

第五章 進階檔案處理



學習目標

- 管理與建立檔案與目錄
- 執行檔案
- 硬碟空間指令操作
- 能進行系統輸出入指令應用
- 搜尋特定檔案
- 過濾複製或多餘的輸出資料
- 使用vim編輯文字檔案

章節大綱

5-1 檔案系統

5-4 搜尋檔案

5-2 硬碟空間使用情形

5-5 篩選內容

5-3 指令的輸出入

5-6 文字編輯器vim

備註：可依進度點選小節

進階檔案處理

- 系統管理工作中和檔案處理有很大的關係
- 檔案的一致性
- 建立連結
- 執行例行性的指令與工具
- 監控磁碟空間的使用情形
- 搜尋特定檔案

3

5-1 檔案系統

- CentOS的檔案系統使用的是最新的「Ext4」檔案系統
- Ext4前身是Ext3，Ext3的前身是Ext2檔案系統
- Ext3是以Ext2為基礎再加上「日誌功能（journaling）」
- 節省了傳統需掃描整顆硬碟所耗費的時間

4

5-1 檔案系統

● Ext4檔案系統

- 以ext3為基礎再加上許多提昇檔案效能的新技術
 - 相容於Ext3
 - 突破檔案系統的大小限制
 - ext4使用了「Extent file writing」範圍性寫入檔案的新技術，減少檔案在寫入磁碟時的游離區塊的程度

5

5-1 檔案系統

● 檔案資訊記錄檔inode

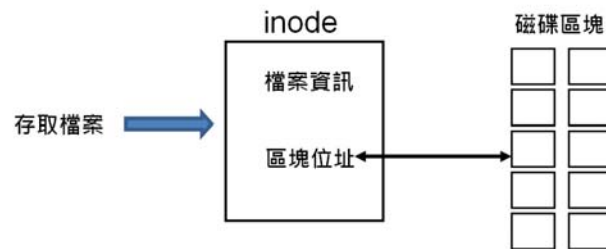
- 記載著這個檔案的所有資訊
- 包括檔名、更動時間、權限與檔案儲存的區塊位置等資料
- 每個inode都有唯一的編號
- `ls -li`可以列出檔案的inode編號


```
[root@proj ch5]# touch file1 file2
[root@proj ch5]# ls -li
1178618 file1 1178619 file2
```

6

5-1 檔案系統

● 檔案資訊記錄檔inode



7

5-1 檔案系統

● 連結Link

- 連結允許多個檔案參考到一個檔案
- 連結是一種指向另一個檔案的特別檔案
- 如Windows的捷徑
- 連結「指向」真實存在的檔案或目錄
- 連結的用途：
 - 不需要多複製一份檔案至其他位置，節省空間。
 - 保持連結指向檔案都是最近資料。

8

5-1 檔案系統

● 連結Link

● 建立符號連結

- 利用ln指令加上選項-s可建立符號連結

- 語法：

- ln -s 目標對象 連結名稱

```
[root@centos ~]# echo "hello" > data
[root@centos ~]# ls -l data
-rw-r--r--. 1 root root 6 Mar 29 17:07 data
[root@centos ~]# ln -s data slink
[root@centos ~]# ls -l
-rw-r--r--. 1 root root 6 Mar 29 17:07 data
lrwxrwxrwx. 1 root root 4 Mar 29 17:07 slink -> data
```

9

5-1 檔案系統

● 連結Link

● 建立硬連結

- 硬連結(hard link)建立的方法也是使用ln指令，不需加選項

- 語法：

- ln 目標對象 連結名稱

- 與符號連結不一樣的是，當真實檔案被刪除或移至其他目錄時，硬連結仍可正常得到檔案的內容，不會變成無效連結

10

5-1 檔案系統

● 連結Link

● 建立硬連結時的限制

- 硬連結不可跨越分割區
- 硬連結不可使用於目錄

11

5-1 檔案系統

● Linux的執行檔

- 沒有特別規定副檔名
- 只要檔案具有可執行(Executable)權限即可執行
- 兩種執行檔案的方式
 - 以絕對路徑執行
 - /usr/bin/ddate
 - /bin/netstat
 - 執行目前所在目錄下的執行檔
 - 先以./開始，代表「目前目錄下」
 - ./runme

