

第五單元

Linux 的基本操作與檔案結構

5.1 Red Hat Linux 的檔案結構

當我們試著處理一筆龐大的資料的時候，總是換先分類以便予以後的尋找和利用，unix 的檔案系統的設計理念就是從這簡單的觀念開始，然而對於其他的作業系統也有相當深遠的影響。

對於從其他的作業系統轉而完 unix 系統的人，都不難發現到 unix 和 DOS 的相似之處，像目錄和檔案這兩個檔案系統的基本組件，unix 不僅都有還更加的方便化，比如說把一般的硬體周邊設備都當作檔案來看待~等，皆是一般作業系統所沒有的，雖然說是先有 unix 再有 Ms 等作業系統，但顯然之後的作業系統都沒有 unix 所發展出來族類好！

一般而言，資料在分類時，都會分類成一層層的架構，這樣的階層式架構稱之為樹狀架構，有用過 dos 的人都不難明白筆者在講什口，爾且還會覺得是在老生常談了！這樣的樹狀分類，最大的優點就是簡單清楚，但是對於 uinx 族類這種龐大的自由發展系統，所包含的意義就不只這些了！用過 DOS 的人應該都知道，\DOS 下放置的通常是 DOS 的工具程式，看到 \WINDOWS 大概已經知道是 MS-Windows 的系統程式，而這就是簡單的『目錄構造』觀念。因為系統的不同，當我們在安裝 DOS、Windows 套裝軟體的時候，這些軟體通常會將所有的檔案裝在自己的目錄下，只有少部份 .DLL .INI 會安裝到 Windows 的目錄下。

5.1.1 Unix 目錄構造的重要性

在許多 UNIX 系統中，目錄都有被定義其下檔案的特性，這不但程式開發者、還有使用者，在安裝檔案或是在找檔案時，有一個重要的依據，尤其是像 UNIX 的系統，是一個巨大樹狀結構的目錄，如果沒有嚴謹的目錄規範，那就不用懷疑，這整個系統看起來一定是亂七八糟的，因此我們一定要瞭解 Red Hat Linux 整個的目錄構造。

然而整個 Linux 檔案系統的最上層是根目錄 "/"，底下再一層一層、分門別類的擁有許多的目錄與檔案，所以稱之為 "階層式" 的檔案系統，其實 Linux 底下的目錄也是一個檔案，只是裡面放的不是使用者的資料或程式，而是放位於此目錄底下一層的檔案的名稱、所在位置... 等資料，要在 Linux 的檔案系統裡面指定一個檔案的名稱有兩種方法，一種是指定絕對路徑，一種是指定相對路徑。

5.1.1.1 絕對路徑

假設我們要用絕對路徑來指明某一個檔案的位址，則此檔案的絕對路徑一定是由"/"開頭，從根目錄開始，一層一層的寫目錄的名稱，目錄跟目錄中間再以"/"相隔。

假設你現在在 root 的目錄有一個叫 iori 的檔案，用絕對路徑來講的話，那門 iori 這個檔案的絕對路徑就是/root/iori。當然，在 Linux 的系統中同一個目錄下是不能有相同檔名的目錄或檔案，但若不是在相同的目錄下，就沒有這個限制，因為若不是在相同的目錄，則這些同名檔案或目錄的絕對路徑都不同，所以不會有衝突，因此 Linux 系統裡，因此所有的絕對路徑檔名都是唯一，不會有重複。

5.1.1.1 相對路徑

指定檔案的名稱的時候假如不用"/"開頭的，就表示是用相對路徑，相對路徑是以當前所在的目錄起算，並且用一個小數點"."表示目前目錄，兩個小數點"..表示上一層目錄。例如，假設在/root/目下有一個執行檔叫 iori，然而在.bash_profile 檔案內容中的參數 PATH 中所設定的執行檔案路徑中也有一個 iori，這時若你執行 iori，則 shell 會執行 PATH 參數執行路徑中的 iori 而不執行目前目錄的 iori，因此若你要執行目前目錄下的 iori，則你應該在執行時在執行檔名稱前加上 ./ 表示指定現行目錄下的檔案，如：./iori。

".."表示上一層目錄，通常我們都用來切換目錄用的，比如說執行 cd .. 這個指令這時就會切換到上一層目錄了。

5.1.2 使用者目錄 (家目錄)

上一章我們曾經說過用 ~ 符號開頭來表示使用者的家目錄，當系統管理者(root)給 Linux 底下使用者帳號的同時，也會給一個家目錄，而 root 本身也有自己的家目錄這個家目錄就是我們第一次登入時所進入的目錄，其他的使用者登入時也是會直接進到自己的家目錄下。

除了給你家目錄之外也會幫你指定一個 shell 來執行命令。每個使用者都可以在自己的目錄進行讀寫的動作，簡直就像自己可以為所欲為，別人不得入侵的勢力範圍一樣，而且當我們要存檔或產生新檔，家目錄也是我們預設的存寫位址。在 Red Hat 系統下，通常使用者的家目錄都統一放在/home 下來方便管理。

對於 ~ 的使用我們，以使用者 root 為例，root 的家目錄是 "/root"，則當我們以 root 的身分執行 ls ~/，這時就會列出所有 /root 底下的檔案，因為~代表家目錄，所以 ls ~/相當於列出家目錄檔案的內容。

講了這門多，讀者應該會利用上面提到的幾種方式在 Linux 系統裡面通行了吧！，

5.2 如何執行指令

在開始介紹指令之前，想必讀者一定不太知道之前筆者所下的指令意義為何？現在開始我們就要開始學習駕馭 Linux 的工具了！也許讀者會想：這時候還要下指令 Y！為什麼不用 X 視窗來管理檔案系統了？這時筆者必須要說的圖形介面的操作雖然好用，但目前用於遠端操控主機還是文字居多，因此學習指令是絕對必要的，而且用下指令的方式讀者才會真正學習到系統運作的觀念，而不只限於按按滑鼠而已！

一般而言下指令必須讓 UNIX 系統了解、接受才能真正的執行，因此下指令的時候，必須遵照一定的格式，當然每個指令的格式都不盡相同，有的指令直接執行即可，有的必須要附加相關的參數才行。因此在下指令前必須先查清楚這個指令的相關使用規則。

底下提供一般自我學習並且查詢指令使用的方法，這些資料通常可以直接由硬碟中取得，無須額外上網，新手們應能從中抓到一些「感覺」：

1. `command -h` 或 `command --help`

傳統 Unix 的指令大多會提供上述的「操作提示」，如果您對於某個指令還算熟悉，只是突然忘了該用哪個選項參數，此法便能指點迷津。

2. `man command` 或 `info command`

`man` 指的就是 manual page 的意思。`man page` 長久以來可說是 Unix 系統裡的「舊約聖經」，其內容條述而精闢，高手們經常喜歡參考或引用。不過，許多舊式的 `man page` 並沒有包含範例指引，硬梆梆的內容常叫新手們望而卻步。而 `info` 則是 GNU 計畫中，打算用來取代 `man page` 的工具，有部份指令的說明內容，目前僅有 `info` 在更新維護了。雖然 `man page` 或 `info` 新手們不易吸收，在下的建議是，看不懂倒沒關係，不過得熟悉它們的操作方式，終究這是 Unix 上最容易取得的資料之一。有時候，您會看到類似 `man signal(7)` 的指示，它表示您應該以 `man 7 signal` 方式來讀取特定章節的內容，7 這個號碼，代表的就是經分類過的章節主題，您可以查詢到 `signal` 的說明，是分散在不同的章節主題中。

3. `cd /usr/doc/command`

Red Hat Linux 在安裝 rpm 檔案之際，通常會同時把該 package 相關的 documentation 置於 `/usr/doc` 當中，有些資料也值得大家參考哦。

一般使用指令的格式如下：

指令名稱 選項 參數 + 【Enter】

指令、選項、參數之間必須用空白鍵隔開，打完後直接按【Enter】執行即可。對於選項、參數的說明如下：

選項：

在指令後面加上選項的目的是要稍微改變指令執行的結果。比如說你用 ls 指令來列出檔案目錄，這時會因為你在 ls 後面加上不同的選項而有不同的列出方式。有的指令有很多的選項，當然你也可以一次同時對一個指令加上好多個選項，或者完全都不用，而選擇項皆以“-”或"--"來引導。當您要使用很多的選擇項時，其前後的次序不會影響指令的動作，但要注意的是大小寫的區分。

參數：

參數簡單的說就是指令執行的對象，通常是一個檔案或目錄的名稱。同樣的，參數也可以同時一次指定多個，但是必須要用空白間隔開喔！

現在我們舉一個指令來說明指令輸入的格式：

```
[root@net53 /root]# ls -la core
```

其中 ls 是指令名稱，-la 是選項，core 是參數。選項-la 是兩個選樣的合成寫法，當然我們也可分開寫成-a -l，等整行輸入完畢後直接按【Enter】執行，若所輸入的字元超過一整行也沒關係，這時會自動跳到下一行，你再繼續輸入而不用按任何的換行鍵，只要結束的時候記得按【Enter】就好。

假設你輸入的指令不和語法或錯誤，這時 shell 會自動判別你所輸入的指令那裡錯誤，而給你不同的錯誤訊息。假設你 ls 打成 is 這時系統就會直接給你

bash: 錯誤的指令名稱: command not found

的錯誤訊息表示沒有這個指令，若你的選項打錯，則會出現

option requires an argument – 錯誤的選項

請用`指令 --help'來顯示本指令的用法

的訊息，告訴你沒有這個選項，若是參數則會出現

指令: 錯誤的參數: No such file or directory

來告訴你沒有這樣的檔案或目錄。

必須注意的是在 `unix` 的系統中，所有字元是有分大小寫的，比如說同一種字母的檔案，`ROOT` 和 `root` 是兩回事，所以在你輸入檔案或目錄名稱的時候，系統一直不接受，這時你可以看看是否按到【`caps Lock`】鍵，導致所輸入的字元變成大寫。

5.3 UNIX 基本指令簡介

UNIX 的指令簡直跟山一樣多，而且包羅萬象，功能強大不在話下，如果想學習它的話，只有多多使用才有可能記得住，最好你在看本節的時候，就已經做在終端機前邊學邊實習，這樣保證事半功倍。在上面我們也可以用 `man` 查詢線上使用手冊，來幫助讀者學習筆者沒介紹的指令。例如：

```
[root@net53 /root]# man ls
Formatting page, please wait...
LS(1)                                FSF                                LS(1)
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
DESCRIPTION
    List information about the FILEs (the current directory by
    default).  Sort entries alphabetically if none of -cftuSUX
    nor --sort.
    -a, --all
        do not hide entries starting with .
    -A, --almost-all
```

這時可按下空白鍵，繼續下頁的說明，當您查詢完畢後，想離開 `man` 可以在：符號下按下“`q`”鍵離開，便會跳到提示符號下。

5.3.1 `more` 和 `less` 指令 閱讀檔案內容一次以一個 `page` 顯示

通常我們要看一篇很長的文件，都希望是從頭看到尾，在 `Linux` 中，`more` 指令可以讓您以一個 `page` 為單位，來看完您的文件。`More` 在執行時也有提供幾個功能，首先示弱是要向下捲一頁，只要按下空白鍵即可，不能向上捲動倒是真的，然而 `less` 可以用上下鍵隨意捲動瀏覽喔！在螢幕最下方我們可以看到一行.....`more`.....(??%)，這代表還有其他的資料，括弧裡的百分比代表著目前所

顯示的最底行位址佔整個檔案長度的百分比。看完了，想結束就按【q】鍵即可！必須注意的是所瀏覽的文字檔必須是 ascii 格式喔！~否則用 binary 格式可以一遍亂碼喔！

more (less) 常用語法為：【more (less) 檔案名稱】

5.3.2 cat 指令 閱讀檔案內容

cat 可不是貓喔！它雖然跟 more 一樣是閱讀檔案內容的指令，其語法格式同 more，但並不能分頁顯示，所以大都不用來當閱讀檔案用，而是用它配合重導符號將把一個檔案內容拷貝到另一個檔案內容，也就是所謂的合併檔案。例如：

```
[root@net53 /root]# cat a.txt > b.txt
```

這時你在用 cat 或 more 指令來看 b.txt 檔，你會發現 a.txt 檔案內容緊跟著 b.txt 的檔案內容喔！

5.3.3 ls 指令 查看目錄及檔案

ls 指令是用來瀏覽檔案與目錄，這個動作相當於 DOS 中的 dir 指令，在這之前我們已經用這個指令做過不少示範了，相信讀者應該不陌生。

鍵入 ls，則會出現該目錄內的內容，且以橫向排列，其中有些名稱的後面有“/”，則為目錄，其顏色為藍色，如：Server/、fonts/；有些名稱的後面有“@”，則為鏈結檔【可鏈結檔案或目錄】，其顏色為青色，如：libX11.so.3@、X11@；有些名稱的後面有“*”，則為可執行檔，其顏色為綠色，如：libX11.so.3.1.0*、libXaw.so.6.0*。

另外還有一些是在下框中所沒有的，像壓縮檔所代表的顏色為紅色，如：.gz、.tar、.zip 等，設備檔所代表的顏色為黃色，如：fd0、hda 等。請看下框所示：

```
[root@net53 /root]# ls /usr/X11R6/lib# ls    ↵ 瀏覽/usr/X11R6/lib 內的內容
Server/      libX11.so.3@    libXaw.so.6@    libXt.sa
X11@         libX11.so.3.1.0*  libXaw.so.6.0*  libXt.so.3@
fonts/       libX11.so.6@    libXdmpc.a      libXt.so.3.1.0*
libFS.a      libX11.so.6.0*  libXext.a       libXt.so.6@
..... ↵ 後面還有檔案，因為太多，故在此省略
```

在上面的框框中，您可能還看不出來（橘紅色）鏈結檔是在做什麼用的，其實鏈結檔是用來指向固定的檔案或目錄，假如有一個檔案的名稱很長【在 Linux

中，檔案系統為長檔名】，而且在很深層的目錄內，這樣每次要使用該檔案時，打起字來就會很累，這時只要用一個很簡單的名稱，且指向該檔案，以便形成鏈結檔，這樣以後只要執行這簡單的名稱，就等於執行該檔案。

至於要如何才能看到鏈結檔是指向那裡呢？其實很簡單，您只要下 **ls -la** 指令即可，其中選擇項 **a** 為要顯示隱藏檔【隱藏檔都是以“.”為開頭的檔案，如：`.netscape/`、`.bash_history`】，選擇項 **l** 為要以長格式顯示。請見下框所示：

```
[root@net53 /root]# ls /usr/X11R6/lib# ls -la | more
total 5850 全部的檔案數
drwxr-xr-x 6 root root 2048 Oct 13 14:30 ./
drwxr-xr-x 7 root root 1024 Oct 13 14:31 ../
drwxr-xr-x 7 root root 1024 Oct 13 14:31 Server/ ↓X11 指向的目錄
lrwxrwxrwx 1 root root 14 Mar 2 17:40 X11 -> /var/X11R6/lib/
drwxr-xr-x 10 root root 1024 Oct 13 14:31 fonts/
檔案擁有人代號↑ ↑檔案擁有人群組代號
-rw-r--r-- 1 root wheel 65520 Oct 10 1998 libFS.a
-rw-r--r-- 1 root wheel 103614 Oct 10 1998 libICE.a
↑ 檔案使用權限
```

