

Linux – 檔案與目錄管理

Reading material

- 連猴子都能懂的Git入門指南
 - <https://backlog.com/git-tutorial/tw/>
- CentOS 7 最小安裝後在VirtualBox的網路設定筆記
 - <http://jimc1682000.blogspot.com/2015/09/centos-7-virtualbox.html>
- 指令下達行為與基礎檔案管理
 - http://linux.vbird.org/linux_basic_train/unit02.php
- 與 vim 初探
 - http://linux.vbird.org/linux_basic_train/unit03.php
 - http://linux.vbird.org/linux_basic/0310vi.php
- 檔案/目錄管理
 - http://linux.vbird.org/linux_basic/0220filemanager.php
 - http://linux.vbird.org/linux_basic/0210filepermission.php

指令格式: 通則，但不是絕對

```
[student@localhost ~]$ command [-options] [parameter1...]
```

- 一行指令中第一個輸入的部分是指令(command)或可執行檔案(例如script)
- 『 command 』：為指令的名稱，例如變換工作目錄的指令為 cd 等等；
- 中括號『 [] 』不在實際的指令中，僅作為一個提示，可有可無的資料之意；
- 『 -options 』：為選項，通常選項前面會帶有減號 (-)，例如 -h ；
- options 有時會提供長選項，此時會使用兩個減號，例如 --help 。
- 選項 -help 通常代表 -h -e -l -p 之意，與 --help 的單一長選項不同。
- 『 parameter1... 』：參數，為依附在選項後面的參數，或者是 command 的參數；
- 指令、選項、參數之間都以空格或 [tab] 作為區分，不論空幾格都視為一格，故空白是特殊字元
- [Enter]按鍵代表著一行指令的開始啟動。
- Linux 的世界中，英文大小寫為不同的字元，例如 cd 與 CD 是不一樣的指令。

```
[student@localhost ~]$ date
四  4月 21 02:43:24 CST 2016
```

```
[student@localhost ~]$ date +%Y/%m/%d
2016/04/21
```

```
[student@localhost ~]$ date --help
Usage: date [OPTION]... [+FORMAT]
    or:  date [-ul--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
Display the current time in the given FORMAT, or set the system date.
```

使用範例：

* 對 `a.c`, `b.c` `c.c` 進行編譯並產出 `file` 檔案.

```
gcc -o file a.c b.c c.c
```

* 對`test.c` 進行編譯並加入除錯訊息與顯示所有警告訊息並產出檔案 `test`

```
gcc -Wall -g -o test test.c
```

* 對`test1.c` 進行編譯且使用 `my_lib` 的 `library` 並到 *library_path* 下找對應的 `library`, 最後 `Include` 進來在*include_path* 下的檔案. 最後輸出成檔案 `test`

```
gcc -Iinclude_path -Llibrary_path -lmy_lib -o test1 test1.c
```

prompt

```
jimmy@ubuntu:~$
```

```
root@ubuntu:~#
```

```
✖ ogre0403@MBP ~/ResilioSync/ProjectSource/OtherProjest/107-1-ntcu-linux | score
```

