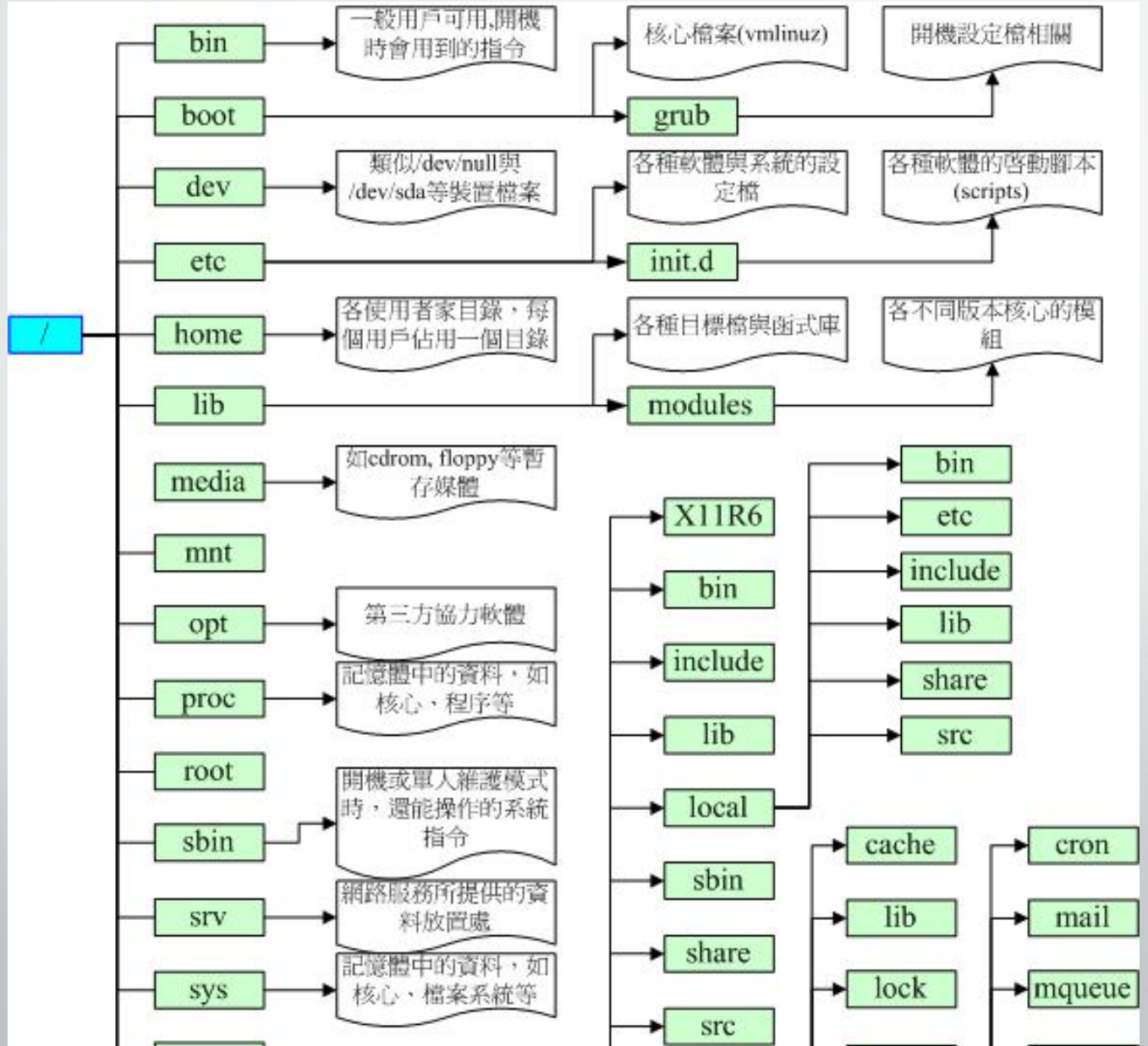
Linux文件种类与扩展名

- 文件种类：正规文件(regular file)，目录(directory)，连结档(link)，设备与装置文件(device)，数据接口文件(sockets)，数据输送文件(FIFO, pipe)
- Linux文件扩展名：*.sh，*.html, *.php，*Z, *.tar, *.tar.gz, *.zip, *.tgz