在上框中，讀者可以看到檔案擁有人代號『root』，檔案擁有人群組代號『root』，鏈結檔 X11 其實是用來指向 `/var/X11R6/lib/` 這個固定的目錄，及檔案使用權限『lrwxrwxrwx』。

在以上範例中所下的指令為【**ls -al | more**】，其中“|”為管道，也就是說 **ls -al** 的動作經由管道送給 **more**，再由 **more** 輸出動作，以達成用長格式來顯示目錄內容，並且輸出一個 page 後就暫停。

而在上框中，您或許還有不明白之處，像這個『drwxrwxrwx』是什麼東西，且用來做什麼？其實這是檔案使用權限，其每一個字母，都有其特別的意義，分別解釋如下：開頭的第一個字元，有很多的字母，像“d”是代表目錄、“l”是代表鏈結、“c”是代表字元設備、“b”是代表區塊設備、“s”是代表 socket、“p”是代表 pipe；接著第一個字元下來的第一組 **rwX** 是代表檔案擁有人的使用權限，其中“r”代表可讀、“w”代表可寫、“x”代表可執行，倘若沒有這三種屬性的其中之一，那便以“-”為代表；第二組與第三組的 **rwX** 則是分別為同群組使用者和其它使用者的使用權限。

drwx	rwX	rwX
『檔案所有人』	『同群組使用者』	『其它使用者』

所謂『同群組使用者』，是指屬於相同群組【group】的使用者，例如所有的學生帳號，就全部都為 student 群組，至於 group 如何設定，我們稍後會作介紹。

而『其它使用者』是指同群組以外的使用者。還有在上框中，您應該有看到一個 root 和一個 wheel，那這是什麼呢？其實這是指該檔案的擁有人，及擁有人所屬的群組，這樣可讓別人知道這檔案是誰的，也就是說能夠修改檔案使用權限的人。

5.3.4 cd 指令 切換目錄

在 Linux 中切換目錄的方法與 Dos 完全相同，其語法如下：

cd 常用語法：【cd 目錄名稱】

cd 除了有切換目錄的功能外，還有一個功能就是當您不管在那個目錄內，只要下 cd 指令不用接任何參數，就可回到您的使用者目錄內【home directory】，通常使用者目錄【home directory】都可用“~”來代表。還有 Linux 的樹狀結構目錄與 Dos 完全相同【其實是 Dos 學 Unix 而來】。

例如：我們現在不管位置在哪，都可以馬上回到自己的使用者目錄，其做法為：
cd ~ 或直接打 cd 即可

5.3.5 mkdir、rmdir 指令 建構目錄、刪除目錄

接下來我們要敘述如何建構目錄、如何刪除目錄，其實這些指令操作方法與 Dos 中的 md、rd 是完全相同，差別只是指令的名稱不同而已。

mkdir 常用語法：【mkdir 目錄名稱】

rmdir 常用語法：【rmdir 目錄名稱】

在使用 rmdir 時，要確保該目錄內已無任何的檔案存在，才可使用該指令。

5.3.6 cp 指令 拷貝檔案

在之前，我們有提到拷貝的動作，接下來我們來看一下拷貝的語法：

cp 常用語法：【cp [-r] 檔案 目的地】

其實 cp 的語法與 Dos 的 copy 語法大致相同，但 cp 的功能還是比 Dos 中的 copy 強大許多，其中選擇項“-r”就是一例【在語法中，使用中括號是表示該選擇項是可使用或可不使用，一切都由使用者來決定】，它的意思是說假如您要拷貝

一個目錄，但該目錄內還有好多個目錄的話，您可以使用“-r”，來幫您拷貝目錄內的所有目錄檔案，並且在拷貝時會自動幫您建目錄，而這功能就相當於 Dos 內的 xcopy。

還有一點須注意的是，在使用 cp 時一定要有目的地才行，在 Dos 中做 copy 時，有時是可省略目的地，但在 Linux 中就無法這樣做。在 cp 中我們也會常用到萬用字元，像*、?等，至於詳細的萬用字元介紹，前面已經介紹了。

例如：我們要將 root 目錄內的所有檔案【包括目錄】，但不包括隱藏檔，拷貝至根目錄下的 tem 內，其作法為：

```
cp -r /root/* /tem
```

若要拷貝所有小寫字母開頭隱藏檔【包括隱藏目錄】的話，其作法為：

```
cp -r /root/.[a-z]* /tem
```

5.3.7 rm 指令 刪除檔案/目錄

rm 指令與 Dos 的 del 指令有些類似之處，不同的是，rm 功能的強大與其相對殺傷力遠不是 Dos 的 del 可以相比。【只要是檔案或目錄，不管有無隱藏，或是檔案使用權限只設定成唯讀，rm 皆可刪除】，而且被刪除的檔案也絕對救不回來，所以讀者要千萬小心使用這個指令，而 rm 的語法為：

rm 常用語法：【rm [-r] 檔案/目錄】

在此要注意的是選擇項“-r”，這個 -r 與 cp 中的 -r 有類似的功能，它通常都用來一併刪除目錄內的目錄檔案，這個功能就相當於 Dos 中的 deltree 功能，通常在使用 deltree 時都會有提示語，但在 FreeBSD 中使用 rm 就沒有任何的提示語，還有最糟糕的是已刪除的檔案是無法救回來的，所以在使用 rm -r 時要特別小心。

例如：我們要刪除剛剛的 tmp 目錄，我們可以用下面的基本方法來做刪除的動作：

- 先到 tmp 內將檔案全部刪除：

```
cd tmp ; rm *
```

【在 Linux 中要一次下兩個不同的指令時，只需在指令與指令之間加上分號即可，這樣 Linux 便會依照排列的先後次序來依續執行，在本例中，會先執行 cd tmp 再執行 rm *。】

- 假如 tmp 內還有目錄，則仍然用上面的方法，來刪除目錄內的檔案，
- 將目錄切換至 tmp 目錄下，然後將每個空目錄刪除
- 退回主目錄，最後將 tmp 目錄刪除：

```
cd / ; rmdir tmp
```

以上是基本的方法，但速度太慢，因為您還要切換目錄，在此要用 rm 指令一次就可完全刪除，這樣速度又快又方便，請見下示：

```
rm -r /tmp
```

5.3.8 mv 指令 檔案/目錄更名或搬移

有的時後我們會做檔案/目錄更名的動作，或是搬移檔案/目錄，其實檔案/目錄更名與搬移檔案/目錄的動作原理是一樣的，差別只是路徑的不同。

mv 常用語法：【mv 檔案/目錄名稱 搬移的目的地】

其實 mv 指令通常都用來搬移檔案，當我們要做檔案更名時，只要改變一下就好。

例如：我們要把現在所在的目錄中的 netscape 檔案，搬到/usr 內，可用：

```
mv netscape /usr
```

例如：我們要把/temp 的目錄，更名為/tmp，其作法為：

```
mv /temp /tmp
```

【注意：舊名稱與新名稱要一樣的路徑才行】

5.3.9 df 指令 查詢磁碟空間使用情形

df 指令可讓您查尋所有的硬碟規劃區的使用情形。在此 df 所呈現的空間其單位，都是以 Kbyte 為單位，若加上-h 的選項則會用 Mbyte 顯示。請見下框為例：

```
[root@net53 /root]# df
```

檔案系統	空間大小	已使用的空間大小	可用的大小	使用率	掛入點
Filesystem	1024-blocks	Used	Available	Capacity	Mounted on
/dev/hda2	197619	181481	5932	97%	/
/dev/hda1	311272	266024	45248	85%	/dosc
/dev/hdb1	118688	78794	39894	66%	/dosd
/dev/sbpcd	335724	335724	0	100%	/cdrom

5.3.10 du 指令 查詢目前目錄所佔用的空間

當我們用 ls 指令來列出檔案的明細時，美中不足的地方就是沒有列出目錄佔用了多少的硬碟空間，不知道各位讀者有沒有注意到這點，現在我們就來介紹用來查詢目前所在目錄佔用多少的硬碟空間的指令 du。

du 常用語法：du -abs 目錄名稱或檔案名稱

-a 的選項是代表顯示所有檔案目錄個別佔有的空間大小，單位是 kilobyte。

-b 的選項是代表將顯示的空間大小以 byte 表示。

-s 的選項是只顯示一個目錄的總空間大小。

我們通常也可不下任何選項來觀看，直接打 du 就可以滿足大部分使用者的需求了，這時會顯示目前使用者所在的目錄下各個檔案目錄所佔的硬

```
[root@net53 /root]# du |more
----->太多故不列出!
```

碟空間，最後便會顯示所有檔案佔的硬碟空間總和。不過檔案數目過多還得配合其他的指令，如 `more` 來做分頁的動作喔！例如：

5.3.11 `clear` 指令 清除螢幕

這個指令的意義就如同說明，使用後立竿見影，便知成效。它就向 `dos` 下的 `cls` 指令一樣，如果你覺得螢幕的字太多太亂，在提示符號執行此指令，便會清的空空如也。

5.3.12 `pwd` 指令 顯示目前工作路徑

在 `Linux` 系統下迷路了嗎？執行這個指令可以馬上讓你知道目前身處何處喔！執行的結果也就是你目前所在位址的絕對路徑。
例如：

```
[root@net53 init.d]# pwd
/etc/rc.d/init.d
[root@net53 init.d]#
```

5.3.13 `ln` 指令 製作鏈結檔

鏈結檔通常用於要編譯程式的時後，通常鏈結檔都是指向某個 `lib` 函式庫，或者 `include` 檔案等，其實這樣做都是為了方便編譯或執行，其語法為：

`ln` 常用語法：【`ln -s 要指向的檔案或目錄 鏈結檔名`】

例如：您要將 X 指向 XF86_W32 的執行檔，其作法為：

```
ln -s XF86_W32 X
```

5.3.14 echo 指令 在螢幕上顯示訊息

echo 指令會把隨在後面的訊息顯示在螢幕上，訊息和訊息間再輸入時要用空白間隔開，並且用引號或雙引號標明此訊息，參數不用此方式標明。指令結束的時候按【Enter】執行即可！通常我們都用此指令來顯示環境參數的值。以下我們用幾個實例來說明！

echo 常用語法：【echo '訊息'】；【echo \$參數】

例如：我們要顯示 I am a good man 的訊息。

```
echo 'I am a good man'
```

若我們要顯示的訊息有包含引號、空白等特殊字元做開頭或結尾，例如：'I am a good man'，則要以雙引號做標明訊息。

```
echo "I am a good man"
```

若我們要顯示環境參數，則方法如下。

```
echo $ →這時可按【Tab】鍵來顯示所有的環境參數。
```

```
echo $PATH →選擇顯示 $PATH 參數。
```

5.3.15 mesg 指令 控制是否接收其他使用者的訊息

當我們擁有帳號後，如果你正在使用終端機，此時其他使用者企圖給你某些資料時，你可以決定是否接收此訊息。

mesg 常用語法：【mesg -y】；【mesg -n】

使用-n可以拒絕其他使用者傳訊息到你的終端機，選項-y可以恢復訊息的顯示。

5.3.16 talk 指令 線上交談

使用此指令，這時終端機畫面分為兩部分，一部份是你的終端機畫面，令一部份是交談對象的，所以你可以看到你和對方的文字出現在螢幕上，因此而進行交談。

talk 常用語法：【talk 對方使用者名稱】；【talk 對方使用者名稱@主機名稱】

在交談的時候，我們除了可以輸入文字交談，結束後按【ctrl】+【d】進行結束，若對方不在本台主機上而在另一台主機，這時你就必須在輸入指令時，順便輸入對方的主機名稱。

5.3.17 who (w) 和 who am i 指令 查詢目前在線上的使用者資料

who 指令會顯示目前在線上的使用者的名稱、終端機名稱和登入系統的時間。若使用 who am I 指令則會顯示你的使用者的名稱、終端機名稱和登入系統的時間。

Who(w) 常用語法：【who(w) 選項】

- a：使用全部的選項。
- b：顯示上一次重新載入系統的資訊。
- d：報告已經終止的處理程序。
- r：顯示執行的層級。
- T：顯示終端機的狀態，有三種狀態：+ 可寫入，- 不可寫入、? 未知情況。

```
[water@net53 water]$ who
root    tty1      Jul 26 11:02 → 在本機登入的使用者
root    tty2      Jul 26 12:01 → 在本機登入的使用者
water   pts/0     Jul 26 12:12 (net49.ee.oit.edu.tw) → 遠端登陸的線上使用者
[water@net53 water]$ who am i
net53.ee.oit.edu.tw!water pts/0    Jul 26 12:12 (net49.ee.oit.edu.tw)
[water@net53 water]$
```

5.3.18 touch 指令 更新檔案被更改的時間或建立新檔案

touch 會把檔案的更改時間改為目前的時間，若指定的檔案不存在，則會產生新檔。

touch 常用語法：【touch -c -f 檔案名稱】

- c：如果指定的檔案名稱不存在則不產生新檔。
- f：即使不能更改，但會強制更改檔案的時間。(但不一定成功)

5.3.19 dmesg 指令 顯示系統資訊

在開機時系統核心會一一偵測您的周邊，有一堆關於系統的資訊，不過這些訊息往往很快就跑過去，來不及看，在提示符號下面打 "dmesg" 就可以看到這些訊息了，還可以用 "dmesg | more" 慢慢看。

5.3.20 gzip、tar 指令 壓縮/解壓縮工具

gzip 在 Linux 中，經常被用做壓縮/解壓縮工具，而 tar 是將一個目錄內的所有東西，壓成一個檔案，這個功能相當於 Dos 中 *.ddi 的東西，這兩個不同的程式之所以放在一起講，是因為這兩個程式經常配合使用，接下來我們就分別介紹。

在早期 Unix 中，都是用 compress 程式來壓縮，其壓縮出來都為 .Z，而現今都是改採用 gzip，其壓縮出來都為 .gz，而 gzip 也可以解以前格式 .Z 的壓縮檔，其常用用法為：

```
壓縮 ⇨ gzip -9 檔名
解壓縮 《1》 ⇨ gzip -d 檔名.gz
解壓縮 《2》 ⇨ gzip -d 檔名.Z
```

在壓縮中的檔名，都事先已用過 tar 壓縮過，然後再用 gzip 來壓縮的。

例如：有一個檔案為 test.tar.gz，其解壓縮的方法為：

```
gzip -d test.tar.gz
```

這樣便可解壓縮，當您用 ls 指令看一下時，會看到解壓縮完後的檔名變為 test.tar，這樣只是將 gzip 的部份解壓縮完而已，還必須用 tar 指令再解壓縮，把 test 還原。

在一般 Unix 環境下，tar 所作的是檔案歸納的動作，並將之存入儲存媒體上，tar 並沒有執行壓縮動作。至於在 Linux 環境下所使用的 tar，則有內建整合且含有壓縮功能，也就是說在經過歸檔之後，還可一併壓縮，這等於兩個動作一次合成。其 tar 的常用用法為：

```
壓縮 《1》 ⇨ tar cvfz test.tar.gz ./
壓縮 《2》 ⇨ tar cvf test.tar ./
```

