# Linux tar打包命令

范例一：将整个 /etc 目录下的文件全部打包成为 /tmp/etc.tar  
[root@linux ~]# tar -cvf /tmp/etc.tar /etc <==仅打包，不压缩！  
[root@linux ~]# tar -zcvf /tmp/etc.tar.gz /etc <==打包后，以 gzip 压缩  
[root@linux ~]# tar -jcvf /tmp/etc.tar.bz2 /etc <==打包后，以 bzip2 压缩  
# 特别注意，在参数 f 之后的文件档名是自己取的，我们习惯上都用 .tar 来作为辨识。  
# 如果加 z 参数，则以 .tar.gz 或 .tgz 来代表 gzip 压缩过的 tar file ～  
# 如果加 j 参数，则以 .tar.bz2 来作为附档名啊～  
# 上述指令在执行的时候，会显示一个警告讯息：  
# 『tar: Removing leading `/” from member names』那是关於绝对路径的特殊设定。

范例二：查阅上述 /tmp/etc.tar.gz 文件内有哪些文件？  
[root@linux ~]# tar -ztvf /tmp/etc.tar.gz  
# 由於我们使用 gzip 压缩，所以要查阅该 tar file 内的文件时，  
# 就得要加上 z 这个参数了！这很重要的！

范例三：将 /tmp/etc.tar.gz 文件解压缩在 /usr/local/src 底下  
[root@linux ~]# cd /usr/local/src  
[root@linux src]# tar -zxvf /tmp/etc.tar.gz  
# 在预设的情况下，我们可以将压缩档在任何地方解开的！以这个范例来说，  
# 我先将工作目录变换到 /usr/local/src 底下，并且解开 /tmp/etc.tar.gz ，  
# 则解开的目录会在 /usr/local/src/etc 呢！另外，如果您进入 /usr/local/src/etc  
# 则会发现，该目录下的文件属性与 /etc/ 可能会有所不同喔！

范例四：在 /tmp 底下，我只想要将 /tmp/etc.tar.gz 内的 etc/passwd 解开而已  
[root@linux ~]# cd /tmp  
[root@linux tmp]# tar -zxvf /tmp/etc.tar.gz etc/passwd  
# 我可以透过 tar -ztvf 来查阅 tarfile 内的文件名称，如果单只要一个文件，  
# 就可以透过这个方式来下达！注意到！ etc.tar.gz 内的根目录 / 是被拿掉了！

范例五：将 /etc/ 内的所有文件备份下来，并且保存其权限！  
[root@linux ~]# tar -zxvpf /tmp/etc.tar.gz /etc  
# 这个 -p 的属性是很重要的，尤其是当您要保留原本文件的属性时！

范例六：在 /home 当中，比 2005/06/01 新的文件才备份  
[root@linux ~]# tar -N ‘2005/06/01′ -zcvf home.tar.gz /home

范例七：我要备份 /home, /etc ，但不要 /home/dmtsai  
[root@linux ~]# tar –exclude /home/dmtsai -zcvf myfile.tar.gz /home/\* /etc

范例八：将 /etc/ 打包后直接解开在 /tmp 底下，而不产生文件！  
[root@linux ~]# cd /tmp  
[root@linux tmp]# tar -cvf – /etc | tar -xvf –  
# 这个动作有点像是 cp -r /etc /tmp 啦～依旧是有其有用途的！  
# 要注意的地方在於输出档变成 – 而输入档也变成 – ，又有一个 | 存在～  
# 这分别代表 standard output, standard input 与管线命令啦！  
# 这部分我们会在 Bash shell 时，再次提到这个指令跟大家再解释啰！

