

基礎學習篇 - CentOS 7.x

第五章、Linux 的檔案權限與目錄配置

Linux 的檔案權限與目錄配置-這真的很重要！鳥哥初次接觸 Linux 時，一堆 rwx 搞不清楚呢！

>> 回到鳥哥本站
>> 回基礎學習版本彙整
關於 CentOS 7.x 基礎學習 - 目錄總覽
第零章、計算機概論
第一章、Linux是什麼與如何學習
第二章、主機規劃與磁碟分割
第三章、安裝 CentOS7.x
第四章、首次登入與線上求助
第五章、Linux 的檔案權限與目錄配置
第六章、Linux 檔案與目錄管理
第七章、Linux 磁碟與檔案系統管理
第八章、檔案與檔案系統的壓縮,打包與備份
第九章、vim 程式編輯器
第十章、認識與學習BASH
第十一章、正規表示法與文件格式化處理
第十二章、學習 Shell Scripts
第十三章、Linux 帳號管理與 ACL 權限設定
第十四章、磁碟配額(Quota)與進階檔案系統管理
第十五章、例行性工作排程(crontab)
第十六章、程序管理與 SELinux 初探
第十七章、認識系統服務 (daemons)
第十八章、認識與分析登錄檔
第十九章、開機流程、模組管理與 Loader
第二十章、基礎系統設定與備份策略
第二十一章、軟體安裝：原始碼與 Tarball
第二十二章、軟體安裝 RPM, SRPM 與 YUM
第二十三章、X Window 設定介紹
第二十四章、Linux 核心編譯與管理

最近更新時間：2015/06/03

Linux最優秀的地方之一就在於他的多人多工環境。而為了讓各個使用者具有較保密的檔案資料，因此檔案的權限管理就變的很重要了。Linux一般將檔案可存取的身份分為三個類別，分別是 owner/group/others，且三種身份各有 read/write/execute 等權限。若管理不當，你的Linux主機將會變的很『不蘇湖！@_@』。另外，你如果首次接觸Linux的話，那麼，在Linux底下這麼多的目錄/檔案，到底每個目錄/檔案代表什麼意義呢？底下我們就來一一介紹呢！

[5.1 使用者與群組](#)

[5.2 Linux檔案權限概念](#)

[5.2.1 Linux檔案屬性, 改變語系的 locale](#)

[5.2.2 如何改變檔案屬性與權限：chgrp, chown, chmod](#)

[5.2.3 目錄與檔案之權限意義：, 資料夾與抽屜, 各項動作所需最小權限](#)

[5.2.4 Linux檔案種類與副檔名](#)

[5.3 Linux目錄配置](#)

[5.3.1 Linux目錄配置的依據--FHS：/, /usr, /var](#)

[5.3.2 目錄樹\(directory tree\)](#)

[5.3.3 絕對路徑與相對路徑](#)

[5.3.4 CentOS 的觀察：lsb_release](#)

[5.4 重點回顧](#)

[5.5 本章練習](#)

[5.6 參考資料與延伸閱讀](#)

針對本文的建議：<http://phorum.vbird.org/viewtopic.php?t=23878>

5.1 使用者與群組

經過[第四章](#)的洗禮之後，你應該可以在Linux的指令列模式底下輸入指令了吧？接下來，當然是要讓你好好的瀏覽一下Linux系統裡面有哪些重要的檔案囉。不過，每個檔案都有相當多的屬性與權限，其中最重要的可能就是檔案的擁有者的概念了。所以，在開始檔案相關資訊的介紹前，鳥哥就先簡單的(1)使用者及(2)群組與(3)非本群組外的其他人等概念作個說明吧～好讓你快點進入狀況的哩！^_^

1. 檔案擁有者

初次接觸Linux的朋友大概會覺得很怪異，怎麼『Linux有這麼多使用者，還分什麼群組，有什麼用？』。這個『使用者與群組』的功能可是相當健全而好用的一個安全防護呢！怎麼說呢？由於Linux是個多人多工的系統，因此可能常常會有多人同時使用這部主機來進行工作的情況發生，為了考慮每個人的隱私權以及每個人喜好的工作環境，因此，這個『檔案擁有者』的角色就顯的相當的重要了！

例如當你將你的e-mail情書轉存成檔案之後，放在你自己的家目錄，你總不希望被其他人看見自己的情書吧？這個時候，你就把該檔案設定成『只有檔案擁有者，就是我，才能看與修改這個檔案的內容』，那麼即使其他人知道你有這個相當『有趣』的檔案，不過由於你有設定適當的權限，所以其他人自然也就無法知道該檔案的內容囉！

2. 群組概念

那麼群組呢？為何要設定檔案還有所屬的群組？其實，**群組最有用的功能之一，就是當你在團隊開發資源的時候啦！**舉例來說，假設有兩組專題生在我的主機裡面，第一個專題組別為projecta，裡面的成員有 class1, class2, class3三個；第二個專題組別為projectb，裡面的成員有class4, class5, class6。這兩個專題之間是有競爭性質的，但卻要繳交同一份報告。每組的組員之間必須要能夠互相修改對方的資料，但是其他組的組員則不能看到本組自己的檔案內容，此時該如何是好？

在Linux底下這樣的限制是很簡單啦！我可以經由簡易的檔案權限設定，就能限制非自己團隊(亦即是群組囉)的其他人不能夠閱覽內容囉！而且亦可以讓自己的團隊成員可以修改我所建立的檔案！同時，如果我自己還有私人隱密的文件，仍然可以設定成讓自己的團隊成員也看不到我的檔案資料。很方便吧！

另外，如果teacher這個帳號是projecta與projectb這兩個專題的老師，他想要同時觀察兩者的進度，因此需要能夠進入這兩個群組的權限時，你可以設定teacher這個帳號，『同時支援projecta與projectb這兩個群組！』，也就是說：**每個帳號都可以有多個群組的支援呢！**

這樣說或許你還不容易理解這個使用者與群組的關係吧？沒關係，我們可以使用目前『家庭』的觀念來進行解說喔！假設有一家人，家裡只有三兄弟，分別是王大毛、王二毛與王三毛三個人，而這個家庭是登記在王大毛的名下的！所以，『王大毛家有三個人，分別是王大毛、王二毛與王三毛』，而且這三個人都有自己的房間，並且共同擁有一個客廳喔！

- **使用者的意義**：由於王家三人各自擁有自己的房間，所以，王二毛雖然可以進入王三毛的房間，但是二毛不能翻三毛的抽屜喔！那樣會被三毛K的！因為抽屜裡面可能有三毛自己私人的東西，例如情書啦，日記啦等等的，這是『私人的空間』，所以當然不能讓二毛拿囉！
- **群組的概念**：由於共同擁有客廳，所以王家三兄弟可以在客廳打開電視機啦、翻閱報紙啦、坐在沙發上面發呆啦等等的！反正，只要是在客廳的玩意兒，三兄弟都可以使用喔！因為大家都是一家人嘛！

這樣說來應該有點曉得了喔！那個『王大毛家』就是所謂的『群組』囉，至於三兄弟就是分別為三個『使用者』，而這三個使用者是在同一個群組裡面的喔！而三個使用者雖然在同一群組內，但是我們可以設定『權限』，好讓某些使用者個人的資訊不被群組的擁有者查詢，以保有個人『私人的空間』啦！而設定群組共享，則可讓大家共同分享喔！

3. 其他人的概念

好了，那麼今天又有個人，叫做張小豬，他是張小豬家的人，與王家沒有關係啦！這個時候，除非王家認識張小豬，然後開門讓張小豬進來王家，否則張小豬永遠沒有辦法進入王家，更不要說進到王三毛的房間啦！不過，如果張小豬透過關係認識了三毛，並且跟王

三毛成為好朋友，那麼張小豬就可以透過三毛進入王家啦！呵呵！沒錯！那個張小豬就是所謂的『其他人，Others』囉！

因此，我們就可以知道啦，在Linux裡面，任何一個檔案都具有『User, Group及Others』三種身份的個別權限，我們可以將上面的說明以底下的圖示來解釋：

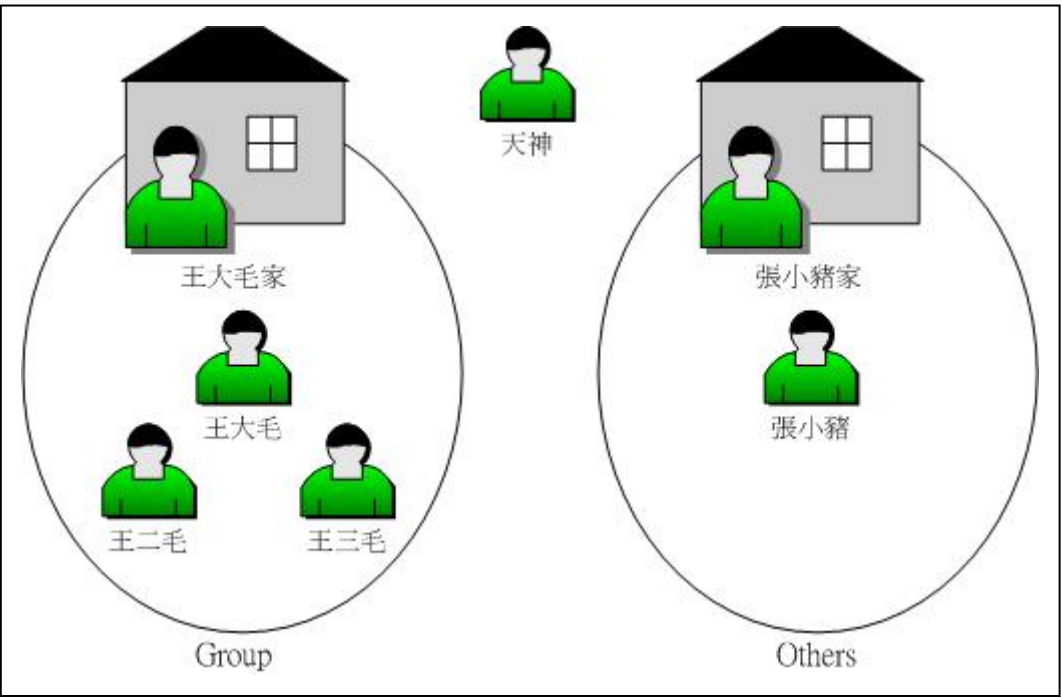


圖5.1.1、每個檔案的擁有者、群組與 others 的示意圖

我們以王三毛為例，王三毛這個『檔案』的擁有者為王三毛，他屬於王大毛這個群組，而張小豬相對於王三毛，則只是一個『其他人 (others)』而已。

不過，這裡有個特殊的人物要來介紹的，那就是『萬能的天神』！這個天神具有無限的神力，所以他可以到達任何他想要去的地方，呵呵！那個人在Linux系統中的身份代號是『root』啦！所以要小心喔！那個root可是『萬能的天神』喔！

無論如何，『使用者身份』，與該使用者所支援的『群組』概念，在Linux的世界裡面是相當的重要的，他可以幫助你讓你的多工Linux環境變的更容易管理！更詳細的『身份與群組』設定，我們將在[第十三章、帳號管理](#)再進行解說。底下我們將針對檔案系統與檔案權限來進行說明。

Tips

現在 (2015年) 鳥哥常以台灣地區常見的社群網站 Facebook 或者是 Google+ 作為解釋。(1)你在 FB 註冊一個帳號，這個帳號可以疊代對比為 Linux 的帳號，(2)你可以新增一個社團，這個社團的隱私權是可以由您自己指定的！看是要公開還是要隱藏。這就可以疊代為 Linux 的群組概念，這個群組的權限可以自己設定。(3)那麼其他在 FB 註冊的人，沒有加入你的社團，他就是 Linux 上所謂的『其他人』！最後，在 FB 上面的每一條留言，就可以想成 Linux 底下的『檔案』囉！



Tips

那麼上面內文談到的群組有啥幫助呢？想想看，你在 FB 上面，你的 StudyArea 社團是隱藏的，你想讓 dmstai 可以進來讀取每一個留言 (想成是 file)，最簡單的作法是什麼？對！讓 dmstai 加入這個社團即可！沒錯！只要讓 Linux 某個帳號加入某個群組，該帳號就可以使用該群組能夠存取的資源！每個帳號可以加入的群組個數基本上是有沒有限制的！



Linux 使用者身份與群組記錄的檔案

在我們Linux系統當中，預設的情況下，所有的系統上的帳號與一般身份使用者，還有那個root的相關資訊，都是記錄在/etc/passwd這個檔案內的。至於個人的密碼則是記錄在/etc/shadow這個檔案下。此外，Linux所有的群組名稱都紀錄在/etc/group內！這三個檔案可以說是Linux系統裡面帳號、密碼、群組資訊的集中地囉！不要隨便刪除這三個檔案啊！^_^

