

## 06.Linux 檔案結構常用指令介紹

### 重點整理

1. Linux的每個檔案中，依據權限分為使用者、群組與其他人三種身份
2. 群組最有用的功能之一，就是當你在團隊開發資源的時候，且每個帳號都可以有多個群組的支援
3. 利用ls -l顯示的檔案屬性中，第一個欄位是檔案的權限，共有十個位元，第一個位元是檔案類型，接下來三個為一組共三組，為使用者、群組、其他人的權限，權限有r,w,x三種
4. 如果檔名之前多一個『.』，則代表這個檔案為『隱藏檔』
5. 更改檔案的群組支援可用chgrp，修改檔案的擁有者可用chown，修改檔案的權限可用chmod
6. chmod修改權限的方法有兩種，分別是符號法與數字法，數字法中r,w,x分數為4,2,1
7. 對檔案來講，權限的效能為：  
r：可讀取此一檔案的實際內容，如讀取文字檔的文字內容等；  
w：可以編輯、新增或者是修改該檔案的內容(但不含刪除該檔案)；  
x：該檔案具有可以被系統執行的權限。
8. 要開放目錄給任何人瀏覽時，應該至少也要給予r及x的權限，但w權限不可隨便給

### 檔案指令

例題：

若有一個檔案的類型與權限資料為『-rwxr-xr--』，請說明其意義為何？

答：

先將整個類型與權限資料分開查閱，並將十個字元整理成為如下所示：

```
[-][rwx][r-x][r--]  
1 234 567 890
```

1 為：代表這個檔名為目錄或檔案，本例中為檔案(-)；  
234為：擁有者的權限，本例中為可讀、可寫、可執行(rwx)；  
567為：同群組使用者權限，本例中為可讀可執行(rx)；  
890為：其他使用者權限，本例中為可讀(r)

同時注意到，rwx所在的位置是不會改變的，有該權限就會顯示字元，沒有該權限就變成減號(-)就是了。

例題：

假設test1, test2, test3同屬於testgroup這個群組，如果有下面的兩個檔案，請說明兩個檔案的擁有者與其相關的權限為何？

```
-rw-r--r-- 1 root    root      238 Jun 18 17:22 test.txt  
-rwxr-xr-- 1 test1   testgroup 5238 Jun 19 10:25 ping_tsai
```

答：

- 檔案test.txt的擁有者為root，所屬群組為root。至於權限方面則只有root這個帳號可以存取此檔案，其他人則僅能讀此檔案；
- 另一個檔案ping\_tsai的擁有者為test1，而所屬群組為testgroup。其中：
  - test1 可以針對此檔案具有可讀可寫可執行的權力；
  - 而同群組的test2, test3兩個人與test1同樣是testgroup的群組帳號，則僅可讀可執行但不能寫(亦即不能修改)；
  - 至於非testgroup這一個群組的人則僅可以讀，不能寫也不能執行！

例題：

如果我的目錄為底下的樣式，請問testgroup這個群組的成員與其他人(others)是否可以進入本目錄？

```
drwxr-xr-- 1 test1  testgroup 5238 Jun 19 10:25 groups/
```

答：

- 檔案擁有者test1[rwx]可以在本目錄中進行任何工作；
- 而testgroup這個群組[r-x]的帳號，例如test2, test3亦可以進入本目錄進行工作，但是不能在本目錄下進行寫入的動作；
- 至於others的權限中[r--]雖然有r，但是由於沒有x的權限，因此others的使用者，並不能進入本目錄。

例題：

如果我的目錄為底下的樣式，請問testgroup這個群組的成員與其他人(others)是否可以進入本目錄？

```
drwxr-xr-- 1 test1 testgroup 5238 Jun 19 10:25 groups/
```

答：

- 檔案擁有者test1[rwx]可以在本目錄中進行任何工作；
- 而testgroup這個群組[r-x]的帳號，例如test2, test3亦可以進入本目錄進行工作，但是不能在本目錄下進行寫入的動作；
- 至於others的權限中[r--]雖然有r，但是由於沒有x的權限，因此others的使用者，並不能進入此目錄！