文件的限制

- Linux文件名长度限制：255
- Linux文件名的限制：*?><;&![]|\'"`(){}

Linux目录


/
bin
一般用戶可用,開機時會用到的指令
boot
核心檔案(vmlinuz)
grub
開機設定檔相關
dev
類似/dev/null與/dev/sda等裝置檔案
etc
各種軟體與系統的設定檔
init.d
各種軟體的啓動腳本(scripts)
home
各使用者家目錄，每個用戶佔用一個目錄
lib
各種目標檔與函式庫
modules
各不同版本核心的模組
media
如cdrom, floppy等暫存媒體
mnt
opt
第三方協力軟體
proc
記憶體中的資料，如核心、程序等
root
sbin
開機或單人維護模式時，還能操作的系統指令
srv
網路服務所提供的資料放置處
sys
記憶體中的資料，如核心、檔案系統等
X11R6
bin
include
lib
local
sbin
share
src
bin
etc
include
lib
share
src
cache
lib
lock
cron
mail
mqueue

目录的相关操作

- cd：变换目录
- pwd：显示目前的目录
- mkdir：创建一个新的目录
- rmdir：删除一个空的目录

cd(变换目录)

```
[root@www ~]# cd [相对路径或绝对路径]
# 最重要的就是目录的绝对路径与相对路径，还有一些特殊目录的符号罗！
[root@www ~]# cd ~vbird
# 代表去到 vbird 这个使用者的家目录，亦即 /home/vbird
[root@www vbird]# cd ~
# 表示回到自己的家目录，亦即是 /root 这个目录
[root@www ~]# cd
# 没有加上任何路径，也还是代表回到自己家目录的意思喔！
[root@www ~]# cd ..
# 表示去到目前的上一级目录，亦即是 /root 的上一级目录的意思；
[root@www /]# cd -
# 表示回到刚刚的那个目录，也就是 /root 罗~
[root@www ~]# cd /var/spool/mail
# 这个就是绝对路径的写法！直接指定要去的完整路径名称！
[root@www mail]# cd ../mqueue
# 这个是相对路径的写法，我们由/var/spool/mail 去到/var/spool/mqueue 就这样写！
```

pwd(显示目前所在的目录)

```
[root@www ~]# pwd [-P]
选项与参数:
-P   : 显示出确实的路径，而非使用连结 (link) 路径。

范例：单纯显示出目前的工作目录:
[root@www ~]# pwd
/root    <== 显示出目录啦~

范例：显示出实际的工作目录，而非连结档本身的目录名而已
[root@www ~]# cd /var/mail   <==注意，/var/mail是一个连结档
[root@www mail]# pwd
/var/mail          <==列出目前的工作目录
[root@www mail]# pwd -P
/var/spool/mail    <==怎么回事？有没有加 -P 差很多~
[root@www mail]# ls -ld /var/mail
lrwxrwxrwx 1 root root 10 Sep  4 17:54 /var/mail -> spool/mail
# 看到这里应该知道为啥了吧？因为 /var/mail 是连结档，连结到 /var/spool/mail
# 所以，加上 pwd -P 的选项后，会不以连结档的数据显示，而是显示正确的完整路径啊！
```

mkdir (创建新目录)

```
[root@www ~]# mkdir [-mp] 目录名称
选项与参数:
-m ：配置文件的权限喔！直接配置，不需要看默认权限 (umask) 的脸色~
-p ：帮助你直接将所需要的目录(包含上一级目录)递回创建起来！

范例：请到/tmp底下尝试创建数个新目录看看:
[root@www ~]# cd /tmp
[root@www tmp]# mkdir test    <==创建一名为 test 的新目录
[root@www tmp]# mkdir test1/test2/test3/test4
mkdir: cannot create directory `test1/test2/test3/test4':
No such file or directory       <== 没办法直接创建此目录啊！
[root@www tmp]# mkdir -p test1/test2/test3/test4
# 加了这个 -p 的选项，可以自行帮你创建多层目录！

范例：创建权限为rwx--x--x的目录
[root@www tmp]# mkdir -m 711 test2
[root@www tmp]# ls -l
drwxr-xr-x  3 root  root 4096 Jul 18 12:50 test
drwxr-xr-x  3 root  root 4096 Jul 18 12:53 test1
drwx--x--x  2 root  root 4096 Jul 18 12:54 test2
# 仔细看上面的权限部分，如果没有加上 -m 来强制配置属性，系统会使用默认属性。
# 那么你的默认属性为何？这要透过底下介绍的 umask 才能了解喔！  ^_^
```

