

3

檔案管理與 vim 初探



前一堂課讀者應該稍微接觸 Linux 檔案的管理，這一堂課我們將較深入的操作 Linux 檔案管理。此外，本堂課亦會介紹未來會一直使用的 vim 程式編輯器，學會 vim 對於系統管理員來說，是相當重要的任務！

3.1 檔案管理

在 Linux 底下，所有的東西都以檔案來呈現，不同的檔案特性會有不同的結果。讀者可以常見的兩種檔案格式為：

- ◆ 一般檔案：實際放置資料的檔案。
- ◆ 目錄檔案：重點在放置『檔名』，並沒有實際的資料。

為何需要目錄檔？讀者可以想像，如果僅有一個櫃子，你將所有書籍全部丟進同一個櫃子中，則未來要找資料時，會很難找尋 (因為單品太多)。若可以有許多個櫃子，將不同的資料分類放置於各別的櫃子中，未來要找某一類別的資料，只要找到該類別的櫃子，就能夠快速的找到資料 (單品較少)，這就是目錄檔案的重點。

3.1.1 目錄的建立與刪除

前一堂課已經談過，目錄的建立主要使用 `mkdir` 這個指令，這個指令將建立一個『空目錄』。所謂的『空目錄』意指該目錄內並沒有其他檔案的存在。至於刪除目錄則使用 `rmdir` 這個指令，但同理，這個指令僅能『刪除空目錄』而已。



例題

1. 前往 `/dev/shm` 目錄。
2. 建立名為 `class3` 的目錄。
3. 觀察 `/dev/shm/class3` 這個目錄的內容，並請說明內部有沒有其他檔案 (註：使用 `ll` 加上顯示隱藏檔的選項)。
4. 透過 `cp /etc/hosts /dev/shm/class3` 將檔案複製到該目錄內，並觀察 `class3` 目錄的內容。
5. 使用 `rmdir /dev/shm/class3` 嘗試刪除該目錄，並說明可以或不行刪除該目錄的原因。

使用 `rm` 可以刪除檔案，但預設 `rm` 僅能刪除一般檔案無法刪除目錄。



例題

1. 承上一個例題，進入到 `/dev/shm/class3` 當中，並且使用 `rm` 刪除掉所有該目錄下的檔案 (非隱藏檔)。
2. 回到 `/dev/shm` 當中，此時能否使用 `rmdir` 刪除 `class3` 目錄？為什麼？

觀察目錄本身的參數

當使用 `ll dirname` 時，預設會顯示出『該目錄下的檔名』，因為目錄的內容就是檔名資料。若讀者需要了解目錄本身的資訊，而不是目錄的內容，可以使用 `-d` 的選項，如下範例：

```
[student@localhost ~]$ ll /etc/cron.d
總計 12
-rw-r--r--. 1 root root 128  7月 27  2015 0hourly
-rw-r--r--. 1 root root 108  9月 18  2015 raid-check
-rw-----. 1 root root 235  3月  6  2015 sysstat

[student@localhost ~]$ ll -d /etc/cron.d
drwxr-xr-x. 2 root root 51  2月 18 02:58 /etc/cron.d
```

承上，讀者可以清楚的看到有沒有加上 `-d` 的選項結果差異相當大。

3.1.2 萬用字元

要查詢某些關鍵字的資訊時，需要透過一些終端機環境下的特殊字元的支援，此即為萬用字元。經常使用的萬用字元有：

符號	意義
*	代表『0個到無窮多個』任意字元
?	代表『一定有一個』任意字元
[]	同樣代表『一定有一個在括號內』的字元(非任意字元)。例如 <code>[abcd]</code> 代表『一定有一個字元，可能是 a, b, c, d 這四個任何一個』
[-]	若有減號在中括號內時，代表『在編碼順序內的所有字元』。例如 <code>[0-9]</code> 代表 0 到 9 之間的所有數字，因為數字的語系編碼是連續的

符號	意義
[^]	若中括號內的第一個字元為指數符號 (^)，那表示『反向選擇』，例如 [^abc] 代表 一定有一個字元，只要是非 a, b, c 的其他字元就接受的意思