解壓縮《1》⇒ tar zxvf test.tar.gz

解壓縮《2》⇒ tar xvf test.tar

tar 中的選擇項“c”代表要建立一個新的壓縮檔；選擇項“v”代表在壓縮或解壓縮時，要顯示路徑與檔案名稱；選擇項“f”代表要開啟您所指定的壓縮檔，不管壓縮或是解壓縮都要有這選擇項“f”；選擇項“x”代表要解壓縮；選擇項“z”代表要處理【壓縮/解壓縮】*.tar.gz 的檔案。

壓縮《1》的意思為壓縮目前目錄下的所有檔案及目錄，成為 test.tar.gz

壓縮《2》的意思與壓縮《1》差不多，只是壓縮成 test.tar

解壓縮《1》的意思為將 test.tar.gz 解壓縮

解壓縮《2》的意思為將 test.tar 解壓縮。

5.3.21 ps 指令 查尋程序執行狀態

當您有時需要查看一下，某一個 server 或程式是否有正在執行【這情形相當於 Dos 中的長駐程式一樣】時，您可以使用 ps 指令。請見下例：

```
[root@net53 /root]# ps uax | more → 可配合more來分頁
```

USER	PID	%CPU	%MEM	SIZE	RSS	TTY	STAT	START	TIME	
root	1	0.0	0.5	44	232	?	S	14:42	0:00	init
root	6	0.0	0.4	32	164	?	S	14:42	0:00	bdflush
(daemon)										
root	7	0.0	0.4	32	168	?	S	14:42	0:00	update
(bdflush)										
root	25	0.0	0.5	60	228	?	S	14:42	0:00	/usr/sbin/crond -l10
root	38	0.0	0.7	73	280	?	S	14:42	0:00	/usr/sbin/syslogd
root	40	0.0	0.6	44	240	?	S	14:42	0:00	/usr/sbin/klogd
root	42	0.0	0.6	72	268	?	S	14:42	0:00	/usr/sbin/inetd
root	44	0.0	0.5	64	216	?	S	14:42	0:00	/usr/sbin/lpd
root	49	0.0	1.1	251	448	?	S	14:43	0:00	


```

sendmail: accepting con
root    53    0.0    1.3    380    540    v01    S      14:43    0:00    -bash
root    54    0.0    1.3    380    540    v02    S      14:43    0:00    -bash
root    55    0.0        0.5     41    220    v03    S      14:43    0:00
/sbin/agetty 38400 tty3
root    56    0.0        0.5     41    220    v04    S      14:43    0:00
/sbin/agetty 38400 tty4
root    57    0.0        0.5     41    220    v05    S      14:43    0:00
/sbin/agetty 38400 tty5
root    58    0.0        0.5     41    220    v06    S      14:43    0:00
/sbin/agetty 38400 tty6
root    59    0.0        0.5     50    228    s00    S      14:43    0:00    gpm
-t ms
root    61    0.0        0.8    300    332    ?      S      14:43    0:00
/usr/local/etc/httpd/sr
root    78    0.0        1.3    445    528    v02    S      14:43    0:00    joe
test
root    88    0.0        0.5     81    224    v01    R      14:46    0:00    ps
uax

```

ps 可以列出系統已經載入的 process，我們來看看最上面一行的意思：

- USER：這是 process 的擁有者。
- PID：這是 process 的編號 (process ID)，編號從 0 開始到 65535 時會再回頭從 0 開始。
- %CPU：這是 process 現在佔用的 CPU 時間百分比。
- %MEM：這是 process 佔用的虛擬記憶體 (VM) 百分比。
- VSZ：這是 process 佔用的虛擬記憶體大小 (bytes)，虛擬記憶體的大小就是您的 DRAM 大小 再加上 SWAP 的大小，也就是可以執行程式的總記憶體大小。
- RSS：這是 process 用掉的真實記憶體 (DRAM) 大小，VSZ 減掉 RSS 就是這個 process 使用 SWAP 的大小。

- TT：這是 process 啟動的 tty。
- STAT：這是 process 現在的狀況，這個狀態的每一個字元都個有其意義。
- STARTED：這是 process 開始執行的時間。
- TIME：這是 process 到目前為止佔掉的 CPU 時間。
- COMMAND：這是 process 的名稱及參數。

5.3.22 top 指令 程序監視

top 可以動態監視程序執行狀態【Process state】，其每五秒鐘自動更新監視畫面，以顯示最新的情況。請見下框為例：

```
[root@net53 /root]# top
```

```
[root@net53 /root]# top

19 processes: 18 sleeping, 1 running, 0 zombie, 1 stopped
CPU states:  5.0% user,  0.5% system,  0.0% nice, 99.5% idle
Mem:  39128K av,  5344K used, 33784K free,  3552K shrd,  2212K buff3
Swap: 13100K av,    0K used, 13100K free

  PID USER  PRI  NI  SIZE  RES  SHRD  STAT  %CPU  %MEM  TIME
COMMAND
   90 root   25   0   89   312   344   R    0.5   0.7   0:00
top3
    1 root    1   0   44   232   312   S    0.0   0.5   0:00
init3
   53 root    2   0  380   540   560   S    0.0   1.3   0:00
-bash3
   54 root    3   0  380   540   560   S    0.0   1.3   0:00
-bash3
   25 root    1   0   60   228   296   S    0.0   0.5   0:00
/usr/sbin/crond -l10
    6 root    1   0   32   164   336   S    0.0   0.4   0:00
bdflush (daemon)3
    7 root    1   0   32   168   340   S    0.0   0.4   0:00
```

```

update (bdflush)3
  49  root    1   0  251  448  452  S      0.0    1.1    0:00
sendmail: accepting
  38  root    1   0   73  280  332  S      0.0    0.7    0:00
/usr/sbin/syslogd3
  40  root    1   0   44  240  320  S      0.0    0.6    0:00
/usr/sbin/klogd3
  42  root    1   0   72  268  324  S      0.0    0.6    0:00
/usr/sbin/inetd3
  44  root    1   0   64  216  312  S      0.0    0.5    0:00
/usr/sbin/lpd3
  61  root    1   0  300  332  524  S      0.0    0.8    0:00
/usr/local/etc/httpd
  78  root   15   0  453  544  496  S      0.0    1.3    0:00  joe
test3
  55  root    1   0   41  220  328  S      0.0    0.5    0:00
/sbin/agetty 38400 t
  56  root    1   0   41  220  328  S      0.0    0.5    0:00
/sbin/agetty 38400 t
  57  root    1   0   41  220  328  S      0.0    0.5    0:00
/sbin/agetty 38400 t
  58  root    1   0   41  220  328  S      0.0    0.5    0:00
/sbin/agetty 38400 t505

```

前四行系統顯示如下：

```

19 processes: 18 sleeping, 1 running, 0 zombie, 1 stopped
CPU states:  5.0% user,  5.0% system,  28.3% nice, 61.7% idle
Mem:  39128K av,  5344K used, 33784K free,  3552K shrd,  2212K buff
Swap: 13100K av,    64K used, 13100K free

```

- 總共有 19 個 process 正在執行，其中有 1 個正在執行中，有 18 個正在休息，有 1 個已經停止。
- CPU 有 5% 的時間正給使用者，有 5.0% 的時間給系統，有 28.3% 的時間在 interrupt，有 61.7% 的時間沒事發呆。
- 記憶體用了 39128K 當 active，2212K 當 Buffer，有 33784K 沒有用到。
- Swap 總共有 13100KB，用掉了 64MB。

其中 PRI 跟 NICE 在前面沒有介紹到，筆者在此說明一下。
在 UNIX 系統的 process 權限從最高的 -20 到最低的 20，一般程式的 NICE 值是 0。不過 NICE 值只是 kernel 排程時的一個參考，真正執行的優先順序是 PRI (Priority)，這個值跟 NICE 一樣也是越低越快。

在 Top 中讀者可以按【k】鍵再輸入「<signal> <PID>」來做出 kill 的動作，也可以按【r】鍵輸入「<nice_value> <PID>」來做出 renice 的動作，按【h】鍵閱讀 top 的說明，要結束正在執行的 top，您可以按下“q”鍵就可離開。

5.3.23 nohup 指令 使處理程序在登出後能繼續執行

nohup 最常用在使幕後的處理程序在登出後不被打擾而執行。讀者可以想像一下，假設有一個很大的檔案正被你用 ftp 傳檔中，這時你誘捕能在旁監視，當初後 ftp 傳檔就會終止，不登出又怕被人冒用帳號，這時就是用這個指令配合幕後處理程序來執行，使登出後的處理程序不備打斷。

nohup 常用語法：【nohup 指令名稱】

這時執行後的結果會放在家目錄下的 nohup.out 檔案中，也可設定放在別的檔案。

例如，執行 sh 並且將執行結果放在 iori 中！：

```
[root@net53 /root]# nohup sh > iori
```

那讀者也許會問使指令在幕後執行的方法為何？其實只要在所執行的指令後加上 & 這個幕後字元，這時系統就會將指令丟到幕後了，舉個例子吧！在幕後執行 find 指令去尋找 iori：

```
[root@net53 /root]# find / -name iori&
```

然後我們在使 find 指令在登出後已能繼續執行，所以我再配合 nohup 指令，順便將執行結果寫到 kyo 裡面。

```
[root@net53 /root]# nohup find > kyo
```

5.3.24 sync、reboot、halt 指令 重新啟動

sync 指令主要的目的就是將最新的資料，盡快從磁碟緩衝區寫入磁碟中，必免資料遺失。通常在關機之前都要做這個動作 3~4 次才行，以確保資料的完整

性。

reboot 指令就是將電腦重新開機，在重開之前，電腦會把一些還在執行的程式，先全部停止，然後將一些 mount 進來的設備，全部都 umount，等一切就緒後，就重新開機。

以上兩個指令，我們都是拿來一起互相配合使用，其用法為：

```
sync;sync;sync;reboot
```

當然我們也可以用手按住【Ctrl】+【Alt】+【Del】，這時也同樣的效果，但是筆者也不建議使用者們這門做，若這是在 MS-Windows 所學到的壞習慣，筆者建議使用者們千萬不要把這同樣的招式用在 Linux！

halt 指令所做的動作跟 reboot 一樣，只是 halt 在關閉 FreeBSD 系統之後，並不會重新啟動，兒會停在那裡等待使用者關閉電源，或是隨便按一個鍵重新啟動，因此如果您是經由遠端網路登陸，想重新啟動電腦，千萬不要下 halt 指令。

5.4 帳號管理

講到帳號管理可是每個系統管理最重要的課題，root 的職責是負責整個 Linux 系統的完整穩定運作，並且增加系統的安全性、過濾使用者的身份，新增使用者或刪除來路不明或會危害系統的使用者，並且安全的定義系統中使用者的權限，所以一個好的 root 必須對一個系統付出相當大的心力。另一種就是每一個使用者都有可能將其工作的內容或是一些機密的檔案放在 Linux 工作站上，所以各個檔案的或是目錄的歸屬和使用權都要有非常明確的規定，要不然機密的文件外洩或遭到破壞，這時所面對的可是相當大的損失喔！

因此每一個使用者自己的家目錄都應該要有合適的權限，才能維護自己的檔案安全，防止有被竊取或破壞的情事發生。所以帳號管理可說是相當重要，因此也要配合一些系統管理的指令才能確實做好 root 的職責。

5.4.1 useradd (或 adduser) 增加使用者帳號

其實建立帳號就是等於增加使用者，而增加使用者很可能是您管理 Linux 第一個必須面對的系統管理工作之一。現在我們使用 useradd 可以建立一個新

的帳號並配合學校教學實例分別探討。

useradd 常用的語法：【useradd 使用者帳號 -p 密碼 -g 群組 -d 指定家目錄位址】

-p 的意義是指定此帳號的密碼，若讀者未加此選項的話，則密碼則由系統內定之，以後再由系統管理者用 passwd 的指令設定密碼即可，若想由使用者設定自己的密碼，則加上此選項時，請將此選項的後面用雙引號將空白括起來，表示設定此帳號的密碼為無，以後可由使用者自行登入後，再自行建立密碼。

-g 是設定此使用者帳號是要定義在哪個群組，若未加此選項指定群組的話，則系統會同時產生一個跟帳號同名的群組。

-d 的意義是要指定所要建立的家目錄的位址，若未加此選項的話，系統會再/home/目錄下開一個跟使用者同名的家目錄。

例如：

```
[root@net53 /root]# useradd iori -p "" -g root
```

執行這個指令，你會發現系統已經幫你建立好一個 iori 帳號，群組隸屬於 root 並且建立一個/home/iori/的家目錄，然而我們用 iori 的帳號登入時，會發現系統並不會詢問你密碼而直經進入家目錄，因為我們已經用雙引號將密碼空白了。讀者或許會問，若我在-p 選項後面加上我要設定的密碼可以嗎？答案是不行的，雖然說理應可行但當你用此密碼時還是不會被接受的。最好的方法還是用 root 設定密碼，或用上述的範例，然後用新設定的帳號進入後自行設定密碼！

```
[root@net53 /root]# useradd kyo
```

使用上面的指令，就會單純的建造一個 kyo 帳號，密碼為系統內定之，以後必須用 root 設定密碼，而群組隸屬於 kyo，家目錄在/home/kyo/下！

```
[root@net53 /root]# useradd kyo -d /home/iori/kyo
```

使用上面的指令，就會單純的建造一個 kyo 帳號，密碼為系統內定之，以後必須用 root 設定密碼，而群組隸屬於 kyo，家目錄在/home/iori/kyo/下，也就是說會在已存在的目錄下建一個新的目錄，作為 kyo 的家目錄。

5.4.2 passwd 設定使用者帳號密碼

剛剛我們一直說設定密碼，現在我們就來介紹設定密碼的方式吧！

Passwd 常用的語法：【passwd (使用者帳號)】

一般來說都是用 root 的方式來設定一般使用者的密碼！所以舉個例：

```
[root@net53 /root]# passwd iori →設定 iori 的密碼
Changing password for user iori
New UNIX password: →鍵入新的密碼，這時太短或太簡單或跟以前的太相似，系統會告訴你重新設一個。
Retype new UNIX password: →重複你所打的密碼，以便確認。
passwd: all authentication tokens updated successfully →成功更改密碼了！
```

以上用 root 設定別人的密碼，效力可說是非常大，不管你是新設定，或者之前已經設定過了，用 root 設定別人的密碼，都會強制重新設定，因此這最適合用來處理一些忘記密碼的豬頭使用者的場合(筆者也當過豬頭！cc)。

那要改自己的該怎門辦？其實只要用該帳號進入後，直接鍵入 passwd 即可，不一樣的是會要求你再輸入一次之前的密碼，以免被有心人竄改。如下框：

```
[a8630257@net53 a8630257]$ passwd
Changing password for a8630257
(current) UNIX password: →請在輸入之前的密碼。
New UNIX password:
BAD PASSWORD: is too similiar to the old one →跟以前的太像了，請換一個。
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
[a8630257@net53 a8630257]$
```

每當我們增加完使用者帳號後，系統都會在 /etc/passwd 檔中新增一使用者的資料。我們可以執行 vipw 指令去觀看或著編輯(此指令 root 才能用)，不過前提是你會使用 vi 編輯器(你也可以用其他的編輯器如 joe)並且你必須是 root 身份才能修改喔！一般的使用者可以用其他的編輯器去觀看/etc/passwd 檔，但是看得到吃不到。以後筆者會介紹簡單的編輯方法。在 /etc/passwd 檔但中的使用者資料的格式如下：

```
a8530257 : x : 1001 : 200 : : 0 : 0 : a8530257 : /home/8630257 : /bin/bash
```