至於更多的與帳號群組有關的設定，還有這三個檔案的格式，不要急，我們在[第十三章的帳號管理](#)時，會再跟大家詳細的介紹的！這裡先有概念即可。

5.2 Linux 檔案權限概念

大致瞭解了Linux的使用者與群組之後，接著下來，我們要來談一談，這個檔案的權限要如何針對這些所謂的『使用者』與『群組』來設定呢？這個部分是相當重要的，尤其對於初學者來說，因為檔案的權限與屬性是學習Linux的一個相當重要的關卡，如果沒有這部份的概念，那麼你將老是聽不懂別人在講什麼呢！尤其是當你在你的螢幕前面出現了『Permission deny』的時候，不要擔心，『肯定是權限設定錯誤』啦！呵呵！好了，閒話不多聊，趕快來瞧一瞧先。

5.2.1 Linux檔案屬性

嗯！既然要讓你瞭解Linux的檔案屬性，那麼有個重要的也是常用的指令就必須要先跟你說囉！那一個？就是『ls』這一個察看檔案的指令囉！在你以dmtsai登入系統，然後使用 su - 切換身份成為root後，下達『ls -al』看看，會看到底下的幾個咚咚：

```
[dmtsai@study ~]$ su - # 先來切換一下身份看看
Password:
Last login: Tue Jun  2 19:32:31 CST 2015 on tty2
[root@study ~]# ls -al
total 48
dr-xr-x---. 5 root root 4096 May 29 16:08 .
dr-xr-xr-x. 17 root root 4096 May  4 17:56 ..
-rw-----. 1 root root 1816 May  4 17:57 anaconda-ks.cfg
-rw-----. 1 root root  927 Jun  2 11:27 .bash_history
-rw-r--r--. 1 root root   18 Dec 29 2013 .bash_logout
-rw-r--r--. 1 root root  176 Dec 29 2013 .bash_profile
-rw-r--r--. 1 root root  176 Dec 29 2013 .bashrc
drwxr-xr-x. 3 root root   17 May  6 00:14 .config          <=範例說明處
drwx-----. 3 root root   24 May  4 17:59 .dbus
-rw-r--r--. 1 root root 1864 May  4 18:01 initial-setup-ks.cfg <=範例說明處
[ 1 ][ 2 ][ 3 ][ 4 ][ 5 ][ 6 ][ 7 ]
[ 權限 ][連結][擁有者][群組][檔案容量][ 修改日期 ][ 檔名 ]
```

Tips

由於本章後續的chgrp, chown等指令可能都需要使用root的身份才能夠處理，所以這裡建議您以root的身份來學習！要注意的是，我們還是不建議你直接使用 root 登入系統，建議使用 su - 這個指令來切換身份喔！離開 su - 則使用 exit 回到 dmtsai 的身份即可！



ls是『list』的意思，重點在顯示檔案的檔名與相關屬性。而選項『-al』則表示列出所有的檔案詳細的權限與屬性 (包含隱藏檔，就是檔名第一個字元為『.』的檔案)。如上所示，在你第一次以root身份登入Linux時，如果你輸入上述指令後，應該有上列的幾個東西，先解釋一下上面七個欄位個別的意思：

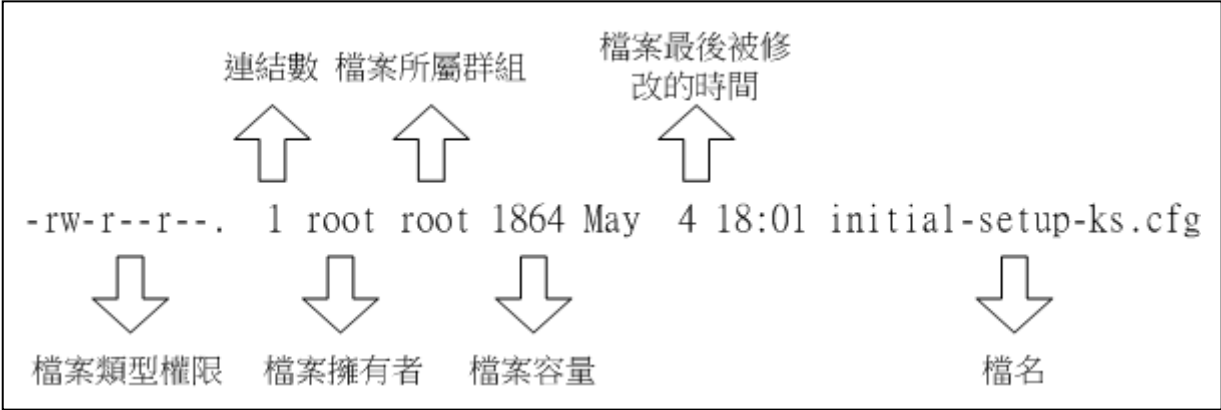


圖5.2.1、檔案屬性的示意圖

- 第一欄代表這個檔案的類型與權限(permission)：

這個地方最需要注意了！仔細看的話，你應該可以發現這一欄其實共有十個字元：(圖5.2.1及圖5.2.2內的權限並無關係)

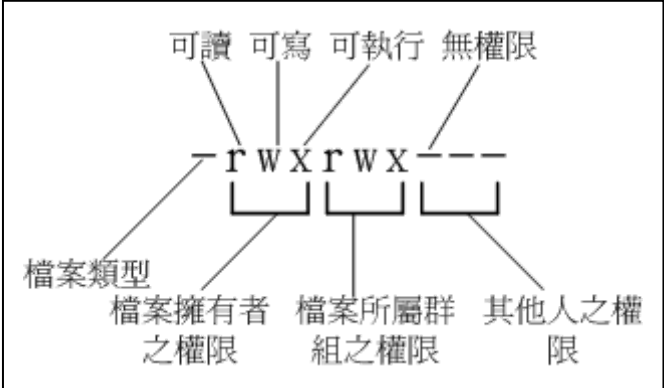


圖5.2.2、檔案的類型與權限之內容

- 第一個字元代表這個檔案是『目錄、檔案或連結檔等等』：
 - 當為[d]則是目錄，例如上表檔名為『.config』的那一行；
 - 當為[-]則是檔案，例如上表檔名為『initial-setup-ks.cfg』那一行；
 - 若是[l]則表示為連結檔(link file)；
 - 若是[b]則表示為裝置檔裡面的可供儲存的周邊設備(可隨機存取裝置)；
 - 若是[c]則表示為裝置檔裡面的序列埠設備，例如鍵盤、滑鼠(一次性讀取裝置)。

- 接下來的字元中，以三個為一組，且均為『**rw**x』的三個參數的組合。其中，[r]代表可讀(read)、[w]代表可寫(write)、[x]代表可執行(execute)。要注意的是，這三個權限的位置不會改變，如果沒有權限，就會出現減號[-]而已。
 - 第一組為『**檔案擁有者可具備的權限**』，以『initial-setup-ks.cfg』那個檔案為例，該檔案的擁有者可以讀寫，但不可執行；
 - 第二組為『**加入此群組之帳號的權限**』；
 - 第三組為『**非本人且沒有加入本群組之其他帳號的權限**』。

Tips

請你特別注意喔！不論是那一組權限，基本上，都是『針對某些帳號來設計的權限』喔！以群組來說，他規範的是『加入這個群組的帳號具有什麼樣的權限』之意，以學校社團為例，假設學校有個童軍社的社團辦公室，『加入童軍社的同學就可以進出社辦』，主角是『學生(帳號)』而不是童軍社本身喔！這樣可以理解嗎？



例題：

若有一個檔案的類型與權限資料為『-**rw**xr-xr--』，請說明其意義為何？

答：

先將整個類型與權限資料分開查閱，並將十個字元整理成為如下所示：

1 為：代表這個檔名為目錄或檔案，本例中為檔案(-)；

234為：擁有者的權限，本例中為可讀、可寫、可執行(rwx)；

567為：同群組使用者權限，本例中為可讀可執行(rx)；

890為：其他使用者權限，本例中為可讀(r)，就是唯讀之意

同時注意到，rwx所在的位置是不會改變的，有該權限就會顯示字元，沒有該權限就變成減號(-)就是了。

另外，目錄與檔案的權限意義並不相同，這是因為目錄與檔案所記錄的資料內容不相同所致。由於目錄與檔案的權限意義非常的重要，所以鳥哥將他獨立到[5.2.3節中的目錄與檔案之權限意義](#)中再來談。

- **第二欄表示有多少檔名連結到此節點(i-node)：**

每個檔案都會將他的權限與屬性記錄到檔案系統的i-node中，不過，我們使用的目錄樹卻是使用檔名來記錄，因此每個檔名就會連結到一個i-node囉！這個屬性記錄的，就是有多少不同的檔名連結到相同的一個i-node號碼去就是了。關於i-node的相關資料我們會在[第七章](#)談到檔案系統時再加強介紹的。

- **第三欄表示這個檔案(或目錄)的『擁有者帳號』**
- **第四欄表示這個檔案的所屬群組**

在Linux系統下，你的帳號會加入於一個或多個的群組中。舉剛剛我們提到的例子，class1, class2, class3均屬於projecta這個群組，假設某個檔案所屬的群組為projecta，且該檔案的權限如圖5.2.2所示(-rwxrwx---)，則class1, class2, class3三人對於該檔案都具有可讀、可寫、可執行的權限(看群組權限)。但如果是不屬於projecta的其他帳號，對於此檔案就不具有任何權限了。

- **第五欄為這個檔案的容量大小，預設單位為bytes；**
- **第六欄為這個檔案的建檔日期或者是最近的修改日期：**

這一欄的內容分別為日期(月/日)及時間。如果這個檔案被修改的時間距離現在太久了，那麼時間部分會僅顯示年份而已。如下所示：

```
[root@study ~]# ll /etc/services /root/initial-setup-ks.cfg
-rw-r--r--. 1 root root 670293 Jun  7 2013 /etc/services
-rw-r--r--. 1 root root  1864 May  4 18:01 /root/initial-setup-ks.cfg
# 如上所示，/etc/services 為 2013 年所修改過的檔案，離現在太遠之故，所以只顯示年份；
# 至於 /root/initial-setup-ks.cfg 是今年（2015）所建立的，所以就顯示完整的時間了。
```

如果想要顯示完整的時間格式，可以利用ls的選項，亦即：『ls -l --full-time』就能夠顯示出完整的時間格式了！包括年、月、日、時間喔。另外，如果你當初是以繁體中文安裝你的Linux系統，那麼日期欄位將會以中文來顯示。可惜的是，[中文並沒有辦法在純文字的終端機模式中正確的顯示](#)，所以此欄會變成亂碼。那你就得要使用『export LC_ALL=en_US.utf8』來修改語系囉！

如果想要讓系統預設的語系變成英文的話，那麼你可以修改系統設定檔『[/etc/locale.conf](#)』，利用第四章談到的[nano](#)來修改該檔案的內容，使LANG這個變數成為上述的內容即可。

• 第七欄為這個檔案的檔名

這個欄位就是檔名了。比較特殊的是：如果檔名之前多一個『.』，則代表這個檔案為『隱藏檔』，例如[上表中的.config那一行](#)，該檔案就是隱藏檔。你可以使用『ls』及『ls -a』這兩個指令去感受一下什麼是隱藏檔囉！

Tips

對於更詳細的ls用法，還記得怎麼查詢嗎？對啦！使用ls --help 或 man ls 或 info ls 去看看他的基礎用法去！自我進修是很重要的，因為『師傅帶進門，修行看個人！』，自古只有天才學生，沒有明星老師呦！加油吧！^_^



這七個欄位的意義是很重要的！務必清楚的知道各個欄位代表的意義！尤其是第一個欄位的九個權限，那是整個Linux檔案權限的重點之一。底下我們來做幾個簡單的練習，你就會比較清楚囉！

例題：

假設test1, test2, test3同屬於testgroup這個群組，如果有下面的兩個檔案，請說明兩個檔案的擁有者與其相關的權限為何？

```
-rw-r--r-- 1 root    root      238 Jun 18 17:22 test.txt
-rwxr-xr-- 1 test1  testgroup 5238 Jun 19 10:25 ping_tsai
```

答：

- 檔案test.txt的擁有者為root，所屬群組為root。至於權限方面則只有root這個帳號可以存取此檔案，其他人則僅能讀此檔案；
- 另一個檔案ping_tsai的擁有者為test1，而所屬群組為testgroup。其中：
 - test1 可以針對此檔案具有可讀可寫可執行的權力；
 - 而同群組的test2, test3兩個人與test1同樣是testgroup的群組帳號，則僅可讀可執行但不能寫(亦即不能修改)；
 - 至於沒有加入testgroup這一個群組的其他人則僅可以讀，不能寫也不能執行！