切换身份

- \$ su -

```
jimmy@ubuntu:~$ su -  
Password: █
```

- \$ sudo su -

```
jimmy@ubuntu:~$ sudo su -  
[sudo] password for jimmy: █
```

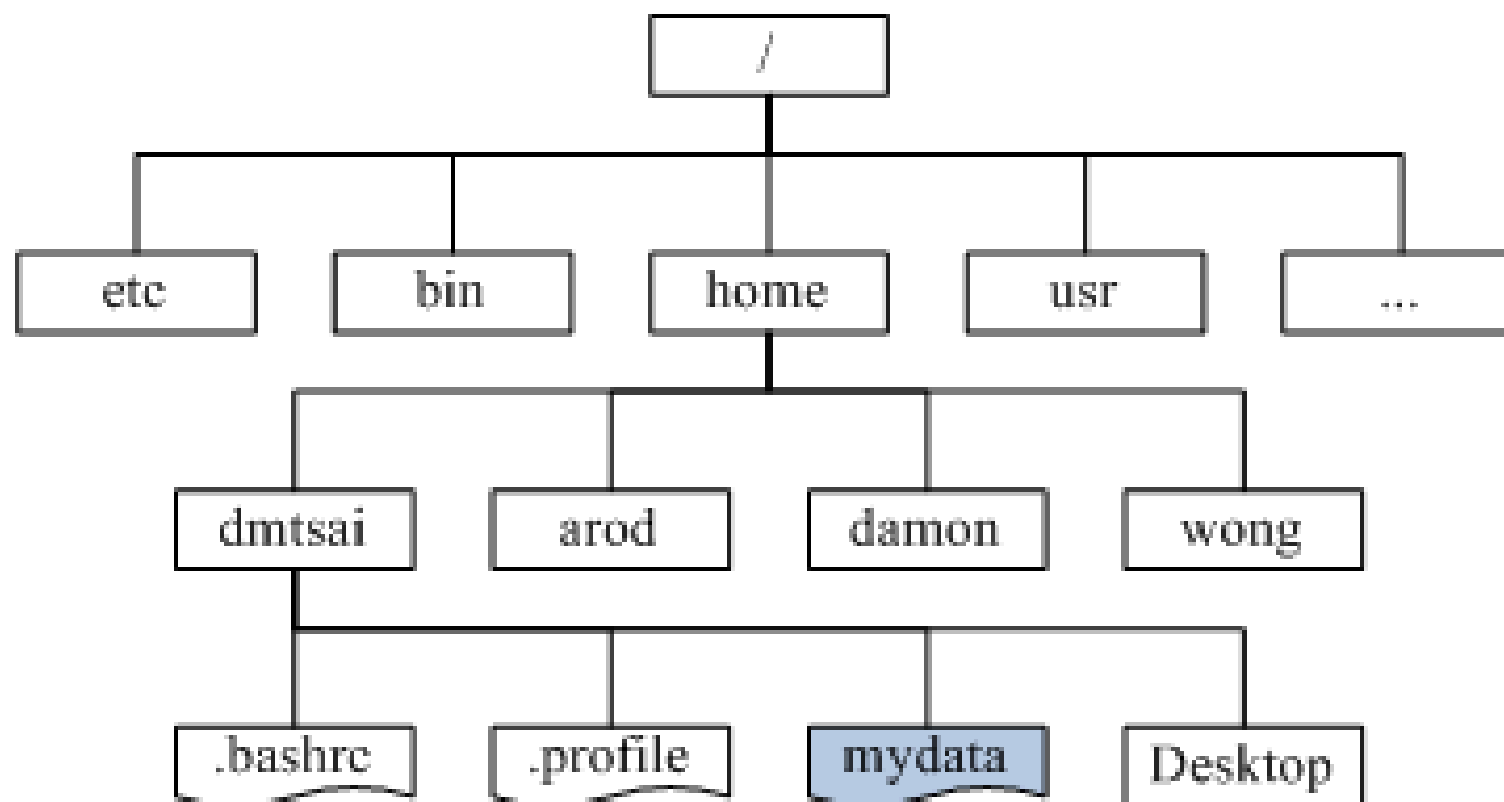
指令別名

- ls 和 ls -al
- ll 和 ls -al
-
- 用 alias 定義別名
 - alias ll='ls -aF'

指令管線 (pipe)

- |
- 將前一個指令輸出的資料，交由後面的指令來處理
- 情境: 指令輸出的資料量很大，要找出所需要資訊
- 並不是所有的指令都支援管線命令的，例如之前談到的 `ls`, `ll`, 能夠支援管線 `|` 的指令，就被稱為管線命令
- Eg:
 - 以 `ifconfig` 指令來觀察系統的所有介面卡 IP
 - 使用管線命令搭配 `grep` 取得關鍵字，來取出有 IP 的那行訊息即可

