## 一 文件系统

1. chgrp groupname filename 改变用户所属组。
2. chown username filename 改变文件所属用户。
3. chmod 权限 filename 改变权限（其中在四位整数的第一位可以设定来修改SUID、SGID和SBIT的值）。
4. pwd 显示当前目录 –P选项显示真是路径（非链接）。
5. mkdir –m选项忽略umask的值，-p选项则递归建立所需目录。
6. rmdir –p选项递归删除所用目录，但该命令要求目录必须为空。
7. cp src dest –a=-pdr d说明复制链接文件属性（如果src为连接文件），p选项复制src属性而非使用文件默认属性，r保持递归复制，s选项复制成为符号链接。
8. rm f不显示提示信息，r递归删除。
9. mv f强制覆盖。
10. basename 显示文件名（不含路径）。
11. dirname显示路径。
12. head –n num filename。
13. od –t (o|c|a|d|x|f)。
14. touch 新建文件或者修改文件时间，ls默认显示mtime，-a修改atime，-c显示ctime且文件不存在时不新建文件。
15. umask 权限。
16. chattr +|-|= Sai filename修改隐藏属性，S标识文件同步写入磁盘，a标识文件除了增加内容外不能进行其它任何操作，i锁定该文件不能进行任何操作（包括root），ai选项只能由root设定。
17. lsattr filename 显示文件隐藏属性。
18. file filename 查看文件类型。
19. which command 在PATH中查找command并列出，-a使得列出所有找到结果，默认只列出第一个找到的。
20. whereis filename 在数据库中寻找文件。
21. locate filename 在数据库中寻找文件，-i选项忽略大小写，-r后跟正规表示法。
22. find path [option] [action] ：当option为-atime|mtime|ctime 时，n表示n天前的当天修改过的文档，+n表示n天之前（不包含当天）修改过的文件，-n表示n天内修改过的文件（包含当天），n从0开始。option也可以为-uid|gid|user|group|perm，perm后加[-|+|]mode，分别表示全部包含，部分包含和恰好等于mode的权限文件。
23. dumpe2fs 分区，用来查看装置的superblock等信息，-h仅列出superblock。
24. df 显示已挂载的装置文件，-h用常用单位显示，-i用inode显示容量。
25. du filename|path 显示filename或path中所有档案的容量。
26. ln src dest默认创建硬连接，-s选项创建符号链接，-f选项在dest存在时删除后建立。
27. fdisk 硬盘装置名 显示装置的分区信息，装置名不包含数字，即不是针对个别分区而是针对整个装置。-l选项显示所有装置的信息并进行操作。
28. ls -l /lib/modules/$(uname -r)/kernel/fs查看当前系统支持的文件系统。
29. cat /proc/filesystems 系统已加载到内存中支持的文件系统。
30. mkfs –t 文件系统格式 装置文件名 格式化文件系统。
31. mkfs[tab][tab] 显示mkfs支持的文件系统。
32. mke2fs [options] 装置名 设置该装置的ext文件系统参数。
33. fsck 装置名 该命令用来检测内存数据与磁盘数据的一致性或者数据的其它问题，-t指明fs类型，-a自动修复，-y同-a，-f强制检测，-C显示检测进度。它会将检查得到的错误信息填入文件系统的lost+found目录。
34. mount 装置名 挂载点：只输入mount会显示已挂载信息。-a使系统按照/etc/fstab中的信息进行自动挂载；-L允许使用label来挂载文件系统；-t选项可以指定挂载装置的文件类型，也可以让系统自动检测文件系统的类型，主要依据两个文件：/etc/filesystems和/proc/filesystems。常用文件类型：ext2, ext3, vfat, reiserfs, iso9660(光盘格式), nfs, cifs, smbfs(此三种为网绚文件系统类型)。
35. Linux 支持的文件系统的驱动程序都写在如下的目录中：/lib/modules/$(uname -r)/kernel/fs/。
36. mount -t vfat -o iocharset=cp950 /dev/sda1 /mnt/flash命令可以挂在u盘，其中-o选项指定了挂载文件系统所用的语系，它还可以指定在该文件系统上允许进行的操作等等，具体参见P276。
37. mount –o remount,attr1,attr2… 装置名：重新挂载文件系统并指定属性attr1等。
38. umount 装置名或者挂载点：-f强制卸载文件系统，-n不更新/etc/mtab。
39. mknod 装置名 bcp major minor：用来指定装置的主次号。
40. e2label 装置名 新label：修改装置的label值，系统允许两个不同分区使用相同label，但这样有时会造成混乱。
41. 可以对非空目录进行挂载，但是会暂时隐藏目录中原数据，直至卸载该文件系统。一般在单用户模式下，默认根目录为只读的，通常要使用指令mount –n –o remount,rw /来修改它的这一属性。
42. 可以通过修改/etc/fstab文件的内容来改变开机默认挂载方式。每次挂载后，系统会将最新的挂载信息填写在/etc/mtab和/proc/mounts两个文件中。
43. mount –o loop iso文件名 挂载点：可以挂载大的DVD的ios文件，并且可以对其内容进行修改。可以先用dd创建一个大文件，再用mkfs将该大文件制作成一个文件系统，最后通过该方法将该文件系统挂载，以此模拟一个磁盘上的文件系统（这种方法适用于磁盘已无空间建立新分区，而已有分区还有大量空间的情况）。
44. partprobe通知内核分区表已更新。
45. mkswap 装置名：创建swap分区。
46. free：显示内存和swap的使用情况。
47. swapon用来启动swap，swapoff关闭swap。
48. ll中的第二栏显示的是该文件占用的block数量，ll中的total是该目录中所有文件占用的block数量，一个block只能属于一个文件，所以文件的实际大小可能不等于它实际占用的空间大小，从而造成空间浪费。
49. parted/mkpart两个命令为新的GNU命令，允许建立2T以上的大分区。
50. /proc/sys/kernel/pid\_max中存储pid的上限。