## chgrp：改變檔案所屬群組

```
[root@www ~]# chgrp [-R] dirname/filename ...
選項與參數：
-R：進行遞迴(recursive)的持續變更，亦即連同次目錄下的所有檔案、目錄
    都更新成為這個群組之意。常常用在變更某一目錄內所有的檔案之情況。
範例：
[root@www ~]# chgrp users install.log
[root@www ~]# ls -l
-rw-r--r-- 1 root users 68495 Jun 25 08:53 install.log
[root@www ~]# chgrp testing install.log
chgrp: invalid group name `testing' <== 發生錯誤訊息囉～找不到這個群組名～
```

## chown：改變檔案擁有者

```
[root@www ~]# chown [-R] 帳號名稱 檔案或目錄
[root@www ~]# chown [-R] 帳號名稱:群組名稱 檔案或目錄
選項與參數：
-R：進行遞迴(recursive)的持續變更，亦即連同次目錄下的所有檔案都變更
範例：將install.log的擁有者改為bin這個帳號：
[root@www ~]# chown bin install.log
[root@www ~]# ls -l
-rw-r--r-- 1 bin users 68495 Jun 25 08:53 install.log
範例：將install.log的擁有者與群組改回為root：
[root@www ~]# chown root:root install.log
[root@www ~]# ls -l
-rw-r--r-- 1 root root 68495 Jun 25 08:53 install.log
```

何候要使用chown或chgrp呢？最常見的例子就是在複製檔案給你之外的其他人時，我們使用最簡單的cp指令來說明好了

```
[root@www ~]# cp .bashrc .bashrc_test
[root@www ~]# ls -al .bashrc*
-rw-r--r-- 1 root root 395 Jul  4 11:45 .bashrc
-rw-r--r-- 1 root root 395 Jul 13 11:31 .bashrc_test <==新檔案的屬性沒變
```

由於複製行為(cp)會複製執行者的屬性與權限，.bashrc\_test還是屬於root所擁有，如此一來，即使你將檔案拿給bin這個使用者了，那他仍然無法修改的(看屬性/權限就知道了吧)，所以你就必須要將這個檔案的擁有者與群組修改。**請將.bashrc\_test擁有者與群組皆改成owner**

## chmod：改變檔案的權限, SUID, SGID, SBIT等等的特性

```
[root@www ~]# chmod [-R] xyz 檔案或目錄
選項與參數：
xyz：就是剛剛提到的數字類型的權限屬性，為 rwx 屬性數值的相加。
-R：進行遞迴(recursive)的持續變更，亦即連同次目錄下的所有檔案都會變更
[root@www ~]# ls -al .bashrc
-rw-r--r-- 1 root root 395 Jul  4 11:45 .bashrc
[root@www ~]# chmod 777 .bashrc
[root@www ~]# ls -al .bashrc
-rwxrwxrwx 1 root root 395 Jul  4 11:45 .bashrc
```

練習: 將剛剛.bashrc這個檔案的權限修改回-rw-r--r-- 的狀態！

例題：

有個目錄的權限如下所示：

```
drwxr--r-- 3 root root 4096 Jun 25 08:35 .ssh
```

系統有個帳號名稱為vbird，這個帳號並沒有支援root群組，請問vbird對這個目錄有何權限？是否可切換到此目錄中？

答：

vbird對此目錄僅具有r的權限，因此vbird可以查詢此目錄下的檔名列表。因為vbird不具有x的權限，因此vbird並不能切換到此目錄內！（相當重要的概念！）

上面這個例題中因為vbird具有r的權限，因為是r乍看之下好像就具有可以進入此目錄的權限，其實那是錯的。能不能進入某一個目錄，只與該目錄的x權限有關啦！此外，工作目錄對於指令的執行是非常重要的，如果你在某目錄下不具有x的權限，那麼你就無法切換到該目錄下，也就無法執行該目錄下的任何指令，即使你具有該目錄的r的權限。

很多朋友在架設網站的時候都會卡在一些權限的設定上，他們開放目錄資料給網際網路的任何人來瀏覽，卻只開放r的權限，如上面的範例所示那樣，那樣的結果就是導致網站伺服器軟體無法到該目錄下讀取檔案(最多只能看到檔名)，最終用戶總是無法正確的查閱到檔案的內容(顯示權限不足啊！)。要注意：要開放目錄給任何人瀏覽時，應該至少也要給予r及x的權限，但w權限不可隨便給！為什麼w不能隨便給，我們來看下一個例子：

