

## 第四章 文字編輯工具 - vi

### 4-1 編輯工具簡介

文字編輯工具是指簡單的文書編輯，大多使用於編寫程式、修改系統設定等等，並非文書排版工具(如 word)。這類型的編輯工具不需要特殊顯示器，但也僅能編輯文字，並無法處理聲音、影像等多媒體文件。許多 Unix/Linux 主機都是透過網路，於遠端從事環境建立與系統管理的工作，一般僅利用簡單的文字處理器即可，目前最普遍使用的文字編輯工具為 vi。vi 編輯器是一套全螢幕的文字編輯工具，它僅能編輯文字，無法接受影像、聲音等多媒體資料，因此利用一般文字終端機(或網路終端機，如 putty)即可。因此，一般透過網路登入主機系統或管理系統時，大多仰賴 vi 工具來編輯各項設定檔，這也是本書介紹 vi 編輯器的主要原因。

早期 vi 是由 Bill Joy 所撰寫，並在柏克萊 Unix 上執行。目前 vi 幾乎成為 Unix 系統的標準配備，也是操作 Unix 必備的工具之一，而 Linux 所採用的是 vi 加強版 vim (vi Improved)。雖然 vim 捨棄許多 vi 繁複的命令，但它保留或加強的命令也幾乎與原來的 vi 相容，就連最基本的命令執行還是保留使用 vi 命令。

### 4-2 vi/vim 編輯工具

#### 4-2-1 vi/vim 操作模式

使用 vi/vim 須注意，它是全螢幕的字元編輯工具，但不使用『滑鼠』，也沒有『下拉式』( Drop-down ) 的視窗操作。它有兩個操作『模式』( Mode ) 來達成命令控制與文字輸入的功能，分別是『命令模式』( Command Mode ) 與『輸入模式』( Input Mode )。由名稱上可以看出，在『命令模式』下使用者可以下達某些編輯控制命令，其中又可區分為『單鍵命令』

與『文字命令』；另一方面，在『輸入模式』下使用者可以直接輸入所欲編輯的文字，兩種模式之間轉換如圖 5-1 所示。使用者在一般終端機上輸入『\$ vi file\_1』，表示呼叫 vi 程式，並編輯 file\_1 檔案，其中 file\_1 可以是已存在或新建立的檔案，系統會開啟 vi 環境並進入『單鍵命令模式』。

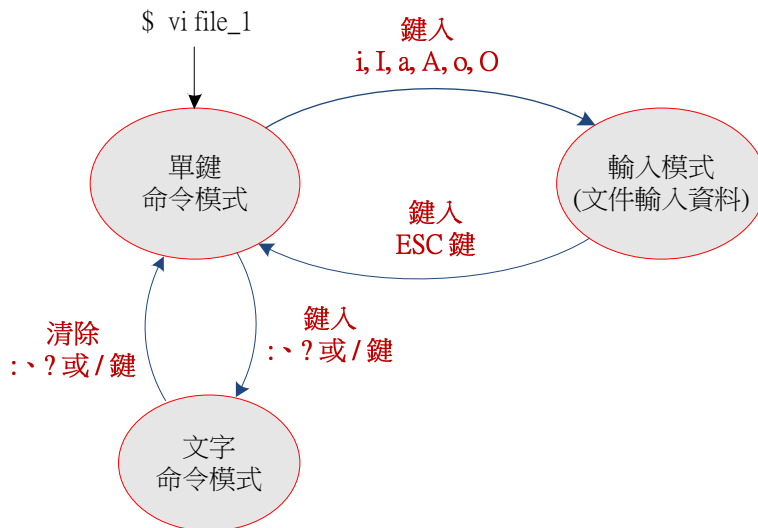


圖 4-1 vi 模式的切換

### 4-2-2 單鍵命令模式

在『單鍵命令』模式下可以直接敲入某些單鍵來達成控制的功能。其中輸入 i、I、a、A、o 與 O 鍵，除了達成該鍵之目的外，還會直接轉換到『輸入模式』，說明如下：

- i：插入文字到游標右邊，並進入『輸入模式』。
- I：插入文字到游標左邊，並進入『輸入模式』。
- a：附加文字到游標右邊，並進入『輸入模式』。
- A：附加文字到該行尾端，並進入『輸入模式』。
- o：目前行的下方開始一行，並進入『輸入模式』。
- O：目前行的上方開始一行，並進入『輸入模式』。
- x：刪除游標上文字並存入緩衝器，另 5x，刪除 5 個字元。
- dd：刪除游標上該行並存入緩衝器，另 2dd 刪除 2 行。

