-- Name：Kevin Liu
-- HomePage：www.mrzhiyi.cn
-- Tencent QQ：2261060260
-- Mobile Phone：18394178594、17715359685
-- Company：昆山杰普软件科技有限公司
-- E-mail：liuhl@briup.com、mrzhiyi@aliyun.com

-- 注：笔记中标题后带“”的表示阅读章节，只需要知道或者了解即可。*

1、关于 Unix/Linux

△ Unix/Linux 能够做什么?

服务器：Web 服务器,Mail 服务器,Database 服务器以及做程序开发等等。

△ 哪些人应该知道 Unix/Linux?

Unix/Linux 管理员,Oracle 管理员,网络工程师以及程序开发者等等。

△ Unix/Linux 的发展史?

更多关于 Unix 和 Linux 的发展史参见：http://dwz.cn/6qU0LQ

2、硬件环境

输入设备(Input Devices)，输出设备(Output Devices)，随机存取存储器(RAM)，中央处理器(CPU)，I/O 设备，硬盘(Hard Disk)等等。

输入设备包括键盘、鼠标、手写笔等等。输出设备包括显示器、打印机、磁带、硬盘等等。

3、软件环境

软件分为操作系统和应用程序。从用户和为用户提供服务的角度来说分为客户端和服务器。

4、Unix/Linux 操作系统

△ Kernel(内核)

○ 管理设备、内存、程序等等。

○ 控制系统程序/工具和硬件之间的功能。

○ 管理交换空间、守护进程、文件系统以及其他功能。

△ Shell

○ Shell 是用户和操作系统内核进行通讯的桥梁，扮演着解释器或翻译器的角色。

○ Shell 主要有一下几种，其中 bash 是 Unix/Linux 操作系统中默认的 Shell

□ Bourne Shell (sh)

◇ Bourne Shell 是 Stephen Bourne 为 AT&T Unix 编写的 Shell 工具。

◇ 最早的比较出名是 Unix Shell

◇ 一个新的版本是 Bourne-Again Shell

□ Korn Shell (ksh)

□ C Shell (csh)

◇ Sun 公司的创始人，Vi 编辑器的作者 Bill Joy 开发出来的，是 BSD Unix 的默认 Shell。

◇ C Shell 具有历史记录功能，所有使用过的命令都会保存下来，可以随时调用处理重新执行。

◇ C Shell 有自己的语法规则，类似于 C 语言。

◇ 在语法上与 Bourne Shell 不兼容。

◇ C Shell 有一个增强的并且完全兼容的新的 Shell，tcsh。

□ Bourne-Again Shell (bash)

△ FileSystem(文件系统)

○ 具有一定层次结构的目录，子目录和文件组合在一起所形成的系统。Linux 文件系统如下：

□ “/” 根目录，是 Linux/Unix 文件系统的最顶级的目录。

□ “/opt” 【Optional Application Software Packages】存放第三方工具或者是应用程序的目录。

□ “/etc” 【Etcetra Directory 】存放和系统相关的配置文件的目录。

□ “/dev” 【Devices】存放设备文件的目录。网卡例外，没有设备文件。

□ “/usr” 【Unix System Resource】Unix 系统资源，存放系统中所有用户都会使用的命令，文档以及开发库。

□ “/kernel” 在 Linux 中是“/boot”，存放内核程序。

□ “/var” 【Variable】存放的是管理员所使用的一些与系统运行过程相关特殊功能。比如操作系统的日志等等。

□ “/home” 系统默认的用户的主目录,root 用户除外。

○ 具体的 Ubuntu 操作系统的目录结构参见：http://dwz.cn/6pTvma

△ Command Line Syntax(命令行语法)

Command [option[s]] [argument[s]]

○ 命令行中的每一条命令的命令、选项和参数之间以空格作为分隔符。

○ 命令行中的每一条命令最长不能超过 265 个字符。

○ 大小写敏感

5、基本命令的使用

△ 登入/登出命令——login/logout

○ 本地登入/登出，在系统启动后，输入用户名和密码进行登入，使用 logout 登出。

○ 远程登入

□ SSH 远程登录

◇ SSH 为 Secure Shell 的缩写，由 IETF 的网络小组（Network Working Group）所制定，SSH 为建立在应用层基础上的安全协议。

◇ SSH 是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议

可以有效防止远程管理过程中的信息泄露问题。

◇ SSH 最初是 UNIX 系统上的一个程序，后来又迅速扩展到其他操作平台。SSH 在正确使用时可弥补网络中的漏洞。SSH 客户端适用于多种平台。

◇ 几乎所有 UNIX 平台—包括 HP-UX、Linux、AIX、Solaris、Digital UNIX、Irix，以及其他平台，都可运行 SSH。

◇ 登录方式：ssh 用户名@IP，回车之后需要输入密码。

□ Telnet 远程登录

◇ Telnet 协议是 TCP/IP 协议族中的一员，是 Internet 远程登陆服务的标准协议和主要方式。

◇ 为用户提供了在本地计算机上完成远程主机工作的能力。在终端使用者的电脑上使用 telnet 程序，用它连接到服务器。

◇ 终端使用者可以在 telnet 程序中输入命令，这些命令会在服务器上运行，就像直接在服务器的控制台上输入一样。可以在本地就能控制服务器。

◇ 要开始一个 telnet 会话，必须输入用户名和密码来登录服务器。

◇ 登录方式：telnet IP，回车之后要求输入用户名，再回车要求出入密码。

□ 几乎所有的 Linux 系统都具备 SSH 远程登录方式，也不推荐使用 Telnet 远程登录方式，因为 Telnet 在数据传输过程中是明文的，也就是不加密的，不安全。

○ 远程登出

□ exit

◇ 如果在当前的 Shell 中只有一个用户登录，则该命令会退出控制台。

◇ 当在同一个 Shell 中有多个用户登录，该命令会逐层退出用户，最后一个用户会退出 Shell。

◇ 该命令不会正常注销用户，只是退出将当前用户退出 Shell。

□ logout

◇ 该命令会正常注销当前用户并且将当前用户退出 Shell。

◇ 在 Shell 中登录，切换 Shell 之后，logout 会失效，即使切换回之前的 Shell，也是失效的。

○ 和登入/登出相关的用户信息

在 Linux 中和用户相关的文件主要是以下三个：

□ “/etc/passwd”,存放用户信息的文件，其中的每一行给出了用户名、密码和用户的其他信息。

◇ 该目录存储的是操作系统用户信息，该文件为所有用户可见。

◇ 打开/etc/passwd，看到其中一行的信息如下“kevin:x:1000:1000:Kevin:/home/kevin:/bin/bash”

◇ 可以看出/etc/passwd 文件存放的是用户的信息,由 6 个分号组成的 7 个信息,解释如下

(1)：用户名。

(2)：密码(已经加密)

(3)：UID(用户标识)，操作系统自己用的

(4)：GID 组标识。

(5)：用户全名或本地帐号

(6)：开始目录

(7)：登录使用的 Shell,就是对登录命令进行解析的工具。

□ “/etc/shadow”，存放用户密码信息的文件，是 passwd 的影子文件。

◇ 在 linux 中，密码存储在/etc/passwd 文件中，早期的这个文件直接存放加密后的密码，前两位是"盐（在密码学中，是指通过在密码任意固定位置插入特定的字符串，让散列后的结果和使用原始密码的散列结果不相符，这种过程称之为“加盐”。）"值，是一个随机数，后面跟的是加密的密码。为了安全，现在的 linux 都提供了/etc/shadow 这个影子文件，密码放在这个文件里面，并且是只有 root 可读的。

◇ 在利用了 shadow 文件的情况下，密码用一个 x 表示，普通用户看不到任何密码信息。影子口令文件保存加密的口令; /etc/passwd 文件中的密码全部变成 x。Shadow 只能是 root 可读，从而保证了安全。

◇ /etc/shadow 文件每一行的格式如下：用户名：加密口令：上一次修改的时间（从 1970 年 1 月 1 日起的天数）：口令在两次修改间的最小天数：口令修改之前向用户发出警告的天数：口令终止后账号被禁用的天数：从 1970 年 1 月 1 日起账号被禁用的天数：保留域。

◇ 例如“test:6hKjqUA40$OelB9h3UKOgnttKgmRpFr/:14316:0:99999:7:::”，共有 9 个栏目

(1)：帐号名称；

(2)：密码；这里是加密过的，也可以解密的，要注意安全问题。

(3)：上次修改密码的日期（距离 1970-01-01 的天数）；

(4)：密码不可被变更的天数；

(5)：密码需要被重新变更的天数(99999 表示不需要变更)；

(6)：密码变更前提前几天警告；

(7)：帐号失效日期。

(8)：帐号取消日期

(9)：保留条目,目前没用

□ “/etc/group”，存放用户组信息的文件，存储有关本地用户组的信息，每一行有一下四部分组成

◇ GroupName GID 到名称的一种映射，组名

◇ Password 保存密码的位置

◇ GID 组 ID

◇ Users 组成员

□ 用户的密码

◇ 对于 Unix 操作系统，用户密码必须在 6 到 8 个字符之间，对于 Linux 操作系统没有上限限制，但是不能少于 6 个字符，但是在使用管理员账户为其他用户设置密码时例外，任何字符数都可以。

◇ 密码要求是字母和其他字符的混合密码，但是至少要包含两个字母和一个数字或者是其他字符，管理设置密码例外，没有限制。

◇ 密码不能和用户名相同。

◇ 在更改用户密码时不能和上次的密码一致。

◇ 密码更改方式：使用 passwd 命令修改密码。

∵ 输入 passwd 用户名回车

∵ 输入当前正在使用的密码

∵ 输入新的密码

∵ 再次输入新的密码

∵ 提示信息，如果提示密码更新成功，则表示密码更改成功。

△ clear 清屏命令

△ pwd 显示当前路径的绝对路径名

△ cd 切换目录

□ 该目录可以是绝对路径，也可以是相对路径

□ cd 后直接回车，会进入到和当前用户相关的家目录

△ ls 列出目录下的目录和文件

□ ls -a 列出改目录下的所有目录和文件，其中包含一“.”开头的隐藏文件

□ ls -R 递归显示某个命令下的所有目录和文件

□ ls -t 将结果以时间顺序显示

□ ls -l 显示某目录下的目录和文件的详细信息

关于目录或者文件的详细信息的说明：例如“-rwxrw-r-- 1 kevin kevin 0 Feb 14 11:27 a.txt”

◇ 该信息由 7 部分组成

◇ 第一部分“-rwxrw-r--”，

∵ 第一个字符表示文件类型，文件类型共有一下几种：

∴ “d” 表示目录文件(Directory File)，第一个属性为 [d]，例如 [drwxrwxrwx]。

∴ “-” 表示常规文件(Regular File)。

∴ “c” 表示字符设备文件(Character Device File)，即串行端口的接口设备，例如键盘、鼠标等等。第一个属性为 [c]。

∴ “b” 表示块设备文件(Block Device File)，存储数据以供系统存取的接口设备，简单而言就是硬盘。第一个属性为 [b]。

∴ “s” 表示套接字文件(Sock File)，这类文件通常用在网络数据连接。启动一个程序来监听客户端的要求，客户端就可以通过套接字来进行数据通信。第一个属性为 [s]。

∴ “l” 表示链接文件(Link File)，类似 Windows 下面的快捷方式。第一个属性为 [l]，例

如 [lrwxrwxrwx]。

∴ “p” 表示管道文件(Pipe File)，是一种特殊的文件类型，它主要的目的是，解决多个程序同时存取一个文件所造成的错误。FIFO 是 first-in-first-out（先进先出）的缩写。第一个属性为 [p]。

∴ 更多关于文件类型和扩展名参见：http://dwz.cn/6rfDQk

∵ 第二、三、四个字符表示拥有者的读、写、执行权限。

∵ 第五、六、七个字符表示同组人的读、写、执行权限。

∵ 第八、九、十个字符表示其他人的读、写、执行权限。

∴ “r” 表示“读”权限

∴ “w” 表示“写”权限，对于目录来说表示可生成子目录或者文件的权限

∴ “x” 表示“执行”权限，对于目录来说表示可以查找该目录下的子目录或者文件的权限

∴ “-” 表示没有权限

◇ 第二部分“1”

∵ 如果该文件是目录文件，则该数字表示该目录下的子目录数

∵ 如果该文件是普通文件，则该数字表示该文件的硬链接(别名)

◇ 第三部分“kevin” 表示该文件的拥有者

◇ 第四部分“kevin” 表示该文件的所属组

◇ 第五部分“0” 表示该文件的所占磁盘空间的大小

◇ 第六部分“Feb 14 11:27” 表示该文件的最后一次修改时间

◇ 第七部分“a.txt” 表示该文件的文件名

△ chmod 修改文件的权限

□ 命令格式：chmod mode filename

◇ 其中“mode”由三部分组成分别是：who、op、permission(s)