例題：

承上一題如果我的目錄為底下的樣式，請問testgroup這個群組的成員與其他(others)是否可以進入本目錄？

```
drwxr-xr-- 1 test1  testgroup 5238 Jun 19 10:25 groups/
```

答：

- 檔案擁有者test1[rwx]可以在本目錄中進行任何工作；
- 而testgroup這個群組[r-x]的帳號，例如test2, test3亦可以進入本目錄進行工作，但是不能在本目錄下進行寫入的動作；
- 至於other的權限中[r--]雖然有r，但是由於沒有x的權限，因此others的使用者，並不能進入此目錄！

Linux檔案權限的重要性：

與Windows系統不一樣的是，在Linux系統當中，每一個檔案都多加了很多的屬性進來，尤其是群組的概念，這樣有什麼用途呢？其實，最大的用途是在『資料安全性』上面的。

- 系統保護的功能：
舉個簡單的例子，在你的系統中，關於系統服務的檔案通常只有root才能讀寫或者是執行，例如/etc/shadow這一個帳號管理的檔案，由於該檔案記錄了你系統中所有帳號的資料，因此是很重要的一個設定檔，當然不能讓任何人讀取(否則密碼會被竊取啊)，只有root才能夠來讀取囉！所以該檔案的權限就會成為[-----]囉！咦！所有人都不能使用？沒關係，root基本上是不受系統的權限所限制的，所以無論檔案權限為何，預設root都可以存取喔！
- 團隊開發軟體或資料共用的功能：
此外，如果你有一個軟體開發團隊，在你的團隊中，你希望每個人都可以使用某一些目錄下的檔案，而非你的團隊的其他人則不予以開放呢？以上面的例子來說，testgroup的團隊共有三個人，分別是test1, test2, test3，那麼我就可以將團隊所需的檔案權限訂為[-rwxrws---]來提供給testgroup的工作團隊使用囉！(怎麼會有s呢？沒關係，這個我們在後續章節再講給你聽！)
- 未將權限設定妥當的危害：
再舉個例子來說，如果你的目錄權限沒有作好的話，可能造成其他人都可以在你的系統上面亂搞囉！例如本來只有root才能做的

開關機、ADSL的撥接程式、新增或刪除使用者等等的指令，若被你改成任何人都可以執行的話，那麼如果使用者不小心給你重新開機啦！重新撥接啦！等等的！那麼你的系統不就會常常莫名其妙的掛掉囉！而且萬一你的使用者的密碼被其他不明人士取得的話，只要他登入你的系統就可以輕而易舉的執行一些root的工作！

可怕吧！因此，在你修改你的linux檔案與目錄的屬性之前，一定要先搞清楚，什麼資料是可變的，什麼是不可變的！千萬注意囉！接下來我們來處理一下檔案屬性與權限的變更吧！

5.2.2 如何改變檔案屬性與權限

我們現在知道檔案權限對於一個系統的安全重要性了，也知道檔案的權限對於使用者與群組的相關性，那麼如何修改一個檔案的屬性與權限呢？又！有多少檔案的權限我們可以修改呢？其實一個檔案的屬性與權限有很多！我們先介紹幾個常用於群組、擁有者、各種身份的權限之修改的指令，如下所示：

- `chgrp`：改變檔案所屬群組
- `chown`：改變檔案擁有者
- `chmod`：改變檔案的權限, SUID, SGID, SBIT等等的特性

改變所屬群組, `chgrp`

改變一個檔案的群組真是很簡單的，直接以`chgrp`來改變即可，咦！這個指令就是change group的縮寫嘛！這樣就很好記了吧！^_^。不過，請記得，要被改變的群組名稱必須要在`/etc/group`檔案內存在才行，否則就會顯示錯誤！

假設你已經是root的身份了，那麼在你的家目錄內有一個名為 `initial-setup-ks.cfg` 的檔案，如何將該檔案的群組改變一下呢？假設你已經知道在`/etc/group`裡面已經存在一個名為users的群組，但是testing這個群組名字就不存在`/etc/group`當中了，此時改變群組成為users與testing分別會有什麼現象發生呢？

```
[root@study ~]# chgrp [-R] dirname/filename ...
選項與參數：
-R：進行遞迴(recursive)的持續變更，亦即連同次目錄下的所有檔案、目錄
    都更新成為這個群組之意。常常用在變更某一目錄內所有的檔案之情況。
範例：
[root@study ~]# chgrp users initial-setup-ks.cfg
[root@study ~]# ls -l
-rw-r--r--. 1 root users 1864 May  4 18:01 initial-setup-ks.cfg
[root@study ~]# chgrp testing initial-setup-ks.cfg
chgrp: invalid group: `testing' <== 發生錯誤訊息囉～找不到這個群組名～
```

發現了嗎？檔案的群組被改成users了，但是要改成testing的時候，就會發生錯誤～注意喔！發生錯誤訊息還是要努力的查一查錯誤訊息的內容才好！將他英文翻譯成為中文，就知道問題出在哪裡了。

改變檔案擁有者, `chown`

如何改變一個檔案的擁有者呢？很簡單呀！既然改變群組是change group，那麼改變擁有者就是change owner囉！BINGO！那就是`chown`這個指令的用途，要注意的是，使用者必須是已經存在系統中的帳號，也就是在`/etc/passwd` 這個檔案中有紀錄的使用者名稱才能改變。

`chown`的用途還滿多的，他還可以順便直接修改群組的名稱呢！此外，如果要連目錄下的所有次目錄或檔案同時更改檔案擁有者的話，直接加上 `-R` 的選項即可！我們來看看語法與範例：

```
[root@study ~]# chown [-R] 帳號名稱 檔案或目錄
[root@study ~]# chown [-R] 帳號名稱:群組名稱 檔案或目錄
選項與參數：
-R：進行遞迴(recursive)的持續變更，亦即連同次目錄下的所有檔案都變更
範例：將 initial-setup-ks.cfg 的擁有者改為bin這個帳號：
[root@study ~]# chown bin initial-setup-ks.cfg
[root@study ~]# ls -l
-rw-r--r--. 1 bin users 1864 May  4 18:01 initial-setup-ks.cfg
範例：將 initial-setup-ks.cfg 的擁有者與群組改回為root：
[root@study ~]# chown root:root initial-setup-ks.cfg
[root@study ~]# ls -l
-rw-r--r--. 1 root root 1864 May  4 18:01 initial-setup-ks.cfg
```

Tips
事實上，chown也可以使用『chown user.group file』，亦即在擁有者與群組間加上小數點『.』也行！不過很多朋友設定帳號時，喜歡在帳號當中加入小數點(例如 vbird.tsai這樣的帳號格式)，這就會造成系統的誤判了！所以我們比較建議使用冒號『:』來隔開擁有者與群組啦！此外，chown也能單純的修改所屬群組呢！例如『chown .sshd initial-setup-ks.cfg』就是修改群組～看到了嗎？就是那個小數點的用途！



知道如何改變檔案的群組與擁有者了，那麼什麼時候要使用chown或chgrp呢？或許你會覺得奇怪吧？是的，確實有時候需要變更檔案的擁有者的，最常見的例子就是在複製檔案給你之外的其他人時，我們使用最簡單的cp指令來說明好了：

```
[root@study ~]# cp 來源檔案 目的檔案
```

假設你今天要將.bashrc這個檔案拷貝成為.bashrc_test檔名，且是要給bin這個人，你可以這樣做：

```
[root@study ~]# cp .bashrc .bashrc_test
[root@study ~]# ls -al .bashrc*
-rw-r--r--. 1 root root 176 Dec 29 2013 .bashrc
-rw-r--r--. 1 root root 176 Jun 3 00:04 .bashrc_test    <==新檔案的屬性沒變
```

由於複製行為(cp)會複製執行者的屬性與權限，所以！怎麼辦？.bashrc_test還是屬於root所擁有，如此一來，即使你將檔案拿給bin這個使用者了，那他仍然無法修改的(看屬性/權限就知道了吧)，所以你就必須要將這個檔案的擁有者與群組修改一下囉！知道如何修改了吧？

改變權限, chmod

檔案權限的改變使用的是chmod這個指令，但是，權限的設定方法有兩種，分別可以使用數字或者是符號來進行權限的變更。我們就來談一談：

- 數字類型改變檔案權限

Linux檔案的基本權限就有九個，分別是owner/group/others三種身份各有自己的read/write/execute權限，先複習一下剛剛上面提到的資料：檔案的權限字元為：『-rwxrwxrwx』，這九個權限是三個三個一組的！其中，我們可以使用數字來代表各個權限，各權限的分數對照表如下：

```
r:4
w:2
x:1
```

每種身份(owner/group/others)各自的三個權限(r/w/x)分數是需要累加的，例如當權限為：[-rwxrwx---] 分數則是：

```
owner = rwx = 4+2+1 = 7
group = rwx = 4+2+1 = 7
others= --- = 0+0+0 = 0
```

所以等一下我們設定權限的變更時，該檔案的權限數字就是770啦！變更權限的指令chmod的語法是這樣的：

```
[root@study ~]# chmod [-R] xyz 檔案或目錄
選項與參數：
xyz : 就是剛剛提到的數字類型的權限屬性，為 rwx 屬性數值的相加。
-R : 進行遞迴(recursive)的持續變更，亦即連同次目錄下的所有檔案都會變更
```

舉例來說，如果要將.bashrc這個檔案所有的權限都設定啟用，那麼就下達：

```
[root@study ~]# ls -al .bashrc
-rw-r--r--. 1 root root 176 Dec 29 2013 .bashrc
[root@study ~]# chmod 777 .bashrc
[root@study ~]# ls -al .bashrc
-rwxrwxrwx. 1 root root 176 Dec 29 2013 .bashrc
```

那如果要將權限變成『-rwxr-xr--』呢？那麼權限的分數就成為[4+2+1][4+0+1][4+0+0]=754 囉！所以你需要下達『chmod 754 filename』。另外，在實際的系統運作中最常發生的一個問題就是，常常我們以vim編輯一個shell的文字批次檔後，他的權限通常是 -rw-rw-r-- 也就是664，如果要將該檔案變成可執行檔，並且不要讓其他人修改此一檔案的話，那麼就需要-rwxr-xr-x這樣的權限，此時就得要下達：『chmod 755 test.sh』的指令囉！

另外，如果有些檔案你不希望被其他人看到，那麼應該將檔案的權限設定為例如：『-rwxr-----』，那就下達『chmod 740

filename』吧！

例題：

將剛剛你的.bashrc這個檔案的權限修改回-rw-r--r--的情況吧！

答：

-rw-r--r--的分數是644，所以指令為：

chmod 644 .bashrc

• 符號類型改變檔案權限

還有一個改變權限的方法呦！從之前的介紹中我們可以發現，基本上就九個權限分別是(1)user (2)group (3)others三種身份啦！那麼我們就可以藉由u, g, o來代表三種身份的權限！此外，a 則代表 all 亦即全部的身份！那麼讀寫的權限就可以寫成r, w, x囉！也就是可以使用底下的方式來看：

chmod	u g o a	+(加入) -(除去) =(設定)	r w x	檔案或目錄
-------	------------------	-------------------------	-------------	-------

來實作一下吧！假如我們要『設定』一個檔案的權限成為『-rwxr-xr-x』時，基本上就是：

- user (u)：具有可讀、可寫、可執行的權限；
- group 與 others (g/o)：具有可讀與執行的權限。

所以就是：

```
[root@study ~]# chmod u=rwx,go=rx .bashrc  
# 注意喔！那個 u=rwx,go=rx 是連在一起的，中間並沒有任何空白字元！  
[root@study ~]# ls -al .bashrc  
-rwxr-xr-x. 1 root root 176 Dec 29 2013 .bashrc
```

那麼假如是『-rwxr-xr--』這樣的權限呢？可以使用『**chmod u=rwx,g=rx,o=r filename**』來設定。此外，如果我不知道原先的檔案屬性，而我只想要增加.bashrc這個檔案的每個人均可寫入的權限，那麼我就可以使用：

```
[root@study ~]# ls -al .bashrc  
-rwxr-xr-x. 1 root root 176 Dec 29 2013 .bashrc  
[root@study ~]# chmod a+w .bashrc  
[root@study ~]# ls -al .bashrc  
-rwxrwxrwx. 1 root root 176 Dec 29 2013 .bashrc
```