12

5-1 檔案系統

● Linux的執行檔

● 為什麼不直接打檔名就執行目前目錄下的執行檔呢？

- 保護系統安全
- 如果有一個惡意的使用者在他的家目錄下新增一個執行檔，執行檔裏面是一些破壞系統的指令
- 管理者如果在該使用者的家目錄中打了ls指令，那就很容易掉入陷阱中

13

5-2 硬碟空間使用情形

● df指令

```
[root@mylinux ~]# df
```

檔案系統	1K- 區段	已用	可用	已用 %	掛載點
/dev/sda3	2016044	722064	1191568	38%	/
/dev/sda6	194442	13190	171213	8%	/home
/dev/sda2	4032124	2458392	1368904	65%	/usr
/dev/sda1	194442	11840	172563	7%	/boot
tmpfs	257532	0	257532	0%	/dev/shm

● 加入選項-h讓顯示的單位以KB、MB、GB來表示

14

5-2 硬碟空間使用情形

● 目錄下已使用的檔案空間

● du指令

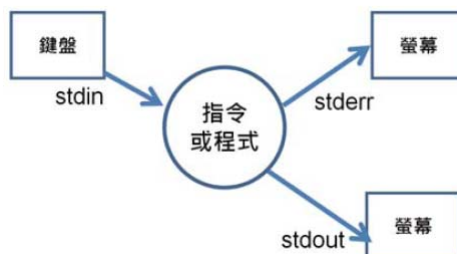
- 想知道目前目錄下（含子目錄）已用空間

```
[root@mylinux ~]# du -h
- 省略 -
16K  ./includes/Cache/Lite
64K  ./includes/Cache
2.9M ./includes
8.0K ./fw
40K  ./ncftp
36K  ./help/css
748K ./help
130M .
```

15

5-3 指令的輸出入

- 輸出設備一般有螢幕、印表機、檔案等
- 輸入設備有鍵盤、讀卡機、掃描器與檔案等
- 「標準輸入」、「標準輸出」與「標準錯誤訊息輸出」



16

5-3 指令的輸出入

● 重導輸出

- 將原本會輸出到螢幕上的文字（stdout 或stderr），改為輸出到檔案成為文字檔
- 使用 >（大於符號，重導）
- 將df指令的結果，重導至myfile檔案
 - df > myfile
- ls /abcde 2> myerror （將錯誤訊息輸出至檔案儲存）

17

5-3 指令的輸出入

● 管線

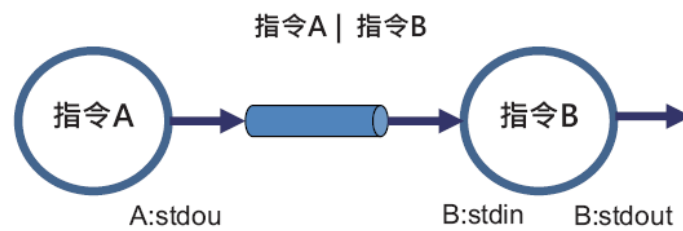
- 單一指令無法做到想要完成的工作
- 可以使用Linux的管線功能「|」
- 結合二個以上的指令，處理較複雜的工作
- ls -l指令輸出很多頁內容，more指令又可以將輸入的資料分頁輸出，結合ls -l的輸出與more指令功能的指令如下：
 - ls -l /usr/lib/ |more

18

5-3 指令的輸出入

● 管線

- 「接收左邊指令的輸出，將資料接至右邊指令，成為其輸入來源」



19

5-4 搜尋檔案

● which

- which來搜尋環境變數PATH裏的所有目錄中是否有特定執行檔
- 只會搜尋PATH變數的目錄
- 知道要搜尋檔案的完整檔名，則用whereis搜尋

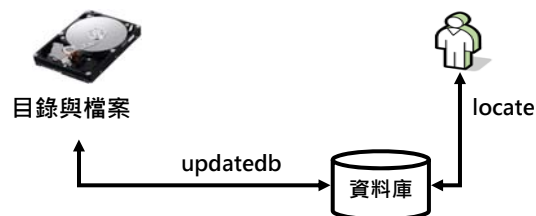
20

5-4 搜尋檔案

● 查詢部份檔名 locate

- 用 locate 檔名來查詢檔案位置
- 查詢部份指令 netst :

```
whereis netst
/usr/bin/netstat
```
- locate向系統內建的資料庫去查詢而得到結果
- 以updatedb指令更新資料庫



