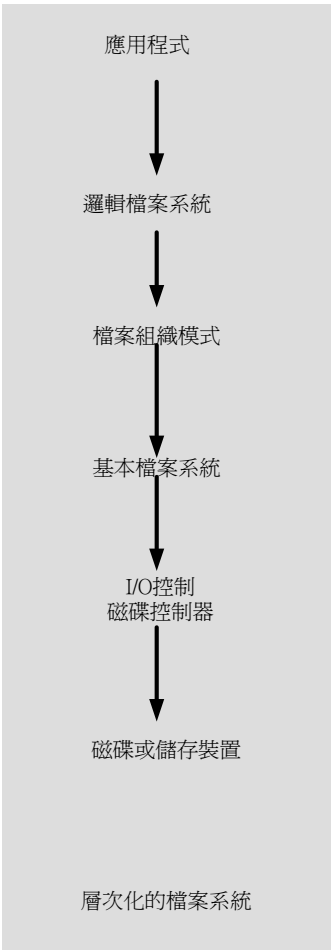
第 15 章
檔案系統管理

第 15 章 檔案系統管理

15-1 檔案和檔案系統結構



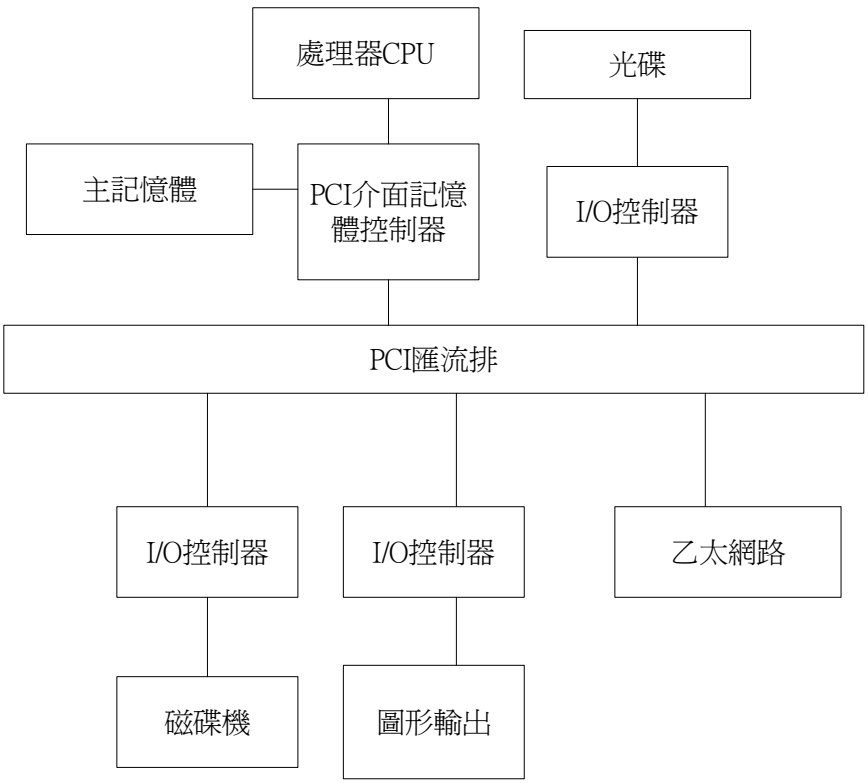
大部份的電腦使用者使用檔案和檔案系統結構。當使用電腦系統時，使用者一般都執行建立檔案、讀取檔案、執行檔案、修改檔案或寫入檔案。因此使用者需要了解在 Linux 上的檔案、檔案如何被管理和組織、檔案在作業系統上如何被實作和檔案如何被儲存在磁碟上。

我們經過作業系統來方便有效的存取磁碟，以及有效的儲存、找到、以及重新取出資料。作業系統如何去定義檔案系統，讓我們使用者能夠輕易的使用檔案系統，以及檔案的定義和它的性質。我們也需要建立一些快速的演算法及資料結構，來快速的從檔案系統存取與找尋資料。檔案系統是由許多不同的操作層次所組成。

- 應用程式：應用程式呼叫邏輯檔案系統來刪除、修改或建立一個檔案。檔案在 Unix 中，應用程式將目錄當成檔案的一種。
- 邏輯檔案系統：包括檔案系統的組織及演算、資料結構，也包含了檔案的保護與安全。邏輯檔案系統使用目錄結構。
- 檔案組織模組：將邏輯檔案系統轉換成基本檔案系統。
- 基本檔案系統：為發出一般的命令給裝置驅動程式，指示要讀取或寫入磁碟的位址的磁區(例如：第

三個磁盤，第六個磁軌，第五個磁區)。

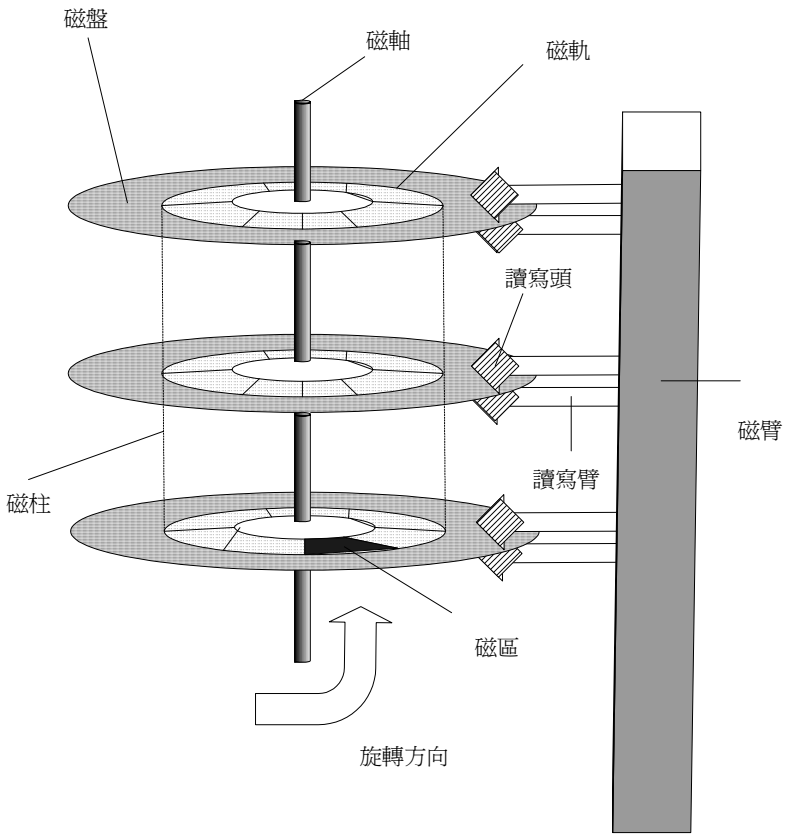
- I/O 輸入輸出控制器：磁碟機介由 I/O 匯流排將磁碟裝置組在電腦上。在匯流排上資料的傳送是由 I/O 控制器所控制。如果要完成一個磁碟的輸入/輸出動作，電腦要將命令給 PCI 介面記憶體控制器，然後再將主控制器的信號傳給 I/O 控制器，然後再由磁碟控制器來控制操作硬碟，來讀取或存入資料。磁碟控制器內一般都有緩衝的快取記憶體，來放置要傳送或輸入的資料。



- 磁碟裝置：磁碟裝置就是存放我們資料的地方。

磁碟是我們儲存檔案資料最主要的地方。磁碟機是由許多的磁盤所組成，就像我們的光碟片一樣，儲存著許多的資料，而磁盤是雙面的儲存資料，我們介著讀取頭來讀取磁盤上的資料。磁盤是由許多的磁軌所組成，而磁軌又由許多的磁區所組成。磁

碟的轉速目前為每分鐘 7200 轉上下，轉速越快讀取資料的速度就越快，但價格越高。一般資料由磁碟傳到我們記憶體的時間可分為傳送時間和定位時間。定位時間是讀寫頭到指定的磁區所需要的時間。傳送時間是傳送資料由磁碟裝置到記憶體的時間。

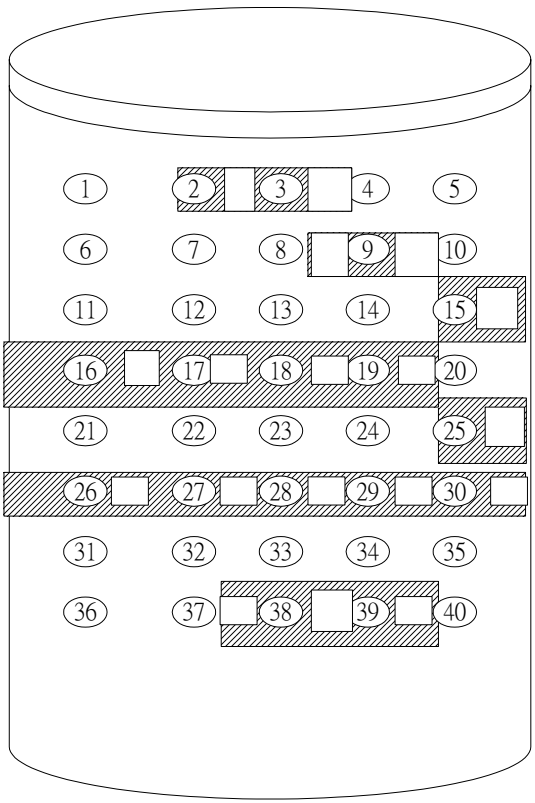


磁碟機裝置

我們可以將檔案資料存放在我們的磁碟上。我們可以將磁碟的儲存空間分配給檔案，方便我們快速的存取檔案，而有連續的配置磁碟空間、鏈接式的配置磁碟空間和索引式的配置磁碟空間。

連續性的分配磁碟空間需要讓檔案使用連續性的磁碟位置。例如 chaiyen 的檔案開始的位置是在 2 的地方，而它使用了兩個區塊的連續磁碟空間。Mail 的檔案開始的位置是在 15 的地方，而它使用了五個區塊的連續磁碟空間。

檔案概念

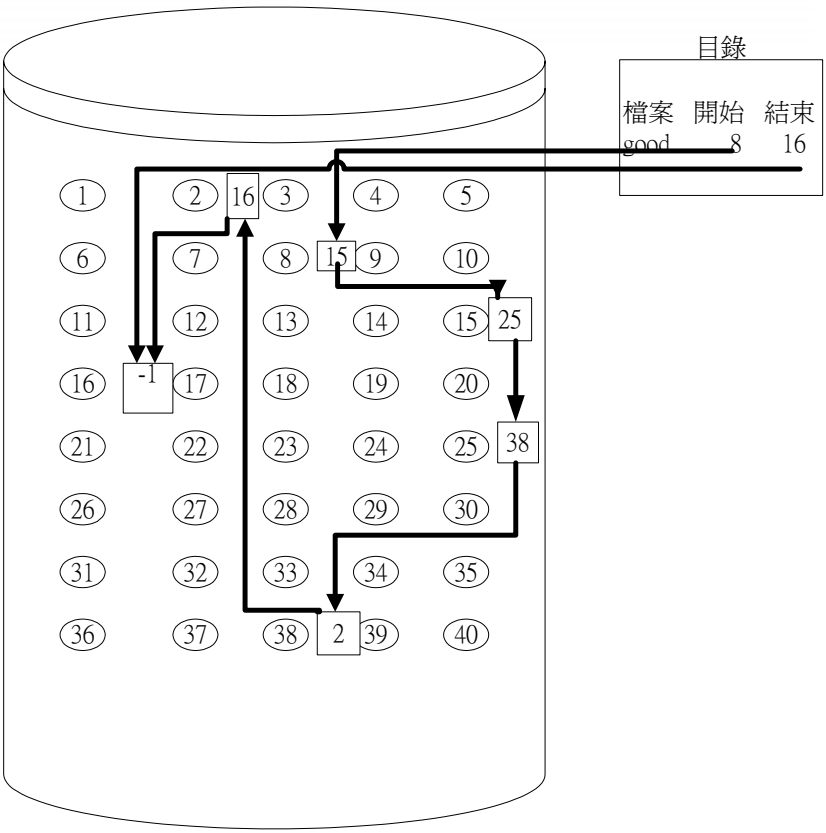


目錄

檔案	開始	長度
chaiyen	2	2
mail	15	5
init.d	8	2
man	25	6
good	37	3

磁碟空間的連續分配

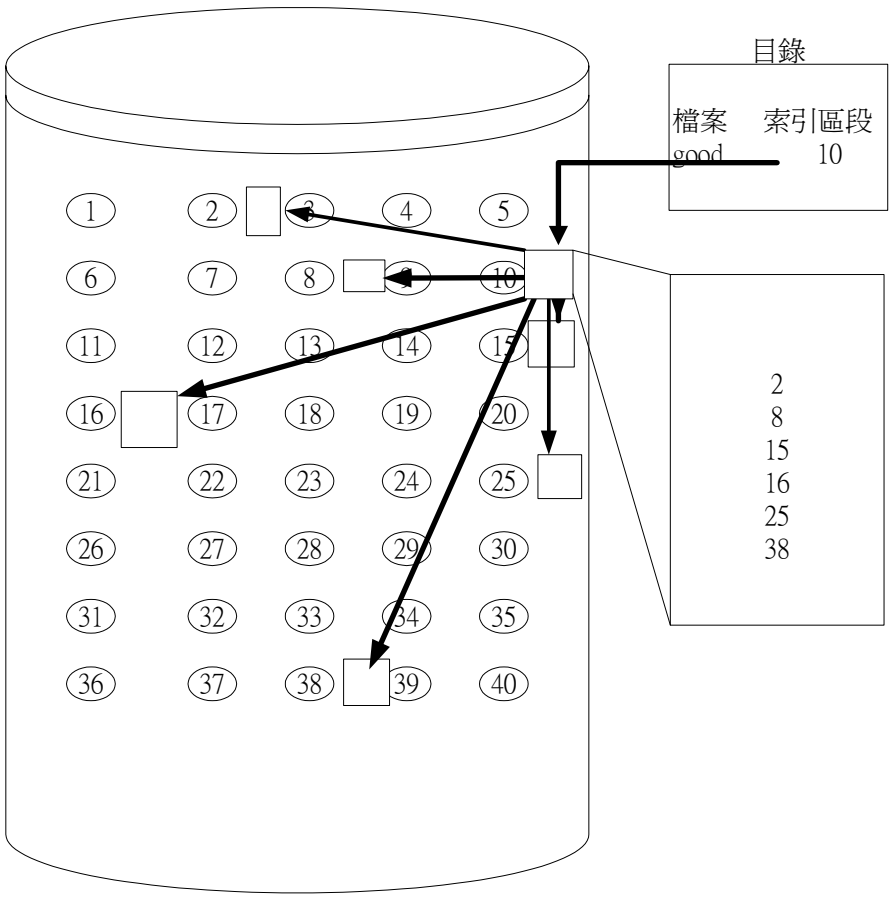
我們使用鏈結來分配檔案的磁碟空間。檔案是用鏈結的方式來表示，因為將硬碟分割成目錄和檔案，目錄標示著所有檔案的資訊，包括檔案的開始與結尾，在下圖中，檔案 good 的開始是在 8 的地方，而結尾是在 16 的地方，結尾以負 1 來表示，而檔案中包含了下一個檔案的位址指標(節點)。



磁碟空間的鏈結分配

我們可以使用索引分配磁碟空間給檔案。每個檔案都有自己的索引區段，那就是一個磁碟中區段位置所組成的陣列。在我們檔案 good 中，其索引區段為 10，而其索引區段放置所有區段的位置指標。我們可以使用多層索引，來增加檔案在磁碟中的儲存分配空間。作業系統使用第一層索引來找到第二層索引區塊，再使用第二層區塊找到檔案的資料區塊。如果我們使用 4096 位元組的區塊，則我們可以儲存 1024 個 4 位元組的指標在索引區塊中。因此我們使用兩層的索引可達 1048576 個資料區塊，則我們可以分配 4G 位元組的磁碟空間給檔案。UNIX 使用組合的方式將 inode 的 12

個指標直接指向直接區段，而這些是分配給較小的檔案。而第十三個指標為指向間接區段，也就是使用單層的索引區段。第十四個指標為指向雙層的索引取段，又稱為雙重間接區段。最後一個指標使用三層的索引，又稱為三重間接區段。



磁碟空間的索引式分配

15-1-1 檔案的型態

檔案的型態分為簡單的檔案型態、目錄、符號連結(symbolic link)、特別檔案裝置、管線。

簡單的檔案型態：簡單的檔案型態一般有執行檔、目的檔、備份或壓縮檔、觀看或圖型檔、函式庫檔、文字檔、批次檔、原始檔、網頁檔。

檔案型態	常用延伸部份	說明
執行檔	Exe、com、bin	可直接被機器執行的檔案。
目的檔	.o、.ob	編譯成機器語言，但還未鏈結的檔案。
備份或壓縮檔	Arc、tar、zip	將檔案壓縮成較小的檔案
觀看或圖型檔	Ps、gif、jpeg	Ascii 或二進位可列印或觀查的檔案。
函式庫檔	Lib、a	放在函式庫中的函式。
文字檔	Txt、doc	文書資料與文件。
批次檔	Bat、sh	命令直譯器的指令。
原始檔	C、cpp、asm、fla、java	程式語言未經過編譯的原始檔。
網頁檔	Html	可以使用瀏覽器閱讀的檔案。

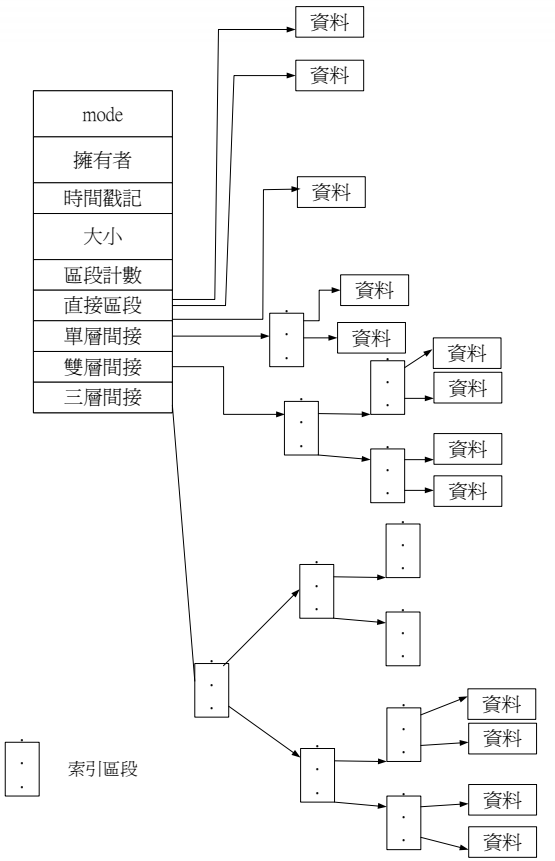
- 目錄檔：目錄檔包含了其它檔案的檔名還有指向這些檔案資訊的指標內容，具有讀取目錄權限的程式都可以讀取指定的目錄內容，但只有核心程式才能寫入目錄檔。目錄是由節點 inode(指向檔案的指標)和檔案名稱所組成。

節點	檔案名稱

- 符號連結檔：符號連結是指向另一個檔案的檔案類型，它的資料內容是存放另外一個檔案的位址。符號連結檔可以讓我們更改檔案的名稱，而不用再複製檔案，因為我們使用符號指標檔指向檔案。

- 特別檔案裝置：用於系統的裝置。特別檔案裝置是存取硬體裝置的方法，包含鍵盤、硬碟、CD-ROM 光碟機、軟碟機、和印表機。每一種硬體裝置都有它自己的特別檔案。特別檔案分為字元特別檔和區塊特別檔。字元特別檔就像是鍵盤，而區塊特別檔則為硬碟裝置。特別檔都是放在/dev 的目錄。這個目錄至少包含一個和電腦連接的裝置。應用程式或指令讀取或寫入特別裝置檔案的方法就像讀取或寫入一般檔案一樣。這是因為在 Linux 中輸出和輸入是獨立於裝置(例如 fd0 為軟碟 0、fda 為硬碟 a、lp0 為印表機 0、tty 為終端機)。我們稱各種不同的裝置模擬這實體的檔案裝置為虛擬裝置(pseudo devices)。虛擬裝置允許我們和 LINUX 作業系統交談。
- 管線：用於行程間相互溝通的檔案。Linux 擁有好幾個機制來允許我們行程間的互相溝通。這個機制稱為內部行程溝通機制 Ineterprocess communication(IPC) mechanisms。一般我們時常使用的稱為管線 pipes、FIFO 和插槽 sockets。Pipe 管線是當作父行程和子行程溝通的緩衝記憶體。FIFO 是兩個行程間的緩衝區。Socket 插槽是核心記憶體緩衝區，它允許兩台電腦經過網路進行溝通。

這是 Unix 也是 Linux 的檔案格式 inode，inode 裏面包含了擁有者、使用者可以存取的權限、也有時間戳記(裏面包含了何時這個檔案被修改)、也包含了檔案的大小。每一個檔案的區段大小為 4k，在 Linux 中系統的方式為保存裝置目錄中的前 15 個索引區段指標，而指標的前 12 個指向直接區段，因此只要檔案小於 48k 的，就可以直接存取，速度較快；而單層間接及雙層間接和三層間接的第一個指向結點，裏面的內容是下一層的位址，這樣可以讓我們的檔案指標指向更多的檔案。在 32 位元的檔案指標最多可以達到 4G 位元組。