而如果是要將權限去掉而不更動其他已存在的權限呢？例如要拿掉全部人的可執行權限，則：

```
[root@study ~]# chmod a-x .bashrc  
[root@study ~]# ls -al .bashrc  
-rw-rw-rw-. 1 root root 176 Dec 29 2013 .bashrc  
[root@study ~]# chmod 644 .bashrc # 測試完畢得要改回來喔！
```

知道 +, -, = 的不同點了嗎？對啦！+ 與 - 的狀態下，只要是沒有指定到的項目，則該權限『不會被變動』，例如上面的例子中，由於僅以 - 拿掉 x 則其他兩個保持當時的值不變！多多實作一下，你就會知道如何改變權限囉！這在某些情況底下很好用的～舉例來說，你想要教一個朋友如何讓一個程式可以擁有執行的權限，但你又不知道該檔案原本的權限為何，此時，利用『**chmod a+x filename**』，就可以讓該程式擁有執行的權限了。是否很方便？

5.2.3 目錄與檔案之權限意義：

現在我們知道了Linux系統內檔案的三種身份(擁有者、群組與其他人)，知道每種身份都有三種權限(rwx)，已知道能夠使用chown, chgrp, chmod去修改這些權限與屬性，當然，利用ls -l去觀察檔案也沒問題。前兩小節也談到了這些檔案權限對於資料安全的重要性。那麼，這些檔案權限對於一般檔案與目錄檔案有何不同呢？有太大的不同啊！底下就讓鳥哥來說清楚，講明白！

權限對檔案的重要性

檔案是實際含有資料的地方，包括一般文字檔、資料庫內容檔、二進位可執行檔(binary program)等等。因此，權限對於檔案來說，他的意義是這樣的：

- **r (read)**：可讀取此一檔案的實際內容，如讀取文字檔的文字內容等；
- **w (write)**：可以編輯、新增或者是修改該檔案的內容(但不含刪除該檔案)；
- **x (eXecute)**：該檔案具有可以被系統執行的權限。

那個可讀(r)代表讀取檔案內容是還好瞭解，那麼可執行(x)呢？這裡你就必須要小心啦！因為在Windows底下一個檔案是否具有執行的能力是藉由『**副檔名**』來判斷的，例如：.exe, .bat, .com 等等，但是在Linux底下，**我們的檔案是否能被執行，則是藉由是否具有『x』這個權限來決定的！跟檔名是沒有絕對的關係的！**

至於最後一個w這個權限呢？當你對一個檔案具有w權限時，你可以具有寫入/編輯/新增/修改檔案的內容的權限，**但並不具備有刪除該檔案本身的權限！**對於檔案的rwx來說，主要都是針對『檔案的內容』而言，與檔案檔名的存在與否沒有關係喔！因為檔案記錄的是實際的資料嘛！

權限對目錄的重要性

檔案是存放實際資料的所在，那麼目錄主要是儲存啥玩意啊？**目錄主要的內容在記錄檔名清單，檔名與目錄有強烈的關連啦！**所以如果是針對目錄時，那個 r, w, x 對目錄是什麼意義呢？

- **r (read contents in directory)**：

表示具有讀取目錄結構清單的權限，所以當你具有讀取(r)一個目錄的權限時，表示你可以查詢該目錄下的檔名資料。所以你就可以利用 ls 這個指令將該目錄的內容列表顯示出來！

- **w (modify contents of directory)**：

這個可寫入的權限對目錄來說，是很了不起的！**因為他表示你具有異動該目錄結構清單的權限**，也就是底下這些權限：

- 建立新的檔案與目錄；
- 刪除已經存在的檔案與目錄(不論該檔案的權限為何！)
- 將已存在的檔案或目錄進行更名；
- 搬移該目錄內的檔案、目錄位置。

總之，目錄的w權限就與該目錄底下的檔名異動有關就對了啦！

- **x (access directory)**：

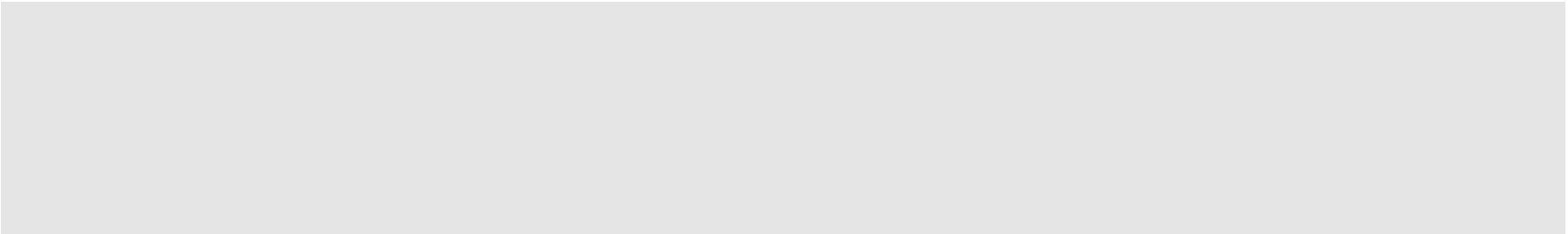
咦！目錄的執行權限有啥用途啊？目錄只是記錄檔名而已，總不能拿來執行吧？沒錯！目錄不可以被執行，**目錄的x代表的是使用者能否進入該目錄成為工作目錄**的用途！所謂的工作目錄(work directory)就是你目前所在的目錄啦！舉例來說，當你登入Linux時，你所在的家目錄就是你當下的工作目錄。而變換目錄的指令是『cd』(change directory)囉！

上面的東西這麼說，也太條列式～太教條了～有沒有清晰一點的說明啊？好～讓我們來思考一下人類社會使用的東西好了！現在假設『檔案是一堆文件資料夾』，所以你可能可以在上面寫/改一些資料。而『目錄是一堆抽屜』，因此你可以將資料夾分類放置到不同的抽屜去。因此抽屜最大的目的是拿出/放入資料夾喔！現在讓我們彙整一下資料：

元件	內容	疊代物件	r	w	x
檔案	詳細資料data	文件資料夾	讀到文件內容	修改文件內容	執行文件內容
目錄	檔名	可分類抽屜	讀到檔名	修改檔名	進入該目錄的權限(key)

根據上述的分析，你可以看到，對一般檔案來說，rwx 主要是針對『檔案的內容』來設計權限，對目錄來說，rwx則是針對『目錄內的檔名列表』來設計權限。其中最有趣的大概就屬目錄的 x 權限了！『檔名怎麼執行』？沒道理嘛！其實，這個 x 權限設計，就相當於『該目錄，也就是該抽屜的 "鑰匙" 』啦！沒有鑰匙你怎麼能夠打開抽屜呢？對吧！

大致的目錄權限概念是這樣，底下我們來看幾個範例，讓你瞭解一下啥是目錄的權限囉！



例題：

有個目錄的權限如下所示：

drwxr--r-- 3 root root 4096 Jun 25 08:35 .ssh

系統有個帳號名為vbird，這個帳號並沒有支援root群組，請問vbird對這個目錄有何權限？是否可切換到此目錄中？

答：

vbird對此目錄僅具有r的權限，因此vbird可以查詢此目錄下的檔名列表。因為vbird不具有x的權限，亦即 vbird 沒有這個抽屜的鑰匙啦！ 因此vbird並不能切換到此目錄內！（相當重要的概念！）

上面這個例題中因為vbird具有r的權限，因為是r乍看之下好像就具有可以進入此目錄的權限，其實那是錯的。能不能進入某一個目錄，只與該目錄的x權限有關啦！此外， 工作目錄對於指令的執行是非常重要的，如果你在某目錄下不具有x的權限，那麼你就無法切換到該目錄下，也就無法執行該目錄下的任何指令，即使你具有該目錄的r或w的權限。

很多朋友在架設網站的時候都會卡在一些權限的設定上，他們開放目錄資料給網際網路的任何人來瀏覽， 卻只開放r的權限，如上面的範例所示那樣，那樣的結果就是導致網站伺服器軟體無法到該目錄下讀取檔案(最多只能看到檔名)， 最終用戶總是無法正確的查閱到檔案的內容(顯示權限不足啊！)。要注意：要開放目錄給任何人瀏覽時，應該至少也要給予r及x的權限，但w權限不可隨便給！ 為什麼w不能隨便給，我們來看下一個例子：

例題：

假設有個帳號名為dmtsai，他的家目錄在/home/dmtsai/，dmtsai對此目錄具有[rwx]的權限。若在此目錄下有個名為the_root.data的檔案，該檔案的權限如下：

-rwx----- 1 root root 4365 Sep 19 23:20 the_root.data

請問dmtsai對此檔案的權限為何？可否刪除此檔案？

答：

如上所示，由於dmtsai對此檔案來說是『others』的身份，因此這個檔案他無法讀、無法編輯也無法執行，也就是說，他無法變動這個檔案的內容就是了。

但是由於這個檔案在他的家目錄下， 他在此目錄下具有rwx的完整權限，因此對於the_root.data這個『檔名』來說，他是能夠『刪除』的！ 結論就是，dmtsai這個用戶能夠刪除the_root.data這個檔案！

Tips

上述的例子解釋是這樣的，假設有個莫名其妙的人，拿著一個完全密封的資料夾放到你的辦公室抽屜中，因為完全密封你也打不開、看不到這個資料夾的內部資料(對檔案來說，你沒有權限)。但是因為這個資料夾是放在你的抽屜中，你當然可以拿出/放入任何資料在這個抽屜中(對目錄來說，你具有所有權限)。所以，情況就是：你打開抽屜、拿出這個沒辦法看到的資料夾、將他丟到走廊上的垃圾桶！搞定了 (順利刪除)！



還是看不太懂？有聽沒有懂喔！沒關係～我們底下就來設計一個練習， 讓你實際玩玩看，應該就能夠比較進入狀況啦！不過，由於很多指令我們還沒有教，所以底下的指令有的先瞭解即可，詳細的指令用法我們會在後面繼續介紹的。

- 先用root的身份建立所需要的檔案與目錄環境

我們用root的身份在所有人都可以工作的/tmp目錄中建立一個名為testing的目錄， 該目錄的權限為744且目錄擁有者為root。另外，在testing目錄下在建立一個空的檔案， 檔名亦為testing。建立目錄可用mkdir(make directory)，建立空檔案可用[touch\(下一章會說明\)](#)來處理。所以過程如下所示：

```
[root@study ~]# cd /tmp                                <==切換工作目錄到/tmp
[root@study tmp]# mkdir testing                        <==建立新目錄
[root@study tmp]# chmod 744 testing                    <==變更權限
[root@study tmp]# touch testing/testing                <==建立空的檔案
[root@study tmp]# chmod 600 testing/testing            <==變更權限
[root@study tmp]# ls -ald testing testing/testing
drwxr--r--. 2 root root 20 Jun  3 01:00 testing
-rw-----. 1 root root  0 Jun  3 01:00 testing/testing
# 仔細看一下，目錄的權限是 744 ，且所屬群組與使用者均是 root 喔！
# 那麼在這樣的情況底下，一般身份使用者對這個目錄/檔案的權限為何？
```

- 一般用戶的讀寫權限為何？觀察中

在上面的例子中，雖然目錄是744的權限設定，一般用戶應該能有 r 的權限， 但這樣的權限使用者能做啥事呢？由於鳥哥的系統中含有一個帳號名為 dmtsai 的，請再開另外一個終端機，使用 dmtsai 登入來操作底下的任務！

```
[dmtsai@study ~]$ cd /tmp
[dmtsai@study tmp]$ ls -l testing/
ls: cannot access testing/testing: Permission denied
total 0
????????? ? ? ? ? ? testing
# 雖然有告知權限不足，但因為具有 r 的權限可以查詢檔名。由於權限不足(沒有x)，所以會有一堆問號。
[dmtsai@study tmp]$ cd testing/
-bash: cd: testing/: Permission denied
# 因為不具有 x，所以當然沒有進入的權限啦！有沒有呼應前面的權限說明啊！
```

- 如果該目錄屬於用戶本身，會有什麼狀況？

上面的練習我們知道了只有r確實可以讓使用者讀取目錄的檔名列表，不過詳細的資訊卻還是讀不到的， 同時也不能將該目錄變成工作目錄(用 cd 進入該目錄之意)。那如果我們讓該目錄變成使用者的， 那麼使用者在這個目錄底下是否能夠刪除檔案呢？底下的練習做看看：

```
# 1. 先用 root 的身份來搞定 /tmp/testing 的屬性、權限設定：
[root@study tmp]# chown dmtsai /tmp/testing
[root@study tmp]# ls -ld /tmp/testing
drwxr--r--. 2 dmtsai root 20  6月  3 01:00 /tmp/testing # dmtsai 是具有全部權限的！

# 2. 再用 dmtsai 的帳號來處理一下 /tmp/testing/testing 這個檔案看看：
[dmtsai@study tmp]$ cd /tmp/testing
[dmtsai@study testing]$ ls -l <==確實是可以進入目錄
-rw-----. 1 root root 0 Jun  3 01:00 testing <==檔案不是dmtsai的！
[dmtsai@study testing]$ rm testing <==嘗試殺掉這個檔案看看！
rm: remove write-protected regular empty file `testing'? y
# 竟然可以刪除！這樣理解了嗎？！
```

