

05.Linux VIM 編輯器

為何要學習Vim?

- 所有的 Unix Like 系統都會內建 vi 文書編輯器，其他的文書編輯器則不一定會存在，很多個別軟體的編輯介面都會主動呼叫 vi
- vim 具有程式編輯的能力，可以主動的以字體顏色辨別語法的正確性，方便程式設計；因為程式簡單，編輯速度相當快速。

綜合以上，同學對於使用webmin使用者介面駕輕就熟，但回過頭來如果要真的深入了解LINUX的奧妙還是必須從指令下手，身為一個技術人員腦袋中沒裝個數十個常用的指令是無法讓能體會到你的專業能力的，以下所列舉出來的vim編輯例子請大家一定要熟記。

- 基本上 vi 共分為三種模式，分別是『一般模式』、『編輯模式』與『指令列命令模式』。這三種模式的作用分別是：

一般模式：

以 vi 打開一個檔案就直接進入一般模式了(這是預設的模式)。在這個模式中，你可以使用『上下左右』按鍵來移動游標，你可以使用『刪除字元』或『刪除整行』來處理檔案內容，也可以使用『複製、貼上』來處理你的文件資料。

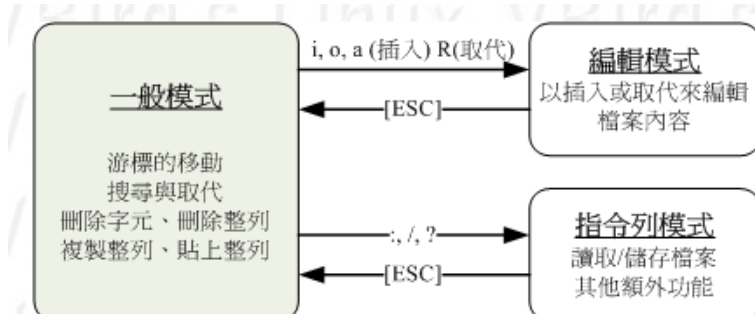
編輯模式：

在一般模式中進行刪除、複製、貼上等等的動作，但是卻無法編輯文件內容的！要等到你按下『i, I, o, O, a, A, r, R』等任何一個字母之後才會進入編輯模式。注意了！通常在 Linux 中，按下這些按鍵時，在畫面的左下方會出現『INSERT 或 REPLACE』的字樣，此時才可以進行編輯。而如果要回到一般模式時，則必須要按下『Esc』這個按鍵即可退出編輯模式。

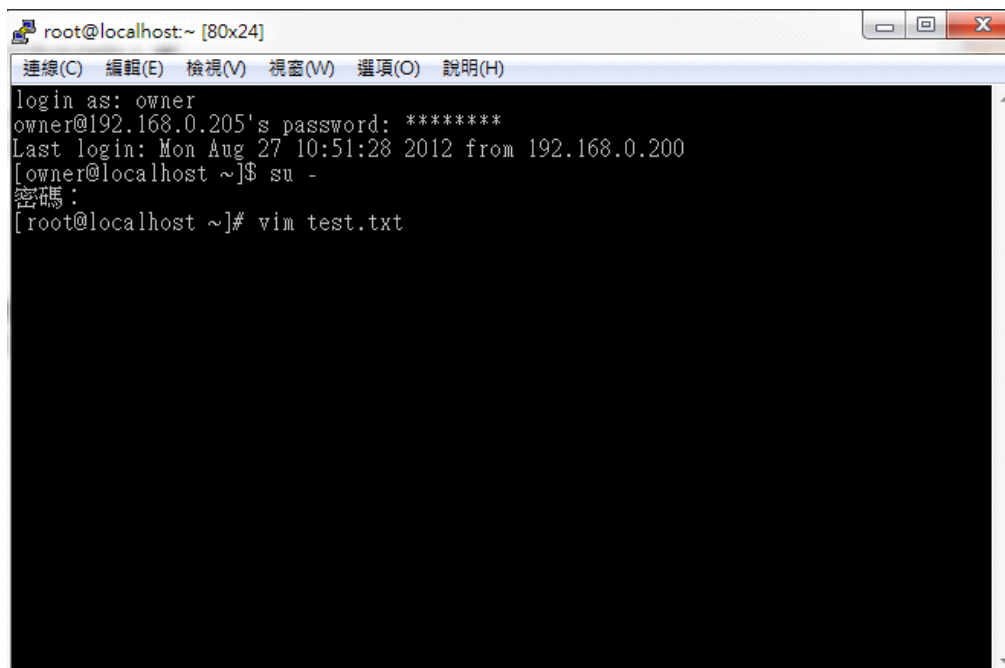
指令列命令模式：

在一般模式當中，輸入『:/?』三個中的任何一個按鈕，就可以將游標移動到最底下那一行。在這個模式當中，可以提供你『搜尋資料』的動作，而讀取、存檔、大量取代字元、離開 vi、顯示行號等等的動作則是在此模式中達成的！