## 二 压缩、打包与数据备份

1. gzip 文件名：-v选项在屏幕上显示压缩比例信息；-d选项解压缩gz文件。默认使用gzip后，源文件会被gz文件替代，可以使用-c选项将数据输出到屏幕，进而使用重定向将内容导入到其它文件，从而保留源文件：gzip –c 文件名>压缩文件名（自己定义）。
2. zcat gz文件：显示gz文件内容。
3. bzip2 文件名：-z是压缩参数，-d是解压缩参数，-v和-c同gzip，-k则保留源文件。bzip2和gzip对目录的压缩效果是对目录内的所有文件分别进行压缩。
4. bzcat bz2文件：直接显示bzip2文件。
5. tar –jcvf filename.tar.bz2 被打包压缩的文件或目录名
6. tar –jtvf filename.tar.bz2
7. tar –jxvf filename.tar.bz2 -C 欲解压的目录
8. j表示bzip2方式，z表示gzip方式，c表示创建压缩，t表示查询，x表示解压缩，v同gzip，f后跟压缩档案名，p表示保存档案原属性，P表示保留绝对路径（默认去掉根目录以防止解压时重要文档被覆盖），--exclude=FILE表示在压缩过程中不将FILE包含入内。解压时，在最后再加一个压缩文件中的文件名则单独解压出该文件而非解压所有。不需要压缩时不添加j和z选项即可。
9. --newer-mtime=时间：仅压缩mtime晚于指定时间的档案，--newer表示ctime和mtime。
10. dump –f 文件名 待备份资料：-S显示备份资料所需空间，-u更新/etc/dumpdates文件，该文件记录系统的dump历史，-j用bzip2等级2压缩备份数据，-level为备份等级（0-9），-W列出/etc/fstabl中有dump设定的分区是否备份过。
11. 一般dump备份整个文件系统，并且较高等价仅备份差异数据。针对非文件系统的目录进行dump只能使用0level即完整备份。
12. restore –f dumpfile：dump用来查看，还原dump创建的文件数据。-t查看dump文件内容，-C比较dump和其中档案现在情况之间的差异，-r还原整个文件系统（应该先创建一个文件系统），-i进入交互模式从而可以部分恢复数据。对restore的操作应该先进入dump备份的文件夹中。
13. 第9章中有iso、cd刻录和DVD等文件的处理命令。
14. dd if=input\_file of=output\_file bs=blocksize count=number：dd命令是针对扇区进行复制的命令，所以可以根据它的这一特性对分区进行复制。bs默认大小为512B（恰好为一个sector大小）。一个应用是备份一个完整的文件系统，流程为：创建一个和待备份文件系统大小相同的分区（或者更大），然后if为待备份文件系统，of为新分区，执行备份；如此备份不需要格式化新建分区，因为复制针对磁盘扇区，所以待备份中的superblock等信息也会备份到新分区，从而两个分区一模一样。还原时只要将if和of互换即可。
15. cpio –ovcB>[file|device] 🡪 备份命令；cpio –ivcdu<[file|device] 🡪 还原； cpio –ivct <[file|device] 🡪 查看。o输出数据到装置货文件，i与o相反，v在屏幕上显示信息，d在还原时自动建立目录，u自动覆盖旧档案，B将预设的blocks从512B增大到5120B，t用于查看cpio档案，c使用较新的portable format方式存储。
16. 一般cpio命令要配合find等查找命令，再利用管道和数据重定向来实现备份，因为cpio不提供待备份文件或目录选项。

## 三 账号管理和ACL权限管理

1. 用户名与账号的对应在/etc/passwd文件中，相应的组对应配置文件在/etc/group中记录，密码信息存储在/etc/shadow文件中，组密码的信息存储在/etc/gshadow文件中。
2. uid为0是root，1-499为系统保留账号（权限和普通类似），其中1-99为distributions自行建立账号，100-499供用户有系统账号需求时使用，500-65535给一般账户使用。
3. 在/etc/shadow中有密码过期、密码失效和账号失效的栏目，密码过期仍然可以登录系统但是会收到系统要求改密码的警告，而密码失效则无法登陆系统，账号失效则是该帐号无法再使用。
4. groups：直接显示当前用户支持的群组。群组分初始群组和有效群组。该命令输出的第一个群组即当时的有效群组。
5. newgrp 组名：切换用户的有效群组，组名必须为用户所支持的群组。切换的本质是创建一个新的shell，因此使用exit时可以退回原shell中，有效群组也切换为原值。
6. useradd 用户名：-u指定uid，-g指定用户组，-G指定次要用户组，-M不建立主目录，-m建立主目录（默认，权限为700），系统账号默认不建立主目录，-d指定主目录，-r建立系统账号，-s指定shell。

-D会显示当前的默认值，这些默认值在/etc/default/useradd文件中，其中的SKEL指定了家目录的默认数据，相关数据在/etc/skel目录中。

对于uid和gid等内容的参考依赖于/etc/login.defs文件。

useradd建立账号后默认是锁定的，需要用passwd命令创建密码再使用。

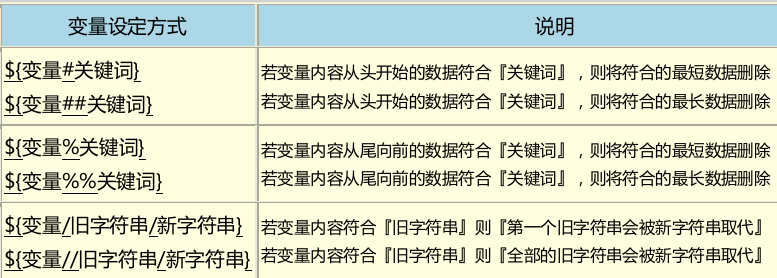
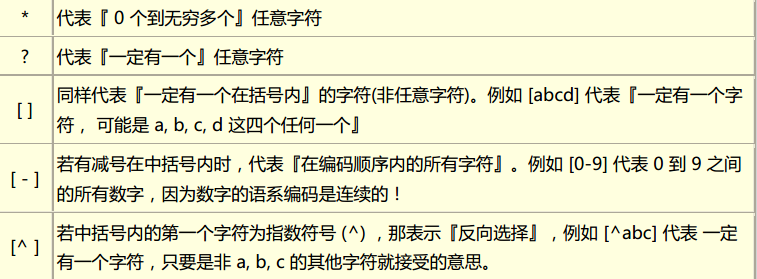
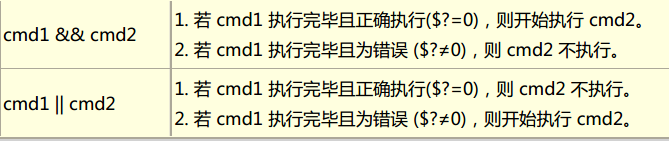
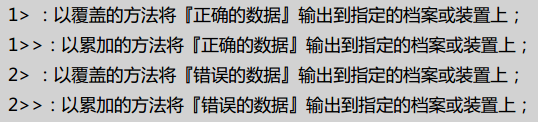
linux的群组机制分为私有群组机制和公共群组机制，不同的distributions采用不同机制，CentOS为私有，即创建一个与账号同名的用户组，忽略/etc/default/useradd中的设置。

1. passwd [帐户名]：修改账户密码。-l选项锁定用户（无法登陆，实质实在/etc/shadow中的密码栏添加个别字符实现），-u选线解锁。配有相关选项来修改/etc/shadow中的时间选项。
2. chage帐户名：各种选项用来修改/etc/shadow中的时间选项。