透過上面這個簡單的步驟，你就可以清楚的知道， x 在目錄當中是與『能否進入該目錄』有關， 至於那個 w 則具有相當重要的權限，因為他可以讓使用者刪除、更新、新建檔案或目錄， 是個很重要的參數啊！這樣可以理解了嗎？！ ^_^！

使用者操作功能與權限

剛剛講這樣如果你還是搞不懂～沒關係，我們來處理個特殊的案例！假設兩個檔名，分別是底下這樣：

- /dir1/file1
- /dir2

假設你現在在系統使用 dmtsai 這個帳號，那麼這個帳號針對 /dir1, /dir1/file1, /dir2 這三個檔名來說，分別需要『哪些最小的權限』才能達成各項任務？ 鳥哥彙整如下，如果你看得懂，恭喜你，如果妳看不懂～沒關係～未來再來繼續學！

操作動作	/dir1	/dir1/file1	/dir2	重點
讀取 file1 內容	x	r	-	要能夠進入 /dir1 才能讀到裡面的文件資料！
修改 file1 內容	x	rw	-	能夠進入 /dir1 且修改 file1 才行！
執行 file1 內容	x	rx	-	能夠進入 /dir1 且 file1 能運作才行！
刪除 file1 檔案	wx	-	-	能夠進入 /dir1 具有目錄修改的權限即可！
將 file1 複製到 /dir2	x	r	wx	要能夠讀 file1 且能夠修改 /dir2 內的資料

你可能會問，上面的表格當中，很多時候 /dir1 都不必有 r 耶！為啥？我們知道 /dir1 是個目錄，也是個抽屜！那個抽屜的 r 代表『這個抽屜裡面有燈光』，所以你能看到的抽屜內的所有資料夾名稱 (非內容)。但你已經知道裡面的資料夾放在哪個地方，那，有沒有燈光有差嘛？你還是可以摸黑拿到該資料夾的！對吧！ 因此，上面很多動作中，你只要具有 x 即可！r 是非必備的！只是，沒有 r 的話，使用 [tab] 時，他就無法自動幫你補齊檔名了！這樣理解乎？

Tips

看了上面這個表格，你應該會覺得很可怕喔！因為，要讀一個檔案時，你得要具有『這個檔案所在目錄的 x 權限』才行！所以，通常要開放的目錄，至少會具備 rx 這兩個權限！現在你知道為啥了吧？



5.2.4 Linux檔案種類與副檔名

我們在基礎篇一直強調一個概念，那就是：任何裝置在Linux底下都是檔案，不僅如此，連資料溝通的介面也有專屬的檔案在負責～所以，你會瞭解到，Linux的檔案種類真的很多～除了前面提到的一般檔案(-)與目錄檔案(d)之外，還有哪些種類的檔案呢？

檔案種類：

我們在剛剛提到使用『ls -l』觀察到第一欄那十個字元中，第一個字元為檔案的類型。除了常見的一般檔案(-)與目錄檔案(d)之外，還有哪些種類的檔案類型呢？

- **正規檔案(regular file)：**

就是一般我們在進行存取的類型的檔案，在由ls -al所顯示出來的屬性方面，第一個字元為[-]，例如[-rwxrwxrwx]。另外，依照檔案的內容，又大略可以分為：

- **純文字檔(ASCII)：**這是Linux系統中最多的一種檔案類型囉，稱為純文字檔是因為內容為我們人類可以直接讀到的資料，例如數字、字母等等。幾乎只要我們可以用來做為設定的檔案都屬於這一種檔案類型。舉例來說，你可以下達『cat ~/.bashrc』就可以看到該檔案的內容。(cat是將一個檔案內容讀出來的指令)
- **二進位檔(binary)：**還記得我們在『[第零章、計算機概論](#)』裡面的[軟體程式的運作](#)中提過，我們的系統其實僅認識且可以執行二進位檔案(binary file)吧？沒錯～你的Linux當中的可執行檔(scripts, 文字型批次檔不算)就是這種格式的啦～舉例來說，剛剛下達的指令cat就是一個binary file。
- **資料格式檔(data)：**有些程式在運作的過程當中會讀取某些特定格式的檔案，那些特定格式的檔案可以被稱為資料檔(data file)。舉例來說，我們的Linux在使用者登入時，都會將登錄的資料記錄在/var/log/wtmp那個檔案內，該檔案是一個data file，他能夠透過last這個指令讀出來！但是使用cat時，會讀出亂碼～因為他是屬於一種特殊格式的檔案。瞭乎？

- **目錄(directory)：**

就是目錄囉～第一個屬性為[d]，例如[drwxrwxrwx]。

- **連結檔(link)：**

就是類似Windows系統底下的捷徑啦！第一個屬性為[l](英文L的小寫)，例如[lrwxrwxrwx]；

- **設備與裝置檔(device)：**

與系統周邊及儲存等相關的一些檔案，通常都集中在/dev這個目錄之下！通常又分為兩種：

- **區塊(block)設備檔：**就是一些儲存資料，以提供系統隨機存取的周邊設備，舉例來說，硬碟與軟碟等就是啦！你可以隨機的在硬碟的不同區塊讀寫，這種裝置就是區塊裝置囉！你可以自行查一下/dev/sda看看，會發現第一個屬性為[b]喔！
- **字元(character)設備檔：**亦即是一些序列埠的周邊設備，例如鍵盤、滑鼠等等！這些設備的特色就是『一次性讀取』的，不能夠截斷輸出。舉例來說，你不可能讓滑鼠『跳到』另一個畫面，而是『連續性滑動』到另一個地方啊！第一個屬性為[c]。

- **資料接口檔(sockets)：**

既然被稱為資料接口檔，想當然爾，這種類型的檔案通常被用在網路上的資料承接了。我們可以啟動一個程式來監聽用戶端的要求，而用戶端就可以透過這個socket來進行資料的溝通了。第一個屬性為[s]，最常在/run或/tmp這些個目錄中看到這種檔案類型了。

- **資料輸送檔(FIFO, pipe)：**

FIFO也是一種特殊的檔案類型，他主要的目的在解決多個程序同時存取一個檔案所造成的錯誤問題。FIFO是first-in-first-out的縮寫。第一個屬性為[p]。

除了設備檔是我們系統中很重要的檔案，最好不要隨意修改之外(通常他也不會讓你修改的啦！)，另一個比較有趣的檔案就是連結檔。如果你常常將應用程式捉到桌面來的話，你就應該知道在Windows底下有所謂的『捷徑』。同樣的，你可以將linux下的連結檔簡單的視為一個檔案或目錄的捷徑。至於socket與FIFO檔案比較難理解，因為這兩個咚咚與程序(process)比較有關係，這個等到未來你瞭解process之後，再回來查閱吧！此外，你也可以透過man fifo及man socket來查閱系統上的說明！

Linux檔案副檔名：

基本上，Linux的檔案是沒有所謂的『副檔名』的，我們剛剛就談過，[一個Linux檔案能不能被執行](#)，與他的第一欄的十個屬性有關，與檔名根本一點關係也沒有。這個觀念跟Windows的情況不相同喔！在Windows底下，能被執行的檔案副檔名通常是 .com .exe .bat 等等，而在Linux底下，只要你的權限當中具有x的話，例如[-rwxr-xr-x] 即代表這個檔案具有可以被執行的能力喔！

Tips

具有『可執行的權限』以及『具有可執行的程式碼』是兩回事！在Linux底下，你可以讓一個文字檔，例如我們之前寫的 text.txt 具有『可執行的權限』(加入 x 權限即可)，但是這個檔案明顯的無法執行，因為他不具備可執行的程式碼！而如果你將上面提到的 cat 這個可以執行的指令，將他的 x 拿掉，那麼 cat 將無法被你執行！



不過，可以被執行跟可以執行成功是不一樣的～舉例來說，在root家目錄下的 initial-setup-ks.cfg 是一個純文字檔，如果經由修改權限成為 -rwxrwxrwx 後，這個檔案能夠真的執行成功嗎？當然不行～因為他的內容根本就沒有可以執行的資料。所以說，這個x代表這個檔案具有可執行的能力，但是能不能執行成功，當然就得要看該檔案的內容囉～

雖然如此，不過我們仍然希望可以藉由副檔名來瞭解該檔案是什麼東西，所以，通常我們還是會以適當的副檔名來表示該檔案是什麼種類的。底下有數種常用的副檔名：

- *.sh：腳本或批次檔 (scripts)，因為批次檔為使用shell寫成的，所以副檔名就編成 .sh 囉；
- *.Z, *.tar, *.tar.gz, *.zip, *.tgz：經過打包的壓縮檔。這是因為壓縮軟體分別為 gunzip, tar 等等的，由於不同的壓縮軟體，而取其相關的副檔名囉！
- *.html, *.php：網頁相關檔案，分別代表 HTML 語法與 PHP 語法的網頁檔案囉！.html 的檔案可使用網頁瀏覽器來直接開啟，至於 .php 的檔案，則可以透過 client 端的瀏覽器來 server 端瀏覽，以得到運算後的網頁結果呢！

基本上，Linux系統上的檔名真的只是讓你瞭解該檔案可能的用途而已，真正的執行與否仍然需要權限的規範才行！例如雖然有一個檔案為可執行檔，如常見的/bin/lsh這個顯示檔案屬性的指令，不過，如果這個檔案的權限被修改成無法執行時，那麼ls就變成不能執行囉！

上述的這種問題最常發生在檔案傳送的過程中。例如你在網路上下載一個可執行檔，但是偏偏在你的Linux系統中就是無法執行！呵呵！那麼就是可能檔案的屬性被改變了！不要懷疑，從網路上傳送到你的Linux系統中，檔案的屬性與權限確實是會被改變的喔！

Linux檔名長度限制(註1)：

在Linux底下，使用傳統的Ext2/Ext3/Ext4檔案系統以及近來被 CentOS 7 當作預設檔案系統的 xfs 而言，針對檔案的檔名長度限制為：

- 單一檔案或目錄的最大容許檔名為 255bytes，以一個 ASCII 英文佔用一個 bytes 來說，則大約可達 255 個字元長度。若是以每個中文字佔用 2bytes 來說，最大檔名就是大約在 128 個中文字之譜！

是相當長的檔名喔！我們希望Linux的檔案名稱可以一看就知道該檔案在幹嘛的，所以檔名通常是很長很長！而用慣了Windows的人可能會受不了，因為檔案名稱通常真的都很長，對於用慣Windows而導致打字速度不快的朋友來說，嗯！真的是很困擾.....不過，只得勸你好好的加強打字的訓練囉！

Linux檔案名稱的限制：

由於Linux在文字介面下的一些指令操作關係，一般來說，你在設定Linux底下的檔案名稱時，最好可以避免一些特殊字元比較好！例如底下這些：

```
* ? > < ; & ! [ ] | \ ' " ` ( ) { }
```

因為這些符號在文字介面下，是有特殊意義的！另外，檔案名稱的開頭為小數點『.』時，代表這個檔案為『隱藏檔』喔！同時，由於指令下達當中，常常會使用到 -option 之類的選項，所以你最好也避免將檔案檔名的開頭以 - 或 + 來命名啊！

5.3 Linux目錄配置

在瞭解了每個檔案的相關種類與屬性，以及瞭解了如何更改檔案屬性/權限的相關資訊後，再來要瞭解的就是，為什麼每套Linux distributions他們的設定檔啊、執行檔啊、每個目錄內放置的咚咚啊，其實都差不多？原來是有一套標準依據的哩！我們底下就來瞧一瞧。

5.3.1 Linux目錄配置的依據--FHS

因為利用Linux來開發產品或distributions的社群/公司與個人實在太多了，如果每個人都用自己的想法來配置檔案放置的目錄，那麼將可能造成很多管理上的困擾。你能想像，你進入一個企業之後，所接觸到的Linux目錄配置方法竟然跟你以前學的完全不同嗎？很難想像吧～所以，後來就有所謂的[Filesystem Hierarchy Standard \(FHS\)](#)標準的出爐了！