例題：

假設有個帳號名稱為dmtsai，他的家目錄在/home/dmtsai/，dmtsai對此目錄具有[rwx]的權限。若在此目錄下有個名為the\_root.data的檔案，該檔案的權限如下：

```
-rwx----- 1 root root 4365 Sep 19 23:20 the_root.data
```

請問dmtsai對此檔案的權限為何？可否刪除此檔案？

答：

如上所示，由於dmtsai對此檔案來說是『others』的身份，因此這個檔案他無法讀、無法編輯也無法執行，也就是說，他無法變動這個檔案的內容就是了。

但是由於這個檔案在他的家目錄下，他在此目錄下具有rwx的完整權限，因此對於the\_root.data這個『檔名』來說，他是能夠『刪除』的！結論就是，dmtsai這個用戶能夠刪除the\_root.data這個檔案！

先用root的身份建立所需要的檔案與目錄環境

```
[root@www ~]# cd /tmp <==切換工作目錄到 /tmp
[root@www tmp]# mkdir testing <==建立新目錄
[root@www tmp]# chmod 744 testing <==變更權限
[root@www tmp]# touch testing/testing <==建立空的檔案
[root@www tmp]# chmod 600 testing/testing <==變更權限
[root@www tmp]# ls -ald testing testing/testing
drwxr--r-- 2 root root 4096 Sep 19 16:01 testing
-rw----- 1 root root 0 Sep 19 16:01 testing/testing
# 仔細看一下，目錄的權限是 744，且所屬群組與使用者均是 root 喔！
# 那麼在這樣的情況底下，一般身份使用者對這個目錄/檔案的權限為何？
```

一般用戶的讀寫權限為何？觀察中

```
[root@www tmp]# su - vbird <==切換身份成為 vbird 囉！
[vbird@www ~]$ cd /tmp <==看一下，身份變了喔！提示字元也變成 $ 了！
[vbird@www tmp]$ ls -l testing/
?----- ? ? ? ? ? testing
# 因為具有 r 的權限可以查詢檔名。不過權限不足(沒有x)，所以會有一堆問號。
[vbird@www tmp]$ cd testing/
-bash: cd: testing/: Permission denied
# 因為不具有 x，所以當然沒有進入的權限啦！有沒有呼應前面的權限說明啊！
```

如果該目錄屬於用戶本身，會有什麼狀況

```
[vbird@www tmp]$ exit <==讓 vbird 變回原本的 root 身份喔！
[root@www tmp]# chown vbird testing <==修改權限，讓vbird擁有此目錄
[root@www tmp]# su - vbird <==再次變成vbird來操作
[vbird@www ~]$ cd /tmp/testing <==可以進入目錄了呢！
[vbird@www testing]$ ls -l
-rw----- 1 root root 0 Sep 19 16:01 testing <==檔案不是vbird的！
[vbird@www testing]$ rm testing <==嘗試殺掉這個檔案看看！
rm: remove write-protected regular empty file `testing'? y
# 竟然可以刪除！這樣理解了嗎？！
```

## 目錄指令

cd：變換目錄 (Change Directory 縮寫)

- 代表此層目錄
- .. 代表上一層目錄
- 代表前一個工作目錄
- ~ 代表『目前使用者身份』所在的家目錄
- ~account 代表 account 這個使用者的家目錄(account是個帳號名稱)

```
[root@www ~]# cd [相對路徑或絕對路徑]
# 最重要的就是目錄的絕對路徑與相對路徑，還有一些特殊目錄的符號囉！
[root@www ~]# cd ~vbird
# 代表去到 vbird 這個使用者的家目錄，亦即 /home/vbird
[root@www vbird]# cd ~
# 表示回到自己的家目錄，亦即是 /root 這個目錄
[root@www ~]# cd
# 沒有加上任何路徑，也還是代表回到自己家目錄的意思喔！
[root@www ~]# cd ..
# 表示去到目前的上層目錄，亦即是 /root 的上層目錄的意思；
[root@www /]# cd -
# 表示回到剛剛的那個目錄，也就是 /root 囉~
[root@www ~]# cd /var/spool/mail
# 這個就是絕對路徑的寫法！直接指定要去的完整路徑名稱！
[root@www mail]# cd ../mqueue
# 這個是相對路徑的寫法，我們由/var/spool/mail 去到/var/spool/mqueue 就這樣寫！
```