rmdir(删除『空』的目录)

```
[root@www ~]# rmdir [-p] 目录名称
选项与参数:
-p ：连同上一级『空的』目录也一起删除

范例：将於mkdir范例中创建的目录(/tmp底下)删除掉!
[root@www tmp]# ls -l   <==看看有多少目录存在?
drwxr-xr-x  3 root  root 4096 Jul 18 12:50 test
drwxr-xr-x  3 root  root 4096 Jul 18 12:53 test1
drwx--x--x  2 root  root 4096 Jul 18 12:54 test2
[root@www tmp]# rmdir test    <==可直接删除掉，没问题
[root@www tmp]# rmdir test1   <==因为尚有内容，所以无法删除!
rmdir: `test1': Directory not empty
[root@www tmp]# rmdir -p test1/test2/test3/test4
[root@www tmp]# ls -l         <==您看看，底下的输出中test与test1不见了!
drwx--x--x  2 root  root 4096 Jul 18 12:54 test2
# 瞧! 利用 -p 这个选项，立刻就可以将 test1/test2/test3/test4 一次删除~
# 不过要注意的是，这个 rmdir 仅能『删除空的目录』喔!
```

关於运行档路径的变量： $PATH

```
范例：先用root的身份列出搜寻的路径为何？
[root@www ~]# echo $PATH
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin
:/bin:/usr/sbin:/usr/bin:/root/bin  <==这是同一行！

范例：用vbird的身份列出搜寻的路径为何？
[root@www ~]# su - vbird
[vbird@www ~]# echo $PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/vbird/bin
# 仔细看，一般用户vbird的PATH中，并不包含任何『sbin』的目录存在喔！
```

Linux 文件与目录管理

- 文件与目录的检视： ls
- 复制、删除与移动： cp, rm, mv
- 取得路径的文件名称与目录名称

文件与目录的检视： ls

```
[root@www ~]# ls [-aAdfFhilnrRSt] 目录名称
[root@www ~]# ls [--color={never,auto,always}] 目录名称
[root@www ~]# ls [--full-time] 目录名称
选项与参数：
-a   ：全部的文件，连同隐藏档( 开头为 . 的文件) 一起列出来(常用)
-A   ：全部的文件，连同隐藏档，但不包括 . 与 .. 这两个目录
-d   ：仅列出目录本身，而不是列出目录内的文件数据(常用)
-f   ：直接列出结果，而不进行排序 (ls 默认会以档名排序！)
-F   ：根据文件、目录等资讯，给予附加数据结构，例如：
       *:代表可运行档； /:代表目录； =:代表 socket 文件； |:代表 FIFO 文件；
-h   ：将文件容量以人类较易读的方式(例如 GB，KB 等等)列出来；
-i   ：列出 inode 号码，inode 的意义下一章将会介绍；
-l   ：长数据串列出，包含文件的属性与权限等等数据；(常用)
-n   ：列出 UID 与 GID 而非使用者与群组的名称 (UID与GID会在帐号管理提到！)
-r   ：将排序结果反向输出，例如：原本档名由小到大，反向则为由大到小；
-R   ：连同子目录内容一起列出来，等於该目录下的所有文件都会显示出来；
-S   ：以文件容量大小排序，而不是用档名排序；
-t   ：依时间排序，而不是用档名。
--color=never   ：不要依据文件特性给予颜色显示；
--color=always  ：显示颜色
--color=auto    ：让系统自行依据配置来判断是否给予颜色
--full-time     ：以完整时间模式 (包含年、月、日、时、分) 输出
--time={atime,ctime} ：输出 access 时间或改变权限属性时间 (ctime)
                       而非内容变更时间 (modification time)
```

cp(复制文件或目录)

```
[root@www ~]# cp [-adfilprsu] 来源档(source) 目标档(destination)
[root@www ~]# cp [options] source1 source2 source3 .... directory
选项与参数:
-a   : 相当於 -pdr 的意思，至於 pdr 请参考下列说明;(常用)
-d   : 若来源档为连结档的属性(link file)，则复制连结档属性而非文件本身;
-f   : 为强制(force)的意思，若目标文件已经存在且无法开启，则移除后再尝试一次;
-i   : 若目标档(destination)已经存在时，在覆盖时会先询问动作的进行(常用)
-l   : 进行硬式连结(hard link)的连结档创建，而非复制文件本身;
-p   : 连同文件的属性一起复制过去，而非使用默认属性(备份常用);
-r   : 递回持续复制，用於目录的复制行为;(常用)
-s   : 复制成为符号连结档 (symbolic link)，亦即『捷径』文件;
-u   : 若 destination 比 source 旧才升级 destination !
最后需要注意的，如果来源档有两个以上，则最后一个目的档一定要是『目录』才行!
```