若讀者想要了解 /etc 底下有多少檔名開頭為 cron 的檔案時，可以使用如下的方式查詢：

```
[student@localhost ~]$ ll /etc/cron*
[student@localhost ~]$ ll -d /etc/cron*
```

如果加上 -d 的選項，則檔名會變得比較單純，但若沒有加上 -d 的選項，則 ll 會列出『許多目錄內的檔名資料』，與預設想要了解的檔名有所差異。因此 -d 選項就顯的更為重要。



例題

1. 列出 /etc/ 底下含有 5 個字元的檔名。
2. 列出 /etc/ 底下含有數字在內的檔名。

3.1.3 檔案及目錄的複製與刪除

前一堂課也稍微介紹過複製，複製主要使用 cp 來處理，相關的選項請自行 man cp 來查詢。預設 cp 僅複製檔案，並不會複製目錄，若需要複製目錄，一般建議直接加上 -r，而如果是需要完整備份，則最好加上 -a 的選項為宜。

另外，除了正常的一般檔案與目錄檔案之外，系統也經常會有連結檔的情況出現，例如底下的資料：

```
[student@localhost ~]$ ll -d /etc/rc0.d /etc/rc.d/rc0.d
lrwxrwxrwx. 1 root root 10 2月 18 02:54 /etc/rc0.d -> rc.d/rc0.d
drwxr-xr-x. 2 root root 43 2月 18 02:56 /etc/rc.d/rc0.d
```

連結檔的特色是，該行開頭的 10 個字元最左邊為 l (link)，一般檔案為減號 (-) 而目錄檔為 d (directory)。如上表所示，其實 /etc/rc0.d 與 /etc/rc.d/rc0.d 是相同的資料，其中 /etc/rc0.d 是連結檔，而原始檔為 /etc/rc.d/rc0.d。此時讀者需要注意，亦即當你進入 /etc/rc0.d 時，代表實際進入了 /etc/rc.d/rc0.d 那個目錄的意思。

複製目錄時

一般來說，複製目錄需要加上 -r 或 -a，兩者的差異如下：

```
[student@localhost ~]$ cd /dev/shm
[student@localhost shm]$ cp -r /etc/rc0.d/ .      <==結尾一定要加上斜線 /
[student@localhost shm]$ ll
drwxr-xr-x. 2 student student      80  5月 19 10:56 rc0.d

[student@localhost shm]$ ll rc0.d /etc/rc0.d/
/etc/rc0.d/:
總計 0
lrwxrwxrwx. 1 root root 20   2月 18 02:56 K50netconsole -> ../init.d/netconsole
lrwxrwxrwx. 1 root root 17   2月 18 02:56 K90network -> ../init.d/network

rc0.d:
總計 0
lrwxrwxrwx. 1 student student 20   5月 19 10:56 K50netconsole -> ../init.d/netconsole
lrwxrwxrwx. 1 student student 17   5月 19 10:56 K90network -> ../init.d/network

[student@localhost shm]$ cp -a /etc/rc0.d/ rc0.d2
[student@localhost shm]$ ll rc0.d2
lrwxrwxrwx. 1 student student 20   2月 18 02:56 K50netconsole -> ../init.d/netconsole
lrwxrwxrwx. 1 student student 17   2月 18 02:56 K90network -> ../init.d/network
```

讀者可以發現 -a 時，連同檔案的時間也都複製過來，而不是使用目前的時間來建立新的檔案。此外，如果以 root 的身份來執行上述指令時，則連同權限 (前面的 root 變成 student) 也會跟原始檔案相同！這就是 -r 與 -a 的差異。因此，當系統備份時，還是建議使用 -a 的。

目標檔案的存在與否

參考底下的範例：



例題

1. 先進入 `/dev/shm`，同時觀察目錄下有無名為 `rc1.d` 的檔名。
2. 使用『`cp -r /etc/rc.d/rc1.d rc1.d`』將 `rc1.d` 複製到本目錄下，然後使用 `ll` 與 `ll rc1.d` 觀察該目錄。
3. 重新執行上述複製的指令一次，然後使用 `ll rc1.d`，觀察一下有什麼變化？

當複製目錄，且目標為未存在的目錄，則系統會建立一個同名的目錄名稱來存放資料。但若目標檔案已存在，則原始目錄將會被放置到目標檔案內，因此目標目錄是否存在，會影響到複製的結果。