① ② ③ ④⑤⑥⑦ ⑧ ⑨ ⑩

在上面的格式中共有七個欄位，而每個欄位都是用冒號來格開，其各個欄位說明如下：

- ① 使用者帳號名字【Login ID】，也就是 user ID，使用者是以此 ID 進入系統。
- ② 密碼欄【Password】，暫時保留空格的時候，是以便使用者自行設定密碼，若有奇怪的文字符號，則為原密碼加以編碼後的密碼內容。
- ③ 使用者的系統編號【UID】，此編號是不可以重覆，而且系統編號是從 0 到 65535，此處的 UID 為 1001，一般使用者的 UID 序號，Slackware Linux 是由 501 開始，而 Red Hat Linux 預設是由 500 開始。在 Red Hat 6.0 中，useradd 已經可以配合 `/etc/login.defs` 檔案，UID 之預設啟始值，可由 `UID_MIN` 來管理。
- ④ 使用者所屬 group 的編號【Munber group】，由於我們是在建立學生帳號，而且在一開始時已經在 `/etc/group` 內定義了 student 群組，故所有屬於 student 群組的 GID 為 200。
- ⑤ 設定您的 Login Class，通常也是空著。
- ⑥ 強迫使用者更改密碼的時間，0 表示關閉此一功能。
- ⑦ 帳號的有效日期，0 表示關閉此一功能。
- ⑧ 使用者的全名【Full name】，以及電話號碼，通常可以不用設定。
- ⑨ 定義該使用者的使用家目錄【Home directory】，這個欄位設定會關係到您後面的建立目錄步驟，故要一致。
- ⑩ 當使用者進入 Linux 時，內定啟動的 shell【login shell】，預設在此我們是

設定為 bash。

這時讀者可能會疑問說，若 /etc/passwd 每個人都看得到，那密碼不是被人看光了嗎？這點到不用擔心，因為看到的都是 x 這個符號，因為已經被隱藏了，所以只看到 x！若想解除隱藏時，可以用 root 的身份下 **pwunconv** 這個指令，這時我們在用 vipw 觀看，就會發現 2 號欄位已經不是 x 了！但這也不代表真正的密碼，因為~cc！我們看到的已經是編碼過後的密碼了！若想再度隱藏只要執行 **pwconv** 即可！你也可以用等會會介紹的指令 chomd 來更改/etc/passwd 此檔的權限，改成 711，這下子除了 root 外其他人都看不到也吃不到了！夠狠吧！

當然啦！假若有一天您的使用者跑去找您，說他忘記密碼，所以無法進入 Linux 系統，您的解決方法也可以下 vipw 指令，把那位使用者密碼刪除，這樣就可以了，那位使用者就可以不用密碼，而直接進入 Linux 內，然後在重新設定密碼就可。

5.4.2.1 vi 編輯器的簡易使用法

vi 是最古老的編輯器，它的功能性與方便性，已經無法何其他的後起之秀相比，但它的功能和編輯文件的安全性是有目共睹的，而且只要有 Unix 就會有它，現在我們稍微的介紹只老而彌堅的編輯程式吧！以免讀者執行 vipw 後，卻不知怎口離開！

若要啟動 vi 編輯檔案，則在提示符號後輸入

vi 檔案名稱

之後便會進入 vi 編輯的程式了，在 vi 中，我們可以輸入資料，做一般文書處理的工作，這時我們所輸入的所有字元都會成為文件的一部份，因為這時我們是在插入模式中。當我們打完文件時，要離開或存檔時，就必須進入命令模式進行命令的執行。

有用過 pe2 或者漢書的讀者應該明白筆者的意思，然而 vi 的初學者常常搞不清楚自己身處於什口模式，畢竟 vi 沒有跟 pe2 或漢書一樣有那口明顯的命令列存在，若插入模式要進入命令模式，則按一下【Esc】鍵即可。若命令進插入模式，則可按【i】、【a】、【o】、【O】其中之一進入插入模式（此四個按鍵，所插入的位址不同），一般筆者都按【Insert】進入插入模式，然後再按【Delete】刪除欲修改的字元，然後再填入正確的字元即可。若有捲動螢幕的必要則按【Ctrl】+【b】可往上捲一頁，【Ctrl】+【f】和【Ctrl】+【d】可以往下捲動一頁和半頁。

若有時當機，在該文件目錄下的提示符號鍵入 **vi -r 檔案名稱**，就可以挽救剛剛的文件了。若不小心修改到不用修改的字元，這時可在命令列模式下按 **u** 即可回覆上一個動作（只有一個動作而已喔！）。

若要離開 vi，可在命令列模式下鍵入：**q**，請注意包含帽號喔！這時若你已經更動過文件了，你用：**q** 是沒用的，必須用：**q!**，加上一個驚嘆號強制放棄修改內容，保持檔案原樣。

那要怎門存檔並且離開？同樣的在命令列下按：**wq**，這時就會存檔並且離開了，你也可以按：**x**，也有同樣的效果！

以上只是筆者所介紹的 vi 基本用法，其他還有很多，讀者可自行參閱其他的書籍或網路上的文件。

5.4.3 groupadd 增加使用者群組

增加群組帳號 (group account)，可以利用群組可以讓一群人對某些檔案有共同的存取權限，在 RedHat 系統底下，當新增一位使用者的時候，useradd 指令也會同時新增一個與使用者帳號同名的群組，作為使用者的預設群組，有需要時可以再加入其他的群組，然而一個使用者可以隸屬於多個不同的群組，也就是說可以一對多，因此同一個帳號可以使用多個群組所設定的權限。

groupadd 常用的語法：【groupadd 群組名稱】

例如：

```
[root@net53 /root]# groupadd student
```

這時就會增加一個名為 student 的群組了，而且每當我們新增群組則也會在 /etc/group 檔案中裡新增一筆資料，我們可以切換目錄到 /etc/ 目錄下然後執行 **vi group** 去瞧瞧。我們可以發現群組資料格式如下：

```
student:x:500:a8630257,a8630222,a8630245
```

① ② ③ ④

- ① 群組名稱。
- ② 群組密碼欄，群組密碼可以用來作群組的管理。
- ③ 群組系統編號【GID】，每個群組都有一個特定的號碼，且號碼都不能重複，而且系統編號是從 0 到 65535，此處的 GID 為 500。
- ④ 在群組中的成員，若想增加群組的成員，可在這裡增加，不過請記得用逗號隔開！

從這裡我們可以看出使用群組也有分類的意味，當您要建立學生帳號的話

【全日間電機科再加上全夜間電機科學生共有 844 人】，在使用 adduser 之前，要先製造出 student 一個群組，來提供給學生來使用，然後在建立學生的帳號，這時當然也不行用 useradd 這個指令，因為像筆者所在的電機科也有 844 人耶！豈不是加到死！所以 useradd 這指令適合初學者使用，當您有少數的使用者要新增的話，也是用這個指令。等會筆者會在介紹自動化增加使用者帳號的指令！

如果您在 /etc/group 檔裡，將某個使用者設定為同屬 disk 與 tty 這兩個群組，那麼這位使用者，就有權限可以存取 CDROM 與 tty?? 了。也就是說，他可以執行 xplaycd 之類的程式，來聽音樂 CD。

5.4.4 gpasswd 管理群組與群組密碼 和 newgrp 登入到新的群組

當我們要將某個使用者只新增到某群組或設定群組密碼時，通常都是用這個指令，該使用者再配合 newgrp 來重新登入新的群組，這樣才會有此新群組的權限，當然我們也可以修改 /etc/group 檔來增加群組的使用者。

gpasswd 常用語法：【gpasswd [-adr] 使用者 群組名稱】

- a 增加使用者到群組中
- d 從群組中刪除某一個使用者
- r 取消群組密碼

newgrp 常用語法：【newgrp 群組名稱】

例如：

```
[root@net53 /root]# gpasswd -a iori root
Adding user iori to group root
[root@net53 /root]# exit → 登出
[iori@net53 /iori]# newgrp root → iori 使用者登入後執行
```

這時 iori 已經增加到 root 的群組中了，但必須注意的是增加後必須執行 newgrp root，這樣 iori 才會有 root 群組的權限。現在我們在來示範刪除群組中的使用者。

```
[root@net53 /root]# gpasswd -d iori root
Removing user iori from group root
```

我們可以從系統訊息得知，iori 已從 root 群組移除了。

假設我們之前有設定群組的密碼，我們可以用 -r 選項取消。

```
[root@net53 /root]# gpasswd root → 設定群組密碼
Changing the password for group root
New Password: → 鍵入新的密碼
Re-enter new password: → 重複密碼以便確認
[root@net53 /root]# gpasswd -r root → 取消群組密碼
```

通常群組密碼是用來管理使用者重新登入群組用的，也就是說如果你有設定群組密碼，該使用者用 newgrp 指令時，就必須鍵入密碼才能登入該群組。

5.4.5 userdel 刪除使用者帳號

刪除使用者可不像建造使用者帳號容易，通常我們使用 userdel 的語法如下：

userdel 常用的語法：【userdel 帳號名稱】

通常我們下這個指令後，系統只會將/etc/passwd 檔案中的此帳號資料刪除，但是並不會將其家目錄刪除，所以我們只好親自到/home 下，執行 **rm -rf 帳號目錄**，將其家目錄刪除。若此使用者有使用電子郵件的話，則 Linux 會將其信箱以使用者的名稱寫成一個文字檔放在 /var/spool/mail/，所以我們必須到 /var/spool/mail/ 下將其信箱殺掉。

5.4.6 groupdel 刪除群組

刪除群組相當簡單，只要照著語法做就好了！

groupdel 常用的語法：【groupdel 帳號名稱】

5.5 自動建立、刪除帳號

剛剛我們說過要建立一大批的使用者，可不是單純用 adduser 這個指令就可以輕易地完成的，好在 Red Hat 提供了 newusers 和 chpasswd 這兩道增加大量帳號和變更密碼的指令。

使用 newusers 之前，我們必須執行 **pwunconv** 將 /etc/passwd 檔案的隱藏功能解除，以免失敗，然後我們必須建造一份文字檔，裡面填寫我們想要做的帳號資料，以便 newusers 去讀取，再用批次的方式建立帳號。此文

件檔的帳號資料格式跟我們用 `vipw` 所看到得格式相同。

假設筆者要為學生們建立一堆帳號，因此筆者首先用 `groupadd` 建立一個 `student` 的群組，然後再去查看此群組的 `GID` 為何，以便帳號資料的填寫，這樣才能將學生的帳號確實加入 `student` 的群組。如下：

```
[root@net53 /root]# groupadd student
```

然後再用 `joe` 或 `vi` 編輯器去建造一份文字檔，裡面填入跟 `/etc/passwd` 檔案一樣的格式。以筆者為例如下：

```
[root@net53 /root]# joe stuacc.txt → 建立新的文字檔
→ 在檔案內容中填入以下筆者要建立的帳號格式，但密碼欄空白，以後由帳號使用者自行建立
或者由 root 用 chpasswd 指令分配之
a8630408::1001:200:a8630408:/home/a8630408:/bin/bash
a8630410::1002:200:a8630410:/home/a8630410:/bin/bash
a8630417::1003:200:a8630417:/home/a8630417:/bin/bash
a8630432::1004:200:a8630432:/home/a8630432:/bin/bash
a8630436::1005:200:a8630436:/home/a8630436:/bin/bash
a8630439::1006:200:a8630439:/home/a8630439:/bin/bash
→ 填完後用 【Ctrl】 + 【k】 + 【x】 存檔並離開。(關於 joe 編輯器的使用，讀者可按 【Ctrl】 +
【k】 + 【h】 看說明文件)
[root@net53 /root]# newusers < stuacc.txt → 把檔案的資料丟給 newusers 做批次處理
```

開始批次建立的時候，若你的 `/etc/passwd` 檔案已經有上述的帳號，則 `newusers` 會將帳號資料給更新並不會重複。現在我們在用 `chpasswd` 指令將學生的密碼給填入。不過首先也是先建立一份文字檔，裡面填入筆者要給學生的密碼。其實最好是讓學生以自己的帳號進入後，再用 `passwd` 自行設定即可，不然一一告知學生密碼為何，也是件麻煩事！
文件內容格式如下：

```
[root@net53 /root]# joe stupass.txt → 建立文件檔
→ 填入格式為 使用者：密碼
a8630408:464456
a8630410:fhfgff
a8630417:qwqtqt
a8630432:xnv,,n
a8630436:yuiyui
a8630439:m,m,.
```

在輸入密碼時，只要輸入一般想要的密碼即可，不需要經過特別加密。若是你的/etc/passwd 檔案中沒有文件檔中的帳號，則在建立過程會發生中斷。

5.5 安裝 Quota 使用者硬碟空間管理

何謂 Quota？Quota 它是可幫您管理硬碟空間的使用，以亞東電機科的 Linux 主機為例，因為全電機科共有八百多人，倘若有幾位學生把 Linux 剩餘的硬碟空間給使用，那麼當其他的學生要儲存資料時，就會因硬碟空間不夠而不能儲存，故我們就使用 Quota 來監視每個使用者，當使用者要儲存資料時，Quota 會檢查該資料的大小是否有超過規定每人能儲存的容量，若有超過時，則 Quota 就不會讓該資料儲存於硬碟中，這樣可以預防學生亂拷貝東西而佔用可貴的硬碟空間，而我們在 Linux 中設定每個學生可用 5Mbytes 的空間，且當各人的資料累積到 4Mbytes 時，Quota 會有訊息通知該學生說他的使用容量快要滿了，這樣該學生就要把不必要的資料刪除，這樣就能夠真正的運用硬碟空間，且不會浪費。

安裝及設定 Quota，我們把它分成幾個步驟來完成：

- 打開檔案系統的 Quota 功能
- 開啟 Quota 的功能
- 設定 user & group 的 Quota 限制

5.5.1 打開檔案系統的 Quota 功能

要開啟 Quota 的功能必須先修改/etc 內的 fstab 檔案，接下來要，將欲加上的硬碟空間限制的分割，增加 userquota 這個參數，通常檔案系統的設定

都是用這個檔案來設定的，由於我們要開啟 Quota 功能，而且我們是用在學生帳號上，故我們將目錄設定到/home/student，並且也增加了兩個參數用來打開 Quota，這兩個參數為 **usrquota**、**grpquota**，至於檔案系統的檔案格式，當然是為 ext2【Linux 常用格式】，底下的範例是我們在 net53 上的/etc/fstab 設定，請見下圖 1 所示：