rm (移除文件或目录)

```
[root@www ~]# rm [-fir] 文件或目录
选项与参数：
-f   ：就是 force 的意思，忽略不存在的文件，不会出现警告信息；
-i   ：互动模式，在删除前会询问使用者是否动作
-r   ：递回删除啊！最常用在目录的删除了！这是非常危险的选项！！！
```

mv(移动文件与目录，或更名)

```
[root@www ~]# mv [-fiu] source destination
[root@www ~]# mv [options] source1 source2 source3 .... directory
选项与参数：
-f  ：force 强制的意思，如果目标文件已经存在，不会询问而直接覆盖；
-i  ：若目标文件 (destination) 已经存在时，就会询问是否覆盖！
-u  ：若目标文件已经存在，且 source 比较新，才会升级 (update)
```

Linux 文件与目录管理

- 直接检视文件内容： cat, tac, nl
- 可翻页检视： more, less
- 数据撷取： head, tail
- 非纯文字档： od
- 修改文件时间与建置新档： touch

文件内容查阅

- cat 由第一行开始显示文件内容
- tac 从最后一行开始显示，可以看出 tac 是 cat 的倒著写！
- nl 显示的时候，顺道输出行号！
- more 一页一页的显示文件内容
- less 与 more 类似，但是比 more 更好的是，他可以往前翻页！
- head 只看头几行
- tail 只看尾巴几行
- od 以二进位的方式读取文件内容！

修改文件时间或建置新档

```
[root@www ~]# touch [-acdmt] 文件
选项与参数:
-a   : 仅修订 access time;
-c   : 仅修改文件的时间，若该文件不存在则不创建新文件;
-d   : 后面可以接欲修订的日期而不用目前的日期，也可以使用 --date="日期或时间"
-m   : 仅修改 mtime ;
-t   : 后面可以接欲修订的时间而不用目前的时间，格式为[YYMMDDhhmm]
```

观察文件类型：file

```
[root@www ~]# file ~/.bashrc
/root/.bashrc: ASCII text  <==告诉我们是 ASCII 的纯文字档啊!
[root@www ~]# file /usr/bin/passwd
/usr/bin/passwd: setuid ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), for GNU/Linux 2.6.9, dynamically linked (uses shared libs), for
GNU/Linux 2.6.9, stripped
# 运行档的数据可就多的不得了!包括这个文件的 suid 权限、兼容於 Intel 386
# 等级的硬件平台、使用的是 Linux 核心 2.6.9 的动态函式库连结等等。
[root@www ~]# file /var/lib/mlocate/mlocate.db
/var/lib/mlocate/mlocate.db: data  <== 这是 data 文件!
```

命令与文件的搜寻

- 命令档名的搜寻：which
- 文件档名的搜寻：whereis, locate, find

which (寻找『运行档』)

```
[root@www ~]# which [-a] command
选项或参数:
-a ：将所有由 PATH 目录中可以找到的命令均列出，而不止第一个被找到的命令名称
```

whereis (寻找特定文件)

```
[root@www ~]# whereis [-bmsu] 文件或目录名
选项与参数:
-b     :只找 binary 格式的文件
-m     :只找在说明档 manual 路径下的文件
-s     :只找 source 来源文件
-u     :搜寻不在上述三个项目当中的其他特殊文件
```

locate

```
[root@www ~]# locate [-ir] keyword
选项与参数:
-i   : 忽略大小写的差异;
-r   : 后面可接正规表示法的显示方式
```

find

```
[root@www ~]# find [PATH] [option] [action]
选项与参数:
1. 与时间有关的选项: 共有 -atime, -ctime 与 -mtime ，以 -mtime 说明
   -mtime  n : n 为数字，意义为在 n 天之前的『一天之内』被更动过内容的文件;
   -mtime +n : 列出在 n 天之前(不含 n 天本身)被更动过内容的文件档名;
   -mtime -n : 列出在 n 天之内(含 n 天本身)被更动过内容的文件档名。
   -newer file : file 为一个存在的文件，列出比 file 还要新的文件档名
```