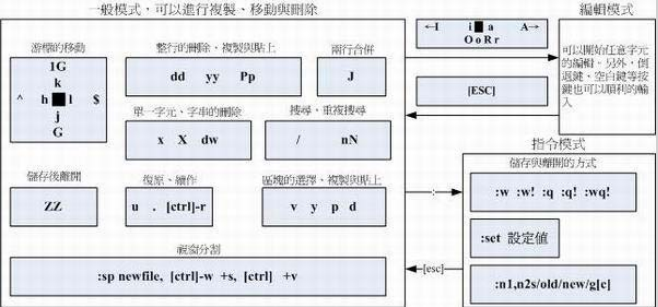
/etc/shadow中各时间栏的含义：第三栏表示密码最后一次修改时间，第四栏表示多久后才能修改密码（相对3），第五栏表示多长时间内要求修改时间（相对3），第六栏表示相对于5发出的警告天数，第七栏表示过了5后宽限日期（过期后密码失效），第八栏表示账号失效日期。其中第三和第八栏的日期都是相对于1970年1月1日来的天数。

1. usermod 帐户名：各种选项实现修改账号属性信息。-a与-G合用可以添加次要群组，仅有G则设定为给定值。
2. userdel 帐户名：-r选项删除主目录。
3. date命令显示时间。date --date='2 days ago' +%Y%m%d后面为设置输出格式。y两位年|Y四位年|m月|d日|H时|M分|S秒|h月的单词表示。
4. finger [帐户名]：显示账户信息（主要依据shadow文件），不加帐户名时显示当前登入系统的账户以及他们的登入时间等信息。
5. chsh：切换当前shell，-l列出当前能使用的shell（即/etc/shells文件中的内容），-s shell修改为shell。因为chsh修改的是shadow的内容，所以它是一个suid程序。
6. id 帐户名：查看账户信息。
7. groupadd –g gid [-r] 组名：r指明该组是一个系统组。
8. groupmod 群组名：-g修改gid，-n 新组名修改组名为新组名。
9. groupdel 组名：待删除的组名必须不能为/etc/passwd中任意用户的初始群组。
10. gpasswd 组名：-A将后接的用户设置为组管理员，-M将后接的用户（可为多个）加入组中，-r移除密码，-R是密码失效，不加选项则创建组密码。以上为root能够执行的操作，下面为组管理员执行的操作：-a将后接用户加入组中，-d将后接用户从组中删除。一个组管理员可以不是这个组中的成员。
11. ACL提供了对单一文件或者目录的权限访问设置，通过命令getfacl和setfacl对其进行设置，设置后的档案权限最后会有一个+号。
12. sudo使得用户可以使用其它账号的权限来执行命令，不需要输入密码，而是root通过visudo命令修改/etc/sudoers文件来控制每个账号的sudo执行权限，文件中不存在的账号不能执行sudo命令，sudo格式通常为：sudo –u用户名命令。
13. PAM（嵌入式认证模块）：它是一组API，可以供任何程序调用来实现自己的验证机制。首先程序调用PAM，PAM到/etc/pam.d目录中寻找与程序同名的配置文件，再依配置文件中的设置进行相关验证并返回验证结果，程序根据验证结果采取相应的应对措施。
14. PAM的常用目录：/etc/securetty 会影响到root可登入的安全终端机， /etc/nologin 会影响到一般使用者是否能够登入功能外（当该文件存在时，系统中的一般用户将不能登陆，已登录一般用户除外），PAM 相关的配置文件在 /etc/pam.d，说明文件在 /usr/share/doc/pam-(版本)，模块实际在 /lib/security/。/etc/security这个目录内也有一些PAM的相关配置文件。
15. /etc/security/limits.conf文件限制了每个用户的资源使用。这个文件设置后即生效而不用重启系统（对已登入的用户需重新登陆）。
16. lastlog指令会读取/var/log/lastlog文件中数据显示用户最近登录信息。
17. write 用户 终端（用w/who查询）：用来向用户发送信息，以eof结束。
18. wall：向所有登录用户发送消息。
19. mesg n/y：关闭/开启消息接收，无法拒绝接受root发来的消息。

## 四 BASH

1. 应注意bash中的通配符和re不同，右图列出shell中的常用通配符
2. /etc/shells文件罗列了系统能够使用的shell种类。
3. alias 别名 “cmd”：设置命令别名。
4. type 命令：判断命令是bash内建命令还是外部文件。
5. 变量中若有空格则可以使用单引号或者双引号括起来，其中双引号中的$等符号将保持原特性，而单引号内的内容则为原样输入。当没有括号时，也即不需要加空格时，默认为保持特殊符号的原特性。
6. 变量中可以使用\符号来对特殊符号进行转义。
7. 在指令中使用其他指令提供的信息，可以使用`cmd`或者$(cmd)来引用，`符号为esc下的按键，称单反引号。如version=`uname –r`或version=$(uname -r)。
8. export var：将var变量设置为环境变量，这样在bash的子程序中也可以使用该变量，比如执行了bash打开新shell后，var仍然有效。
9. 自定义变量在子程序中是不可见的，而环境变量在子程序中可见。
10. unset var：将变量var删除。
11. set显示当前环境变量和自变量的内容，其中PS1显示命令提示符，其中有各种转义字符，$为当前shell的pid，?显示上一个命令的执行返回结果。
12. 默认LANG和LC\_ALL的设定是各种默认的编码，但可以被各单独覆盖，locale显示各种编码的选择，-a显示当前支持的语系。语系文件都放在/usr/lib/locale中。
13. 系统的整体默认语系保存在/etc/sysconfig/i18n文件中，可以通过修改它来改变当前语系。在tty下要想显示中文必须安装一些第三方软件，因为默认tty是无法显示中文的，但在xtty下是可以的。
14. read var：输入后会提示输入变量值，并将该值赋予var。-p选项后可接一句提示语句（加引号），-t选项后跟数字表示等待时间。
15. declare var：-a声明var为数组类型，-i声明var为整数，-x将var变为环境变量，-r将var变为只读（取消只读只能注销），+x将var还原为自定义变量，-p显示单独变量的类型。在bash中两个整数相除得到结果仍未整数。数组类型可以直接赋值使用而不用显示声明-a，比如var[1]=hello。对该数组元素的调用为${var[1]}。数组的下标最小为0。
16. 一些变量设置方式。
17. ulimit：用来显示或者设置允许一个shell使用的系统资源数量P376。
18. 这些并不会改变原变量的值，而是生成一个修改后的副本返回，其中的关键词可以使用通配符。
19. history指令：~/.bash\_history文件，注销时更新文件，！！执行上一条指令，！com执行com开头的指令（最近执行过），！n执行最近第n条指令。
20. 修改/etc/issue来修改登录提示语。使用telnet登录时显示的是/etc/issue.net中的内容。修改/etc/motd来显示用户登录后shell显示的提示信息。
21. bash的默认环境设置分为2个流程。先读取/etc/profile文件，该文件参考/etc/profile.d目录中的脚本来进行全局默认设置，然后读取~/.bash\_profile或~/.bash\_login或~/.profile（依顺序仅读取一个）来进行每个用户自己的设置，该文件最终还是调用~/.bashrc中的设置，它可以覆盖前面文件中的设置。该文件的本质也是去调用/etc/bashrc文件。
22. source filename：执行filename脚本。
23. 第3条所列为login-shell的情况，还有nonlogin-shell，即不需要输入密码验证的shell（比如执行bash和在x窗口下新建一个xtty），这种shell仅读取~/.bashrc文件。
24. /etc/man.config中设置了MANPATH变量的值，作为查找man文件的依据。
25. ~/.bash\_logout文件规定了用户注销时系统执行的工作。
26. stty命令用来设置tty的属性信息。
27. 
28. 使用cat > file可以新建一个file并且提示输入信息，默认crtl+d（eof）结束输入。也可以使用重定向<从一个文件输入数据，当使用<<时后接终止符，当输入该符号时（可以是一个字符串）终止输入。
29. 
30. 由11显示的性质可以由cmd1&&cmd2||cmd3来表示条件表达式。
31. last：显示最后登录的用户登录信息。
32. cut –d ‘分隔字符’ –f n：cut是针对行来进行文字处理的。对每一行依据分隔字符将文本分成若干段，然后指定n来显示第n段（每行都显示），多个段可以用’,’连接。也可以用cut –c n1,n2来显示每一行中的第n1到第n2个字符。
33. echo的-e选项对后面内容中的特殊字符进行转义，而且需要使用双引号，尤其是在引用一个变量的时候。
34. grep “str” filename：与cut不同，cut将每行的内容做定制显示出来，而grep则将包含str的行的整行信息显示出来（仍然分行）。-n显示行号，-i忽略大小写（默认区分大小写），-v反向选择即不包含str的行显示，-c计算显示找到str的次数（不再显示该行）。--color=auto会使得str高亮显示，-An和-Bn分别显示搜索行的后n行和前n行。
35. sort对结果进行排序，它有许多排序选项也包括一些字段选择项，但也可以使用grep或者cut现将信息处理然后用管道导入sort的标准输入再进行排序。P402
36. uniq：将显示的信息去重，-c显示计数，-i忽略大小写。
37. wc [filename]：显示字数统计（默认全部显示），-l显示行数，-m显示字符数，-w显示英文单词数。
38. tee filename：将数据既导向stdout又保存到filename文件中。
39. 若要将两者同时导入同一文件可以使用>list 2>&1或者&>list。若是>list 2>list会导致文件结构错乱。