pwd：顯示目前的目錄

mkdir：建立一個新的目錄

```
[root@www ~]# mkdir [-mp] 目錄名稱
選項與參數：
-m：設定檔案的權限喔！直接設定，不需要看預設權限 (umask) 的臉色~
-p：幫助你直接將所需要的目錄(包含上層目錄)遞迴建立起來！

範例：請到 /tmp 底下嘗試建立數個新目錄看看：
[root@www ~]# cd /tmp
[root@www tmp]# mkdir test <==建立一名為 test 的新目錄
[root@www tmp]# mkdir test1/test2/test3/test4
mkdir: cannot create directory `test1/test2/test3/test4':
No such file or directory <== 沒辦法直接建立此目錄啊！
[root@www tmp]# mkdir -p test1/test2/test3/test4
# 加了這個 -p 的選項，可以自行幫你建立多層目錄！

範例：建立權限為 rwx--x--x 的目錄
[root@www tmp]# mkdir -m 711 test2
[root@www tmp]# ls -l
drwxr-xr-x 3 root root 4096 Jul 18 12:50 test
drwxr-xr-x 3 root root 4096 Jul 18 12:53 test1
drwx--x--x 2 root root 4096 Jul 18 12:54 test2
# 仔細看上面的權限部分，如果沒有加上 -m 來強制設定屬性，系統會使用預設屬性。
# 那麼你的預設屬性為何？這要透過底下介紹的 umask 才能瞭解喔！ ^_^
```

rmdir：刪除一個空的目錄

rm -r test

```
[root@www ~]# rm -rf 目錄名稱
```

選項與參數：

**-p** : 連同上層『空的』目錄也一起刪除

範例：將於mkdir範例中建立的目錄(/tmp底下)刪除掉！

```
[root@www tmp]# ls -l <==看看有多少目錄存在？
drwxr-xr-x 3 root root 4096 Jul 18 12:50 test
drwxr-xr-x 3 root root 4096 Jul 18 12:53 test1
drwx--x--x 2 root root 4096 Jul 18 12:54 test2
[root@www tmp]# rm -rf test <==可直接刪除掉，沒問題
[root@www tmp]# rm -rf test1 <==因為尚有內容，所以無法刪除！
rm: `test1': Directory not empty
[root@www tmp]# rm -rf test1/test2/test3/test4
[root@www tmp]# ls -l <==您看看，底下的輸出中test與test1不見了！
drwx--x--x 2 root root 4096 Jul 18 12:54 test2
# 瞧！利用 -p 這個選項，立刻就可以將 test1/test2/test3/test4 一次刪除～
# 不過要注意的是，這個 rm 僅能『刪除空的目錄』喔！
```

## cp (複製檔案或目錄)

```
[root@www ~]# cp [-adfilprsuv] 來源檔(source) 目標檔(destination)
[root@www ~]# cp [options] source1 source2 source3 .... directory
```

選項與參數：

- a** : 相當於 **-pdr** 的意思，至於 **pdr** 請參考下列說明；(常用)
- d** : 若來源檔為連結檔的屬性(link file)，則複製連結檔屬性而非檔案本身；
- f** : 為強制(force)的意思，若目標檔案已經存在且無法開啟，則刪除後再嘗試一次；
- i** : 若目標檔(destination)已經存在時，在覆蓋時會先詢問動作的進行(常用)
- l** : 進行硬式連結(hard link)的連結檔建立，而非複製檔案本身；
- p** : 連同檔案的屬性一起複製過去，而非使用預設屬性(備份常用)；
- r** : 遞迴持續複製，用於目錄的複製行為；(常用)
- s** : 複製成為符號連結檔 (symbolic link)，亦即『捷徑』檔案；
- u** : 若 destination 比 source 舊才更新 destination ！

最後需要注意的，如果來源檔有兩個以上，則最後一個目的檔一定要是『目錄』才行！

範例一：用root身份，將家目錄下的 .bashrc 複製到 /tmp 下，並更名為 bashrc

```
[root@www ~]# cp ~/.bashrc /tmp/bashrc
[root@www ~]# cp -i ~/.bashrc /tmp/bashrc
cp: overwrite `/tmp/bashrc'? n <==n不覆蓋，y為覆蓋
# 重複作兩次動作，由於 /tmp 底下已經存在 bashrc 了，加上 -i 選項後，
# 則在覆蓋前會詢問使用者是否確定！可以按下 n 或者 y 來二次確認呢！
```