- yy：將游標所在整行複製到緩衝區內，另 2 yy 複製 2 行。
- p：將緩衝區內資料複製到游標所在位置。

### 4-2-3 輸入模式

在『輸入模式』之下，可以直接輸入文字，並利用『backspace』或『delete』鍵刪除字元，也可利用『↑、↓、←、→』鍵來移動游標。重要的是，直接敲入『ESC』鍵，就可以轉換到『單鍵命令模式』，而且無論鍵入幾次都一樣。

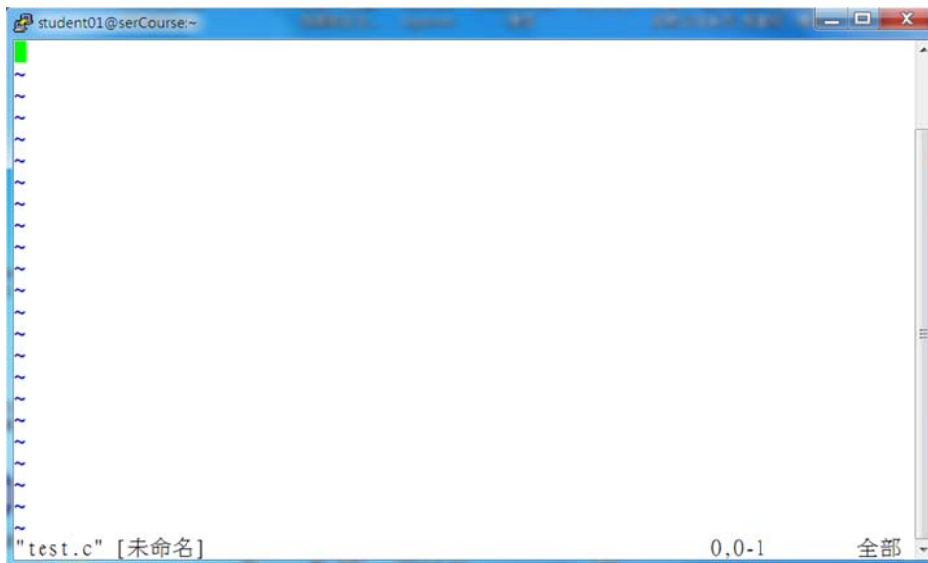


圖 4-2 vi 命令模式

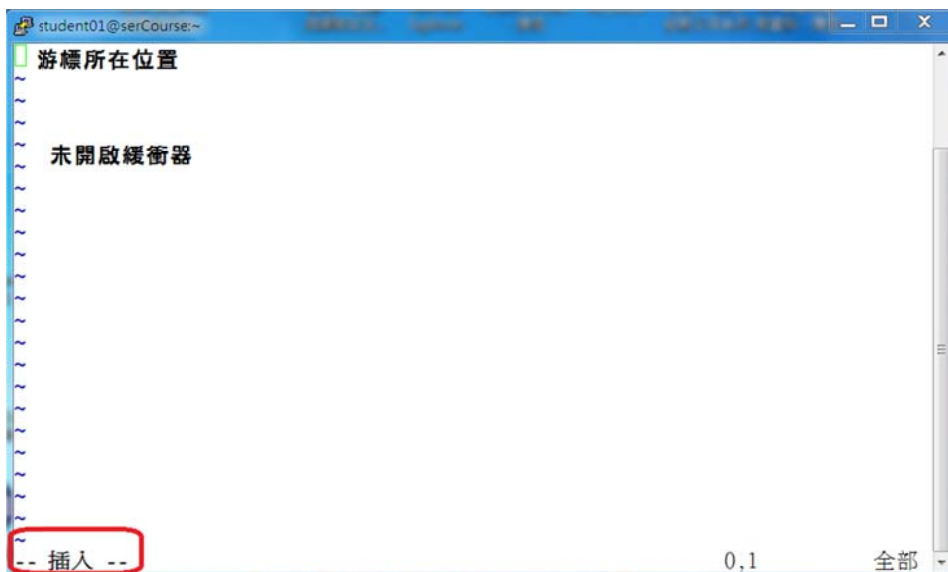


圖 4-3 vi 輸入模式

## 4-2-4 文字命令模式

在『單鍵命令模式』下輸入『:』(冒號)、『/』(斜線)或『?』(問號)，則可切換到『文字命令模式』，常用命令如下：

- 『:e file\_1』：編輯 file\_1 檔案 (編輯功能)。
- 『:w file\_2』：存檔寫入 file\_2 檔案 (寫入功能)。
- 『:q』：離開。
- 『:wq』：寫入並離開。
- 『:q!』：強迫離開 (不儲存)。
- 『:!ls -l』：執行 Shell 命令 (如 ls -l)；驚嘆號 (!) 表示可以執行 Unix/Linux 命令。
- 『:e』：回到全螢幕編輯。
