

第 2 章：Linux 主机规划

1、了解硬件与硬件代号，第 39 页

硬件设备	Linux 中的代号	
IDE 硬盘	/dev/hd【a-d】	特别要注意这个
SCSI 硬盘	/dev/sd【a-p】	
光驱	/dev/cdrom	
软驱	/dev/fd【0-1】	
打印机	/dev/lp【0-2】	
鼠标	/dev/mouse	

2、安装 Windows 和 Linux 双系统必须先装 Windows，再安装 Linux。第 40 页

3、装系统时的分区原则，第 42 页

(1) 最简单的分区模式：Linux 安装的过程中，至少有两个分区才行，一个是根目录“/”，另一个是虚拟内存“Swap”，如果你的硬盘很小（例如小于 1GB），那么最好使用这个分区方法。

(2) 稍微麻烦一点的方式：默认情况下，由于 Linux 操作系统都是放在/usr/中，所以，你可以将这一部分分大一点，另外，由于用户的信息都是放在/home 下，因此，这个也可以大一些，而/var 下是记录所有默认服务器的登录文件，且 mail 与 WWW 的默认蹇也在/var 下，因此这个空间也可以加大一些。所以，需要的目录有：

```
/
/usr
/home
/var
Swap
```

第 3 章：Linux 安装与多重启动技巧

1、关于硬盘的相关知识，第 47~50 页

第 4 章：开机、关机、在线求助与指令输入

- 1、Linux 的命令区分大小写
- 2、bc 指令，打开系统的计算器
- 3、cal，显示当前月历；cal 2012 显示整年的月历
- 4、在线求助，man 指令和 info 指令
- 5、正确的关机是先通知客户，再 shutdown

第 5 章 Linux 文件权限与目录配置

1、ls -al 显示所的文件（包括隐藏文件）

```
-rwxrwxrwx 1 root root 293 Oct 19 21: 24 test
```

文件属性 连接数 文件拥有者 文件所属群组 文件容量

第一个属性代表这个文件是个目录、文件或者链接文件：【d】代表目录，【-】代表文件，【l】代表链接文件，【b】代表设备文件中可供储存的接口设备，【c】代表设备文件中

的串行端口设备，例如键盘

如果是一个目录，就算某用户对该目录有 `r` 权限但没有 `x` 权限，也是无法进入该目录的。 注：一个文件能否执行与其后缀无关，而与其读写权限有关

2、更改文件权限

(1) 改变群组(change group)

语法：Chgrp 群组名称 文件或目录 （注意：群组名称必须在 `/etc/group` 中存在，否则会显示错误）

(2) 改变文件拥有者 (change owner)

语法：chown [-R] 账号名称 文件或目录

chown [-R] 账号名称: 群组名称 文件或目录

注：拥有者名称在 `/etc/passwd` 中存在才能改变

例：chown test: test /home/test/testint 将文件的所有者和群组一起改变

(3) 改变权限

语法：chmod [-R] xyz 文件或目录 （xyz 是文件权限总和）

3、Linux 能否支持某个文件格式与内核是否将基编译进去有关，所以你可以到你的 Linux 系统的 `/lib/modules/uname-r/kernel/fs` 下查看，如果有你想要的文件格式，那么这个内核就支持它

Ps：从网上传送文件到你的 Linux 系统中，文件的属性（读写权限）可能会发生变化

4、需要注意的目录：

(1) `/etc`：这个目录相当重要，启动与系统数据文件均在这个目录下，这个目录被破坏意味着你的系统也差不多死掉了

(2) `/bin`、`/sbin`、`/usr/bin`、`/usr/sbin`：系统默认的执行文件的放置目录

(3) `/usr/local`：系统预留的让你安装后来的升级套件的目录

(4) `/home`：系统默认的用于存放用户账号的根目录

(5) `/var`：这个目录很重要。登入、各类服务发生问题时的记录，以及常规性的服务记录等都记在此目录下。系统出现问题时，就要在这个目录文件记录的数据中查看问题所在。

(6) `/usr/share/man`、`/usr/local/man`：放置各类套件说明文档的目录

第 6 章 文件与目录管理

1、绝对路径：路径的写法一定是从根目录“/”写起的。如 `/usr/local`

相对路径：路径写法不是从根目录写起

2、查看文件内容

cat 由第一行开始显示文件内容 cat [-n] 连行号一起输出屏幕

tac 从最后一行开始显示，可以看出，tac 是 cat 的倒写 tac 文件名

more 一页一页地显示文件内容 语法：more 【文件名】

less 与 more 类似，但其优点是，可以往前翻页 语法：less 【文件名】

head 只看头几行 语法：head [-n number] 【文件名】（只显示某一行）

tail 只看末尾几行 语法：tail [-n number] 【文件名】

nl 显示时同时输出行号 语法：nl 【文件名】

od 以二进制方式读取文件内容 语法：od 【文件名】

注：要同时使用两个命令，只有将两个相“|”即可，如显示 11 到 20 行 `head -n 20 ~/.bashrc | tail -n 10` 即可

3、Block 是记录文件内容数据的区域，inode 则是记录该文件的属性及其放置在哪个 Block 之内的信息。所以，每个文件都会占用一个 inode。当 Linux 系统要查找某个文件时，它会先搜寻 inode table 找到这个文件的属性及数据放置地点，然后再查找数据存放的 Block 进而将数据取出。（Block 相当于一本书，inode 就相当于这本书的目录）

4、链接 ln

链接分为硬链接和符号链接。

(1) 硬链接就是再建立一个 inode 链接到文件放置的 Block 块

硬链接有两大限制：一是不能跨文件系统，二是不能链接目录

(2) 符号链接是一个独立文件，而这个文件会让数据读取操作指向它链接的那个文件。

语法：ln [-s] 【源文件】 【目标文件】

参数：-s 提供符号链接，不加则属于硬链接

5、文件与目录权限

(1)、umask 002 新建文件时默认取消其他用户的写权限，umask 用于指定需要取消的权限

(2) chattr 指令

语法：chattr 【+ -=】 【选项】 文件或目录名称

参数：

+ -=：分别是增加、减少、设定属性

这个指令比较重要，对文件的安全性很有帮助。该指令的作用体现在选项里，由于这些属性属于隐藏属性，所以要使用 lsattr 才能看到这些属性

(3)、lsattr

语法：lsattr [-aR]