範例二：變換目錄到/tmp，並將/var/log/wtmp複製到/tmp且觀察屬性：

```
[root@www ~]# cd /tmp
[root@www tmp]# cp /var/log/wtmp . <==想要複製到目前的目錄，最後的 . 不要忘
[root@www tmp]# ls -l /var/log/wtmp wtmp
-rw-rw-r-- 1 root utmp 96384 Sep 24 11:54 /var/log/wtmp
-rw-r--r-- 1 root root 96384 Sep 24 14:06 wtmp
# 注意上面的特殊字體，在不加任何選項的情況下，檔案的某些屬性/權限會改變；
# 這是個很重要的特性！要注意喔！還有，連檔案建立的時間也不一樣了！
# 那如果你想要將檔案的所有特性都一起複製過來該怎辦？可以加上 -a 喔！如下所示：
```

```
[root@www tmp]# cp -a /var/log/wtmp wtmp_2
[root@www tmp]# ls -l /var/log/wtmp wtmp_2
-rw-rw-r-- 1 root utmp 96384 Sep 24 11:54 /var/log/wtmp
-rw-rw-r-- 1 root utmp 96384 Sep 24 11:54 wtmp_2
# 瞭了吧！整個資料特性完全一模一樣～！真是了不賴～這就是 -a 的特性！
```

範例三：複製 /etc/ 這個目錄下的所有內容到 /tmp 底下

```
[root@www tmp]# cp /etc/ /tmp
cp: omitting directory `/etc' <== 如果是目錄則不能直接複製，要加上 -r 的選項
[root@www tmp]# cp -r /etc/ /tmp
# 還是要再次的強調喔！ -r 是可以複製目錄，但是，檔案與目錄的權限可能會被改變
# 所以，也可以利用『 cp -a /etc /tmp 』來下達指令喔！尤其是在備份的情況下！
```

範例四：將範例一複製的 bashrc 建立一個連結檔 (symbolic link)

```
[root@www tmp]# ls -l bashrc
-rw-r--r-- 1 root root 176 Sep 24 14:02 bashrc <==先觀察一下檔案情況
[root@www tmp]# cp -s bashrc bashrc_slink
[root@www tmp]# cp -l bashrc bashrc_hlink
[root@www tmp]# ls -l bashrc*
-rw-r--r-- 2 root root 176 Sep 24 14:02 bashrc <==與原始檔案不太一樣了！
-rw-r--r-- 2 root root 176 Sep 24 14:02 bashrc_hlink
lrwxrwxrwx 1 root root 6 Sep 24 14:20 bashrc_slink -> bashrc
```



## rm (移除檔案或目錄)

```
[root@www ~]# rm [-fir] 檔案或目錄
選項與參數：
-f : 就是 force 的意思，忽略不存在的檔案，不會出現警告訊息；
-i : 互動模式，在刪除前會詢問使用者是否動作
-r : 遞迴刪除啊！最常用在目錄的刪除了！這是非常危險的選項！！

範例一：將剛剛在 cp 的範例中建立的 bashrc 刪除掉！
[root@www ~]# cd /tmp
[root@www tmp]# rm -i bashrc
rm: remove regular file `bashrc'? y
# 如果加上 -i 的選項就會主動詢問喔，避免你刪除到錯誤的檔名！

範例二：透過萬用字元*的幫忙，將/tmp底下開頭為bashrc的檔名通通刪除：
[root@www tmp]# rm -i bashrc*
# 注意那個星號，代表的是 0 到無窮多個任意字元喔！很好用的東西！

範例三：將 cp 範例中所建立的 /tmp/etc/ 這個目錄刪除掉！
[root@www tmp]# rmdir /tmp/etc
rmdir: etc: Directory not empty <== 刪不掉啊！因為這不是空的目錄！
[root@www tmp]# rm -r /tmp/etc
rm: descend into directory `/tmp/etc'? y
...(中間省略)....
# 因為身份是 root，預設已經加入了 -i 的選項，所以你要一直按 y 才會刪除！
# 如果不想繼續按 y，可以按下『[ctrl]-c』來結束 rm 的工作。
# 這是一種保護的動作，如果確定要刪除掉此目錄而不要詢問，可以這樣做：
[root@www tmp]# \rm -r /tmp/etc
# 在指令前加上反斜線，可以忽略掉 alias 的指定選項喔！至於 alias 我們在bash再談！

範例四：刪除一個帶有 - 開頭的檔案
[root@www tmp]# touch ./-aaa- <==touch這個指令可以建立空檔案！
[root@www tmp]# ls -l
-rw-r--r-- 1 root root 0 Sep 24 15:03 -aaa- <==檔案大小為0，所以是空檔案
[root@www tmp]# rm -aaa-
Try `rm --help' for more information. <== 因為 "-" 是選項嘛！所以系統誤判了！
[root@www tmp]# rm ./-aaa-
```