∵ “who”表示用户或者组，又分为 u、g、o、a

∴ “u” 表示文件所有者

∴ “g” 表示文件所属组

∴ “o” 表示文件除当前拥有者和所属组之外的其他用户以及组

∴ “a” 表示上面所有的情况，即“a”表示文件的拥有者、所属组以及除当前拥有者和所属组之外的其他用户以及组

∵ “op” 表示修改用户对于文件权限的运算符，分为=、-、+

∴ “=” 表示设置权限，覆盖原来的权限

∴ “+” 表示当原来没有某个权限时，赋予改权限

∴ “-” 表示移除某个已拥有的权限

∵ “permission(s)” 表示用户或组对于文件的权限，分为 r、w、x

◇ 例如：chmod g-r file1

chmod u+x,go+w file2

chmod a=rw file3

※ 练习：创建三个文件分别为 a.txt、b.txt、c.txt，对于这三个文件修改权限为用户本人具有所有权限，同组人具有读写权限，，其他人没有任何权限。

□ 八进制数来表示权限，命令格式：chmod 八进制权限 filename

◇ 八进制表示权限的解释：r=4，w=2，x=1，由此八进制 0 到 7 的权限如下：

∵ 7：rwx；6：rw-；5：r-x；4：r--；3：–wx；2：-w-；1：--x；0：---

◇ 其中“777”也是由三部分组成

∵ 第一个 7 表示文件拥有者的读、写、执行权限

∵ 第二个 7 表示文件同组人的读、写、执行权限

∵ 第三个 7 表示文件其他人的读、写、执行权限

◇ 例如：chmod 777 file1

chmod 655 file2

△ touch 新建文件命令，当文件不存在是则创建文件，当文件已经存在时，则更新该文件的最后修改时间为当前时间。

△ mkdir 创建目录命令，命令格式：mkdir [-p] directory_name(s)

□ 一次可以创建多个目录

□ 也可以一次创建一个多级目录

◇ 当上级目录存在时，则直接创建相应的目录

◇ 当上级目录不存在时，则需要加“-p”选项

△ cp 拷贝命令，既能拷贝文件也能拷贝目录

□ cp file1 file2

◇ 拷贝文件 file1 到 file1 所在的目录，并重命名为 file2

□ cp file1 dir1

◇ 拷贝文件 file1 到目录 dir1 中

□ cp -r file1 file2 dir1 dir2

◇ 当 cp 命令后面的参数大于两个的时候，最后一个参数必须是“目录”，表示将所有参数(最后一个目录除外)所表示的文件(或目录)拷贝到最后一个参数所表示的目录中

◇ 拷贝文件可以不加选项“-r”

◇ 拷贝目录必须加选项“-r”

△ mv 移动命令，既能移动文件，也能移动目录，跟 cp 命令很相似，尤其是 cp 命令的第三条，但是不用加“-r”选项。

△ rm 删除文件命令，当删除目录时需要加“-r”选项，在删除目录或文件时原则上是需要有对于该文件或目录的写权限，但是在没有写权限的情况下也是可以删除的，只不过会出现提示信息，但是前提是该文件的所有者是当前用户。

△ rmdir 删除空目录，不需要加“-r”选项

△ 压缩和解压缩

□ 常用的压缩格式有两种：gzip 和 bzip，在 bzip 的基础上出现了新的压缩格式 bzip2

□ gzip 压缩时间短，但是压缩比低。压缩用 gzip 命令，

□ bzip2 压缩时间长，但是压缩比高

△ 归档和解档

□ 归档和解档都用的是 tar 命令，只不过跟的选项不一样。

□ 归档：tar [zcvf] package_name file(s)

□ 解档：tar [zxvf] package_name

◇ 当解档时需要指定解档路径时使用“-C”选项，即 tar zxvf package_name -C destination_dir

△ ln 创建链接文件命令

□ 链接又分为硬链接(Hard Link)和软链接(Soft Link)

◇ 硬链接

∵ 硬链接相当于给文件起别名，当改硬链接文件修改时，对应的源文件内容也会随之更改。

∵ 具体硬链接参见：http://dwz.cn/6pTr6j

∵ 建立硬链接可以防止误删

∵ 硬链接不能跨分区

∵ 一般对于目录不能建立硬链接

◇ 软链接

∵ 软链接相当于 Windows 文件系统中快捷方式。

∵ 软链接也叫符号链接(Symbolic)

∵ 具体参见：http://dwz.cn/6pTrBZ

□ 命令格式：ln [-s] source_file destination_file

◇ 说明：加选项“-s”为创建软链接，不加为创建硬链接

□ 其他参数/选项说明：

◇ 必要参数：

● -b 删除，覆盖以前建立的链接

● -d 允许超级用户制作目录的硬链接

● -f 强制执行

- -i 交互模式，文件存在则提示用户是否覆盖
- -n 把符号链接视为一般目录
- -s 软链接(符号链接)
- -v 显示详细的处理过程

◇ 选择参数：

- -S "-S<字尾备份字符串> "或 "--suffix=<字尾备份字符串>"
- -V "-V<备份方式>"或"--version-control=<备份方式>"
- --help 显示帮助信息
- --version 显示版本信息

△ man 查看其它命令的帮助文档 http://dwz.cn/6pTvma

6、元字符

△ 元字符（Meta Character）是指键盘上可输入的对于 Shell 来说具有其他特殊含义的字符被称为元字符，不同的 Shell 元字符不一定相同。

△ “Asterisk (*)” 星号，表示 0 到若干个任意字符

△ “Question Mask (?)” 问号，表示任意的单个字符

△ “Square Brackets ([])” 中括号，限制范围，表示在该元字符内的所有元素的任意一个，例如：[b-f]表示 b 到 f(包括 b、f)中的任意一个字符

△ “Semicolon (;)” 分号，当多条命令在 Shell 中的一行时，用来分割每一条命令

△ “Piping (|)” 管道，将两个命令分割开，会将第一个命令的结果作为第二个命令的输入传递给第二个命令

△ “Angle Brackets (< > >>)” 重定向符号

7、文本操作

△ 文本文件

□ 文本文件是一种由若干行字符构成的计算机文件。文本文件存在于计算机文件系统中。通常，通过在文本文件最后一行后放置文件结束标志来指明文件的结束。

□ 文本文件通常有系统配置文件，网页，程序源代码等等。

△ 文本文件的浏览

□ cat 命令

◇ 命令格式：cat filename(s)

◇ 命令说明：将文件的内容显示到屏幕上

◇ 将文件内容进行重定向到其他文件或者设备中。例如：cat /etc/passwd > a.txt

◇ 扩展

∵ 命令格式：cat /etc/passwd | awk -F: '{print $1 "\t" $6}' | sort > ~/userinfo

∵ 命令说明：将 passwd 的内容按照“:”进行分割，然后提取第一项和第六项交给 sort 命令，sort 命令进行排序后重定向输出到 userinfo 文件中。

∵ 空格和元字符作为分隔符时，在使用上述命令时，需要将空格或者元字符用单引号括起来。

□ more 命令

◇ 命令格式：more filename(s)

◇ 命令说明：显示文件内容到屏幕上，但是一屏内容满了之后会暂停下来。此时可以通过其他功能键进行查看。

□ head 命令

◇ 命令格式：head [-n] filename(s)

◇ 命令说明：显示某个文件的前 n 行，如果没有选项-n(n 是一个具体的数值)，则默认显示前 10 行。

□ tail 命令 具有和 head 命令相类似的功能，不同的是从显示某个文件尾部的行。

※ 练习：显示/etc/passwd 文件的倒数第五行到倒数第十行的内容。tail -10 /etc/passwd | head -5

□ sort 命令

◇ 读取文件中的内容，按行排序。

◇ 读取管道中的内容，按行排序。

◇ 可以加选项“-u”，表示重复行(相同并且相邻则视为重复行)只显示一次。

※ 练习：提取系统中所有的用户名，按用户名(行)排序。

答案：cat /etc/passwd | awk -F: '{print $1}' | sort -u

□ uniq 命令

◇ 清除文件或者管道中的重复行

□ diff 命令

◇ 比较并显示两个文件的不同之处

◇ 命令格式：diff [-u] filename1 filename2

□ file 命令

◇ 该命令用来查看文件类型

□ echo 命令

◇ 显示命令后面的内容到屏幕上

□ script 命令

◇ 录屏命令，该命令会执行之后会将控制台上所有从该命令之后的输入和输入全部显示到指定的文件中。

◇ 可以加选项“-a”，表示追加到文件中。

◇ 录屏结束之后使用“exit”命令退出录屏。

8、全屏幕文本编辑器

△ Vi 编辑器（全屏幕文本编辑器，Full Screen Text Editor）

□ Vi 的工作模式

◇ 编辑模式

Vi 在初始启动后首先进入编辑模式，这时用户可以利用一些预先定义的按键来移动光标、删除文字、复制或粘贴文字等。这些按键均是普通的字符，例如 l 是向右移动光标，相当于向右箭头键，k 是向下移动光标，相当于向下箭头键。在编辑模式下，用户还可以利用一些特殊按键选定文字，然后再进行删除、或复制等操作。

◇ 插入模式

当用户在编辑模式下键入 i/a/o 等命令之后，可进入插入模式。在该模式下，用户随后输入的，除 Esc 之外的任何字符均将被看成是插入到编辑缓冲区中的字符。按 Esc 之后，从插入模式切换到编辑模式。

◇ 命令模式

在插入模式下，键入:可进入命令模式。在命令模式，Vi 将把光标挪到屏幕的最下方，并在第一个字符的位置显示一个 “:” (冒号)。这时，用户就可以键入一些命令。这些命令可用来保存文件、读取文件内容、执行 Shell 命令、设置 Vi 参数、以正则表达式的方式查找字符串或替换字符串等。

□ Vi 编辑器的简单使用

◇ Vi 的启动　输入 vi 命令后，便进入全屏幕编辑环境，此时的状态为命令模式。

∵ vi 进入 vi 的一个临时缓冲区，光标定位在该缓冲区第 1 行第 1 列的位置上。

∵ vi file1 如果 file1 文件不存在，将建立此文件；如该文件存在，则将其拷贝到一个临时缓冲区。光标定位在该缓冲区第 1 行第 1 列的位置上。

∵ vi + file1 如果 file1 文件不存在，将建立此文件；如该文件存在，则将其拷贝到一个临时缓冲区。光标定位在文件最后 1 行第 1 列的位置上。

∵ vi +N file1(N：为数字) 如果 file1 文件不存在，将建立此文件；如该文件存在，则将其拷贝到一个临时缓冲区。光标定位在文件第 N 行第 1 列的位置上。

∵ vi +/string file1 如果 file1 文件不存在将建立此文件；如该文件存在则将其拷贝到一个临时缓冲区。光标定位在文件中第一次出现字符串 string 的行首位置。

◇ 退出 Vi　建议在退出 vi 前，先按 ESC 键，以确保当前 vi 的状态为命令方式，然后再键入“：”(冒号)，输入下列命令，退出 vi。

∵ :w 将编辑缓冲区的内容写入文件，则新的内容就替代了原始文件。这时并没有退出 vi，必须进一步输入下述命令才能退出 vi：:w filename(存入指定文件) :q

∵ :wq 即将上面的两步操作可以合成一步来完成，先执行 w，后执行 q。

∵ :x 和 ZZ（注意：ZZ 前面没有“:”）功能与“:wq”等价。

∵ :q!(或:quit) 强行退出 vi，使被更新的内容不写回文件中。仅键入命令:q 时，如 vi 发现文本内容已被更改，将提示用户使用“:quit”命令退出。

◇ 另外请参见：http://dwz.cn/6pTuTA

□ Vi 编辑器的常用命令

◇ 在文件中移动光标

- h：向左移动一个字符
- l：向右移动一个字符
- j：向下移动一行
- k：向上移动一行
- ^（即 Shift+6）：移动到当前行的开头处
- $（即 Shift+4）：移动到当前行的结尾处
- Ctrl+b：上滚一屏。

- Ctrl+f：下滚一屏。
- Ctrl+d：下滚半屏。
- Ctrl+u: 上滚半屏。

◇ 插入文本

∵ 添加：

- 输入 a 后，在光标的右边插入文本
- 输入 A，在一行的结尾处添加文本

∵ 插入：

- 通过在命令模式下输入 i，在光标的左边插入文本
- 通过在命令模式下输入 I，在行首插入文本

∵ 插入新行：

- 输入 o，在当前光标位置下面打开一行
- 输入 O，在当前光标位置上面打开一行

◇ 撤消更改

∵ 撤消前一个命令：在最后一个命令之后立即输入 u 来撤消该命令

∵ 重复某个命令：“.”

∵ 撤消对一行的更改：输入 U 来撤消你对一行所做的所有更改，这个命令只有在你没将光标移动到该行以外时才生效。

◇ 删除文本

∵ 删除一个字符：

- 为删除一个字符，需将光标放置在要删除的字符上并输入 x
- 为删除光标之前（其左边）的一个字符，需输入 X

∵ 删除一个词或词的部分内容：

- 为删除一个词，需将光标放置到该词的开头并输入 dw
- 为删除词的部分内容，将光标放到该词要保存部分的右边。输入 dw 来删除余下的部分

∵ 删除 1 行：将光标放置到该行的任意处并输入 dd；删除多行：ndd

∵ 删除行的部分内容：将光标放置到该行要保存部分的右边，并输入 D。为删除光标左边的所有内容，须将光标放置到该行要删除部分的右边，并输入 d0（d-零）。

∵ 删除到文件的结尾：为删除从当前行到文件结尾的所有内容，需输入 dG

◇ 复制和移动文本

- 复制一行命令：yy
- 粘贴命令：p
- 移动文本：先将要移动的部分用删除命令删除，然后再粘贴就可以了
- 复制指定文件的内容（使用底行模式）：:r filename

◇ 查找一个字符串

- 输入/，并在/后面输入要查找的串，然后按下回车
- 输入“n”跳转到该串的下一个出现处
- 输入“N”跳转到该串的上一个出现处

◇ 替换一个字符串

- 在一行内替换头一个字符串 old 为新的字符串 new：:s/old/new
- 在一行内替换所有的字符串 old 为新的字符串 new：:s/old/new/g
- 在两行内替换所有的字符串 old 为新的字符串 new：:#,#s/old/new/g

- 在文件内替换所有的字符串 old 为新的字符串 new：:%s/old/new/g
- 进行全文替换时询问用户确认每个替换需添加 c 选项：:%s/old/new/gc（需按两次回车）

◇ 设置 vi

- 显示行号:set number
- 取消行号显示:set nonumber
- 设置显示用户模式:set showmode
- 设置文件只读:set readonly

◇ 更多关于 vim 编辑器的内容请参见：http://www.dwz.cn/6pTuTA

9、在 Linux 系统中查找文件

△ 自动搜索

□ 通过“echo $PATH”命令可以查看自动搜索路径有哪些。

□ 一般自动搜索都是针对于系统命令而言的。

△ 手动搜索

□ find 命令

◇ find 命令的一般形式为：find pathname -options [-print -exec -ok ...]