## 五 vim编辑器的使用

1. 使用vim编辑程序时会自动生成.filename.swp用来保存文件发生的变化，当vim正常退出时，该文件会被自动删除。
2. vim中的行选择，v是光标移过的区域选择，V是光标移过的行选择，crtl+v是矩形选择（和v不同，它是基于形状而不是基于行的）。
3. 在vim命令后跟多个文件名，可以同时打开多个文件。使用:files查看vim打开的文件列表，:n开始编辑下一个（列表中）文件，:N开始编辑上一个文件。不同于多窗口，该方法仅有一个窗口，即只能编辑当前文件。
4. 在vim中输入:sp [filename]将分割窗口打开另一个文件filename（缺省为本文件），可以打开多个，使用crtl+w+j向下移动，crtl+w+k向上移动。在对应的窗口输入:q退出当前窗口。
5. ~/.viminfo记录了vim的使用记录（里面存储了高亮信息），可以编辑查看，~/.vimrc设置了自己的vim设置，/etc/vimrc则存储了vim的全局设置。
6. 一些常用的设置（可以写到~/.vimrc文件中）：set hlsearch（默认）高亮搜索，输入nohl取消当前高亮搜索结果，set autoindent自动缩进，set backup自动生成备份文件filename~，syntax on/off是否高亮关键字，set backspace=(012)设置删除键可以删除的字符（0和1只能删除刚输入的字符，2可以删除任意字符）。
7. 可以设置vim的自动缩进，分为三种cindent，smartindent和autoindent，还有设置tab的宽度（set tabstop=n），缩进所占字符长度（set shiftwidth=n），输入n<<减少缩进（以shiftwidth为单位，即shiftwidth\*n），同理，n>>增加缩进。set expandtab将tab转换为相应数量的空格，修改当前文件的tab则需执行:retab!。应注意，expandtab并不是将现有tab转换为空格，而是将以后输入的tab转换为空格，retab相当于:set noexpandtab命令。如果想将已输入的tab转换为空格应用文本替换（如第10条所示）。相关资料可见有道笔记。
8. 常用命令如图：
9. x相当于delete，X相当于backspace，ZZ直接存储退出，J把当前行和下一行合并。$到该行末尾，0到该行起始处。
10. :n1,n2s/old/new/g：文本替换，在n1到n2行寻找，若是真个文件则直接s。
11. dos2unix file [newfile]：文件格式转换，-n选项生成新文件newfile。类似的还有unix2dos命令；-k保留原文件的mtime。
12. iconv –f 原语系 –t目标语系 原文件 [-o 目标文件]；将原文件的编码改为目标编码，-o生成新文件，-l显示当前支持的编码。
13. vim中关于语系编码的选项主要为encoding，fileencoding，fileencodings和termencoding四个选项，前三个的设置较为重要，详细见有道笔记。

## 六 正规表示法和文件格式化处理

1. 每一个代表一类字符。
2. 当使用-符号表示一个范围时，应该注意语系对其的影响。只有在当前语系中编码连续时，使用-符号才有意义。
3. []代表选取其中的一个字符，[^]中的^符号代表反向选择，可以在[]中使用-符号。
4. ^不用在[]中时表示一行的开头，$符号表示一行的结尾。^$表示一个空行。
5. re中的\*和通配符\*不同，re中的\*表示任意个前面的字符，.代表一个任意字符，所以re中的.\*效果和通配符中的\*相同。如g.\*g表示两个g中有任意个任意字符（可以为0个）。
6. {n1,n2}表示前面的字符出现n1到n2次，如果是在bash中直接输入，则需要对{和}进行转义（\{和\}）。
7. sed命令是一个主要以行为基础进行文本处理的命令。命令分为两部分，一部分是命令的参数，一部分是对文本执行的操作。对于参数，-n表示安静模式，只显示修改的行而不显示所有数据。-i直接修改文件内容，-e（默认）在命令模式下进行编辑（不修改文件内容和-i相对），多个操作需要分别使用-e。-r支持延伸的re。-f filename将sed的动作写入filename文件。 对于行的动作为’n1,n2func [text]’，即对n1到n2行执行func操作，a为在下一行插入一行text内容，i为在上一行插入一行text，c为将n1到n2内容替换为text，d则直接删除相关行，p为显示相关行。 sed也可以对文件中的内容进行替换修改，用法同vim中的文本替换类似：’n1,n2s/old/new/g’。
8. grep默认仅支持基础re，-E选项支持延伸re，功能相当于egrep命令。延伸re以下几个：+表示1个到多个前面字符，?表示0个或1个前面的字符，|表示or（非管道），()表示一个群组（如(abc)表示串中有abc），()+表示一个或多个()的串。
9. awk ‘条件1{动作1}条件2{动作2}…’：awk是一个文本处理工具，它是以列为操作对象的（类似cut），默认的分隔符为空格和tab，awk先将文本依据分隔符将每一行进行分割，然后用$1,$2…依次代替各字段的内容，$0代表该行内容。几个重要的变量：NF为每一行拥有的字段数量，NR为当前行号，FS为分隔符，BEGIN表示第一行，END表示最后一行，FS可以进行设置（在动作中）。条件则是一些判断式。一般将awk和printf命令结合使用。详见鸟哥P435。
10. diff fromfile tofile：diff用来比较两个不同的文本文件并且以行为比较单位，比较结果形式为n1 func n2（没有空格，仅为表示清楚），n1表示fromefile中的行号，n2为tofile中的行号，func与sed命令中的func相似。-b忽略同一行中的空格差异，-B忽略空白行差异，-i忽略大小写。diff –Naur fromfile tofile>patchfile用来生成patch文件。
11. cmp file1 file2：用来依据字节比较两个文件，-s显示所有不同点，默认显示第一个。
12. patch –pN<patchfile用补丁文件patchfile更新文件，加上-R选项用来还原文件。p表示取消基层目录的意思。