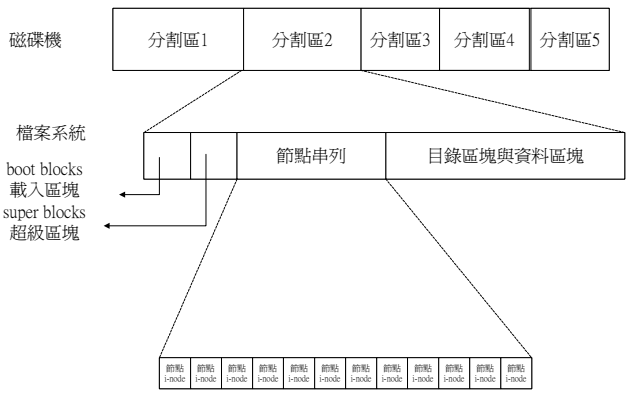
這是檔案系統的結構。我們將磁碟機分割成數個分割區，而每個分割區都有自己的檔案系統。每個檔案系統都有載入區塊(boot blocks)、超級區塊、節點串列和目錄區塊及資料區塊。

載入區塊包含了該分割區被系統啟動的資訊。

超級區塊包含了分割區的詳細資料，包含了分割區的區塊數目、區塊的大小、空的區塊數目、節點的數目。

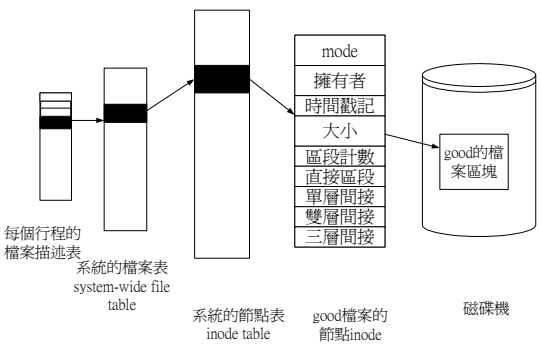
目錄區塊通常是用來組織檔案。

節點包含了存取權、擁有者、大小、資料區塊的位址。



磁碟機,磁碟分割和檔案系統

當應用程式要作檔案的輸入輸出時，它一定要開啟檔案，然後再作讀取或寫入的動作。當我們在 Linux 上要開啟檔案時，都會和檔案描述表 file descriptor 有關。如下圖。檔案描述表的描述符號 0 是表示標準輸入 stdin、檔案描述表的描述符號 1 是表示標準輸出、檔案描述表的描述符號 3 是表示錯誤 stderr。核心使用的檔案描述符號索引每個行程的檔案描述表，並且得到指標來指向系統的檔案表。然後，系統檔案表包含的指標指向系統節點表的檔案節點。當檔案的節點 inode 被存取時，就可以讀取整個檔案了。



讀取檔案的過程

15-1-2 檔案系統結構

作業系統的檔案系統結構有檔案系統如何被組織、檔案系統如何被儲存和如何管理檔案系統這三點。

Linux 的檔案系統結構是屬於階層式。因此，檔案系統的開始是由根目錄 root 開始往下長，就像一棵倒長的樹一樣。Linux 作業系統包含了數百個目錄和檔案，就像下列的檔案系統結構。

了解 LINUX 的檔案結構。

LINUX 的檔案結構就像是一棵數(TREE)，它是由/(根)開始往下發展。在圖中方型代表著目錄，圓形代表著檔案。

LINUX 的檔案結構就像一棵樹(TREE)，而微軟將檔案分割成好幾個磁區例如 C:/(C 碟)、D:/(D 碟)、E:/(E 碟)，而我們 LINUX 是使用掛載(MOUNT)的方式來讀區不同磁區。

我們使用 mount 指令來掛載/mnt 目錄底下的 CDROM 光碟機。

/根目錄，它是整個 LINUX 檔案系統的起點。在 WINDOWS 上它是以 C 碟，D 碟、E 碟，而在 LINUX 上則是以掛載(mount)的方式。

/lib 是放置函式(library)的地方。

/etc 存放和系統有關的程式和檔案的地方。

/tmp 放置暫時存放的檔案。

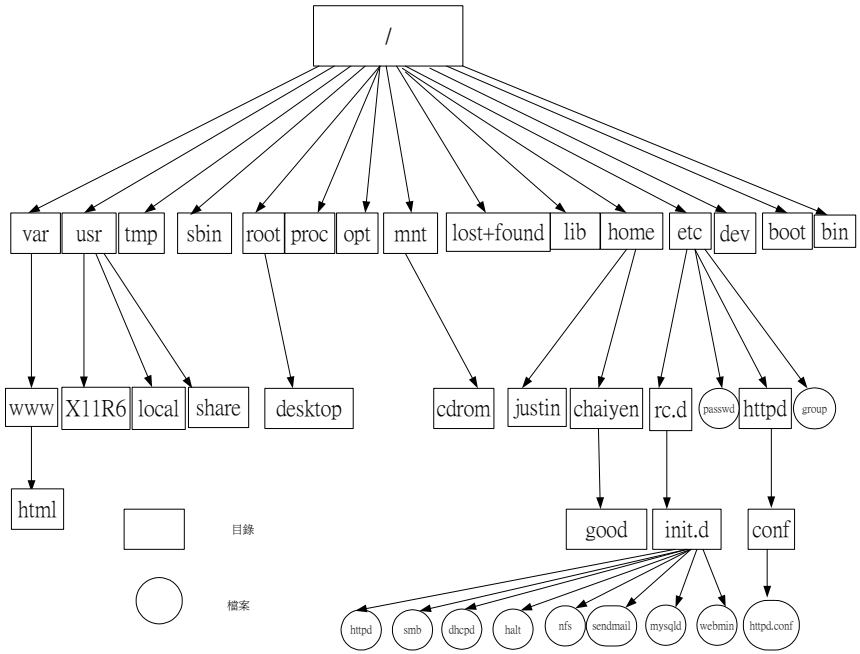
/var 放置一些內容時常改變的檔案。

網頁的預設放置地點就是在/var/www/html。

/mnt 放置光碟機或軟碟機的地方，我們經常使用 mount 指令來掛載此外部 device。

/usr 存放給使用者用的重要子目錄。

/bin 存放基本的 linux 指令的地方。



典型的檔案系統結構

當我們登錄作業系統時，作業系統會將我們放到指定的目錄/home/使用者。例如我們使用者 chaiyen 登錄時，就會到作業系統指定的目錄/home/chaiyen 中。我們目前所在的目錄稱為逗點 dot．，而我們的上一層目錄稱為逗點逗點 dotdot．．。

檔案路徑分為兩種，一種為絕對路徑，一種為相對路徑。路徑名稱可以用三種方式來指示，從根目錄開始(/)、從目前的工作目錄開始、和從使用者的家目錄開始。當路徑名稱是由家目錄開始時，我們稱為這是絕對路徑。例如/etc/rc.d/init.d/httpd 就是絕對路徑。

路徑從目前的目錄開始或使用者的家目錄開始時，我們稱為相對路徑。當我們在/etc/rc.d 的目錄中，我們要執行啟動 httpd 的程式，我們可以使用 . /init.d/httpd　start 相對路徑來啟動。我們也可以使用絕對路徑/etc/rc.d/init.d/httpd　start 來啟動。我們也可也可以在任何地方使用 cd ~/來跳到我們的家目錄。

在 Linux 中我們可以使用掛載 mount 到不同的分割區來使用檔案系統，而不用像微軟使用命名 A：、B：、C：、D：、E：分割區。在 Linux 中，我們可以介由指定路徑來使用檔案系統，而不用考慮它們的分割區。

/(根目錄)：根目錄是檔案系統最上層的目錄，它包含所有的目錄和檔案。

我們可以使用 mount 指令來掛載系統。

語法：

指令:mount

參數：

-a：掛載/etc/fstab 中所設定的所有檔案系統。

-f：模擬掛載，但不會真的掛載進去。

-F：使用平行掛載，來加快掛載的時間。

-L 標籤：掛載指定標籤的檔案系統。

-r：以唯讀的方式掛載。

-w：以可讀寫的方式掛載檔案系統。

-o 選項：

async 為以非同步方式作輸入/輸出。

atime 每次存取後更新節點 inode 的時間。

auto 為自動掛載設定為 auto 的裝置。

dev 可解讀檔案系統上的區塊裝置。

exec 可執行二進位檔。

nouser 讓一般使用者無法掛載。

ro 以唯讀模式掛載。

rw 以可讀寫方式掛載。

Suid 開啟 SUID(設定使用者 ID)。

sync 以同步的方式啟動執行輸入/輸出的動作。

user 讓一般使用者可以掛載。

我們使用 mount 來掛入檔案系統而使用 umount 來解除掛載。

Linux 是使用掛載的方式來將檔案掛入使用。

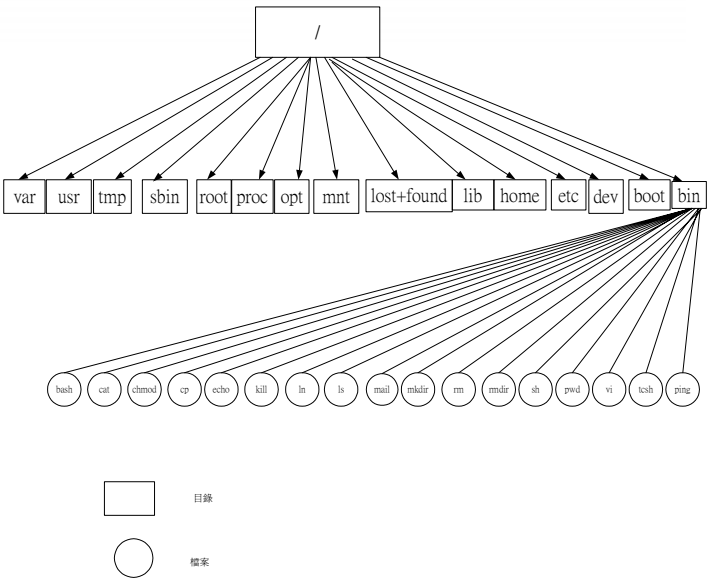
我們使用 mount /mnt/cdrom 來掛載光碟，而使用 umount /mnt/cdrom 來解除掛載。

```
[root@flash chaiyen]# mount /mnt/cdrom
[root@flash chaiyen]# umount /mnt/cdrom
```

我們使用 mount –a 來掛入所有在/etc/fstab 設定的檔案系統。

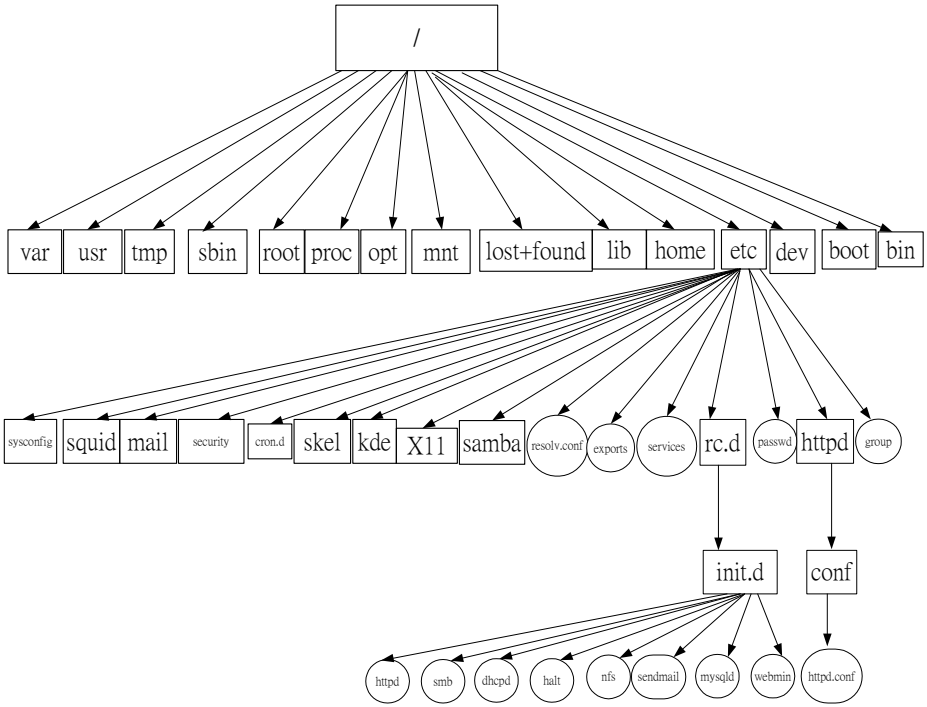
```
[root@flash chaiyen]# mount -a
```

- /bin：為主要執行檔的目錄。所有在這目錄的檔案不是執行檔，就是連結到可執行檔的符號檔案(symbolic links)。一些主要的指令像是 bash、cat、chmod、cp、echo、kill、ln、ls、mail、mkdir、rm、rmdir、sh、stty、su、tcsh、uname、more、move、ps、pwd 和 vi 檔都在這/bin 的目錄下。在/bin 下也包含一些重要的網路指令 ping、netstat、hostname。



/bin目錄下的檔案

- /dev：這/dev 目錄為裝置目錄，它包含相關裝置的檔案系統，如終端機、磁碟機、光碟機、軟碟機和印表機。我們可以將這些裝置分為字元特別檔和區塊特別檔。字元特別檔為以字元輸入輸出的裝置，如鍵盤。區塊特別檔為以區塊輸入輸出的裝置，如磁碟機。
- /etc：/etc 目錄存放和主機系統有關的程式和檔案的地方。這些檔案和目錄包含系統的組態。一些在這裏的檔案和目錄為 X11、bashrc、csh.login、crontab、group、inittab、lilo.conf、linuxconf、localtime、motd、passwd、pine.conf、profile、securetty、skel、shells 和 zshrc。這 X11 目錄包含 X 視窗的組態檔案。一些在/etc 的網路相關檔案或目錄包含 exports、ftpusers、gateways、host.conf、hosts、hosts.allow、hosts.deny、hosts.equiv、hosts.lpd、httpd、inetd.conf、inputrc、lynx.cfg、mail.rc、networks、rc.d、resolv.conf、services、snmp、uucp、news、printcap、protocols。/etc/passwd 是放置使用者資料的檔案，/etc/shadow 是放置密碼加密的檔案、/etc/group 是放置群組資料的檔案、/etc/shells 是放置指令直譯器的目錄(BASH、TCSH)、/etc/skel 是放置登錄 shell 時所初始化的檔案。

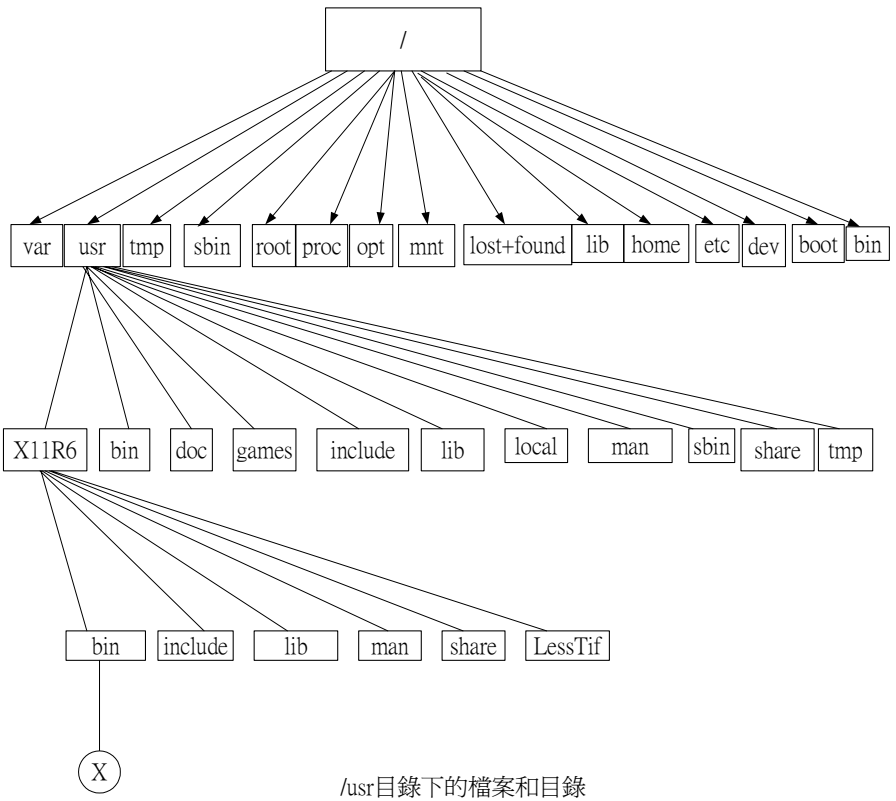


- /home：/home 目錄存放使用者的家目錄。在我們的個人電腦系統，家目錄存放著使用者們的家目錄，如/home/chaiyen 為使用者 chaiyen 的目錄。我們也可以將/home 目錄當作是網路檔案系統 NFS 輸出的目錄。
- /lib：/lib 是放置各種語言的函式庫。Linux 系統包含 C、C++和 FORTRAN 的函式庫。我們在開發軟體時，可以使用這些函式庫。標準的 C 函式庫為/lib/libc.so*，數學函式庫 libm.so.*和可動態連結的/lib/ld/so。/lib/modules 目錄包含可載入的核心模組，而/lib 包含所有基本的函式。
- /mnt：/mnt 是被系統管理者暫時使用來掛載檔案系統。這個目錄/mnt 包含光碟機 cdrom、磁碟機 disk 和軟碟機 floppy 的掛載點。因此掛載光碟機裝置/mnt/cdrom，可以讓我們使用在/cdrom 上的檔案。

- /opt：/opt 目錄是給我們安裝一些其它的軟體套件。
- /proc：/proc 目錄包含了行程和系統的資訊。
- /root：/root 目錄是用來當做超級使用者 root 的家目錄。這個目錄只有超級使用者 root 可以使用，因此受到保護。
- /sbin：/sbin、/usr/sbin、usr/local/sbin 包含系統管理者的工具和只有超級使用者可以使用的指令。超級使用者的指令為 getty、init、update、swapon 和 swapoff。暫停或關閉系統的指令為 halt、reboot 和 shutdown。系統管理工具為 fdisk、fsck、fsck.ext2、fsck.minx、mkfs、mkfs.ext2、mkfs.minix、mkfs.msdos 和 mkfs.vfat。網路指令為 ifconfig 和 route。
- /temp：/temp 目錄為放置暫時檔案的目錄。我們也可以將我們的暫時檔案放置到/temp 目錄中。在/temp 目錄下的檔案每經過一段時間就會被系統移除。我們可以使用 sticky bit 來設定檔案，只有檔案的擁有者才能刪除它。
- /usr：/usr 目錄包含 X11R6、bin、doc、games、include、lib、local、man、sbin、share、src 和 tmp。

/usr 的子目錄	說　明
X11R6	為 X 視窗系統的目錄。
bin	包含大部份的使用者指令、perl 或 python 直譯器。
doc	包含各種工具、函式庫、應用程式和直譯器的程式說明文件。如 gcc、cc、Xfree86、GNOME 說明文件。
games	包含遊戲軟體。
include	包含 C 或 C++的標頭檔，如/usr/include/netdb 和 /usr/include/netinet。系統指定的標頭檔/usr/include/sys。
lib	包含目地檔、函式檔、和內部二進位檔。一些子目錄包含該應用程式或工具的函式庫，如 gcc-lib、xemacs、perl5、netscape。
local	包含系統使用者在本地端安裝的軟體。
man	包含 Linux 指令、工具和應用程式的手冊。
sbin	包含系統使用者和 daemon 專門使用的指令。

share	包含可被其它電腦平台使用的唯讀檔案。
src	包含 Linux 和包裝管理軟體的原始碼。
tmp	包含符號連結到/tmp。



/usr目錄下的檔案和目錄

- /var：/var 目錄存放著當系統執行時會更動的變數資料。
  /var/spool/mail 存放著郵件，當我們讀取過這郵件，這郵件會被放到使用者家目錄的 mbox 檔案中。

15-1-3 瀏覽檔案結構

我們可以使用一些指令來瀏覽 Linux 檔案系統、建立檔案和目錄、設定檔案的屬性、設定使用者家目錄的路徑、和檔案型態。

我們可以使用 pwd 指令來顯示我們目前所在的地方。這顯示我們目前所在的目錄是/usr/X11R6/bin。

```
[root@flash bin]# pwd
/usr/X11R6/bin
```

我們可以使用 pwd 指令來顯示目前的工作目錄。

語法：

指令：pwd

參數：

--help：pwd 指令說明

我們可以使用 echo $HOME 指令來顯示我們的家目錄。因為我們使用超級使用者，所以顯示超級使用者的家目錄為/root。

```
[root@flash bin]# echo $HOME
/root
```

我們可以使用 echo 指令來在螢幕上顯示字串。

語法：

指令：echo 參數 字串

參數：

-E：不直譯跳脫字元 \ 。

-e：加強直譯跳脫字元

\b：刪除前一個字元。

\c：不換行送出。

\f：換行但游標不移動。

\n：換行且游標移置行首。

\\：插入\字元。

\t：插入 Tab 鍵。

我們可以使用 cd 指令來切換目錄。我們的工作區從/home 目錄跳到/usr 目錄。

```
[root@flash home]# cd /usr
[root@flash usr]#
```