可由下圖來表示vim的指令切換模式需熟記



- 簡單執行示範
開檔 test.txt



A terminal window titled 'root@localhost:~ [80x24]' with a menu bar containing '連線(C)', '編輯(E)', '檢視(V)', '視窗(W)', '選項(O)', and '說明(H)'. The terminal output shows a login sequence: 'login as: owner', 'owner@192.168.0.205's password: *****', 'Last login: Mon Aug 27 10:51:28 2012 from 192.168.0.200', and '[owner@localhost ~]\$ su -'. It then prompts for a password ('密碼:') and shows the root prompt '[root@localhost ~]# vim test.txt'.

```
root@localhost:~ [80x24]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)
login as: owner
owner@192.168.0.205's password: *****
Last login: Mon Aug 27 10:51:28 2012 from 192.168.0.200
[owner@localhost ~]$ su -
密碼:
[root@localhost ~]# vim test.txt
```



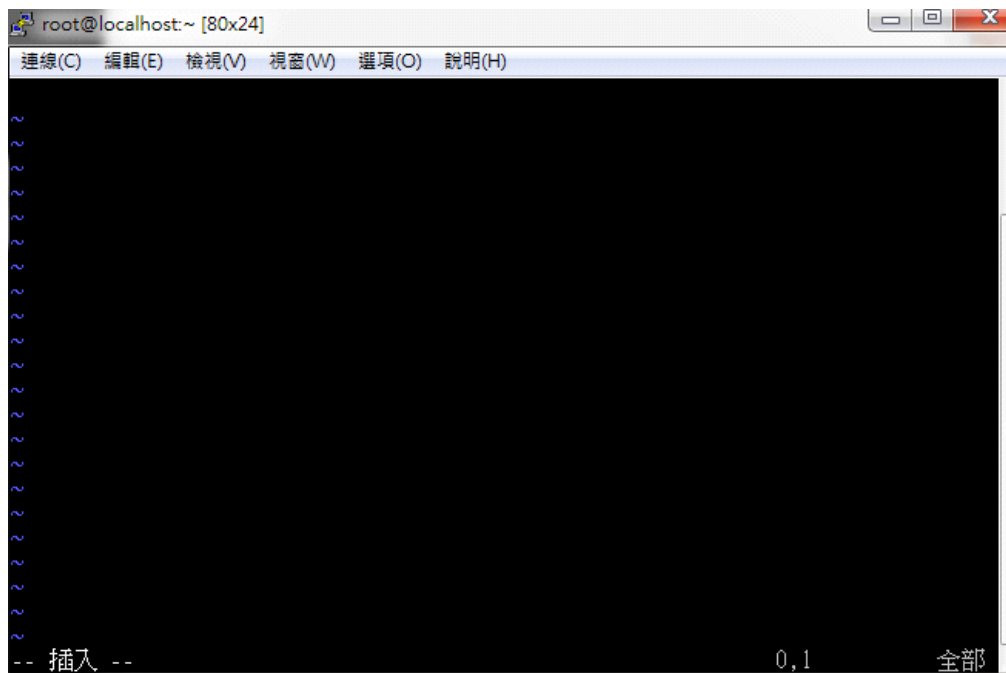
A Vim editor window titled 'root@localhost:~ [80x24]' with the same menu bar. The editor is in normal mode, showing a file named 'test.txt' with 1 line and 0 characters. The status bar at the bottom indicates '"test.txt" [未命名]' and '0,0-1 全部'.

```
root@localhost:~ [80x24]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

"test.txt" [未命名] 0,0-1 全部
```

輸入字元 切換指令 I

此時下面的提示字元變成"插入" 表示為可輸入狀態



root@localhost:~ [80x24]