## 七 SHELL编程

1. 直接用脚本名执行需要rx权限，使用sh来执行只需要有r权限。
2. 在脚本中可以使用exit n来设定一个返回值，该值可由$?返回。注意若要使用source则会推出当前bash，而是用另几种方式是退出子bash，返回值会被父进程获取。
3. var=$((运算内容))这样可以直接使var作为整型变量，而不用declare –i来声明var。
4. 应注意，直接执行脚本或者使用sh/bash来执行脚本的实质是新建一个bash子进程来运行该脚本，因此脚本中的变量在执行完回返后不存在（无论是否环境变量）；而使用source来执行脚本时是在当前 bash中执行，因此变量在执行完后仍存在。
5. test用来测试一个文件的属性信息，两个数的大小，两个字符串是否相等等，并给出true或false结果。详见P453
6. 可以使用一对[]来代替test来进行条件判断，前提是中括号两端和内容之间应该至少有一个空格。这种方法经常用在if语句中。
7. 执行脚本时，系统自动设置脚本程序中一些默认变量（内置变量），输入的第一个字符串（通常为文件名）为$0，后面的参数依次为$1,$2…参数的总数量（不含文件名）为$#，$@代表[ “$1” “$2” ... ]等。
8. shift n：直接将参数列表进行偏移，即删去前n个参数。用来修改脚本内置变量。
9. if语句：if[]; then 语句 elif[]; then 语句 else 语句 fi
10. case语句：case $变量 in 常量1)语句;; 常量2)语句;; … \*)语句;;esac
11. 函数function：function funcname(){语句}。函数不用显示地声明参数，只要在调用的时候在函数名后加上即可。函数也有内置变量，$0代表函数名，$1,$2…代表参数。比如funcname var1，则函数中var1就作为$1，funcname作为$0，函数调用时不用加括号。
12. while [ condition ] do 语句 done：当condition成立时循环。
13. until [ condition ] do 语句 done：当condition成立时结束循环。
14. for var in var1,var2…/$变量 do 语句 done
15. for (( 初始值;条件;步长 )) do 语句 done
16. sh 脚本：-x则将每一步执行过程都输出，-n不执行脚本仅检查语法，-v在执行前将脚本内容输出。

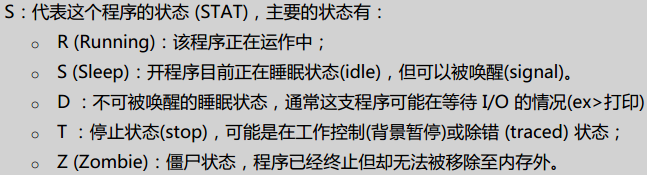
## 八 Quota

1. quota指磁盘配额，能够针对每一个用户或者每一个用户组，来面向一整个文件系统进行磁盘容量使用限制或者文件数量限制。并非所有文件系统支持quota，支持quota的文件系统在挂载时应加入支持选项usrquota和grpquota，文件系统的顶层目录会建立aquota.user和aquota.group文件记录该文件系统中每个用户和群组的使用情况。
2. 针对每个user/group有两个限制值soft/hard和宽限时间（默认7天），hard表示用户/群组实际拥有的空间容量，soft表示超过该值时显示警告，在超过宽限期限后而没有把将使用容量降至soft以下则用户仅能使用soft大小容量。0表示无限制。
3. 当同时设定了user和group的quota时，同组内的所有user容量相加不能超过group的设置，即使所有user的给定值的和比group大。
4. quotacheck [挂载点]：扫描文件系统并建立quota记录文件（如1中所述）。-a扫描所有文件系统，-v显示扫描过程，-u针对用户的使用情况在文件系统建立quota.user，-g针对群组建立quota.group，-f强制扫描文件系统并写入新quota文件，-M强制以读写的方式扫面文件系统。
5. quota配置文件非纯文本文件，由系统自动维护，不应主动去编辑它。
6. quotaon [挂载点]：参数和quotacheck类似，用来启动quota。启动一次后，系统的初始化脚本/etc/rc.d/rc.sysinit会在以后的系统启动过程中自动执行该操作。
7. quotaoff [挂载点]：参数同上，用来关闭quota。
8. edquota [-u username] [-g groupname]：用来设置每个用户或者群组的quota，会进入相应的vim修改界面，分别设置文件大小和inode的soft/hard（以kb为单位）。-t修改宽限之间，-p用来指定范本用户，如edquota –p user1 –u user2表示参照user1来设定user2。
9. quota [-u username] [-g groupname]：用来显示username或者groupname的quota信息，-s以K/M/G等单位显示，-v显示输出。
10. repquota选项与quotacheck类似，功能同quota类似，只是它依据/etc/mtab中的信息来显示所有支持quota的文件系统的quota信息。
11. warnquota：向所有磁盘超过soft的用户发送邮件提醒。修改/etc/warnquota.conf可以定义邮件格式。当用户的邮箱所在文件系统也超额时将无法收到邮件。
12. setquota [–u|-g] username/groupname 大小soft 大小hard inode的soft inode的hard 文件系统：直接使用指令修改指定用户或者群组的quota设置。