- 『/text』：由游標位置開始往前，尋找 text 文字的所在位置 (尋找功能)。
- 『//』：重複上一個尋找。
- 『?text』：由游標開始倒回，尋找 text 文字所在功能。
- 『??』：重複上一個尋找。
- 利用 Backspace 鍵移除文字命令 (:、?、/)，則回到單鍵命令模式下。

## 4-3 vi/vim 開啟與儲存檔案

### 4-3-1 開啟檔案

無論主控台或透過連線 (telnet 或 putty) 的終端機，都可以利用 vi 開啟新檔案，命令格式如下：

```
$ vi Fedora
```

上述命令為開啟 vi 工作平台，如果檔案 Fedora 已存在的話，則會開啟 vi 並載入 Fedora 檔案；如果不存在的話，則開啟新檔案 Fedora。執行後會出現全螢幕的編輯環境，並直接進入『命令模式』，新檔案如圖 5-2 所示。



圖 4-3-1 vi 建立新檔案

進入 vi 之後，如再鍵入 i、I、a 或 A 按鍵，則可切換到『輸入模式』並可開始輸入資料，輸入文字當中敲入 ESC 鍵可切換回『命令模式』，任何時候都可以利用這些按鍵切換工作模式，如圖 5-3 所示。在輸入模式下，全螢幕最下面一行會出現『-- insert --』訊息；而命令模式則不會出現。如果使用者不知道目前處於何種模式底下，可以多按幾次 ESC 鍵，使其進入命令模式，再利用其他單鍵命令回到輸入模式。