連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

-- 插入 -- 0.1 全部



root@localhost:~ [80x24]

連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

i am richie

-- 插入 -- 1,12 全部

存檔 :wq 觀察檔案

```

root@localhost:~ [80x24]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)
login as: owner
owner@192.168.0.205's password: *****
Last login: Mon Aug 27 10:51:28 2012 from 192.168.0.200
[owner@localhost ~]$ su -
密碼:
[root@localhost ~]# vim test.txt
[root@localhost ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog  test.txt
[root@localhost ~]#

```

3. 按鍵說明

第一部份：一般模式可用的按鈕說明，游標移動、複製貼上、搜尋取代等

h 或 向左方向鍵(←)	游標向左移動一個字元
j 或 向下方向鍵(↓)	游標向下移動一個字元
k 或 向上方向鍵(↑)	游標向上移動一個字元
l 或 向右方向鍵(→)	游標向右移動一個字元
如果你將右手放在鍵盤上的話，你會發現 hjkl 是排列在一起的，因此可以使用這四個按鈕來移動游標。如果想要進行多次移動的話，例如向下移動 30 行，可以使用 "30j" 或 "30↓" 的組合按鍵，亦即加上想要進行的次數(數字)後，按下動作即可！	
[Ctrl] + [f]	螢幕『向下』移動一頁，相當於 [Page Down]按鍵 (常用)
[Ctrl] + [b]	螢幕『向上』移動一頁，相當於 [Page Up] 按鍵 (常用)
[Ctrl] + [d]	螢幕『向下』移動半頁
[Ctrl] + [u]	螢幕『向上』移動半頁
+	游標移動到非空白字元的下一列
-	游標移動到非空白字元的上一列
n<space>	那個 n 表示『數字』，例如 20 。按下數字後再按空白鍵，游標會向右移動這一行的 n 個字元。例如 20<space> 則游標會向後面移動 20 個字元距離。
0 或功能鍵[Home]	這是數字『0』：移動到這一行的最前面字元處 (常用)
\$ 或功能鍵[End]	移動到這一行的最後面字元處(常用)
H	游標移動到這個螢幕的最上方那一行的第一個字元
M	游標移動到這個螢幕的中央那一行的第一個字元
L	游標移動到這個螢幕的最下方那一行的第一個字元
G	移動到這個檔案的最後一行(常用)
nG	n 為數字。移動到這個檔案的第 n 行。例如 20G 則會移動到這個檔案的第 20 行(可配合 :set nu)
gg	移動到這個檔案的第一行，相當於 1G 啊！ (常用)
n<Enter>	n 為數字。游標向下移動 n 行(常用)

搜尋與取代	
/word	向游標之下尋找一個名為 word 的字串。例如要在檔案內搜尋 vbird 這個字串，就輸入 /vbird 即可！(常用)
?word	向游標之上尋找一個字串名為 word 的字串。
n	這個 n 是英文按鍵。代表『重複前一個搜尋的動作』。舉例來說，如果剛剛我們執行 /vbird 去向下搜尋 vbird 這個字串，則按下 n 後，會向下繼續搜尋下一個名為 vbird 的字串。如果是執行 ?vbird 的話，那麼按下 n 則會向上繼續搜尋名為 vbird 的字串！
N	這個 N 是英文按鍵。與 n 剛好相反，為『反向』進行前一個搜尋動作。例如 /vbird 後，按下 N 則表示『向上』搜尋 vbird。
使用 /word 配合 n 及 N 是非常有幫助的！可以讓你重複的找到一些你搜尋的關鍵字！	
:n1,n2s/word1/word2/g	n1 與 n2 為數字。在第 n1 與 n2 行之間尋找 word1 這個字串，並將該字串取代為 word2！舉例來說，在 100 到 200 行之間搜尋 vbird 並取代為 VBIRD 則： 『:100,200s/vbird/VBIRD/g』。(常用)
:1,\$s/word1/word2/g	從第一行到最後一行尋找 word1 字串，並將該字串取代為 word2！(常用)
:1,\$s/word1/word2/gc	從第一行到最後一行尋找 word1 字串，並將該字串取代為 word2！且在取代前顯示提示字元給使用者確認(confirm) 是否需要取代！(常用)
刪除、複製與貼上	
x, X	在一行字當中，x 為向後刪除一個字元(相當於 [del] 按鍵)，X 為向前刪除一個字元(相當於 [backspace] 亦即是倒退鍵)(常用)
nx	n 為數字，連續向後刪除 n 個字元。舉例來說，我要連續刪除 10 個字元，『10x』。
dd	刪除游標所在的那一整列(常用)
ndd	n 為數字。刪除游標所在的向下 n 列，例如 20dd 則是刪除 20 列(常用)
d1G	刪除游標所在到第一行的所有資料
dG	刪除游標所在到最後一行的所有資料
d\$	刪除游標所在處，到該行的最後一個字元
d0	那個是數字的 0，刪除游標所在處，到該行的最前面一個字元
yy	複製游標所在的那一行(常用)
nyy	n 為數字。複製游標所在的向下 n 列，例如 20yy 則是複製 20 列(常用)
y1G	複製游標所在列到第一列的所有資料
yG	複製游標所在列到最後一列的所有資料
y0	複製游標所在的那個字元到該行行首的所有資料
y\$	複製游標所在的那個字元到該行行尾的所有資料
p, P	p 為將已複製的資料在游標下一行貼上，P 則為貼在游標上一行！舉例來說，我目前游標在第 20 行，且已經複製了 10 行資料。則按下 p 後，那 10 行資料會貼在原本的 20 行之後，亦即由 21 行開始貼。但如果是按下 P 呢？那麼原本的

