

二、UNIX 常用的指令：

以下只說明各指令的基本用法，若需詳細說明，請用 `man` 去讀詳細的 `manual`。

a. 關於檔案/目錄處理的指令：

1. `ls`

這是最基本的檔案指令。`ls` 的意義為 "list"，也就是將某一個目錄或是某一個檔案的內容顯示出來。

如果你在下 `ls` 指令後頭沒有跟著任何的檔名，它將會顯示出目前目錄中所有檔案。

也可以在 `ls` 後面加上所要察看的目錄名稱或檔案的名稱，如

```
% ls /home2/X11R5
```

```
% ls first
```

`ls` 有一些特別的參數，可以給予使用者更多有關的資訊，如下：

`-a`：在 UNIX 中若一個目錄或檔案名字的第一個字元為 "."，則使用 `ls` 將不會顯示出這個檔案的名字，我們稱此類檔案為隱藏檔。如 `tcsh` 的初設檔 `.tcshrc`；如果我們要察看這類檔案，則必須加上參數 `-a`。

`-l`：這個參數代表使用 `ls` 的長(`long`)格式，可以顯示更多的資訊，如檔案存取權，檔案擁有者(`owner`)，檔案大小，檔案最後更新日期，甚而 `symbolic link` 的檔案是 `link` 那一個檔等等。如下

```
% ls -l

drwx--x--x  2 jjtseng      512 Aug  8 05:08 18
drwx--x--x  2 jjtseng      512 Aug  8 22:00 19
-rw-----  1 jjtseng      566 Aug  8 05:28 makefile
```

2. `cp`

`cp` 這個指令的意義是複製("COPY")，也就是將一個或多個檔案複製成另一個檔案或者是將其複製到另一個目錄去。

`cp` 的用法如下：

```
cp f1 f2 : 將檔名為 f1 的檔案複製一份為檔名為 f2 的檔案。
cp f1 f2 f3 ... dir : 將檔案 f1 f2 f3 ... 都以相同的檔名複製一份放到目錄 dir 裡面。
cp -r dir1 dir2 : 將 dir1 的全部內容全部複製到 dir2 裡面。
```

`cp` 也有一些參數，如下：

`-i`：此參數是當已有檔名為 `f2` 的檔案時，若逕自使用 `cp` 將會將原來 `f2` 的內容掩蓋過去，因此在要蓋過之前必須先詢問使用者一下。如使用者的回答是 `y(yes)` 才執行複製的動作。

`-r`：此參數是用來做遞迴複製用，可將一整顆子樹都複製到另一個目錄中。

3. `mv`

`mv` 的意義為 `move`，主要是將一檔案改名或換至另一個目錄。如同 `cp`，它也有三種格式：

```
mv f1 f2 : 將檔名為 f1 的檔案變更成檔名為 f2 的檔案。
mv dir1 dir2 : 將檔名為 dir1 的目錄變更成檔名為 dir2 的目錄。
mv f1 f2 f3 ... dir : 將檔案 f1 f2 f3 ... 都移至目錄 dir 裡面。
```

`mv` 的參數有兩個，`-f` 和 `-i`，其中 `-i` 的意義與 `cp` 中的相同，均是 `interactive` 詢問之意。而 `-f` 為強迫(`force`)，就是不管有沒有同名的檔案，反正我就是要搬過去，所有其他的參數遇到 `-f` 均會失效。

4. `rm`

`rm` 的意義是 `remove`，也就是用來殺掉一個檔案的指令。在 `UNIX` 中一個被殺掉的檔案除非是系統恰好有做備份，否則無法像 `DOS` 裡面一樣還能夠救回來的。所以在做 `rm` 動作的時候使用者應該要特別小心。

`rm` 的格式如下：

```
rm f1 f2 f3 .....
```

而 `rm` 的參數比較常用的有幾個：`-f`，`-i`，與 `-r`

- `-f`：將會使得系統在刪除時，不提出任何警告訊息。
- `-i`：在除去檔案之前均會詢問是否真要除去。
- `-r`：遞迴式的刪除。

小心不要隨便使用 `rm -rf`，否則有一天你會"欲哭無淚".....