刪除檔案

刪除檔案使用 `rm`，其中需要特別注意，不要隨便使用 `rm -rf` 這樣的選項，因為 `-r` 為刪除目錄，`-f` 為不詢問直接刪除，因此若後續的檔名寫錯時，將會有相當大的影響（一般來說，檔案刪除是無法救援回來的。）



例題

1. 進入 `/dev/shm`，觀察到前一個例題 `/dev/shm/rc1.d` 的目錄存在後，請將它刪除。

3.1.4 特殊檔名的處理方式

在 windows 底下經常會有比較特別的檔名出現，最常出現者為檔名含空白字元的情況。由於指令操作行為下，空白鍵亦為特殊字元，因此操作上需要將這些特殊字元改為一般字元後，方可進行處理。常見的處理方式有這些情況：

空白字元的檔名

一般可以使用單引號或雙引號或反斜線 (\) 來處理這樣的檔名。例如建立一個名為『class one』的檔名時，可以這樣做：

```
[student@localhost ~]$ cd /dev/shm
[student@localhost shm]$ mkdir "class one"
[student@localhost shm]$ ll
drwxrwxr-x. 2 student student    40  5月 19 11:23 class one
```

讀者可以發現最右邊出現了 class one 的檔名，但這個檔名要如何刪除呢？

```
[student@localhost shm]$ rmdir class one
rmdir: failed to remove 'class': 沒有此一檔案或目錄
rmdir: failed to remove 'one': 沒有此一檔案或目錄

[student@localhost shm]$ rmdir class\ one
```

如果僅單純的補上檔名，則 rmdir 會誤判有兩個名為 class 與 one 的目錄要刪除，因為找不到，所以回報錯誤。此時你可以使用成對雙引號或單引號來處理，也可以透過反斜線將空白變成一般字元即可 (其實透過按下 [tab] 按鈕也可以找到上述的方式來刪除！)

加號與減號開頭的檔名

讀者應該知道指令下達時，在指令後的選項為開頭是 + 或 - 的項目，如果檔名被要求建成立成 -newdir 時，該如何處理？

```
[student@localhost shm]$ mkdir -newdir
mkdir: 不適用的選項 -- n
Try 'mkdir --help' for more information.
```

此時會回報錯誤，若嘗試使用單引號來處理時，同樣回報錯誤！使用反斜線，同樣回報錯誤。是否無法建立此類檔名呢？其實讀者可以透過『絕對/相對路徑』的作法來處理，例如：

```
[student@localhost shm]$ mkdir /dev/shm/-newdir
[student@localhost shm]$ mkdir ./-newdir2
[student@localhost shm]$ ll -d ./-*new*
```

```
drwxrwxr-x. 2 student student 40 5月 19 11:32 ./-newdir
drwxrwxr-x. 2 student student 40 5月 19 11:32 ./-newdir2
```

這樣就可以建立開頭為 + 或 - 的檔名。刪除同樣得要使用這樣的檔名撰寫方式來處理。



例題

1. 將剛剛建立的 -newdir, -newdir2 刪除。

3.1.5 隱藏檔的觀察與檔案類型的觀察

觀察隱藏檔案

要觀察隱藏檔時，可以使用如下的方式來處理：

```
[student@localhost shm]$ cd
[student@localhost ~]$ ll
drwxr-xr-x. 2 student student 6 4月 14 19:46 下載
drwxr-xr-x. 2 student student 6 4月 14 19:46 公共
drwxr-xr-x. 2 student student 6 4月 14 19:46 圖片
.....

[student@localhost ~]$ ll -a
drwx-----. 16 student student 4096 5月 12 11:31 .
drwxr-xr-x. 17 root root 4096 5月 3 21:43 ..
-rw-----. 1 student student 4202 5月 18 17:43 .bash_history
-rw-r--r--. 1 student student 18 11月 20 13:02 .bash_logout
-rw-r--r--. 1 student student 193 11月 20 13:02 .bash_profile
.....

[student@localhost ~]$ ll -d .*
```