	第 20 行會被推到變成 30 行。(常用)
J	將游標所在列與下一列的資料結合成同一列
c	重複刪除多個資料，例如向下刪除 10 行，[10cj]
u	復原前一個動作。(常用)
[Ctrl]+r	重做上一個動作。(常用)
這個 u 與 [Ctrl]+r 是很常用的指令！一個是復原，另一個則是重做一次～ 利用這兩個功能按鍵，你的編輯，嘿嘿！很快樂的啦！	
.	不要懷疑！這就是小數點！意思是重複前一個動作的意思。如果你想要重複刪除、重複貼上等等動作，按下小數點『.』就好了！(常用)

第二部份：一般模式切換到編輯模式的可用的按鈕說明

i, I	進入插入模式(Insert mode)： i 為『從目前游標所在處插入』，I 為『在目前所在行的第一個非空白字元處開始插入』。(常用)
a, A	進入插入模式(Insert mode)： a 為『從目前游標所在的下一個字元處開始插入』，A 為『從游標所在行的最後一個字元處開始插入』。(常用)
o, O	進入插入模式(Insert mode)： 這是英文字母 o 的大小寫。o 為『在目前游標所在的下一行處插入新的一行』；O 為在目前游標所在處的上一行插入新的一行！(常用)
r, R	進入取代模式(Replace mode)： r 只會取代游標所在的那一個字元一次；R 會一直取代游標所在的文字，直到按下 ESC 為止；(常用)
[Esc]	退出編輯模式，回到一般模式中(常用)

第三部份：一般模式切換到指令列模式的可用的按鈕說明

指令列的儲存、離開等指令

:w	將編輯的資料寫入硬碟檔案中(常用)
:w!	若檔案屬性為『唯讀』時，強制寫入該檔案。不過，到底能不能寫入，還是跟你對該檔案的檔案權限有關啊！
:q	離開 vi (常用)
:q!	若曾修改過檔案，又不想儲存，使用！為強制離開不儲存檔案。
驚嘆號 (!)	常常具有『強制』的意思～
:wq	儲存後離開，若為 :wq! 則為強制儲存後離開 (常用)
ZZ	這是大寫的 Z 喔！若檔案沒有更動，則不儲存離開，若檔案已經被更動過，則儲存後離開！
:w [filename]	將編輯的資料儲存成另一個檔案（類似另存新檔）
:r [filename]	在編輯的資料中，讀入另一個檔案的資料。亦即將『filename』這個檔案內容加到游標所在行後面
:n1,n2 w [filename]	將 n1 到 n2 的內容儲存成 filename 這個檔案。
:! command	暫時離開 vi 到指令列模式下執行 command 的顯示結果！例如『:! ls /home』即可在 vi 當中察看 /home 底下以 ls 輸出的檔案資訊！