根據FHS([註2](#))的標準文件指出，他們的主要目的是希望讓使用者可以瞭解到已安裝軟體通常放置於那個目錄下，所以他們希望獨立的軟體開發商、作業系統製作者、以及想要維護系統的使用者，都能夠遵循FHS的標準。也就是說，FHS的重點在於規範每個特定的目錄下應該要放置什麼樣子的資料而已。這樣做好處非常多，因為Linux作業系統就能夠在既有的面貌下(目錄架構不變)發展出開發者想要的獨特風格。

事實上，FHS是根據過去的經驗一直再持續的改版的，FHS依據檔案系統使用的頻繁與否與是否允許使用者隨意更動，而將目錄定義成為四種交互作用的形態，用表格來說有點像底下這樣：

	可分享的(shareable)	不可分享的(unshareable)
不變的(static)	/usr (軟體放置處)	/etc (設定檔)
	/opt (第三方協力軟體)	/boot (開機與核心檔)
可變動的(variable)	/var/mail (使用者郵件信箱)	/var/run (程序相關)
	/var/spool/news (新聞群組)	/var/lock (程序相關)

上表中的目錄就是一些代表性的目錄，該目錄底下所放置的資料在底下會談到，這裡先略過不談。我們要瞭解的是，什麼是那四個類型？

- **可分享的**：可以分享給其他系統掛載使用的目錄，所以包括執行檔與使用者的郵件等資料，是能夠分享給網路上其他主機掛載用的目錄；
- **不可分享的**：自己機器上面運作的裝置檔案或者是與程序有關的socket檔案等，由於僅與自身機器有關，所以當然就不適合分享給其他主機了。
- **不變的**：有些資料是不會經常變動的，跟隨著distribution而不變動。例如函式庫、文件說明檔、系統管理員所管理的主機服務設定檔等等；
- **可變動的**：經常改變的資料，例如登錄檔、一般用戶可自行收受的新聞群組等。

事實上，FHS針對目錄樹架構僅定義出三層目錄底下應該放置什麼資料而已，分別是底下這三個目錄的定義：

- **/ (root, 根目錄)**：與開機系統有關；
- **/usr (unix software resource)**：與軟體安裝/執行有關；
- **/var (variable)**：與系統運作過程有關。

為什麼要定義出這三層目錄呢？其實是有意義的喔！每層目錄底下所應該要放置的目錄也都又特定的規定喔！由於我們尚未介紹完整的Linux系統，所以底下的介紹你可能會看不懂！沒關係，先有個概念即可，等到妳將基礎篇全部看完後，就重頭將基礎篇再看一遍！到時候你就會豁然開朗啦！^_^

Tips

這個 root 在 Linux 裡面的意義真的很多很多～多到讓人搞不懂那是啥玩意兒。如果以『帳號』的角度來看，所謂的 root 指的是『系統管理員！』的身份，如果以『目錄』的角度來看，所謂的 root 意即指的是根目錄，就是 / 啦～要特別留意喔！



根目錄 (/) 的意義與內容：

根目錄是整個系統最重要的一個目錄，因為不但所有的目錄都是由根目錄衍生出來的，同時根目錄也與開機/還原/系統修復等動作有關。由於系統開機時需要特定的開機軟體、核心檔案、開機所需程式、函式庫等等檔案資料，若系統出現錯誤時，根目錄也必須要包含有能夠修復檔案系統的程序才行。因為根目錄是這麼的重要，所以在FHS的要求方面，他希望根目錄不要放在非常大的分割槽內，因為越大的分割槽妳會放入越多的資料，如此一來根目錄所在分割槽就可能有較多發生錯誤的機會。

因此FHS標準建議：根目錄(/)所在分割槽應該越小越好，且應用程式所安裝的軟體最好不要與根目錄放在同一個分割槽內，保持根目錄越小越好。如此不但效能較佳，根目錄所在的檔案系統也較不容易發生問題。

有鑑於上述的說明，因此FHS定義出根目錄(/)底下應該要有底下這些次目錄的存在才好，即使沒有實體目錄，FHS也希望至少有連結檔存在才好：

目錄	應放置檔案內容
第一部份：FHS 要求必須要存在的目錄	
/bin	系統有很多放置執行檔的目錄，但/bin比較特殊。因為/bin放置的是在單人維護模式下還能夠被操作的指令。在/bin底下的指令可以被root與一般帳號所使用，主要有：cat, chmod, chown, date, mv, mkdir, cp, bash等等常用的指令。
/boot	這個目錄主要在放置開機會使用到的檔案，包括Linux核心檔案以及開機選單與開機所需設定檔等等。Linux kernel 常用的檔名為：vmlinuz，如果使用的是grub2這個開機管理程式，則還會存在/boot/grub2/這個目錄喔！
/dev	在Linux系統上，任何裝置與周邊設備都是以檔案的型態存在於這個目錄當中的。你只要透過存取這個目錄底下的某個檔案，就等於存取某個裝置囉～ 比要重要的檔案有/dev/null, /dev/zero, /dev/tty, /dev/loop*, /dev/sd*等等
/etc	系統主要的設定檔幾乎都放置在這個目錄內，例如人員的帳號密碼檔、各種服務的啟始檔等等。一般來說，這個目錄下的各檔案屬性是可以讓一般使用者查閱的，但是只有root有權力修改。FHS建議不要放置可執行檔(binary)在這個目錄中喔。比較重要的檔案有：/etc/modprobe.d/, /etc/passwd, /etc/fstab, /etc/issue 等等。另外 FHS 還規範幾個重要的目錄最好要存在 /etc/ 目錄下喔： <ul style="list-style-type: none">/etc/opt(必要)：這個目錄在放置第三方協力軟體 /opt 的相關設定檔/etc/X11/(建議)：與 X Window 有關的各種設定檔都在這裡，尤其是 xorg.conf 這個 X Server 的設定檔。/etc/sgml/(建議)：與 SGML 格式有關的各項設定檔/etc/xml/(建議)：與 XML 格式有關的各項設定檔
/lib	系統的函式庫非常的多，而/lib放置的則是在開機時會用到的函式庫，以及在/bin或/sbin底下的指令會呼叫的函式庫而已。什麼是函式庫呢？妳可以將他想成是『外掛』，某些指令必須要有這些『外掛』才能夠順利完成程式的執行之意。另外 FHS 還要求底下的目錄必須要存在： <ul style="list-style-type: none">/lib/modules/：這個目錄主要放置可抽換式的核心相關模組(驅動程式)喔！
/media	media是『媒體』的英文，顧名思義，這個/media底下放置的就是可移除的裝置啦！包括軟碟、光碟、DVD等等裝置都暫時掛載於此。常見的檔名有：/media/floppy, /media/cdrom等等。
/mnt	如果妳想要暫時掛載某些額外的裝置，一般建議妳可以放置到這個目錄中。在古早時候，這個目錄的用途與/media相同啦！只是有了/media之後，這個目錄就用來暫時掛載用了。
/opt	這個是給第三方協力軟體放置的目錄。什麼是第三方協力軟體啊？舉例來說，KDE這個桌面管理系統是一個獨立的計畫，不過他可以安裝到Linux系統中，因此KDE的軟體就建議放置到此目錄下了。另外，如果妳想要自行安裝額外的軟體(非原本的distribution提供的)，那麼也能夠將你的軟體安裝到這裡來。不過，以前的Linux系統中，我們還是習慣放置在/usr/local目錄下呢！
/run	早期的 FHS 規定系統開機後所產生的各項資訊應該要放置到 /var/run 目錄下，新版的 FHS 則規範到 /run 底下。由於 /run 可以使用記憶體來模擬，因此效能上會好很多！
/sbin	Linux有非常多指令是用來設定系統環境的，這些指令只有root才能夠利用來『設定』系統，其他使用者最多只能用來『查詢』而已。放在/sbin底下的為開機過程中所需要的，裡面包括了開機、修復、還原系統所需要的指令。至於某些伺服器軟體程式，一般則放置到/usr/sbin/當中。至於本機自行安裝的軟體所產生的系統執行檔(system binary)，則放置到/usr/local/sbin/當中了。常見的指令包括：fdisk, fsck, ifconfig, mkfs等等。
/srv	srv可以視為『service』的縮寫，是一些網路服務啟動之後，這些服務所需要取用的資料目錄。常見的服務例如 WWW, FTP等等。舉例來說，WWW伺服器需要的網頁資料就可以放置在/srv/www/裡面。不過，系統的服務資料如果尚未要提供給網際網路任何人瀏覽的話，預設還是建議放置到 /var/lib 底下即可。
/tmp	這是讓一般使用者或者是正在執行的程序暫時放置檔案的地方。這個目錄是任何人都能夠存取的，所以你需要定期的清理一下。當然，重要資料不可放置在此目錄啊！因為FHS甚至建議在開機時，應該要將/tmp下的資料都刪除唷！
/usr	第二層 FHS 設定，後續介紹
/var	第二層 FHS 設定，主要為放置變動性的資料，後續介紹
第二部份：FHS 建議可以存在的目錄	
/home	這是系統預設的使用者家目錄(home directory)。在你新增一個一般使用者帳號時，預設的使用者家目錄都會規範到這裡來。比較重要的是，家目錄有兩種代號喔： <ul style="list-style-type: none">~：代表目前這個使用者的家目錄~dmtsai：則代表 dmtsai 的家目錄！
/lib<qual>	用來存放與 /lib 不同的格式的二進位函式庫，例如支援 64 位元的 /lib64 函式庫等

/root	系統管理員(root)的家目錄。之所以放在這裡，是因為如果進入單人維護模式而僅掛載根目錄時，該目錄就能夠擁有root的家目錄，所以我們會希望root的家目錄與根目錄放在同一個分割槽中。
-------	--

事實上FHS針對根目錄所定義的標準就僅有上面的咚咚，不過我們的Linux底下還有許多目錄你也需要瞭解一下的。 底下是幾個在Linux當中也是非常重要的目錄喔：

目錄	應放置檔案內容
/lost+found	這個目錄是使用標準的ext2/ext3/ext4檔案系統格式才會產生的一個目錄，目的在於當檔案系統發生錯誤時，將一些遺失的片段放置到這個目錄下。不過如果使用的是 xfs 檔案系統的話，就不會存在這個目錄了！
/proc	這個目錄本身是一個『虛擬檔案系統(virtual filesystem)』喔！他放置的資料都是在記憶體當中，例如系統核心、行程資訊(process)、周邊裝置的狀態及網路狀態等等。因為這個目錄下的資料都是在記憶體當中，所以本身不佔任何硬碟空間啊！比較重要的檔案例如： /proc/cpuinfo , /proc/dma , /proc/interrupts , /proc/ioports , /proc/net/* 等等。
/sys	這個目錄其實跟/proc非常類似，也是一個虛擬的檔案系統，主要也是記錄核心與系統硬體資訊較相關的資訊。包括目前已載入的核心模組與核心偵測到的硬體裝置資訊等等。這個目錄同樣不佔硬碟容量喔！

早期 Linux 在設計的時候，若發生問題時，救援模式通常僅掛載根目錄而已，因此有五個重要的目錄被要求一定要與根目錄放置在一起，那就是 `/etc`, `/bin`, `/dev`, `/lib`, `/sbin` 這五個重要目錄。現在許多的 Linux distributions 由於已經將許多非必要的檔案移出 `/usr` 之外了，所以 `/usr` 也是越來越精簡，同時因為 `/usr` 被建議為『即使掛載成為唯讀，系統還是可以正常運作』的模樣，所以救援模式也能同時掛載 `/usr` 喔！例如我們的這個 CentOS 7.x 版本在救援模式的情況下就是這樣。因此那個五大目錄的限制已經被打破了呦！例如 CentOS 7.x 就已經將 `/sbin`, `/bin`, `/lib` 通通移動到 `/usr` 底下了哩！

好了，談完了根目錄，接下來我們就來談談/usr以及/var囉！先看/usr裡面有些什麼東西：

/usr 的意義與內容：

依據FHS的基本定義，`/usr`裡面放置的資料屬於可分享的與不可變動的(`shareable`, `static`)，如果你知道如何透過網路進行分割槽的掛載(例如在伺服器篇會談到的[NFS伺服器](#))，那麼/usr確實可以分享給區域網路內的其他主機來使用喔！

很多讀者都會誤會usr為user的縮寫，其實usr是Unix Software Resource的縮寫，也就是『Unix作業系統軟體資源』所放置的目錄，而不是使用者的資料啦！這點要注意。FHS建議所有軟體開發者，應該將他們的資料合理的分別放置到這個目錄下的次目錄，而不要自行建立該軟體自己獨立的目錄。

因為是所有系統預設的軟體(distribution發佈者提供的軟體)都會放置到/usr底下，因此這個目錄有點類似Windows 系統的『`C:\Windows\` (當中的一部份) + `C:\Program files\`』這兩個目錄的綜合體，系統剛安裝完畢時，這個目錄會佔用最多的硬碟容量。一般來說，`/usr`的次目錄建議有底下這些：