## 九 at/crontab

1. at是一次执行的工具，后台服务为atd。at产生的指令以文本文件的形式写入/var/spool/at/ 目录内，等待系统执行。
2. at首先查询/etc/at.allow，不在其中的用户无法使用at，此时at.deny文件被忽略；若不存在/etc/at.allow，则参照/etc/at.deny，除了其中的其他用户都可以使用at；若两个文件都不存在，则只有root能使用at。（通常系统中都默认有一个at.deny。）
3. at TIME：在TIME后执行指定的指令，TIME格式有1.HH:MM，2.HH:MM YYYY-MM-DD，3.HH:MM/now+n[minutes|hours|days|weeks]。执行后提示编辑执行指令，最后以EOT（crtl+d）来结束输入。
4. at的执行是与当前的bash无关的，所以指令为echo时不会在屏幕上显示，而是会将所有输出发送至执行at指令的用户邮箱，若要显示可以重定向>/dev/ttyn（n为当前的tty编号）。如果没有输出则不发送邮件，-m选项可以强制发送一封邮件。-c n列出号码为n的at任务的实际工作内容（每个at任务都有一个编号）。
5. atq：用来查询at任务，等价于at –l。
6. atrm n用来取消编号为n的任务，等价于at –d n。
7. batch：利用at的机制，和at用法相同，仅仅是加入了一些参数控制系统在不忙碌时再执行指定的任务。
8. cron的服务为crond，权限文件为/etc/cron.allow和/etc/cron.deny，用法和at相似。
9. cron建立的任务会被记录到/etc/spool/cron目录中，每个使用cron的用户在该目录下有一个以用户名命名的文件，其中记录了他的cron任务，不应用vim直接编辑它。每个cron执行的任务都会被记录到/var/log/cron文件中。
10. crontab –e：执行指令的当前用户的cron任务编辑，分为6个字段：分 时 日 月 周 指令。\*代表任意，逗号分隔多个时间点，-代表一段时间内，/n代表每隔n时间（不同的字段代表不同的时长）。-l显示当前用户的cron任务，-r删除当前用户所有任务，删除单个任务应该使用-e去删除一行，-u username只能由root为其他用户指定任务。
11. root修改/etc/crontab文件，可以指定系统定时执行的任务。run-parts为/usr/bin/run-parts脚本，它后面加一个目录执行该目录中的所有脚本。
12. /etc/crontab中分别定义了每小时，每天，每月和每周执行对的任务（通过run-parts实现），实质是执行/etc/cron.hourly，/etc/cron.daily，/etc/cron.monthly和/etc/cron.weekly四个目录下的脚本，所以可以将自己要执行的脚本放在这些目录下由系统定时执行。crontab中还可以设定执行脚本的用户。系统每分钟会扫描该文件。
13. 指定任务时，周与日月不可并存。
14. anacron：-s开始根据时间文件记录选择执行各项任务，-f强制执行，-n无延迟执行，-u仅更新时间戳而不执行，job由/etc/anacrontab定义的各项工作。几个重要的文件/etc/anacron，/var/spool/anacron/cron.\*以及/etc/cron.\*目录下的0anacron文件。详见P596

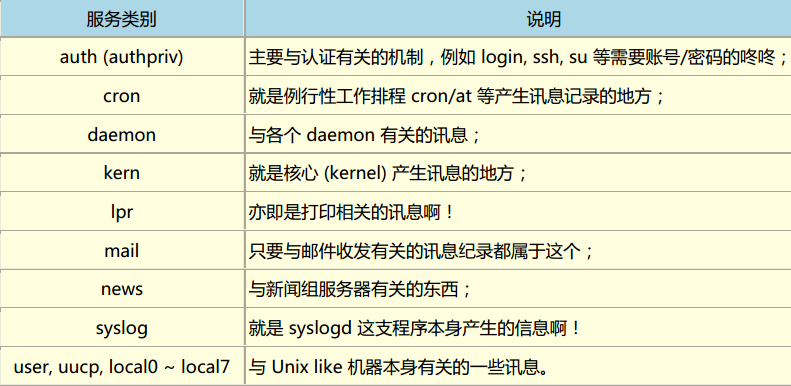
## 十 程序管理

1. bash下使用&执行后台（当前bash的后台，非系统后台）操作时，会显示该作业在该bash下的工作号和它的pid。
2. 程序执行时，crtl+z会使其暂停编程后台程序，带+表示最近一个被丢进后台的工作，-表示最近第二个被丢进的。后台程序通常有stopped和running两种状态。
3. jobs：显示当前bash的后台程序状态属性，-l显示pid，-s仅显示stopped进程，-r仅显示running进程。
4. fg [%n]：默认将带+的程序取出执行，加%n将job号为n的取出执行，fg –将带-的程序取出执行。
5. bg %n：在后台执行job号为n的程序。
6. kill –n %jobn：-l（小写L）为列出当前能使用的信号，-1重新读取配置文件（reload），-2类似ctrl+c，-9强制关闭，-15正常终止，-17暂停程序运行。jobn为job号，也可以直接使用pid（不加%）。
7. 上面的后台执行都与当前bash有关，当bash退出后，所有前后台程序都会终止。若要交给系统后台运行，使用：nohup 指令 [&]，即使bash退出程序也将继续运行。并且程序的输出会导入到~/nohup.out文件中。
8. ps –l：显示当前bash下的程序运行信息（执行ps指令的这个时间点）。
9. ps aux：显示系统所有程序信息。cmd后显示defunct表示为僵尸进程。详见P611
10. top：动态显示进程信息，-bn n1显示n1次进城信息，-d n2每隔n2秒刷新一次，-p指定具体pid进行观察。进入top后可以输入一些指令，q退出，P/M/N分别以CPU/内存使用和pid进行排序，k给某进程信号，r给某进程nice，?显示指令。
11. pstree显示进程树。
12. killall -n cmd：所有以cmd启动的程序都会被发送信号。
13. 程序的优先级一般由PRI和nice值指定，PRI由内核维护用户无法修改，用户仅能修改nice，root能任意修改任何人nice且范围-20-19，普通用户仅能修改自己nice且范围0-19，且用户的nice只能越调越大。
14. nice –n num cmd：直接使用num作为nice执行cmd指令。
15. renice n pid：重新修改pid进程的nice。
16. /var/log/messages是系统日志文件，纯文本，每行代表一个事件。
17. free –b|m|g：分别以kb|mb|gb为单位查看内存使用情况。
18. uname：查看系统信息。详见P622
19. uptime：显示系统的负载。
20. netstat：输出分本系统状态和网络程序连接状态信息两部分。-t显示tcp包数据，-u为udp，-l列出正在监听的服务，-p列出pid，-n显示端口号而非服务名，-a显示所有。其中的Recv-Q表示在缓冲区尚未recv的字节数，Send-Q表示已发送但尚未收到确认的字节数。
21. dmesg：显示内核从启动开始到现在的所有输出信息。
22. /proc/\*：文件夹主要存储了系统的一些重要信息，每个进程会以pid为名有一个文件夹，里面存储了该进程的一些信息，如cmdline文件记录了启动命令，environ记录了环境变量内容。详见P628
23. fuser file/dir：列出file或者dir的使用进程，-u显示进程所属用户，-m会将file和dir直接上升至所在文件系统的最顶层，-v列出进程的命令的相关属性，-k找出使用档案的pid后试图向他发送SIGKILL，-i配合k进行询问。
24. lsof：列出某用户使用的文件和目录，-a条件连接需要同时成立，-U仅列出Unix Like系统的socket文件类型，-u username列出username开启的文件和目录，+d dir列出dir下被打开的文件和目录。没有参数则输出系统当前所有打开的文件和目录。
25. pidof cmd：查询执行cmd生成进程的pid（可能有多个，有时需要使用cmd的绝对路径）。