参数：

-a：将隐藏文件的属性也显示出来

-R：连同子目录的数据一并显示出来

6、搜寻文件或目录

(1) which：查看可执行文件的位置

whereis：查看文件的位置

locate：配合数据库查看文件位置

find：实际搜寻硬盘查询文件名称

(2)、which

语法：which 【文件名称】

(3)、whereis

语法：whereis [-bmsu] 【目录名称】

参数：

-b：只找二进制文件

-m：只找在说明文件-manual 路径下的文件

-s：只找 source 源文件

-u：没有说明文档的文件

(4)、locate

语法：locate 【目录名称】

(5)、find

语法：find 【路径】 【参数】

参数：

时间：

- atime n：将 n*24 小时内被存取过的文件列出来
- ctime n：将 n*24 小时内被改变、新增的文件或目录列出来
- mtime n：将 n*24 小时内被修改过的文件列出来
- newer file：把比 file 还要新的文件列出来

使用名称：

- gid n：寻找群组 id 为 n 的文件
- group name：寻找群组名称为 name 的文件
- uid n：寻找拥有者 id 为 n 的文件
- user name：寻找用户名称为 name 的文件
- name file：寻找文件名为 file 的文件名称（可使用通配符）
- type type：寻找文件属性为 type 的文件

7、（1）SUID：SUID 的主要功能是在某个文件执行期间具有文件拥有者的权限，因此，权限中 s 可以替代 x 可执行属性的位置。

（2）、file

语法：file 【文件名】

功能：查看文件的类型，以及查看文件是否被加入 SUID 等信息

第 7 章 硬盘与硬件管理

1、基本名词：

主引导扇区：硬盘中最重要的地方，记录了所有硬盘的分割信息，以及启动时可进行的启动管理程序的写入等。它只能提供最多 4 个分区的记忆，这就是主分区与扩展分区数量之和最多只能是 4 个的原因（主引导扇区相当于一本书的目录）

块（Block）：文件在磁盘中会被存储在一个固定大小的块中，即 Block

Superblock：进行硬盘分区时，每个磁盘分区就是一个文件系统，而每个文件系统起始位置的 Block 就称为 superblock

2、查看硬盘或目录的容量

（1）、df

语法：df -【ikm】

参数：

- i：使用 i-nodes 显示结果
- k：使用 KBytes 显示结果
- m：使用 Mbytes 显示结果

功能：查看硬盘的总容量、已用容量与 inode 等

（3）du

语法：du 【-abckms】 【目录名称】

参数：

- 【目录名称】可以省略，如果省略，表示要统计当前所在目录的文件容量
- a：列出所有的文件与目录，默认值是列出目录的值
- b：列出的值以 bytes 输出
- c：：最后求总 total
- k：列出的值以 KB 输出

-m：列出的值以 MB 输出

-s：只列出最后求总的值

功能：查看文件已用容量

3、磁盘分割与格式化

(1)、硬盘分区工具 fdisk

语法：fdisk 【-l】 【设备名称】

注意：设备名称不要加数字

参数：

-l：直接列出该硬盘设备的分区表

(2)、mke2fs

语法：mke2fs 【-b block-size】 【-i inode-size】

Mke2fs 【-cLj】

参数：

-b：设定每个数据块占用的大小，目前支持的大小有 1024、2048、4096

三种

-i：设定 inode 值

-c：检查磁盘错误，会比较慢

-L：设定这个扇区的 label(表头名称)

-j：建立 ext3 这个具有日志管理功能的文件格式

功能：这是用来将磁盘格式化成 Linux 系统文件的指令

(3)、e2label

语法：e2label 【/dev/hd...】 【label_name】

功能：主要用来更改磁盘的表头数据，也就是类似资源管理器中的每个磁盘驱动器代号后面的名称

(4)、mknod

语法：mknod 【device】 【bcp】 【Major】 【Minor】

参数：

b：设定节点为外部设备文档

c：设定节点为接口设备、如鼠标等

p：建立 FIFO

4、检查硬盘坏轨与数据同步写入

(1)、fsck

语法：fsck 【-Aar】 【设备名称】

参数说明：

-A：依据/etc/fstab 的内容，将所有的设备都扫描一次

-a：自动修复检查到的问题扇区，所以你不用一直按 y

-r：一定要让用户决定是否需要修复，与-a 刚好相反

功能：检查硬盘有没有坏轨

(2) sync

功能：将内存中的数据同步写入硬盘

5、关于启动盘

(1)、mkbootdisk

语法：mkbootdisk --device /dev/fd0 'uname -r'

参数：

--device：后面接设备，通常接的就是软盘/dev/fd0

功能：制作启动软盘

(2)、fdformat

语法：fdformat 【设备名称】

语法：低级格式化软盘的指令

6、硬盘的装载

mount 指令和 unmount 指令

7、硬盘效能测试与启用

hdparm

语法：hdparm 【-cdmXTt】 【设备名称】

参数：省略

第 8 章 文件系统的装载

1、建立与装载虚拟内存文件

基本上，虚拟内存就是将硬盘规划出一个空间，让内存数据可以经由硬盘读取，那么有 swap 也就够了，所以我们使用两种方法建立 swap 的扩增：

- (1) 直接加一块硬盘，并将其中某个分区规划为 swap 文件系统，步骤为：
 - a、用 fdisk /dev/hd【a-d】先建立一个分区。具体说，就是：建立一个分区，并将该分区的 ID 改为 82，这是 swap 文件格式代号；
 - b、用 mkswap /dev/hd【a-d】【1-16】将你刚刚配置的分区格式化为 swap 文件格式
 - c、启动 swap。启动指令为 swapon /dev/hd【a-d】【1-16】
- (2)、如果不想新增一个分区，可以使用 swap 文件方式，基本流程是：
 - a、以 dd 指令建立 swapfile；
 - b、以 mkswap 将 swapfile 格式化为 swap 文件格式
 - c、以 swapon 启动该文件，使之成为 swap
 - d、以 swapoff 关闭该文件