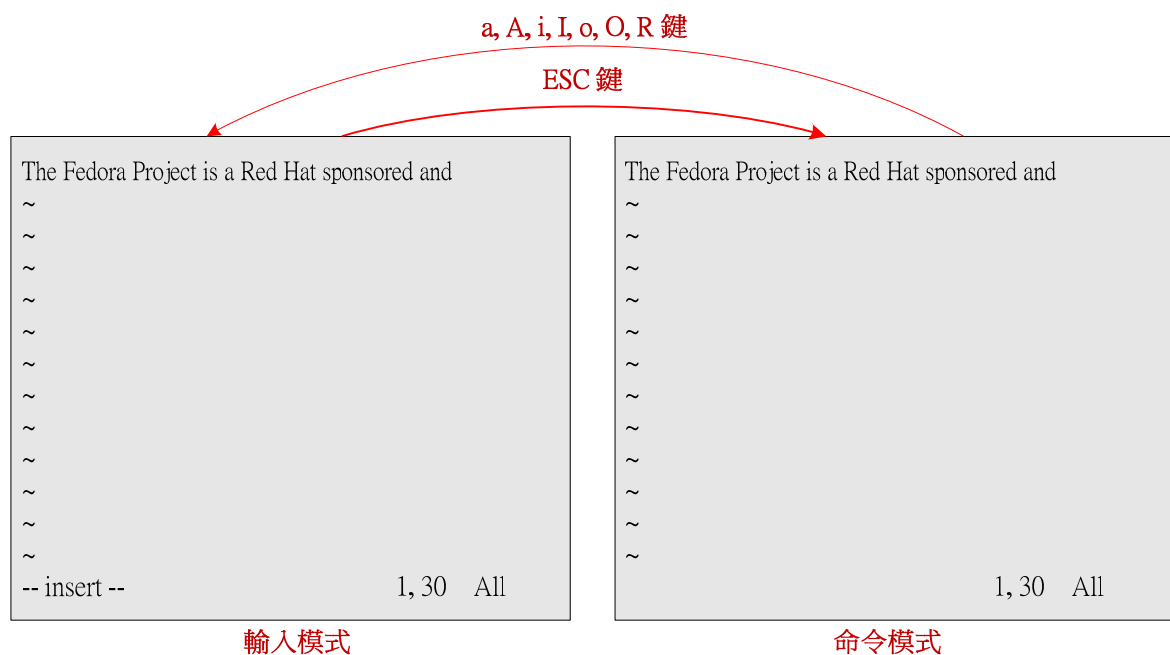


圖 4-4 vi 模式的切換

## 4-3-2 存檔並離開

編輯完文件後欲離開 vi 時，首先需進入命令模式，再輸入『:』(冒號)使其進入文字命令模式，再輸入『:wq!』，表示寫入檔案並強制離開的意思。

```
The Fedora Project is a Red Hat sponsored and
community-supported open source
project. It is not a supported product of Red Hat, Inc. The
goal? Work with the
Linux community to build a complete, general purpose
operating system
exclusively from free software. Public forum. Open
processes. A proving ground
for new technology that may eventually make its way into
Red Hat products.
~
~
~
~
:wq!
```

圖 4-4-1 vi 存檔並離開

## 4-4 vi/vim 命令

### 4-4-1 單鍵編輯命令

以下列出一些較常用的單鍵編輯命令 (命令模式下)，如下：

- x：刪除游標上的字元。
- u：復原 (undo) 最近一次修改。
- U：復原任何在目前所做的修改。
- dd：將目前游標所在的整行刪除，並存入緩衝器內 (連續按兩次 d 鍵)，3 dd 則為 3 行。
- yy：將目前游標所在的整行複製到緩衝器上，2yy 則為 2 行。
- p：將緩衝區內資料複製到游標所在位置。

上述命令中可以看出，vi 也具有 cut/copy/pasty 的功能，剪下 (cut) 即是 dd 命令，複製 (copy) 是 yy 命令，而貼上 (pasty) 是 p 命令。

## 4-4-2 文字編輯命令

在單鍵命令模式下，輸入『:』(冒號)、『/』(斜線)或『?』(問號)，則進入文字編輯命令，『冒號』或『斜線』會出現在全螢幕的最底下一行，表示已進入文字命令模式(如上圖 5-4 所示)。在此模式可以輸入較複雜命令來編輯文件，也可以設定 vi 的工作環境，以下說明較常用的命令。

### A. 尋找文字

- 『/text』：從游標所在位置開始，往下尋找 text 字串，如找到則停留在該字串位置上。
- 『//』：從游標位置開始，重複上一次往下尋找的字串(注意不用輸入冒號)。
- 『?text』：從游標位置開始，往回尋找 text 字串。
- 『?』：重複上一次往回尋找字串(注意不用輸入冒號)。

### B. 替代字串

- 『:s/old\_text/new\_text』：從游標位置開始尋找一個 old\_text 字串，並將其改為 new\_text 字串。
- 『:s/old\_text/new\_text/g』：從游標開始往下尋找，將所有 old\_text 字串改為 new\_text 字串。