我們使用 cd 指令來切換到想要到的目錄中。

語法：

指令：cd 目錄

我們可以使用 ls 指令來顯示在目錄中的檔案的內容。

語法：

指令：ls 參數 路徑名稱

參數：

-F：用來辨示檔案的型態。顯示在目錄後面為/；顯示在符號連結後面為@；顯示在執行檔後面為*。

-a：顯示所有檔案的名稱，包含隱藏檔。

-i：顯示節點的數量。

-l：顯示檔案的屬性及內容，包含存取檔案的權限、連結數、擁有者、群組、檔案大小、和修改時間。

-c：以最後修改的時間來排序檔案。搭配-l 參數使用。

-R：遞迴的顯示子目錄。

我們使用 ls -F 用來辨示檔案的型態。bin 為目錄/，tmp 為符號連結檔案@。

```
[root@flash usr]# ls -F
bin/   games/              include/   libexec/     mtab   src/
dict/  good.c              kerberos/  local/       sbin/  tmp@
etc/   i386-glibc21-linux/ lib/       lost+found/  share/ X11R6/
```

在 root 目錄中，我們使用 ls –a 來觀看所有在 root 目錄下的檔案。．逗點表示現在的目錄。．．逗點逗點表是上一層目錄。

```
[root@flash root]# ls -a
.                               .gconfd             .mozilla
..                              .gimp-1.2           .nautilus
anaconda-ks.cfg                 .gnome              .qt
.bash_history                   .gnome-desktop      .rhn-applet.conf
.bash_logout                    .gnome_private      .sawfish
.bash_profile                   .gtkrc              .tcshrc
.bashrc                         .gtkrc-kde          .viminfo
.cshrc                          .ICEauthority       .Xauthority
.DCOPserver_flash.aasir.com__0  install.log         .xcdroast
.DCOPserver_flash.aasir.com_:0  install.log.syslog  .xftcache
dead.letter                     .kde                .Xresources
Desktop                         mbox                .xsession-errors
.first_start_kde                .mcop               快照1.png
.gconf                          .mcoprc
```

這是在使用者目錄下的一些重要隱藏檔案。

檔案名稱	說明
.	逗點為目前工作的目錄
. .	逗點逗點為上一層目錄
. bash_history	我們先前下達 Bash 指令的歷史記錄。
. bashrc	設定 Bourne Again shell 的檔案。
. cshrc	設定 C shell 的檔案。
. exrc	設定 vi 編輯器的檔案。
. login	當登入作業系統時的 C shell 設定檔。
. profile	當登入作業系統時的 Bourne 或 Korn shell 的設定檔。

我們使用 ls -i 顯示檔案節點的數量。

```
[root@flash root]# ls -i
 310736 anaconda-ks.cfg   310690 install.log          310849 快照1.png
 310845 dead.letter       310691 install.log.syslog
  98420 Desktop           310846 mbox
```

我們使用 ls -l 顯示檔案的屬性及內容，包含存取檔案的權限、連結數、擁有者、

群組、檔案大小、和修改時間。

```
[root@flash root]# ls -l
total 140
-rw-r--r--    1 root     root         1138  8月 23 00:04 anaconda-ks.cfg
-rw-------    1 root     root            3  8月 26 22:58 dead.letter
drwx------    3 root     root         4096  8月 31 23:17 Desktop
-rw-r--r--    1 root     root        42595  8月 23 00:01 install.log
-rw-r--r--    1 root     root         4096  8月 22 23:51 install.log.syslog
-rw-------    1 root     root        35588  8月 31 17:32 mbox
-rw-r--r--    1 root     root        41114  8月 30 08:55 快照1.png
```

我們可以使用 echo ~來顯示我們使用者自己的家目錄。

```
[root@flash home]# echo ~
/root
```

我們可以使用 mkdir 來建立目錄，也可以使用 rmdir 來移除目錄。我們可以使用 vi 編輯器、pico 編輯器或 emacs 編輯器來編輯或建立檔案。

我們使用 mkdir 來建立目錄。

語法：

指令：mkdir　參數　目錄

參數：

-m <目錄存取權限>：建立目錄時，同時建立目錄的存取權限。

-p <指令目錄路徑>：同時建立在指定目錄路徑中沒有存在的上一層目錄。

我們使用 mkdir foo 來建立 foo 目錄。

```
[root@flash home]# mkdir foo
```

我們使用 mkdir –p /home/be/good 來建立 good 目錄和上一層的 be 目錄。

```
[root@flash home]# mkdir -p /home/be/good
```

我們使用 mkdir ~/best 來在使用者家目錄下建立 best 目錄。

```
[root@flash home]# mkdir ~/best
```

我們使用 rmdir 來刪除目錄。

語法：

指令：rmdir 參數 目錄

參數：

-p：同時移除上一層的空目錄。

我們使用 rmdir foo 來移除 foo 目錄。

```
[root@flash pop]# rmdir foo
```

我們使用 rmdir –p /home/be/good 同時移除上一層的空目錄 be 和目錄 good。

```
[root@flash home]# rmdir -p /home/be/good
```

我們使用 rmdir ~/best 將使用者家目錄下的 best 目錄給移除。

```
[root@flash home]# rmdir ~/best
```

我們可以使用 ls -l 指令來觀看檔案的屬性。我們可以看到 dhcpc 為目錄，而 dhcpcd 為符號連結檔案。符號連結檔案 dhcpcd->(連結到)dhcpc 檔。

```
[root@flash etc]# ls -l
drwxr-xr-x  2 root root  4096 4月  16 01:19 dhcpc
lrwxrwxrwx  1 root root     5 8月  22 23:13 dhcpcd -> dhcpc
-rw-r--r--  1 root root  2434 3月  25 09:23 DIR_COLORS
-rw-rw----  1 root root   857 4月   2 22:31 diskcheck.conf
-rw-rw-r--  1 root disk     0 3月   1  2002 dumpdates
```

第一欄位　第四欄位　第五欄位　第七欄位

欄位	說明
第一欄位的第一個字元	檔案的類型 d 代表目錄 - 代表一般檔案 l 代表 symbolic link 符號檔案 b 代表區塊特別檔案 c 代表字元特別檔案 s 代表 socket 插槽檔案 p 代表 FIFO 管線檔案
第一欄位其它的字元	2~4：檔案擁有者對檔案的權限 5~7：群組使用者對檔案的權限 8~10：其他使用者對檔案的權限
第二欄位	檔案的連結數目
第三欄位	檔案的所有者
第四欄位	檔案所有者所在的群組
第五欄位	檔案大小
第六欄位	檔案建立或最後修改時間
第七欄位	檔名

我們可以使用 file /*指令來識別檔案類型。

```
[root@flash etc]# file /*
/bin:         directory
/boot:        directory
/dev:         directory
/etc:         directory
/home:        directory
/home.lock:   ASCII text
/initrd:      directory
/lib:         directory
/lost+found:  directory
/misc:        directory
/mnt:         directory
/net:         directory
/opt:         directory
```

語法：

指令：file 參數 檔案串列

參數：

-f 檔案：指令名稱檔。

我們在/etc 目錄下使用 file init.d 來檢查 init.d 的檔案類型。這是一個符號連結檔。

```
[root@flash etc]# file init.d
init.d: symbolic link to rc.d/init.d
```

15-2 檔案安全管理

在 Linux 作業系統上，我們有些檔案很重要，只有系統或經過授權的人才能使用，這樣才能保護我們系統的安全。因為有一些檔案是只有部份指定的人才能存取，以免不小心被他人刪除或修改，因此檔案的安全管理是非常重要的。

15-2-1 存取權限

當使用者被分配帳號時，Linux 管理系統也會分配該使用者的群組給使用者。一個使用者可以屬於多個群組。所有的群組和它的成員都在/etc/group 的檔案中。Linux 擁有一個特別的使用者，他就是系統的超級使用者 root，而他的使用者 ID 為 0。在 Linux 上有三種和檔案存取權有關的型態：讀取 read(r)、寫入 write(w)、執行 execute(x)。讀取 read 權允許我們讀取檔案。寫入 write 權允許我們寫入檔案。執行(execute)權允許我們執行檔案。

Linux 的檔案使用者可以分為檔案的擁有者、群組和一般使用者這三種型態。而 Linux 的檔案存取權限可分為讀取(r)、寫入(w)和執行(x)三種。因此 Linux 的檔案型態總共有九種不同的使用者存取權。

X 的值可以為 true(1)或 false(0)。當為 true(1)時表是允許其使用檔案的權力，當為 false(0)時表示不允許其使用檔案的權力。因為一個使用者的每一個存取檔案型態都要有一個位元來表示，所以用三個位元來表示該使用者的檔案存取權，因此該使用者就有八種可能型態的檔案存取權(數字代表從 0 到 7)。0 代表不可寫、不可讀、不可執行。7 代表可讀、可寫、可執行。

這九個位元可以用來表示這三種使用者存取檔案的三種存取權。而我們可以使用 3 個數值來表示這個檔案的使用者存取權(從 000 到 777)。第一個數值是擁有者的檔案存取權，第二個數值是群組的檔案存取權，第三個數值是其他使用者的檔案存取權。不允許存取檔案 false 也代表了-(dash)，而允許存取檔案 true 則為 r、w 或 x。因此檔案的存取權 0 代表了---，而檔案的存取權 7 則代表了 rwx。

在 Linux 上檔案的存取權。

使用者型態	讀取(r)	寫入(w)	執行(x)
使用者(u)	X	X	X
群組(g)	X	X	X
其他使用者	X	X	X

檔案存取的權限。這是二進位的檔案存取權限。x 代表 1，w 代表 2，r 代表 4。

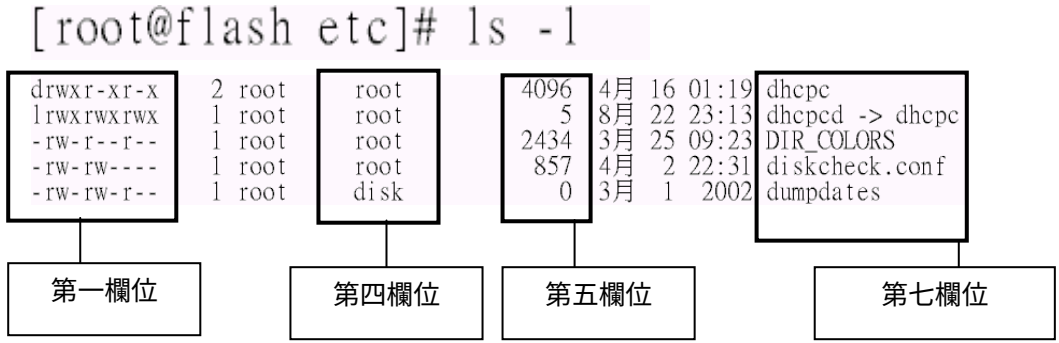
r	w	x	代表數字	說明
0	0	0	0	不可讀、不可寫、不可執行
0	0	1	1	只可以執行
0	1	0	2	只可以寫入
0	1	1	3	可以寫入、可以執行
1	0	0	4	只可以讀取
1	0	1	5	可以讀取、可以執行，但不可寫入
1	1	0	6	可以讀取、可以寫入，但不可執行
1	1	1	7	可讀、可寫、可執行

我們可以使用 ls -l 指令來觀看檔案的屬性。第一欄位的第二到第八個字元是存取檔案的權限。第一個 dhcpc 的目錄檔案權限為 rwxr-xr-x 表示檔案擁有者 root 可以讀取檔案、寫入檔案和執行檔案，但 root 群組只可以讀取和執行檔案，而其他使用者也一樣只可以讀取和執行檔案。

對於 dhcpcd 符號連結檔，使用者存取權限為 rwxrwxrwx。這代表 root 使用者、群組和其它使用者都可以讀取、寫入和執行。

對於 diskcheck.conf 一般檔案，使用者存取權限為 rw-rw----。這代表 root 使用者

可以讀取與寫入檔案，root 群組也可以讀取與寫入檔案，但是其他使用者則沒有任何存取檔案的權限。



15-2-2 改變檔案的存取權限

我們可以使用 chmod 指令來改變檔案的存取權限。

語法：

指令：chmod 參數 權限代表數字 檔案或目錄

chmod 參數 權限符號 檔案或目錄

參數：

-R：遞迴的處理所有的檔案和子目錄。

-f：強制指定存取權。

權限符號

使用者	運算子	權限
u(檔案擁有者)	+增加權限	r 讀取位元
g(群組)	-移除權限	w 寫入位元
o(其他使用者)	=設定權限	x 執行位元
a(全部使用者)		u 檔案擁有者目前權限
ugo(全部使用者)		g 群組目前權限
		o 其他使用者目前權限
		l 鎖定權限
		設定使用者會群組 id 模式位元
		t 固定位元

我們使用 chmod 700 *將 chaiyen 目錄下的所有檔案和目錄都改成只有 root 可以讀、寫與執行 rwx------。

```
[root@flash chaiyen]# chmod 700 *
-rwx------  1 root root 47978  8月 30 08:33 快照1.png
-rwx------  1 root root 41114  8月 30 08:55 快照2.png
-rwx------  1 root root 58315  8月 31 09:10 快照3.png
-rwx------  1 root root 26265  8月 31 09:20 快照5.png
-rwx------  1 root root 53834  8月 31 14:39 快照6.png
```

我們使用 chmod 740 webmin 將 webmin 設定成只有擁有者 root 可以讀取、寫入或執行，而設定 root 群組只能讀取。

```
[root@flash chaiyen]# chmod 740 webmin

drwxr-----  2 root root 4096  8月 22 23:12 webmin
```

我們使用 chmod 700 ~來設定使用者 root，只有其在家目錄 root 為可讀可寫可執行。

```
[root@flash chaiyen]# chmod  700 ~
drwx------  17 root root 4096  9月  2 10:58 root
```

我們使用 chmod u=rwx webmin 來將可讀、可寫和可執行的權限設定給 webmin 目錄擁有者 root。

```
[root@flash chaiyen]# chmod u=rwx webmin
drwxr-----    2 root     root         4096  8月 22 23:12 webmin
```

我們使用 chmod u-wx webmin 來將 root 檔案擁有者可寫和可執行的權限移除。

```
[root@flash chaiyen]# chmod u-wx webmin
dr--------    2 root     root         4096  8月 22 23:12 webmin
```

我們使用 chmod u+x webmin 來增加 root 檔案擁有者可執行檔案的權限。

```
[root@flash chaiyen]# chmod u+x webmin
dr-x------    2 root     root         4096  8月 22 23:12 webmin
```

我們可以使用 chgrp 指令來改變檔案或目錄的所屬群組。

語法：

指令：chgrp　檔案的群組　檔案或目錄

參數：

-c：只有當群組改變時才報告。

-R：遞迴的處理所有的檔案和子目錄。

-f：隱藏所有錯誤訊息。

我們使用 chgrp –c root webmin 將 webmin 目錄的檔案群組改成 root 群組。

```
[root@flash chaiyen]# chgrp -c root  webmin
changed group of `webmin' to root
drwxr-----    2 root     root         4096  8月 22 23:12 webmin
```

我們可以使用 chown 指令來更改檔案或目錄的擁有者。

語法：

指令：chown　參數　擁有者　檔案或目錄

參數：

-c：只有當群組改變時才報告。

-R：遞迴的處理所有的檔案和子目錄。

-f：隱藏所有錯誤訊息。

我們使用 chown –c webmin 來將 webmin 目錄的擁有者改成 root。

```
[root@flash chaiyen]# chown -c root webmin
changed ownership of `webmin' to root
```

當我們要建立檔案時，也會一起設定檔案的存取權限。我們使用權限遮罩將檔案的權限給限制住。。在建立檔案時預設的權限遮罩。權限遮照的預設值為 022。

語法：

指令：umask 權限遮罩

我們的檔案存取權限為預設的存取權限(777)減去權限遮罩。

檔案的存取權限=預設的存取權限 - 權限遮罩

我們使用 umask 指令來觀看目前權限遮罩的設定值。預設為 022。所以它設定的權限為 777-022=755(rwxrw-rw-)。當我們建立檔案或目錄時，它也會建立其存取權限為 755(rwxrw-rw-)。

```
[root@flash chaiyen]# umask
0022
```

我們使用 umask 013 來設定權限遮罩為 013。

```
[root@flash chaiyen]# umask 013
```

因為我們設定權限遮罩為 013，所以設定建立檔案目錄時，他的存取權限為 777-013=764(rwxrw-r--)。我們使用 mkdir best 來建立目錄。

```
[root@flash chaiyen]# mkdir best
```

這是我們所建立的 best 目錄。其權限為 rwxrw-r--。

```
drwxrw-r--    2 root     root         4096  9月  2 23:38 best
```

15-2-3 特殊權限 SUID、SGID、Sticky

SUID(Set-User-ID)為設定執行檔擁有該擁有者的存取系統資源權限。

我們可以使用 chmod u+s 指令來設定檔案擁有該擁有者存取系統資源的權限。

語法：

指令：chmod　4***　檔案或目錄

chmod　u+s　檔案或目錄

我們使用 ls –al 來顯示我們 power2 的檔案。

```
[root@flash chaiyen]# ls -al power2
-rwx--x---    1 root     root        14331  8月  9 11:36 power2
```

我們可以使用 chmod u+s 指令來設定 power2 檔案擁有 root 存取系統資源的權限。這時會變成 rws—x---。使用者的 x 變成小寫的 s 了。我們在這裏使用 chmod 4710 指令來設定 power2 的 SUID 權限。

```
[root@flash chaiyen]# chmod 4710 power2
[root@flash chaiyen]# ls -al power2
-rws--x---    1 root     root        14331  8月  9 11:36 power2
```

我們也可以使用 chmod u-s 將檔案的 SUID 的權限給移除。

```
[root@flash chaiyen]# chmod u-s power2
[root@flash chaiyen]# chmod u-x power2
[root@flash chaiyen]# ls -al power2
-rw---x---    1 root     root        14331  8月  9 11:36 power2
```

我們使用 chmod u+s power2 來設定 SUID，因為我們已經將擁有者的執行權力給刪除 chmod u –x，所以這時檔案的存取權會變成大寫的 S(rwS—x---)。

```
[root@flash chaiyen]# chmod u+s power2
[root@flash chaiyen]# ls -al power2
-rwS--x---    1 root     root        14331  8月  9 11:36 power2
```

SGID(Set-Group-ID)為設定執行檔擁有該擁有群組的存取系統資源權限。

我們可以使用 chmod g+s 指令來設定檔案擁有該擁有群組存取系統資源的權限

語法：

指令：chmod　2***　檔案或目錄

chmod　g+s　檔案或目錄

我們使用 ls –al 來顯示我們 power2 的檔案。我們可以使用 chmod g+s 指令來設定 power2 檔案擁有 root 群組的存取系統資源權限。我們在這裏使用 chmod 2751 指令來設定 power2 的 SGID 權限。

```
[root@flash chaiyen]# ls -al power2
-rw---x---    1 root     root        14331  8月  9 11:36 powe
[root@flash chaiyen]# chmod 2751 power2
```

這時會變成 rwxr-s--x。群組的檔案執行權 x 變成小寫的 s 了。

```
[root@flash chaiyen]# ls -al power2
-rwxr-s--x    1 root     root        14331  8月  9 11:36 power2
```

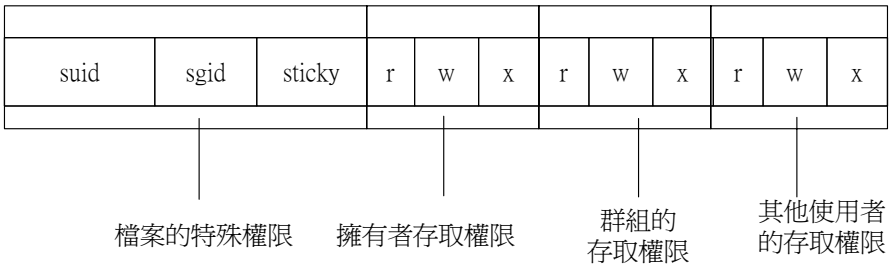
我們也可以使用 chmod g-s 將檔案的 SGID 的權限給移除。

```
[root@flash chaiyen]# chmod g-s power2
[root@flash chaiyen]# chmod g-x power2
[root@flash chaiyen]# ls -al power2
-rwxr----x    1 root     root        14331  8月  9 11:36 power2
```

我們使用 chmod g+s powre2 來設定 SGID，因為我們已經將群組的執行權力給刪除 chmod g –x，所以這時檔案的群組執行權會變成大寫的 S(rwS—x---)。

```
[root@flash chaiyen]# chmod g+s power2
[root@flash chaiyen]# ls -al power2
-rwxr-S--x    1 root     root        14331  8月  9 11:36 power2
```

這是我們檔案的存取權限。



Sticky 固定位元為特殊的檔案權限，它讓檔案只有是它的擁有者或超級使用者才能去搬動這個檔案或刪除這個檔案。我們可以使用 chmod 1***指令，就可以加入這個 Sticky 固定位元。或者我們也可以使用 chmod +t 檔案或目錄來加入這個 Sticky 固定位元。