第 155 页

第 9 章 vi 文本处理器

1、(1) 纯文本文件：可以直接查看的文件，而不会打开出现乱码

(2) Vi 的使用

命令行模式下输入 r 表示替换光标所指的那个字符，R 会一直替换光标所指的那个字符

第 10 章 学习 Bash

(1)、指令补全功能：如输入一个 l 再连续按两个 TAB 键，系统会把所有以 l 开头的指令列出来。

给指令命名别名：alias lm='ls -al'，在 shell 里输入 lm 就等于输入 ls -al

取消命名别名用 unalias

(2)、echo \$variable：显示当前的 PATH 变量

env：显示所有环境变量

set：显示当前的环境变量的自定义变量

定义 shell 变量时，单引号只作为一般字符，如 `echo '$name is me'`，只会输出“\$name is me”，而不会输出 name 变量所代表的值

取消变量定义：unset，如 `unset name`

(3)、历史指令服务 history

语法：history

下一行：【! number】【!command】【!!】

参数：

number：第几个指令

command：指令开头几个字母

!：上一个指令

(4)、个人设定值

个人的自定义设置在个人根目录的几个隐藏文件中设定，~代表个人根目录

~/bash_profile：存放个人路径（PATH）与环境变量的文件名称，可在此修改个人路径

~/bashrc：可在此修改个人变量，如别名的设定和路径的重新定义等

(5)、source

语法：source 变量配置文件

功能：在不注销登录的情况下让更改的配置生效

(6)、命令重定向

标准输入：代码为 0，划称为 stdin，使用方式为 <

标准输出：代码为 1，或称为 stdout，使用方式为 1> (或者把 1 省略)

错误输出：代码为 2，或称为 stderr，使用方式为 2>

例：ls -al 1>list.txt 2>&1 错误和正确信息输出同一个文件中

3、管线命令

注：管线命令只能用于“|”界定符之后，也就是只能作为其它指令的附属指令，如：
cat /etc/passwd | sort

(1)、管线命令使用“|”界定符，管线命令仅能处理由前一个指令传来的正确信息，也就是标准输出信息，对标准错误信息没有直接处理能力。每个管线部分都是指令，而后一个指令的输入是前一个指令的输出

(2)、cut

语法：cut -d “分隔符” [-cf] fields

参数：

-d：后面接的是分隔符，默认是空格符

-c：后面接的是第几个字符

-f：后面接的是第几个区块

功能：主要的用途在于将同一行里的数据进行分解，最常用在分析一些数据或文字数据的时候，这是因为有时候我们会以某些字符当作分割对数，然后将数据加以分割，以取得我们需要的数据

(3)、sort

语法：sort [-t 分隔符] [(+起始) (-结束)] [-nru]

参数：

-t 分隔符：使用分隔符隔开不同的区块，默认是 tab

+start -end：由第 start 区块排序到 end 区块

-n：使用纯数字排序（否则会按字母排序）

-r：反向排序

-u：相同出面的一行，只列出一次

(4) \ wc

语法：wc [-lmw]

参数：

-l：多少行

-m：多少字符

-w：多少字

功能：计算文件内容

(5) \ uniq

语法：uniq

功能：删除重复的行从而只显示一个

(6) \ tee

(7) \ tr

语法：tr [-ds] SET1

参数：

-d：删除 SET1 这个字符串

-s：取代重复的字符

(8) \ split

语法：split [-bl] 输入文件 输出文件前导字符

参数：

-b：以文件 size 来分

-l：以行数来分

功能：分割文件

第 11 章 压缩指令与正规压缩表示法

1、压缩指令

几个常见的压缩文件后缀名如下：

*.Z compress 程序压缩的文件

*.bz2 bzip2 程序压缩的文件

*.gz gzip 程序压缩的文件

*.tar tar 程序打包的数据，没有经过压缩

*.tar.gz tar 程序打包的文件，且经过 gzip 压缩

下面介绍几个压缩程序的使用方法

(1) \ compress (*.Z)

语法：compress [-d] filename

参数：

-d：解压缩参数

(2) \ bzip2, bzip2 (*.bz2)

语法：bzip2 [-dz] filename

bzip2 filename.bz2

压缩解压缩指令

读取压缩文件内容

参数：

-d：解压缩

-z：压缩

(3) 、gzip, zcat (*.gz)

语法: gzip 【-d#】 filename 压缩与解压缩
zcat filename.gz 读取压缩文件内容

参数:

- d : 解压缩的参数
- # : 压缩等级, 1 最不好, 9 最好, 6 默认值

(4) 、tar

语法: tar 【-zxcvfpP】 filename
tar -N 'yyyy/mm/dd' /path -zcvf target.tar.gz source

参数:

- z : 是否同时具有 gzip
- x : 解开一个压缩文件
- t : 查看 tarfile 里面的文件
- c : 建立一个压缩文件
- v : 压缩过程中的显示文件
- f : 使用文件名
- p : 使用原文件的原有属性 (属性不会依据用户而变)
- P : 可以使用绝对路径
- N : 比后面接的日期 (yyyy/mm/dd) 还要新的文件才会被包进新建

的文件中

(5) 、cpio

语法: cpio -covB > 【file | device】 备份
cpio -icduv < 【file | device】 还原

参数:

- o : 将数据复制输出到文件或设备上
 - i : 将数据自文件或设备复制出到系统中
 - t : 查看 cpio 建立的文件或设备的内容
 - c : 以一种较新的便携格式储存
 - v : 让储存过程中文件名称可以在屏幕上显示
 - B : 让默认的 Blocks 可以增至 5120bytes, 默认是 512bytes, 这样的
- 好处是可以让大文件的储存速度加快
- d : 自动建立目录
 - u : 自动用较新的文件覆盖较旧的文件