目錄結構



目錄名稱	應放置檔案內容(一定要知道的内容)
/bin /sbin	/bin 主要放置一般用戶可操作的指令 /sbin 主要放置系統管理員可操作的指令 這兩個資料目前都是連結檔，分別連結到 /usr/bin, /usr/sbin 當中
/boot	與開機有關的檔案，包括核心檔案 / 開機管理程式與設定檔
/dev	是 device 的縮寫，放置裝置檔，包括硬碟檔、鍵盤滑鼠終端機檔案等
/etc	一堆系統設定檔，包括帳號、密碼與各式服務軟體的設定檔大多在此目錄內
/home /root	/home 是一般帳號的家目錄預設放置位置 /root 則是系統管理員的家目錄了！
/lib /lib64	系統函式庫與核心函式庫，其中 /lib 包含核心驅動程式，而其他軟體的函式庫若為 64 位元，則使用 /lib64 目錄內的函式庫檔案。這兩個目錄目前也都是連結到 /usr/lib, /usr/lib64 內。
/proc	將記憶體內的資料做成檔案類型，放置於這個目錄下，連同某些核心參數也能手動調整
/sys	跟 /proc 類似，只是比較針對硬體相關的參數方面。
/usr	是 usr 不是 user 喔！是 unix software resource 的縮寫，與 Unix 程式有關。從 CentOS 7 開始，系統相關的所有軟體、服務等，均放置在這個目錄中了！因此不能與根目錄分離。
/var	是一些變動資料，系統運作過程中的服務資料、暫存資料、登錄資料等等。
/tmp	一些使用者操作過程中會啟用的暫存檔，例如 X 軟體相關的資料等等。

工作目錄的切換

- 在系統管理時非常的重要，若去錯目錄，會導致檔案修訂的錯誤
- 登入後，會進使用者的家目錄 `/home/user-name`
- Change directory (`cd`)
- print working directory (`pwd`)

特別的目錄代號

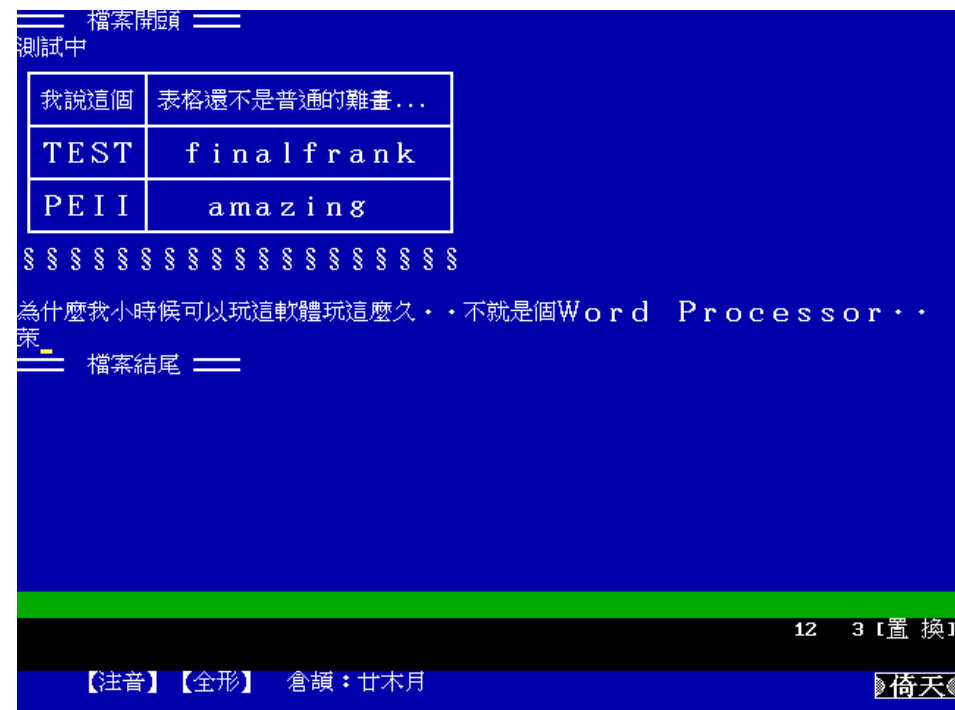
目錄名稱	目錄意義
/	根目錄，從根目錄寫起的檔名只會存在一個
~	使用者的家目錄，不同用戶的家目錄均不相同
.	一個小數點，代表的是『本目錄』，亦即目前的工作目錄之意
..	兩個小數點，代表的是『上一層目錄』
-	一個減號，代表『上一次的工作目錄』之意