5. `mkdir`

`mkdir` 是一個讓使用者建立一個目錄的指令。你可以在一個目錄底下使用 `mkdir` 建立一個子目錄，使用的方法如下：

```
mkdir dirname1 [ dirname2 ... ]
```

如此你就可以建立一個或多個目錄。

6. `chdir (cd)`

這是讓使用者用來轉移工作目錄用的。

`chdir` 的用法如下：

```
chdir dirname
```

如此你就可以將目前的目錄轉移到 `dirname` 這一個目錄去。
或使用 "`chdir ..`" 來轉移到上一層目錄。

7. `rmdir`

相對於 `mkdir`，`rmdir` 是用來將一個"空的"目錄殺掉的。如果一個目錄下面沒有任何檔案，你就可以用 `rmdir` 指令將其除去。`rmdir` 的使用法如下：

```
rmdir dirname1 [ dirname2 .... ]
```

如果一個目錄底下有其他的檔案，`rmdir` 將無法將這個目錄殺掉，除非使用 `rm` 指令的 `-r` 選項。

8. `pwd`

`pwd` 會將目前目錄的路徑(`path`)顯示出來，例如：

9. `cat/more/less`

以上三個指令均為察看檔案內容的指令。`cat` 的意義是貓....不不不，是 `concatenate`，在字典上的意思是"連結，將...串成鎖狀"(語出資工電子詞典 `cdict`)，其實就是把檔案的內容顯示出來的意思。`cat` 有許多奇怪的參數，較為人所使用的是 `-n` 參數，也就是把顯示出來的內容加上行號。`cat` 的用法如下：

`cat [-n]`：自標準輸入讀進內容，你可以用 `pipe` 將別的程式的輸出轉向給 `cat`。

`cat [-n] filename`：將 `filename` 的內容讀進來，顯示在標準輸出上。

問題在於 `cat` 它是不會停下來的，因此並不好用(試想如果一個螢幕二十四行，而一個檔案四百行，`cat` 一出來將會劈哩啪啦不斷的捲上去，使用者

很難據此得到他們所需的資訊。) 所以才有人又寫了 `more` 出來。

`more` , 跟據蔡文能老師的說法是"再多一點就好", `more` 可以將所觀察的檔案跟據終端機的形態一頁頁的顯示出來, 再根據使用者的要求換頁或捲行。如果使用者要在某一個檔案中搜尋一個特定的字串, 則按 `/` 然後跟著打所要搜尋的單字即可進行搜尋。`more` 也可以找得到。`more` 的使用法如下:

```
more filename
```

如果你在使用中覺得已經看到了所要看的部份, 可以按 `'q'` 離開 `more` 的使用。在使用中按 `'v'` 亦可以使用編輯器來編輯所觀看的檔案。

`less` 的用法與 `more` 極類似, 原先它就是為了彌補 `more` 只能往前方捲頁的缺點而設計。`less` 的用法如下:

```
less filename
```

其與 `more` 不同的是它可以按 `y` 來往上捲一行, 並且可以用 `"?"` 來往回搜尋你所要找的單字。

10. `chmod`

`chmod` 為變更檔案模式用(`change mode`)。這個指令是用來更改檔案的存取模式(`access mode`)。在 UNIX 一個檔案上有可讀(`r`)可寫(`w`)可執行(`x`)三種模式, 分別針對該檔案的擁有者(`owner`)、同群者(`group member`)(你可以 `ls -lg` 來觀看某一檔案的所屬的 `group`), 以及其他他人(`other`)。一個檔案如果改成可執行模式則系統就將其視為一個可執行檔, 而一個目錄的可執行模式代表使用者有進入該目錄之權利。`chmod` 就是用來變更一些檔案的模式, 其使用方式如下:

```
chmod [ -fR ] mode filename ...
```

其參數的意義如下:

- `-f` Force. `chmod` 不會理會失敗的動作。
- `-R` Recurive. 會將所有子樹下的所有子目錄及檔案改為你所要改成的模式。

`mode` 可以為一個三位或四位的八進位數字, 來表示對某些對象的存取權。詳情可參閱 `chmod(1)` 的 manual page 中有關 Absolute Modes 的說明。

或是用一個字串來表示, 請參考 `chmod(1)` 的說明。

b. 關於 Process 處理的指令:

1. `ps`

`ps` 是用來顯示目前你的 process 或系統 processes 的狀況。以下列出比較常用的參數:

其選項說明如下:

- `-a` 列出包括其他 users 的 process 狀況。
- `-u` 顯示 user - oriented 的 process 狀況。
- `-x` 顯示包括沒有 terminal 控制的 process 狀況。
- `-w` 使用較寬的顯示模式來顯示 process 狀況。

我們可以經由 `ps` 取得目前 processes 的狀況, 如 `pid` , `running state` 等。

2. `kill`

`kill` 指令的用途是送一個 signal 給某一個 process。因為大部份送的都是用來殺掉 process 的 `SIGKILL` 或 `SIGHUP`, 因此稱為 `kill`。`kill` 的用法為:

```
kill [ -SIGNAL ] pid ...
kill -l
```

`SIGNAL` 為一個 signal 的數字, 從 0 到 31, 其中 9 是 `SIGKILL`, 也就是一

般用來殺掉一些無法正常 `terminate` 的訊號。其餘訊號的用途可參考 `sigvec(2)` 中對 `signal` 的說明。

你也可以用 `kill -l` 來察看可代替 `signal` 號碼的數目字。`kill` 的詳細情形請參閱 `man kill`。

c. 關於字串處理的指令：

1. `echo`

`echo` 是用來顯示一字串在終端機上。?

`echo -n` 則是當顯示完之後不會有跳行的動作。

2. `grep/fgrep`

`grep` 為一過濾器，它可自一個或多個檔案中過濾出具有某個字串的行，或是自標準輸入過濾出具有某個字串的行。

`fgrep` 可將欲過濾的一群字串放在某一個檔案中，然後使用 `fgrep` 將包含有屬於這一群字串的行過濾出來。

`grep` 與 `fgrep` 的用法如下：

```
grep [-nv] match_pattern file1 file2 ....
fgrep [-nv] -f pattern_file file1 file2 ....
```

`-n` 把所找到的行在行前加上行號列出

`-v` 把不包含 `match_pattern` 的行列出

`match_pattern` 所要搜尋的字串

`-f` 以 `pattern_file` 存放所要搜尋的字串

d. 網路上查詢狀況的指令：

1. `man`

`man` 是手冊 (`manual`) 的意思。UNIX 提供線上輔助 (`on-line help`) 的功能，`man` 就是用來讓使用者在使用時查詢指令、系統呼叫、標準程式庫函式、各種表格等的使用所用的。`man` 的用法如下：

```
man [-M path] [[section] title ] .....
man [-M path] -k keyword ...
```

`-M path` `man` 所需要的 `manual database` 的路徑。

我們也可以用設定環境變數 `MANPATH` 的方式來取代 `-M` 選項。

`title` 這是所要查詢的目的物。

`section` 為一個數字表示 `manual` 的分類，通常 1 代表可執行指令，2 代表系統呼叫 (`system call`)，3 代表標準函數，等等。

像下面 `man` 查詢的片段：

```
SEE ALSO
apropos(1), cat(1V), col(1V), eqn(1), lpr(1), more(1),
nroff(1), refer(1), tbl(1), troff(1), vgrind(1), vtroff(1),
whatis(1), eqnchar(7), man(7), catman(8)
```

我們如要參考 `eqnchar(7)` 的資料，則我們就輸入 `man 7 eqnchar`，便能取得我們所要的輔助訊息。

`-k keyword`

用來將含有這項 `keyword` 的 `title` 列出來。

`man` 在 UNIX 上是一項非常重要的指令，我們在本講義中所述之用法均僅只是一個大家比較常用的用法以及簡單的說明，真正詳細的用法與說明還是要請你使用 `man` 來得到。