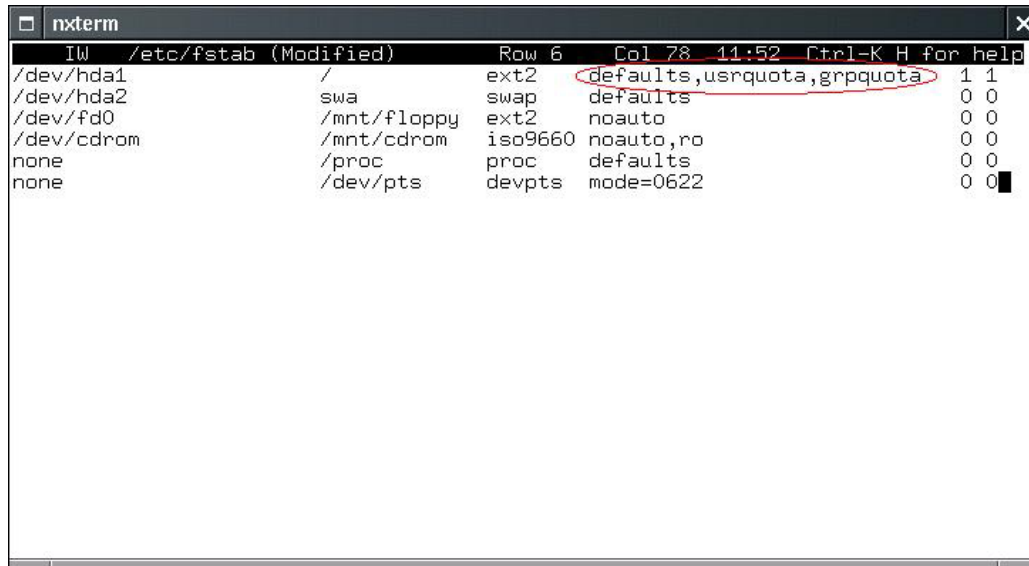


圖 1

5.5.2 開啟 Quota 功能

如果是第一次安裝 Quota，則必須要先到要設定 Quota 的目錄中，在本例為/home/student，執行 **quotacheck -avug**，這樣子 Quota 就開始檢查/home/student下的所有目錄【此目錄為使用者目錄】，第一次在做這個掃描動作的時後會比較慢。掃描完後會在 / 下產生 quota.group 和 quota.user 這兩個檔。

- a：代表掃描全部的磁碟
- v：代表在掃描時顯時掃描過程的訊息
- u：代表掃描紀錄磁碟上個使用者所使用的檔案和目錄
- g：代表掃描紀錄磁碟上每個群組所使用的檔案和目錄

由於一開機時，我們便要使用 Quota 來管理硬碟空間，故我們到/etc 底下，有一個檔案叫作 rc.local，此檔的作用是當 FreeBSD 開機後會自動去讀取 rc.local 的檔案，並且執行該檔案的內容，此功能相當於 Dos 中的 autoexec.bat，在/etc/rc.local 內再加入二行設定為：

echo "Quota turn on...."

quotaon -avug

這樣以後只要一開機就會自動的去啟動 Quota 的功能。

5.5.3 設定 user & group 的 Quota 限制

在設定 user 的 Quota 限制，是使用

edquota -u username

來設定 user Quota 的限制，而要設定 group 的 Quota 限制，是使用

edquota -g groupname

來設定的，通常我們的作法是先設定單一 user 的 Quota 後，再設定所有的使用者，使用剛才所設定的單一 user 的 Quota 值，這樣每一個使用者的 Quota 限制就會相同。假如我們先設定一位學生帳號 a8530257 的 Quota，可用

```
edquota -u a 8530257
```

執行後會出現一個 vi 【Unix & Linux 的基本文書編輯器】畫面，其畫面如下所示：

```
Quotas for user a8530257 :
/usr: blocks in use : 8 , limits ( soft = 4000 , hard = 5000 )
      inodes in use : 8 , limits ( soft = 200 , hard =
300 )
```

在上面的框框中，所要更改的只有 soft 和 hard 項中的數字，而 blocks（容量）數字是以「Kbyte」為單位，inodes（檔案）數字是以「個」為單位，其中 soft 是指當該使用者總共所儲存的資料已超過 4000K，或是檔案已超過 200 個時，Quota 會出現一個訊息通知該使用者，說能夠所儲存的空間已經快要滿了，必須要清除一些已不用的資料，這樣才會有多餘的空間能讓該使用者儲存資料，七日後如果仍無法改善的話，將禁止寫入任何檔案。其中 hard 是指該使用者當使用硬碟到達 5000K，或是檔案超過 300 個時，則 Quota 就不會讓該使用者儲存該資料，立刻禁止寫入檔案。除非使用者去清除一些沒用的資料，騰出足夠的空間才能儲存。

假如您想要關閉 Quota 功能，您只要將 soft、hard 的值，改成 0 就可以

關閉 Quota。在設定完單一使用者的 Quota 後，我們就要把 user1、user2、user3 使用者都設定成為剛剛所設定單一使用者的 Quota 值，可用

```
edquota -p user_source user1_target,user2_target,user3_target
```

其中 user_source 是指要 copy 的來源 user，而 user_target 是指要 copy 的目的 user。如果要一次來完成所有學生帳號的 Quota 設定，我們的作法是先切換到使用者目錄所在後，再執行

```
edquota -p a8530257 *
```

這樣所有使用者的 Quota 值，就會與 a8530257 的 Quota 值相同。而 group 的設定和 user 也是一樣的，假設我們要設 student 這個 group 的 quota 設定，你可以這樣執行：

```
edquota -g student
```

但是你必須注意一點是【群組的 quota 必須大於這個群組底下的所有使用者的 quota 的總和】，如果小於的話，則此群組的所有使用者都不能在自己的下寫入檔案喔！所以請自行估算一下大概要給這個群組多大的空間限制，這點可是非常重要的。

假若您想要查看一位 user 或一個 group 的 Quota 設定，您可以使用：

```
quota [-qv] -u username
```

```
quota [-qv] -g groupname
```

上面語法中，“-q 參數”為顯示當已超過您所設定的 soft，以及 hard 已到限制；“-v 參數”是使用於您還沒有啟動檔案系統的 Quota，或者使用於您已啟動 Quota，但還沒有分配任何的 blocks。

範例如下框所示：

```
[iori@net53 iori]$ quota -vu iori
Disk quotas for user iori (uid 502):
Filesystem      blocks  quota  limit  grace  files  quota  limit  grace
/dev/hda2        1032   4000  5000           63   300   400
net53:(pid368)    0       0     0              0     0     0
```

在此範例中,iori 這個使用者的總容量 5000K,而當資料已經儲存到 4000K 時則會出現訊息通知使用者所儲存的空間快要滿了,而 iori 這個使用者所使用的空間還沒超過 quota 所設定的範圍,所以 iori 還有空間可以使用。

系統管理者或是具有 operator group 的使用者,可以執行 **repquota -v -a | more** 來查詢所有使用者的 QUOTA 使用情形。如下框所示：

```
[root@net58 /root]# repquota -v -a | more
*** Report for group quotas on /dev/hda2 (/)

          Block limits                File limits
User      used  soft  hard  grace  used  soft  hard  grace
root     -- 1177739    0    0      78977    0    0
bin      --   247      0    0         9    0    0
daemon  --   22      0    0        18    0    0
sys      --    0      0    0        282    0    0
tty      --  132      0    0        653    0    0
disk     --    0      0    0       3490    0    0
lp       --   76      0    0         4    0    0
floppy   --    0      0    0        20    0    0
```

5.5.4 Quota 注意事項

在建立使用者 quota 硬碟容量配額時,有一個問題時常發生,即是 UID 和使用者帳號搞混了,問題發生的背景是一般教育單位常利用免費的 FreeBSD 或 Linux 提供學生電子信箱和個人首頁服務,此時系統管理者大多會使用學生的學號來當作帳號,方便日後的管理,如果不巧的是學生的學號全部均是阿拉伯數字的話,問題便發生了。當我們使用 **edquota -u username** 來設定使用者 quota 硬碟容量配額時,如果 username 為阿拉伯數字的話則系統設定的是 UID (使用者識別號),而非該位使用者!所以筆者建議不妨在全部均是阿拉伯數字的學生學號前面加一個英文字母來當作帳號,避開此一問題。

有時我們可能新增加使用者後,卻忘了同時限制使用者 quota 硬碟容量配額,使得系統有崩潰的危機,所以日常的檢查是必需的,首先我們切換至使用者有使用者的目錄中,一般是 /home,

執行 **cd /home ; quota -u * | more**, 檢查一下是否有漏網之魚。

接下來執行 `cd /home ; quota -q -v -u * | more`，檢查一下超過提醒配額的使用者。

5.6 基本系統管理指令介紹

系統管理指令基本上是 root 的必學課程，但筆者只說明一些重要且基本的系統指令，若得者想知道到更多的指令，讀者可自行購買一寫有關 Unix 的指令介紹書籍。

5.6.1 chmod 指令 變更檔案使用權限

有時候您在 Linux 作業系統中檔案，要分享一些系統管理者的重要資源，但並不想分享太多的權限給使用者或同一群組的使用者，我們都將其檔案使用權限改設為【`-rw-r--r--`】，只可以做讀取的動作，可以有效防止檔案被修改。而您要修改的地方是劃底線的地方，通常第一個字元是不能修改，而在本例中，文件檔通常都不可寫入與執行，所以要把 wx 去除，修改的方法有兩種如下：

1. chmod 常用語法：**【chmod 權限 檔案/目錄】**

2. chmod 另一語法：**【chmod <-R> 『權限』 < 檔案/目錄>】**

在上面第一種方法是用 9 個 bit 來描述，剛好一個字元對應一個 bit，而其中每一個 bit 為 0 代表該字元功能 disable，為 1 代表該字元功能 enable。在語法中的權限通常是用八進位來表示，表示的方法是每 3 個 bit 為單位【`rwX` 為一組】，然後把 3 個 bit 轉成八進位，如此就有 3 個八進位，這就是權限；例如：輸入欄位的數字為 647，則 6 為【所有者】的權限，4 為【群組】的權限，7 為【其他使用者】的權限，然而【可讀】的權限代表數字為 4，【可寫】的權限代表數字為 2，的權限代表數字為 1，所以【所有者】的權限是 4+2=6，表示有【可讀】、【可寫】的權限，而沒有【可執行】的權限。

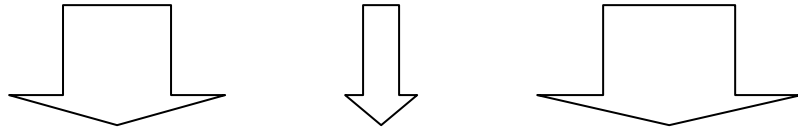
r w X	r w X	r w X
4+ 2+ 1	4+2+ 1	4+ 2+ 1
『檔案所有人』	『同群組使用者』	『其它使用者』

假設欲得到的權限為檔案所有人可讀寫，但不可執行。同群組使用者可讀，但不可寫及執行。其他使用者可讀寫並執行，則在本例中，指令的下法為：

```
chmod 647 book
```

而第二種方法，權限則不是使用 bit 方式來描述，而是用加減法來描述，其中 -R 表示包含目錄下的所有子目錄與檔案，權限如下圖所示：

「 u , g , o 」 「 + , - 」 「 r , w , x , s 」



分別代表「Owner, Group, Other」「增加, 移除」「read, write, execute, setid」

在本例題中，指令如下：

```
chmod u-x book
chmod g-wx book
chmod o-wx book
```

【我們把使用者的 x 和同群組使用者 wx 及其他使用者 wx 屬性去除，其餘按照原樣不動】

此外 u、g、o 也可用 a 來代替，也既是說 a 是 u+g+o 的總稱，例如：

```
chmod a+rwx book
```

【這表示每個使用者、群組都可以對 book 做讀、寫、執行的權利】

在此您也可以設定同群組使用者的使用權限【第二組 rwx】，讓同群組的使用者能否來讀、寫、執行檔案等，而其它使用者【第三組 rwx】也是一樣的設定。該指令除了 root 可使用以外，還有該檔案的擁有人也可使用。

但是有時候，我們會有一些特別的需要，例如說：setuid、setgid，那ㄇ chmod 的權限數字就會由 3 位數變為 4 位數了，4 位數的第一個數字代表除了基本的 rwx 權限外的特殊權限設定。這時 0 代表無權限制，跟普通的 3 位數的權限定一樣，1 代表設定 sticky 位元(必須跟 u 字元合用)，2 代表 setgid，讓任何人都可以擁有群組的身分來執行，4 代表 setuid，讓任何人都可以擁有使用者的身分來執行。

例如：

```
[root@net53 /root]# chmod 0647 filename
```

說明：表示普通的權限設定。

```
[root@net53 /root]# chmod 2647 filename
```

說明：做 setgid 特殊權限設定，用 ls -la 會發現設定後的 group 的 x 執行權限變成大的 S。使用【chmod g+s filename】也有同樣的效果。

```
[root@net53 /root]# chmod 4647 filename
```

5.6.2 chown 指令 變更檔案的擁有人及所屬群組

chown 指令只有 root 帳號才能使用，其指令的語法如下：

chown 常用語法：【chown 擁有人.擁有人所屬群組 檔案/目錄】

例如：有個帳號為 a8530257，其所屬群組為 student，想將一個文件檔【原本是 root 的】轉給 a8530614 使用，其作法是先將文件檔拷貝給 a8530257 的所在目錄內，然後再將檔案的擁有人及所屬的群組改成 a8530257 的，如此一來，a8530257 的使用者就有權使用該文件檔，其修改方式如下：

```
chown a8530257 .student test
```

即可。

5.6.3 chgrp 指令 變更檔案的所屬群組

這個指令和 chown 用法差不多，但是其功能是專對於檔案的群組改成令一個群組。

chgrp 常用語法：【chgrp 群組名稱 檔案/目錄】

例如，我要將 iori 這個檔案群組更改成 root，這時我只要輸入：

```
chgrp root iori
```

5.6.4 chsh 指令 變更使用者所用的 shell

雖然說在 root 新增使用者的同時，已經預設將好用的 bash 指定給使用者了，但是 shell 的種類可是非常多，也許有別的使用者會用到別的 shell，這時就可以

用 chsh 來變更所用的 shell，這個指令也是每個使用者都可以用的。

chsh 常用語法：【chsh】

例如我們要更改 shell，則在提示符號下直接輸入 chsh，然後再鍵入自己的密碼和欲改成 shell 路徑即可。

```
[root@net53 /root]# chsh
Changing shell for root.
New shell [/bin/bash]: → 輸入欲改成的 shell 路徑
[root@net53 /root]#
```

5.6.5 chfn 指令 修改使用者資料

建立個人的資料是培養良好的人際關係的第一步，因此建立好帳號後我們要為此帳號輸入其基本的個人資料。

chfn 常用語法：【chfn 使用者名稱】

在指令後面若沒有加上使用者名稱，則改自己的資料喔！~您可以用 chfn 指令修改使用者全名、公司電話號碼等等。如下框所示：

```
[root@net53 /root]# chfn
Changing finger information for root
Name []: iori → 輸入姓名
Office []: ee.oit → 公司行號或位址
Office Phone []: 0939-61565X → 輸入公司的電話
Home Phone []: 0939-61565X → 家裡的電話

Finger information changed.
[root@net53 /root]#
```

5.6.6 finger 指令 查詢使用者的基本資料

用這個指令可以查詢使用者的資料，若在之前這個使用者沒有用 chfn 設定過個人資料的話，查起來就是空空如也。

finger 常用語法：【finger 使用者名稱】

在指令後面若沒有加上使用者名稱，則顯示所有正在線上使用者的名稱讓你挑一個來查詢！例如我們查詢剛剛所設定好的 root 的資料。

```
[root@net53 /root]# finger root
Login: root      Name:iori
Directory: /root  Shell: /bin/bash
Never logged in.
No mail.
No Plan.
[root@net53 /root]##
```