相對/絕對路徑

- 絕對路徑：
 - 由根目錄(/)開始寫起的檔名或目錄名稱，例如 `/home/student/.bashrc`
- 相對路徑：
 - 開頭不是 / 就屬於相對路徑的寫法
 - 相對於目前路徑的檔名寫法。
 - 例如 `./home/student` 或 `../../home/student/`
 - 相對於家目錄的寫法
 - `~/.bashrc`

文字編輯器

- 命令列下編輯檔案的工具
- vim、emacs、nano、...
 - 請務必要熟悉一種工具



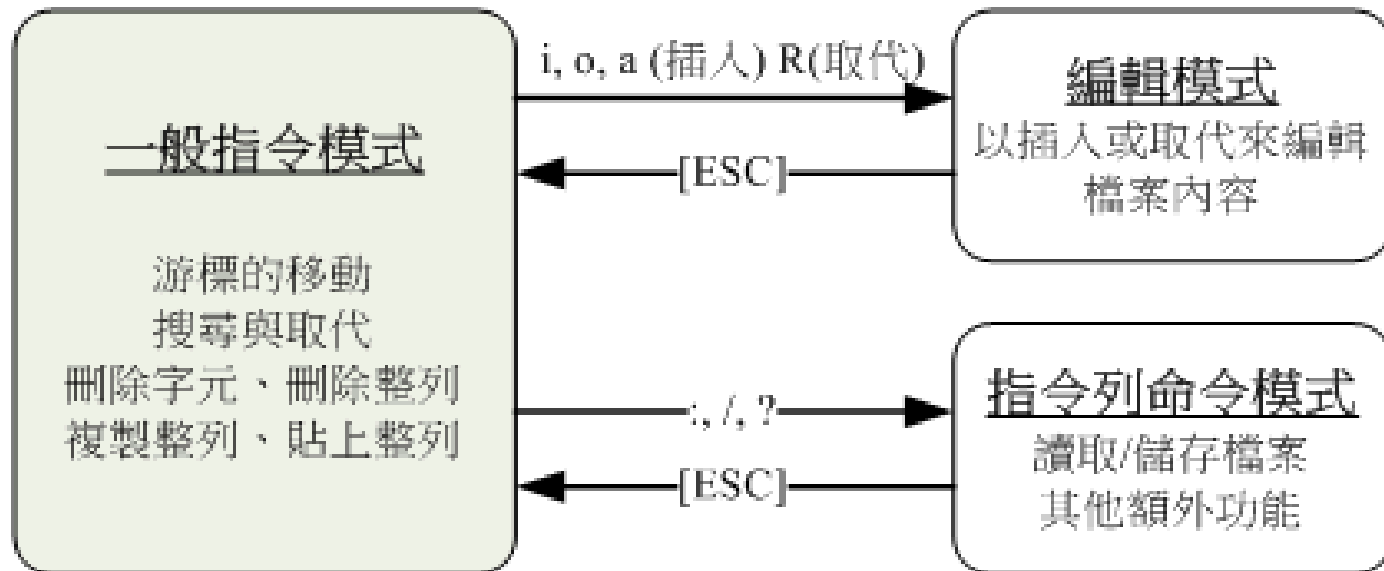
vi or vim

- \$ vim 檔名
 - 進入 vi 的一般指令模式

"test" [未命名]