2. who

who 指令是用來查詢目前有那些人在線上。

3. w

w 指令是用來查詢目前有那些人在線上，同時顯示出那些人目前的工作。

4. ku

ku 可以用來搜尋整個網路上的 user，不像 w 跟 who 只是針對 local host 的查詢。而且 ku 提供讓使用者建立搜尋特定使用者名單的功能。你可以建立一個檔案 information-file 以條列的方式存放你的朋友們的資料，再建立一個檔案 hosts-file 來指定搜尋的機器名稱。ku 的指令格式可由 ku -h 得到。

E. 網路指令：

UNIX 提供網路的連接，使得你可以在各個不同的機器上做一些特殊的事情，如你可以在系上的 iris 圖形工作站上做圖形的處理，在系上的 Sun 上讀 News，甚至到學校的計中去找別系的同學 talk。這些工作可以利用 UNIX 的網路指令，在你的位子上連到各個不同的機器上工作。如此一來，即使你在寢室，也能輕易的連至系上或計中來工作，不用像以前的人必須泡在冷冰冰的機房面。

這些網路的指令如下所述：

1. rlogin 與 rsh

rlogin 的意義是 remote login，也就是經由網路到另外一部機器 login。rlogin 的格式是：

```
rlogin host [ -l username ]
```

選項 -l username 是當你在遠方的機器上的 username 和 local host 不同的時候，必須輸入的選項，否則 rlogin 將會假設你在那邊的 username 與 local host 相同，然後在第一次 login 時必然會發生錯誤。

rsh 是在遠方的機器上執行某些指令，而把結果傳回 local host。rsh 的格式如下：

```
rsh host [ -l username ] [ command ]
```

如同 rlogin 的參數 -l username，rsh 的 -l username 也是指定 remote host 的 username。而 command 則是要在 remote host 上執行的指令。如果沒有指定 command，則 rsh 會去執行 rlogin，如同直接執行 rlogin。

不過 rsh 在執行的時候並不會像一般的 login 程序一樣還會問你 password，而是如果你沒有設定 trust table，則 remote host 將不會接受你的 request。

rsh 須要在每個可能會做為 remote host 的機器上設定一個檔案，稱為 .rhosts。這個檔案每一行為兩個部份，第一個是允許 login 的 hostname，第二個部份則是允許 login 的 username。例如，在 ccsun7.csie.nctu.edu.tw 上頭你的 username 為 ysjuang，而你的 home 下面的 .rhost 有以下的一行：

```
ccsun6.cc.nctu.edu.tw u8217529
```

則在 ccsun6.cc.nctu.edu.tw 機器上的 user u8217529 就可以用以下的方法來執行 rsh 程式：

```
% rsh ccsun7.csie.nctu.edu.tw -l ysjuang cat mbox
```

將 ysjuang 在 ccsun7.csie.nctu.edu.tw 上的 mbox 檔案內容顯示在 local host ccsun6.cc.nctu.edu.tw 上。

而如果 .rhost 有這樣的一行，則 ccsun6.cc.nctu.edu.tw 上的 user u8217529 將可以不用輸入 password 而直接經由 rsh 或 rlogin login 到

ccsun7.csie.nctu.edu.tw 來。

注意：

.rhost 是一個設定可以信任的人 login 的表格，因此如果設定不當將會讓不法之徒有可以乘機侵入系統的機會。如果你閱讀 man 5 rhosts，將會發現你可以在第一欄用 + 來取代任何 hostname，第二欄用 + 來取代任何 username。

如一般 user 喜歡偷懶利用 " + username " 來代替列一長串 hostname，但是這樣將會使得即使有一台 PC 上跑 UNIX 的 user 有與你相同的 username，也可以得到你的 trust 而侵入你的系統。這樣容易造成系統安全上的危險。因此本系禁止使用這樣子的方式寫你的 .rhost 檔，如果發現將予以停機直到你找中心的工作人員將其改正為止。同理，如果你的第二個欄位為 +，如 " hostname + "，則你是允許在某一部機器上的"所有" user 可以不用經由輸入 password 來進入你的帳號，是壹種更危險的行為。所以請自行小心。