◇ find 命令的参数；

● pathname: find 命令所查找的目录路径。例如用.来表示当前目录，用/来表示系统根目录。

● -print： find 命令将匹配的文件输出到标准输出。

● -exec： find 命令对匹配的文件执行该参数所给出的 shell 命令。相应命令的形式为'command' {} \;，注意{ }和；之间的空格。

● -ok：和-exec 的作用相同，只不过以一种更为安全的模式来执行该参数所给出的 shell 命令，在执行每一个命令之前，都会给出提示，让用户来确定是否执行。

◇ find 命令选项

● -name 按照文件名查找文件。

● -perm 按照文件权限来查找文件。

● -prune 使用这一选项可以使 find 命令不在当前指定的目录中查找，如果同时使用-depth 选项，那么-prune 将被 find 命令忽略。

- -user 按照文件属主来查找文件。
- -group 按照文件所属的组来查找文件。
- -mtime -n +n 按照文件的更改时间来查找文件，-n 表示文件更改时间距现在 n 天以内，+n 表示文件更改时间距现在 n 天以前。find 命令还有-atime 和-ctime 选项，但它们都和-m time 选项。
- -nogroup 查找无有效所属组的文件，即该文件所属的组在/etc/groups 中不存在。
- -nouser 查找无有效属主的文件，即该文件的属主在/etc/passwd 中不存在。
- -newer file1 ！file2 查找更改时间比文件 file1 新但比文件 file2 旧的文件。
- -type 查找某一类型的文件，诸如：

  b - 块设备文件。

  d - 目录。

  c - 字符设备文件。

  p - 管道文件。

  l - 符号链接文件。

  f - 普通文件。

- -size n：[c] 查找文件长度为 n 块的文件，带有 c 时表示文件长度以字节计。-depth：在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找。
- -fstype：查找位于某一类型文件系统中的文件，这些文件系统类型通常可以在配置文件/etc/fstab 中找到，该配置文件中包含了本系统中有关文件系统的信息。
- -mount：在查找文件时不跨越文件系统 mount 点。
- -follow：如果 find 命令遇到符号链接文件，就跟踪至链接所指向的文件。
- -cpio：对匹配的文件使用 cpio 命令，将这些文件备份到磁带设备中。
- 另外,下面三个的区别:
  - ∵ -amin n 查找系统中最后 n 分钟访问的文件
  - ∵ -atime n 查找系统中最后 n*24 小时访问的文件
  - ∵ -cmin n 查找系统中最后 n 分钟被改变文件状态的文件
  - ∵ -ctime n 查找系统中最后 n*24 小时被改变文件状态的文件
  - ∵ -mmin n 查找系统中最后 n 分钟被改变文件数据的文件
  - ∵ -mtime n 查找系统中最后 n*24 小时被改变文件数据的文件

◇ 使用 exec 或 ok 来执行 shell 命令

● 使用 find 命令时，只要把想要的操作写在一个文件里，就可以用 exec 来配合 find 命令查找，很方便的

● 在有些操作系统中只允许-exec 选项执行诸如 l s 或 ls -l 这样的命令。大多数用户使用这一选项是为了查找旧文件并删除它们。建议在真正执行 rm 命令删除文件之前，最好先用 ls 命令看一下，确认它们是所要删除的文件。

● exec 选项后面跟随着所要执行的命令或脚本，然后是一对儿{}，一个空格和一个“\”，最后是一个分号。为了使用 exec 选项，必须要同时使用 print 选项。如果验证一下 find 命令，会发现该命令只输出从当前路径起的相对路径及文件名。

※ 例如：用 ls -l 命令列出所匹配到的文件，可以把 ls -l 命令放在 find 命令的-exec 选项中

```
# find . -type f -exec ls -l { } \;
```

上面的例子中，find 命令匹配到了当前目录下的所有普通文件，并在-exec 选项中使用 ls -l 命令将它们列出。

※ 例如：在/logs 目录中查找更改时间在 5 日以前的文件并删除它们：

$ find logs -type f -mtime +5 -exec rm { } \;

记住：在 shell 中用任何方式删除文件之前，应当先查看相应的文件，一定要小心！当使用诸如 mv 或 rm 命令时，可以使用-exec 选项的安全模式。它将在对每个匹配到的文件进行操作之前提示你。

※ 例如：在下面的例子中， find 命令在当前目录中查找所有文件名以.LOG 结尾、更改时间在 5 日以上的文件，并删除它们，只不过在删除之前先给出提示。

$ find . -name "*.conf" -mtime +5 -ok rm { } \;

< rm ... ./conf/httpd.conf > ? n

按 y 键删除文件，按 n 键不删除。任何形式的命令都可以加-exec 选项中使用。

※ 例如：在下面的例子中我们使用 grep 命令。find 命令首先匹配所有文件名为“ passwd*”的文件，例如 passwd、passwd.old、passwd.bak，然后执行 grep 命令看看在这些文件中是否存在一个 sam 用户。

find /etc -name "passwd*" -exec grep "sam" {} \;

sam:x:501:501::/usr/sam:/bin/bash

◇ find 命令的例子；

- 查找当前用户主目录下的所有文件：

  $ find $HOME -print

  $ find ~ -print

- 让当前目录中文件属主具有读、写权限，且文件所属组的用户和其他用户具有读权限的文件；

  $ find . -type f -perm 644 -exec ls -l {} \;

- 为了查找系统中所有文件长度为 0 的普通文件，并列出它们的完整路径；

  $ find / -type f -size 0 -exec ls -l {} \;

- 查找/var/logs 目录中更改时间在 7 日以前的普通文件，并在删除之前询问它们；

  $ find /var/logs -type f -mtime +7 -ok rm {} \;

- 为了查找系统中所有属于 root 组的文件；

  $ find . -group root -exec ls -l {} \;

- find 命令将删除当目录中访问时间在 7 日以来、含有数字后缀的 admin.log 文件。

  该命令只检查三位数字，所以相应文件的后缀不要超过 999。先建几个 admin.log*的文件 ，才能使用下面这个命令

$ find . -name "admin.log[0-9][0-9][0-9]" -atime -7 -ok rm {} \;

< rm ... ./admin.log001 > ? n

< rm ... ./admin.log002 > ? n

< rm ... ./admin.log042 > ? n

< rm ... ./admin.log942 > ? n

- 为了查找当前文件系统中的所有目录并排序；

  $ find . -type d | sort

- 为了查找系统中所有的 rmt 磁带设备；

  $ find /dev/rmt -print

- xargs（*）

  ○ xargs - build and execute command lines from standard input

  ○ 在使用 find 命令的-exec 选项处理匹配到的文件时， find 命令将所有匹配到的文件一起传递给 exec 执行。但有些系统对能够传递给 exec 的命令长度有限制，这样在 find 命令运行几分钟之后，就会出现溢出错误。错误信息通常是“参数列太长”或“参数列溢出”。这就是 xargs 命令的用处所在，特别是与

find 命令一起使用。

○ find 命令把匹配到的文件传递给 xargs 命令，而 xargs 命令每次只获取一部分文件而不是全部，不像-exec 选项那样。这样它可以先处理最先获取的一部分文件，然后是下一批，并如此继续下去。

○ 在有些系统中，使用-exec 选项会为处理每一个匹配到的文件而发起一个相应的进程，并非将匹配到的文件全部作为参数一次执行；这样在有些情况下就会出现进程过多，系统性能下降的问题，因而效率不高；

○ 而使用 xargs 命令则只有一个进程。另外，在使用 xargs 命令时，究竟是一次获取所有的参数，还是分批取得参数，以及每一次获取参数的数目都会根据该命令的选项及系统内核中相应的可调参数来确定。

○ 来看看 xargs 命令是如何同 find 命令一起使用的，并给出一些例子。

○ 下面的例子查找系统中的每一个普通文件，然后使用 xargs 命令来测试它们属于哪类文件

```
# find . -type f -print | xargs file
./.kde/Autostart/Autorun.desktop: UTF-8 Unicode English text
./.kde/Autostart/.directory: ISO-8859 text
```

.....

○ 在系统中查找内存信息转储文件(core dump)，然后把结果保存到/tmp/core.log 文件中：

$ find / -name "core" -print | xargs echo "" >/tmp/core.log

○ 上面这个执行太慢，改成在当前目录下查找

find . -name "file*" -print | xargs echo "" > /temp/core.log

cat /temp/core.log

○ 在当前目录下查找所有用户具有读、写和执行权限的文件，并收回相应的写权限：

find . -perm -7 -print | xargs chmod o-w

○ 用 grep 命令在所有的普通文件中搜索 hostname 这个词：

find . -type f -print | xargs grep "hostname"

○ 用 grep 命令在当前目录下的所有普通文件中搜索 hostnames 这个词：

find . -name * -type f -print | xargs grep "hostnames"

○ 注意，在上面的例子中，用来取消 find 命令中的*在 shell 中的特殊含义。

○ find 命令配合使用 exec 和 xargs 可以使用户对所匹配到的文件执行几乎所有的命令。

◇ find 命令的参数

● 使用 name 选项

∵ 文件名选项是 find 命令最常用的选项，要么单独使用该选项，要么和其他选项一起使用。

∵ 可以使用某种文件名模式来匹配文件，记住要用引号将文件名模式引起来。

∵ 不管当前路径是什么，如果想要在自己的根目录$HOME 中查找文件名符合*.txt 的文件，使用~作为 'pathname'参数，波浪号~代表了你的$HOME 目录。

```
$ find ~ -name "*.txt" -print
```

∵ 想要在当前目录及子目录中查找所有的‘ *.txt’文件，可以用：

```
$ find . -name "*.txt" -print
```

∵ 想要的当前目录及子目录中查找文件名以一个大写字母开头的文件，可以用：

```
$ find . -name "[A-Z]*" -print
```

∵ 想要在/etc 目录中查找文件名以 host 开头的文件，可以用：

```
$ find /etc -name "host*" -print
```

∵ 想要查找$HOME 目录中的文件，可以用：

$ find ~ -name "*" -print 或 find . -print

∵ 要想让系统高负荷运行，就从根目录开始查找所有的文件。

$ find / -name "*" -print

∵ 如果想在当前目录查找文件名以两个小写字母开头，跟着是两个数字，最后是.txt 的文件，下面的命令就能够返回名为 ax37.txt 的文件：

$find . -name "[a-z][a-z][0--9][0--9].txt" -print

- 用 perm 选项

∵ 按照文件权限模式用-perm 选项,按文件权限模式来查找文件的话。最好使用八进制的权限表示法。

∵ 如在当前目录下查找文件权限位为 755 的文件，即文件属主可以读、写、执行，其他用户可以读、执行的文件，可以用：

$ find . -perm 755 -print

∵ 还有一种表达方法：在八进制数字前面要加一个横杠-，表示都匹配，如-007 就相当于 777，-006 相当于 666

ls -l

find . -perm 006

find . -perm -006

∵ -perm mode:文件许可正好符合 mode

∵ -perm +mode:文件许可部分符合 mode

∵ -perm -mode: 文件许可完全符合 mode

- 忽略某个目录

∵ 如果在查找文件时希望忽略某个目录，因为你知道那个目录中没有你所要查找的文件，那么可以使用-prune 选项来指出需要忽略的目录。在使用-prune 选项时要当心，因为如果你同时使用了-depth 选项，那么-prune 选项就会被 find 命令忽略。

∵ 如果希望在/apps 目录下查找文件，但不希望在/apps/bin 目录下查找，可以用：

$ find /apps -path "/apps/bin" -prune -o -print

- 使用 find 查找文件的时候怎么避开某个文件目录

∵ 比如要在/usr/sam 目录下查找不在 dir1 子目录之内的所有文件

find /usr/sam -path "/usr/sam/dir1" -prune -o -print

find [-path ..] [expression] 在路径列表的后面的是表达式

-path "/usr/sam" -prune -o -print 是 -path "/usr/sam" -a -prune -o

-print 的简写表达式按顺序求值, -a 和 -o 都是短路求值，与 shell 的 && 和 || 类似如果 -path "/usr/sam" 为真，则求值 -prune ，-prune 返回真，与逻辑表达式为真；否则不求值 -prune，与逻辑表达式为假。如果 -path "/usr/sam" -a -prune 为假，则求值 -print ，-print 返回真，或逻辑表达式为真；否则不求值 -print，或逻辑表达式为真。

∵ 避开多个文件夹

find /usr/sam (-path /usr/sam/dir1 -o -path /usr/sam/file1) -prune -o -print

∵ 查找某一确定文件，-name 等选项加在-o 之后

find /usr/sam (-path /usr/sam/dir1 -o -path /usr/sam/file1) -prune -o -name "temp" -print

- 使用 user 和 nouser 选项

∵ 按文件属主查找文件，如在$HOME 目录中查找文件属主为 sam 的文件，可以用：

$ find ~ -user sam -print

∵ 在/etc 目录下查找文件属主为 uucp 的文件：

$ find /etc -user uucp -print

∵ 为了查找属主帐户已经被删除的文件，可以使用-nouser 选项。这样就能够找到那些属主在/etc/passwd 文件中没有有效帐户的文件。在使用-nouser 选项时，不必给出用户名； find 命令能够为你完成相应的工作。