語法：

指令：chmod 1*** 檔案或目錄

chmod +t 檔案或目錄

我們使用 chmod 1775 power2 指令來加入 Sticky 固定位元。

```
[root@flash chaiyen]# chmod 1775 power2
[root@flash chaiyen]# ls -al
```

power2 的檔案權限變成 rwxrwxr-t，這表示只有 root 超級使用者可以移動 power2 這個檔案，而 power2 也將固定在這個目錄中。

```
-rwxrwxr-t    1 root     root        14331  8月  9 11:36 power2
```

我們使用 chmod 750 power2 來刪除 Sticky 固定位元和其他使用者執行檔案的權限。我們然後使用 chmod +t power2 來將固定位元加到 power2 中，所以它的權限變成(rwxr-x—T)這時就變成大 T。

```
[root@flash chaiyen]# chmod 750 power2
[root@flash chaiyen]# chmod +t power2
[root@flash chaiyen]# ls -al power2
-rwxr-x--T    1 root     root        14331  8月  9 11:36 power2
```

15-3 檔案處理

我們可以使用 vi 編輯器編輯建立檔案, 我們也可以使用 touch 指令來新建立檔案。

我們可以使用 touch 指令來建立一個新檔。

語法：

指令：touch 參數 檔名

參數：

-c：若檔案真的不存在，也不要新增此檔

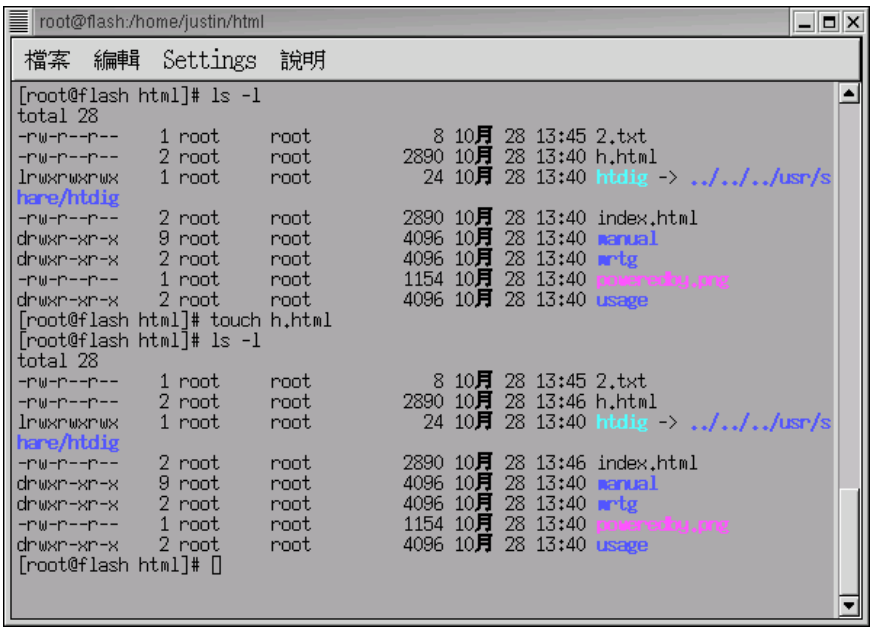
-f：強迫做 touch 的動作

-m：改變檔案修改的時間

我們使用 touch bb.c 指令來建立 bb.c 的檔案。

```
[root@flash chaiyen]# touch bb.c
```

我們使用 touch h.html 將檔案 h.html 的時間由 13:40 改成現在的時間 13:46。



15-3-1 觀看檔案的內容

cat 指令可以將多個檔案結合，並將所有內容輸出到標準輸出設備，若不指定任何檔案名稱，則 cat 指令會從鍵盤讀取 Key-in 的訊息，然後再輸出到螢幕。

語法：

指令：cat 參數 檔案串列

參數：

-b：在每一個空白列開頭編上編號。

-E：在每一行最後顯示$。

-n：把每一行都編號。

-s：當空白行數超過一行時，則以一行空白表示。

-T：顯示 tab 字元為^I。

我們使用 cat -b 在 g.c 檔案每一個空白列開頭編上編號。

```
[root@flash chaiyen]# cat -b g.c
     1  iiiii
```

我們使用 cat –E 在 g.c 檔案每一行最後顯示$。

```
[root@flash chaiyen]# cat -E g.c
iiiii$
```

我們使用 cat –n 在 g.c 檔案把每一行前都編號。

```
[root@flash chaiyen]# cat -n g.c
     1  we are good.
     2  we are the best.
     3  we will be the fisrt one.
     4  I love my country.
     5  I will protect my country.
```

我們使用 cat –s 在 g.c 檔案空白行數超過一行時，則以一行空白表示。

```
[root@flash chaiyen]# cat -s g.c
we are good.
we are the best.

we will be the fisrt one.
I love my country.
I will protect my country.
```

我們使用 cat –T 在 po.c 檔案中顯示 tab 字元為^I。

```
[root@flash chaiyen]# cat po.c
I am a good man.
        yes.
[root@flash chaiyen]# cat -T po.c
I am a good man.
^Iyes.
```

我們使用 more 指令來一頁一頁的顯示檔案內容。

語法：

指令：more 參數 檔案串列

參數

-d：在畫面提示”按下空白鍵來繼續，按下 q 來離開”。

-f：計算實際行數。

-l：取消遇到^L 會暫停的功能。

-s：合併連續空白的行數為一行。

-u：隱藏文字的底線。

-<行數>：指定每次顯示的行數。

+/<字串>：搜尋指定的字串。

+<行數>：從指定行數開始顯示。

我們使用 more +50 love.txt 來指定從第 50 行開始顯示內容。

```
[root@flash chaiyen]# more +50 love.txt
最不會調情的人：     天秤座男生、魔羯座女生
最愛好和平的人：     金牛座男生、巨蟹座女生
最愛引起爭端的人：   獅子座男生、牡羊座女生
```

我們可以一次以一個螢幕的方式來顯示檔案資料。

語法：

指令：less 參數 檔案串列

參數：

-N：顯示列數編號。

-o 檔案：將 less 指令讀入的資料輸出成檔案儲存。

-p 範本：從指定的範本開始執行。

-c：重心繪製整個畫面。

-m：顯示百分比模式。

-n：忽略列數編號。

-N：顯示行數編號。

我們使用 less –N g.c 來顯示檔案 g.c 的內容並且顯示編號。

```
[root@flash chaiyen]# less -N g.c
       1 we are good.
       2 we are the best.
       3
       4
       5
       6 we will be the fisrt one.
       7 I love my country.
       8 I will protect my country.
```

我們可以使用 head 指令輸出檔案內容最前面十行的部份。

語法：

指令：head 參數 檔案串列

參數：

-c 顯示數目：顯示以位元組來計算。

-n<顯示列數>：我們可以指定要顯示的列數。

我們使用 head love.txt 就可以顯示 love.txt 檔案前十行。

```
[root@flash chaiyen]# head love.txt
智力測驗平均最高者：水瓶座男生、天蠍座女生
智力測驗平均最低者：雙魚座男生、魔羯座女生
機靈反應能力最佳者：雙子座男生、處女座女生
機靈反應能力最差者：金牛座男生、金牛座女生
最重視倫理道德的人：巨蟹座男生、魔羯座女生
最漠視倫理道德的人：射手座男生、獅子座女生
最懂得羅曼蒂克的人：雙子座男生、雙魚座女生
最不懂羅曼蒂克的人：處女座男生、魔羯座女生
用情最爲專一者：    巨蟹座男生、魔羯座女生
用情最爲善變者：    雙子座男生、水瓶座女生
```

我們使用 head –n 2 love.txt 來顯示檔案前兩行。

```
[root@flash chaiyen]# head -n 2 love.txt
智力測驗平均最高者：水瓶座男生、天蠍座女生
智力測驗平均最低者：雙魚座男生、魔羯座女生
```

我們使用 tail 指令來顯示檔案最後十行的內容。

語法：

指令：tail 參數 檔案串列

參數：

-c 顯示數目。

-f：當讀取到檔案最末端，不斷的反覆嘗試讀取更多資料。

-n 行數：顯示最末端的行數

我們使用 tail –n 3 love.txt 來顯示最後三行的內容。

```
[root@flash chaiyen]# tail -n 3 love.txt
最不會調情的人：     天秤座男生、魔羯座女生
最愛好和平的人：     金牛座男生、巨蟹座女生
最愛引起爭端的人：   獅子座男生、牡羊座女生
```

15-3-2 拷背、移除和移動檔案

我們可以使用 cp 來拷背檔案。

語法：

指令：cp 參數 來源檔案 目地檔案

參數：

-b：當移除或覆蓋目地檔之前先備份。

-d：當複製符號連結時，把目錄或目錄也建立符號連結，並指向來源檔案或目錄。

-f：強制複製檔案或目錄。

-i：當覆蓋檔案前，先詢問使用者。

-l：對來源檔案建立硬連結(hard link)。

-p：保留來源目錄或檔案的屬性。

-r：遞迴複製檔案。

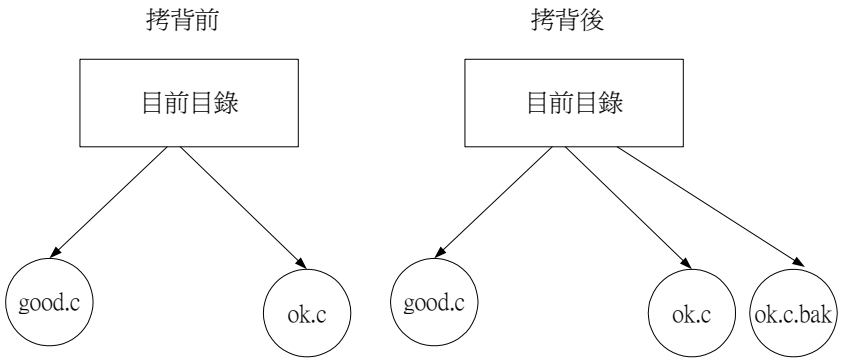
-s：對來源檔案或目錄建立符號連結。

-u：當來源檔案的更動時間比目地檔案更新時才複製檔案。

-x：不處理其它分割區的檔案。

我們使用 cp ok.c ok.c.bak 來覆製 ok.c.的檔案到 ok.c.bak。在我們拷背後就會增加 ok.c.bak 的檔案到我們目錄之下了。

```
[root@flash chaiyen]# cp ok.c ok.c.bak
```



我們使用 mv 指令來更改檔案名稱，或移動檔案到指定目錄。

語法

指令：mv 參數 來源檔案 目地檔案

mv 參數 檔案串列 目錄

參數：

-b：當需要覆寫檔案時，則先行備份。

-f：強制覆寫檔案。

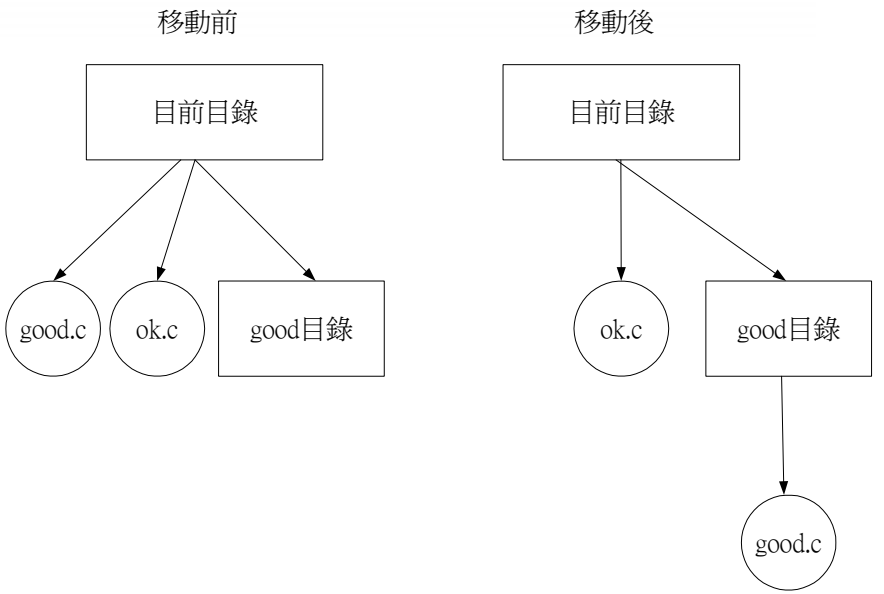
-i：覆寫前先詢問使用者。

-S 字尾：指定備份檔附加的字尾。

-u：只有當來源檔比目地檔比目地檔較新才覆寫目地檔。

我們使用 mv 指令將 good.c 的檔案移動到 good 目錄下。

```
[root@flash chaiyen]# mv good.c good/good.c
```



我們使用 rm 指令來移除指定的檔案。

語法

指令：rm 參數 檔案串列

參數：

-d：移除 Unlink 指定目錄，並將其硬連結數刪除。

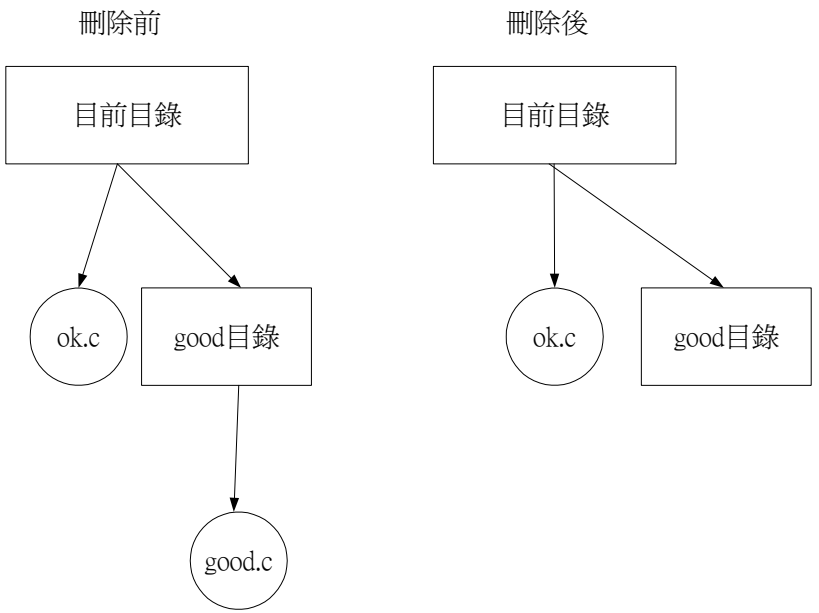
-f：強制刪除目錄或檔案。

-i：在刪除檔案或目錄前，先詢問使用者。

-r：刪除檔案時使用遞迴處理。

我們使用 rm good/good.c 來刪除在 good 目錄下的 good.c 檔案。

```
[root@flash chaiyen]# rm good/good.c
rm: remove `good/good.c'? y
```



我們使用 wc 來計算檔案大小、檔案的行數或字數。

語法：

指令：wc 參數 檔案串列

參數：

-c：計算檔案位元組 Byte 數目。

-m：計算檔案字元數目。

-l：計算檔案行數。

-L：計算檔案最長行數的長度。

-w：計算檔案文字的數目。

我們使用 wc –c love.txt 來計算檔案位元組 Byte 數目。

```
[root@flash chaiyen]# wc -c love.txt
   2335 love.txt
```

我們使用 wc –l love.txt 來計算檔案 love.txt 的行數。

```
[root@flash chaiyen]# wc -l love.txt
     51 love.txt
```

當目錄為空的時後，我們可以使用 rmdir 來移除目錄。

語法

指令：rmdir 參數 目錄

參數：

-p：移除指定目錄和指定目錄路徑中組成的目錄。

-verbose：顯示指令執行的過程。

--help：顯示說明。

```
[root@flash chaiyen]# rmdir -p  good/best
```

我們使用 rmdir –help 來顯示 rmdir 指定的說明。

```
[root@aasir chaiyen]# rmdir --help
用法: rmdir [選項]... DIRECTORY...
Remove the DIRECTORY(ies), if they are empty.
```

我們使用 mkdir 來建立目錄。

語法：

指令：mkdir 參數 目錄

參數：

-m 目錄權限：按照目錄權限來設定目錄。

-p：當建立目錄時，若上一層目錄還未建立，則一起建立。

--help：顯示線上說明。

--verbose 顯示指令執行的過程。

我們使用 mkdir –p good/best 來建立 best 目錄和其上一層目錄 good。

```
[root@flash chaiyen]# mkdir -p good/best
```

15-3-3 添加到檔案

我們可以使用 cat 檔案串列 >> 目地檔案，來將檔案串列的檔案加到目地檔案中。

語法

指令：cat 檔案串列 >> 目地檔案

我們可以使用 cat 檔案串列 >> 目地檔案，來將檔案串列的檔案 g.c 和 love.txt 加到目地檔案 best.c 中。我們使用 more best.c 就可以看到添加的結果。

```
[root@flash chaiyen]# cat g.c love.txt >> best.c
[root@flash chaiyen]# more best.c
we are good.
we are the best.



we will be the fisrt one.
I love my country.
I will protect my country.
智力測驗平均最高者：水瓶座男生、天蠍座女生
```

15-3-4 組合檔案

cat 指令可以將多個檔案結合，並將所有內容輸出到標準輸出設備，若不指定任何檔案名稱，則 cat 指令會從鍵盤讀取 Key-in 的訊息，然後再輸出到螢幕。

語法：

指令：cat 參數 檔案串列

參數：

-b：在每一個空白列開頭編上編號。

-E：在每一行最後顯示$。

-n：把每一行都編號。

-s：當空白行數超過一行時，則以一行空白表示。

-T：顯示 tab 字元為^I。

```
[root@flash chaiyen]# more good.c
I am the best.

[root@flash chaiyen]# more ok.c
I am ok.
```

我們使用 cat 指令來合併顯示 good.c 和 ok.c 的檔案。

```
[root@flash chaiyen]# cat good.c ok.c
I am the best.

I am ok.
```

15-3-5 比較檔案、移除重複的行數

我們可以使用 diff 指令來比較指定的檔案間的相異之處，假如我們指定比較目錄，則只會比較目錄中相同檔名的檔案。

語法

指令：diff 參數 檔案 1 檔案 2

參數：

-a：把他們當作文字檔一行一行的比較檔案。

-b：忽略空白字元。

-B：忽略空白行數。

-c：列出所有行數，並標出不同之處。

-d：改變演算法式來找尋更小的改變。

-f：按照原先檔案順序，顯示檔案不同之處。

-H：忽略小地方來加快檔案的比較。

-i：忽略大小寫的不同。

-l：結果交由 pr 分頁。

-q：僅顯示其差異性。

-r：以遞迴方式比較目錄及其子目錄。

-s：當二個檔案相同時提出報告。

我們使用 diff good.c ok.c 來比較 good.c 和 ok.c 的相異之處。

```
[root@flash chaiyen]# diff good.c ok.c
1,2c1
< I am the best.
<
---
> I am ok.
```

我們使用 diff –q good.c ok.c 來顯示 good.c 和 ok.c 的差異性。

```
[root@flash chaiyen]# diff -q good.c ok.c
Files good.c and ok.c differ
```

我們使用 uniq 指令從輸入檔來檢查檔案中重覆出現的行列。

語法

指令：uniq 參數 輸入檔 輸出檔

參數：

-c：每一行的前面顯示其重覆出現的次數。

-d：只有顯示重覆的行數。

-f 欄位：忽略比較重覆的欄位。

-s 字元位置：忽略比較重覆的字元。

-u：顯示只有出現過一次的行列。

-w 字元數：比較不得多於字元數。

--help：顯示說明。

我們使用 cat 來顯示 good.c 的檔案內容，然後我們使用 uniq –d good.c 來顯示檔案中重覆出現的行列。

```
[root@flash chaiyen]# cat good.c
I am the best.
I am the best.
I am the best.
I am fine.
are you ok?
I am fine.
[root@flash chaiyen]# uniq -d good.c
I am the best.
```

我們使用 uniq –c good.c 來在每一行的前面顯示其重覆出現的次數。

```
[root@flash chaiyen]# uniq -c good.c
      3 I am the best.
      1 I am fine.
      1 are you ok?
      1 I am fine.
```

15-3-6 列印檔案

lpr 使用 spooling 的方式來列印檔案。Spooling 就是將所要列印的檔案暫時放到磁碟中存放，等要用時再載入到印表機。

語法

指令：lpr 參數 檔案串列

參數：

-#列印份數：設定列印的份數。

-P 印表機：指定要列印的印表機。

-T 工作標題：設定列印文件的工作標題。

-m 使用者：列印完成後 mail 通知使用者。

-p：使用 pr 指令將列印檔案格式化。

我們使用 lpr -#5 good.c 來列印 5 份 good.c 的檔案。

```
[root@flash chaiyen]# lpr -#5 good.c
```

我們使用 lpq 指令可以顯示列印的作業情況。

語法：

指令：lpq 參數

參數：

-l：顯示列印的資訊。

-P 印表機：顯示指定印表機的工作。

我們可以使用 lprm 指令來移除在 spooling 中等待的列印工作。

語法

指令：lprm 參數 工作串列 使用者

參數：

-a：移除所有在列印序列中的檔案。

-P 印表機：指定印表機。

我們使用 lprm 移除最後一個列印的工作。

```
[root@flash chaiyen]# lprm
```

我們使用 lprm –a all 移除所有在列印序列中的檔案。

```
[root@flash chaiyen]# lprm -a all
```

15-4 進階檔案處理

15-4-1 壓縮檔案、解壓縮檔案與打包檔案

檔案壓縮工具

壓縮檔案	解壓縮檔案	副檔名
compress	uncompress	.Z
pack	unpack	.z
Tar+gzip	Tar+gzip	.tgz
Gzip	Gzip – d(或 gunzip)	.gz
Zip	Unzip	.zip
Bzip2	Bzip2 – d	.bz2