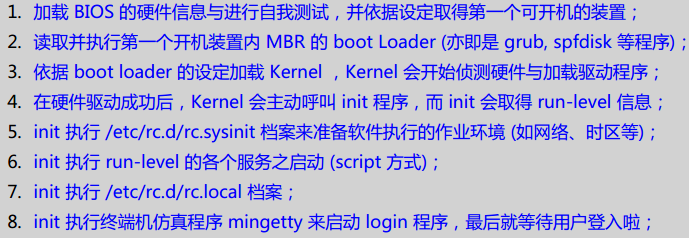
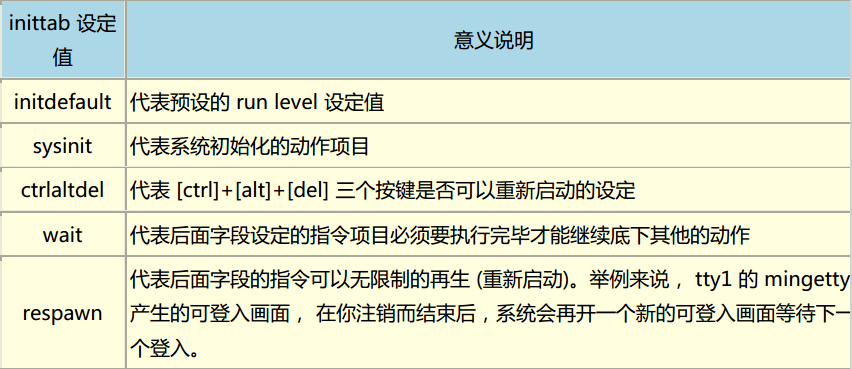
## 十一 系统服务

1. 服务分为stand alone和xinetd同一管理，前者一直在内存中，后者仅在运行时由xinetd加载至内存，运行结束后在从内存移除。
2. 服务与端口号的对应信息存储在/etc/services文件中。
3. /etc/init.d/\*：放置启动脚本的目录。centos中的实际位置是/etc/rc.d/init.d/\*。
4. /etc/sysconfig/\*：各个服务的初始化脚本文件。syslog的配置文件/etc/sysconfig/syslog，网络的配置文件/etc/sysconfig/network。
5. /etc/xinetd.conf为xinetd的配置文件，/etc/xinetd.d/\*为xinetd管理的各服务的配置文件。
6. /etc/lib/\*：各服务产生的数据库，mysql的为/etc/lib/mysql/目录。
7. /var/run/\*各服务的pid记录处，分别记录在服务名.pid文件中。
8. 开启stand alone服务方法：/etc/init.d/\* start|stop|restart|status|crondstart
9. service 服务名（非路径） start|…：功能同上，--status-all会显示所有stand alone服务状体，service是/sbin/service脚本。
10. xinetd也是一个stand alone服务，所有可以使用/etc/init.d/xinetd文件进行服务操作，若要管理xinetd下的服务，可以修改/etc/xinetd.d/\*下的相应服务配置脚本，然后重新启动xinetd服务生效。
11. 在/etc/init.d/xinetd中的脚本中可以对同一服务设置多段，分别针对不同的情况（如多个ip）。脚本中的only\_from和no\_access选项实现了一种防火墙的功能。
12. TCP Wrappers是一种tcp的防火墙机制，linux中使用/usr/sbin/tcpd来分析tcp网络封包，因此根据tcp包中的ip和端口信息决定其是否被处理。该程序的配置文件是根据/etc/hosts.allow和/etc/hosts.deny两个配置文件来设定防火墙。所有有xinetd管控的服务都受这两个文件的控制（该服务的脚本中可以覆盖该设置）。支持Wrappers的程序也受这两个文件的限制，查看一个程序是否支持Wrappers的方法见13条。
13. ldd 程序名（绝对路径）：查看程序使用的动态函数库。如查看一个程序是否支持Wrappers，就是ldd cmd，看cmd是否支持libwrap.so这个函数库。
14. hosts.allow中格式为：服务名 ip段/固定ip [allow]，hosts.deny中的设置与此相同（除最后为[deny]），第三个字段的allow/deny可以省略。系统先检查allow文件再检查deny文件。若一个网段不在allow中，再以deny作为依据。
15. 安装tcp\_wrappers软件后，可以在allow/deny文件中的第三字段添加信息来对特殊网段的访问进行后续的处理，spawn后接shell指令在服务器上执行一些指令，twist后接指令可以直接在客户机上执行一些指令并且终止该次联机。
16. chkconfig --level [0-9] [--list] service on|off：chkconfig命令用来设置开机启动项，--level指定启动等级（0-6），--list来查看指定等级的启动项。它仅修改开机启动设置，而不对服务的当前运行状况产生影响。xinetd的start/restart也会参照chkconfig的设置。
17. chkconfig --add|--del service：可以添加/删除自己编写的服务到/从开机启动项，牵扯到顺位的概念，详见开机流程部分，P761。 18章末尾有系统常用服务。
18. ntsysv --level n：用图形化的界面设置等级n的开机启动项。

