使用者與群組



在開始檔案相關資訊的介紹前，先就簡單的(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！

1. 檔案擁有者:『使用者與群組』的功能是安全防護，由於Linux是多人多工的系統，常會有多人同時用這部主機，考慮每人的隱私權及喜好的工作環境，因此『檔案擁有者』的角色就相當的重要了！
2. 群組概念: 群組最有用的功能之一，就是當團隊開發資源的時候。
3. 其他人。  
   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。

Linux 檔案權限概念



當螢幕面出現了『Permission deny』的時候，『肯定是權限設定錯誤』。

Linux檔案屬性



『ls』這一個察看檔案的指令！在你以root的身份登入Linux之後，下達『ls -al』看看，會看到底下的幾個咚咚：

|  |
| --- |
| [root@www ~]# ls –al  total 156  drwxr-x--- 4 root root 4096 Sep 8 14:06 .  drwxr-xr-x 23 root root 4096 Sep 8 14:21 ..  drwx------ 3 root root 4096 Sep 5 10:37 .gconf <=範例說明處  -rw-r--r-- 1 root root 42304 Sep 4 18:26 install.log <=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 ] |

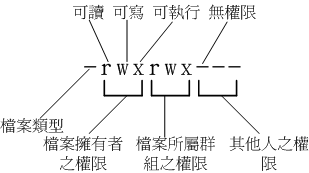
ls是『list』的意思，重點在顯示檔案的檔名與相關屬性。而選項『-al』則表示列出所有的檔案詳細的權限與屬性 (包含隱藏檔，就是檔名第一個字元為『.』的檔案)。解釋一下上面七個欄位個別的意思：

圖2.1.1、檔案屬性的示意圖



* 第一欄代表檔案的類型與權限(permission)：這一欄共有十個字元：(圖2.1.1及圖2.1.2內的權限並無關係)

圖2.1.2、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』：
  + [ d ]是目錄，例如[上表](http://linux.vbird.org/linux_basic/0210filepermission.php#table2.1.1)檔名為『.gconf』的那一行；
  + [ - ]是檔案，例如[上表](http://linux.vbird.org/linux_basic/0210filepermission.php#table2.1.1)檔名為『install.log』那一行；
  + [ l ]為連結檔(link file)；
  + [ b ]為裝置檔裡面的可供儲存的周邊設備(可隨機存取裝置)；
  + [ c ]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，均為『rwx』的三個參數的組合。其中，[r]表可讀(read)、[w]表可寫(write)、[x]表可執行(execute)。這三個權限的位置不會改變，如果沒有權限，就會出現減號[ - ]。
  + 第一組為『檔案擁有者的權限』，以『install.log』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『同群組的權限』；
  + 第三組為『其他非本群組的權限』。
* 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！這個屬性記錄有多少不同的檔名連結到相同的一個i-node號碼。

* 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；
* 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。如果檔案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@www ~]# ls -l /etc/termcap /root/install.log  -rw-r--r-- 1 root root 807103 Jan 7 2007 /etc/termcap  -rw-r--r-- 1 root root 42304 Sep 4 18:26 /root/install.log  # 如上所示，/etc/termcap為2007年修改過，離現在太遠； install.log是今年建立的，所以就顯示完整的時間了。 |

* 第七欄為這個檔案的檔名

這個欄位就是檔名。如果檔名前多一個『.』，則這個檔案為『隱藏檔』。使用『ls』及『ls -a』來看隱藏檔！

|  |
| --- |
|  |
| 例題：假設test1, test2, test3同屬testgroup群組，如有下面的檔案，其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：   * test.txt的擁有者為root，所屬群組為root。權限只有root可存取此檔案，其他人僅能讀檔案； * ping\_tsai的擁有者為test1，而所屬群組為testgroup。其中：   + test1具有可讀可寫可執行的權力；   + test2, test3與test1同是testgroup的群組帳號，則可讀可執行但不能寫(亦即不能修改)；   + 非testgoup這一個群組的人則僅可以讀，不能寫也不能執行！ | | |

|  |
| --- |
| 例題：如果目錄為底下的樣式，testgroup這個群組的成員與其他人(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； * testgroup群組[r-x]的帳號，如test2, test3可以進入本目錄工作，但不能在目錄下進行寫入動作； * 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！ |