文件的压缩与打包

- *.zip
- *.tar, *.tar.gz, *.tgz, *.gz, *.Z, *.bz2

软件安装：RPM, YUM 功能

平台名称	适合平台说明
i386	几乎适用於所有的 x86 平台，不论是旧的 pentum 或者是新的 Intel Core 2 与 K8 系列的 CPU 等等，都可以正常的工作！那个 i 指的是 Intel 兼容的 CPU 的意思，至於 386 不用说，就是 CPU 的等级啦！
i586	就是针对 586 等级的计算机进行最佳化编译。那是哪些 CPU 呢？包括 pentum 第一代 MMX CPU， AMD 的 K5, K6 系列 CPU (socket 7 插脚) 等等的 CPU 都算是这个等级；
i686	在 pentun II 以后的 Intel 系列 CPU ，及 K7 以后等级的 CPU 都属於这个 686 等级！ 由於目前市面上几乎仅剩 P-II 以后等级的硬件平台，因此很多 distributions 都直接释出这种等级的 RPM 文件。
x86_64	针对 64 位的 CPU 进行最佳化编译配置，包括 Intel 的 Core 2 以上等级 CPU ，以及 AMD 的 Athlon64 以后等级的 CPU ，都属於这一类型的硬件平台。
noarch	就是没有任何硬件等级上的限制。一般来说，这种类型的 RPM 文件，里面应该没有 binary program 存在， 较常出现的就是属於 shell script 方面的软件。

RPM 安装 (install)

```
[root@www ~]# rpm -ivh package_name
选项与参数：
-i ：install 的意思
-v ：察看更细部的安装资讯画面
-h ：以安装资讯列显示安装进度
```

RPM 查询 (query)

```
[root@www ~]# rpm -qa                          <==已安装软件
[root@www ~]# rpm -q[icdR] 已安装的软件名称      <==已安装软件
[root@www ~]# rpm -qf 存在於系统上面的某个档名    <==已安装软件
[root@www ~]# rpm -qp[icdR] 未安装的某个文件名称  <==查阅RPM文件
选项与参数：
查询已安装软件的资讯：
-q  ：仅查询，后面接的软件名称是否有安装；
-qa ：列出所有的，已经安装在本机 Linux 系统上面的所有软件名称；
-qi ：列出该软件的详细资讯 (information)，包含开发商、版本与说明等；
-ql ：列出该软件所有的文件与目录所在完整档名 (list)；
-qc ：列出该软件的所有配置档 (找出在 /etc/ 底下的档名而已)
-qd ：列出该软件的所有说明档 (找出与 man 有关的文件而已)
-qR ：列出与该软件有关的相依软件所含的文件 (Required 的意思)
-qf ：由后面接的文件名称，找出该文件属於哪一个已安装的软件；
查询某个 RPM 文件内含有的资讯：
-qp[icdlR]：注意 -qp 后面接的所有参数以上面的说明一致。但用途仅在於找出
        某个 RPM 文件内的资讯，而非已安装的软件资讯！注意！
```

RPM 卸载

```
#1.找出与 pam 有关的软件名称，并尝试移除 pam 这个软件：
[root@www ~]# rpm -qa | grep pam
pam-devel-0.99.6.2-3.27.el5
pam_passwdqc-1.0.2-1.2.2
pam_pkcs11-0.5.3-23
pam_smb-1.1.7-7.2.1
pam-0.99.6.2-3.27.el5
pam_ccreds-3-5
pam_krb5-2.2.14-1
[root@www ~]# rpm -e pam
error: Failed dependencies:  <==这里提到的是相依性的问题
        libpam.so.0 is needed by (installed) coreutils-5.97-14.el5.i386
        libpam.so.0 is needed by (installed) libuser-0.54.7-2.el5.5.i386
....(以下省略)....

#2.若仅移除 pam-devel 这个之前范例安装上的软件呢？
[root@www ~]# rpm -e pam-devel  <==不会出现任何信息！
[root@www ~]# rpm -q pam-devel
package pam-devel is not installed
```

yum 进行查询

```
[root@www ~]# yum [option] [查询工作项目] [相关参数]
选项与参数：
[option]：主要的选项，包括有：
 -y ：当 yum 要等待使用者输入时，这个选项可以自动提供 yes 的回应；
 --installroot=/some/path ：将该软件安装在 /some/path 而不使用默认路径
[查询工作项目] [相关参数]：这方面的参数有：
 search ：搜寻某个软件名称或者是描述 (description) 的重要关键字；
 list   ：列出目前 yum 所管理的所有的软件名称与版本，有点类似 rpm -qa；
 info   ：同上，不过有点类似 rpm -qai 的运行结果；
 provides：从文件去搜寻软件！类似 rpm -qf 的功能！
```

安装/升级功能： yum [install|update] 软件

```
[root@www ~]# yum [option] [查询工作项目] [相关参数]
选项与参数：
 install ：后面接要安装的软件！
 update  ：后面接要升级的软件，若要整个系统都升级，就直接 update 即可
```