目錄	應放置檔案內容
第一部份：FHS 要求必須要存在的目錄	
/usr/bin/	所有一般用戶能夠使用的指令都放在這裡！目前新的 CentOS 7 已經將全部的使用者指令放置於此，而使用連結檔的方式將 <code>/bin</code> 連結至此！也就是說， <code>/usr/bin</code> 與 <code>/bin</code> 是一模一樣了！另外，FHS 要求在此目錄下不應該有子目錄！
/usr/lib/	基本上，與 <code>/lib</code> 功能相同，所以 <code>/lib</code> 就是連結到此目錄中的！
/usr/local/	系統管理員在本機自行安裝自己下載的軟體(非distribution預設提供者)，建議安裝到此目錄，這樣會比較便於管理。舉例來說，你的distribution提供的軟體較舊，你想安裝較新的軟體但又不想移除舊版，此時你可以將新版軟體安裝於/usr/local/目錄下，可與原先的舊版軟體有分別啦！你可以自行到usr/local去看看，該目錄下也是具有bin, etc, include, lib...的次目錄喔！
/usr/sbin/	非系統正常運作所需要的系統指令。最常見的就是某些網路伺服器軟體的服務指令(daemon)囉！不過基本功能與 <code>/sbin</code> 也差不多，因此目前 <code>/sbin</code> 就是連結到此目錄中的。
/usr/share/	主要放置唯讀架構的資料檔案，當然也包括共享文件。在這個目錄下放置的資料幾乎是不分硬體架構均可讀取的資料，因為幾乎都是文字檔案嘛！在此目錄下常見的還有這些次目錄： <ul style="list-style-type: none"><code>/usr/share/man</code>：線上說明文件<code>/usr/share/doc</code>：軟體雜項的文件說明<code>/usr/share/zoneinfo</code>：與時區有關的時區檔案
第二部份：FHS 建議可以存在的目錄	

/usr/games/	與遊戲比較相關的資料放置處
/usr/include/	c/c++等程式語言的檔頭(header)與包含檔(include)放置處，當我們以tarball方式 (*.tar.gz 的方式安裝軟體)安裝某些資料時，會使用到裡頭的許多包含檔喔！
/usr/libexec/	某些不被一般使用者慣用的執行檔或腳本(script)等等，都會放置在此目錄中。例如大部分的 X 視窗底下的操作指令，很多都是放在此目錄下的。
/usr/lib<qual>/	與 /lib<qual>/功能相同，因此目前 /lib<qual> 就是連結到此目錄中
/usr/src/	一般原始碼建議放置到這裡，src有source的意思。至於核心原始碼則建議放置到/usr/src/linux/目錄下。

/var 的意義與內容：

如果/usr是安裝時會佔用較大硬碟容量的目錄，那麼/var就是在系統運作後才會漸漸佔用硬碟容量的目錄。因為/var目錄主要針對常態性變動的檔案，包括快取(cache)、登錄檔(log file)以及某些軟體運作所產生的檔案，包括程序檔案(lock file, run file)，或者例如MySQL資料庫的檔案等等。常見的次目錄有：

目錄	應放置檔案內容
第一部份：FHS 要求必須要存在的目錄	
/var/cache/	應用程式本身運作過程中會產生的一些暫存檔；
/var/lib/	程式本身執行的過程中，需要使用到的資料檔案放置的目錄。在此目錄下各自的軟體應該要有各自的目錄。舉例來說，MySQL的資料庫放置到/var/lib/mysql/而rpm的資料庫則放到/var/lib/rpm去！
/var/lock/	某些裝置或者是檔案資源一次只能被一個應用程式所使用，如果同時有兩個程式使用該裝置時，就可能產生一些錯誤的狀況，因此就得要將該裝置上鎖(lock)，以確保該裝置只會給單一軟體所使用。舉例來說，燒錄機正在燒錄一塊光碟，你想一下，會不會有兩個人同時在使用一個燒錄機燒片？如果兩個人同時燒錄，那片子寫入的是誰的資料？所以當第一個人在燒錄時該燒錄機就會被上鎖，第二個人就得要該裝置被解除鎖定(就是前一個人用完了)才能夠繼續使用囉。目前此目錄也已經挪到 /run/lock 中！
/var/log/	重要到不行！這是登錄檔放置的目錄！裡面比較重要的檔案如/var/log/messages, /var/log/wtmp(記錄登入者的資訊)等。
/var/mail/	放置個人電子郵件信箱的目錄，不過這個目錄也被放置到/var/spool/mail/目錄中！通常這兩個目錄是互為連結檔啦！
/var/run/	某些程式或者是服務啟動後，會將他們的PID放置在這個目錄下喔！至於PID的意義我們會在後續章節提到的。與 /run 相同，這個目錄連結到 /run 去了！
/var/spool/	這個目錄通常放置一些佇列資料，所謂的『佇列』就是排隊等待其他程式使用的資料啦！這些資料被使用後通常都會被刪除。舉例來說，系統收到新信會放置到/var/spool/mail/中，但使用者收下該信件後該封信原則上就會被刪除。信件如果暫時寄不出去會被放到/var/spool/mqueue/中，等到被送出後就被刪除。如果是工作排程資料(crontab)，就會被放置到/var/spool/cron/目錄中！

建議在你讀完整個基礎篇之後，可以挑戰FHS官方英文文件(參考本章[參考資料](#))，相信會讓你對於Linux作業系統的目錄有更深入的瞭解喔！

針對FHS，各家distributions的異同，與 CentOS7 的變化

由於FHS僅是定義出最上層(/)及次層(/usr, /var)的目錄內容應該要放置的檔案或目錄資料，因此，在其他次目錄層級內，就可以隨開發者自行來配置了。舉例來說，CentOS的網路設定資料放在 /etc/sysconfig/network-scripts/ 目錄下，但是SuSE則是將網路放置在 /etc/sysconfig/network/ 目錄下，目錄名稱可是不同的呢！不過只要記住大致的FHS標準，差異性其實有限啦！

此外，CentOS 7 在目錄的編排上與過去的版本不同喔！本節稍早之前已經有介紹過，這裡做個彙整。比較大的差異在於將許多原本應該要在根目錄 (/) 裡面的目錄，將他內部資料全部挪到 /usr 裡面去，然後進行連結設定！包括底下這些：

- /bin --> /usr/bin
- /sbin --> /usr/sbin
- /lib --> /usr/lib
- /lib64 --> /usr/lib64
- /var/lock --> /run/lock
- /var/run --> /run

5.3.2 目錄樹(directory tree)

另外，在Linux底下，所有的檔案與目錄都是由根目錄開始的！那是所有目錄與檔案的源頭～然後再一個一個的分支下來，有點像是樹枝狀啊～因此，我們也稱這種目錄配置方式為：『目錄樹(directory tree)』這個目錄樹有什麼特性呢？他主要的特性有：

- 目錄樹的啟始點為根目錄 (/ , root) ；
- 每一個目錄不止能使用本地端的 partition 的檔案系統，也可以使用網路上的 filesystem 。舉例來說， 可以利用 Network File System (NFS) 伺服器掛載某特定目錄等。
- 每一個檔案在此目錄樹中的檔名(包含完整路徑)都是獨一無二的。

好，談完了FHS的標準之後，實際來看看CentOS在根目錄底下會有什麼樣子的資料吧！我們可以下達以下的指令來查詢：

```
[dmtsai@study ~]$ ls -l /
lrwxrwxrwx.    1 root root    7 May  4 17:51 bin -> usr/bin
dr-xr-xr-x.    4 root root 4096 May  4 17:59 boot
drwxr-xr-x.   20 root root 3260 Jun  2 19:27 dev
drwxr-xr-x.  131 root root 8192 Jun  2 23:51 etc
drwxr-xr-x.    3 root root   19 May  4 17:56 home
lrwxrwxrwx.    1 root root    7 May  4 17:51 lib -> usr/lib
lrwxrwxrwx.    1 root root    9 May  4 17:51 lib64 -> usr/lib64
drwxr-xr-x.    2 root root    6 Jun 10  2014 media
drwxr-xr-x.    2 root root    6 Jun 10  2014 mnt
drwxr-xr-x.    3 root root   15 May  4 17:54 opt
dr-xr-xr-x.  154 root root    0 Jun  2 11:27 proc
dr-xr-x---.    5 root root 4096 Jun  3 00:04 root
drwxr-xr-x.   33 root root   960 Jun  2 19:27 run
lrwxrwxrwx.    1 root root    8 May  4 17:51/sbin -> usr/sbin
drwxr-xr-x.    2 root root    6 Jun 10  2014 srv
dr-xr-xr-x.   13 root root    0 Jun  2 19:27 sys
drwxrwxrwt.   12 root root 4096 Jun  3 19:48 tmp
drwxr-xr-x.   13 root root 4096 May  4 17:51 usr
drwxr-xr-x.   22 root root 4096 Jun  2 19:27 var
```

上述目錄相關的介紹都在上一個小節，要記得回去查看看。如果我們將整個目錄樹以圖示的方法來顯示，並且將較為重要的檔案資料列出來的話，那麼目錄樹架構有點像這樣：

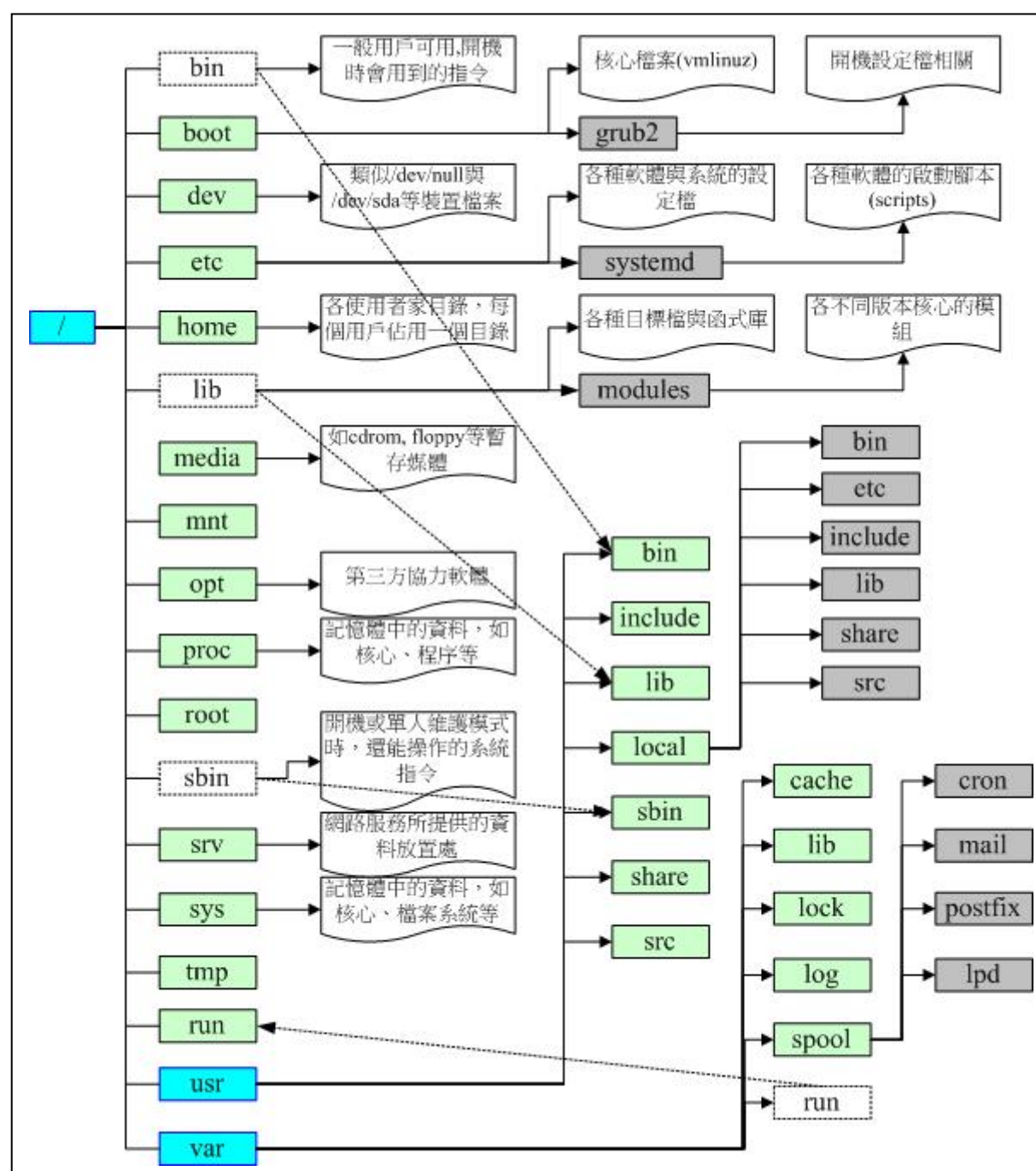


圖5.3.1、目錄樹架構示意圖

鳥哥只有就各目錄進行簡單的解釋，看看就好，詳細的解釋請回到剛剛說明的表格中去查閱喔！看完了FHS標準之後，現在回到[第二章裡面去看看安裝前Linux規劃的分割情況](#)，對於當初為何需要分割為這樣的情況，有點想法了嗎？^_^。根據FHS的定義，妳最好能夠將/var獨立出來，這樣對於系統的資料還有一些安全性的保護呢！因為至少/var死掉時，你的根目錄還會活著嘛！還能夠進入救援模式啊！