2. telnet

telnet 是一個提供 user 經由網路連到 remote host。
telnet 的格式如下：

```
telnet [ hostname | ip-address ] [ port ]
```

hostname 為一個像 ccsun1 或是 ccsun1.cc.nctu.edu.tw 的 name address，ip-address 則為一個由四個小於 255 的數字組成的 ip address，如 ccsun1 的 ip-address 為 140.113.17.173，ccsun1.cc.nctu.edu.tw 的 ip-address 為 140.113.4.11。你可以利用 telnet ccsun1 或 telnet 140.113.17.173 來連到 ccsun1。

port 為一些特殊的程式所提供給外界的溝通點，如資工系的 MUD 其 server 便提供一些 port 讓 user 由這些 port 進入 MUD 程式。詳情請參閱 telnet(1) 的說明。

3. ftp

ftp 的意義是 File Transfer Program，是一個很常應用在網路檔案傳輸的程式。ftp 的格式如下：

```
ftp [ hostname | ip-address ]
```

其中 hostname | ip-address 的意義跟 telnet 中的相同。

在進入 ftp 之後，如果與 remote host 連接上了，它將會詢問你 username 與密碼，如果輸入對了就可以開始進行檔案傳輸。

在 ftp 中有許多的命令，詳細的使用方式請參考 ftp(1)，這裡僅列出較常用的 cd，lcd，mkdir，put，mput，get，mget，binary，ascii，prompt，help 與 quit 的使用方式。

ascii 將傳輸模式設為 ascii 模式。通常用於傳送文字檔。

binary 將傳輸模式設為 binary 模式，通常用於傳送執行檔，壓縮檔與影像檔等。

cd remote-directory 將 remote host 上的工作目錄改變。

lcd [directory] 更改 local host 的工作目錄。

ls [remote-directory] [local-file] 列出 remote host 上的檔案。

get remote-file [local-file] 取得遠方的檔案。

`mget remote-files` 可使用通用字元一次取得多個檔案。

`put local-file [remote-file]` 將 local host 的檔案送到 remote host。

`mput local-files` 可使用通用字元一次將多個檔案放到 remote host 上。

`help [command]` 線上輔助指令。

`mkdir directory-name` 在 remote host 造一個目錄。

`prompt` 更改交談模式，若為 on 則在 `mput` 與 `mget` 時每作一個檔案之傳輸時均會詢問。

`quit/bye` 離開ftp。

利用 `ftp`，我們便可以在不同的機器上將所需要的資料做轉移，某些特別的機器更存放大量的資料以供各地的使用者抓取，本校較著名的 `ftp server` 有 NCTUCCCA 與系上的 `ftp.csie.nctu.edu.tw`。這些 `ftp server` 均有提供一個 user 稱為 `anonymous`，一般的"外來客"可以利用這個 `username` 取得該 `server` 的公共資料。不過 `anonymous` 在詢問 `password` 時是要求使用 `anonymous` 的使用者輸入其 `email address`，以往有許多台灣的使用者在使用國外的 `ftp server` 時並沒有按照人家的要求輸入其 `email address`，而僅是隨便打一些字串，引起許多 `internet user` 和管理者的不滿，對台灣的使用者的風評變得很差，因此遵循各 `ftp server` 的使用規則也是一件相當重要的事。

f. 關於通訊用的指令：

1. write

這個指令是提供使用者傳送訊息給另一個使用者，使用方式：

```
write username [tty]
```

2. talk/ytalk/cytalk/ctalk

UNIX 專用的交談程式。會將螢幕分隔開為你的區域和交談對象的區域，同時也可和不同機器的使用者交談。使用方式：

```
talk username[@host] [tty]
```

3. mesg

選擇是否接受他人的 `messege`，若為 `messege no` 則他人的 `messege` 將無法傳送給你，同時他也無法干擾你的工作。使用方法：

```
mesg [-n|-y]
```