.tar  
　　解包：tar xvf FileName.tar  
　　打包：tar cvf FileName.tar DirName  
　　（注：tar是打包，不是压缩！）  
　　———————————————  
　　.gz  
　　解压1：gunzip FileName.gz  
　　解压2：gzip -d FileName.gz  
　　压缩：gzip FileName  
　　.tar.gz 和 .tgz  
　　解压：tar zxvf FileName.tar.gz  
　　压缩：tar zcvf FileName.tar.gz DirName  
　　———————————————  
　　.bz2  
　　解压1：bzip2 -d FileName.bz2  
　　解压2：bunzip2 FileName.bz2  
　　压缩： bzip2 -z FileName  
　　.tar.bz2  
　　解压：tar jxvf FileName.tar.bz2 或tar –bzip xvf FileName.tar.bz2  
　　压缩：tar jcvf FileName.tar.bz2 DirName  
　　———————————————  
　　.bz  
　　解压1：bzip2 -d FileName.bz  
　　解压2：bunzip2 FileName.bz  
　　压缩：未知  
　　.tar.bz  
　　解压：tar jxvf FileName.tar.bz  
　　压缩：未知  
　　———————————————  
　　.Z  
　　解压：uncompress FileName.Z  
　　压缩：compress FileName  
　　.tar.Z  
　　解压：tar Zxvf FileName.tar.Z  
　　压缩：tar Zcvf FileName.tar.Z DirName  
　　———————————————  
　　.zip  
　　解压：unzip FileName.zip  
　　压缩：zip FileName.zip DirName  
　　压缩一个目录使用 -r 参数，-r 递归。例： $ zip -r FileName.zip DirName  
　　———————————————  
　　.rar  
　　解压：rar x FileName.rar  
　　压缩：rar a FileName.rar DirName  
　　rar请到：<http://www.rarsoft.com/download.htm> [下载](http://www.2cto.com/soft" \t "_blank)！  
　　解压后请将rar\_static拷贝到/usr/bin目录（其他由$PATH环境变量指定的目录也可以）：