延申: last 命令可以查看当月哪些用户登陆过系统

2 、grep

语法: grep "word" filename

功能: 在 filename 里找到包含关键词 word 的一行

第 12 章 学习 Shell scripts

1 、脚本文件的执行方法有两种:

- i. 一个是将该文件改成可以执行的属性, 如 `chmod 755 scripts.file`, 然后执行该文件
- ii. 另一种则是直接以 `sh` 这个执行文件来执行脚本内容, 如 `sh scripts.file`

2、(1)、定义变量 declare

语法: declare 【-afirx】

参数:

- a : 定义为数组 array
- f : 定义为函数 function
- i : 定义为整数 integer
- r : 定义为只读
- x : 定义为通过环境输出变量

例: [test @test test]#declare -i c=\$a*\$b

变量 a 和 b 都要用 declare 定义, a=3,b=5 时输出 15, 而不用 declare 时输出 3*5

(2)、命令行: sh test pa1 pa2 pa3

其中 test 参数代号是 \$0, 第一个参数 pa1 为 \$1, 依此类推, \$0 也就是脚本的文件名

注意: 在 shell 里面, 每个独立的组件间要用空格键格开, 如 if["\$syn" = ""]

2、if...then...fi

```
语法: if 【条件判断 1】 && (||) 【条件判断 2】; then
      elif 【条件判断 3】 && (||) 【条件判断 4】; then
      else
      fi
```

3、case...esac

语法: case 种类方式 (string) in

种类方式一)

程序执行段

;;

种类方式二)

程序执行段

;;

*)

echo "Usage: {t 种类方式二 | 种类方式二}"

exit 1

esac

种类方式的格式有两种:

直接输入: 就是以“执行文件+string”的方式执行

交互式: 就是由屏幕输出可能的项, 然后由用户输入, 通常必须配合 read variable, 然后 string 写成 \$variable 的格式

4、循环

for 语句

while 语句

until 语句

5、调试脚本 sh

语法: sh 【-nvx】 scripts

参数:

- n: 不执行脚本, 查询脚本内的语法, 若有错误则列出

-v：在执行脚本之前，先将脚本的内容显示在屏幕上

-x：将用到的脚本内容显示在屏幕上，与-v 稍微不同

第 13 章

1、认识/etc/passwd 文件与/etc/shadow 文件

a) passwd 的构造

在这个文件中，每一行代表一个账号，有几行就代表在你的系统中有几个账号，但如 bin 等很多账号本来就是系统必要的。UID=0 的账号是系统管理员

b) shadow 的构造

用户的账号密码以加密的形式存放在 shadow 里，在这个文件里，如果密码栏的第一个字符为“*”，表示这个账号不会被登入。所以如果忘记 root 密码，可以以软盘启动，进入单人维护系统，清空 root 的密码

2、认识/etc/group 与/etc/gshadow 文件

这两个文件和群组有关。

a) /etc/group 文件可以让你直接将所要支持的群组加进来

3、（1）、groupadd 指令、groupdel 指令、useradd 指令、userdel 指令

（2）、chsh 指令

语法：chsh [-l] [-s shellname]

参数：

-l：列出当前机器上能用的 shell 名称

-s：将当前的 shell 变为 shellname

4、用户身份切换

（1）、su 指令

（2）、sudo 和 visudo

语法：sudo [-u username] [command]

参数：-u：将身份变成 username 的身份

功能：使用 sudo 指令可以在当前的用户身份的情况下执行 root 身份可以操作的事，但要用 visudo 编辑/etc/sudoers 文件，在文件里加上 test ALL=(ALL) ALL（单个用户 test 可以使用 sudo）或 wheel ALL=(ALL) ALL（wheel 群组可以使用）

5、用户查询

id：查询用户的 UID,GID 及所拥有的群组

groups：查询用户能够支持的群组

finger：查询用户的一些相关信息，如电话号码等

（1）、id

语法：直接输入 id

（2）、finger

语法：finger [-s] username

参数：完整列出

注：finger 不是很安全，所以系统默认不安装这个套件

（3）、groups

语法：直接输入 groups

6、手动添加用户步骤：（如新增全是数字的账号）

a、先建立需要的群组（vi /etc/group）

b、建立账号的各个属性（vi /etc/passwd）

- c、将 passwd 与 shadow 同步 (pwconv)
- d、建立该账号的密码 (passwd account)
- d、建立用户根目录 (cp -r /etc/account.group /home/account)
- e、更改根目录属性 (chown -R account.group /home/account)

第 14 章 磁盘配额

使用磁盘配额有如下几个限制

- a、实际动作时，配额是针对整个分区进行的限制
- b、Linux 系统内核必须支持 quota 模块

Quota 程序的限制内容主要分为以下几个部分：

- a、soft：这是最低限制容量（相当于警告容量）
- b、hard：这是绝对不能超过的容量
- c、宽限时间：宽限时间就是你的用户使用空间超过 soft limit，但还没到达 hard limit

1、基本的 quota 指令

Quota 指令基本上分两种，一种是查询功能（quota, quotacheck, quotastats, warnquota, repquota），另一种则是编辑 quota 的内容（edquota, setquota）

(1)、quota

语法：quota [-guvs] [user,group]

参数：

- g：显示群组
- u：显示用户
- v：显示 quota 的值
- s：选择显示 inod 或硬盘空间

功能：显示当前某个群组或某个用户的 quota 限值

(2)、quotacheck

语法：quotacheck [-auvg] /yourpath

参数：

- a：扫描/etc/mstab 中所有已经安装并支持 quota 的硬盘
- u：扫描用户的文件与目录
- v：显示扫描过程
- g：扫描群组使用的文件与目录

功能：扫描某个磁盘的 quota 空间

(3)、edquota

语法：edquota [-u user] [-g group] [-t]

edquota -p user_demo -u user

参数：

- u：编辑 user 的 quota
- g：编辑 group 的 quota
- t：编辑宽限时间（就是超过 quota 值后，还能使用硬盘的宽限期限）
- p：复制模板（已建立的用户或群组）到另一个用户（或群组）