例如，希望在/home 目录下查找所有的这类文件，可以用：

$ find /home -nouser -print

- 使用 group 和 nogroup 选项

∵ 就像 user 和 nouser 选项一样，针对文件所属于的用户组， find 命令也具有同样的选项，为了在/apps 目录下查找属于 gem 用户组的文件，可以用：

$ find /apps -group gem -print

∵ 要查找没有有效所属用户组的所有文件，可以使用 nogroup 选项。下面的 find 命令从文件系统的根目录处查找这样的文件

$ find / -nogroup-print

- 按照更改时间或访问时间等查找文件

∵ 如果希望按照更改时间来查找文件，可以使用 mtime,atime 或 ctime 选项。如果系统突然没有可用空间了，很有可能某一个文件的长度在此期间增长迅速，这时就可以用 mtime 选项来查找这样的文件。

∵ 用减号-来限定更改时间在距今 n 日以内的文件，而用加号+来限定更改时间在距今 n 日以前的文件。

∵ 希望在系统根目录下查找更改时间在 5 日以内的文件，可以用：

```
$ find / -mtime -5 -print
```

∵ 为了在/var/adm 目录下查找更改时间在 3 日以前的文件，可以用：

```
$ find /var/adm -mtime +3 -print
```

- 查找比某个文件新或旧的文件

∵ 如果希望查找更改时间比某个文件新但比另一个文件旧的所有文件，可以使用-newer 选项。它的一般形式为：newest_file_name ！ oldest_file_name

○ 其中，！是逻辑非符号。

○ 查找更改时间比文件 sam 新但比文件 temp 旧的文件：

```
# find -newer httpd1.conf ! -newer temp -ls
```

∵ 查找更改时间在比 temp 文件新的文件：

```
$ find . -newer temp -print
```

● 使用 type 选项

∵ 在/etc 目录下查找所有的目录，可以用：

```
$ find /etc -type d -print
```

∵ 在当前目录下查找除目录以外的所有类型的文件，可以用：

```
$ find . ! -type d -print
```

∵ 在/etc 目录下查找所有的符号链接文件，可以用

```
$ find /etc -type l -print
```

● 使用 size 选项

∵ 可以按照文件长度来查找文件，这里所指的文件长度既可以用块(block)来计量，也可以用字节来计量。以字节计量文件长度的表达形式为 N c；以块计量文件长度只用数字表示即可。

∵ 在按照文件长度查找文件时，一般使用这种以字节表示的文件长度，在查看文件系统的大小，因为这时使用块来计量更容易转换。

∵ 在当前目录下查找文件长度大于 1 M 字节的文件：

```
$ find . -size +1000000c -print
```

∵ 在/home/apache 目录下查找文件长度恰好为 100 字节的文件

```
$ find /home/apache -size 100c -print
```

∵ 在当前目录下查找长度超过 10 块的文件(一块等于 512 字节)

```
$ find . -size +10 -print
```

- 使用 depth 选项

∵ 在使用 find 命令时，可能希望先匹配所有的文件，再在子目录中查找。使用 depth 选项就可以使 find 命令这样做。这样做的一个原因就是，当在使用 find 命令向磁带上备份文件系统时，希望首先备份所有的文件，其次再备份子目录中的文件。

∵ 在下面的例子中，find 命令从文件系统的根目录开始，查找一个名为 CON.FILE 的文件。

∵ 它将首先匹配所有的文件然后再进入子目录中查找。

$ find / -name "CON.FILE" -depth -print

● 使用 mount 选项

∵ 在当前文件系统中查找文件（不进入其他文件系统），可以使用 find 命令的 mount 选项。

∵ 从当前目录开始查找位于本文件系统中文件名以 XC 结尾的文件：

$ find . -name "*.XC" -mount -print

□ grep 命令

◇ 命令作用

● Linux 系统中 grep 命令是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。

● grep 全称是 Global Regular Expression Print，表示全局正则表达式版本，它的使用权限是所有用户。

◇ 命令格式

● grep [options]

◇ 主要参数

- 主要选项

  －c：只输出匹配行的计数。

  －I：不区分大 小写(只适用于单字符)。

  －h：查询多文件时不显示文件名。

  －l：查询多文件时只输出包含匹配字符的文件名。

  －n：显示匹配行及 行号。

  －s：不显示不存在或无匹配文本的错误信息。

  －v：显示不包含匹配文本的所有行。

- pattern 正则表达式主要参数：

  \：　忽略正则表达式中特殊字符的原有含义。

  ^：匹配正则表达式的开始行。

  $: 匹配正则表达式的结束行。

  \<：从匹配正则表达 式的行开始。

  \>：到匹配正则表达式的行结束。

[ ]：单个字符，如[A]即 A 符合要求 。

[-]：范围，如[A-Z]，即 A、B、C 一直到 Z 都符合要求 。

。：所有的单个字符。

* ：有字符，长度可以为 0。

◇ grep 命令使用简单实例

- 显示所有以 d 开头的文件中包含 test 的行。

  $ grep ‘test’ d*

- 显示在 aa，bb，cc 文件中匹配 test 的行。

  $ grep ‘test’ aa bb cc

- 显示所有包含每个字符串至少有 5 个连续小写字符的字符串的行。

  $ grep ‘[a-z]\{5\}’ aa

□ wc 命令

◇ 统计文件里面有多少单词，多少行，多少字符。

◇ 命令格式：wc [option(s)] filename

◇ 可选选项

-l 统计行

-w 统计单词

-c 统计字符数

□ df 命令

◇ 命令格式：

df [选项] [文件]

◇ 命令功能：

显示指定磁盘文件的可用空间。如果没有文件名被指定，则所有当前被挂载的文件系统的可用空间将被显示。

◇ 命令参数：

● 必要参数：

-a 全部文件系统列表

-h 方便阅读方式显示

-H 等于“-h”，但是计算式，1K=1000，而不是 1K=1024

-i 显示 inode 信息

-k 区块为 1024 字节

-l 只显示本地文件系统

-m 区块为 1048576 字节

--no-sync 忽略 sync 命令

-P 输出格式为 POSIX

--sync 在取得磁盘信息前，先执行 sync 命令

-T 文件系统类型

- 选择参数：

--block-size=<区块大小> 指定区块大小

-t<文件系统类型> 只显示选定文件系统的磁盘信息

-x<文件系统类型> 不显示选定文件系统的磁盘信息

--help 显示帮助信息

--version 显示版本信息

□ fdisk 命令

◇ 命令说明：观察硬盘使用情形与硬盘分区用。

◇ 使用方法：

● 在 console 上输入 fdisk -l /dev/sda ，观察硬盘之实体使用情形。

● 在 console 上输入 fdisk /dev/sda，可进入分割硬盘模式。

① 输入 m 显示所有命令列示。

② 输入 p 显示硬盘分割情形。

③ 输入 a 设定硬盘启动区。

④ 输入 n 设定新的硬盘分割区。

○ 输入 p 硬盘为[主要]分割区(primary)。

○ 输入 e 硬盘为[延伸]分割区(extend)。

⑤ 输入 t 改变硬盘分割区属性。

⑥ 输入 d 删除硬盘分割区属性。

⑦ 输入 q 结束不存入硬盘分割区属性。

⑨ 输入 w 结束并写入硬盘分割区属性

● partprobe 更新当前分区表给内核，这一步非常重要，否则你的分区重启才能看到。

□ mkfs 命令

◇ 命令格式：mkfs [-V] [-t fstype] [fs-options] filesys [blocks]

◇ 参数说明

device ： 预备检查的硬盘分区，例如：/dev/sda1

-V ：详细显示模式

-t ：给定档案系统的型式，Linux 的预设值为 ext2

-c ：在制做档案系统前，检查该 partition 是否有坏轨

-l bad_blocks_file ：将有坏轨的 block 资料加到 bad_blocks_file 里面

block ：给定 block 的大小

※ 练习：在/dev/hda5 上建一个 msdos 的档案系统，同时检查是否有坏轨存在，并且将过程详细列出来:

mkfs -V -t msdos -c /dev/hda5

※ 练习：将 sda6 分区格式化为 ext3 格式

mfks -t ext3 /dev/sda6

※ 注意：这里的文件系统是要指定的，比如 ext3 ； reiserfs ； ext2 ； fat32 ； msdos 等。

□ mount 命令

◇ 命令说明：Linux mount 命令是经常会使用到的命令，它用于挂载 Linux 系统外的文件。

◇ 命令语法

mount [-hV]

mount -a [-fFnrsvw] [-t vfstype]

mount [-fnrsvw] [-o options [,...]] device | dir

mount [-fnrsvw] [-t vfstype] [-o options] device dir

◇ 参数说明

-V：显示程序版本

-h：显示辅助讯息

-v：显示较讯息，通常和 -f 用来除错。

-a：将 /etc/fstab 中定义的所有档案系统挂上。

-F：这个命令通常和 -a 一起使用，它会为每一个 mount 的动作产生一个行程负责执行。在系统需要挂上大量 NFS 档案系统时可以加快挂上的动作。

-f：通常用在除错的用途。它会使 mount 并不执行实际挂上的动作，而是模拟整个挂上的过程。通常会和 -v 一起使用。

-n：一般而言，mount 在挂上后会在 /etc/mtab 中写入一笔资料。但在系统中没有可写入档案系统存在的情况下可以用这个选项取消这个动作。

-s-r：等于 -o ro

-w：等于 -o rw

-L：将含有特定标签的硬盘分割挂上。

-U：将档案分割序号为 的档案系统挂下。-L 和 -U 必须在/proc/partition 这种档案存在时才有意义。

-t：指定档案系统的型态，通常不必指定。mount 会自动选择正确的型态。

-o async：打开非同步模式，所有的档案读写动作都会用非同步模式执行。

-o sync：在同步模式下执行。

-o atime、-o noatime：当 atime 打开时，系统会在每次读取档案时更新档案的『上一次调用时间』。当我们使用 flash 档案系统时可能会选项把这个选项关闭以减少写入的次数。

-o auto、-o noauto：打开/关闭自动挂上模式。

-o defaults:使用预设的选项 rw, suid, dev, exec, auto, nouser, and async.

-o dev、-o nodev-o exec、-o noexec 允许执行档被执行。

-o suid、-o nosuid：允许执行档在 root 权限下执行。

-o user、-o nouser：使用者可以执行 mount/umount 的动作。

-o remount：将一个已经挂下的档案系统重新用不同的方式挂上。例如原先是唯读的系统，现在用可读写的模式重新挂上。

-o ro：用唯读模式挂上。

-o rw：用可读写模式挂上。

-o loop=：使用 loop 模式用来将一个档案当成硬盘分割挂上系统。

※ 练习 1：将/dev/hda1 挂在/mnt 之下。

mount /dev/hda1 /mnt

※ 练习 2：将/dev/hda1 用唯读模式挂在/mnt 之下。

#mount -o ro /dev/hda1 /mnt

※ 练习 3：将/tmp/image.iso 这个光碟的 image 档使用 loop 模式挂在/mnt/cdrom 之下。用这种方法可以将一般网络上可以找到的 Linux 光碟 ISO 档在不烧录成光碟的情况下检视其内容。

#mount -o loop /tmp/image.iso /mnt/cdrom

□ unmount 命令

◇ 命令格式：unmount 挂载点

◇ 命令说明：对某个挂载点进行卸载

□ 永久挂载磁盘

◇ 说明：通过上述的 mount 命令只是临时的挂载，在及其重启之后需要再次进行挂载，很不方便，于是就需要进行永久挂载或者是开机自动挂载。

◇ 方法：

- 首先得到到/dev/vdb1 这个分区的 UUID,使用以下命令：sudo blkid /dev/vdb1
- 用 vi 编辑器打开/etc/fstab 文件，参照该文件中已存在的挂载信息再添加一条来挂载 vdb1。
  - 其中第一列为 UUID
  - 第二列为挂载目录，该目录必须为空目录
  - 第三列为文件系统类型
  - 第四列为参数
  - 第五列 0 表示不备份
  - 最后一列必须为 2 或 0(除非引导分区为 1)
- 最后使用 mount -a 命令来检测挂载，该命令用来检测 fstab 文件是否有错，如果有错误则不会挂载成功。

□ 其他和系统更新相关的命令

◇ sudo apt-get upgrade 更新系统安装的应用软件包

◇ sudo apt-get dist-upgrade 更新系统内核以及安装的应用软件包

◇ sudo apt-get update 修改源之后用该命令更新源(源文件：/etc/apt/source.list 文件)

◇ sudo apt-get install xxx 安装某个软件，通 tab 键可以提示软件包详细名称

◇ sudo apt-cache search xxx 搜索某个软件

◇ sudo apt-get remove xxx 卸载某个软件，通 tab 键可以提示软件包详细名称

◇ sudo apt-get autoremove 查找系统中已经不再使用的软件依赖包并进行清理

◇ 使用命令更新下载的软件包缓存在/var/cahce/apt/archives 中

◇ sudo apt-get clean 清空上述目录中的更新包的缓存

◇ 涉及到软件包的安装或者更新是单进程的，所以在同一个操作系统新同时进行多个软件的安装或者更新的时候会阻塞

□ ps 命令

◇ 命令说明：用于显示当前进程 (process) 的状态

◇ 命令语法：ps [options] [--help]

◇ 参数说明：ps 命令的参数非常多, 在此仅列出几个常用的参数并大略介绍含义

● -A 列出所有的行程

● -w 显示加宽可以显示较多的资讯

- -au 显示较详细的资讯
- -aux 显示所有包含其他使用者的行程
- au(x) 输出格式：
- USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
- USER: 行程拥有者
- PID: pid
- %CPU: 占用的 CPU 使用率
- %MEM: 占用的记忆体使用率
- VSZ: 占用的虚拟记忆体大小
- RSS: 占用的记忆体大小
- TTY: 终端的次要装置号码 (minor device number of tty)
- STAT: 该行程的状态:
- D: 不可中断的静止 (进行 I/O 动作)
- R: 正在执行中