## mv (移動檔案與目錄，或更名)

```
[root@www ~]# mv [-fiu] source destination
[root@www ~]# mv [options] source1 source2 source3 .... directory
選項與參數：
-f : force 強制的意思，如果目標檔案已經存在，不會詢問而直接覆蓋；
-i : 若目標檔案 (destination) 已經存在時，就會詢問是否覆蓋！
-u : 若目標檔案已經存在，且 source 比較新，才會更新 (update)

範例一：複製一檔案，建立一目錄，將檔案移動到目錄中
[root@www ~]# cd /tmp
[root@www tmp]# cp ~/.bashrc bashrc
[root@www tmp]# mkdir mvtest
[root@www tmp]# mv bashrc mvtest
# 將某個檔案移動到某個目錄去，就是這樣做！

範例二：將剛剛的目錄名稱更名為 mvtest2
[root@www tmp]# mv mvtest mvtest2 <== 這樣就更名了！簡單～
# 其實在 Linux 底下還有個有趣的指令，名稱為 rename，
# 該指令專職進行多個檔名的同時更名，並非針對單一檔名變更，與mv不同。請man rename。

範例三：再建立兩個檔案，再全部移動到 /tmp/mvtest2 當中
[root@www tmp]# cp ~/.bashrc bashrc1
[root@www tmp]# cp ~/.bashrc bashrc2
[root@www tmp]# mv bashrc1 bashrc2 mvtest2
# 注意到這邊，如果有多個來源檔案或目錄，則最後一個目標檔一定是『目錄！』
# 意思是說，將所有的資料移動到該目錄的意思！
```

## 修改檔案時間或建置新檔：touch

```
[root@www ~]# touch [-acdm] 檔案
選項與參數：
-a : 僅修訂 access time；
-c : 僅修改檔案的時間，若該檔案不存在則不建立新檔案；
-d : 後面可以接欲修訂的日期而不用目前的日期，也可以使用 --date="日期或時間"
-m : 僅修改 mtime；
-t : 後面可以接欲修訂的時間而不用目前的時間，格式為 [YYMMDDhhmm]

範例一：新建一個空的檔案並觀察時間
[root@www ~]# cd /tmp
[root@www tmp]# touch testtouch
[root@www tmp]# ls -l testtouch
-rw-r--r-- 1 root root 0 Sep 25 21:09 testtouch
# 注意到，這個檔案的大小是 0 呢！在預設的狀態下，如果 touch 後面有接檔案，
# 則該檔案的三個時間 (atime/ctime/mtime) 都會更新為目前的時間。若該檔案不存在，
# 則會主動的建立一個新的空的檔案喔！例如上面這個例子！

範例二：將 ~/.bashrc 複製成為 bashrc，假設複製完全的屬性，檢查其日期
[root@www tmp]# cp -a ~/.bashrc bashrc
[root@www tmp]# ll bashrc; ll --time=atime bashrc; ll --time=ctime bashrc
-rw-r--r-- 1 root root 176 Jan  6  2007 bashrc <==這是 mtime
-rw-r--r-- 1 root root 176 Sep 25 21:11 bashrc <==這是 atime
-rw-r--r-- 1 root root 176 Sep 25 21:12 bashrc <==這是 ctime
```