移除功能： yum [remove] 软件

```
[root@www ~]# yum remove pam-devel
Setting up Remove Process
Resolving Dependencies  <==同样的，先解决属性相依的问题
--> Running transaction check
---> Package pam-devel.i386 0:0.99.6.2-4.el5 set to be erased
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch       Version          Repository        Size
================================================================================
Removing:
 pam-devel              i386       0.99.6.2-4.el5   installed         495 k

Transaction Summary
================================================================================
Install      0 Package(s)
Update       0 Package(s)
Remove       1 Package(s)  <==还好，并没有属性相依的问题，单纯移除一个软件

Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing    : pam-devel                       ######################### [1/1]
```

sudo启用

- 添加普通用户到sudo组，并且免密码登录

课程大纲

- Linux环境搭建
- Linux常用命令
- Linux系统管理
- Shell命令编程

Shell预览

- shell变量
- shell运算符
- shell数组
- shell判断
- shell循环
- shell函数

Shell变量

- 注意，变量名和等号之间不能有空格，这可能和你熟悉的所有编程语言都不一样。同时，变量名的命名须遵循如下规则：
- 首个字符必须为字母（a-z，A-Z）。
- 中间不能有空格，可以使用下划线（_）。
- 不能使用标点符号。
- 不能使用bash里的关键字（可用help命令查看保留关键字）。

shell算术运算符

算术运算符列表

运算符	说明	举例
+	加法	`expr $a + $b` 结果为 30。
-	减法	`expr $a - $b` 结果为 10。
*	乘法	`expr $a \* $b` 结果为 200。
/	除法	`expr $b / $a` 结果为 2。
%	取余	`expr $b % $a` 结果为 0。
=	赋值	a=$b 将把变量 b 的值赋给 a。
==	相等。用于比较两个数字，相同则返回 true。	[$a == $b] 返回 false。
!=	不相等。用于比较两个数字，不相同则返回 true。	[$a != $b] 返回 true。

shell关系运算符

关系运算符列表

运算符	说明	举例
-eq	检测两个数是否相等，相等返回 true。	[$a -eq $b] 返回 true。
-ne	检测两个数是否相等，不相等返回 true。	[$a -ne $b] 返回 true。
-gt	检测左边的数是否大于右边的，如果是，则返回 true。	[$a -gt $b] 返回 false。
-lt	检测左边的数是否小于右边的，如果是，则返回 true。	[$a -lt $b] 返回 true。
-ge	检测左边的数是否大等于右边的，如果是，则返回 true。	[$a -ge $b] 返回 false。
-le	检测左边的数是否小于等于右边的，如果是，则返回 true。	[$a -le $b] 返回 true。

shell布尔运算符

布尔运算符列表

运算符	说明	举例
!	非运算，表达式为 true 则返回 false，否则返回 true。	[! false] 返回 true。
-o	或运算，有一个表达式为 true 则返回 true。	[$a -lt 20 -o $b -gt 100] 返回 true。
-a	与运算，两个表达式都为 true 才返回 true。	[$a -lt 20 -a $b -gt 100] 返回 false。

shell字符串运算符

字符串运算符列表

运算符	说明	举例
=	检测两个字符串是否相等，相等返回 true。	[$a = $b] 返回 false。
!=	检测两个字符串是否相等，不相等返回 true。	[$a != $b] 返回 true。
-z	检测字符串长度是否为0，为0返回 true。	[-z $a] 返回 false。
-n	检测字符串长度是否为0，不为0返回 true。	[-z $a] 返回 true。
str	检测字符串是否为空，不为空返回 true。	[$a] 返回 true。

Shell数组

- 在Shell中，用括号来表示数组，数组元素用“空格”符号分割开。定义数组的一般形式为：
- array_name=(value1 ... valuen)
- 例如：
- array_name=(value0 value1 value2 value3)

Shell if else语句

- if ... else 语句的语法：
- if [expression]
- then
- Statement(s) to be executed if expression is true
- fi

Shell while循环

- while循环用于不断执行一系列命令，也用于从输入文件中读取数据；命令通常为测试条件。其格式为：
- while command
- do
- Statement(s) to be executed if command is true
- done

Shell函数

- Shell 函数的定义格式如下：
- function_name () {
- list of commands
- [return value]
- }
- 如果你愿意，也可以在函数名前加上关键字 function：
- function function_name () {
- list of commands
- [return value]
- }