21

5-4 搜尋檔案

● 搜尋特定檔案 find

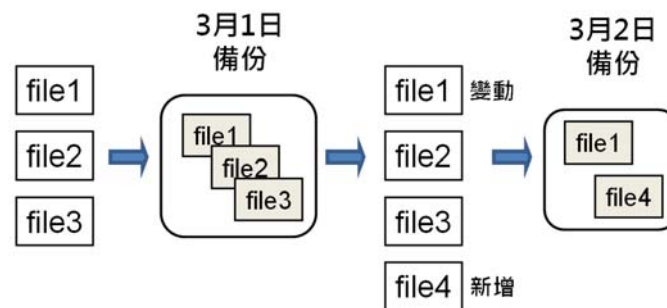
- 直接在檔案系統內搜尋特定檔案
- find的用法是在指令後面空白後加上搜尋的起始目錄，再加上想要查詢的條件
- 想要從/home以下的子目錄，搜尋檔名為install的檔案
 - `find /home -name install`
- 想要從/var/spool/下，搜尋檔案名稱稱為jack的檔案：
 - `find /var/spool/ -name jack`

22

5-4 搜尋檔案

● 增量備份的實務應用

- 只有在初次備份時，備份目錄內的所有檔案
- 爾後備份時，只需處理自上次備份時間之後有變動或新增的檔案



23

5-4 搜尋檔案

● 增量備份的實務應用

- 初次備份（3月1日）的指令：
 - `tar cvfz backup-0301.tar.gz /root/testdir/*`
 - `touch /root/testdir/timebase`
- 增量備份
 - `tar cvfz backup-0302.tar.gz 'find /root/testdir/ -cnewer /root/testdir/timebase -type f'` → 錯誤做法
 - `find /root/testdir/ -cnewer /root/testdir/timebase -type f | tar cvfz backup-0302.tar.gz -T -`
 - `touch /root/testdir/timebase`

24

5-5 篩選內容

- 遇到某個檔案或指令輸出內容很多時
- 需要篩選符合特定資料的結果
- 只顯示我們想要的資料
- `grep` 特定文字 檔案(對象)
 - `grep sam install.log`
 - `grep sam *`

25

5-6 文字編輯器vim

- 所有UNIX與Linux都內建的文字編輯器
- 只要學好vim，在任何Linux都能夠利用它來編輯文字檔案

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

VIM - Vi Improved

版本 7.2.411

維護者：Bram Moolenaar et al.
修改者為 <bugzilla@redhat.com>

Vim 為可自由散佈的開放原始碼軟體

請幫助烏干達的可憐孩童！

進一步說明請輸入 :help iccf<Enter>

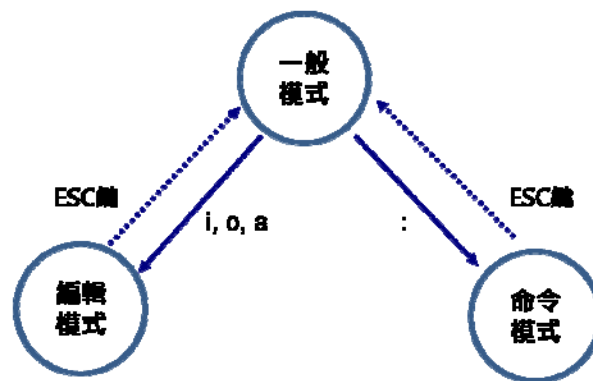
要離開請輸入 :q<Enter>
線上說明請輸入 :help<Enter>

type :help version7<Enter> for version info

26

5-6 文字編輯器vim

● 重要的三個模式



27

5-6 文字編輯器vim

● 一般模式 (Common mode)

- 可以使用具有特殊意義的按鍵
- 在文件內容中移動、刪除、複製、貼上等功能
- 在一般模式中，按下i、o或a鍵可以進入編輯模式
- 按下冒號「:」則會進入命令模式