5.6.7 su 指令 變成超級使用者

說到 root 還有一個指令是要知曉的，就是 su 指令，su 的功能是能讓一個普通的使用者能夠轉變為 root 身份，一樣享有系統最高的使用權限，或者暫時化身成其他使用者，在 Red hat 中，root 也無法使用遠端登入，因此如果讀者要想由遠端登入，就得使用 su 指令，但是當您要轉變為 root 身份時，您還是必須要有 root 的 password，才能轉變成功，為了考慮系統安全問題，所以不能用 root 來進行 telnet 的 login 動作，如果您有不得已的「苦衷」，必須用 root 直接 telnet 及 login，那麼請在 /etc/securetty 檔案裡多加上一些 tty 的 device 名稱，如 tty0、ttyS0。

su 常用語法：【su - 使用者名稱】

"su" 後面沒有給任何參數的時候，預設就是化身成為 root，這個時候系統會要求你輸入 root 的密碼，只要輸入密碼正確，就會擁有 root 身分了，假如有加一個使用者名稱作為參數，則會化身成這個指定的使用者。另外，"su" 的後面加一個減號的話，表示啟動的環境會跟真的用 root 帳號登入的一樣，不加減號則會維持目前的環境，當然若暫時化成其他的使用者也加上減號的話，也會將目錄環境轉成該使用者的家目錄。

例如：

```
[water@net53 water]$ su - →轉成超級使用者並改變目錄
Password: →輸入密碼
[root@net53 /root]# exit →離開此身份，轉回原來的身份
[water@net53 water]$ su - a8630257 →再轉成另一個普通使用者，並改變目錄
Password:
[a8630257@net53 a8630257]$ logout →功能同 exit
```

要是你發現用 su 轉成 root 的身分時，不能用增加帳號或群組等指令時，執行 echo \$PATH，來看看 PATH 環境參數有無 /usr/sbin/ 這個路徑，若沒有的話請修改你家目錄下的 shell 起始設定檔將 PATH 的參數增加這個路徑，始可用增加帳號或群組等指令，這是因為這個目錄放了很多常用的指令，不過大部分都是給 root 用的，因此在指令的收循路徑若沒有此路徑的話，你執行的時候 shell 就會告訴你沒有這個指令！

5.6.8 crontab 指令 定時執行程式

假設一個系統管理者每天都要做同樣的事情，如刪除一些暫存檔案啦~等等，諸如此類做久了會煩的的動作，你可以利用此指令幫你定時做好你要做的事，可以說非常方便。

Red Hat Linux 裡，較特殊的是，系統的 crontab 以「更加強大且具有彈性的方式」存在於 /etc 目錄下。藉用了來自於 Debian 的 run-parts 構想，Red Hat Linux 預設使用 hourly、daily、weekly、monthly 等，不同時間的 cron 設定，您可以視自己的需要，為系統增加適當的 crontab 排程。首先，您應該發現到 cron daemon 已經隨著系統啟動而執行待命，crond 若未啟動的話，您得查看 vixie-cron 此一套件程式是否安裝無誤。接著，在 crond 的 manpage 中提到，除了使用者各自的 crontab 排程檔會以自己的帳號為名「存放」在 /var/spool/cron 目錄下，crond 也會檢查 /etc/crontab 檔案的狀況，不過，此檔案是歸另一個套件程式所管理。然而我們現在所介紹的指令就是要教使用者設定自己的工作排程，好讓 crond 去幫你執行。

crontab 常用語法：【crontab 選項】

- e：編輯目前的 crontab 排程檔案，排入你想要定期執行的工作。
- l：列出目前 crontab 檔案中所定義的工作
- r：將存在於 /var/spool/cron 的使用者排程檔移除。

-d：刪除 crontab 檔案中所定義的工作。

在我們使用 crontab 的功能前，我們必須建立一份工作排程檔，這時我們可以用 joe 或 vi 建造一份。在檔案的內容中我們必須依照一定的格式寫出我們要讓 crond 幫我們做的工作。然而每個使用者只有一個工作排程檔，當我用 crontab 執行此檔時，這個檔案就會以使用者帳號的名稱存在於 /var/spool/cron 目錄下，比如說使用者 iori 建立一份名為 work 的排程檔，然後我們在執行 crontab work，這時就會在 /var/spool/cron 目錄下產生一個名為 iori 的排程檔，供 crond 執行用。

然而我們也可以直接執行 crontab -e 這個指令編寫排程，而不先建立一份排程檔，這是最常用的。然而編寫時是用 vi 編輯器編寫。編寫的格式如下：

分	時	日	月	星期	你所要執行的指令
分：0~59					
時：0~23					
日：1~31					
月：1~12					
星期：0~6 (0 代表星期日，其餘類推)					

例如：

執行 crontab -e 後，會出現 vi 的編輯畫面，然後依照自己的需要填入

```
1 5 3 * * 1-5 find $HOME -name core 2 > /dev/null | rm -f
```

這就代表每星期 1 到 5 的凌晨 3 點 15 分就會刪除使用者家目錄下的 core 檔案，而時間欄位以 * 表示，就是表示該欄位代表時間的任何時刻，如上面的範例：日和月都用 * 表示，這就代表每個月的每一天！

5.6.9 id 指令 查詢 id 碼

使用 id 指令，相當簡單只要在提示符號下輸入 id 即可，就會顯示目前使用者的所有 id 碼。例如：

```
[root@net53 /root]# id
uid=0(root) gid=0(root) 群組=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
[root@net53 /root]#
```

在 Linux 中由於檔案分佈太廣，且目錄又很多，通常要找一個檔案都會很麻煩，這時可用 find 來幫您尋找該檔案在那個目錄內，由於 find 指令功能很強大，

選擇項很多，在此我們只列出常用的。

find 常用語法：【find 起始路徑 -name 要尋找的名稱 [-print]】

其中起始路徑是說您要從那個路徑開始找起，是要從根目錄【/】，還是從現在所在的目錄【.】皆可，至於選擇項-print，在 find 指令中是為預設值，意思是說當找到檔案時，把所在目錄列印出來，知道以上這些，這樣便可使用 find 指令。

例如：要從根目錄開始尋找一個叫做 netscape 的檔案，便可用：

```
find / -name netscape
```

例如：要從根目錄開始尋找 net 開頭的檔案，便可用：

```
find / -name "net*"
```

5.6.11 kill 指令 移除 Process

ps 指令筆者有說過，這是在查看 Process 的動作情形，但是當我們的系統遇到衝突時，那就必須運用 kill 指令來將衝突的 Process 做個適當的處理。接下來我們開始介紹 kill 指令。

kill 簡便語法：【kill <process_ID>】

例如：kill 678

kill 常用語法：【kill -<signal_number> <process_ID>】

例如：kill -1 678

kill 另一語法：【kill -<signal_name> <process_ID>】

例如：kill -HUP 678

關於 signal 的編號以及其意義，您可以參考/usr/include/sys/signal.h 中的定義，或是執行 man kill，在這裡筆者僅介紹比較重要的一些 signal。

- ★ 1 (HUP)：對於許多 daemon 而言，會通知 daemon 更新設定檔；而對於其他程式而言，會通知程式結束。
- ★ 3 (QUIT)：會通知該 process 結束。
- ★ 9 (KILL)：會直接通知 kernel 把這個 process 移除。

當您更新某些 daemon process 的設定檔，您必須執行『kill -1 <PID>』，來通知這一些 process 更新資料，可以利用 ps 指令來找出它們的 PID，當您發現某個 process 不正常而想將它移除的時候，您可以利用 ps 指令來找出它們的 PID，再用 kill 將它移除，您可以先試試 kill -1、再試 kill -3，都不行的話就用 kill -9 強制移除。如果不想那麼麻煩的話，也可以直接執行 kill <PID>，免除一切麻煩。

5.6.12 mount、umount 指令 掛上、卸下檔案系統「設備」

通常我們要使用軟碟、硬碟或光碟機內的檔案系統時，就要先把檔案系統給外掛進來，這樣才能使用，但在外掛之前，您必須要知道您 Linux 的 kernel【核心程式】，是否有支援到您所要外掛的檔案系統格式，若沒有您必須要先更新 kernel 才行，kernel 的更新請看前面的章節。

在 Linux 中，我們常用到的檔案系統為 msdos、ext2、hpfs、iso9660 等格式，至於還有其他的格式都不太常用，故我們省略。ext2 檔案格式是 Linux 系統格式之一，也是最廣為使用的；hpfs 檔案格式是 OS/2 的檔案格式；iso9660 檔案格式是為 CD-ROM 裝置的檔案格式，假如 CD-ROM 內的光碟片的檔案都是 MS-Dos 的格式，當要使用該光碟片時，仍然要使用 iso9660 的檔案格式。其語法為：

mount 常用語法：mount -t 檔案格式 [-o conv=auto] 設備檔名 要外掛的目錄。

選擇項“-t 檔案格式”，是指定您要外掛那一種的檔案系統格式，在使用 mount 時，一定要有“-t 檔案格式”選擇項；選擇項“-o conv=auto”，是說將外掛進來的檔案系統做自動轉換，轉換成 Linux 格式，這個自動轉換只對文字檔有用，原本在 Dos 中用文書編輯器【pe2、漢書】寫成的文件檔，拷到 Linux 裡後，再用 Linux 的文書編輯器【vi、joe】來看該文件檔，會明顯的發現到在每行結尾處有“^M”符號，這時只要用“-o conv=auto”選擇項，就可轉換成 Linux 文件格式。

設備檔名是指您要外掛的軟碟、硬碟或是光碟機等，像軟碟的 A 槽為 /dev/fd0、B 槽就為 /dev/fd1【在此是不分磁碟容量，因為 Linux 會自動偵測】；而硬碟的 C 槽就為 /dev/hda1、D 槽就為 /dev/hda2 或 /dev/hdb1，這個就要看您有幾個硬碟，以及您如何規畫 partitions 而定，通常 hda 是指第一個硬碟，那 hdb 就是指第二個硬碟，而 1、2 是指一顆硬碟內的各個 partitions 代號；光碟機通常都為 /dev/cdrom 或 /dev/sbpcd，這個就要看當初選 bootdisk 時所選的光碟機驅動程式而定。

在 Linux 中有一個特定的目錄，是專門給您外掛檔案系統專用的，這個目錄為 /mnt，通常 /mnt 都是空目錄；而 CD-ROM 也是有固定的目錄，就是為 /cdrom。請看下例：

例如：要外掛 Dos 的磁片【在 A 磁碟機】，並且做自動轉換的格式，其作法為：

```
mount -t msdos -o conv=auto /dev/fd0 /mnt/fd0
```

假如要外掛 OS/2 的磁片【在 B 磁碟機】，其作法為：

```
mount -t hpfs /dev/fd1 /mnt/fd1
```

假如要外掛 Linux 的磁片【在 B 磁碟機】，其作法為：

```
mount -t ext2 /dev/fd1 /mnt/fd1
```

假如要外掛 CD-ROM，其作法為：

```
mount -t iso9660 /dev/hdb /mnt/cdrom
```

以上這下掛入點目錄，Red Hat 都已經在 /mnt 下面做好了，讀者可以情況增修，如果您想查尋現在電腦上有掛上那些東西，您可以直接鍵入 mount 來查尋，請見下例：

```
linux1:~#mount
/dev/hda2 on / type ext2 (rw)
/dev/hda1 on /dosc type msdos (rw)
/dev/hdb1 on /dosd type msdos (rw)
/dev/sbpcd on /cdrom type iso9660 (ro)
none on /proc type proc (rw)
```

當您想要將某一個檔案系統退出時，您可以使用 umount 指令，其語法為：【所有的檔案系統，都可使用該語法來卸下檔案系統】

umount 常用語法：umount 掛檔案系統的目錄

例如：我們要將/cdrom 退出，其用法為：

```
umount /mnt/cdrom
```

上框中，如果是在 /etc/fstab 中有定義的目錄，例如：/mnt/cdrom 、/proc、/...等等，mount 的時候您可以不需要指定它的來源。

5.6.13 fsck 指令 掃描檔案

當您想要檢查檔案的時候，您一定要記住，先 umount 它，但通常會因為有人正在該 mount point 中，而無法 mount 掉，所以通常要檢查檔案系統最好的方法，就是重新啟動進入單使用者模式。

fsck 常用語法：【fsck <-y> r 『想檢查的位置』 】

例如：您現在想要檢查 hda2，其作法為：

```
fsck <-y> /dev/hda2
```

<-y>：這一個參數是當您每檢察到一個錯誤時，您就要按一次 Y，使用 -y 參數則會自動回答所有的 Y。

如果你之前有不常關機，導致開機時掃描檔案無法掃描過去，而不能正常的開機，這時你也

5.6.14 file 指令 顯示檔案資訊

file 可以顯示檔案相關訊息，例如：

```
[root@eef /root]# file vmware-forlinux-103.tar.gz
vmware-forlinux-103.tar.gz: gzip compressed data, deflated, last modified: Tue Jul 20 08:14:59 1999, os: Unix
[root@eef /root]# file vmware-tools131.exe
vmware-tools131.exe: MS Windows PE 32-bit Intel 80386 GUI executable not relocatable
[root@eef /root]# file -z vmware-forlinux-103.tar.gz
vmware-forlinux-103.tar.gz: GNU tar archive (gzip compressed data, deflated, last modified: Tue Jul 20 08:14:59 1999, os: Unix)
```

5.6.15 grep 指令 找尋檔案中資訊

grep 可以顯示檔案中相關訊息，例如：

```
ps x | grep httpd
```

5.6.16 whereis 指令 找尋執行檔，原始檔和手冊

whereis 可以顯示執行檔，原始檔和手冊檔案在系統中位置，例如：

```
whereis ls
ls: /bin/ls /usr/man/man1/ls.1
```

5.6.17 locate 指令 找尋檔案

locate 也可以找尋符合條件之檔案，但是由資料庫中，速度較快。語法如下：

```
updatedb 建立檔案由資料庫
locate core
```

5.6.18 alias 指令 建立代名

alias 可以建立代名，方便使用者的習慣，系統預設之代名可執行 alias 取得，為：

```
alias cls='clear'
alias copy='cp -iv'
alias cp='cp -i'
alias date='date +%Y\ %B\ %d\ %A\ %T\ %Z'
alias del='rm -iv'
alias dir='ls -N --color'
alias econfig='vi /etc/profile.d/cle-config.sh ; source /etc/profile.d/cle-confi
g.sh'
alias ll='ls -N -la --color'
alias ls='ls -N --color'
alias md='mkdir'
alias move='mv -iv'
alias mv='mv -i'
alias rd='rmdir'
alias rm='rm -i'
alias which='type -path'
alias xcin='xcin -in1 cj.tab -in2 liu5.tab'
```