## 十二 日志系统

1. 日志系统可以交由系统进行统一管理，也可以有软件自己进行管理并设计日志文件的内容格式。
2. /var/log/messages主要记录了系统发生的错误信息和一些重要信息，/var/log/secure文件记录了账号认证的日志，/var/log/sendmail记录邮件来往信息，/var/log/maillog记录了邮件软件的运行日志，/var/log/httpd/\*存放http软件的日志，/var/log/wtmp记录了成功登录的信息，/var/log/lastlog记录了所有账号最后一次登录信息，/var/log/dmesg记录了开机过程中的硬件检测信息。
3. 系统提供的日志设施主要由syslogd来管理通用日志系统信息（登录、网络等），klogd管理内核产生的信息，logrotate主要进行对日志文件的维护，如备份并建新文件等。Logrotata在日志文件太大时会将其保存为旧日志文件，然后新建文件保存新的日志。
4. syslogd产生的日志格式可以在/etc/syslog.conf（centos中是/etc/rsyslog.conf）文件中进行设置。格式为：服务名称[.=!]信息等级 日志文件存放位置。
5. 同时设置多个服务可以在第一个字段服务名称内分别设置用分号隔开，和都是允许的。
6. 显示了syslog识别的主要服务。不同软件可以根据自己的需求设置交由syslog处理的日志属于什么类型。local7是开机时应显示在屏幕上的信息。
7. 信息等级按照紧急程度分为7等：info，notice，warning，err，crit，alert，emerg（\*代表所有）。还有两个特殊的debug，none（不需要记录）。
8. 服务名称后面的3个符号意义：.代表只要是比后面还要高的等级的信息都记录下来，.=代表仅记录后面等级的日志，.!代表不记录后面等级的信息。
9. 日志文件的存放位置可以是绝对路径（常用/var/log），也可以是设备，还可以是远程主机，\*代表目前在线的所有人（类似于用wall指令将信息发送给所有登录用户）。
10. 综上日志的流程是syslog将日志分类，每个软件分别设置了自己产生日志所属的类，通过修改/etc/rsyslog.conf设置不同类别的日志的存储位置。
11. syslog管理的任何日志文件被编辑过（如在vim中用wq离开）就无法继续记录直至重新启动syslog服务。（待验证）P688讲述了日志服务器的设置方法。
12. logrotate是一个cron管理的脚本，在/etc/cron.daily/目录下。
13. logrotate的原理是每个固定时间将log改名为log.1，log.1改为log.2等，并新建log文件记录新日志，超过设置数量的旧日志将被删除。
14. logrotate的配置主要在/etc/logrotate.conf文件和/etc/logrotate.d/目录下。前者将后者中的所有内容包含。前者设置了所有日志文件的默认配置，每个日志文件可以对其进行覆盖，后者主要是为了降低前者的复杂程度设置的。设置的格式为：日志路径{参数}。能设置的参数包括备份周期，备份数量，最小备份大小，是否压缩和新建日志所属用户等。
15. 承14点，可以在logrotate配置文件中添加脚本文件格式为在大括号内添加：sharedscripts prerotate|postrotate cmds endscript。prerotate和postrotate分别表示在日志rotate之前或之后执行cmds。
16. logrotate file：执行file文件中的设定，-v选项显示过程输出，-f强制执行rotate而忽略file中的设置。file为14条中的配置文件。
17. logwatch软件每天分析一次系统中的日志文件，并将分析结果通过邮件发送给root。

## 十三 开机流程

1. 表示了开机流程。
2. 每个文件系统都有一个boot sector。
3. linux的驱动等模块放在/lib/modules/目录内，/和lib两个目录必须同在一个文件系统中，且开机过程中根目录是以只读的方式进行挂载的。
4. /boot/目录下重要文件：config-2.6.xx-xx.elx存储此版本内核编译时选择的功能与模块配置文件，grub/开机管理程序grup的相关数据，initrd-2.6.xx-xx.elx.img是虚拟文件系统镜像（详见P704），System.map-2.6.xx-xx.elx为核心功能所在内存地址的对应表，vmlinuz-2.6.xx-xx.elx就是内核文件。
5. 若是使用SATA硬盘，则需加载SATA驱动，这是由initrd解压缩为一个目录树，再由其中的linuxrc程序加载驱动程序，然后挂载真是的根目录并且移除initrd解压出的目录，交由/sbin/init程序开始执行。
6. 7个run level：0表示直接关机，1为单用户维护模式，2与3类似，但不带NFS功能，3为含完整网络功能的多用户纯文本模式，4位系统保留（设定后系统重启），5位X桌面，6重新启动。
7. /sbin/init的主要功能是在内核已经加载完驱动模块后，来设定系统的各种软件环境，它的参考依据是/etc/inittab文件。
8. inittab文件中每行的格式为冒号分隔开的4个字段，第一个字段工作说明（最多4字符），第二字段为run level，表示对那些level有效执行，第三字段为init的动作，第四字段为命令。CentOS 6中inittab文件中仅保存初始run level的设置（initdefault），其它设置位置见inittab文件中。
9. /sbin/mingetty为终端程序，/etc/X11/prefdm –nodaemon为启动X环境。
10. 系统会先/etc/rc.d/rc.sysinit将系统设置好，该程序主要依据/etc/sysconfig/目录当中的配置文件进行设置。对模块的设置脚本放置在/etc/sysconfig/modules/目录下，其中文件以.modules结尾。
11. /etc/rc.d/rc N：执行/etc/rcN.d/目录下的脚本。rcN.d目录下大都是指向/etc/init.d/目录下脚本的链接文件。它们的的格式为Sxx\*\*/Kxx\*\*，xx为一个两位数，代表执行的顺序，S/K分别代表执行和关闭，\*\*为init.d目录下的脚本名。其中S99local指向/etc/rc.d/rc.local脚本，用户的自定义开机执行命令可以写入该脚本。
12. /etc/sysconfig/目录下的几个文件：authconfig主要是配置身份认证机制；clock用来设置本机的时区，时区参考文件在/usr/share/timeinfo/目录下的相对路径中，修改时区时还需将该目录下的文件复制为/etc/localtime；i18n为语系设置；network设置是否启动网络以及主机名和网关；network-scripts/下脚本设置网络。
13. run level的切换使用init n指令，仅参照两个level的启动脚本，先关闭多的K开头脚本服务，再打开多的S开头的脚本服务。所以两个level中都开启的服务则不会重启。
14. 内核与内核模块P713
15. /urs/src/linux/为源码安装处。
16. /lib/modules/`uname –r`/kernel/目录下存储核心模块，不同模块被组织到不同目录中去。模块文件通常为ko结尾。/lib/modules/`uname –r`/modules.dep文件记录了系统中所有的模块（但不一定都已加载）以及它们之间的依赖关系，depmod命令可以更新该文件（本质是扫描kernel目录下搜索所有模块并根据modules.dep文件中原内容进行更新）。
17. modprobe module\_name：modprobe依据modules.dep文件中的记录将模块module\_name加载，-r选项为卸载模块，-f为强制加载。
18. 总上，要添加一个新模块（可能是一个驱动），先将该模块复制到/lib/modules/`uname –r`/kernel/下的适当目录下，然后执行depmod更新modules.dep文件，再使用modprobe来加载该模块。所以系统中存在的模块并不一定都加载到了内存，这需要modprobe来管理。
19. lsmod列出系统已加载的模块，modinfo module显示module的信息。
20. insmod module\_path [parms]：将module\_path加载到内存，和modprobe功能相同，但是需要输入module的完整路径名，容易造成错误。rmmod module\_name是卸载模块，不需要完整路径名，-f选项强制卸载。
21. /etc/modprobe.d/目录下也有一些内核模块的额外设置，需查资料。P717
22. grub中的硬盘代号格式为：(hdn1,n2)，n1代表系统按顺序检测到的硬盘序号，n2代表该硬盘的第n2个磁盘分区。