- S: 静止状态
- T: 暂停执行
- Z: 不存在但暂时无法消除
- W: 没有足够的记忆体分页可分配
- <: 高优先序的行程
- N: 低优先序的行程
- L: 有记忆体分页分配并锁在记忆体内 (实时系统或捱 A I/O)
- START: 行程开始时间
- TIME: 执行的时间
- COMMAND:所执行的指令

※ 练习 1：显示进程信息：ps -A

※ 练习 2：显示指定用户信息：ps -u root

□ kill 命令

◇ 命令说明：用于删除执行中的程序或工作。kill 可将指定的信息送至程序。预设的信息为 SIGT

ERM(15)，可将指定程序终止。若仍无法终止该程序，可使用 SIGKILL(9)信息尝试强制删除程序。程序或工作的编号可利用 ps 指令或 jobs 指令查看。

◇ 命令语法：kill [-s <信息名称或编号>][程序]或 kill [-l <信息编号>]

◇ 参数说明：

● -l <信息编号> 若不加<信息编号>选项，则-l 参数会列出全部的信息名称。（*）

```
(1) SIGHUP       (2) SIGINT       (3) SIGQUIT      (4) SIGILL       (5) SIGTRAP
(6) SIGABRT      (7) SIGBUS       (8) SIGFPE       (9) SIGKILL      (10) SIGUSR1
(11) SIGSEGV     (12) SIGUSR2     (13) SIGPIPE     (14) SIGALRM     (15) SIGTERM
(16) SIGSTKFLT   (17) SIGCHLD     (18) SIGCONT     (19) SIGSTOP     (20) SIGTSTP
(21) SIGTTIN     (22) SIGTTOU     (23) SIGURG      (24) SIGXCPU     (25) SIGXFSZ
(26) SIGVTALRM   (27) SIGPROF     (28) SIGWINCH    (29) SIGIO       (30) SIGPWR
(31) SIGSYS      (34) SIGRTMIN    (35) SIGRTMIN+1  (36) SIGRTMIN+2  (37) SIGRTMIN+3
(38) SIGRTMIN+4  (39) SIGRTMIN+5  (40) SIGRTMIN+6  (41) SIGRTMIN+7  (42) SIGRTMIN+8
(43) SIGRTMIN+9  (44) SIGRTMIN+10 (45) SIGRTMIN+11 (46) SIGRTMIN+12 (47) SIGRTMIN+13
(48) SIGRTMIN+14 (49) SIGRTMIN+15 (50) SIGRTMAX-14 (51) SIGRTMAX-13 (52) SIGRTMAX-12
(53) SIGRTMAX-11 (54) SIGRTMAX-10 (55) SIGRTMAX-9  (56) SIGRTMAX-8  (57) SIGRTMAX-7
(58) SIGRTMAX-6  (59) SIGRTMAX-5  (60) SIGRTMAX-4  (61) SIGRTMAX-3  (62) SIGRTMAX-2
(63) SIGRTMAX-1  (64) SIGRTMAX
```

● -s <信息名称或编号> 指定要送出的信息。

● [程序] [程序]可以是程序的 PID 或是 PGID，也可以是工作编号。

10、网络基础

△ 基本命令

□ ping 命令

◇ 命令说明：用于检测主机。执行 ping 指令会使用 ICMP 传输协议，发出要求回应的信息，若远端主机的网络功能没有问题，就会回应该信息，因而得知该主机运作正常。

◇ 命令说明：ping [-dfnqrRv][-c<完成次数>][-i<间隔秒数>][-I<网络界面>][-l<前置载入>][-p<范本样式>][-s<数据包大小>][-t<存活数值>][主机名称或 IP 地址]

◇ 参数说明：

- -d 使用 Socket 的 SO_DEBUG 功能。
- -c<完成次数> 设置完成要求回应的次数。
- -f 极限检测。
- -i<间隔秒数> 指定收发信息的间隔时间。
- -I<网络界面> 使用指定的网络界面送出数据包。

- -l<前置载入> 设置在送出要求信息之前，先行发出的数据包。
- -n 只输出数值。
- -p<范本样式> 设置填满数据包的范本样式。
- -q 不显示指令执行过程，开头和结尾的相关信息除外。
- -r 忽略普通的 Routing Table，直接将数据包送到远端主机上。
- -R 记录路由过程。
- -s<数据包大小> 设置数据包的大小。
- -t<存活数值> 设置存活数值 TTL 的大小。
- -v 详细显示指令的执行过程。

□ ifconfig 命令

◇ 命令说明：显示或设置网络设备。ifconfig 可设置网络设备的状态，或是显示目前的设置。

◇ 命令语法：ifconfig [网络设备][down up -allmulti -arp -promisc][add<地址>][del<地址>][<hw<网络设备类型><硬件地址>][io_addr<I/O 地址>][irq<IRQ 地址>][media<网络媒介类型>][mem_start<内存地址>][metric<数目>][mtu<字节>][netmask<子网掩码>][tunnel<地址>][-broadcast<地址>][-pointopoint<地址>][IP

地址]

◇ 参数说明：

- add<地址> 设置网络设备 IPv6 的 IP 地址。
- del<地址> 删除网络设备 IPv6 的 IP 地址。
- down 关闭指定的网络设备。
- hw<网络设备类型><硬件地址> 设置网络设备的类型与硬件地址。
- io_addr<I/O 地址> 设置网络设备的 I/O 地址。
- irq<IRQ 地址> 设置网络设备的 IRQ。
- media<网络媒介类型> 设置网络设备的媒介类型。
- mem_start<内存地址> 设置网络设备在主内存所占用的起始地址。
- metric<数目> 指定在计算数据包的转送次数时，所要加上的数目。
- mtu<字节> 设置网络设备的 MTU。
- netmask<子网掩码> 设置网络设备的子网掩码。
- tunnel<地址> 建立 IPv4 与 IPv6 之间的隧道通信地址。

- up 启动指定的网络设备。
- -broadcast<地址> 将要送往指定地址的数据包当成广播数据包来处理。
- -pointopoint<地址> 与指定地址的网络设备建立直接连线，此模式具有保密功能。
- -promisc 关闭或启动指定网络设备的 promiscuous 模式。
- [IP 地址] 指定网络设备的 IP 地址。
- [网络设备] 指定网络设备的名称。

※ 练习 1：显示网络设备信息

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:0A:0B:0C
          inet addr:192.168.0.3  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe0a:b0c/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500  Metric:1
          RX packets:172220  errors:0  dropped:0 overruns:0  frame:0
          TX packets:132379  errors:0  dropped:0 overruns:0  carrier:0
```

collisions:0 txqueuelen:1000

RX bytes:87101880 (83.0 MiB) TX bytes:41576123 (39.6 MiB)

Interrupt:185 Base address:0x2024

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:16436 Metric:1

RX packets:2022 errors:0 dropped:0 overruns:0 frame:0

TX packets:2022 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:2459063 (2.3 MiB) TX bytes:2459063 (2.3 MiB)

※ 练习 2：启动关闭指定网卡

ifconfig eth0 down

ifconfig eth0 up

※ 练习 3：为网卡配置和删除 IPv6 地址

ifconfig eth0 add 33ffe:3240:800:1005::2/ 64 //为网卡设置 IPv6 地址

ifconfig eth0 del 33ffe:3240:800:1005::2/ 64 //为网卡删除 IPv6 地址

※ 练习 4：用 ifconfig 修改 MAC 地址

ifconfig eth0 down //关闭网卡

ifconfig eth0 hw ether 00:AA:BB:CC:DD:EE //修改 MAC 地址

ifconfig eth0 up //启动网卡

ifconfig eth1 hw ether 00:1D:1C:1D:1E //关闭网卡并修改 MAC 地址

ifconfig eth1 up //启动网卡

※ 练习 5：配置 IP 地址

ifconfig eth0 192.168.1.56

// 给 eth0 网卡配置 IP 地址

ifconfig eth0 192.168.1.56 netmask 255.255.255.0

// 给 eth0 网卡配置 IP 地址,并加上子掩码

ifconfig eth0 192.168.1.56 netmask 255.255.255.0 broadcast 192.168.1.255

// 给 eth0 网卡配置 IP 地址,加上子掩码,加上个广播地址

※ 练习 6：启用和关闭 ARP 协议

ifconfig eth0 arp //开启

ifconfig eth0 -arp //关闭

※ 练习 7：设置最大传输单元

ifconfig eth0 mtu 1500

//设置能通过的最大数据包大小为 1500 bytes

□ netstat 命令

◇ 命令说明：用于显示网络状态。利用 netstat 指令可让你得知整个 Linux 系统的网络情况。

◇ 命令语法：netstat [-acCeFghilMnNoprstuvVwx][-A<网络类型>][--ip]

◇ 参数说明：

● -a 或--all 显示所有连线中的 Socket。

- -A<网络类型>或--<网络类型> 列出该网络类型连线中的相关地址。
- -c 或--continuous 持续列出网络状态。
- -C 或--cache 显示路由器配置的快取信息。
- -e 或--extend 显示网络其他相关信息。
- -F 或--fib 显示 FIB。
- -g 或--groups 显示多重广播功能群组组员名单。
- -h 或--help 在线帮助。
- -i 或--interfaces 显示网络界面信息表单。
- -l 或--listening 显示监控中的服务器的 Socket。
- -M 或--masquerade 显示伪装的网络连线。
- -n 或--numeric 直接使用 IP 地址，而不通过域名服务器。
- -N 或--netlink 或--symbolic 显示网络硬件外围设备的符号连接名称。
- -o 或--timers 显示计时器。
- -p 或--programs 显示正在使用 Socket 的程序识别码和程序名称。

- -r 或--route 显示 Routing Table。
- -s 或--statistice 显示网络工作信息统计表。
- -t 或--tcp 显示 TCP 传输协议的连线状况。
- -u 或--udp 显示 UDP 传输协议的连线状况。
- -v 或--verbose 显示指令执行过程。
- -V 或--version 显示版本信息。
- -w 或--raw 显示 RAW 传输协议的连线状况。
- -x 或--unix 此参数的效果和指定"-A unix"参数相同。
- --ip 或--inet 此参数的效果和指定"-A inet"参数相同。

※ 练习 1：显示详细的网络状况

```
# netstat -a
```

※ 练习 2：显示当前户籍 UDP 连接状况

```
# netstat -nu
```

□ 远程控制命令

◇ telnet 命令

○ 命令说明：用于远端登入。执行 telnet 指令开启终端机阶段作业，并登入远端主机。

○ 命令语法：telnet [-8acdEfFKLrx][-b<主机别名>][-e<脱离字符>][-k<域名>][-l<用户名称>][-n<记录文件>][-S<服务类型>][-X<认证形态>][主机名称或 IP 地址<通信端口>]

○ 参数说明：

- -8 允许使用 8 位字符资料，包括输入与输出。
- -a 尝试自动登入远端系统。
- -b<主机别名> 使用别名指定远端主机名称。
- -c 不读取用户专属目录里的.telnetrc 文件。
- -d 启动排错模式。
- -e<脱离字符> 设置脱离字符。
- -E 滤除脱离字符。
- -f 此参数的效果和指定"-F"参数相同。
- -F 使用 Kerberos V5 认证时，加此参数可把本地主机的认证数据上传到远端主机。

- -k<域名> 使用 Kerberos 认证时，加上此参数让远端主机采用指定的领域名，而非该主机的域名。
- -K 不自动登入远端主机。
- -l<用户名称> 指定要登入远端主机的用户名称。
- -L 允许输出 8 位字符资料。
- -n<记录文件> 指定文件记录相关信息。
- -r 使用类似 rlogin 指令的用户界面。
- -S<服务类型> 设置 telnet 连线所需的 IP TOS 信息。
- -x 假设主机有支持数据加密的功能，就使用它。
- -X<认证形态> 关闭指定的认证形态。

※ 练习：登录远程主机

telnet 192.168.0.5

//登录 IP 为 192.168.0.5 的远程主机，执行该命令后需要输入用户名和密码

◇ ssh 命令

- 命令说明：用于远端登入。执行 ssh 指令开启终端机阶段作业，并登入远端主机。
- 命令语法：ssh 用户名@IP，执行该命令后需要输入密码
- 和 ssh 相关的远程拷贝命令 scp：
  - 命令说明：进行远程拷贝
  - 命令语法：scp [-1246BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file] [-l limit] [-o ssh_option] [-P port] [-S program] [[user@]host1:]file1 [...] [[user@]host2:]file2
  - 简易写法：scp [option(s)] source destination
  - 参数说明：
    - -1： 强制 scp 命令使用协议 ssh1
    - -2： 强制 scp 命令使用协议 ssh2
    - -4： 强制 scp 命令只使用 IPv4 寻址
    - -6： 强制 scp 命令只使用 IPv6 寻址
    - -B： 使用批处理模式（传输过程中不询问传输口令或短语）
    - -C： 允许压缩。（将-C 标志传递给 ssh，从而打开压缩功能）

∵ -p：保留原文件的修改时间，访问时间和访问权限。

∵ -q： 不显示传输进度条。

∵ -r： 递归复制整个目录。

∵ -v：详细方式显示输出。scp 和 ssh(1)会显示出整个过程的调试信息。这些信息用于调试连接，验证和配置问题。

∵ -c cipher：以 cipher 将数据传输进行加密，这个选项将直接传递给 ssh。

∵ -F ssh_config：指定一个替代的 ssh 配置文件，此参数直接传递给 ssh。

∵ -i identity_file：从指定文件读取传输时使用的密钥文件，此参数传递给 ssh。

∵ -l limit： 限定用户所能使用的带宽，以 Kbit/s 为单位。

∵ -o ssh_option： 如果习惯于使用 ssh_config(5)中的参数传递方式，

∵ -P port：注意是大写的 P, port 是指定数据传输用到的端口号

∵ -S program：指定加密传输时所使用的程序。此程序必须能够理解 ssh 的选项。

○ 配置 ssh 免密码登录：

● 前提条件：进行 SSH 免密登录的前提条件是两台机器的用户名相同。例如有 A、B

两台机器，这两台机器上都有一个用户名同为 briup 的用户。现在的需求是在 A 机器上通过免密(不输入密码，即免密)方式登录 B 机器。

- 第一步：分别在 A 和 B 机器上使用命令“ssh-keygen”生成公钥和私钥。

执行该命令之后会在 briup 用户的家目录下生成“.ssh/”目录和在该“.ssh/”目录下生成公钥“id_rsa.pub”文件和私钥“id_rsa”文件。

注意：公钥用来加密，私钥用来解密。

- 第二步：将 A 机器上的 briup 用户家目录下的.ssh/目录下的 id_rsa.pub 拷贝到 B 机器的 briup 用户的家目录下(注意是家目录，不是.ssh 目录)。

命令：scp /home/briup/.ssh/id_rsa.pub briup@B:~

命令说明：将上述命令中的“B”更换为具体的主机名或者是 IP。注意是拷贝到 B 机器上的 briup 用户的家目录，不是“.ssh”目录。

- 第三步：在 B 机器上，将刚刚从 A 机器上拷贝过来的“id_rsa.pub”(即 B 机器 briup 用户家目录下的“id_rsa.pub”)的内容复制到 B 机器上的 briup 家目录下的“.ssh”目录下的“authorized_keys”文件中。

命令：cat /home/briup/id_rsa.pub >> /home/briup/.ssh/authorized_keys

命令说明：上述命令是在 B 机器上执行

- 第四步：更改第三步执行命令所生成的“authorized_keys”文件的操作权限为 600.

命令：chmod 600 /home/briup/.ssh/authorized_keys

命令说明：上述命令在 B 机器上执行

- 第五步：在 A 机器上通过无密码登录 B 机器。

命令：ssh B

命令说明：上述命令在 B 机器上执行，将“B”更换为具体的主机名或 IP 地址。

- 额外说明：在使用 SSH 工具进行远程登录的时候，如果登录的两个机器的用户名相同，可以直接使用命令“ssh 主机名/IP”来远程登录，而不需要输入用户名，即不需要以“ssh briup@主机名/IP”这样的方式登录。

□ 远程传输命令

◇ ftp 命令

○ 命令说明：设置文件系统相关功能。FTP 是 ARPANet 的标准文件传输协议，该网络就

是现今 Internet 的前身。

○ 命令语法：ftp [-dignv][主机名称或 IP 地址]

○ 参数说明：

- -d 详细显示指令执行过程，便于排错或分析程序执行的情形。
- -i 关闭互动模式，不询问任何问题。
- -g 关闭本地主机文件名称支持特殊字符的扩充特性。
- -n 不使用自动登陆。
- -v 显示指令执行过程。

※ 练习：例如使用 ftp 命令匿名登录 ftp.kernel.org 服务器，该服务是 Linux 内核的官方服务器，可以使用如下命令：