4. mail/elm

在網路上的 `email` 程式，可經由此程式將信件 `mail` 給他人。使用方式：

```
mail [username]
mail -f mailboxfile
```

如有信件，則直接鍵入 `mail` 可以讀取你的 `mail`。

`elm` 提供較 `mail` 更為方便的介面，而且可做線上的 `alias`。你可以進入 `elm` 使用上下左右鍵來選讀取的信件，並可按 `h` 取得線上的 `help` 文件。

使用方式：

```
elm [usermae]
elm -f mailboxfile
```

g. 編譯器(Compiler)：

`Compiler` 的用處在於將你所撰寫的程式翻譯成一個可執行檔案。在資工系常用的程式語言是 `C`，`pascal`，`FORTRAN` 等。你可以先寫好一個 `C` 或 `Pascal` 或 `FORTRAN` 的原始程式檔，再用這些 `compiler` 將其翻成可執行檔。你可以

用這個方法來製造你自己的特殊指令。

1. `cc/gcc` (C Compiler)
`/usr/bin/cc`
`/usr/local/bin/gcc`

語法: `cc [-o execfile] source`
`gcc [-o execfile] source`

`execfile` 是你所希望的執行檔的名稱，如果沒有加上 `-o` 選項編譯出來的可執行檔會以 `a.out` 做為檔名。 `source` 為一個以 `.c` 做為結尾的 C 程式檔。請參閱 `cc(1)` 的說明。

2. `pc` (Pascal Compiler)
`/usr/local/bin/pc`

語法: `pc [-o execfile] source`

`execfile` 是你所希望的執行檔的名稱，如果沒有加上 `-o` 選項編譯出來的可執行檔會以 `a.out` 做為檔名。 `source` 為一個以 `.p` 做為結尾的 Pascal 程式檔。請參閱 `/net/home5/lang/man` 中 `pc(1)` 的說明。

3. `f77` (Fortran Compiler)
`/net/home5/lang/f77`

語法: `f77 [-o execfile] source`

`execfile` 是你所希望的執行檔的名稱，如果沒有加上 `-o` 選項編譯出來的可執行檔會以 `a.out` 做為檔名。 `source` 為一個以 `.p` 做為結尾的 FORTRAN 程式檔。

h. 有關列印的指令：

以下為印表所會用到的指令，在本系的印表機有 `lp1` , `lp2` (點矩陣印表機) , `lw` , `sp` , `ps` , `compaq` (雷射印表機) , 供使用者使用。

1. `lpr`
`lpr` 為用來將一個檔案印至列表機的指令。

用法：

`lpr -P[printer] file1 file2 file3`

或

`lpr -P[printer] < file1`

例子：

`lpr -Plp1 hello.c hello.lst hello.map`

`lpr -Plp1 < hello.c`

前者以參數輸入所要印出的檔案內容，後者列印標準輸入檔案(`standard input`) 的內容，因已將 `hello.c` 轉向到標準輸入，故會印出 `hello.c` 的檔案內容。

2. `lpq`
`lpq` 是用來觀察 `printer queue` 上的 Jobs 。

用法：

`lpq -P[printer]`

3. `lprm`
`lprm` 是用來取消列印要求的指令。通常我們有時會印錯，或是誤送非文字檔資料至 `printer` , 此時就必須利用 `lprm` 取消列印 `request` , 以免造成資源的浪費。

用法：

```
lprm -P[ printer ] [ Jobs id | username ]
```

lprm 用來清除 printer queue 中的 Jobs , 如果你使用 Job Id 作為參數, 則它將此 Job 自printer queue 清除, 如果你用 username作為參數, 則它將此 queue 中所有 Owner 為此 username 的 Jobs 清除。

i. 更改個人使用資料：

1. passwd

passwd 是用來更改你的使用密碼, 用法為：

```
passwd [ username ]
```

在使用 passwd 的時候, 它會先問你的舊密碼, 然後詢問兩次要更改的密碼, 確定無誤後才將你的密碼改掉。

2. chsh

chsh 是提供使用者更換 login shell 的指令, 你可經由此更換自己使用的 shell 。

.