摘自：webshell blog

VIM编辑器的使用（大小写区分）

在命令行下输入切换到打开VIM，按下i进入编辑模式，开始编辑文本，按下esc进入命令模式。

初步使用：

        打开在图形界面下打开VIM，按下i建，进入编辑模式，编辑文字，按下esc建进入命令模式，按下:wq保存并且退出，注意命令此命令有冒号。

继续使用：

    Vim 的模式有3种，分为一般模式，编辑模式，命令行切换模式。

（1）：一般模式下的按键说明：

           h 建向左移动

           j 向下移动

           k 向上移动

           l 向右移动

          CTRL f 向下移动一页

          CTRL b向上移动一页

          CTRL d 向下移动半页

          CTRL u 向上移动半页

          + 光标移动到非空格的下一行

          -光标移动到非空格的上一行

          n <space> n表示数字，加上空格，向当前移动多少个字符。

**0 或者shift+6（^）移动到改行的开始处**

**shift+4（$）移动到行尾**

          H 移动到上行的第一个字符，

          M 移动到中间那行的第一个字符。

          L 移动到屏幕最下方那一行的第一个字符。

          G 移动到文件最后一行（常用）

          nG移动到文件第几行

          gg移动到这个文件的第一行。

          n<enter> 光标移动到这个文件的第一行

          /word 查找文件为WOED 的字符。向下

          ?word 向上寻找一个味WORD 的字符。

          n 此按键位英文按键，表示继续重复前一个查找动作，

          N 此按键为英文按键，表示继续相反地方向重复前一个查找动作。

          :n1,n2s/word1/word2/g表示n1-n2行的word1被替换为word2

          :n1,$s/word1/word2/g 便是从n1-末行的word1被替换为word2

          :n1,$s/word1/word2/gc 功能同上给予确认的替换。

          x 删除光标所在的的后面一个字符。相当于DEL键

          nx 此处n为数字，表示删除n个字符。

          dd 删除光标所在行

          ndd 删除光标所在行下面n行

          d1G删除光标所在行到第一行的所有数据。

          dG 删除光标所在行到最后一行的所有数据。

          d$ 删除光标所在处到该行的所有数据。

          d0 删除光标所在行到第一行的所有数据

         c重复删除多个数据，

         yy 复制光标所在那一行的所有数据

         nyy 复制光标所在的所在的向下n行，n为数字

         y1G 复制光标所在行到第一行的所有数据。

         yG 复制光标所在行到最后一行的搜有数据。

         y0 复制光标处到第一行的所有数据。

         y$ 复制光标到末行的所有数据。

         p 将复制的数据向下面一行粘贴

         P将复制的数据向上面粘贴

         J 将光标所在行与下一行结合成一行。

         u 复原上一个操作。

         Ctrl +r 重复上面一个操作。

（1）：编辑模式：

             进入到编辑模式的常用命令。

             i 从目前光标所在处插入。

             I 从目前光标所在处第一个非空格开始插入。

             a 从光标所在处的下一个字符开始插入。

             A 从光标所在处行的最后一个字符卡是插入

             o 从目前光标所在处的下一行插入新的一行。

             O 为在目前光标所在处的上面一行插入新的一行。

             r 进入一次替换插入，即在目前光标的下一行插入新的一行。

             R 一直替换插入，直到按下esc为止。

   （2）：命令模式

           :wq保存文本，并且退出，如果文件没有指定文件名字，则需要:wq 文件名，

             :w 写入

             :wq! 强制写入。

             :set nu 显示行号

             :set nonu 取消行号

             :n1,n2 w <filename> 将n1,n2的内容保存到filename 这个文件。

             :r <filename> 将filename 这个文件的内容读到当前这个光标所在行的最后。

             :! Commad command 为命令即离开暂时离开当前进入命令模式。

                                                                                            VIM使用的注意事项

              如果你正在使用vim编辑的时候，vim会在被编辑的的文件下新建一个名字为filename.swp的文件，当正在执行编辑过程发生了VIM不正常结束 时候，再次打开被编辑的文件就会出现警告信息，其中O为打开只读，E以正常方式打开，R加载缓存文件打开，D删除缓存文件Q离开vim,A忽略编辑行为，

                                                                                            Vim 的功能

       目前的distributions都以vim代替vi的功能了。终端输入alias可以找到alias  vi=’vim’，如果没有则需要自己设置，

                                                                                                   块选择

            按下V 则可以使光标所在行进行翻白操作，ctrl +v 对块进行翻白操作，使用y 或者d操作可以对翻白的进行操作，可以整行整列的复制。

          :n编辑下一个文件，

         :N 编辑上一个文件。

         :files 列出这个vime的打开的所有文件。

                                                                               Vim的多窗口功能

           如果文件很大，如果使用CTRL+F或者使用CTR+M翻阅文件也很麻烦，可以将一个文件切割为多个窗口的展现，其命令为:sp +filename其中如果只是为:sp则只是分割当前文件，如果是:sp+filename则分割多个文件，多窗口之间的移动，CTRL +W + j向下移动，ctrl +w +k向上移动，ctrl +w +q离开当前窗口。

第一次用windows的edit时赶脚edit的功能好强大，结果现在发现vim 更是碉堡了。。。。­­­­­­­­­

                                                                      Vim 的环境配置

:set nu            设置行号

:set nonu          取消行号

:set hlsearch       设置高亮查找

:set nohlsearch      取消高亮查找

:set backup        设置备份

:set ruler           对于状态栏的说明

:set showmode      是否显示左下角的状态栏

:set all             显示目前所有环境参数设置值

:set               显示与系统默认值不同的设置参数

:syntax on          是否依据程序相关语法显示不同颜色，开

:syntax off          是否一句成语相关语法显示不同颜色，关

:set bg=dark       显示不同的颜色色调

:set bg=light        显示不同的颜色色调

:set autoindent    自动缩进

对于这些设置，如果你想要默认设置好，这需要在/etc/vimrc中修改，但是听说建议不要字节修改这个文件，修改~/.vimrc，默认不存在，自己创建，

# ~用法

~代表你的/home/用户明目录  
假设你的用户名是x，那么~/就是/home/x/  
.是代表此目录本身，但是一般可以不写  
所以cd ~/. 和cd ~ 和cd ~/效果是一样的  
但是.后面有东西又是另外一个问题，点在文件名头部，代表一个隐藏文件  
~/.local是你的主目录下一个.local的文件夹的路径，并且从.可以看出，这是一个饮藏文件，如果不用ls -a的话，一般ls是无法看到的

# clear命令用于Terminal清屏

# 查询python目录

type python

# 为所有人添加\*\*\*的可执行权限

chmod a+x \*\*\*.py

（在首行加入#!usr/bin/python这样就可以直接使用 ./\*\*\*.py直接执行程序）

# 汉字编码

服务器端使用

#encoding:utf-8解码（注意不要含空格）

客户端使用

#coding:utf-8编码

shell进程是执行其它进程的进程，但在执行一个命令时，shell进程会指示内核把指定的程序作为另一个进程分开执行，并将输出写到终端。

通过命令选项，我们可以修改命令执行的行为。命令行选项可以分为短命令行选项和长命令行选项两种。短命令行选项是由字母组成，长命令行选项是由单词组成。短命令行选项在选项前使用单横杠“-”，长命令行选项前使用双横杠“--”。如果选项后面需要输入选项的参数，短命令行选项和参数之间使用空格分隔，而长命令行选项使用等号“=”连接选项和参数。

通过在命令后面添加选项“--help”、“-h”或“-？”，我们可以获得一个命令的帮助信息。