```
ftp ftp.kernel.org #发起链接请求
```

11、Shell 脚本

△ 具体的 Shell 脚本编程请参见：http://dwz.cn/4sAupl

△ crontab 命令

□ 命令说明：

○ 用来定期执行程序的命令。当安装完成操作系统之后，默认便会启动此任务调度命令。

○ crond 命令每分锺会定期检查是否有要执行的工作，如果有要执行的工作便会自动执行该工作。

○ 而 linux 任务调度的工作主要分为以下两类：

● 系统执行的工作：系统周期性所要执行的工作，如备份系统数据、清理缓存

● 个人执行的工作：某个用户定期要做的工作，例如每隔 10 分钟检查邮件服务器是否有新信，这些工作可由每个用户自行设置

□ 命令语法：crontab [-u user] file 或者 crontab [-u user] { -l | -r | -e }

□ 参数说明：

○ -e ：执行文字编辑器来设定时程表，内定的文字编辑器是 VI，如果你想用别的文字编辑器，则请先设定 VISUAL 环境变数来指定使用那个文字编辑器(比如说 setenv VISUAL joe)

○ -r ：删除目前的时程表

○ -l ：列出目前的时程表

□ 具体说明：

○ crontab 是用来让使用者在固定时间或固定间隔执行程序之用，换句话说，也就是类似使用者的时程表。

○ -u user 是指设定指定 user 的时程表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 -u user 的话，就是表示设定自己的时程表。

○ 时程表的格式如下：f1 f2 f3 f4 f5 program

● 其中 f1 是表示分钟，f2 表示小时，f3 表示一个月份中的第几日，f4 表示月份，f5 表示一个星期中的第几天。program 表示要执行的程序。

● 当 f1 为 * 时表示每分钟都要执行 program，f2 为 * 时表示每小时都要执行程序，其馀类推

● 当 f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时表示从第 a 到第 b 小时都要执行，其馀类推

● 当 f1 为 */n 时表示每 n 分钟个时间间隔执行一次，f2 为 */n 表示每 n 小时个时间

间隔执行一次，其馀类推

● 当 f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示第 a, b, c...个小时要执行，其馀类推

○ 使用者也可以将所有的设定先存放在文件中，用 crontab file 的方式来设定时程表。

※ 示例

○ 每月每天每小时的第 0 分钟执行一次 /bin/ls

0 7 * * * /bin/ls

○ 在 12 月内, 每天的早上 6 点到 12 点中，每隔 20 分钟执行一次 /usr/bin/backup

0 6-12/3 * 12 * /usr/bin/backup

○ 周一到周五每天下午 5:00 寄一封信给 alex@domain.name

0 17 * * 1-5 mail -s "hi" alex@domain.name < /tmp/maildata

○ 每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分....执行 echo "haha"

20 0-23/2 * * * echo "haha"

○ 下面再看看几个具体的例子：

0 */2 * * * /sbin/service httpd restart 意思是每两个小时重启一次 apache

50 7 * * * /sbin/service sshd start 意思是每天 7：50 开启 ssh 服务

50 22 * * * /sbin/service sshd stop 意思是每天 22：50 关闭 ssh 服务

0 0 1,15 * * fsck /home 每月 1 号和 15 号检查/home 磁盘

1 * * * * /home/bruce/backup 每小时的第一分执行 /home/bruce/backup 这个文件

00 03 * * 1-5 find /home "*.xxx" -mtime +4 -exec rm {} \; 每周一至周五 3 点钟，在目录/home 中，查找文件名为*.xxx 的文件，并删除 4 天前的文件。

30 6 */10 * * ls 意思是每月的 1、11、21、31 日是的 6：30 执行一次 ls 命令

※ 注意：当程序在你所指定的时间执行后，系统会寄一封信给你，显示该程序执行的内容，若是你不希望收到这样的信，请在每一行空一格之后加上 > /dev/null 2>&1 即可。

12、设置初始化文件

△ 环境变量的设置

□ 全局设置：修改/etc/profile 文件

□ 用户设置：修改用户家目录下的.bashrc 文件

□ 设置方式：export key=value

□ 当有环境变量引用时，被引用的环境变量必须用“$”标注

△ which 命令

□ 命令说明：用于查找文件。which 指令会在环境变量$PATH 设置的目录里查找符合条件的文件。

□ 命令语法：which [文件...]

□ 参数说明：

- -n<文件名长度> 指定文件名长度，指定的长度必须大于或等于所有文件中最长的文件名。
- -p<文件名长度> 与-n 参数相同，但此处的<文件名长度>包括了文件的路径。
- -w 指定输出时栏位的宽度。
- -V 显示版本信息。

※ 练习 使用指令"which"查看指令"bash"的绝对路径。

$ which bash，该命令执行后，输出信息如下所示：/bin/bash #bash 可执行程序的绝对路径

△ history 命令 查看使用过的命令的历史记录

△ whereis 命令

□ 命令说明：用于查找文件。该指令会在特定目录中查找符合条件的文件。这些文件应属于原始代码、二进制文件，或是帮助文件。该指令只能用于查找二进制文件、源代码文件和 man 手册页，一般文件的定位需使用 locate 命令。

□ 命令语法：whereis [-bfmsu][-B <目录>...][-M <目录>...][-S <目录>...][文件...]

□ 参数说明：

- -b 只查找二进制文件。
- -B<目录> 只在设置的目录下查找二进制文件。
- -f 不显示文件名前的路径名称。
- -m 只查找说明文件。
- -M<目录> 只在设置的目录下查找说明文件。
- -s 只查找原始代码文件。
- -S<目录> 只在设置的目录下查找原始代码文件。
- -u 查找不包含指定类型的文件。

※ 练习 1：使用指令"whereis"查看指令"bash"的位置，输入如下命令：

$ whereis bash 指令执行后，输出信息如下所示： bash:/bin/bash/etc/bash.bashrc/usr/share/man/man1/bash.1.gz

注意：以上输出信息从左至右分别为查询的程序名、bash 路径、bash 的 man 手册页路径。

※ 练习 2：如果用户需要单独查询二进制文件或帮助文件，可使用如下命令：

$ whereis -b bash

$ whereis -m bash

输出信息如下：

$ whereis -b bash #显示 bash 命令的二进制程序

bash: /bin/bash /etc/bash.bashrc /usr/share/bash # bash 命令的二进制程序的地址

$ whereis -m bash #显示 bash 命令的帮助文件

bash: /usr/share/man/man1/bash.1.gz #bash 命令的帮助文件地址

△ 建立别名

□ 命令格式：alias alias-name value

□ 应用举例：