我們使用 gzip 指令來壓縮檔案。

語法

指令：gzip 參數 檔案串列

參數：

-a：Ascii 文字模式。轉換每一行的最後，使用本地端的協定。

-c：壓縮後檔案輸出至標準輸出設備，而不更動原始檔案。

-d：解壓縮檔案。

-f：強制執行壓縮。

-h：顯示說明。

-l：顯示壓縮檔欄位訊息。

-L：顯示 gzip 版權。

-n：當壓縮檔案時，不儲存原來的檔案名稱及時間。

-N：壓縮檔案時，儲存原來的檔案名稱及時間。

-q：不顯示警告內容。

-r：遞迴式處理目錄結構。

-t：測試壓縮檔。

-v：顯示指令執行過程。

-壓縮率：介於 1 到 9，數值越大壓縮率越高。預設值為 6。

--best：最佳壓縮。

--fast：快速壓縮。

這是 start.txt 的檔案，它的大小為 2335 個位元組。

```
[root@flash chaiyen]# ls -al start.txt
-rwxr--r--    1 root     root         2335  9月  4 22:15 start.txt
```

我們使用 gzip 來壓縮 start.txt，則它的壓縮檔會變成 start.txt.gz。

```
[root@flash chaiyen]# gzip --best start.txt
```

這時壓縮檔變成 start.txt.gz，而其檔案大小為 783 個位元組，約為原來檔案大小的三分之一。

```
[root@flash chaiyen]# ls -al start.txt.gz
-rwxr--r--    1 root     root          783  9月  4 22:15 start.txt.gz
```

我們使用 gzip –d start.txt.gz 來解壓縮檔，來原來的檔名 start.txt。

```
[root@flash chaiyen]# gzip -d start.txt.gz
```

我們使用 gunzip 指令來解壓縮檔案。

語法

指令：gunzip 參數　檔案串列

參數：

-a：Ascii 文字模式。轉換每一行的最後，使用本地端的協定。

-c：壓縮後檔案輸出至標準輸出設備，而不更動原始檔案。

-d：解壓縮檔案。

-f：強制執行壓縮。

-h：顯示說明。

-l：顯示壓縮檔欄位訊息。

-L：顯示版權。

-n：當壓縮檔案時，不儲存原來的檔案名稱及時間。

-N：壓縮檔案時，儲存原來的檔案名稱及時間。

-q：不顯示警告內容。

-r：遞迴式處理目錄結構。

-t：測試壓縮檔。

-v：顯示指令執行過程。

我們使用 gunzip start.txt.gz 來解壓縮 start.txt.gz 檔案。

```
[root@flash chaiyen]# ls -al start.txt
-rwxr--r--    1 root     root         2335  9月  4 22:15 start.txt
[root@flash chaiyen]# gzip --best start.txt
[root@flash chaiyen]# ls -al start.txt.gz
-rwxr--r--    1 root     root          783  9月  4 22:15 start.txt.gz
[root@flash chaiyen]# gunzip start.txt.gz
[root@flash chaiyen]# ls -al start.txt
-rwxr--r--    1 root     root         2335  9月  4 22:15 start.txt
```

我們使用 gunzip –l *.gz 來顯示所有壓縮檔的壓縮資訊。

```
[root@flash chaiyen]# gunzip -l *.gz
         compressed        uncompressed  ratio uncompressed_name
                783                2335  67.7% start.txt
```

語法：

指令：zip 參數 壓縮檔 檔案串列

參數：

-a：將檔案轉換成 ASCII 的格式。

-b 目錄：指定暫存檔案的目錄。

-c：每一個檔案都加上一個註解。

-d：從壓縮檔內刪除指定的檔案。

-D：壓縮檔內不建立目錄名稱。

-f：更新現有的檔案，假如有些檔案原本不存在，可以將它加入壓縮檔中。

-F：試著去修護已損壞的壓縮檔。

-g：將檔案壓縮後附加到現有的壓縮檔後。

-h：使用說明

-i 檔案：壓縮指定的檔案。

-j：移除壓縮檔前不需的資料。

-k：使用符合 MS-DOS 格式的檔案名稱。

-m：將檔案壓縮並加入壓縮檔後，刪除原始檔案。

-n 字尾字串：不壓縮指定的字尾字串。

-o：將壓縮檔的異動時間設為和該檔案相同。

-P 密碼：將壓縮檔加密。

-q：不顯示指令執行過程。

-r：遞迴處理壓縮過程。

-T：檢查備份檔內的每個檔案是否正確。

-u：將較新的檔案壓縮後，來更換較舊的檔案。

-x 檔案：明確的包含指定的檔案。

我們使用 zip 來壓縮/hom/chaiyen/第九章底下的所有目錄及檔案，壓縮的名子為 book.zip。

```
[root@flash chaiyen]# zip -r book /home/chaiyen/第九章
  adding: home/chaiyen/第九章/ (stored 0%)
  adding: home/chaiyen/第九章/第九章　討論區PHORUM7.3.doc (deflated 13%)
```

這是我們壓縮的檔案 book.zip。

```
[root@flash chaiyen]# ls -al book.zip
-rw-r--r--    1 root     root      2613804  9月  5 14:42 book.zip
```

我們使用 zip ok ok.c 將 ok.c 的檔案壓縮成 ok.zip。

```
[root@flash chaiyen]# zip ok ok.c
  adding: ok.c (stored 0%)
```

這是我們壓縮成 ok.zip 的情況。

```
[root@flash chaiyen]# ls -al ok.zip
-rw-r--r--    1 root     root          149  9月  5 14:59 ok.zip
```

我們可以使用 zip　pic　*.tif 把所有*.tif 的圖檔壓縮成 pic.zip 的壓縮檔。

```
[root@flash chaiyen]# zip pic *.tif
  adding: 002.tif (deflated 98%)
  adding: 003.tif (deflated 99%)
  adding: 005.tif (deflated 99%)
[root@flash chaiyen]# ls -al pic.zip
-rw-r--r--    1 root     root        31670  9月  5 15:04 pic.zip
```

我們可以使用 unzip 來解壓縮*.zip 檔。一般這些*.zip 檔都是來自微軟的作業系統。

語法：

指令：unzip　參數　壓縮檔　檔案串列

參數：

-a：轉換文字檔。

-b：把所有檔案當作是二進位檔。

-C：區分壓縮檔案名稱的大小寫。

-c：顯示解壓縮的檔案。

-f：更新現有的檔案。

-j：不在重新建立原來壓縮檔案的目錄。

-l：顯示壓縮檔中，所有的檔案。

-L：將壓縮檔中的所有檔案轉成小寫。

-n：不覆寫現存的檔案。

-o：直接覆寫現存的檔案，而不詢問使用者。

-P 密碼：使用解密來直接解密壓縮的檔案。

-t：測試壓縮檔是否正確。

-u：更新和建立新的檔案。

-v：顯示執行時的資訊。

-Z：顯示壓縮檔資訊。

我們可以使用 unzip ok.zip 來解壓縮我們的 zip 檔。

```
[root@flash chaiyen]# unzip ok.zip
Archive:  ok.zip
replace ok.c? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
 extracting: ok.c
```

我們使用 zip –P 353766 g g.c 將 g.c 檔案壓縮成 g.zip，再加上鎖定的密碼檔 353766。我們使用 unzip 來解壓縮 g.zip 檔時需輸入密碼 353766。

```
[root@flash chaiyen]# zip -P 353766 g g.c
  adding: g.c (deflated 26%)
[root@flash chaiyen]# ls -al g.zip
-rw-r--r--    1 root     root          244  9月  5 15:41 g.zip
[root@flash chaiyen]# unzip g.zip
Archive:  g.zip
[g.zip] g.c password:
replace g.c? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: g.c
```

我們也可以使用 unzip –P 353766 g.zip 來解壓縮 g.zip 檔。

```
[root@flash chaiyen]# unzip -P 353766 g.zip
Archive:  g.zip
```

我們使用 unzip –Z g.zip 來看 g.zip 壓縮檔的詳細資訊。

```
[root@flash chaiyen]# unzip -Z g.zip
Archive:  g.zip   244 bytes   1 file
-rwxrwxrwx  2.3 unx      105 TX defN  3-Sep-02 16:00 g.c
1 file, 105 bytes uncompressed, 78 bytes compressed:  25.7%
```

我們可以使用 zipinfo 來顯示壓縮檔的資訊。

語法

指令：zipinfo 參數　壓縮檔案

參數：

-1：每一行只列出檔名。

-2：每一行只列出檔名、標頭、結尾和壓縮說明。

-s：列出壓縮檔的資訊，就像是"ls –l"較短的格式。

-m：列出壓縮檔的資訊，就像是"ls –l"一般的格式。

-l：列出壓縮檔的資訊，就像是"ls –l"一般的格式。

-v：顯示壓縮檔較長的資訊。

我們使用 zipinfo –v g.zip 來顯示 g.zip 的詳細資訊。

```
[root@flash chaiyen]# zipinfo -v g.zip
Archive:  g.zip   244 bytes   1 file

End-of-central-directory record:
```

我們使用 bzip2 來壓縮檔案。它的副檔名是.bz2。它壓縮檔案是使用 Huffman 來編碼。

語法：

指令：bzip2 參數　檔案串列

參數：

-c：顯示壓縮或解壓縮的標準輸出。

-d：執行解壓縮。

-z：強制壓縮。

-f：解壓縮時，強制覆寫檔案。

-t：測試壓縮。

-k：保留原被壓縮檔。

-s：壓縮時減少記憶體的使用。

-L：壓縮程式的版權。

我們使用 bzip2 –v find.c 來將 find.c 檔案壓縮成 find.c.bz2。

```
[root@flash chaiyen]# bzip2 -v find.c
  find.c:   0.281:1, 28.500 bits/byte, -256.25% saved, 16 in, 57 out.
[root@flash chaiyen]# ls -al find.c.bz2
-rw-r--r--    1 root     root           57  9月  4 15:13 find.c.bz2
```

我們使用 bzip2 –d find.c.bz2 來解壓縮 find.c.bz2 到 find.c 檔。

```
[root@flash chaiyen]# bzip2 -d find.c.bz2
```

我們使用 bzip2 –k conb.c，就可以壓縮檔案成 conb.c.bz2 壓縮檔，並且保留 conb.c 原始檔案。

```
[root@flash chaiyen]# bzip2 -k conb.c
```

我們使用 bunzip2 來解壓縮*.bz2 的檔案。

語法：

指令：bunzip2 參數　檔案串列

參數：

-c：顯示壓縮或解壓縮的標準輸出。

-d：執行解壓縮。

-f：解壓縮時，強制覆寫檔案。

-k：解壓縮後保留原壓縮檔。

我們可以使用 bunzip2 –k bb.c.bz2 解壓縮檔 bb.c.bz2 成 bb.c，我們並且可以保留原來的壓縮檔 bb.c.bz2。

```
[root@flash chaiyen]# bunzip2 -k bb.c.bz2
[root@flash chaiyen]# ls
002.tif    book.zip     good       pic.zip       快照3.png  第九章     第十一章
003.tif    conb.c       g.zip      proftpd       快照5.png  第二章     第十章
005.tif    conb.c.bz2   love.txt   start.txt.gz  快照6.png  第五章     第四章
bb.c       feeling.eml  ok.c.bak   webmin        第一章     第五章pic
bb.c.bz2   find.c.bz2   ok.c.bz2   快照1.png     第七章     第八章
best.c     g.c          ok.zip     快照2.png     第三章     第六章
```

zcat 把解壓縮的檔案輸出到標準輸出設備來顯示。

語法：

指令：zcat 參數 檔案串列

參數：

-f：強制解壓縮檔案。

我們使用 zcat 指令將 best.c.gz 和 conb.c.gz 解壓縮，並將它們輸出到 goodman.gz 的檔案中。

```
[root@flash chaiyen]# zcat best.c.gz conb.c.gz > goodman.gz
[root@flash chaiyen]# ls
002.tif    best.c.gz    g.c.gz     ok.c.bz2      快照1.png
003.tif    book.zip     good       ok.zip        快照2.png
005.tif    conb.c.bz2   goodman.gz pic.zip       快照3.png
bb.c       conb.c.gz    g.zip      proftpd       快照5.png
bb.c.bz2   feeling.eml  love.txt   start.txt.gz  快照6.png
best.c~    find.c.bz2   ok.c.bak   webmin        第一章
```

我們可以使用 cmpress 指令來壓縮檔案。經過壓縮後其附檔名會有.Z。

語法：

指令：compress 參數 檔案或目錄

參數：

我們可以使用 cmpress 指令來壓縮檔案。經過壓縮後其附檔名會有.Z。圖檔使用 compress 可以壓縮成原來的三十分之一。

```
[root@flash chaiyen]# ls -al 003.tif
-rwx------    1 root     root       622250  8月 30 07:49 003.tif
[root@flash chaiyen]# compress 003.tif
[root@flash chaiyen]# ls -al 003.tif.Z
-rwx------    1 root     root        19269  8月 30 07:49 003.tif.Z
```

我們可以使用 comress –d 003.tif.Z 來解壓縮檔案。

```
[root@flash chaiyen]# compress -d 003.tif.Z
[root@flash chaiyen]# ls -al 003.tif
-rwx------    1 root     root       622250  8月 30 07:49 003.tif
```

我們可以使用 tar 指令來新建打包成備份檔或從備份檔取出檔案。

語法：

指令：tar [key][options][name]

參數：

-c ：建立新的備份檔

--delete： 刪除 tar 中的檔案

-r ：將檔案附加在備份檔後面

-f ：指定檔案名稱

-t ：列出檔名

-u ：將把比備份檔中更新的檔案加入到備份檔中

-x ：從備份檔中取出檔案

-z ：用 gzip 指令檔案來處理備份檔。

-b ：指定區塊的數目

-v：處理檔案時列出檔名

-A：將備份檔加到已存在的備份檔後。

-d：比較備份檔和檔案系統的不同

-k：保留舊的檔案。

-i：忽略備份檔中 0 個區塊。

-O：抽取檔案然後標準輸出。

-p：抽許所有保護的資訊。

-S：有效的處理稀疏檔。

--group=群組名稱：將加入備份檔案中的所屬群組，設成指定群組。

-w：每一步都會詢問使用者。

我們使用 tar –zxvf 來解開 PHP-NUKE 的包裝檔案。

```
[root@flash chaiyen]# tar -zxvf PHP-Nuke-5.2.tar.gz
```

我們使用 tar –cvf 來將所有*.png 的檔案打包成 pic.tar 的備份檔。因為把六個*.png 的檔案打包成一個 pic.tar 的備份檔，所以我們稱 tar 指令為打包。

```
[root@flash chaiyen]# tar -cvf pic.tar *.png
快照1.png
快照2.png
快照3.png
快照5.png
快照6.png
```

這是我們可以觀看我們使用 tar 打包成備份檔的情況。

```
[root@flash chaiyen]# ls -al *.tar
-rw-r--r--    1 root     root      16013308  9月  5 19:51 doc.tar
-rw-r--r--    1 root     root        235520  9月  5 23:27 pic.tar
[root@flash chaiyen]# ls -al *.png
-rwx------    1 root     root         47978  8月 30 08:33 快照1.png
-rwx------    1 root     root         41114  8月 30 08:55 快照2.png
-rwx------    1 root     root         58315  8月 31 09:10 快照3.png
-rwx------    1 root     root         26265  8月 31 09:20 快照5.png
-rwx------    1 root     root         53834  8月 31 14:39 快照6.png
```

15-4-2 排列檔案

我們可以使用 sort 加上關鍵字來排列檔案。排列按照大小順序可以分為遞增排列或遞減排列。{1、3、5、7、9}為遞增排列。{9、7、5、3、1}為遞減排列。排列是以鍵值的大小來作排列。

語法：

指令：sort 參數 檔案串列

參數：

-b：忽略主要的空白。

-d：考慮只有空白或字母字元。

-f：將小寫字元當作是大寫字元。

-g：根據一般的數值來比較。

-i：考慮只可列印的字元。

-n：比較根據字串的數值。

+欄位 n：指定欄位來當作比較排列的關鍵值。

-r：反轉比較的結果。

-c：檢查輸入是否已排列。

-m：合併已排列的檔案。

-o 檔案：將結果寫入檔案，而不是標準輸出。

-s：穩定排列。

我們使用 cat 來觀看我們 student.c 的內容。

```
[root@flash home]# cat student.c
david Sarwar    david@msa.hinet.net     61.218.29.2
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6
```

我們使用 sort 來排列我們 student.c 檔案的內容。

```
[root@flash home]# sort student.c

david Sarwar    david@msa.hinet.net     61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
```

我們使用 sort +1 來按照第二欄位來排列 student.c 的檔案。

```
[root@flash home]# sort +1 student.c
Michal Jackson Michal@yahoo.com.tw     61.218.29.5
Nabeel Johnsen Nabeel@sina.com.tw      61.218.29.3
Hassan Kendal  Hassan@cml.hinet.net    61.218.29.0
david Sarwar   david@msa.hinet.net     61.218.29.2
goddess tsen   goddess@ddm.org.tw      61.218.29.6
Mary  wu       Mary@yahoo.com.tw       61.218.29.2
```

15-4-3 尋找檔案

有時後我們需要找尋是否特定的檔案或指令存在於我們的檔案系統結構。我們可以使用 find、whereis 和 which 來搜尋檔案或指令。

我們可以使用 find 指令來搜尋符合運算式的目錄串列。Find 指令使用遞迴的方式來搜尋檔案目錄。

語法：

指令：find 目錄串列 運算式

參數：

-exec 指令：尋找符合指令，如果傳回 0 則指令存在。

-inum 節點 inode 編號：搜尋節點 inode 編號的目錄或檔案。

-links 連結數目：搜尋指定連結數目的檔案。

-name 範本：搜尋指定範本的指令或檔案。

-newer 檔案：搜尋比指定檔案更新的指令或檔案。

-ok 指令：和-exec 參數一樣，但使用前會先詢問使用者。

-perm 權限數值：搜尋指令權限數值的檔案或指令。

-print ：顯示搜尋的結果。

-size +區塊：搜尋指定區塊大小的檔案。

-user 使用者：尋找使用者所擁有的檔案。

-amin 分鐘：搜尋在指定分鐘內被存取的檔案。

-anewr 檔案：搜尋最近被存取的檔案。

-atime 小時：搜尋幾小時內被存取的檔案。

我們使用 find /home –user chaiyen 來找尋 chaiyen 使用者在/home 目錄下所擁有的檔案。

```
[root@flash chaiyen]# find /home -user chaiyen
/home/chaiyen
/home/chaiyen/.kde
/home/chaiyen/.kde/Autostart
/home/chaiyen/.kde/Autostart/Autorun.desktop
/home/chaiyen/.kde/Autostart/.directory
/home/chaiyen/.gtkrc
/home/chaiyen/.bash_logout
/home/chaiyen/.bash_profile
/home/chaiyen/.bashrc
/home/chaiyen/.emacs
```

找尋/home/chaiyen 目錄下權限為 777 的檔案或目錄。

```
[root@flash chaiyen]# find /home/chaiyen -perm 777
/home/chaiyen
/home/chaiyen/g.c.gz
/home/chaiyen/proftpd/proftpd-1.2.5/contrib/README.linux-privs
/home/chaiyen/proftpd/proftpd-1.2.5/contrib/README.mod_sql
```

我們使用 find /home/chaiyen –atime -2 搜尋 2 時內在/home/chaiyen 目錄中被存取的檔案。

```
[root@flash chaiyen]# find /home/chaiyen -atime  -2
/home/chaiyen
/home/chaiyen/.kde
/home/chaiyen/.kde/Autostart
/home/chaiyen/.bash_profile
/home/chaiyen/.bashrc
/home/chaiyen/good.c
/home/chaiyen/.xvpics
/home/chaiyen/003.tif
/home/chaiyen/doc.tar
/home/chaiyen/005.tif
```

我們可以使用 whereis 指令來找尋系統是否有我們指定的指令或檔案，以及它所在的目錄及地方。我們可以使用 whereis 指令來找尋檔案所在的位置，和更新搜尋的路徑。我們也可以使用 wheris 指令來搜尋到我們所要指令的說明。

語法：

指令：whereis 參數 檔案串列

參數：

-b：只搜尋執行檔。

-m：同時搜尋指令的說明檔。

-s：僅搜尋原始檔。

-u：搜尋不包含指定類型的檔案。

-f：中斷最後的目錄串列，而且通知檔名的開始。

-B 目錄：在指定的目錄下搜尋執行檔。

-M 目錄：在指定的目錄下搜尋說明檔。

-S 目錄：在指定的目錄下搜尋原始碼檔。

我們使用 whereis ftp 來搜尋 ftp 伺服器的所在目錄。

```
[root@flash chaiyen]# whereis ftp
ftp: /usr/bin/ftp /usr/bin/ftp.expect /usr/share/man/man1/ftp.1.gz
```

我們使用 whereis –b cat 來搜尋 cat 指令的所在目錄。

```
[root@flash chaiyen]# whereis -b cat
cat: /bin/cat
```