由於隱藏檔是檔名開頭為小數點的檔名，因此可以透過 -a 來查詢所有的檔案，或者是透過 .* 來找隱藏檔而已。不過得要加上 -d 的選項才行。

觀察檔案的類型

但如果需要觀察檔案的類型與型態，就需要使用 `file` 這個指令來觀察。例如分別找出 `/etc/passwd` 即 `/usr/bin/passwd` 這兩個檔案的格式為何？

```
[student@localhost ~]$ ll /etc/passwd /usr/bin/passwd
-rw-r--r--. 1 root root 2945  5月  3 21:43 /etc/passwd
-rwsr-xr-x. 1 root root 27832  6月 10  2014 /usr/bin/passwd

[student@localhost ~]$ file /etc/passwd /usr/bin/passwd
/etc/passwd:      ASCII text
/usr/bin/passwd: setuid ELF 64-bit LSB shared object, x86-64, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.6.32, BuildID[sha1]=
1e5735bf7b317e60bcb907f1989951f6abd50e8d, stripped
```

讀者即可知道這兩個檔案分別是文字檔 (ASCII text) 及執行檔 (ELF 64-bit LSB...)。



例題

1. 觀察 `/etc/rc0.d` 及 `/etc/rc.d/rc0.d` 的檔案類型為何？

3.1.6 檔名的移動與更名

若檔案建立到錯誤的位置時，可以使用 `mv` 來處理。同時若檔名鍵錯，也能夠使用 `mv` 來更名。



例題

1. 讓 student 回到家目錄。
2. 將 `/etc/rc3.d` 複製到本目錄。
3. 該目錄移動錯誤，請將本目錄的 `rc3.d` 移動到 `/dev/shm`。
4. 檔名依舊錯誤，請將 `/dev/shm` 底下的 `rc3.d` 更名為 `init3.d`。

3.1.7 大量建置空白檔案的方式

有時候為了測試系統，管理員可能需要建立許多的檔名來測試，此時可以透過 `touch` 這個指令來處理。例如到 `/dev/shm` 建立名為 `testdir` 與 `testfile` 兩個『目錄檔與一般檔』，可以這樣處理。

```
[student@localhost ~]$ cd /dev/shm
[student@localhost shm]$ mkdir testdir
[student@localhost shm]$ touch testfile
[student@localhost shm]$ ll -d test*
drwxrwxr-x. 2 student student 40  5月 19 13:04 testdir
-rw-rw-r--. 1 student student  0  5月 19 13:04 testfile
```

如果需要建立較多的檔名，例如 `test1`, `test2`, `test3`, `test4` 時，可以透過大括號的方式來處理。例如在 `/dev/shm` 底下建立上述的四個檔案，可以這樣處理：

```
[student@localhost shm]$ touch test{1,2,3,4}
[student@localhost shm]$ ll -d test?
-rw-rw-r--. 1 student student 0  5月 19 13:06 test1
-rw-rw-r--. 1 student student 0  5月 19 13:06 test2
-rw-rw-r--. 1 student student 0  5月 19 13:06 test3
-rw-rw-r--. 1 student student 0  5月 19 13:06 test4
```

如果所需要的檔名或輸出資訊是有用到連續數字時，假設由 1 到 10 這組數字，雖然能使用 `{1,2,3,4,5,6,7,8,9,10}` 來處理，然而輸入太繁瑣。此時可以使用 `{1..10}` 來取代上述的輸出。若需要輸出 `01`, `02` 這樣的字樣 `{01..10}` 來處理。



例題

- ◆ 我需要在 `/dev/shm/testing` 目錄下建立名為 `mytest_XX_YY_ZZ` 的檔案，其中 `XX` 為 `jan`, `feb`, `mar`, `apr` 四個資料，`YY` 為 `one`, `two`, `three` 三個資料，而 `ZZ` 為 `a1`, `b1`, `c1` 三個資料，如何使用一個指令就建立出上述的 36 個檔案？

- ◆ 我需要在 `/dev/shm/student/` 目錄下，建立檔名為 `4070C001` 到 `4070C050` 的檔案，如何使用一個指令來完成這 50 個檔案的建置？
-