## which (尋找『執行檔』)

```
[root@www ~]# which [-a] command
選項與參數：
-a : 將所有由 PATH 目錄中可以找到的指令均列出，而不止第一個被找到的指令名稱

範例一：分別用 root 與一般帳號搜尋 ifconfig 這個指令的完整檔名
[root@www ~]# which ifconfig
/sbin/ifconfig <==用 root 可以找到正確的執行檔名喔！
[root@www ~]# su - vbird <==切換身份成為 vbird 去！
[vbird@www ~]$ which ifconfig
/usr/bin/which: no ifconfig in (/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin
:/home/vbird/bin) <==見鬼了！竟然一般身份帳號找不到！
# 因為 which 是根據使用者所設定的 PATH 變數內的目錄去搜尋可執行檔的！所以，
# 不同的 PATH 設定內容所找到的指令當然不一樣啦！因為 /sbin 不在 vbird 的
# PATH 中，找不到也是理所當然的啊！瞭乎？
[vbird@www ~]$ exit <==記得將身份切換回原本的 root
```

## whereis (尋找特定檔案)

```
[root@www ~]# whereis [-bmsu] 檔案或目錄名
選項與參數：
-b : 只找 binary 格式的檔案
-m : 只找在說明檔 manual 路徑下的檔案
-s : 只找 source 來源檔案
-u : 搜尋不在上述三個項目當中的其他特殊檔案

範例一：請用不同的身份找出 ifconfig 這個檔名
[root@www ~]# whereis ifconfig
ifconfig: /sbin/ifconfig /usr/share/man/man8/ifconfig.8.gz
[root@www ~]# su - vbird <==切換身份成為 vbird
[vbird@www ~]$ whereis ifconfig <==找到同樣的結果喔！
ifconfig: /sbin/ifconfig /usr/share/man/man8/ifconfig.8.gz
[vbird@www ~]$ exit <==回歸身份成為 root 去！
# 注意看，明明 which 一般使用者找不到的 ifconfig 卻可以讓 whereis 找到！
# 這是因為系統真的有 ifconfig 這個『檔案』，但是使用者的 PATH 並沒有加入 /sbin
# 所以，未來你找不到某些指令時，先用檔案搜尋指令找找看再說！

範例二：只找出跟 passwd 有關的『說明文件』檔名(man page)
[root@www ~]# whereis -m passwd
passwd: /usr/share/man/man1/passwd.1.gz /usr/share/man/man5/passwd.5.gz
```

## Linux 系統常見的壓縮指令

```
*.Z      compress 程式壓縮的檔案；
*.gz     gzip 程式壓縮的檔案；
*.bz2    bzip2 程式壓縮的檔案；
*.tar     tar 程式打包的資料，並沒有壓縮過；
*.tar.gz  tar 程式打包的檔案，其中並且經過 gzip 的壓縮
*.tar.bz2 tar 程式打包的檔案，其中並且經過 bzip2 的壓縮
```

## gzip(壓縮), zcat(解壓縮)

```
[root@www ~]# gzip [-cdtv#] 檔名
[root@www ~]# zcat 檔名.gz
選項與參數：
-c    : 將壓縮的資料輸出到螢幕上，可透過資料流重導向來處理；
-d    : 解壓縮的參數；
-t    : 可以用來檢驗一個壓縮檔的一致性～看看檔案有無錯誤；
-v    : 可以顯示出原檔案/壓縮檔案的壓縮比等資訊；
-#    : 壓縮等級，-1 最快，但是壓縮比最差、-9 最慢，但是壓縮比最好！預設是 -6

範例一：將 /etc/man.config 複製到 /tmp，並且以 gzip 壓縮
[root@www ~]# cd /tmp
[root@www tmp]# cp /etc/man.config .
[root@www tmp]# gzip -v man.config
man.config:      56.1% -- replaced with man.config.gz
[root@www tmp]# ll /etc/man.config /tmp/man*
-rw-r--r-- 1 root root 4617 Jan  6 2007 /etc/man.config
-rw-r--r-- 1 root root 2684 Nov 10 17:24 /tmp/man.config.back.Z
-rw-r--r-- 1 root root 2057 Nov 10 17:14 /tmp/man.config.gz <==gzip壓縮比較佳
```

```
範例二：由於 man.config 是文字檔，請將範例一的壓縮檔的內容讀出來！
[root@www tmp]# zcat man.config.gz
# 由於 man.config 這個原本的檔案是文字檔，因此我們可以嘗試使用 zcat 去讀取！
# 此時螢幕上會顯示 man.config.gz 解壓縮之後的檔案內容！

範例三：將範例一的檔案解壓縮
[root@www tmp]# gzip -d man.config.gz
# 不要使用 gunzip 這個指令，不好背！使用 gzip -d 來進行解壓縮！
# 與 gzip 相反，gzip -d 會將原本的 .gz 刪除，產生原本的 man.config 檔案。

範例四：將範例三解開的 man.config 用最佳的壓縮比壓縮，並保留原本的檔案
[root@www tmp]# gzip -9 -c man.config > man.config.gz
```