功能：编辑每个用户或群组的 quota 数值

(4)、quotaon

语法: quotaon [-a] [-uvvg directory]

参数:

-a : 启动所有的 quota 设定 (会自动寻找/etc/mtab 的设定)

-u : 激活用户的 quota

-g : 启动群组的 quota 设定

-s : 显示信息

功能: 用于启动 quota

(5) 、quotaoff

语法: quotaoff -a

参数:

-a: 关闭所有的 quota 设定 (会自动寻找/etc/mtab 的设定)

功能: 关闭 quota 限制

2、使用 quota 的几个主要步骤:

a、开启磁盘的 quota 功能: 编辑/etc/fstab, 使得准备开放的 quota 磁盘可以支持 quota

(在 parameters 一栏增加 usrquota 和 grpquota 两项)

b、预扫描磁盘的使用情况: 使用 quotacheck 扫描我们要使用的磁盘

c、建立用户的 quota: 使用 edquota 来编辑每个用户或群组的可使用空间

d、启动 quota: 使用 quotaon

注: 开机自启动在/etc/rc.d/rc.local 下。要将 quota 设置成开机自启动, 要在该文件里加上/sbin/quotaon -aug 即可

第 15 章 例行性命令 at 与 crontab

1、at : 这个工作仅执行一次就从 Linux 系统的流程中取消

crontab : 这个工作将持续例行性地做下去

2、仅执行一次: at

语法: at [-m] TIME (输入的工作指令)

atq (查看当前的工作流程)

atrm [jobnumber] (删除流程)

参数:

-m : 执行 at 规范的工作流程时, 将屏幕输出结果 mail 给输出指令的用户

TIME : 时间格式, 有如下几个:

HH: MM YYYY-MM-DD ex>04: 00 2002-05-30

HH [pm;am] +number [hours;days;week] ex>4pm+3days (3 天后的下午 4 点)

HH: MM ex>12: 00 (今天 12 点执行)

HH [pm;am] [Month] [Day] ex>1pm May 30

Jobnumber : 这即是每个 at 工作流程都有的编排顺序

注: 按下 Ctrl+D 退出 at

3、循环执行: crontab

这个例行命令其实就是 cron 服务

语法: crontab [-u user] [-l|-e|-r]

参数:

-u user：这个指令只有 root 能输入，查看或编译其他用户的 crontab 内容

-l：列出 crontab 的内容

-e：编辑 crontab 的内容

crontab -e 进入 crontab 编辑内容，使用 vi

0 12 * * * mail test < /home/test/test.txt

前面的 5 类数字代表的意义如下

数字代表的意义	分钟	小时	日期	月份	周
范围	0~59	0~23	1~31	1~12	0~6(0 是星期天)

如果出现*号，代表所有数字都适用。所以，上述指令意思是无论何年、何月、何日、星期几的 12 点，都执行后面的指令。

注：

a、指令的路径最好是绝对路径

b、第一个数字*/5 表示第 5 分钟执行一次

c、删除用 crontab -r

4、系统的 crontab 设定：/etc/crontab

Crontab 服务的最低检测时间单位是分钟。只要编辑完/etc/crontab 文件并保存之后，crontab 的设定会自动执行。如不会马上执行可以执行命令：/etc/rc.d/init.d/crond restart

(1) \etc/crontab 支持两种格式输入指令方式，一种是直接以指令形式输入，一种则是以目录形式输入

指令形式：

01 * * * * test mail -s test test < /home/test/test.txt

用户是 test,且每小时执行一次指令 mial

目录形式：

* /5 * * * * root run-parts /root/runcron

建立一个/root/runcron 目录，每隔 5 分钟执行一次可执行文件都写在该目录下，就可以让系统每 5 分钟执行一次该目录下的所有可执行文件

第 16 章 程序与资源管理器

1、man passwd

man 5 passwd

加上 5 以后就变成自动查询/etc/passwd 这个文件的设定功能了。

Linux 使用 man 时，对数配置文件的查询都是以 5 来设定的。

2、多用户环境：Linux 允许不同的用户使用，而且每个有其特殊的权限，但只有一个人（系统管理员）具有至高无上的权力，除了他之外，其他人都必须受到一定的限制

多任务环境：Linux 允许不同的画面，同时由不同的人开启或修改同一个文件，只有你有权限

7 个基本窗口：Linux 默认提供了 6 个文字界面登入窗口，以及一个图形界面，你可以使用 Alt+F1~F7 来切换不同的终端界面。（ctrl+alt+F1~F7??）

3、背景工作原理(即后台执行)

(1)、&

语法: `command &`

例: `find / -name testint &`

将该执行程序丢到后台执行

`fg`

将该程序拉回屏幕前执行

(2)、`ctrl+z`

语法: 在 `vi` 的一般模式中键入 `ctrl+z`

功能: 在不保存退出 `vi` 的情况下, 暂时退出 `vi`, 把数据放到背景

(后台)

(3) `jobs`

语法: `jobs`

功能: 显示当前有多少任务工作在背景 (后台) 下

(4)、`fg`

语法: `fg %number`

`bg %number`

参数:

`%`: 后面接数字, 表示 `jobs` 的工作代号

`Number`: 工作代号

功能: `bg %number` 将数据放在背景中执行, `fg %number` 将代号为 `number` 的程序移支屏幕前的操作

(4)、`kill`

语法: `kill -signal %number`

参数:

`%number`: 背景工作的代号, 可使用 `jobs` 查询

`-1`: 重新读取参数配置文件 (类似 `reload`)

`-2`: 用户中断该工作, 类似 `ctrl+c` 中断一个工作

`-9`: 立刻杀掉一个工作, 不论该工作是否为死亡程序

`-15`: 停止一个工作 (这是默认值)

4、程序与资源管理

(1)、`ps`

语法: `ps -aux`

参数:

`-a`: 选择列出所有的程序

`-u`: 列出所有用户的程序

`-x`: 列出所有 `tty` 的程序

功能: 查询当前主机环境中在背景执行的相关程序

(2)、`top`

语法: `top`

参数:

在执行 `top` 的程序中, 可以输入下面的字符进行排序

`A`: 以 `age` 亦即执行的先后顺序进行排序

`T`: 由启动的时间排序

`M`: 以所点的内在大小排序

`P`: 以所耗的 `CPU` 资源排序

功能: 动态显示当前主机环境中在背景执行的相关程序

注: **PS** 是静态的, **top** 是动态的 (每 5 秒更新一次)

(3) 、free

语法：free

参数：

-k：以 Kbytes 显示内存

-m：以 Mbytes 显示内存

功能：检查当前内存的使用情况

(4) 、sar

语法：sar 【-u】 【-r】 秒数 次数

参数：

-u：看 CPU 的使用情况

-r：看内存的使用情况

秒数：几秒钟观测一次？

次数：共观测几次？

注：Mandrake 默认没有安装这个套件，如果需要，可到相关网站下载 sysstat 这个套件

(5) 、kill

语法：kill -signal PID

参数：

-signal 跟上面的 kill 一样

-l：让该 PID 重新读取它的配置文件

-9：杀掉该程序

-15：停止该程序

(6) 、uname

语法：uname 【-apnr】

参数：

-a：列出所有的系统信息

-p：列出 CPU 信息

-n：列出主机名

-r：列出内核版本信息

功能：列出内核版本、主机名称、CPU 等系统信息

5、程序的优先级

在使用 ps -l 指令后显示出来的信息有两个和优先级相关的信息，即 PRI 和 NI，PRI 是该程序被 CPU 执行的先后顺序，所以当 CPU 忙碌时，PRI 值越小就会越先执行。NI 代表 nice 值，该值是系统可被执行的修正数值。注意：最终 PRI 的值为 $PRI(new)=PRI+NI$

(1) 、nice

语法：nice 【-n number】 command

参数：

-n：后面那个 number 即为 nice 值

功能：调整程序的执行优先级

(2) 、renice

语法：renice 【number】 PID

功能：用于改变一个正在进行中的程序的优先级，所以必须先取得该程序的 PID

6、信息管理

(1)、dmesg

语法：dmesg

功能：系统在启动时会显示很多信息，该指令用于观察这些系统信息，通常加上“| more”使用

(2)、uptime

语法：uptime

功能：查看系统开机时间、平均负载等相关信息

(3)、who,w,whoami

语法：who

w

功能：用来查看当前系统的用户的指令，whoami 可以帮助解决由于使用 su 变换身份而导致忘记自己的真实身份的问题

(4)、last, hostname

语法：last

参数：

-number : number 为数字，如果你的登入信息太多，可以使用这个指令

功能：想知道有没有被入侵，常常使用的就是 last 这个指令，包括 ftp、telnet 等都会被记录到这个信息中，不过当前只记录一个月的量

第 17 章 开机关机流程与多重启动

1、开机流程分析

(1)、在 rc3.d(或其他目录下，rc0.d~rc6.d)目录中，以 S 开头的文件为执行该服务文件，以 K 开头的文件则是杀掉该服务的意思，那些数字代表启动的顺序，如 S12syslog 会比 S90crond 先执行（启动时的配置文件）

(2)、加载内核之后，由内核执行的第一个程序是/sbin/init

(3)、Linux 的登入模式有如下几种

a)、0——halt(DO NOT set initdefault to this) 关机

b)、单用户模式（系统有问题时的登入模式，类似 windows 的安全模式）

c)、多用户但无网络

d)、从文字界面登入的多用户系统

e)、系统保留

f)、x11——X-Windows 图形界面登入的多用户系统

g)、重新启动

(4)、想要系统启动时默认进入文字界面，只要用 vi 编辑器打开/etc/inittab 文件，然后将“id: 5: initdefault”更改为“id: 3: initdefault”即可

第 18 章 套件管理 RPM 与 Tarball

1、RPM：目前使用最广泛的套件管理程序之一，是利用数据库管理的方式进行套件的安装，具有相当容易的操作界面，而且套件查询验证功能相当强大，不过麻烦的地方在于它的属性相依问题

Tarball：直接在源代码经过编译后进行的安装

2、（1）RPM：RPM 是以一种数据库记录的方式将你需要的套件安装

到 Linux 主机的一套管理程序，其最大的特点是将你要安装的套件先包装好，通过包装好的套件中默认的数据库记录记录这个套件要安装时必需的相依模块（就是你的 Linux 主机需要首先存在几个必要的套件）

SRPM: SRPM 也是一种 RPM，但是由于它连同当初编译之前的源代码一起带在里面，所以可以重新编译。

(2)、RPM 和 SRPM 的格式

```
*****.RPM          RPM 的格式
*****.src.RPM       SRPM 的格式
```

(3)、Tarball 管理方法

Tarball 就是以 *.tar.gz 压缩之后的二进制原始文档。Tarball 的安装十分轻松，但其反安装（即删除）就十分困扰，因为它并没有记录当初安装文件的数据库，所以可能要手动删除每个文件