### C. 設定行號

- 『:set all』：觀察 vi 環境下可設定的有哪些環境變數：
- 『:set number』：將 vi 環境設定有行號顯示。
- 『:set nonumber』：取消行號表示。

## 4-4-3 自動產生行號

雖然利用 set 命令(如 set number)可以設定 vi 的工作模式(如產生行號)，但使用

者離開 vi 再進入時，這些設定值將不復存在。因此，我們必須將這些設定寫成一個檔案，只要再進入 vi 時，vi 就會自動執行這些設定值，如此設定才會保持有效，而該檔案即是 .exrc 檔。假設期望每次進入 vi 都會出現行號，則所制定的 .exrc 檔案的內容如下：( 利用 \$vi .exrc 編輯 )

```
$ cat .exrc
set number
```

此檔案必須儲存於使用者的家目錄上。編輯完成之後，再執行 vi 則出現圖 4-5 的格式，操作如下：

```
$ cp /etc/passwd file1
$ vi file1
```

```
student01@serCourse:~
1 root:x:0:0:root:/root:/bin/bash
2 bin:x:1:1:bin:/bin:/sbin/nologin
3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
4 自動產生行號 var/adm:/sbin/nologin
5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
6 sync:x:5:0:sync:/sbin:/bin/sync
7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
8 halt:x:7:0:halt:/sbin:/sbin/halt
9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
10 operator:x:11:0:operator:/root:/sbin/nologin
11 games:x:12:100:games:/usr/games:/sbin/nologin
12 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
13 nobody:x:99:99:Nobody:/:/sbin/nologin
14 systemd-bus-proxy:x:999:998:systemd Bus Proxy:/:/sbin/nologin
15 systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
16 dbus:x:81:81:System message bus:/:/sbin/nologin
17 polkitd:x:998:997>User for polkitd:/:/sbin/nologin
18 abrt:x:173:173:/:etc/abrt:/sbin/nologin
19 unbound:x:997:995:Unbound DNS resolver:/etc/unbound:/sbin/nologin
20 tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:
/dev/null:/sbin/nologin
21 libstoragemgmt:x:996:994:daemon account for libstoragemgmt:/var/run/lsm:/sbi
n/nologin
"file1" 44L, 2296C 1,1 頂端
```

圖 4-5 具有行號的編輯器

## 4-5 程式範例練習

接下來我們用幾個 C 語言程式，希望讀者能依樣畫葫蘆方式輸入，若編譯當中檢視出錯誤，請讀者再利用 vi 修飾原來程式，一直到可以正常執行為止。安裝 gcc，如下：



```
$ su                【更換 root 身分】

密碼：             【輸入 root 密碼】

# yum -y install gcc  【安裝 gcc 編譯器】

Loaded plugins: fastestmirror, langpacks
Reposdata is over 2 weeks old. Install yum-cron? Or run: yum makecache fast
base                                                    | 3.6 kB      00:00
extras                                                  | 3.4 kB      00:00
updates                                                 | 3.4 kB      00:00
(1/2): extras/7/x86_64/primary_db                      | 121 kB     00:00
(2/2): updates/7/x86_64/primary_db                    | 2.2 MB     00:00
.....
.....

# exit                【回到原來帳戶身分】

$
```

### 4-5-1 範例一：hello.c

請鍵入一個 hello.c 檔案，命令如下：

```
$ vi hello.c
```

則進入 vi 編輯螢幕，並輸入下列程式（圖 4-11）。

```
# include <stdio.h>

int main(void) {
    printf("Good Lucky To You\n");
    return 0;
}
~
~
~
~
~
~
~
~
~
: wq!
```

圖 4-11 輸入範例一



```
int main(void) {
    int i, j;
    for (i=1; i<=9; i++) {
        for(j=2; j<=9; j++) {
            printf("%d*%d=%d\t", j, i, i*j);
        }
        printf("\n");
    }
    return 0;
}
```