我們可以查尋我們所指定的指令和執行檔的所在位置。

```
[root@flash chaiyen]# whereis gzip gunzip ps
gzip: /bin/gzip /usr/bin/gzip /usr/share/man/man1/gzip.1.gz
gunzip: /bin/gunzip /usr/bin/gunzip /usr/share/man/man1/gunzip.1.gz
ps: /bin/ps /usr/share/man/man1/ps.1.gz
```

我們使用 which 來在環境變數$PATH 中找尋符合的檔案。

語法：

指令：which 檔案

參數：

-a：列出所有符合的檔案。

我們使用 which ps 來在環境變數$PATH 中找尋符合的檔案。

```
[root@flash chaiyen]# which ps
/bin/ps
```

我們使用 grep 指令來搜尋在檔案串列中的字串、運算式或範本。

語法：

指令：grep 參數 範本 檔案串列

egrep 參數 字串 檔案串列

fgrep 參數 運算式 檔案串列

參數：

-c：僅列出符合行數的編號。

-i：在搜尋中忽略大小寫。

-l：僅列出合乎檔案的名稱。

-n：列出該行的編號。

-s：不顯示錯務資訊。

-v：列出不符合字串的行數。

-w：顯示符合字串的行數。

-a：把二進位檔當作是文字檔處理。

-f 範本檔：指定範本。

我們使用 grep Sarwar student.c 來搜尋在 student.c 檔案中符合 Sarwar 字串的那一行。

```
[root@flash chaiyen]# grep Sarwar student.c
david Sarwar    david@msa.hinet.net     61.218.29.2
[root@flash chaiyen]# more student.c
david Sarwar    david@msa.hinet.net     61.218.29.2
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6
```

我們使用 grep –n Sarwar student.c 來搜尋在 student.c 檔案中符合 Sarwar 字串的那一行，並列出該行的編號。

```
[root@flash chaiyen]# grep -n Sarwar student.c
1:david Sarwar    david@msa.hinet.net     61.218.29.2
```

我們使用 fgrep Sarwar student.c 來搜尋在 student.c 檔案中符合 Sarwar 字串的那一行。我們使用 fgrep –n Sarwar student.c 來搜尋在 student.c 檔案中符合 Sarwar 字串的那一行，並列出該行的編號。

```
[root@flash chaiyen]# fgrep Sarwar student.c
david Sarwar    david@msa.hinet.net     61.218.29.2
[root@flash chaiyen]# fgrep -n Sarwar student.c
1:david Sarwar    david@msa.hinet.net     61.218.29.2
```

我們使用 egrep –n Sarwar student.c 來搜尋在 student.c 檔案中符合 Sarwar 字串的那一行，並列出該行的編號。我們使用 egrep Sarwar student.c 來搜尋在 student.c 檔案中符合 Sarwar 字串的那一行。

```
[root@flash chaiyen]# egrep -n Sarwar student.c
1:david Sarwar    david@msa.hinet.net     61.218.29.2
[root@flash chaiyen]# egrep Sarwar student.c
david Sarwar    david@msa.hinet.net     61.218.29.2
```

15-4-4 剪下和貼上

我們可以在 Linux 上使用 cut 剪下指令及 paste 貼上指令，來處理和儲存檔案。

我們使用 cut 指令從檔案剪下表格的欄位。

語法

指令：cut 參數　檔案

參數：

-b 位元組：只有輸出指定位元組。

-c 字元：只有輸出這些字元。

-d 分界字元：指定欄位分界字元。

-f 欄位：只有輸出這些欄位。

我們使用 cut –c 2 student.c 來輸出每行第二個字元。

```
[root@flash chaiyen]# cut -c 2 student.c
a
a
a
i
a
o

[root@flash chaiyen]# more student.c
david Sarwar    david@msa.hinet.net     61.218.29.2
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6
```

我們可以使用 paste 指令，將多個檔合併輸出到一個檔中。

語法：

指令：paste 參數 檔案串列

參數：

-d 字元：使用指定字元當分隔檔案，而不是 TABs。

-s：不是以平行的方式貼上檔案。

--help：說明

我們使用 paste student.c love.txt > good.c 將 student.c 和 love.txt 合併到 good.c 的檔案中。

```
[root@flash chaiyen]# paste student.c love.txt > good.c
```

15-4-5 加密和解密

uuencode 指令傳送加密的 Ascii 檔案到標準輸出。

我們使用 uuencode 來加密二進位的原始檔到 ASCII 檔。

語法：

指令：uuencode 檔案 輸出裝置

參數：

我們使用 uudecode 指令來解密從 ASCII 到二進位。

語法：

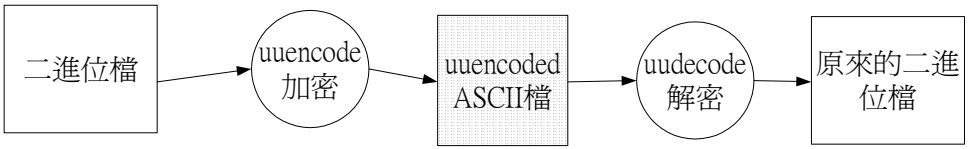
指令：uudecode 參數 加密檔案

參數：

-p：傳送加密版本的二進位檔到標準輸出。

-o 檔案：將解密的檔案轉換成指定檔案。

這是加密解密的過程，我們將二進位檔加密成 ASCII 檔，再將 ASCII 檔解密成原來的二進位檔。



這是加密前 student.c 的檔案。

```
[root@flash chaiyen]# more student.c
david Sarwar    david@msa.hinet.net     61.218.29.2
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6
```

我們使用 uuencode student.c /dev/stdout > tt.c 將 student.c 加密，並且將它輸出到 tt.c 的檔案中。這時 tt.c 就是經過加密的檔案，它的編碼方式變了。

```
[root@flash chaiyen]# uuencode student.c /dev/stdout > tt.c
[root@flash chaiyen]# more tt.c
begin 644 /dev/stdout
M9&%V:60@4V%R=V%R("`@9&%V:61`;7-A+FAI;F5T+FYE="`@("`V,2XR,3@N
M,CDN,@I(87-S86X@2V5N9&%L("!(87-S86Y`8VTQ+FAI;F5T+FYE="`@(#8Q
M+C(Q."XR.2XP"DYA8F5E;"!*;VAN<V5N($YA8F5E;$!S:6YA+F-O;2YT=R`@
M("`@-C$N,C$X+C(Y+C,*36EC:&%L($IA8VMS;VX@36EC:&%L0'EA:&]O+F-O
M;2YT=R`@("`V,2XR,3@N,CDN-0I-87)Y("!W=2`@("`@("!-87)Y0'EA:&]O
M+F-O;2YT=R`@("`@(#8Q+C(Q."XR.2XR"F=O9&1E<W,@='-E;B`@(&=O9&1E
@<W-`9&1M+F]R9RYT=R`@("`@-C$N,C$X+C(Y+C8@"@H`
`
end
```

我們使用 uudecode –o yy.c tt.c 將 tt.c 解密到 yy.c 的檔中。這時 yy.c 的檔案就回到 student.c 的檔案一樣。

```
[root@flash chaiyen]# uudecode -o yy.c tt.c
[root@flash chaiyen]# more yy.c
david Sarwar    david@msa.hinet.net     61.218.29.2
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6
```

15-4-6 指令記錄

我們可以使用 history 指令來讀取最近使用的指令。

語法：

指令：history 參數　檔案名稱

參數：

-a 檔案名稱：history 檔案中加上記錄。

-N：最近 N 次所下的指令。

-c：清除歷史檔案。

-w 檔案：將目前的歷史串列寫到檔案。

我們使用 history 10 來顯示最近 10 次所使用的指令。

```
[root@flash chaiyen]# history 10
 1079  ls
 1080  uudecode -o gigi.c ll.c
 1081  vi gigi.c
 1082  clear
 1083  uuencode student.c /dev/stdout > tt.c
 1084  more tt.c
 1085  uudecode -o yy.c tt.c
 1086  more yy.c
 1087  more student.c
 1088  history 10
```

我們使用 fc –l 來顯示歷史的指令並且顯示編號。

語法：

指令：fc 參數

參數：

-l：顯示歷史的指令並且顯示編號。

我們使用 fc –l 來顯示歷史的指令並且顯示編號。

```
[root@flash chaiyen]# fc -l
2074     rm 001.tif
2075     clear
2076     vi student.c
2077     clear
2078     cat students
2079     cat student
2080     ls
2081     cat student.c
2082     vi student.c
2083     clear
2084     cat student.c
2085     cp student.c /home/chaiyen
2086     sort student.c
2087     sort -n student.c
2088     sort +1 student.c
2089     man find
2090     clear
```

15-4-7 RPM 管理套件

RPM 指的是 RedHat Package Manager 的縮寫，他是由紅帽公司所開發的工具。

透過 rpm 的管理，我們可以將某個新軟體的 source code 重新包裝成另一種 source 與 binary 的檔案型態，如此我們可以由 binary(二進位)的檔案，進行安裝與套件追蹤管理的工作，而 source 的檔案，也很方便的能夠再被重新整理包裝。

Rpm 管理了一份資料庫，裡面包含了所有的程式套件的檔案資料，透過這份資料庫，我們可以進行程式的確認與查詢工作。

當我們要安裝、查詢、辨視、更新、移除套件時可以使用 rpm 這個指令

語法：

指令： rpm –I(ihv,uhv) 套件名稱

參數：

-v：顯示執行過程。

-h：安裝套件實列出五十個雜湊標記。

-a：查詢所有安裝的套件。

-f 檔案：查尋擁有檔案的套件。

-p 套件：查詢指定的套件。

-c：列出所有組態檔。

-d：只有列出文件檔案。

-i：顯示套件資訊，包含名稱、版本、和描述。

-l：列出套件的檔案。

-R：列出相依的套件。

-s：顯示套件內檔案的狀態。

--nodes：不顯示相依性質。

-U 套件：升級指定的套件。

-q：使用交談模式。

-e 套件：刪除指定的套件。

-F 套件：更新指定的套件。

-i 套件：安裝指定的套件。

裝上 apache-1.3.20

裝上並更新 apache-1.3.20

移除，當我們要移除套件時可以使用這個指令

指令： rpm –e 套件名稱

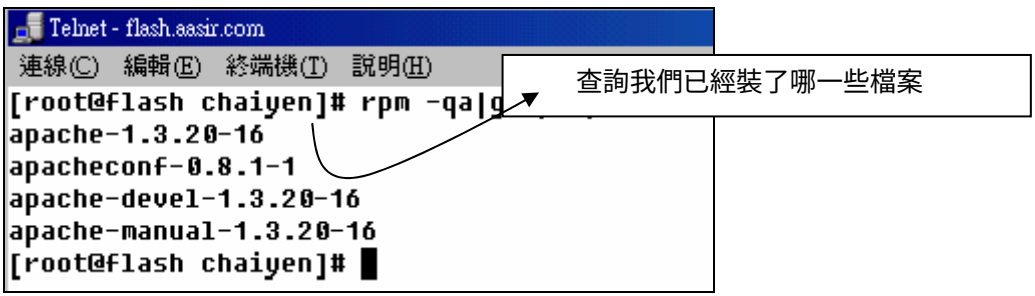
```
[root@flash chaiyen]# rpm -e apache-1.3.20-16
```

驗證，當我們不小心刪除了檔案，但卻不知道是哪些檔案，我們可以使用驗證來了解已經遺失或損壞的檔案。

指令：rpm –va

查詢：假如我們碰到一個認不出來的檔案，而我們卻想知道他是屬於哪一個套件，可以使用查尋指令。

指令：Rpm –qa | grep 檔案名

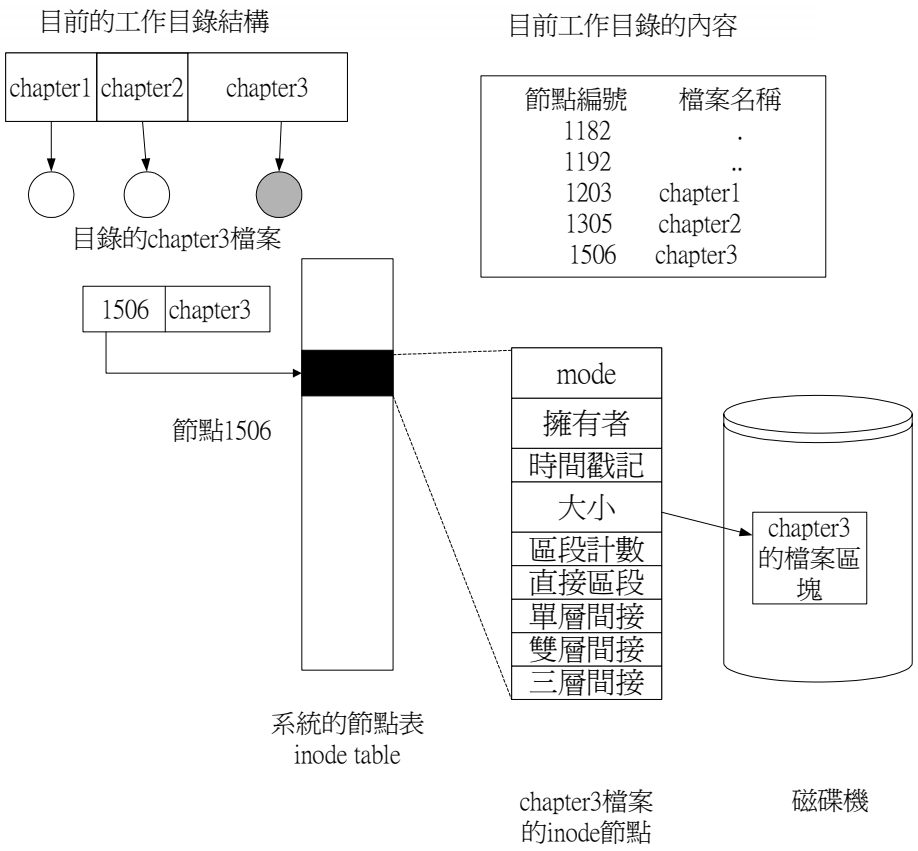


15-5 檔案分享與管線和重新指向

15-5-1 硬連結

硬連結(hard link)就是指向檔案的節點。當在 Linux 建立檔案時，系統會分配唯一的檔案節點，而且會在建立檔案的目錄上建立進入點。檔案的節點 inode 通常是指向節點的內容，它是包含著檔案的屬性(連結數、擁有者、檔案更新時間、檔案在磁碟的位置、檔案的大小、檔案的使用者權限……)。假如我們在目前的工作目錄上建立 chapter3 的檔案，而且系統分配編號 1506 的節點給它。因此這個檔案的目錄進入點就是(節點 1506 ,chapter3)。我們預設目前的工作目錄已有 chapter1、chapter2 的檔案，這個灰色的檔案就是我們新建立的。目前目錄的工作內容就是由節點編號和檔案名稱所組成，而節點編號通常指向系統的節點表(system inode table)。系統的節點表通常是在記憶體中，這節點表指出在磁碟機上的檔案內容、與檔案的屬性。

在 Linux 作業系統，我們可以使用 ln 指令來建立檔案的連結。這個連結給我們一個目錄的進入點，來指向這個節點編號，也就是指向這個檔案的節點表。



節點,檔案內容和目錄的進入點的關係

我們使用 ln 指令來建立硬連結(hard link)從來源檔案到目地檔。

語法：

指令：ln 參數 已存在的檔案或目錄 新的檔案或目錄

參數：

-f ：強迫建立連結

-s ：建立符號連結 symbolic link(又稱為軟連結)

-b：在移除檔案前先備份

-f：移除已存在的檔案或目錄。

-n：當新的檔案或目錄已存在時，則不建立連結。

-i：詢問是否覆蓋檔案。

-s：設定符號連結 symbolic link 而不是硬連結。

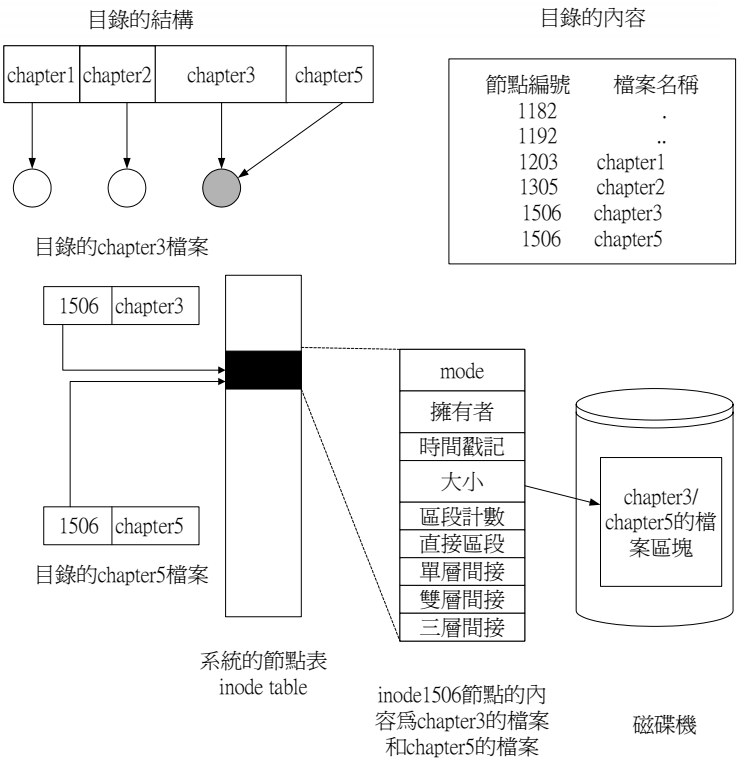
-v：在連結前列印每一個的檔案名稱。

Chapter5 為我們的硬連結(hard link)檔案。我們使用 ln chapter3 chapter5，來將新建立的檔案 chapter5 連結到 chapter3。Chapter5 擁有和 chapter3 相同的節點編號，因為它們都是指向同一個 chapter3 的檔案內容，因此它們的檔案大小都相同為 244 個位元組，而且它們 chapter3 和 chapter5 的節點編號都為 555975。Ls –il 指令顯示了在我們目前工作目錄檔案的屬性(包含它們的節點編號)。在下圖中它也顯示了 chapter3 和 chapter5 的連結數為 2，因為 chapter3 和 chapter5 連結到這檔案。

```
[root@flash now]# ls -il
total 15668
 555973 -rw-r--r--    1 root     root          302  9月  7 16:30 chapter1
 555974 -rw-r--r--    1 root     root     16013308  9月  7 16:31 chapter2
 555975 -rw-r--r--    1 root     root          244  9月  7 16:31 chapter3
[root@flash now]# ln chapter3 chapter5
[root@flash now]# ls -il
total 15672
 555973 -rw-r--r--    1 root     root          302  9月  7 16:30 chapter1
 555974 -rw-r--r--    1 root     root     16013308  9月  7 16:31 chapter2
 555975 -rw-r--r--    2 root     root          244  9月  7 16:31 chapter3
 555975 -rw-r--r--    2 root     root          244  9月  7 16:31 chapter5
```

我們現在將 chapter3 給移除，則 chapter5 的連結數會變成 1。

```
[root@flash now]# rm chapter3
[root@flash now]# ls -il
total 15668
 555973 -rw-r--r--    1 root     root          302  9月  7 16:30 chapter1
 555974 -rw-r--r--    1 root     root     16013308  9月  7 16:31 chapter2
 555975 -rw-r--r--    1 root     root          244  9月  7 16:31 chapter5
```



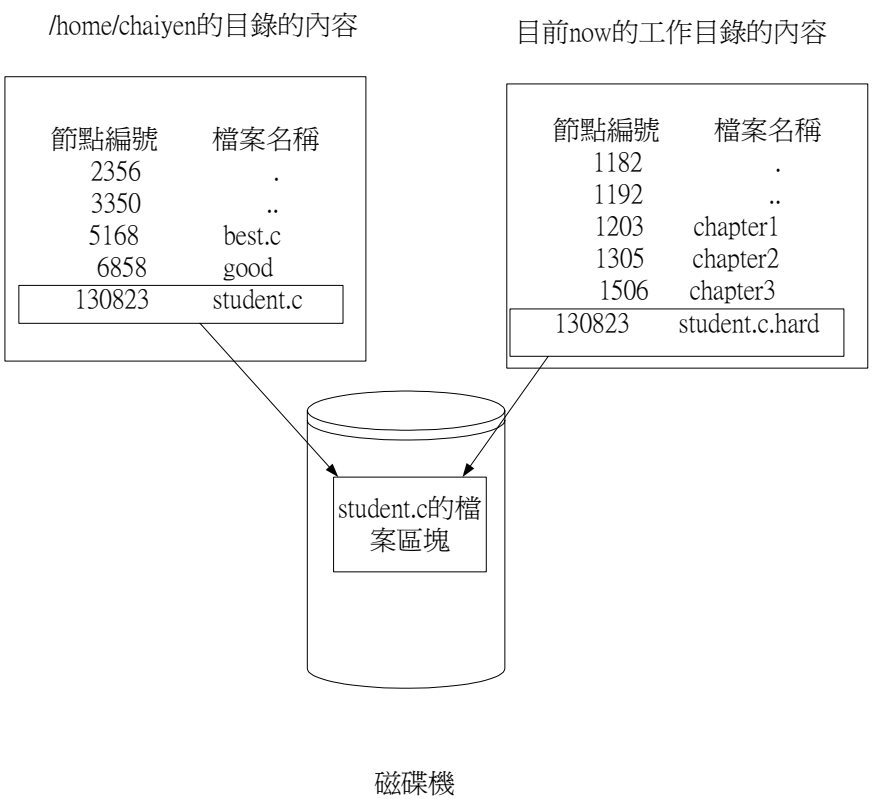
檔案的硬連結hard link

我們可以觀看我們/home/chaiyen 的目錄下 student.c 檔案的屬性，它的節點編號為 130823，而它的檔案連結數為 1，檔案的大小為 302，而檔案建立的時間為 9 月 6 日。

```
[root@flash now]# ls -il /home/chaiyen/student.c
 130823 -rw-r--r--    1 root     root          302  9月  6 09:31 /home/chaiyen/s
tudent.c
```