● 編輯模式 (Insert mode)

- 按下內容會新增在文件游標所在處
- 像打字一樣，按鍵不再具有特殊意義
- 按下[ESC]鍵可回到一般模式

28

5-6 文字編輯器vim

● 命令模式 (Command line mode)

- 可以儲存、開啟檔案、結束vi、取代特定文字等檔案處理功能
- 按下ESC鍵可回到一般模式
- 觀察左下方是否出現「:」冒號

29

5-6 文字編輯器vim

● 一般模式下的功能按鍵

- 瀏覽文件
- 利用以下按鍵，我們可以在vi內瀏覽文件內容：
- h：游標向左移，與[左鍵]的功能相同。
- j：游標向下移，與[下鍵]的功能相同。
- k：游標向上移，與[上鍵]的功能相同。
- l：游標向右移，與[右鍵]的功能相同。
- PageUp：游標向上一頁
- PageDown：游標向下一頁
- G：游標移到文件最後一行。
- gg：游標移到文件第一行。

30

5-6 文字編輯器vim

● 一般模式下的功能按鍵

- Ctrl+g：顯示游標目前位置的資訊，包括第幾行，第幾個字元，位於文件中的百分比。
- 顯示行號：按下冒號後，打入set nu
- 取消顯示行號：按下冒號後，打入set nonu
- 複製游標所在該行
- p：將先前複製的內容，貼在游標所在下行，原文字會往下移
- dd：刪除游標所在行，原該行下方資料往上移。
- u：復原上次做的更動
- .：重覆上一個動作

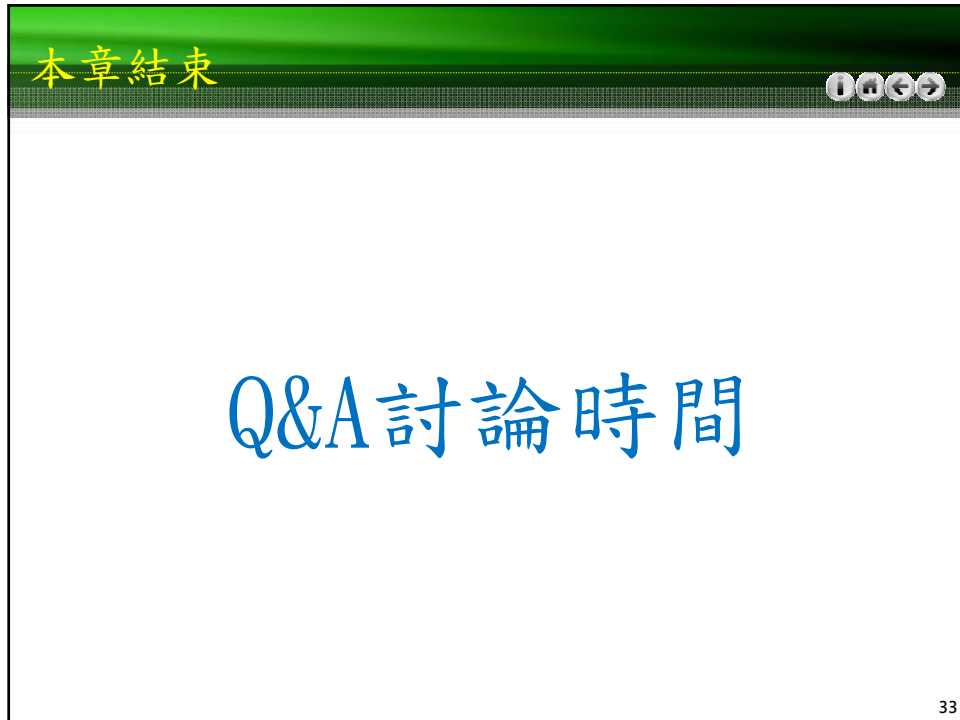
31

5-6 文字編輯器vim

● 命令模式功能

- 開啟另一個檔案 :o 檔案名稱
- 儲存目前內容 :w
- 將目前編輯內容另存為其他檔案 :w 檔名
- 離開q
- 不存檔離開q!

32



本章結束

Q&A討論時間

33