0,0-1

全部

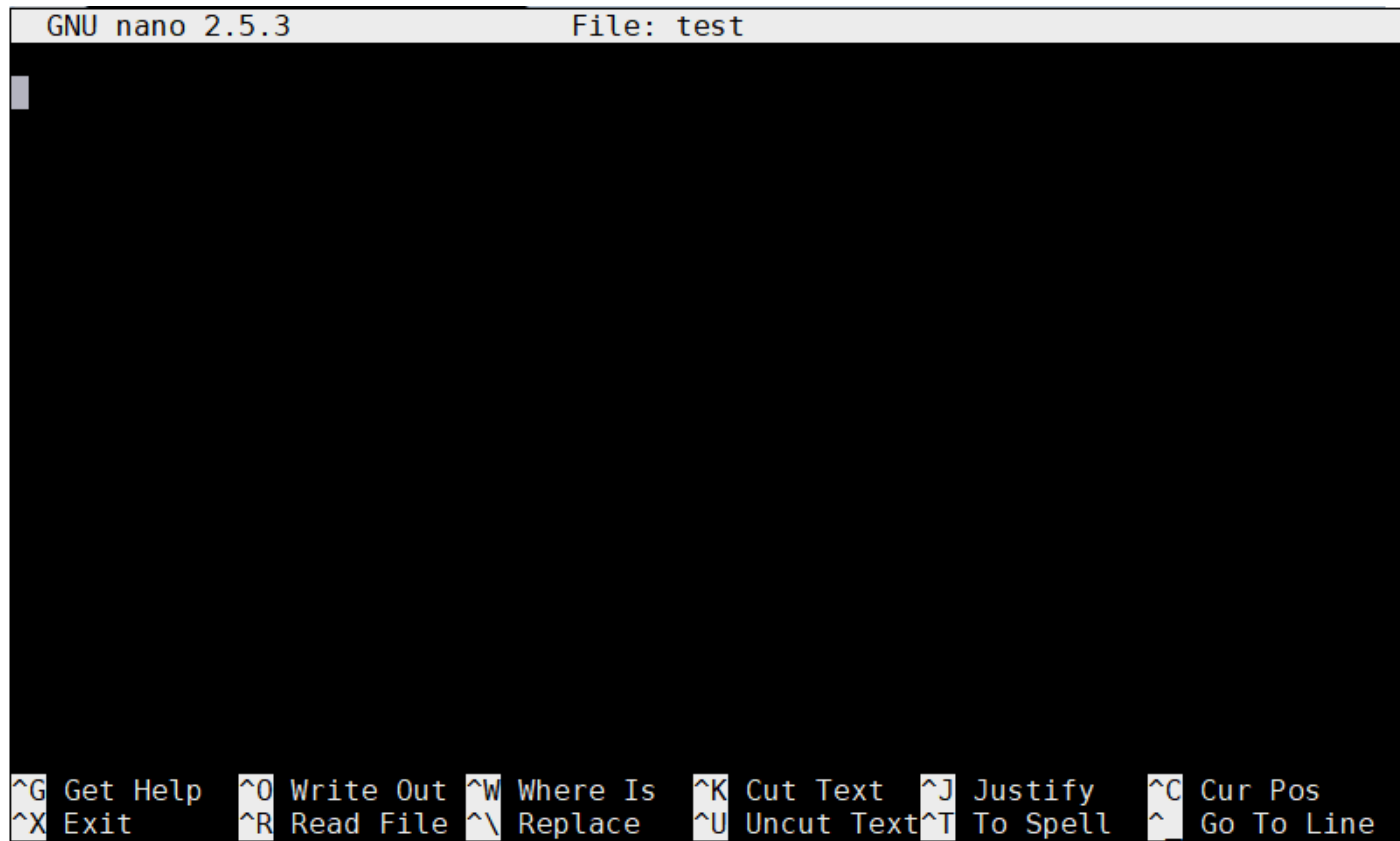


Vi 指令列命令

指令列模式的儲存、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。
注意一下啊，那個驚嘆號(!)在 vi 當中，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)

nano

- 和vim不同，沒有模式的區別。用^做操作
- \$ nano 檔名
- ^ = ctrl



The screenshot shows the GNU nano 2.5.3 text editor interface. The title bar at the top displays "GNU nano 2.5.3" on the left and "File: test" on the right. The main editing area is a large black rectangle. At the bottom, there is a status bar with a grid of keyboard shortcuts. The shortcuts are arranged in two rows: the first row includes ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, and ^C Cur Pos; the second row includes ^X Exit, ^R Read File, ^\ Replace, ^U Uncut Text, ^T To Spell, and ^_ Go To Line.

GNU nano 2.5.3	File: test				
^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^\ Replace	^U Uncut Text	^T To Spell	^_ Go To Line