我們使用 ln /home/chaiyen/student.c student.c.hard 來建立硬連結，將 student.c.hard 的檔案連結到/home/chaiyen/student.c 的檔案上。這時我們觀看 student.c 的屬性會和 student.c.hard 的屬性相同，它們有相同的節點編號、相同的連結數…。

```
[root@flash now]# ln /home/chaiyen/student.c student.c.hard
[root@flash now]# ls -il /home/chaiyen/student.c
 130823 -rw-r--r--    2 root     root          302  9月  6 09:31 /home/chaiyen/s
tudent.c
[root@flash now]# ls -il /home/chaiyen/now/student.c.hard
 130823 -rw-r--r--    2 root     root          302  9月  6 09:31 /home/chaiyen/n
ow/student.c.hard
```



student.c和student.c.hard共同分享檔案內容

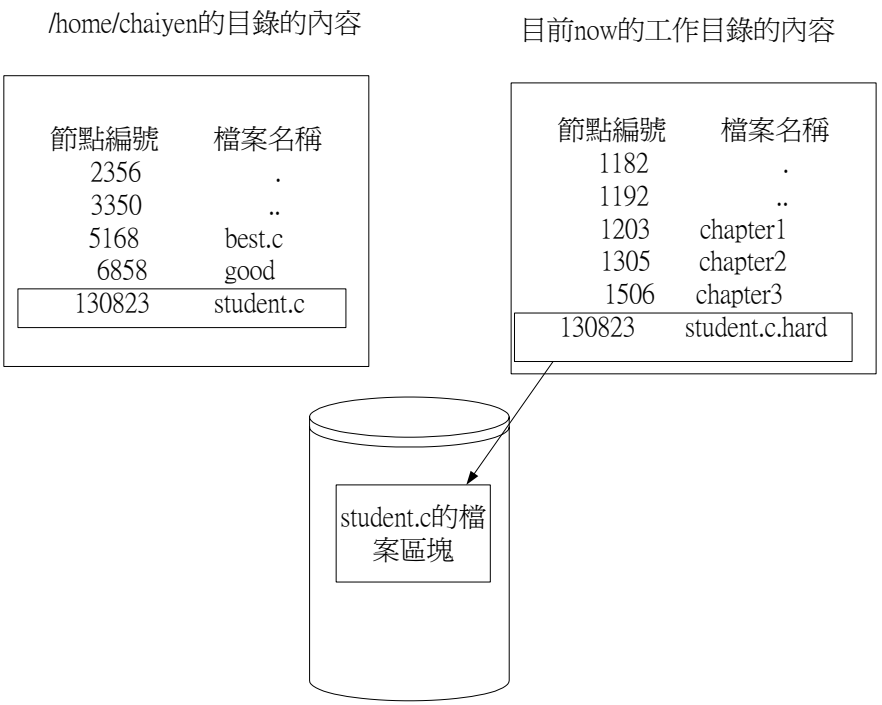
我們使用 more 就可以看到這兩個檔案的內容都相同，因為它們的檔案節點都是指到這個內容。

```
[root@flash now]# more student.c.hard
david Sarwar    david@msa.hinet.net     61.218.29.2
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6

[root@flash now]# more /home/chaiyen/student.c
david Sarwar    david@msa.hinet.net     61.218.29.2
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6
```

當我們使用 rm student.c 將 student.c 的檔案刪除，這時/home/chaiyen/student.c 的節點表也就被刪除。但觀看 student.c.hard 硬連結，還可以看到檔案的內容。因為 student.c.hard 的節點表還是可以連結到 student.c 在磁碟機上的檔案。

```
[root@flash chaiyen]# rm student.c
rm: remove `student.c'? y
[root@flash chaiyen]# cd now
[root@flash now]# ls -il student.c.hard
 130823 -rw-r--r--    1 root     root          302  9月  6 09:31 student.c.hard
[root@flash now]# more student.c.hard
david Sarwar    david@msa.hinet.net     61.218.29.2
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6
```



在/home/chaiyen目錄下的student.c檔案已被刪除,其結點也就沒有指向student.c,但是student.c.hard的節點表依舊指向

15-5-2 符號連結

我們使用 ln -s 指令來建立符號連結(sybolic link)從已存在的檔案或目錄到新的檔或目錄。

語法：

指令：ln -s 已存在的檔案或目錄 新的檔案或目錄

參數：

-f ：強迫建立連結

-s ：建立符號連結 symbolic link(又稱為軟連結)

-b：在移除檔案前先備份

-f：移除已存在的檔案或目錄。

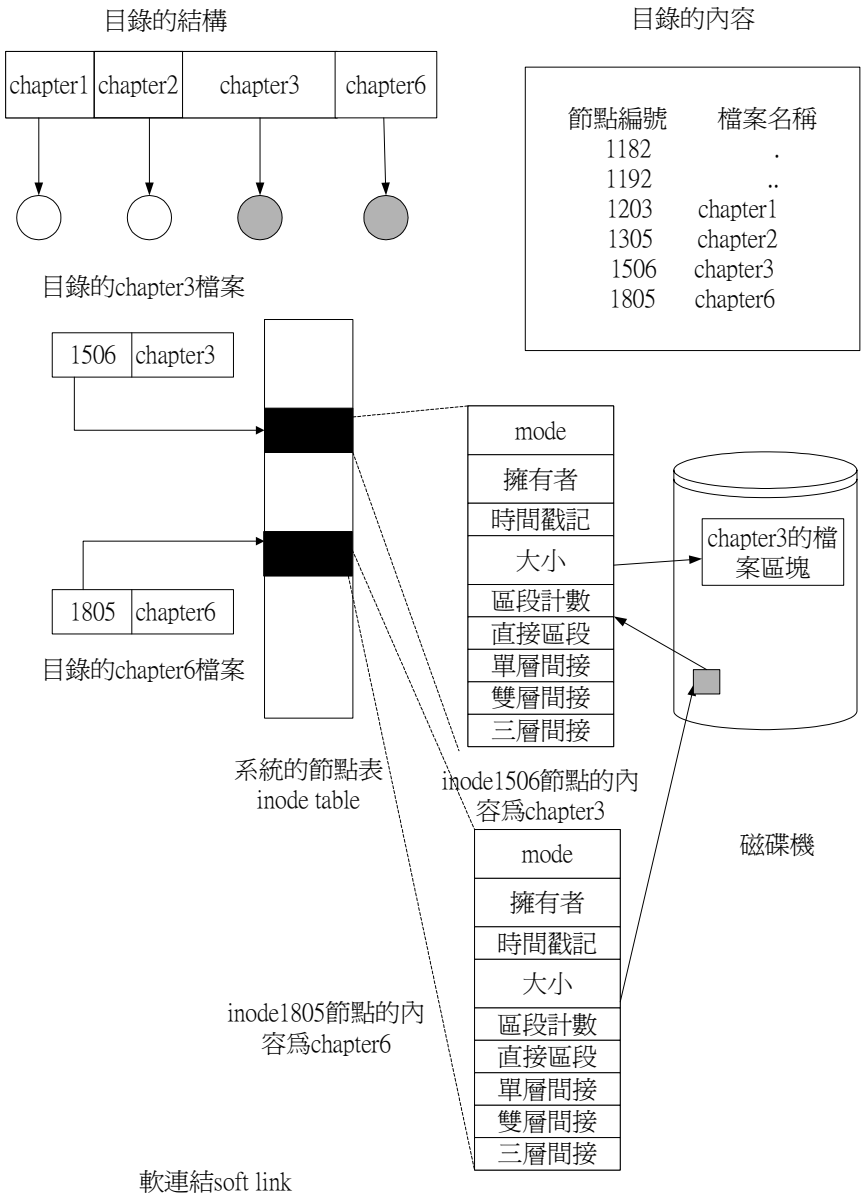
-n：當新的檔案或目錄已存在時，則不建立連結。

-i：詢問是否覆蓋檔案。

-s：設定符號連結 symbolic link 而不是硬連結。

-v：在連結前列印每一個的檔案名稱。

這是符號連結 symbolic link(又稱軟連結)的示意圖，我們使用 ln –s chapter3 chapter6 指令，來將 chapter6 符號連結 chapter3。這時它們個自有不同的檔案名稱和節點編號。chapter3 的節點指向到 chapter3 的檔案區塊。而 chapter6 的節點是指向 chapter6 的檔案區塊。這個 chapter6 的檔案區塊裝的是 chapter3 區塊的檔案位置。因此 chapter3 和符號連結 chapter6 共同分享 chapter3 的檔案區塊。


目錄的結構
chapter1
chapter2
chapter3
chapter6
目錄的內容
節點編號 檔案名稱
1182 .
1192 ..
1203 chapter1
1305 chapter2
1506 chapter3
1805 chapter6
目錄的chapter3檔案
1506 chapter3
1805 chapter6
目錄的chapter6檔案
mode
擁有者
時間戳記
大小
區段計數
直接區段
單層間接
雙層間接
三層間接
chapter3的檔案區塊
系統的節點表
inode table
inode1506節點的內容爲chapter3
磁碟機
mode
擁有者
時間戳記
大小
區段計數
直接區段
單層間接
雙層間接
三層間接
inode1805節點的內容爲chapter6
軟連結soft link

我們可以使用 ln 指令加上-s 參數來建立符號連結(軟連結)。

```
[root@flash now]# ls -il
total 15672
 555973 -rw-r--r--    1 root     root          302  9月  7 16:30 chapter1
 555974 -rw-r--r--    1 root     root     16013308  9月  7 16:31 chapter2
 555975 -rw-r--r--    1 root     root          244  9月  7 16:31 chapter5
 130823 -rw-r--r--    2 root     root          302  9月  6 09:31 student.c.hard
[root@flash now]# ln -s chapter5 chapter5.soft
```

chapter5.soft 就是我們的符號連結，它會出現 chapter5.soft->chapter5 的檔案，這表示 chapter5.soft 是指向 chapter5 節點屬性的檔案。符號連結 symbolic link 檔案所存放的內容是指向 chapter5 檔案的內容節點，因此我們可以看到 chpter5.soft 的檔案大小只有 8 個 bytes。符號連結 Chapter5.soft 的節點編號和 chapter5 不同，而且 chapter5.soft 的檔案內容是放置 chapter5 的節點內容。符號連結 chapter5.soft 的節點編號是 555976 和 chapter5 的節點編號 555975 不同。

```
[root@flash now]# ls -il
total 15672
 555973 -rw-r--r--    1 root     root          302  9月  7 16:30 chapter1
 555974 -rw-r--r--    1 root     root     16013308  9月  7 16:31 chapter2
 555975 -rw-r--r--    1 root     root          244  9月  7 16:31 chapter5
 555976 lrwxrwxrwx    1 root     root            8  9月  7 23:04 chapter5.soft -> chapter5
 130823 -rw-r--r--    2 root     root          302  9月  6 09:31 student.c.hard
```

我們也可以跨過不同檔案系統來建立符號連結。我們先將 chapter5 拷背到/usr 目錄下，我們再用 ln –s /usr/chapter5 usr.soft 來符號連結。我們可以看到 chapter5 的節點編號是 3807，而 usr.soft 的節點編號是 555977，這兩個檔案的節點編號不一樣。

```
[root@flash now]# ls
chapter1  chapter2  chapter5  chapter5.soft  student.c.hard
[root@flash now]# cp chapter5 /usr
[root@flash now]# ln -s /usr/chapter5 usr.soft
```

```
[root@flash now]# ls -il /usr/chapter5 usr.soft
   3807 -rw-r--r--    1 root     root          244  9月  8 07:46 /usr/chapter5
 555977 lrwxrwxrwx    1 root     root           13  9月  8 07:53 usr.soft -> /usr/chapter5
```

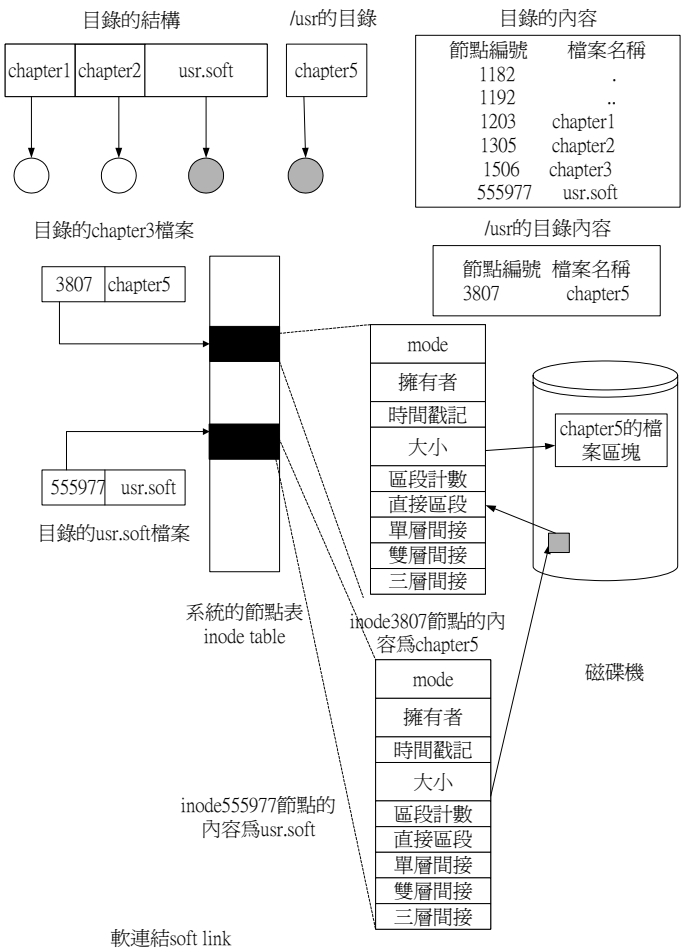
我們可以使用 more 來觀看 usr.soft 和 chapter5 的檔案內容都一樣，因為它們在磁碟機的檔案內容都是同一個。而 usr.soft 符號連結檔案的內容就是 chapter5 檔案的位址。

```
[root@flash now]# more usr.soft
PK       #-  +酆i  g.cUT          lt=
w=Ux          宣敨 N ;%凍鏟  g 祛L迕e驌凫 挸伽矍+嶠U/i |逭妲重聱       糲 諴i捽
 g.cUT   lt=UxPK     > PK          #-  +酆i
[root@flash now]# more /usr/chapter5
PK       #-  +酆i  g.cUT          lt=
w=Ux          宣敨 N ;%凍鏟  g 祛L迕e驌凫 挸伽矍+嶠U/i |逭妲重聱       糲 諴i捽
 g.cUT   lt=UxPK     > PK          #-  +酆i
```

當我們將檔案/usr/chapter5 給刪除，則符號連結檔 usr.soft 將無法指向/usr/chapter5 的檔案，因為 chapter5 的節點表及檔案位址已經被刪除，usr.soft 符號連結檔就無法連結到 chapter5 的檔案位置了。這種情況稱為"不連結"dangling link。

```
[root@flash now]# rm /usr/chapter5
rm: remove `/usr/chapter5'? y
```

```
[root@flash now]# ls -il usr.soft
 555977 lrwxrwxrwx    1 root     root           13  9月  8 07:53 usr.soft -> /usr/chapter5
```



15-5-3 符號連結的搜尋

我們可以使用 symlinks 指令來搜尋指定目錄串列中的符號連結目錄, 而且顯示有關它們的資訊。

語法：

指令：symlinks 參數 目錄串列

參數：

-c：轉換絕對路徑到相對路徑

-d：刪除”不連結”(dangling link)的符號連結。

-r：在目錄串列中遞迴搜尋目錄。

-s：偵測 lengthy 符號連結。Lengthy 符號連結是在路徑上使用”../”來指向目標檔案。

-v：搜尋相對路徑的符號連結。

-t：測試-c 參數是否在 symlinks 可用。

我們使用 symlinks –r /home/chaiyen 來觀察/home/chaiyen 目錄下的符號連結情況。這 dangling 表示/usr/chapter5 這個檔案已經遺失了，因此產生 dangling 不連結的情況。

```
[root@flash chaiyen]# symlinks -r  /home/chaiyen
dangling: /home/chaiyen/now/usr.soft -> /usr/chapter5
```

我們使用 symlinks –r /home 來遞迴的搜尋符號連結。Absolute 是絕對路徑的符號連結，它的目地檔案開始是從根目錄(/)開始。

```
[root@flash chaiyen]# symlinks -r /home
dangling: /home/chaiyen/now/usr.soft -> /usr/chapter5
absolute: /home/temp.soft -> /home/chaiyen/yy.c
```

我們使用 symlinks –v 來顯示所有相對路徑 relative 的符號連結。相對路徑 relatvie 的符號連結是相對於我們所指向檔案的相對路徑而言。

```
[root@flash chaiyen]# symlinks -v /bin |more
relative: /bin/dnsdomainname -> hostname
relative: /bin/nisdomainname -> hostname
relative: /bin/domainname -> hostname
relative: /bin/sh -> bash
relative: /bin/ypdomainname -> hostname
relative: /bin/bash2 -> bash
relative: /bin/red -> ed
relative: /bin/awk -> gawk
relative: /bin/bsh -> ash
relative: /bin/gtar -> tar
relative: /bin/csh -> tcsh
relative: /bin/ex -> vi
relative: /bin/rvi -> vi
relative: /bin/rview -> vi
relative: /bin/view -> vi
```

other_fs 的符號連結是指我們符號連結的檔案系統和我們 other_fs 符號連結檔案系統不同。

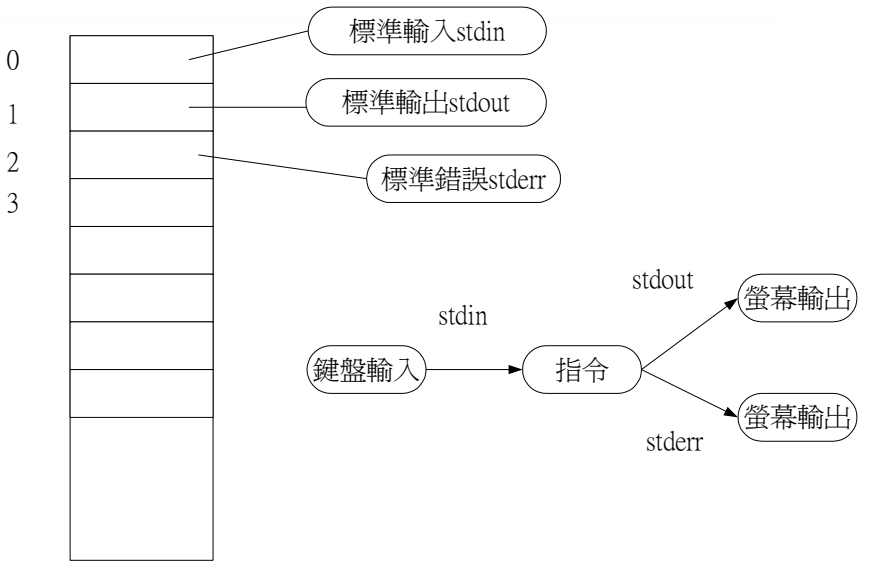
```
[root@flash chaiyen]# symlinks  /usr
other_fs: /usr/tmp -> ../var/tmp
```

在 messy 符號連結，它包含了不需要的逗點或 / (slash)，在路徑上。

```
[root@flash /]# symlinks -r /usr |more
messy:    /usr/bin/redhat-config-apache -> ./apacheconf
messy:    /usr/bin/apacheconf -> ./consolehelper
```

15-6 管線和轉向

輸入、輸出與處理是典型的檔案操作。在 Linux 上指令的輸入、輸出和錯誤訊息可以轉向到標準的檔案。這可以讓我們將好幾個複雜的指令合起來使用。在 Linux 上有三個檔是自動被核心打開來給每個指令當作輸出、輸入和錯誤資訊使用。這些檔案像是標準輸入檔 stdin、標準輸出檔 stdout、和標準錯誤檔 stderr。這些檔與在預設的指令執行終端有關。而鍵盤的輸入為標準輸入 stdin、螢幕的顯示為標準輸出。在標準的檔案中，指令的輸入一般都是使用鍵盤，而輸出大部份都在螢幕的顯示。



每個行程的檔案描述表

15-6-1 輸入轉向

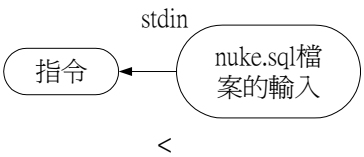
我們可以使用指令 < 輸入檔，從輸入檔輸入資料到指令。小於<的符號為輸入的符號。