```
$ alias  h  history
$ alias  c  clear
$ alias  home  cd;ls
$ alias  ls  ls  -l
$ alias  copy  cp  -i
$ alias
$ unalias  copy
```

13、用户管理

△ Linux 系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。

△ 用户的账号一方面可以帮助系统管理员对使用系统的用户进行跟踪，并控制他们对系统资源的访问；另一方面也可以帮助用户组织文件，并为用户提供安全性保护。

△ 每个用户账号都拥有一个惟一的用户名和各自的口令。

△ 用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。

△ 实现用户账号的管理，要完成的工作主要有如下几个方面：

□ 用户账号的添加、删除与修改。

□ 用户口令的管理。

□ 用户组的管理。

△ Linux 系统用户账号的管理

□ 用户账号的管理工作主要涉及到用户账号的添加、修改和删除。

□ 添加用户账号就是在系统中创建一个新账号，然后为新账号分配用户号、用户组、主目录和登录 Shell 等资源。刚添加的账号是被锁定的，无法使用。

□ 添加新的用户账号使用 useradd 命令，其语法如下：

○ $ useradd 选项 用户名

○ 参数说明：

● -c comment 指定一段注释性描述。描述信息会保存在 passwd 的备注栏位中。

    - -d 目录 指定用户主目录，如果此目录不存在，则同时使用-m 选项，可以创建主目录。
    - -g 用户组 指定用户所属的用户组。
    - -G 用户组，用户组 指定用户所属的附加组。
    - -s Shell 文件 指定用户的登录 Shell。
    - -u 用户号 指定用户的用户号，如果同时有-o 选项，则可以重复使用其他用户的标识号。
- -D 变更预设值.
- -e 指定帐号的有效期限。
- -f 指定在密码过期后多少天即关闭该帐号。
- -m 自动建立用户的登入目录。
- -M 不要自动建立用户的登入目录。
- -n 取消建立以用户名称为名的群组.
- -r 建立系统帐号。
- -s 指定用户登入后所使用的 shell。
- -u 指定用户 ID。

※ 实例 1

useradd –d /usr/sam -m sam

此命令创建了一个用户 sam，其中-d 和-m 选项用来为登录名 sam 产生一个主目录/home/sam（/home 为默认的用户主目录所在的父目录）。

※ 实例 2

useradd -s /bin/bash -g group –G adm,root gem

此命令新建了一个用户 gem，该用户的登录 Shell 是 /bin/bash，它属于 group 用户组，同时又属于 adm 和 root 用户组，其中 group 用户组是其主组。

这里可能新建组：#groupadd group 及 groupadd adm

增加用户账号就是在/etc/passwd 文件中为新用户增加一条记录，同时更新其他系统文件如/etc/shadow, /etc/group 等。

Linux 提供了集成的系统管理工具 userconf，它可以用来对用户账号进行统一管理。

□ 删除帐号

○ 如果一个用户的账号不再使用，可以从系统中删除。删除用户账号就是要将/etc/passwd 等系

统文件中的该用户记录删除，必要时还删除用户的主目录。

○ 删除一个已有的用户账号使用 userdel 命令，其格式如下：

userdel 选项 用户名

○ 常用的选项是-r，它的作用是把用户的主目录一起删除。

※ 例如：

userdel sam

此命令删除用户 sam 在系统文件中（主要是/etc/passwd, /etc/shadow, /etc/group 等）的记录，同时删除用户的主目录。

□ 修改帐号

○ 修改用户账号就是更改用户的有关属性，如用户号、主目录、用户组、登录 Shell 等。

○ 修改已有用户的信息使用 usermod 命令，其格式如下：

usermod 选项 用户名

○ 常用的选项包括-c, -d, -m, -g, -G, -s, -u 以及-o 等，这些选项的意义与 useradd 命令中的选项一样，可以为用户指定新的资源值。

○ 另外，有些系统可以使用选项：-l 新用户名

○ 这个选项指定一个新的账号，即将原来的用户名改为新的用户名。

※ 例如：

usermod -s /bin/ksh -d /home/z –g developer sam

此命令将用户 sam 的登录 Shell 修改为 ksh，主目录改为/home/z，用户组改为 developer。

□ 用户密码的管理

○ 用户管理的一项重要内容是用户口令的管理。用户账号刚创建时没有口令，但是被系统锁定，无法使用，必须为其指定口令后才可以使用，即使是指定空口令。

○ 指定和修改用户口令的 Shell 命令是 passwd。超级用户可以为自己和其他用户指定口令，普通用户只能用它修改自己的口令。命令的格式为：

passwd 选项 用户名

○ 可使用的选项：

-l 锁定口令，即禁用账号。

-u 口令解锁。

-d 使账号无口令。

-f 强迫用户下次登录时修改口令。

○ 如果默认用户名，则修改当前用户的口令。

※ 例如，假设当前用户是 sam，则下面的命令修改该用户自己的口令：

$ passwd

Old password：******

New password：*******

Re-enter new password：*******

※ 如果是超级用户，可以用下列形式指定任何用户的口令：

passwd sam

New password：*******

Re-enter new password：*******

○ 普通用户修改自己的口令时，passwd 命令会先询问原口令，验证后再要求用户输入两遍新口令，如果两次输入的口令一致则将这个口令指定给用户；而超级用户为用户指定口令时，不需要知道原口令。

○ 为了系统安全起见，用户应该选择比较复杂的口令，例如最好使用 8 位长的口令，口令中包含有大写、小写字母和数字，并且应该与姓名、生日等不相同。

○ 为用户指定空口令时，执行下列形式的命令：

passwd -d sam

此命令将用户 sam 的口令删除，这样用户 sam 下一次登录时，系统就不再询问口令。

○ passwd 命令还可以用-l(lock)选项锁定某一用户，使其不能登录，例如：

passwd -l sam

△ Linux 系统用户组的管理

□ 每个用户都有一个用户组，系统可以对一个用户组中的所有用户进行集中管理。不同 Linux 系统对用户组的规定有所不同，如 Linux 下的用户属于与它同名的用户组，这个用户组在创建用户时同时创建。

□ 用户组的管理涉及用户组的添加、删除和修改。组的增加、删除和修改实际上就是对/etc/group 文件的更新。

□ 增加一个新的用户组使用 groupadd 命令。其格式为：groupadd 选项 用户组

□ 可以使用的选项有：

-g GID 指定新用户组的组标识号（GID）。

-o 一般与-g 选项同时使用，表示新用户组的 GID 可以与系统已有用户组的 GID 相同。

※ 实例 1：

groupadd group1

此命令向系统中增加了一个新组 group1，新组的 GID 是在当前已有的最大的 GID 基础上加 1。

※ 实例 2：

groupadd -g 101 group2

此命令向系统中增加了一个新组 group2，同时指定新组的组标识号是 101。

□ 如果要删除一个已有的用户组，使用 groupdel 命令，其格式如下：

groupdel 用户组

※ 例如：

groupdel group1

此命令从系统中删除组 group1。

□ 修改用户组的属性使用 groupmod 命令。其语法如下：

groupmod 选项 用户组

□ 常用的选项有：

-g GID 为用户组指定新的组标识号。

-o 与-g 选项同时使用，用户组的新 GID 可以与系统已有用户组的 GID 相同。

-n 新用户组 将用户组的名字改为新名字

※ 实例 1：

groupmod -g 102 group2

此命令将组 group2 的组标识号修改为 102。

※ 实例 2：

groupmod –g 10000 -n group3 group2

此命令将组 group2 的标识号改为 10000，组名修改为 group3。

□ 如果一个用户同时属于多个用户组，那么用户可以在用户组之间切换，以便具有其他用户组权限。

□ 用户可以在登录后，使用命令 newgrp 切换到其他用户组，该命令的参数就是目的用户组。例如：

$ newgrp root

这条命令将当前用户切换到root用户组，前提条件是root用户组确实是该用户的主组或附加组。

类似于用户账号的管理，用户组的管理也可以通过集成的系统管理工具来完成。

△ 与用户账号有关的系统文件

□ 完成用户管理的工作有许多种方法，但是每一种方法实际上都是对有关的系统文件进行修改。

□ 与用户和用户组相关的信息都存放在一些系统文件中，如/etc/passwd，/etc/shadow，/etc/group 等。

□ /etc/passwd 文件是用户管理工作涉及的最重要的一个文件。

□ Linux 系统中的每个用户都在/etc/passwd 文件中有一个对应的记录行，它记录了这个用户的一些基本属性。

□ 这个文件对所有用户都是可读的。

```
# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

sync:x:4:65534:sync:/bin:/bin/sync

games:x:5:60:games:/usr/games:/usr/sbin/nologin

man:x:6:12:man:/var/cache/man:/usr/sbin/nologin

lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin

mail:x:8:8:mail:/var/mail:/usr/sbin/nologin

news:x:9:9:news:/var/spool/news:/usr/sbin/nologin

uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin

proxy:x:13:13:proxy:/bin:/usr/sbin/nologin

www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin

backup:x:34:34:backup:/var/backups:/usr/sbin/nologin

list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin

irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin

gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin

nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin

syslog:x:101:104::/home/syslog:/bin/false

messagebus:x:102:106::/var/run/dbus:/bin/false

landscape:x:103:109::/var/lib/landscape:/bin/false

sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin

briup:x:1000:1000:briup,,,:/home/briup:/bin/bash

□ 从上面的例子我们可以看到，/etc/passwd 中一行记录对应着一个用户，每行记录又被冒号(:)分隔为 7 个字段，其格式和具体含义如下：

用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录 Shell

○ "用户名"是代表用户账号的字符串。

● 通常长度不超过 8 个字符，并且由大小写字母和/或数字组成。登录名中不能有冒号(:)，因为冒号在这里是分隔符。

● 为了兼容起见，登录名中最好不要包含点字符(.)，并且不使用连字符(-)和加号(+)打头。

○ “口令”一些系统中，存放着加密后的用户口令字。

● 虽然这个字段存放的只是用户口令的加密串，不是明文，但是由于/etc/passwd 文件对所

有用户都可读，所以这仍是一个安全隐患。因此，现在许多 Linux 系统（如 SVR4）都使用了 shadow 技术，把真正的加密后的用户口令字存放到/etc/shadow 文件中，而在/etc/passwd 文件的口令字段中只存放一个特殊的字符，例如“x”或者“*”。

○ “用户标识号”是一个整数，系统内部用它来标识用户。

● 一般情况下它与用户名是一一对应的。如果几个用户名对应的用户标识号是一样的，系统内部将把它们视为同一个用户，但是它们可以有不同的口令、不同的主目录以及不同的登录 Shell 等。

● 通常用户标识号的取值范围是 0～65 535。0 是超级用户 root 的标识号，1～99 由系统保留，作为管理账号，普通用户的标识号从 100 开始。在 Linux 系统中，这个界限是 500。

○ “组标识号”字段记录的是用户所属的用户组。

● 它对应着/etc/group 文件中的一条记录。

○ “注释性描述”字段记录着用户的一些个人情况。

● 例如用户的真实姓名、电话、地址等，这个字段并没有什么实际的用途。在不同的 Linux 系统中，这个字段的格式并没有统一。在许多 Linux 系统中，这个字段存放的是一段任意的注释性描述文字，用做 finger 命令的输出。

○ “主目录”，也就是用户的起始工作目录。

● 它是用户在登录到系统之后所处的目录。在大多数系统中，各用户的主目录都被组织在同一个特定的目录下，而用户主目录的名称就是该用户的登录名。各用户对自己的主目录有读、写、执行（搜索）权限，其他用户对此目录的访问权限则根据具体情况设置。

○ 用户登录后，要启动一个进程，负责将用户的操作传给内核，这个进程是用户登录到系统后运行的命令解释器或某个特定的程序，即 Shell。

● Shell 是用户与 Linux 系统之间的接口。Linux 的 Shell 有许多种，每种都有不同的特点。常用的有 sh(Bourne Shell), csh(C Shell), ksh(Korn Shell), tcsh(Type C Shell), bash(Bourne Again Shell)等。

● 系统管理员可以根据系统情况和用户习惯为用户指定某个 Shell。如果不指定 Shell，那么系统使用 sh 为默认的登录 Shell，即这个字段的值为/bin/sh。

● 用户的登录 Shell 也可以指定为某个特定的程序（此程序不是一个命令解释器）。

● 利用这一特点，我们可以限制用户只能运行指定的应用程序，在该应用程序运行结束后，用户就自动退出了系统。有些 Linux 系统要求只有那些在系统中登记了的程序才能出现在这个字段中。

□ 系统中有一类用户称为伪用户（Psuedo Users）。

○ 这些用户在/etc/passwd 文件中也占有一条记录，但是不能登录，因为它们的登录 Shell 为空。它们的存在主要是方便系统管理，满足相应的系统进程对文件属主的要求。常见的伪用户如下所示：

- bin 拥有可执行的用户命令文件
- sys 拥有系统文件
- adm 拥有帐户文件
- uucp UUCP 使用
- lp lp 或 lpd 子系统使用
- nobody NFS 使用

□ /etc/shadow 中的行与/etc/passwd 一一对应，它由 pwconv 命令根据/etc/passwd 中的数据自动产生

○ 它的文件格式与/etc/passwd 类似，由若干个字段组成，字段之间用":"隔开。

○ "登录名"是与/etc/passwd 文件中的登录名相一致的用户账号

○ "口令"字段存放的是加密后的用户口令字，长度为 13 个字符。如果为空，则对应用户没有口令，登录时不需要口令；如果含有不属于集合 { ./0-9A-Za-z }中的字符，则对应的用户不能登录。

○ "最后一次修改时间"表示的是从某个时刻起，到用户最后一次修改口令时的天数。时间起点

对不同的系统可能不一样。例如在 SCO Linux 中，这个时间起点是 1970 年 1 月 1 日。

○ "最小时间间隔"指的是两次修改口令之间所需的最小天数。

○ "最大时间间隔"指的是口令保持有效的最大天数。

○ "警告时间"字段表示的是从系统开始警告用户到用户密码正式失效之间的天数。

○ "不活动时间"表示的是用户没有登录活动但账号仍能保持有效的最大天数。

○ "失效时间"字段给出的是一个绝对的天数，如果使用了这个字段，那么就给出相应账号的生存期。期满后，该账号就不再是一个合法的账号，也就不能再用来登录了。

※ 下面是/etc/shadow 的一个例子：

```
# cat /etc/shadow
sys:*:16652:0:99999:7:::
sync:*:16652:0:99999:7:::
games:*:16652:0:99999:7:::
man:*:16652:0:99999:7:::
lp:*:16652:0:99999:7:::
```

briup:6hTDplD7m$213DG3m7raGEnbIBvo6nHgmiVS.v3nfuMTq0./2RcL.cZHoggYZAAhuFDVL8yTzpTvHTfRLZxHxHcLJ/Jg5pd/:16875:0:99999:7:::

kevin:6ZKZ8.5.E$lz7ZKiWjOn700bgs4tdyeXDUPTxqsgmsoQhGUJHDzytwAWw1rBTkB9crSwGsJ5N1uGEX8ru36TPU7XMBW4HoB1:17212:0:99999:7:::

test:6XnlFasVr$SZloMrPk4TGiLC29pyueV8DptTtQQnYuVGhOWtjD7MAnRMdn4n9yhm1VL4/WYc0WIRBdGmkmShm0bmsaNkQ0r1:16924:0:99999:7:::

□ 用户组的所有信息都存放在/etc/group 文件中。

○ 将用户分组是 Linux 系统中对用户进行管理及控制访问权限的一种手段。

○ 每个用户都属于某个用户组；一个组中可以有多个用户，一个用户也可以属于不同的组。

○ 当一个用户同时是多个组中的成员时，在/etc/passwd 文件中记录的是用户所属的主组，也就是登录时所属的默认组，而其他组称为附加组。

○ 用户要访问属于附加组的文件时，须先使用 newgrp 命令使自己成为所要访问的组中的成员。

○ 用户组的所有信息都存放在/etc/group 文件中。此文件的格式也类似于/etc/passwd 文件，由冒号(:)隔开若干个字段，这些字段有：

组名:口令:组标识号:组内用户列表

○ "组名"是用户组的名称，由字母或数字构成。与/etc/passwd 中的登录名一样，组名不应重复。

○ "口令"字段存放的是用户组加密后的口令字。一般 Linux 系统的用户组都没有口令，即这个字段一般为空，或者是*。

○ "组标识号"与用户标识号类似，也是一个整数，被系统内部用来标识组。

○ "组内用户列表"是属于这个组的所有用户的列表，不同用户之间用逗号(,)分隔。这个用户组可能是用户的主组，也可能是附加组。

※ 示例：/etc/group 文件的一个例子如下：

root::0:root

bin::2:root,bin

sys::3:root,uucp

adm::4:root,adm

daemon::5:root,daemon

lp::7:root,lp

users::20:root,sam

△ 批量添加用户

□ 添加和删除用户对每位 Linux 系统管理员都是轻而易举的事，比较棘手的是如果要添加几十个、上百个甚至上千个用户时，我们不太可能还使用 useradd 一个一个地添加，必然要找一种简便的创建大量用户的方法。Linux 系统提供了创建大量用户的工具，可以让您立即创建大量用户，方法如下：