vim 環境的變更

:set nu	顯示行號，設定之後，會在每一行的字首顯示該行的行號
:set nonu	與 set nu 相反，為取消行號！

4. 範例練習 參照第4點的表格練習下面例子

1. 在/tmp目錄下建立一個名為vimtest目錄
2. 進入vimtest這個目錄中
3. 將/etc/manpath.config複製到本目錄底下
4. 使用vim開啟本目錄下的/manpath.config檔案
5. 設定行號
6. 移動到第67行，向右移動26個字元，請問看到的目錄為何？
7. 移動到第一行，並往下搜尋 **adjacent** 這個字串，請問在第幾行？
8. 將第125行到131行註解刪除
9. 刪除後反悔了，想要復原有什麼方法？
10. 到102行並刪除10個字元，結果出現的第一個單字為？
11. 在第一行新增一行，該行內容輸入「I am a student...」
12. 儲存後離開

1. mkdir /tmp/vimtest
2. cp /tmp/vimtest
3. cp /etc/manpath.config .
4. vim manpath.config
5. :set nu
6. 先輸入67G 再輸入 26->，可看到路徑為 /var/cache/man
7. 先輸入1G 或 gg，再輸入/adjacent，會出現在109行
8. 先輸入125G再輸入7dd
9. 輸入u 或者不存檔離開 :q!
10. 先輸入102G再輸入10x，會出現單字 names
11. 先輸入1G到第一行，再輸入O為新增一行插入模式
12. :wq

1. 『mkdir /tmp/vitest』
2. 『cd /tmp/vitest』
3. 『cp /etc/man.config .』
4. 『vi man.config』
5. 『:set nu』然後你會在畫面中看到左側出現數字即為行號。
6. 先按下『58G』再按下『40→』會看到『/dir/bin/foo』這個字樣在雙引號內；
7. 先執行『1G』或『gg』後，直接輸入『/bzip2』，則會去到第 118 行才對！
8. 直接下達『:50,100s/man/MAN/gc』即可！若一直按『y』最終會出現『在 23 行內置換 25 個字串』的說明。
9. (1)簡單的方法可以一直按『u』回復到原始狀態，(2)使用不儲存離開『:q!』之後，再重新讀取一次該檔案；
10. 『65G』然後再『9yy』之後最後一行會出現『複製九行』之類的說明字樣。按下『G』到最後一行，再給他『p』貼上九行！
11. 因為 21~42 22 行，因此『21G』→『22dd』就能刪除 22 行，此時你會發現游標所在 21 行的地方變成 MANPATH 開頭囉，註解的 # 符號那幾行都被刪除了。
12. 『:w man.test.config』，你會發現最後一行出現 "man.test.config" [New].. 的字樣。
13. 『27G』之後，再給他『15x』即可刪除 15 個字元，出現『you』的字樣；
14. 先『1G』去到第一行，然後按下大寫的『O』便新增一行且在插入模式；開始輸入『I am a student...』後，按下[Esc]回到一般模式等待後續工作；
15. 『:wq』

1. 請在 /tmp 這個目錄下建立一個名為 vitest 的目錄；
 2. 進入 vitest 這個目錄當中；
 3. 將 /etc/man.config 複製到本目錄底下(或由上述的連結下載 [man.config](#) 檔案)；
 4. 使用 vi 開啟本目錄下的 man.config 這個檔案；
 5. 在 vi 中設定一下行號；
 6. 移動到第 58 行，向右移動 40 個字元，請問你看到的雙引號內是什麼目錄？
 7. 移動到第一行，並且向下搜尋一下『bzip2』這個字串，請問他在第幾行？
 8. 接著下來，我要將 50 到 100 行之間的『小寫 man 字串』改為『大寫 MAN 字串』，並且一個一個挑選是否需要修改，如何下達指令？如果在挑選過程中一直按『y』，結果會在最後一行出現改變了幾個 man 呢？
 9. 修改完之後，突然反悔了，要全部復原，有哪些方法？
 10. 我要複製 65 到 73 這九行的內容(含有MANPATH_MAP)，並且貼到最後一行之後；
 11. 21 到 42 行之間的開頭為 # 符號的註解資料我不要了，要如何刪除？
 12. 將這個檔案另存成一個 man.test.config 的檔名；
 13. 去到第 27 行，並且刪除 15 個字元，結果出現的第一個單字是什麼？
 14. 在第一行新增一行，該行內容輸入『I am a student...』；
 15. 儲存後離開吧！
- Cp

以上動作熟悉即可很容易的運用vim來編輯文件

以上教材 引用參考鳥哥的私房菜