3.2 檔案內容的查詢

很多時候管理員只是需要知道檔案內容，並沒有進行編輯。此時可以透過一些簡易的指令來查詢文件檔案的內容。

3.2.1 連續輸出檔案內容

最簡單的查詢檔案內容的方式為透過 `cat`、`head` 與 `tail` 等指令。`cat` 為較常用的指令，但是 `cat` 會將檔案完整的重現在螢幕上，因此若管理員想要查詢最後幾行時，以 `tail` 指令查詢會較佳。

- ◆ `cat`：將檔案內容全部列出。
- ◆ `head`：預設只列出檔案最前面 10 行。
- ◆ `tail`：預設只列出檔案最後面 10 行。



1. 列出 `/etc/hosts` 檔案的內容。
 2. 列出 `/etc/profile` 檔案的內容。
 3. 承上，第二次列出 `/etc/profile` 時加上行號輸出。
 4. 讀者僅須列出 `/etc/profile` 的最前面 10 行。
 5. 讀者僅須列出 `/etc/passwd` 最後面 10 行的內容。
 6. 讀者僅須列出 `/etc/services` 最後 5 行的內容。
-

3.2.2 可檢索檔案內容

上述的 `cat/head/tail` 需要查詢資料時，得要人工眼力查詢，因此，如果資料量比較大，而且需要查詢資訊時，可以透過 `more` 與 `less` 來處置。`more`

預設會一頁一頁向後翻動，而 less 則可以向前、向後翻頁，事實上，man page 就是呼叫 less 指令的函數處理的方式。

more 軟體內常用指令：

- ◆ /關鍵字：可以查詢關鍵字。
- ◆ 空白鍵：可以向下/向後翻頁。
- ◆ q：結束離開不再查詢文件。

less 軟體內常用指令：

- ◆ /關鍵字：可以查詢關鍵字。
- ◆ 空白鍵：可以向下/向後翻頁。
- ◆ [pageup]：可以向前/向上翻頁。
- ◆ [pagedown]：可以向下/向後翻頁。
- ◆ g：直接來到第一行。
- ◆ G：直接來到最後一行。
- ◆ q：結束離開不再查詢文件。



1. 使用 more /etc/services 一頁一頁翻動資料。
2. 承上，請找出 http 這個關鍵字，之後直接離開不再查閱。
3. 使用 less /etc/services 查詢檔案內容。
4. 承上，請找出 http 這個關鍵字，之後直接離開不再查閱。

若需要查詢資料的行號時，可以透過 cat -n 配合管線命令來處理。例如，先將 /etc/services 的輸出加上行號，然後交由 less 處理，再去搜尋 http 所在行，要執行這個指令則為：

```
[student@localhost ~]$ cat -n /etc/services | less
```

關於管線命令的使用，後續的章節會談論更多，在此讀者僅須知道在管線 (|) 之前所輸出的資訊，會傳給管線後的指令繼續讀入處理的意思。亦即訊息資料並不是來自檔案，而是來自於前一個檔案的輸出。

3.3 vim 程式編輯器

管理員總是會需要變動系統設定檔，或者是進行純文字檔的編輯，此時就需要 vi/vim 的支援。因為 vi/vim 是 Linux 很多指令預設會去呼叫的編輯程式，因此管理員『務必』要學會這個編輯器。另外，vim 會有顏色的支援，vi 僅為文書編輯器，故我們建議讀者們，應該要熟悉 vim 較佳。

3.3.1 簡易的 vim 操作

vim 有三種基本的模式，亦即是：

- ◆ 一般指令模式 (command mode)：使用『vim filename』進入 vim 之後，最先接觸到的模式。在這個模式底下使用者可以進行複製、刪除、貼上、移動游標、復原等任務。
- ◆ 編輯模式 (insert mode)：在上述模式底下輸入『i』這個按鈕，就可以進入編輯模式，終於可以開始打字了。
- ◆ 指令列命令模式 (command-line mode)：回到一般模式後，可以進行儲存、離開、強制離開等動作。