5.3.3 絕對路徑與相對路徑

除了需要特別注意的FHS目錄配置外，在檔名部分我們也要特別注意喔！因為根據檔名寫法的不同，也可將所謂的路徑(path)定義為絕對路徑(absolute)與相對路徑(relative)。這兩種檔名/路徑的寫法依據是這樣的：

- **絕對路徑**：由根目錄(/)開始寫起的檔名或目錄名稱，例如 /home/dmtsai/.bashrc；
- **相對路徑**：相對於目前路徑的檔名寫法。例如 ./home/dmtsai 或 ../../home/dmtsai/ 等等。反正開頭不是 / 就屬於相對路徑的寫法

而你必須要瞭解，相對路徑是以『你當前所在路徑的相對位置』來表示的。舉例來說，你目前在 /home 這個目錄下，如果想要進入 /var/log 這個目錄時，可以怎麼寫呢？

1. cd /var/log (absolute)
2. cd ../var/log (relative)

因為你在 /home 底下，所以要回到上一層 (../) 之後，才能繼續往 /var 來移動的！特別注意這兩個特殊的目錄：

- **.**：代表當前的目錄，也可以使用 ./ 來表示；
- **..**：代表上一層目錄，也可以 ../ 來代表。

這個.與..目錄概念是很重要的，你常常會看到 cd .. 或 ./command 之類的指令下達方式，就是代表上一層與目前所在目錄的工作狀態喔！很重要的吶！

例題：

如何先進入/var/spool/mail/目錄，再進入到/var/spool/cron/目錄內？

答：

由於/var/spool/mail與/var/spool/cron是同樣在/var/spool/目錄中，因此最簡單的指令下達方法為：

1. cd /var/spool/mail
2. cd ../cron

如此就不需要在由根目錄開始寫起了。這個相對路徑是非常有幫助的！尤其對於某些軟體開發商來說。一般來說，軟體開發商會將資料放置到/usr/local/裡面的各相對目錄，妳可以參考圖5.3.1的相對位置。但如果使用者想要安裝到不同目錄呢？就得要使用相對路徑囉！^_^

例題：

網路文件常常提到類似『./run.sh』之類的資料，這個指令的意義為何？

答：

由於指令的執行需要變數(bash章節才會提到)的支援，若你的執行檔放置在本目錄，並且本目錄並非正規的執行檔目錄(/bin,/usr/bin等為正規)，此時要執行指令就得要嚴格指定該執行檔。『./』代表『本目錄』的意思，所以『./run.sh』代表『執行本目錄下，名為run.sh的檔案』囉！

5.3.4 CentOS 的觀察

如同在[第一章](#)談到的 Linux distribution 的差異性，除了 FHS 之外，還有個 Linux Standard Base (LSB) 的標準是可以依循的！我們可以簡單的使用 ls 來查看 FHS 規範的目錄是否正確的存在於你的 Linux 系統中，那麼 Linux 核心、LSB 的標準又該如何查閱呢？基本上，LSB 團隊是有列出正確支援 LSB 標準的 distribution 在如下的網頁中：

- https://www.linuxbase.org/lsb-cert/productdir.php?by_lsb

不過，如果你要知道確切的核心與 LSB 所需求的幾種重要的標準的話，恐怕就得要使用諸如 uname 與 lsb_release 等指令來查閱了。不過，這個 lsb_release 指令已經不是預設安裝的軟體了，所以你得要自己安裝該軟體才行。因為我們尚未講到網路與掛載等動作，所以底下的安裝流程在你的機器上面應該是無法執行的(除非你確實可以連上 Internet 才行!)，因為 CentOS7 在這個軟體上面實

在有太多的相依軟體，所以無法單純使用 rpm 來安裝！若你有公開的網路，那麼底下的指令才能夠順利運作！

```
# 1. 透過 uname 檢查 Linux 核心與作業系統的位元版本
[dmtsai@study ~]$ uname -r    # 查看核心版本
3.10.0-229.el7.x86_64
[dmtsai@study ~]$ uname -m    # 查看作業系統的位元版本
x86_64

# 2. 假設你的 CentOS 7 確實有網路可以使用的情況下（要用 root 的身份）
[root@study ~]# yum install redhat-lsb    # yum 的用法後面章節才會介紹
.....(前面省略).....
Install 1 Package (+85 Dependent packages)
Upgrade ( 4 Dependent packages)

Total size: 47 M
Total download size: 31 M
Is this ok [y/d/N]: y
.....(後面省略).....
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importing GPG key 0xF4A80EB5:
  Userid      : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"
  Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
  Package     : centos-release-7-0.1406.el7.centos.2.3.x86_64 (@anaconda)
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Is this ok [y/N]: y
.....(後面省略).....

[root@study ~]# lsb_release -a
LSB Version:      :core-4.1-amd64:core-4.1-noarch:cxx-4.1-amd64:cxx-4.1-noarch:
desktop-4.1-amd64:desktop-4.1-noarch:languages-4.1-amd64:languages-4.1-noarch:
printing-4.1-amd64:printing-4.1-noarch  # LSB 的相關版本
Distributor ID: CentOS
Description:      CentOS Linux release 7.0.1406 (Core)
Release:          7.0.1406
Codename:         Core
```

這個 lsb_release 的東西大家先看看就好，因為有牽涉到後面的 yum 軟體安裝的東西，這部份我們還沒有談到啊～而且如果你現在就直接安裝，未來我們談網路與軟體的階段時，恐怕有些地方會跟我們的測試機環境不同～所以...先看看就好喔！ ^_^

Tips

在這裡要跟大家說抱歉，因為不想要破壞整體測試機器的環境，所以鳥哥使用了另一部虛擬機來安裝 redhat-lsb 這套軟體，而另一部虛擬機是透過 CentOS 7.0 而非 CentOS 7.1 的版本，因此你應該會發現到上面使用 lsb_release 指令的輸出中，竟然出現了 7.0.1406 的東東～真是不好意思～



5.4 重點回顧

- Linux的每個檔案中，可分別給予擁有者、群組與其他人三種身份個別的 rwx 權限；
- 群組最有用的功能之一，就是當你在團隊開發資源的時候，且每個帳號都可以有多個群組的支援；
- 利用ls -l顯示的檔案屬性中，第一個欄位是檔案的權限，共有十個位元，第一個位元是檔案類型，接下來三個為一組共三組，為擁有者、群組、其他人的權限，權限有r,w,x三種；
- 如果檔名之前多一個『.』，則代表這個檔案為『隱藏檔』；
- 若需要root的權限時，可以使用 su - 這個指令來切換身份。處理完畢則使用 exit 離開 su 的指令環境。
- 更改檔案的群組支援可用chgrp，修改檔案的擁有者可用chown，修改檔案的權限可用chmod
- chmod修改權限的方法有兩種，分別是符號法與數字法，數字法中r,w,x分數為4,2,1；
- 對檔案來講，權限的效能為：
 - r：可讀取此一檔案的實際內容，如讀取文字檔的文字內容等；
 - w：可以編輯、新增或者是修改該檔案的內容(但不含刪除該檔案)；
 - x：該檔案具有可以被系統執行的權限。
- 對目錄來說，權限的效能為：
 - r (read contents in directory)
 - w (modify contents of directory)
 - x (access directory)
- 要開放目錄給任何人瀏覽時，應該至少也要給予r及x的權限，但w權限不可隨便給；
- 能否讀取到某個檔案內容，跟該檔案所在的目錄權限也有關係 (目錄至少需要有 x 的權限)。

- Linux檔名的限制為：單一檔案或目錄的最大容許檔名為 255 個英文字元或 128 個中文字元；
- 根據FHS的官方文件指出，他們的主要目的是希望讓使用者可以瞭解到已安裝軟體通常放置於那個目錄下
- FHS訂定出來的四種目錄特色為：shareable, unshareable, static, variable等四類；
- FHS所定義的三層主目錄為：/, /var, /usr三層而已；
- 絕對路徑檔名為從根目錄 / 開始寫起，否則都是相對路徑的檔名。

5.5 本章練習

(要看答案請將滑鼠移動到『答：』 底下的空白處，按下左鍵圈選空白處即可察看)

- 早期的 Unix 系統檔名最多允許 14 個字元，而新的 Unix 與 Linux 系統中，檔名最多可以容許幾個字元？
- 當一個一般檔案權限為 -rwxrwxrwx 則表示這個檔案的意義為？
- 我需要將一個檔案的權限改為 -rwxr-xr-- 請問該如何下達指令？
- 若我需要更改一個檔案的擁有者與群組，該用什麼指令？
- 請問底下的目錄與主要放置什麼資料：
/etc/, /boot, /usr/bin, /bin, /usr/sbin, /sbin, /dev, /var/log, /run
- 若一個檔案的檔名開頭為『.』，例如 .bashrc 這個檔案，代表什麼？另外，如何顯示出這個檔名與他的相關屬性？

5.6 參考資料與延伸閱讀

- 註1：各種檔案系統的檔名長度限制，維基百科：http://en.wikipedia.org/wiki/Comparison_of_file_systems
- 註2：FHS 標準的相關說明：
維基百科簡易說明：http://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard
FHS 2.3 (2004 年版)的標準文件：<http://www.pathname.com/fhs/pub/fhs-2.3.html>
FHS 3.0 (2015 年版)的標準文件：http://refspecs.linuxfoundation.org/FHS_3.0/fhs-3.0.pdf
- 關於 Journaling 日誌式文章的相關說明 <http://www.linuxplanet.com/linuxplanet/reports/3726/1/>

修改歷史：

- 2002/07/18：第一次完成
- 2003/02/06：重新編排與加入FAQ
- 2005/06/28：將舊的資料移動到 [這裡](#)
- 2005/07/15：呼呼～終於改完成了～這次的修訂當中，加入了 FHS 的說明，希望大家能夠比較清楚 Linux 的目錄配置！
- 2005/08/05：修訂了最大檔名字元，應該是 255 才對！另外，加入了『檔名限制』的部分！
- 2005/09/03：修訂了目錄權限相關的說明，將原本僅具有 r 卻寫成無法使用 ls 瀏覽的說明資料移除！
- 2008/09/08：舊的針對FC4所寫的文章移動到 [此處](#)
- 2008/09/20：針對FHS加強說明了一下，分為/, /usr, /var三層來個別說明！並非抄襲官網的資料而已喔！
- 2008/09/23：經過一場大感冒，停工了四、五天，終於還是給他完工了！^ ^
- 2008/10/21：原本的第四小節 Linux 的檔案系統，因為與第八章重複性太高，將他移除了！
- 2009/08/01：加入了 lsb_release 的相關說明！
- 2009/08/18：調整一下顯示的情況，使得更易讀～
- 2015/06/02：將原本基於 CentOS 5.x 撰寫的就文章放在 [這裡](#) 了喔！

2002/02/18以來統計人數

3100759

基礎學習篇文件

[各版本彙整說明](#)

[CentOS 7.x](#)

[CentOS 5.x](#)

[Fedora Core 4](#)

[Mandrake 9](#)

[Red Hat 6.1](#)

[其他舊文件](#)

基礎訓練篇文件

[各版本彙整說明](#)

[CentOS 8.x - 基礎訓練](#)

[CentOS 7.x - 基礎訓練](#)

伺服器篇文件

[各版本彙整說明](#)

[RockyLinux 8.x](#)

[CentOS 6.x](#)

[CentOS 5.x](#)

[CentOS 4.x](#)

[RedHat 9](#)

其他連結

[環境工程模式篇](#)

[鳥園討論區](#)

[鳥哥舊站](#)

今日 04807
昨日 06572
本月 050555
上月 165847

[RedHat 6.1](#)
[其他舊文件](#)



<https://linux.vbird.org> is designed by VBird during 2001~2022.