執行 `econfig` 可以修改。

`alias` 的語法為：

```
alias alias_name='command'
```

解除 `alias` 為 `unalias`。

5.7 基本網路指令簡介

請熟記以下的網路指令，才能真正了解自己的伺服器的狀況和使用網路的便利，並且駕輕就熟的排除故障點。

5.7.1 telnet 指令 與遠端終端機連線

在 Internet 上由於有很多的網點位址，當您想要進入遠端的電腦時，可以利用 `telnet` 來做與遠端終端機連線的動作。其語法如下：

telnet 常用語法：【telnet 網路位址/該機名稱】

以亞東工專電機科工作站的網路為例，位址如下所示：

該機名稱	網路位址
ntl	192.192.73.6
www	192.192.73.3
gateway	192.192.73.1
dns	192.192.73.2
ipproxy	192.192.73.9
www1	192.192.73.4
亞東工專 bbs 站	192.192.69.134

5.7.2 ftp 指令 下傳/上載檔案

`ftp` 是用於 Internet 上的檔案傳輸工具，其語法與 `telnet` 指令完全相同，在此就不加贅述。如果讀者還想知道更多語法，可以進入該 `ftp` 站後，再按 **help** 或？

來尋求查詢指令，若要查詢指令的意義可按 **? 指令名稱** 即可。

ftp 語法：【ftp 網路位址/該機名稱】

假設您用 ftp 想要去交大下傳檔案到您自己的電腦內，可以這樣執行

```
ftp nctuccca.edu.tw
```

一開始交大的 ftp server 會詢問您的 login name，由於我們都沒有在交大有帳號，故在此的 login name 都鍵入 anonymous 或者 ftp，接下來會詢問您 password，這時 password 就輸入您自己的 E-mail 帳號即可，這樣就能進入交大的 ftp server 去提取您想要的檔案。

在提取檔案之前，我們都會做幾個指令動作，第一個指令為 **bin**，這是設定在傳輸檔案時，用二進位【binary】來傳輸，第二個指令為 **hash**，此指令是說當每傳輸 1024bytes，就會出現一個#，用此可觀察出在傳輸時是否有斷線的情形發生，第三個指令為 **prom**，此指令是在傳輸檔案時關掉交談功能，以增加傳輸速率。

在做完上面的動作後，就要真正的提取檔案，其捉檔的指令有兩個，一個為 **get**，這個指令只能一次提取一個檔案，而且在下指令時，其提取的檔名不能使用萬用字元；另一個為 **mget**，這個指令一次可提取數個檔案，而且在下指令時，其提取的檔名可使用萬用字元。

至於上載檔案的指令為 **put** 與 **mput**，其用法與 **get**、**mget** 完全相同，在此就不多加累述。而在上載的時後，是有一些限制，其限制為您必須在該機上有您的帳號或 server 提供此權限 ftp server，這樣才行上載檔案。

要離開 ftp，請打 bye 或 exit 離開。

例如：

```
[root@net58 /root]# ftp 192.192.73.4 →輸入此提供FTP服務的伺服器
Connected to 192.192.73.4.
220 cle.ee.oit.edu.tw FTP server (Version wu-2.4.2-VR17(1) Mon Apr 19 09:21:53 E
DT 1999) ready.
Name (192.192.73.4:iori): download →輸入帳號名稱
331 Password required for download.
Password: →密碼
```


5.7.3 mail 指令 收信與寄信

當讀者要在站上寫信給你的朋友時，你可以這樣執行：

mail (對方的) E-mail Address

假設筆者要寄一封信給 net53.ee.oit.edu.tw 這台機器的 fang 使用者，範例如下框：

```
[root@net53 /root]# mail a8630257@net53.ee.oit.edu.tw  
  
Subject: test  
  
This is test letter.  
.  
EOT
```

Subject 為這封信的標題，我們寫 test，按 Enter 後，就是這封信的內容，我們是寫 this is test letter.，寫完內容後，按 Enter，再按[.]號，再按 Enter，這封信就寄出去了。假如你要寄信的對象，其 Mail Server 是跟你同一台機器的，你只要執行

mail 使用者的名字

就可以了。又或者你是要寄一個文字檔或是你編好的檔案，你可以這樣執行：

mail 使用者的 E-mail Address < 文字檔案

假設我要寄 test.doc 給 net53.ee.oit.edu.tw 這台機器 fang 這個使用者，則是要這樣寫：

```
mail a8630257@net53.ee.oit.edu.tw < test.doc
```

當你有新的信件時，系統會通知你有新信件進來，你只要執行

mail

不加任何參數，就可以看信了。範例如下框所示：

```
You have new mail.      ←系統通知你有新信件
[root@net53 /root]# mail      ←執行mail 看信
Mail version 5.5 6/1/90.  Type ? for help.
"/var/spool/mail/alumi": 2 messages 1 new 2 unread
U  1 slakware.bbs@bbs.ntu  Tue Jul  1 15:11 138/5119  "宏佳你好"
>N  2 root@net53.ee.oit.  Tue Jul 29 15:56  11/428  "test"
  & 2                    ←執行2，看第二封信
Message 2:
From root@net53.ee.oit.edu.tw  Tue Jul 29 15:56:17 1998      ←信件的來處和日期
Date: Tue, 29 Jul 1998 15:56:16 +0800                        ←信件到達的日期
From: root <root@net53.ee.oit.edu.tw>                       ←信件的來處
To: a8630257@net53.ee.oit.edu.tw                            ←收信者的 E-mail Address
Subject: test        ←信件的標題

this is test later.  ←信件的內容

&
```

您要再看信時，還可以在提示符號">"之下加一些參數，來處理信件，範例如下：

當你看完一封信，要刪除它時，用

& d

假設你收到很多封信，想刪除第一封到第五封，則用

& d 1-5

刪掉全部則用

& d*

將信另儲存成一個檔案(假設我們要將信另存一個新檔 letters 則用

& s letters

回信則用

& r

你想知道其它的參數功能，則用

& ?

離開系統，但不將所看過的信儲存成家目錄下的 mbox，則用

& x

離開系統，將所看過的信儲存成 mbox，則用

& q

5.7.4 ping 指令 測試遠端電腦是否在線上

當您想要連線到某一個遠端電腦之前，可以先利用 ping 來測試遠端的電腦是否有在線上，若有在線上的話，則可以正常的使用，若沒有在線上時，可能是遠

端的電腦已關掉，或者是網路線上有台主機已當機，所造成的結果。

假若您在使用網路時，覺得網路異常，您可先自行測試一下，便可知道那一台主機有當機的情形發生。

底下是 ping 192.192.73.1，且有在線上的範例，如下框所示：

```
[root@net53 /root]# ping 192.192.69.30
PING 192.192.69.30 ( 192.192.69.30 ) : 56 data bytes
64 bytes from 192.192.69.30 : icmp_seq=0 ttl=128 time=1.6ms
64 bytes from 192.192.69.30 : icmp_seq=0 ttl=128 time=1.1ms
64 bytes from 192.192.69.30 : icmp_seq=2 ttl=128 time=0.6 ms
64 bytes from 192.192.69.30 : icmp_seq=3 ttl=128 time=0.6 ms
      :  ↵ 可按下 ctrl + c 結束
    --- 192.192.69.30 ping statistics ---
4 Packets transmitted , 4 packets received, 0% packet loss
round-trip min/avg/max=1.1/1.3/1.6ms
```

5.7.5 traceroute 指令 列出所經過的閘道器

這個指令是列出當你要連線到遠方機器時，所需要經過的閘道器，通常是用來查看網路是否能正常運作。

traceroute 常用的語法: 【 traceroute IP 位址】

假設我們要查出從本身機器連到 192.192.69.254 這個位址時，要經過哪些閘道器，你可以這樣執行。如下圖所示：

```
[root@net53 /root]# traceroute 192.192.69.254

traceroute to 192.192.69.254 (192.192.69.254), 30 hops max, 40 byte packets
 1  192.192.73.1 (192.192.73.1)          0.857 ms  0.625 ms  0.603 ms
 2  router.oit.edu.tw (192.192.69.254)  2.923 ms  2.619 ms  2.328 ms
```

從以上的範例，我們可以知道從本身的機器要連到 192.192.69.254 這個位址時，要經過 192.192.73.1 和 192.192.69.254 這兩個閘道器。

5.7.7 ifconfig 指令 設定與管理網路介面

這是用來設定與管理網路介面的指令，網路介面泛指網路卡、PPP、SLIP、Loopback 等等。接下來我們介紹 ifconfig 的一般用法：

ifconfig 常用語法：【ifconfig -a】

這可以查詢所有的網路介面情形，如下圖所示：

```
[root@net53 /root]# ifconfig -a

lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
            UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
            RX packets:22 errors:0 dropped:0 overruns:0 frame:0
            TX packets:22 errors:0 dropped:0 overruns:0 carrier:0 coll:0

eth0       Link encap:Ethernet  HWaddr 08:00:39:01:1A:5B
            inet addr:192.192.73.4  Bcast:192.192.73.255  Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:207607 errors:0 dropped:0 overruns:0 frame:1930
            TX packets:590 errors:0 dropped:0 overruns:0 carrier:0 coll:0
            Interrupt:11 Base address:0x6000
```

舉例:如果讀者懷疑自己的網路介面當掉了，則可以執行以下指令。

```
[root@net53 /root]# ifconfig de0 down      停止網路介面。
[root@net53 /root]# ifconfig de0 up        再啟動網路介面。
[root@net53 /root]# ping 192.192.73.1     隨便一台遠端機器。
```

5.7.8 netstat 指令 查詢網路使用情形

這一個指令可以讓讀者查詢現在的網路使用以及連線的情形。

netstat 常用指令：【netstat 「參數」】

1. 現在我們來網路介面流量，鍵入 netstat -i ，如下框所示：

```
[root@net53 /root]# netstat -i
Kernel Interface table
Iface    MTU  Met  RX-OK RX-ERR RX-DRP RX-OVR   TX-OK TX-ERR
TX-DRP TX-OVR  Flags
lo       3584   0    26     0     0     0     26     0     0
0 BLRU
eth0     1500   0 216134     0     0     0     741     0     0
0 BRU
```

2. 查詢 TCP 網路連線的情形，鍵入 netstat -t ，如下圖所示：

```
[root@net53 /root]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp                0          0 net53.ee.oit.edu.:ftp  192.192.73.8:2513
ESTABLISHED
[root@net53 /root]#
```

上框中，Active Internet connections，您可以看到從 net53.ee.oit.edu..ftp 到 192.192.73.8 已經建立了連線的 prot 2513，經由 ESTABLISHED ，所以讀者可一知道有哪些機器連到您機器的哪一個 port 。

3. 查詢所有網路連線的情形，鍵入 netstat -a | more ，如下圖所示：

```
[root@net53 /root]# netstat -a |more
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp                0          0 net53.ee.oit.edu.:ftp  192.192.73.8:2513
ESTABLISHED
tcp                0          0 *:www                   *.*
LISTEN
tcp                0          0 *:smtp                   *.*
LISTEN
tcp                0          0 *:2049                   *.*
```

```

LISTEN
tcp          0          0  *:678      *.*
LISTEN
tcp          0          0  *:printer  *.*
LISTEN
tcp          0          0  *:auth     *.*
LISTEN
tcp          0          0  *:netstat  *.*
LISTEN
tcp          0          0  *:systat   *.*
LISTEN
tcp          0          0  *:finger   *.*
LISTEN
tcp          0          0  *:imap2    *.*
LISTEN
tcp          0          0  *:pop3     *.*
LISTEN
tcp          0          0  *:login    *.*
LISTEN
tcp          0          0  *:shell    *.*
LISTEN
tcp          0          0  *:telnet   *.*
LISTEN
tcp          0          0  *:ftp      *.*
LISTEN
tcp          0          0  *:time     *.*
LISTEN
tcp          0          0  *:sunrpc   *.*
LISTEN
udp          0          0  *:2049     *.*
udp          0          0  *:675      *.*
udp          0          0  *:ntalk    *.*
--More--

```

讀者可以看見 netstat -a 比 -t 多了一些正在等待連線的 service daemon 資訊，圖中的 http port、smtp port、imap port、pop3 port....等等都在等待連線。

5.7.9 nslookup 指令 查詢機器名稱與 ip 位址

如果你以已經知道某台機器的 ip 位址，而你想要知道此台機器的 hostname，或者反過來你知道某台機器的 hostname 而想要知道此機器的 ip 位址，這時你又可以此指令去查詢，不過先決條件是你有設定 Name Server 才查得到喔！

Nslookup 常用語法：【nslookup IP 或機器名稱】

其實我們也可以在提示符號下直接下 nslookup 的指令，這時就會出現 > 的符號，我們可以在這個符號下輸入 ip 或者機器的名稱，程式就會幫我連到網域名稱伺服器去查詢了！要是不想查了，按【Ctrl】+【d】或者輸入 exit 即可。

例如：

```
[root@net53 /root]# nslookup
Default Server:  www2.ee.oit.edu.tw
Address:  192.192.73.2

> 192.192.73.1
Server:  www2.ee.oit.edu.tw
Address:  192.192.73.2

Name:  gateway.ee.oit.edu.tw
Address:  192.192.73.1

> exit
```

5.8 系統資訊管理

筆者認為身為一個系統管理者，必須對你所管理系統要有充分的了解，所以

筆者在這一章節所要介紹的就是系統資訊的管理，讓你充分的掌握系統的資訊，而這一類的資料是放在/var/log/這個目錄中，筆者在這一章節會簡略的敘述這個目錄裡的檔案內容。

5.8.1 last 指令

這個指令的功能是列出系統的登入紀錄，詳細的列出使用者於何時登入，從哪裡登入，使用哪一個 tty 登入，範例如下框所示：

```
[root@net53 /root]# last
student  ftp          192.192.73.8    Mon Apr 13 15:44 - 16:01 (00:16)
student  ftp          192.192.73.8    Mon Apr 13 15:06 - 15:17 (00:11)
root     tty2           Mon Apr 13 14:38  still logged in
root     tty1           Mon Apr 13 13:38  still logged in
root     tty0          h136.s84.ts.hine Sun Apr 12 22:16 - 22:16 (00:00)
runlevel ~             Fri Apr 10 12:40
reboot   ~             Fri Apr 10 12:40
runlevel ~             Fri Apr 10 11:17
reboot   ~             Fri Apr 10 11:17
runlevel ~             Fri Apr 10 11:00
reboot   ~             Fri Apr 10 11:00
student  tty1          192.192.73.49   Thu Apr 9 16:57 - 17:15 (00:17)
student  tty0          192.192.73.39   Thu Apr 9 13:23 - 13:30 (00:07)
```

從以上的範例中，我們可以清楚的知道最近的使用者登錄情況，都會詳細的列印出來。而這個指令所對應的檔案資料是/var/log/wtmp 這個檔案，所以讀者想刪除紀錄系統登入情形的紀錄檔，只要刪掉/var/log/wtmp 這個檔案，系統會重新再重新紀錄使用者的登入情形，所以保持這個檔案的更新是系統使用者的責任。

此外這個指令還有所多參數可以使用，參數說明如下：

參數： -t ttyX

功能： 列出系統中有哪些使用者使用哪一個 ttyX 登入系統

範例： last -t tty1

```
[root@net53 /root]# last -t tty1
student  tty1      192.192.73.49   Thu Apr  9 16:57 - 17:15  (00:17)
root     tty1      192.192.73.41   Wed Apr  8 19:41 - 20:07  (00:25)
student  tty1      192.192.73.41   Wed Apr  8 19:41 - 19:41  (00:00)
root     tty1      192.192.73.57   Wed Apr  8 18:39 - 19:06  (00:27)
root     tty1      192.192.73.57   Wed Apr  8 18:12 - crash  (00:25)
wtmp begins Wed Apr  8 18:04
[root@net53 /root]#
```