(4)、选择 RPM 还是 Tarball

如果你的版本支持 RPM，并且没有严重的相依属性的问题，选择 RPM 安装是一个比较好的解决方案

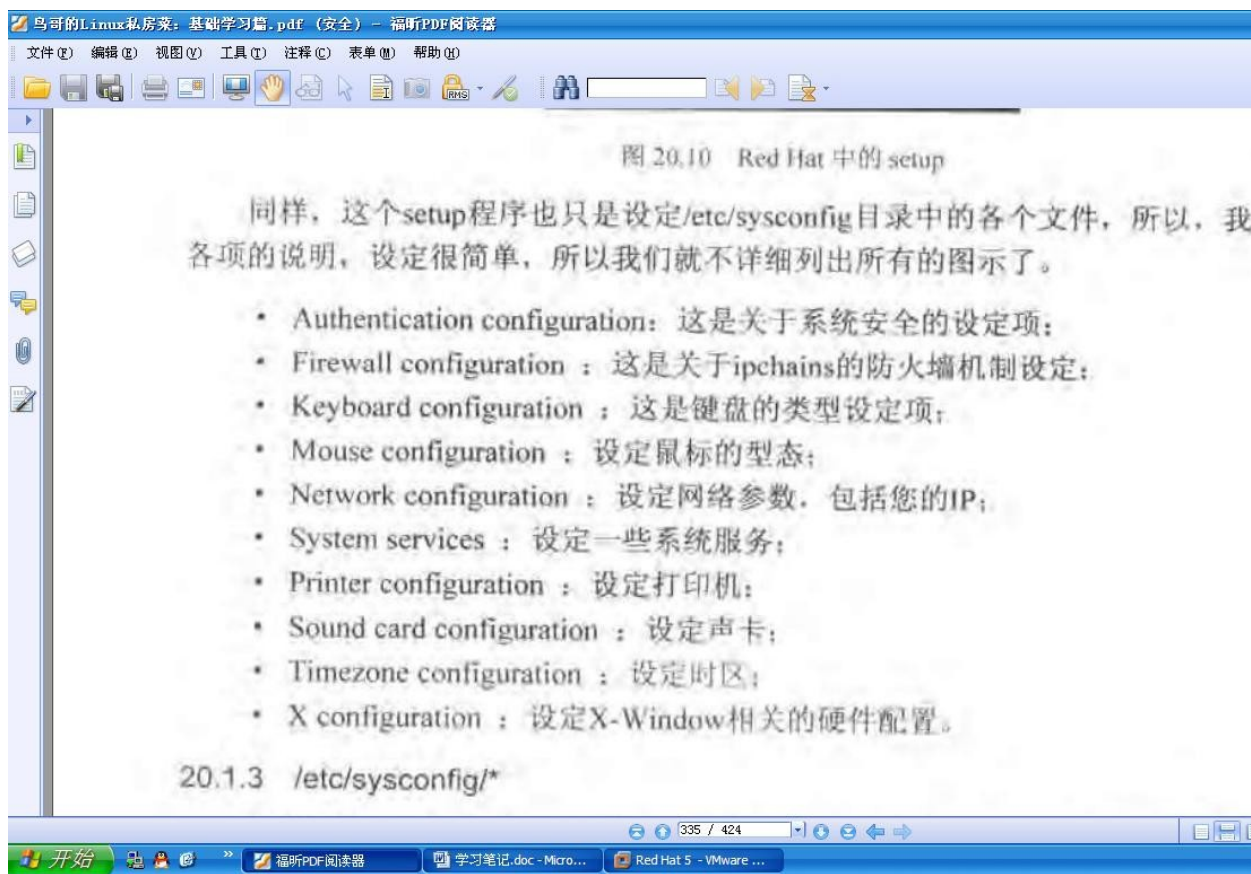
第 19 章 内核编译与多重启动

- 1、请特别注意，不同版本的内核之间最好不要互相升级，否则容易发生问题。如 Red Hat 6.x 使用的是 2.2.xx 版本的内核，那么就不要用 2.4.xx 的内核编译它
- 2、编译内核时，内核一定要放在 /usr/src/linux 目录下，但由于不论哪一版内核都是在这个目录下编译的，所以强烈将每一版的内核都使用不同的目录来指定，然后以链接方式将该目录链接到 Linux，后续动作在 linux 下进行

```
例： a) `cd /usr/src          //内核的默认 source 文件放置目录
      mv linux linux.old      //将旧的内核编译的目录搬移到其他目录
      tar -zxvf /root/linux-2.4.19.tar.gz    //将内核文件解压缩
      ln -s linux-2.4.19 linux    //建立链接
      cd linux                //进入工作目录
```

第 20 章 基本的系统设定指令

- 1、Red Hat 系统设置工具：setup



2、系统启动服务工具

(1)、service

语法：service 【service name】 【start | stop | restart】

参数：

service name：这个服务名称必须实际存在于/etc/rc.d/init.d 目录下才行

start | stop | restart：基本上，这个与/etc/rc.d/init.d/service_name 所允许的启动形式有关

功能：打开、关闭或重启相关服务

(2)、chkconfig

语法：chkconfig 【--level levels】 【--list | add | del】 【service name】

参数：

--level：同样地说明了可以设定不同 run-level 的东西

--list：列出 run -level 情况下该套件名称是否被开机启动

--add：增加一个 service_name 在开机时启动

--del：删除一个开机启动服务

Service name：这个服务名称必须实际存在于/etc/rc.d/init.d 目录下

功能：chkconfig 程序可以将/etc/rc.d/init.d 中的可执行服务与/etc/xinetd.d 中的相关服务链接到当前 run-level 中，或你可以定义你的 run-level 让你的该项服务在开机时启动。不过请注意，service 是马上让你的该项服务立刻启动，但 chkconfig 则只是设定某个服务在开机时可以被启动，并不一样

3、检查软件正确性：md5sum

语法：md5sum 【-bct】 filename

Mc5sum [--status | --warn] --check filename

参数:

-b : 使用 binary 的读取方式, 默认为 Windows/DOS 文件型态的读取方式;

-c : 检验 mc5sum 文件指纹

-t : 以文字型态读取 md5sum 的文件指纹

功能: 检查指定的文件有没有被修改过, 如果被修改过, 其指纹数据库会不同。必须先为系统重要文件建立指纹数据库

4、模块管理

(1) 、depmod

语法: depmod [-aens]

参数:

-a : 将 /etc/modules.conf 可以查询到的模块的相关性都写入 /lib/modules/'uname -r' 中

-e : 显示当前已加载的不可执行的模块名称

-n : 将各模块的相依性直接输出屏幕上, 而不是输出到 /lib/modules/'uname -r' 中

-s : 将一些结果信息输出到 syslog 的登录文件

(2) 、lsmod

语法: lsmod

功能: 查看当前系统中有哪些模块

(3) 、modinfo

语法: modinfo [-adnp] module_name

参数:

-a : 显示作者

-d : 显示这个模块的基本说明

-n : 显示这个模块的文件旋转路径与名称

(4) 、modprobe

语法: modprobe [-l|-c]

Modprobe modules_name //直接加载该模块与其相依属性的模块

参数:

-l : 列出当前系统所有模块

-c : 列出当前系统所有的模块 (更详细的代号对应表)

(5) 、insmod

语法: insmod [-fkps] module_name

参数:

-f : 强制加载模块, 要加载不同版本的模块时可能会用到这个参数

-k : 在内核没有乃至该模块时先自动清除干净

-p : 测试模块是否可以被加载, 不会直接加载到系统中

-s : 将一些执行信息写到 syslog 登录文件中, 而不是写到 terminal 中

功能: 加载模块

(6) 、rmmod

语法: rmmod modules_name

功能: 删除模块

第 21 章 认识系统服务

1、(1) daemons 目前有两种基本的模式:

- ◆ stand_alone : 该 daemon 启动之后一直常驻内存, 一直占用系统资源, 反应速度快
- ◆ super daemon : 当有请求过来时, 该 daemon 才会启动 (被从睡眠状态唤醒)。没有启动时不会一直占据系统资源。这种服务由 inet(xinet)负责

(2)、ftp 等相关服务的端口号在 /etc/services 文件里有说明

2、脚本存放位置:

- ◆ stand alone : 这个放置在 /etc/rc.d/init.d 目录里
- ◆ super daemon : xinet 也是一个 daemon, 它是 stand alone 启动的。Xinet 的启动脚本写在 /etc/rc.d/init.d/xinetd 这个脚本里, 挂在这个 daemon 里的服务的设定项写在 /etc/xinetd.conf 与 /etc/xinetd.d 目录下的任何文件

3、TCP_Wrappers: /etc/hosts.allow 与 /etc/hosts.deny

/etc/hosts.allow 与 /etc/hosts.deny 这两个文件的内容的基本语法是:

<service_name>:<ip,domain,hostname...>:<allow|deny>

例:

不想让 140.116.44.202 这个地址及 140.116.32.0/255.255.255.0 这个 C 类网域进入你的主机, 可以在 /etc/hosts.deny 中做如下设定:

telnetd:140.116.44.202:deny

telnetd:140.116.32.0/255.255.255.0:deny

这样, 对方就无法以 telnet 进入你的主机。

4、系统开启的服务

(1)、netstat

语法: netstat [-r] [-i interface]

netstat [-antulp]

参数:

- r : 显示 route
- i : 显示 interface 的内容, 跟 ifconfig 类似
- a : 显示当前所有的网络联机状态
- n : 默认情况下, 显示出的 host 会以 host name 表示。若为 n 则可以使端口与 host 都以数字表示
- t : 仅显示 tcp 包的联机行为
- l : 仅显示 LISTEN 的内容
- p : 同时显示此联机服务的 PID (只有 root 才能使用此功能)

第 22 章 分析登陆文件

1、登陆文件有几个重要的功能:

(1)解决系统错误 (2)解决网络服务问题 (3)记录登录信息

常见的登录文件有下面几个:

- /var/log/secure : 记录登入系统存取数据的文件, 如 ftp
- /var/log/wtmp : 记录登入者的信息数据, 必须使用 last 这个指令来取出文件的内容
- /var/log/messages : 这个文件很重要, 记录了系统发生的所有错误信息

(或重要信息)

- /var/log/maillog 或 /var/log/mail/* : 记录邮件存取或往来的用户记录
- /var/log/cron : 用来记录 crontab 这个例行性服务的内容

2、针对 log 文件的服务有以下两个

- syslogd : 进行系统或网络服务的登录文件记录工作 (默认开启的后台程序)
- logrotate : 将旧的数据更名, 并建立新的登录文件, 以保持登录文件的“新鲜”, 并根据设定将最旧的登录文件删除

3、(1)、syslogd 记录参数

打开 /var/log/secure 文件可以看到下面内容

```
Feb 5 10: 55: 58 localhost gdm[3550]: pam_unix(gdm: session): session closed for user root
```

事件发生时间 主机名称 启动此事件的服务名称 信息数据内容

(2)、syslog 配置文件 /etc/syslog.conf

这个文件规定什么服务要被记录、该服务的信息等级如何, 基本上, 可以使用下面的句法来说时:

格式: <服务名称. 信息等级> <存放或显示地点>

服务名称: 如 mail,http 等

信息等级: 总共分成下列几种等级:

- info : 提供一些信息数据
- notice : 需要比较留意的信息
- warint 或 warn : 警告信息
- err 或 error : 出现错误
- crit : 临界点, 再不处理可就伤脑筋了
- alert : 错误信息一再地警告, 你将完蛋了
- emerg 或 panic : 系统已进入混乱阶段, 真的完蛋了
- 特殊等级: 如 debug (显示较多信息) 及 none (不要记录该服务的内容) 等

4、登录文件的轮替: logrotate

(1)、Logrotate 参数的配置文件考虑两个地方:

/etc/logrotate.conf

/etc/logrotate.d

其中, logrotate.conf 才是主要的参数文件, 而 logrotate.dj 是一个目录, 里面的所有文件都会被主动读入 /etc/logrotate.conf 中

(2)、下面来看一下默认的 logrotate.conf 的内容

```
weekly           //每周进行一次轮替工作
rotate 4         //保留几个登录文件? 默认 4 个
create           //是否建立新的登录文件进行记录? 是
include /etc/logrotate.c //将这个目录中的所有文件都读进来执行 rotate 的工作
```

下面是 /var/log/wtmp 文件的 rotate 情况:

```
/var/log/wtmp{
    monthly       //每个月进行一次 log rotate 的工作
    create 0664 root utmp //将文件的权限设定为 664
    rotate 1      //仅保存前一个月的 rotate 备份, 这个项目可以改大一些
}
```