簡單的說，讀者可以將三種模式使用底下的圖示來思考一下相關性：

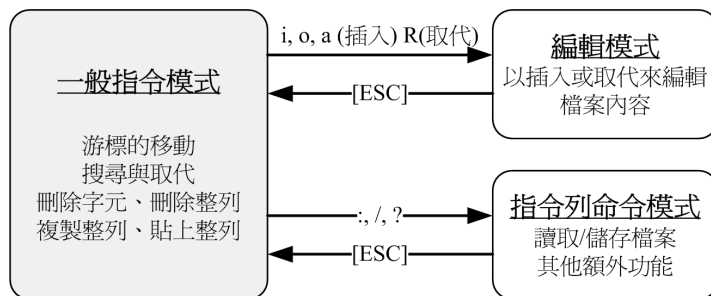


圖 3.3-1 vi 三種模式的相互關係

假設讀者想要嘗試編輯 `/etc/services` 這個檔案，可以這樣嘗試處理看看：

1. 使用『`cd /dev/shm`』將工作目錄移動到記憶體當中。
2. 使用『`cp /etc/services .`』將檔案複製一份到本目錄下。
3. 使用『`vim services`』開始查閱 `services` 的內容，並請回答：
 - a. 最底下一行顯示的『`"services" 11176L, 670293C`』是什麼意思？
 - b. 為什麼在 `#` 之後的文字顏色與沒有 `#` 的那行不太一樣？
4. 使用方向鍵，移動游標到第 100 行，並回答，你怎麼知道游標已經在第 100 行？
5. 回到第 5 行，按下『`i`』之後，你看到畫面中最底下一行的左邊出現什麼？
6. 按下 `Enter` 按鈕以新增一行，然後方向鍵重回第 5 行，之後隨便輸入一串文字。輸入完畢後，直接按下 `[esc]` 按鈕，螢幕最下方左下角會有什麼變化？
7. 要離開時，記得關鍵字是 `quit`，此時請按下『`:q`』，看一下游標跑到什麼地方去了？
8. 在輸入『`:q`』並且按下 `Enter` 之後，螢幕最下方出現什麼資訊？怎麼會這樣？
9. 按下『`:w`』儲存，然後重新『`:q`』離開，這時可以離開了嗎？



例題

1. 承上，繼續 `vim services`。
2. 請在第 1 行加上『`Welcome to my linux server`』的字樣，輸入完畢後請回到一般指令模式。
3. 在一般指令模式底下，跑到第 5 行，按下『`dd`』，看看發生什麼事情？
4. 回到第 1 行，按下『`p`』又出現什麼資訊？
5. 連續按下 5 次『`p`』，然後又按一次『`5p`』又出現什麼？
6. 按下『`u`』會出現什麼狀況？

7. 跑到第 1 行，按下『yy』然後跑到第 10 行，按下『p』，又出現什麼情況？
8. 按下『G』(注意大小寫)，游標跑到哪裡去？
9. 按下『gg』(注意大小寫)，游標跑到哪裡去？
10. 現在不想要編輯這個檔案了 (因為亂搞一通)，不儲存離開時，按下『:q』會一直出現僅告尚未存檔的訊息，你輸入『:q!』之後，可以不儲存離開了嗎？

3.3.2 常用的 vim 一般指令模式與指令列模式列表

通過上述的練習，讀者應該會對 vim 有初步的認識。vim 的功能其實不只這些，不過管理員會經常用到的大概就是上述的這些資訊而已。底下為常用的指令列表：

慣用的指令	說明
i, [esc]	i 為進入編輯模式，[esc] 為離開編輯模式
G	移動到這個檔案的最後一列
gg	移動到這個檔案的第一列
dd	dd 為刪除游標所在行，5dd 為刪除 5 行，n dd 為刪除 n 行
yy	yy 為複製游標所在行，5yy 為複製 5 行，nyy 為複製 n 行
p	在游標底下貼上剛剛刪除/複製的資料
u	復原前一個動作
:w	將目前的資料寫入硬碟中
:q	離開 vim
:q!	不儲存 (強制) 離開 vim

讀者大概之要知道這幾個按鈕即可，其他的更進階的功能有用到的時候再查閱《鳥哥的 Linux 私房菜-基礎學習篇》即可。



例題

讀者經常有需要紀錄自己輸入資料的習慣，可以使用 `history` 來輸出歷史命令。那該如何紀錄有效的資訊？

1. 使用 `student` 的身份輸入 `history`，查閱這次有效的歷史命令有幾個，假設有 50 筆新的指令好了。
2. 使用『`history 50 >> ~/history.log`』將指令紀錄到 `history.log` 檔案中。
3. 使用『`vim ~/history.log`』編輯該檔案，將無效的指令移除，只剩下需要的檔案，同時在指令後說明該指令的用途。