語法：

指令：指令 < 輸入檔

我們可以使用 mysql nuke < nuke.sql 來將 nuke.sql 的資料輸入到 mysql 的 nuke 資料庫中。<為輸入的符號。

```
[root@flash chaiyen]# mysql nuke < nuke.sql
```



15-6-2 輸出轉向

我們可以使用指令 > 輸出檔，將指令產生的資料輸出到輸出檔。大於>的符號為輸出的符號。

語法：

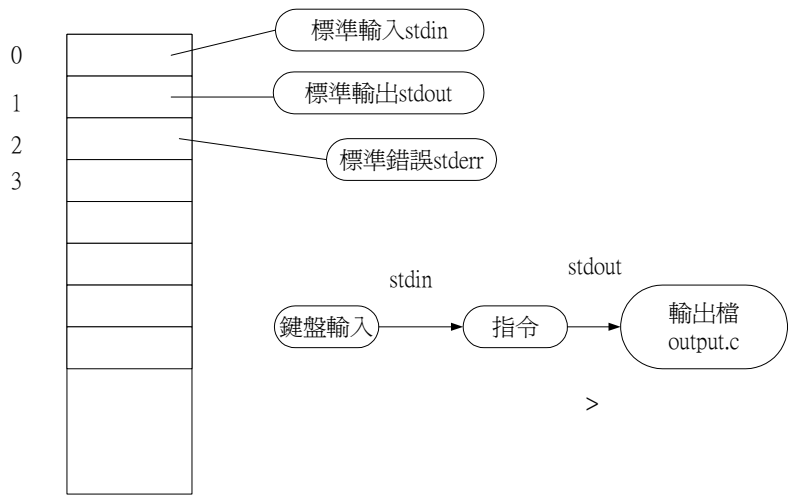
指令：指令 > 輸出檔案

參數：

我們將 cat yy.c 的資料輸出到 output.c。我們使用>大於的符號來表示輸出。

```
[root@flash chaiyen]# cat yy.c > output.c
[root@flash chaiyen]# more output.c
david Sarwar   david@msa.hinet.net    61.218.29.2
Hassan Kendal  Hassan@cml.hinet.net   61.218.29.0
Nabeel Johnsen Nabeel@sina.com.tw     61.218.29.3
Michal Jackson Michal@yahoo.com.tw    61.218.29.5
Mary  wu       Mary@yahoo.com.tw      61.218.29.2
goddess tsen   goddess@ddm.org.tw     61.218.29.6
iii
```

這是我們將資料輸出到 output.c 的示意圖。我們的指令經過鍵盤輸入，經過執行，在將資料標準輸出到輸出檔 output.c。



每個行程的檔案描述表

15-6-3 組合輸出和輸入轉向

我們可以使用指令 < 輸入檔，從輸入檔輸入資料到指令。小於<的符號為輸入的符號。我們可以使用指令 > 輸出檔，將指令產生的資料輸出到輸出檔。大於>的符號為輸出的符號。我們可以使用<小於的符號為輸入的符號，我們使用>輸出的符號將資料輸出。

語法：

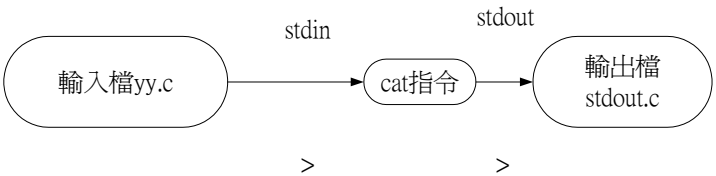
指令：指令 < 輸入檔 > 輸出檔

指令 > 輸出檔 < 輸入檔

我們使用 cat < yy.c > stdout.c 將 yy.c 的檔案資料輸入到 cat 指令，cat 指令再將資料傳送到 stdout.c 的檔案中。

```
[root@flash chaiyen]# cat < yy.c > stdout.c
[root@flash chaiyen]# more stdout.c
david Sarwar    david@msa.hinet.net     61.218.29.2
Hassan Kendal   Hassan@cml.hinet.net    61.218.29.0
Nabeel Johnsen  Nabeel@sina.com.tw      61.218.29.3
Michal Jackson  Michal@yahoo.com.tw     61.218.29.5
Mary  wu        Mary@yahoo.com.tw       61.218.29.2
goddess tsen    goddess@ddm.org.tw      61.218.29.6
iii
```

我們使用 cat < yy.c > stdout.c 將 yy.c 的檔案資料輸入到 cat 指令，cat 指令再將資料輸送到 stdout.c 的檔案中。



當錯誤的資訊由我們的指令產生時，我們可以將它輸入到錯誤儲存檔中。

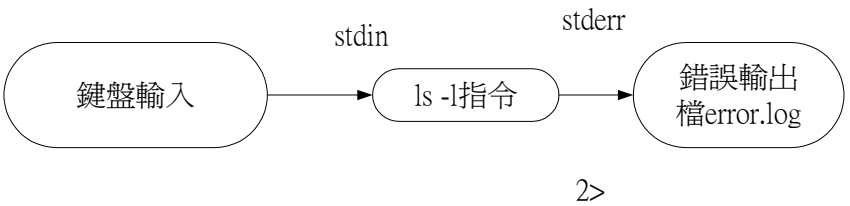
語法：

指令：指令 2> 錯誤儲存檔

當我們的目錄沒有 bibo 檔案，而我們又用 ls 指令來尋找 bibo 檔案，這時錯誤的

資訊由我們的指令 ls 產生，我們可以將它輸入到錯誤儲存檔 error.log 中。而我們使用 ls –l bibo 2 > error.log 將錯誤資訊輸入到 error.log 的檔案中。

```
[root@flash chaiyen]# ls -l bibo 2> error.log
[root@flash chaiyen]# vi error.log
ls: bibo: 沒有此一檔案或目錄
```



我們使用>>增添符號，將輸出的資料添加到指定檔的後面。

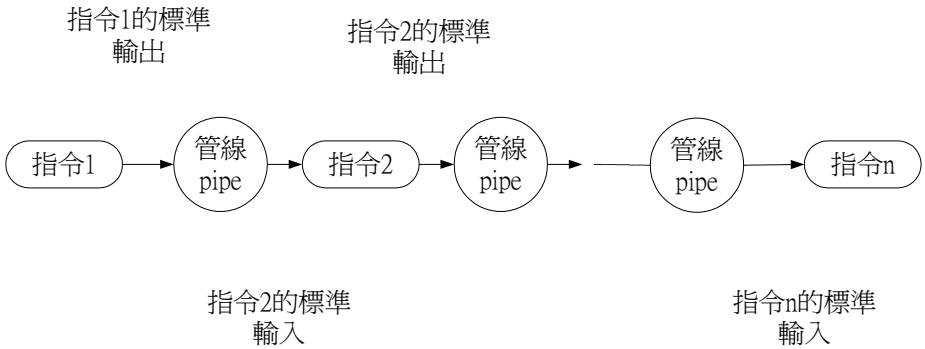
```
[root@flash chaiyen]# ls -l bibo 2> error.log
[root@flash chaiyen]# ls >> error.log
[root@flash chaiyen]# more error.log
ls: bibo: 沒有此一檔案或目錄
002.tif.Z
003.tif
005.tif
bb.c
```

15-6-4 管線 PIPES

我們可以使用指令 1|指令 2|指令 3|.....|指令 N 來將指令 1 的標準輸出連結到指令 2 的標準輸入，指令 2 的標準輸出連結到指令 3 的標準輸入，指令 3 的標準輸出連結到,,,,,,,,,，指令 n-1 的標準輸出連結到指令 n 的標準輸入。我們經常使用到管線，來將指令 1 的標準輸出處理，和指令 2 的標準輸入處理，這個管線處理過程稱為過濾。我們的指令有 cat、compress、crypt、grep、gzip、lp、pr、sort.tr 和 uniq 與 wc 常運用到管線。

語法：

指令：指令 1|指令 2|指令 3|.....|指令 N

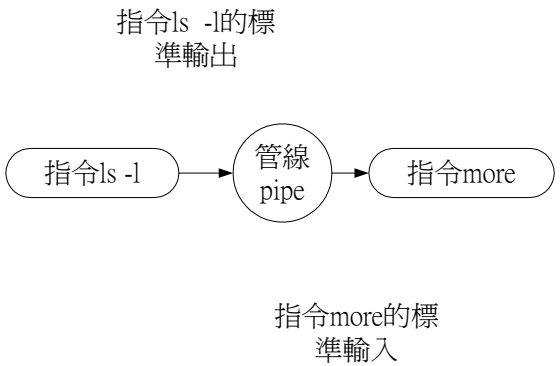


這是我們將 ls –l 輸出到 log 檔案，再從 log 檔案輸入到 more 指令來顯示。

```
[root@flash good]# ls -l >log
[root@flash good]# more < log
total 360
drwxr-xr-x    2 root     root         4096  9月  4 13:40 best
-rw-r--r--    1 root     root            0  9月  9 14:28 log
-rwx------    1 root     root        47978  9月  9 14:27 快照1.png
-rwx------    1 root     root        41114  9月  9 14:27 快照2.png
-rwx------    1 root     root        58315  9月  9 14:27 快照3.png
```

我們使用管線 ls –l | more 來顯示內容。當資料過多時，我們可以將資料存在管線中，再慢慢讀出來。

```
[root@flash good]# ls -l | more
total 364
drwxr-xr-x    2 root     root         4096  9月  4 13:40 best
-rw-r--r--    1 root     root          669  9月  9 14:28 log
-rwx------    1 root     root        47978  9月  9 14:27 快照1.png
-rwx------    1 root     root        41114  9月  9 14:27 快照2.png
-rwx------    1 root     root        58315  9月  9 14:27 快照3.png
```



15-6-5 管線組合和轉向

我們可以使用 tee 指令來同時使用轉向運算子和管線。我們可以用 tee 指令來讀取輸入檔，在將它送到標準輸出和指定的檔案串列中。

語法：

指令：tee　參數　檔案串列

參數：

-a：將資料添加到檔案串列後面，而不去覆寫檔案。

-i：忽略中斷信號。

--help：顯示說明。

這是使用 tee 指令來作轉向和管線，我們按下<Ctrl+D>就可以跳出 tee 指令。我們可以使用鍵盤輸入資料，並同時將資料顯示，而且輸入到 foo 的檔案中。

```
[root@flash good]# tee foo
這是使用tee指令來作轉向和管線
這是使用tee指令來作轉向和管線
我們按下<Ctrl+D>就可以跳出我們按下<Ctrl+D>就可以跳出

[root@flash good]# more foo
這是使用tee指令來作轉向和管線
我們按下<Ctrl+D>就可以跳出
```

我們可以使用指令 >& 檔案，轉向指令的輸出或轉向指令的錯誤到檔案中。

語法：

指令：指令　>&　檔案

當在我們目錄上沒有 good 檔案，我們使用 ls –l good >& log，這會將標準錯誤輸出到 log 中。

```
[root@flash good]# ls -l good >& log
[root@flash good]# more log
ls: good: 沒有此一檔案或目錄
```

我們使用 ls　–l　>&　log，這會將標準輸出輸出到 log 中。

```
[root@flash good]# ls -l >& log
[root@flash good]# more log
total 372
drwxr-xr-x    2 root     root         4096  9月  4 13:40 best
-rw-r--r--    1 root     root            1  9月  9 14:52 file
-rw-r--r--    1 root     root           57  9月  9 15:03 foo
-rw-r--r--    1 root     root           58  9月  9 14:58 goo
-rw-r--r--    1 root     root            0  9月  9 15:22 log
```

15-7 檢查並修護檔案系統

我們可以使用 fsck 指令來檢查並修正檔案系統的錯誤

語法：

指令：fsck 參數 <檔案系統類型> [檔案系統]

參數：

-a：自動修護檔案系統。

-A：檢查在/etc/fstab 內的所有檔案系統。

-N：不執行修護檔案系統但列出執行修護的動作。

-P：搭配-A 參數同步執行檢查所有檔案系統。

-r：以交談式的方式修護檔案系統。

-R：搭配-A 參數檢查檔案系統時略過根目錄。

-s：依序執行檔案系統。

-t<檔案串列>：指定要檢查的檔案系統類型。

-T：在開始時不顯示標題。

-V：指令執行過程。

我們使用 fsck –As 以循序的方式自動檢查及修護在/etc/fstab 所列的檔案系統。這是我們最常使用修護檔案系統的方式，而且最有效。

```
[root@flash root]# fsck -As
fsck 1.27 (8-Mar-2002)
e2fsck 1.27 (8-Mar-2002)
/dev/hda1 is mounted.

WARNING!!!  Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.

Do you really want to continue (y/n)? yes

/ was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Deleted inode 228997 has zero dtime.  Fix<y>? yes

Inode 310745, i_blocks is 64, should be 8.  Fix<y>? yes

Inode 327199, i_blocks is 64, should be 48.  Fix<y>? yes
```

我們使用 df 來查詢檔案系統，再使用 fsck –a /dev/hda1 來檢查整個檔案系統。

```
[root@flash chaiyen]# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda1             12096724    226204  11256036   2% /
/dev/hda3              6048352    216820   5524292   4% /home
none                    257148         0    257148   0% /dev/shm
/dev/hda5              6048320   3282576   2458504  58% /usr
aasir.com:/home/chaiyen
                      12096728   3546448   7935792  31% /home/people
```

我們使用 fsck –Ap 以同步方式自動檢查及修護在/etc/fstab 所列的檔案系統。

```
[root@flash root]# fsck -Ap
fsck 1.27 (8-Mar-2002)
/dev/hda1 is mounted.

WARNING!!!  Running e2fsck on a mounted filesystem may c
ause
SEVERE filesystem damage.

Do you really want to continue (y/n)? yes

/ was not cleanly unmounted, check forced.
/: Deleted inode 228997 has zero dtime.  FIXED.
/: Inode 310745, i_blocks is 64, should be 8.  FIXED.
```

我們使用 fsck –V /dev/hda1 來作檢查及修護檔案執行過程

```
[root@flash root]# fsck -V /dev/hda1
fsck 1.27 (8-Mar-2002)
[/sbin/fsck.ext2 (1) -- /] fsck.ext2 /dev/hda1
e2fsck 1.27 (8-Mar-2002)
/dev/hda1 is mounted.

WARNING!!!  Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.

Do you really want to continue (y/n)? yes

/ contains a file system with errors, check forced.
Pass 1: Checking inodes, blocks, and sizes
Inode 310745, i_blocks is 64, should be 8.  Fix<y>? yes

Inode 327210, i_blocks is 1160, should be 1144.  Fix<y>? yes

Inode 327212, i_blocks is 2120, should be 2064.  Fix<y>? yes

Inode 327215, i_blocks is 1256, should be 1240.  Fix<y>? yes

Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Unattached inode 147442
Connect to /lost+found<y>? yes
```

15-8 在 Linux 上新增硬碟

我們可以使用 dmesg 指令來搜詢 grep 和磁碟 hd 相關的裝置資訊。had 為我們第一顆 QUANTUM 磁碟。Hdb 為我們光碟機，RW-241040 為我們的 CD/DVD-ROM 光碟機。hdc 為我們第二顆硬碟，其為 Maxtor。dmesg 為顯示或控制核心資訊的指令。

```
[root@mandrake /]# dmesg|grep hd|more
Kernel command line: ro root=/dev/md2 hdd=ide-scsi
ide_setup: hdd=ide-scsi
    ide0: BM-DMA at 0xd400-0xd407, BIOS settings: hda:DMA, hdb:DMA
    ide1: BM-DMA at 0xd408-0xd40f, BIOS settings: hdc:DMA, hdd:pio
hda: QUANTUM FIREBALLP AS40.0, ATA DISK drive
hdb: RW-241040, ATAPI CD/DVD-ROM drive
hdc: Maxtor 5T030H3, ATA DISK drive
hda: host protected area => 1
hda: 78177792 sectors (40027 MB) w/1902KiB Cache, CHS=4866/255/63, UDMA(100)
hdc: host protected area => 1
hdc: 60030432 sectors (30736 MB) w/2048KiB Cache, CHS=59554/16/63, UDMA(33)
 hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 >
 hdc: hdc1 hdc2
 hdc1: <bsd: hdc5 hdc6 hdc7 hdc8 hdc9 >
 hdc2: <bsd: hdc10 >
```

我們使用 fdisk 指令來分割我們第二顆磁碟。/sbin/fdisk /dev/hdc 為分割我們 /dev/hdc 的裝置。我們使用 m 來觀看選項。

/sbin/fdisk /dev/hdc

```
Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

先刪除磁碟的舊資料，我們使用 d 的指令。

```
Command (m for help): d
Partition number (1-4): 1
```

這是列印出我們的磁碟資訊，這裏顯示有 30GB 的硬碟。

```
Command (m for help): p

Disk /dev/hdc: 30.7 GB, 30735581184 bytes
16 heads, 63 sectors/track, 59554 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
```

我們使用 n 的指令來新增加分割，我們選取為主要分割 p，並且要分割 3000M。我們使用+3000M 來表示要分割的大小。

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-59554, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-59554, default 59554): +3000M
```

這時後我們使用 p 來顯示磁碟資訊，/dev/hdc1 裝置，其有 3000M，而其系統為 Linux。

```
Command (m for help): p

Disk /dev/hdc: 30.7 GB, 30735581184 bytes
16 heads, 63 sectors/track, 59554 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdc1               1        5814     2930224+  83  Linux
```

我們將資料寫到磁碟，並且離開，我們使用 w 指令。

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.
```

我們現在使用 mke2fs –j 來在/dev/hdc1 的裝置上建立 ext3 的檔案系統。

```
[root@mandrake /]# /sbin/mke2fs -j /dev/hdc1
mke2fs 1.32 (09-Nov-2002)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
366528 inodes, 732556 blocks
36627 blocks (5.00%) reserved for the super user
First data block=0
23 block groups
32768 blocks per group, 32768 fragments per group
15936 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 34 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

這時我們使用 df 來觀看目前的檔案系統，因為我們新增的檔案系統還未掛載 mount 上去，所以我們將掛載/dev/hdc1 的裝置。

```
[root@mandrake /]# df
檔案系統                1K-區段      已用      可用 已用% 掛載點
/dev/md2               3020048    720696   2145940   26% /
/dev/md0                147692      9284    130783    7% /boot
none                    256916         0    256916    0% /dev/shm
```

我們要將我們的/dev/hdc1 磁碟裝置掛載到特定目錄上，所以我們新增要掛載的目錄 newdisk。

mkdir /newdisk

我們使用 mount 指令來將/dev/hdc1 裝置上的 ext3 檔案系統掛載到/newdisk 目錄上。

mount -t ext3 /dev/hdc1 /newdisk

```
[root@mandrake /]# df
檔案系統                1K-區段      已用      可用 已用% 掛載點
/dev/md2              3020048    720700   2145936   26% /
/dev/md0               147692      9284    130783    7% /boot
none                   256916         0    256916    0% /dev/shm
/dev/hdc1             2884176     32828   2704840    2% /newdisk
```

我們也可以編輯/etc/fstab 讓它在開機時自動掛載我們/dev/hdc1 的裝置。我們在/etc/fstab 最後一行新增/dev/hdc1 /newdisk ext3 defaults 1 1，來開機自動掛載/dev/hdc1 的裝置。

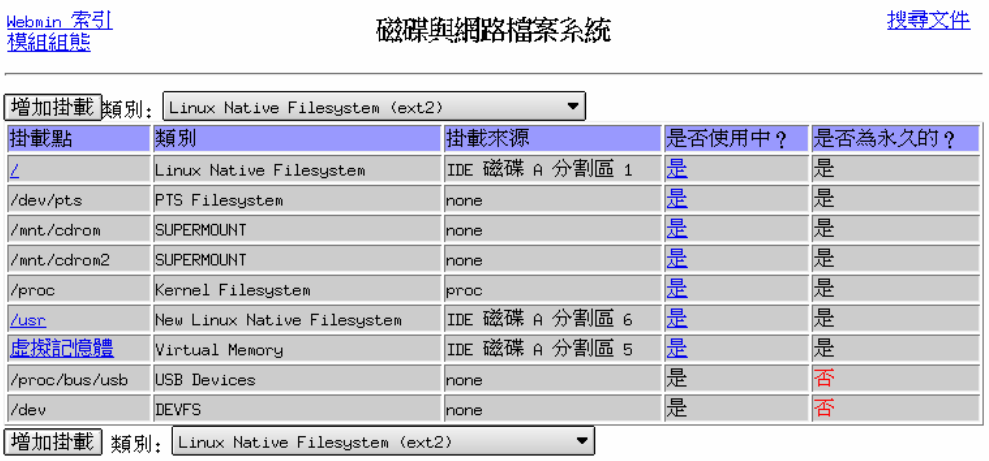
vi /etc/fstab

```
/dev/md2                /                       ext3    defaults        1 1
/dev/md0                /boot                   ext3    defaults        1 2
none                    /dev/pts                devpts  gid=5,mode=620  0 0
none                    /proc                   proc    defaults        0 0
none                    /dev/shm                tmpfs   defaults        0 0
/dev/md1                swap                    swap    defaults        0 0
/dev/cdrom              /mnt/cdrom              udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/hdc1               /newdisk                ext3    defaults        1 1
```

15-9 使用 Webmin 管理檔案系統

我們選取系統→磁碟與網路檔案系統。

這就是我們的磁碟與檔案系統。



NOTE