## bzip2(壓縮), bzip2(解壓縮)

```
[root@www ~]# bzip2 [-cdkzv#] 檔名
[root@www ~]# bzcat 檔名.bz2
選項與參數：
-c    : 將壓縮的過程產生的資料輸出到螢幕上！
-d    : 解壓縮的參數
-k    : 保留原始檔案，而不會刪除原始的檔案喔！
-z    : 壓縮的參數
-v    : 可以顯示出原檔案/壓縮檔案的壓縮比等資訊；
-#    : 與 gzip 同樣的，都是在計算壓縮比的參數，-9 最佳，-1 最快！

範例一：將剛剛的 /tmp/man.config 以 bzip2 壓縮
[root@www tmp]# bzip2 -z man.config
# 此時 man.config 會變成 man.config.bz2 !

範例二：將範例一的檔案內容讀出來！
[root@www tmp]# bzcat man.config.bz2
# 此時螢幕上會顯示 man.config.bz2 解壓縮之後的檔案內容！！

範例三：將範例一的檔案解壓縮
[root@www tmp]# bzip2 -d man.config.bz2

範例四：將範例三解開的 man.config 用最佳的壓縮比壓縮，並保留原本的檔案
[root@www tmp]# bzip2 -9 -c man.config > man.config.bz2
```

## tar 打包(不是壓縮)



```

[root@www ~]# tar [-j|-z] [cv] [-f 建立的檔名] filename... <==打包與壓縮
[root@www ~]# tar [-j|-z] [tv] [-f 建立的檔名] <==察看檔名
[root@www ~]# tar [-j|-z] [xv] [-f 建立的檔名] [-C 目錄] <==解壓縮
選項與參數：
-c : 建立打包檔案，可搭配 -v 來察看過程中被打包的檔名(filename)
-t : 察看打包檔案的內容含有哪些檔名，重點在察看『檔名』就是了；
-x : 解打包或解壓縮的功能，可以搭配 -C (大寫) 在特定目錄解開
    特別留意的是，-c, -t, -x 不可同時出現在一串指令列中。
-j : 透過 bzip2 的支援進行壓縮/解壓縮：此時檔名最好為 *.tar.bz2
-z : 透過 gzip 的支援進行壓縮/解壓縮：此時檔名最好為 *.tar.gz
-v : 在壓縮/解壓縮的過程中，將正在處理的檔名顯示出來！
-f filename: -f 後面要立刻接要被處理的檔名！建議 -f 單獨寫一個選項囉！
-C 目錄 : 這個選項用在解壓縮，若要在特定目錄解壓縮，可以使用這個選項。

其他後續練習會使用到的選項介紹：
-p(小寫) : 保留備份資料的原本權限與屬性，常用於備份(-c)重要的設定檔
-P(大寫) : 保留絕對路徑，亦即允許備份資料中含有根目錄存在之意；
--exclude=FILE: 在壓縮的過程中，不要將 FILE 打包！

```

## 最常用的部分

壓縮：tar -jcv -f filename.tar.bz2 要被壓縮的檔案或目錄名稱

查詢：tar -jtv -f filename.tar.bz2

解壓縮：tar -jxv -f filename.tar.bz2 -C 欲解壓縮的目錄

.zip

解壓：unzip FileName.zip

壓縮：zip FileName.zip DirName

以上教材 引用參考鳥哥的私房菜

指令:

chgrp

chown

chmod

ls -l

ls -al

cd

cp

touch

mkdir

rmdir

rm

mv

pwd

which

whereis

wget

unzip

kill