列出哪些使用者使用 tty1 登入系統。

參數： last 使用者名稱

功能： 列出此使用者的登入紀錄

範例： last student

```
[root@net53 /root]# last student
student  ftp      192.192.73.8    Mon Apr 13 15:44 - 16:01  (00:16)
student  ftp      192.192.73.8    Mon Apr 13 15:06 - 15:17  (00:11)
student  tty1     192.192.73.49   Thu Apr  9 16:57 - 17:15  (00:17)
student  tty0     192.192.73.39   Thu Apr  9 13:23 - 13:30  (00:07)
student  ftp      192.192.73.28   Thu Apr  9 09:43 - 09:58  (00:15)
student  ftp      192.192.73.28   Thu Apr  9 09:26 - 09:27  (00:01)
student  tty2     micro10.ee.oit.e Thu Apr  9 08:49 - 08:55  (00:06)
```

列出 student 這個使用者登入系統的情形

參數： -h 主機名稱

功能： 列出所定義的主機登入系統的情形

範例： last -h 192.192.73.8

```
[root@net53 /root]# last -h 192.192.73.8
student  ftp      192.192.73.8    Mon Apr 13 15:44 - 16:01  (00:16)
student  ftp      192.192.73.8    Mon Apr 13 15:06 - 15:17  (00:11)
fang     ftp      192.192.73.8    Mon Apr 15 16:22 - 13:20  (00:14)
wtmp begins Wed Apr  8 18:04
[root@net53 /root]#
```

列出從 192.192.73.8 這台主機登入系統的情形。

參數： -y

功能： 詳細列出系統登入的年月日

範例： last -y

```
[root@net53 /root]# last -y |more
student  ftp          192.192.73.8    Mon Apr 13 1998 15:44 - 16:01 (00:16)
student  ftp          192.192.73.8    Mon Apr 13 1998 15:06 - 15:17 (00:11)
root     tty2           Mon Apr 13 1998 14:38  still logged in
root     tty1           Mon Apr 13 1998 13:38  still logged in
root     tty0          h136.s84.ts.hine Sun Apr 12 1998 22:16 - 22:16 (00:00)
runlevel ~             Fri Apr 10 1998 12:40
reboot   ~             Fri Apr 10 1998 12:40
runlevel ~             Fri Apr 10 1998 11:17
reboot   ~             Fri Apr 10 1998 11:17
runlevel ~             Fri Apr 10 1998 11:00
reboot   ~             Fri Apr 10 1998 11:00
student  tty1          192.192.73.49   Thu Apr 9 1998 16:57 - 17:15 (00:17)
student  tty0          192.192.73.39   Thu Apr 9 1998 13:23 - 13:30 (00:07)
root     tty0          192.192.73.34   Thu Apr 9 1998 10:13 - 10:15 (00:01)
student  ftp          192.192.73.28   Thu Apr 9 1998 09:43 - 09:58 (00:15)
student  ftp          192.192.73.28   Thu Apr 9 1998 09:26 - 09:27 (00:01)
root     tty3          192.192.73.40   Thu Apr 9 1998 08:58 - 09:05 (00:06)
```

參數： last -n 數目

功能： 列出最近新紀錄的第幾筆資料

範例： last -5

```
[root@net53 /root]# last -5
student  ftp          192.192.73.8    Mon Apr 13 15:44 - 16:01 (00:16)
student  ftp          192.192.73.8    Mon Apr 13 15:06 - 15:17 (00:11)
root     tty2           Mon Apr 13 14:38  still logged in
root     tty1           Mon Apr 13 13:38  still logged in
root     tty0          h136.s84.ts.hine Sun Apr 12 22:16 - 22:16 (00:00)
[root@net53 /root]#
```

列出最新紀錄的前五筆資料。

如果讀者你想知道更詳細的 last 指令功能，你可以執行 **【man last】** 來查看

線上說明資料。

5.9 RPM 套件的安裝

rpm 是 Red hat 特有檔案格式，RPM 指的是 **Red Hat Package Manager** 之縮寫，雖然名稱上包含著 Red Hat 的字號，但其原始設計理念是採開收式的，用意在於讓這個程式，能提供給所有人使用。透過 rpm 的管理，使用者可以將某個新軟體的 source code，重新整理包裝成另一種 source 與 binary 的檔案型式，如此一來，我們可以輕鬆地由 binary 型式的檔案，進行安裝與套件追蹤管理的工作，而 source 型式的檔案，也很方便地能夠再被重新整理包裝。同時，rpm 管理著一份資料庫，裡頭包含了所有的程式套件的檔案資料，透過這份資料庫，我們可以進程式套件之確認與查詢工作。

不過在之前你不是用客戶式安裝(custom)的【every thing】話，相信你有很多好用的工具沒有裝上去，雖然說我們可以用光碟選擇 update 來升級，但要重新開機，所以已經違反了一個性能良好的伺服器原則。因此筆者將介紹如何在文字模式下安裝或昇級套件。

首先我們可以把光碟機掛進來，然後切換到/mnt/cdrom/Redhat/RPMS 這個目錄下，在使用 rpm 這個工具程式將我們要的軟體安裝進來。或者從網站安裝。

RPM 常用語法：【rpm 選項 軟體名稱】

- U：是使 rpm 程式做新增/更新的安裝。
- i：是標準的安裝選項。
- vh：可以用來顯示安裝的過程，跟-U 和-i 選項配合用。
- i --replacepkgs：強制安裝並覆蓋已安裝的軟體。
- U --oldpackage：升級或安裝舊版軟體。
- Uvh --force：強制升級或安裝軟體，force 選項是 replacepkgs 和 oldpackage 的結合。
- e：解除安裝。
- q：查詢已安裝的軟體。
- Vf：檢查單一已安裝的軟體狀態。
- Va：檢查所有已安裝的軟體狀態。
- recompile：從原始檔安裝。
- rebuild：從原始檔安裝後，在產生函可執行檔的 rpm 檔。

RPM 用於安裝程式套件上，最簡單的操作格式如下：

```
example# rpm -i foobar-1.0-1.i386.rpm
```

另一個簡單的操作範例，是用於解除某一個程式套件的安裝：

```
example# rpm -e foobar
```

接下來的指令範例較為複雜，但它非常有用，可以讓您透過 FTP 來安裝程式套件。如果您能夠連上網路，想安裝某個新的程式套件時，所需下達的指令，便是在檔案前加上適當的 URL 格式，例如：

```
example# rpm -i
```

```
ftp://ftp.pht.com/pub/linux/redhat/rh-2.0-beta/RPMS/foobar-1.0-1.i386.rpm
```

請特別注意，RPM 目前已允許透過 FTP 來進行查詢或安裝。

```
example% rpm -qlp ftp://some.ftp.site/pub/redhat/RPMS/mem_test-0.10-1.i386.rpm
```

```
example# rpm -Uvh -v ftp://some.ftp.site/pub/redhat/RPMS/wget-1.4.5-1.i386.rpm
```

上述都是些簡單的指令，rpm 還有許多不同的選項方式，您可以在 RPM 的 man page 裡，找到這些選項的詳細說明。在光碟片中指定軟體名稱時，並不需要需入全名，只要輸入幾個代表性的字母即可，例如：

```
rpm -Uvh joe*
```

這樣它就會去安裝 joe 文書處理器了。

RPM 是個非常有用的工具，而且如您所知的，它具有許多選項功能。讓您了解這些功能的最好方式，便是實際介紹幾個範例，前面我們已經提過安裝與解除安裝的簡單例子，底下是一些額外的範例：

- 假設說，您不小心誤刪了幾個檔案，但不確定到底是哪些檔案，您想對整個系統進行確認動作，以了解哪些部份可能已經遺損，您可以這樣做：
- rpm -Va
- 假設說，您碰到一個認不出來的檔案，想要知道它是屬於哪一個程式套件的話，您可以這樣做：
- example% rpm -qf /usr/X11R6/bin/xjewel
其結果會得到：
xjewel-1.6-1
- 如果您找到一個新的 RPM 檔案，卻不清楚它的內容，想要獲得相關的資訊，可以這樣做：
- example% rpm -qpi koules-1.2-2.i386.rpm
其結果會得到：

Name : koules Distribution: Red Hat
Linux Colgate
Version : 1.2 Vendor: Red Hat
Software
Release : 2 Build Date: Mon Sep 02
11:59:12 1996
Install date: (none) Build Host: porky.redhat.com
Group : Games Source RPM:
koules-1.2-2.src.rpm
Size : 614939
Summary : SVGAlib action game with multiplayer, network, and sound support

Description :

This arcade-style game is novel in conception and excellent in execution. No shooting, no blood, no guts, no gore. The play is simple, but you still must develop skill to play. This version uses SVGAlib to run on a graphics console.

- 如果您想了解某個 RPM 檔案，會在系統裡安裝哪些檔案，您可以這樣做：

- `example% rpm -qpl koules-1.2-2.i386.rpm`

其結果會是：

```
/usr/doc/koules
/usr/doc/koules/ANNOUNCE
/usr/doc/koules/BUGS
/usr/doc/koules/COMPILE.OS2
/usr/doc/koules/COPYING
/usr/doc/koules/Card
/usr/doc/koules/ChangeLog
/usr/doc/koules/INSTALLATION
/usr/doc/koules/Icon.xpm
/usr/doc/koules/Icon2.xpm
/usr/doc/koules/Koules.FAQ
/usr/doc/koules/Koules.xpm
/usr/doc/koules/README
/usr/doc/koules/TODO
/usr/games/koules
/usr/games/koules.svga
/usr/games/koules.tcl
```

/usr/man/man6/koules.svga.6

- 如果您有機會取得大批的 SRPM 檔案，下列的方法可以讓您檢查出，哪些檔案可能是壞掉的。
- `example$ rpm -qp SRPMS/*.rpm 1> /dev/null 2> ~/srpm.fail`

上述只是一些範例，當您對 RPM 愈加熟悉時，應該能想出其他更多的創意組合

5.10 Red Hat Linux 目錄構造

UNIX 大致分為 BSD 與 SYSV 兩個類別，這兩者的指令、目錄構造、作業系統的特性，都有些許的差異，不過沒有什麼關係，筆者還是一一的介紹 Linux 的目錄構造。

- / 這是 UNIX 系統的根目錄。UNIX 不像 DOS 一樣有 C:D:E:..... 等等的硬碟代分別，UNIX 是由根目錄開始擁有一大堆子目錄，而某個割區可能只有掛在某個子目錄上面，我們稱這些掛上另一分割區的目錄為「掛入點」(Mount Point)
- /bin/ 這是在『單使用者 及 多使用用者』模式下都用的到的一般使用者指令，而這些指令都是開機時所必備的，所以這通常都是靜態連結的。此目錄是用來放對系統最重要的基本指令，例如 "mount"、"cp"、"ls"... 等指令就放在這裡。
- /boot/ 顧名思義，"/boot" 是拿來放跟開機有關的檔案，Linux 系統核心就放在這個目錄底下。
- /dev/ 這是裝置節點檔，/dev/MAKEDEV 是產生節點檔的工作程式。這個目錄底下是放硬體設備檔用的，Linux 系統裡面，每一個硬體裝置都會有一個對應的設備檔，對這個檔案作存取的动作，就是對這個硬體進行存取，這裡面比較重要的檔案除了前一章提到的硬碟、光碟機以外，還有：
- | 裝置檔 | 裝置 |
|------------|------------|
| /dev/fd0 | A: 槽軟碟機 |
| /dev/lp0 | 第一個印表機並列埠 |
| /dev/ttyS0 | 序列埠 COM 1. |
| /dev/ttyS2 | 序列埠 COM 2. |
| /dev/psaux | PS/2 滑鼠 |

`/etc/` 系統設定檔以及 script，除了設定檔以外，還有使用者密碼檔、群組檔，所以 `/etc/` 可以說是最重要的目錄了。

`/home/` 用來放使用者的家目錄。

`/sbin/` 這是在『單使用者 及 多使用用者』模式下都用的到系統程式以及管理指令，這些指令都是開機時所必備的，以這通常都是靜態連結的。`"sbin"` 也是放指令，但性質上比較偏向放跟系統管理有關的指令，有許多是只有 `"root"` 才可以執行的指令。

`/tmp/` 這是暫存檔案區，有許多程式都會建立暫存檔，因此個目錄是絕對必要的。我們都會將 `/tmp/` 連結到硬碟空比較大的分割區，可以減少執行時所會遇到的一些小問題。如果讀者必須要定時清除 `/tmp/` 下的暫存檔，則請修改 `/etc/daily` 中的設定。

`/lib` 開機時必要的一些動態連結函式庫。

`/mnt` 外掛的檔案，預設底下有 `cdrom` 跟 `floppy` 兩個目錄，一個用來掛光碟機，一個用來掛軟碟機，正常安裝後，只要執行 `"mount /mnt/cdrom"` 就可以把光碟片的內容掛在 `"/mnt/cdrom"` 底下。

`/proc` `"/proc"` 是一個假的檔案系統，他完全不佔硬碟空間，事實上是存在記憶體裡面，存放系統所有的資料；您可以用 `"ls /proc"` 看看這底下有什麼，在試試打 `"cat /proc/meminfo"` 看看，有好奇心的人可以一一的看看這些檔案裡面擺了哪些東西。

`/root` 超級使用者 `"root"` 的家目錄。

`/var` 放置系統的紀錄檔 (logfile) 跟一些雜七雜八的資料，比較重要的有：

`/var/log`

系統的紀錄檔，整個系統發生過什麼事，什麼人什麼時候從哪裡有用網路來取得系統的什麼服務，都有紀錄。

`/var/lib`

放置應用程式產生的資料，例如 "rpm" 的 rpm 資料庫就放在 `/var/lib/rpm` 底下，記錄了您系統上所有程式套件的資料。

`/var/spool`

讓系統上一些重要資料暫時放置的地方，例如 `/var/spool/mail` 底下是使用者的郵件信箱，個人還沒刪除的電子郵件都會放在這裡，`/var/spool/lpd` 底下則放置等待列印的資料。

`/usr` "`/usr`" 是 RedHat 裡面放最多東西的目錄，用 rpm 裝上去的程式幾乎都是放在 `/usr` 底下，這裡面比較重要的目錄有：

`/usr/bin`

放置主要的系統指令，RedHat Linux 底下常用的指令與應用程式都在這裡了。

`/usr/include`

放置系統的標頭檔 (header file)，寫 C 語言的人不該對這個目錄感到陌生。

`/usr/games`

顧名思義，放遊戲的地方。

`/usr/share`

放置一些重要的系統資料，其中的 `/usr/share/chinese` 目錄放置了 CLE 程式會用到的檔案資料。

`/usr/man`

放系統說明頁 (manual page) 的地方。

`/usr/doc`

放置各種文件與應用程式的說明檔，包含 HOWTO 檔案、Red Hat 安裝手冊，FAQ 等等都放在這裡。

`/usr/src`

放置系統核心與 RedHat rpm 程式的程式原始碼。

`/usr/X11R6/`

放置跟 X 視窗系統有關的所有檔案。

`/usr/local`

用來放置自行安裝的應用程式。