□ 先编辑一个文本用户文件。每一列按照/etc/passwd 密码文件的格式书写，要注意每个用户的用户名、UID、宿主目录都不可以相同，其中密码栏可以留做空白或输入 x 号。一个范例文件 user.txt 内容如下：

user001::600:100:user:/home/user001:/bin/bash

user002::601:100:user:/home/user002:/bin/bash

user003::602:100:user:/home/user003:/bin/bash

user004::603:100:user:/home/user004:/bin/bash

user005::604:100:user:/home/user005:/bin/bash

user006::605:100:user:/home/user006:/bin/bash

□ 以 root 身份执行命令 /usr/sbin/newusers，从刚创建的用户文件 user.txt 中导入数据，创建用户：

newusers < user.txt

□ 然后可以执行命令 vipw 或 vi /etc/passwd 检查 /etc/passwd 文件是否已经出现这些用户的数据，并且用户的宿主目录是否已经创建。

□ 执行命令/usr/sbin/pwunconv。

□ /etc/shadow 产生的 shadow 密码解码，然后回写到 /etc/passwd 中，并将/etc/shadow 的 shadow 密码栏删掉。这是为了方便下一步的密码转换工作，即先取消 shadow password 功能。

pwunconv

□ 编辑每个用户的密码对照文件。

范例文件 passwd.txt 内容如下：

user001:密码

user002:密码

user003:密码

user004:密码

user005:密码

user006:密码

□ 以 root 身份执行命令 /usr/sbin/chpasswd。创建用户密码，chpasswd 会将经过 /usr/bin/passwd 命令编码过的密码写入 /etc/passwd 的密码栏。

chpasswd < passwd.txt

□ 确定密码经编码写入/etc/passwd 的密码栏后。执行命令 /usr/sbin/pwconv 将密码编码为 shadow password，并将结果写入/etc/shadow。

pwconv

□ 这样就完成了大量用户的创建了，之后您可以到/home 下检查这些用户宿主目录的权限设置是否都正确，并登录验证用户密码是否正确。

14、Linux NFS 服务管理

△ NFS 简介

□ NFS(网络文件系统,Network File System)采用 CS 的工作模式

□ NFS 是分布式计算系统的一个组成部分，可以实现在异种网络上共享和装配远程文件系统

□ NFS 提供了一种在类 Unix 系统上的共享文件的方法

□ NFS 还可以结合远程网络启动实现

○ 无盘工作站(PXE 启动，所有数据都在服务器的磁盘阵列上)

○ 瘦客户工作站(本地启动系统，本地磁盘存储常用系统工具，而所有/home 目录的用户数据被放在 NFS 服务器上并且在网络上处处可见)

△ NFS 协议以及各版本的组要差别

□ V3 相对 V2 的主要区别：

○ 文件尺寸

V2 最大只支持 32BIT 的文件大小,而 NFS V3 新增加了支持 64BIT 文件大小的技术。

○ 文件传输尺寸

V3 没有限定传输尺寸，V2 最多只能设定为 8k，可以使用-rsize and -wsize 来进行设定。

○ 完整的信息返回

V3 增加和完善了许多错误和成功信息的返回，对于服务器的设置和管理能带来很大好处。

○ 增加了对 TCP 传输协议的支持

V2 只提供了对 UDP 协议的支持，在一些高要求的网络环境中有很大限制，V3 增加了对 TCP 协议的支持

○ 异步写入特性

● NFS V3 能否使用异步写入，这是可选择的一种特性。

● NFS V3 客户端发发送一个异步写入请求到服务器，在给客户端答复之前服务器并不是必须要将数据写入到存储器中（稳定的）。

● 服务器能确定何时去写入数据或者将多个写入请求聚合到一起并加以处理，然后写入。客户端能保持一个数据的 copy 以防万一服务器不能完整的将数据写入。

● 当客户端希望释放这个 copy 的时候，它会向服务器通过这个操作过程，以确保每个操作步骤的完整。

● 异步写入能够使服务器确定最好的同步数据的策略使数据尽可能的同步的提交。

● 与 V2 比较来看，这样的机制能更好的实现数据缓冲和更多的平行（平衡）。而 NFS V2 的 SERVER 在将数据写入存储器之前不能再相应任何的写入请求。

○ 改进了 SERVER 的 mount 性能

○ 有更好的 I/O WRITES 性能。

○ 更强网络运行效能，使得网络运作更为有效。

○ 更强的灾难恢复功能。

□ V4 相对 V3 的改进：

○ 改进了 INTERNET 上的存取和执行效能

○ 在协议中增强了安全方面的特性

○ 增强的跨平台特性

△ RPC(Remote Procedure Call Protocol，远程过程调用)

□ RPC（Remote Procedure Call Protocol）——远程过程调用协议，它是一种通过网络从远程计算机程序上请求服务，而不需要了解底层网络技术的协议。RPC 协议假定某些传输协议的存在，如 TCP 或 UDP，为通信程序之间携带信息数据。在 OSI 网络通信模型中，RPC 跨越了传输层和应用层。RPC 使得开发包括网络分布式多程序在内的应用程序更加容易。

□ RPC 采用客户机/服务器模式。请求程序就是一个客户机，而服务提供程序就是一个服务器。首先，客户机调用进程发送一个有进程参数的调用信息到服务进程，然后等待应答信息。在服务器端，进程保持

睡眠状态直到调用信息到达为止。当一个调用信息到达，服务器获得进程参数，计算结果，发送答复信息，然后等待下一个调用信息，最后，客户端调用进程接收答复信息，获得进程结果，然后调用执行继续进行。

- □ NFS 协议本身没有网络传输功能，而是基于 RPC 实现的。
- □ RPC 提供了一个面向过程的远程服务结构。
- □ RPC 可以通过网络从远程主机上请求服务，而不需要了解底层网络技术的协议。
- □ RPC 是工作在 OSI 模型的会话层，它可以为遵循 RPC 协议应用层协议提供端口注册功能。
- □ 很多服务都可以向 RPC 注册端口。
- □ RPC 使用网络端口 111 来监听客户端的请求。

△ NFS 的守护进程

- □ NFS 的不同功能由不同的守护进程提供。
- □ NFS 的每个功能都有 RPC 分配分配的端口监听。
  - ○ rpc.nfsd：基本的 NFS 守护进程（2049 端口），主要负责登录权限检测。
  - ○ rpc.mountd：负责管理 NFS 的文件系统，对客户端存取服务器的文件进行一系列的管理。
  - ○ rpc.rquotad：提供远程磁盘限额服务。

- ○ rpc.lockd：用于管理文件的锁定，防止多个客户端同时写入某个文件时产生的冲突。
- ○ rpc.statd：用来检查共享目录的一致性

△ NFS 相关工具

- □ exportfs：NFS 服务器端功能，维护共享资源
- □ showmount：NFS 客户端功能，查看服务器共享的目录。
- □ nfsstat：显示 NFS 的状态统计信息。
- □ rpcinfo：显示由 RPC 维护的端口映射，显示已注册的 RPC 服务列表

△ NFS 的安装与启动

- □ sudo apt-get install nfs-common nfs-kernel-server
- □ sudo service nfs-kernel-server restart

△ NFS 服务端

- □ 主要配置文件/etc/exports
  - ○ 共享目录 [主机 1(参数项)] [主机 2(参数项)] ...
  - ○ 主机：ip 地址或主机名

○ 参数项：

- ro　　设置共享目录为只读
- rw　　设置共享目录可读写
- sync　　所有数据在请求时写入共享
- async　　NFS 在写入数据前可以相应请求
- secure　　NFS 通过 1024 以下的安全 TCP/IP 端口发送
- insecure　　NFS 通过 1024 以上的端口发送
- wdelay　　如果多个用户写入 NFS 目录，则归组写入（默认）
- no_wdelay　　如果多个用户写入 NFS 目录，则立即写入，使用 async 时无需此设置。
- hide　　在 NFS 共享目录中不共享其子目录
- no_hide　　共享 NFS 目录的子目录
- subtree_check　　如果共享/usr/bin 之类的子目录时，强制 NFS 检查父目录的权限（默认）
- no_subtree_check　　和上面相对，不检查父目录权限
- all_squash　　共享文件的 UID 和 GID 映射匿名用户 anonymous，适合公用目录。

- no_all_squash　　保留共享文件的 UID 和 GID（默认）
- root_squash　　root 用户的所有请求映射成如 anonymous 用户一样的权限（默认）
- no_root_squash　　root 用户具有根目录的完全管理访问权限
- anonuid=xxx　　指定 NFS 服务器/etc/passwd 文件中匿名用户的 UID
- anongid=xxx　　指定 NFS 服务器/etc/passwd 文件中匿名用户的 GID

□ exportfs 命令

○ 用于维护 NFS 共享的目录列表

○ 当修改了/etc/exports 之后，无需重启 nfs 服务

○ exportfs [-aruv]

- -a:全部挂载或卸载配置文件中的设置
- -r:重新挂载配置文件中的设置。
- -u:卸载共享目录
- -v:显示输出列表同时，显示设定参数。

△ NFS 客户端

□ showmount 命令

○ 查看 NFS 服务器上所有共享目录 showmount －e ip_address

○ 查看服务器上哪些共享目录被挂载 showmount －d ip_address

□ NFS 的挂载与卸载

○ 挂载 mount －t nfs [-o 参数] server_adr:/共享目录 /本机挂载点

○ 卸载 umount /本机挂载点 (加选项-f 进行强制卸载)

□ 启动时挂载

○ 修改/etc/fstab 文件 192.168.0.200:/share /share nfs hard,intr 0 0

△ 更多的关于 NFS 的相关内容请参见：http://linux.vbird.org/linux_server/0330nfs.php#What_NFS_NFS

15、内核级虚拟化技术

内核级虚拟化技术：Kernel-based Virtual Machine，简称 KVM

△ KVM 的特性

□ 嵌入到 Linux 正式 Kernel（提高兼容性）

- □ 代码级资源调用（提高性能）
- □ 虚拟机就是一个进程（内存易于管理）
- □ 直接支持 NUMA 技术（提高扩展性）

△ KVM+Qemu+Libvirt 实践

- □ KVM 支持检测及模块安装
  - ○ CPU 检测
    - ● sudo kvm-ok
    - ● egrep －c ‘(vmx|svm)’ /proc/cpuinfo
  - ○ 模块安装
    - ● sudo apt-get install kvm cpu-checker
  - ○ 模块启动检测
    - ● lsmod | grep kvm
- □ 虚拟磁盘管理
  - ○ 软件包安装

- sudo apt-get install qemu-utils

- 创建虚拟磁盘
  - sudo qemu-img create -f {type} {VM_FILE} {size}
- 虚拟磁盘格式转换
  - sudo qemu-img convert -f {type1} {VM_FILE1} -O {type2} {VM_FILE2}
- 虚拟磁盘检查
  - sudo qemu-img info {VM_FILE}
- 虚拟磁盘空间调整
  - sudo qemu-img resize -q {VM_FILE} {NEW_SIZE}
- 查看帮助
  - qemu-img --help

- libvirtd 配置
  - 安装软件包

    sudo apt-get install virtinst

○ sudo vi /etc/libvirt/libvirtd.conf，输入以下内容

listen_tls = 0

listen_tcp = 1

tcp_port = "16509”

auth_tcp = "none”

mdns_adv = 0

○ sudo vi /etc/default/libvirt-bin，修改如下内容

libvirtd_opts=“-d –l”

○ sudo vi /etc/libvirt/qemu.conf，修改如下

vnc_listen = “0.0.0.0”

○ 重启 libvirt 服务

sudo service libvirt-bin restart

□ 创建虚拟网卡

○ virsh iface-bridge eth0 virbr0

□ 创建及克隆虚拟机

○ 创建虚拟机 virt-install，在/etc/libvirt/qemu 目录下会生成配置文件。

○ 创建命令如下，亦可以将该命令写到脚本里进行执行，在写脚本是直接将下面的内容复制粘贴即可，但是在命令行中直接执行的时候需要注意每一行结尾的续行符“ \”。

```
sudo virt-install
--connect qemu:///system        #固定写法
--name=VMNAME       #要创建的虚拟机的名字
--ram 2048          #设置虚拟机的内存大小
--vcpus=2           #设置虚拟机的核数
--disk path=/xxxx/us.qcow2,format=qcow2,size={SIZE},bus=virtio      #设置安装虚拟机的虚拟磁盘的路径
--cdrom /xxx/ubuntu-14.04.3-server-amd64.iso        #设置镜像文件路径
--vnc               #设置是否支持 VNC 模式，如果是按照此种方法安装虚拟机的话，最好设置，因为在安装的时候有一些步骤是需要人为手动干预的，如果不通过 VNC 是没有办法看到安装界面的
```

--os-type linux　　#设置安装的虚拟机的系统类型

--accelerate　　#设置是否加速安装

--hvm

--network bridge=virbr0,model=virtio　　#设置网卡信息

--noautoconsole　　#设置不输出到控制台

- ○ 克隆虚拟机 virt-clone －o VM1 －n VM_NEW －f VM_NEW_DISK_PATH
- ○ 修改配置文件 virsh edit {VM}

□ 虚拟机管理

- ○ 启动虚拟机 virsh start VM
- ○ 关闭虚拟机 virsh shutdown VM
- ○ 强制关闭虚拟机 virsh destroy VM
- ○ 删除虚拟机 virsh undefine VM
- ○ 查看虚拟机 virsh list --all
- ○ 迁移虚拟机 virsh migrate --live VM qemu+tcp://ip/system --unsafe

□ 虚拟机域管理

○ 查询虚拟机 domain 连接信息 virsh domdisplay VM

○ 查询虚拟机磁盘信息 virsh domblklist VM

○ 查询虚拟机网卡 virsh domiflist VM

□ 虚拟磁盘管理

○ 添加虚拟磁盘 virsh attach-disk –domain {VM} –source {XXX.qcow2} --target {vdX} --persistent --subdriver=qcow2

○ 删除虚拟磁盘 virsh detach-disk –domain {VM} –target {vdX} –persistent

□ 当管理虚拟机是，可以使用 virsh 命令进入 virsh 控制台进行管理，而不必每次都在命令之前输入 virsh。