3.4 課後練習操作

前置動作：請使用 `unit03` 的硬碟進入作業環境，並請先以 `root` 身分執行 `vbird_book_setup_ip` 指令設定好你的學號與 IP 之後，再開始底下的作業練習。

- ◆ 實作題：請使用 `student` 的身份進行如下實作的任務。直接在系統上面操作，操作成功即可，上傳結果的程式會主動找到你的實作結果。
 1. 使用 `vim` 建立一個名為 `~/myname.txt` 的檔案，內容填寫你的學號與姓名，並在找出底下的任務後，寫下正確答案：
 - a. 請先實際找出你系統中的 `/etc/passwd`, `/etc/pam.d` `/etc/rc.local`, `/dev/vda` 這 4 個檔案的『檔案類型 (如一般檔案、目錄檔案等等)』後，將這 4 個的類型寫入 `~/myname.txt` 檔案中。
 - b. 找出 `/usr/lib64` 這個目錄內，有個檔名長度為 5 個字元的一般檔案，將該檔案檔名寫入 `~/myname.txt` 中。
 - c. 找出 `/etc` 底下，檔名含有 4 個數字 (數字不一定連接在一起) 的資料，寫下 (1)該檔案的『絕對路徑』檔名，(2)該檔案的類型。
 2. 在 `/opt` 底下，有個以減號 (-) 為開頭的檔名，該檔案做錯了，因此，請將它刪除。(可能需要 `root` 的權限)

3. 在 student 家目錄下，新增一個名為 class03 的目錄，並進入該目錄成為工作目錄。並且完成底下的工作：
 - a. 在當前的目錄下，新建 mytest_XX_YY_ZZ.txt，其中 XX 為『class1, class2, class3』，而 YY 為『week1, week2, week3』，ZZ 則為『one, two, three, four』。
 - b. 建立一個名為 class1/week2 的目錄，將當前目錄中，含有 class1_week2 的檔名通通『複製』到 class1/week2 目錄下。
 - c. 將檔名含有 class1 檔案，通通『移動』到 class1 目錄下。
 - d. 新建一個名為 one 的目錄，將當前目錄中，所有檔名含有 one 的檔案通通移動到 one 底下。
 - e. 建立一個名為 others 的目錄，將當前檔名開頭為 mytest 的檔案，通通移動到該目錄下。
4. 在 student 家目錄下，建立一個 userid 的子目錄，並將工作目錄移動到 userid 內，在 userid 這個目錄內，嘗試以一個指令建立 ksuid001 ksuid002 ... 到 ksuid020 等 20 個『空白目錄』。
5. 回到 student 家目錄，並且完成底下的任務：
 - a. 在 student 家目錄底下，建立一個名為 -myhome 的目錄，並將 student 家目錄中，以 b 為開頭的『隱藏檔』複製到 -myhome 目錄內。
 - b. 將工作目錄移動到 -myhome 目錄內，並將 /etc/sysconfig 目錄複製到當前目錄下 (會有一些錯誤訊息是正確的！)。
 - c. 將當前目錄下的 sysconfig/cbq 目錄刪除。
 - d. 列出 /etc/profile 與 /etc/services 的最後 5 行，並將這 10 行轉存到當前目錄下的 myetc.txt 檔案中。
 - e. 將 myetc.txt 複製成為 myetc2.txt，並使用 vim 編輯 myetc2.txt，第一行加入『I can use vim』的字樣即可。

6. 在 student 家目錄下，有個名為 mytext.txt 的檔案，請使用 vim 開啟該檔案，並將第一行複製後，貼上 100 次，之後『強制儲存』即可離開 vim。

作業結果傳輸：請以 root 的身分執行 vbird_book_check_unit 指令上傳作業結果。正常執行完畢的結果應會出現【XXXXXX;aa:bb:cc:dd:ee:ff;unitNN】字樣。若需要查閱自己上傳資料的時間，請在作業系統上面使用：<http://192.168.251.250> 檢查相對應的課程檔案。