passwd



使用useradd建立了帳號之後，在預設的情況下該帳號無法登入，使用passwd設定密碼！

|  |
| --- |
| [root@www ~]# passwd [--stdin] <==所有人均可使用來改自己的密碼  [root@www ~]# passwd [-l] [-u] [--stdin] [-S] \  > [-n 日數] [-x 日數] [-w 日數] [-i 日期] 帳號 <==root 功能  選項與參數：  --stdin ：可以透過來自前一個管線的資料，作為密碼輸入，對 shell script 有幫助！  -l ：是 Lock 的意思，會將 /etc/shadow 第二欄最前面加上 ! 使密碼失效；  -u ：與 -l 相對，是 Unlock 的意思！  -S ：列出密碼相關參數，亦即 shadow 檔案內的大部分資訊。  -n ：後面接天數，shadow 的第 4 欄位，多久不可修改密碼天數  -x ：後面接天數，shadow 的第 5 欄位，多久內必須要更動密碼  -w ：後面接天數，shadow 的第 6 欄位，密碼過期前的警告天數  -i ：後面接『日期』，shadow 的第 7 欄位，密碼失效日期  範例一：請 root 給予 vbird2 密碼  [root@www ~]# passwd vbird2  Changing password for user vbird2.  New UNIX password: <==這裡直接輸入新的密碼，螢幕不會有任何反應  BAD PASSWORD: it is WAY too short <==密碼太簡單或過短的錯誤！  Retype new UNIX password: <==再輸入一次同樣的密碼  passwd: all authentication tokens updated successfully. <==竟然還是成功修改了！ |

當要給予使用者密碼時，透過root來設定。root 可設定各式各樣的密碼，系統幾乎一定會接受！這個是root幫忙設定的結果，那如果是使用者自己要改密碼呢？包括 root 也是這樣修改！

|  |
| --- |
| 範例二：用 vbird2 登入後，修改 vbird2 自己的密碼  [vbird2@www ~]$ passwd <==後面沒有加帳號，就是改自己的密碼！  Changing password for user vbird2.  Changing password for vbird2  (current) UNIX password: <==這裡輸入『原有的舊密碼』  New UNIX password: <==這裡輸入新密碼  BAD PASSWORD: it is based on a dictionary word <==密碼檢驗不通過，請再想個新密碼  New UNIX password: <==這裡再想個來輸入吧  Retype new UNIX password: <==通過密碼驗證！所以重複這個密碼的輸入  passwd: all authentication tokens updated successfully. <==有無成功看關鍵字 |

passwd的使用要很注意，尤其是root！一般帳號建立密碼使用『passwd 帳號』的格式，使用『passwd』表示修改自己的密碼！與root不同，一般帳號在更改密碼時要先輸入自己的舊密碼(current 那一行)，再輸入新密碼 (New那一行)。

如何改變檔案屬性與權限



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：

* chown ：改變檔案擁有者
* chmod ：改變檔案的權限, SUID, SGID, SBIT等等的特性
* 改變檔案擁有者, chown

chown可改變一個檔案的擁有者，使用者須是已存在系統中的帳號，也就是在/etc/passwd中紀錄的使用者名稱才能改變。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，加上-R即可！來看看語法與範例：

|  |
| --- |
| [root@www ~]# chown [-R] 帳號名稱 檔案或目錄  [root@www ~]# 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install.log的擁有者改為bin這個帳號：  [root@www ~]# chown bin install.log  [root@www ~]# ls -l  -rw-r--r-- 1 bin users 68495 Jun 25 08:53 install.log  範例：將install.log的擁有者與群組改回為root：  [root@www ~]# chown root:root install.log  [root@www ~]# ls -l  -rw-r--r-- 1 root root 68495 Jun 25 08:53 install.log |

什麼時候要使用chown或chgrp呢？最常見的例子就是在複製檔案給你之外的其他人時：

|  |
| --- |
| [root@www ~]# cp 來源檔案 目的檔案 |

假設要將.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@www ~]# cp .bashrc .bashrc\_test  [root@www ~]# ls -al .bashrc\*  -rw-r--r-- 1 root root 395 Jul 4 11:45 .bashrc  -rw-r--r-- 1 root root 395 Jul 13 11:31 .bashrc\_test <==新檔案的屬性沒變 |

複製行為(cp)會複製執行者的屬性與權限，所以.bashrc\_test還是root所擁有，即使將檔案拿給bin使用者，仍無法修改屬性/權限，所以必須要將這個檔案的擁有者與群組修改一下！

* 改變權限, chmod

檔案權限的改變使用的是chmod指令，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。

* 數字類型改變檔案權限  
  Linux檔案的基本權限有九個，分別是owner/group/others三種身份各有自己的read/write/execute權限，檔案的權限字元為：『-rwxrwxrwx』，這九個權限是三個三個一組的！可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。每種身份各自的三個權限分數是累加的，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。所以設定權限的變更時，該檔案的權限數字就是770！變更權限的指令chmod是這樣的：

|  |
| --- |
| [root@www ~]# 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@www ~]# ls -al .bashrc  -rw-r--r-- 1 root root 395 Jul 4 11:45 .bashrc  [root@www ~]# chmod 777 .bashrc  [root@www ~]# ls -al .bashrc  -rwxrwxrwx 1 root root 395 Jul 4 11:45 .bashrc |

將權限變成『 -rwxr-xr-- 』，權限的分數就為[4+2+1][4+0+1][4+0+0]=754！所以要下達『chmod 754 filename』。如果有些檔案不希望被其他人看到，那檔案的權限應為如：『-rwxr-----』！

* 符號類型改變檔案權限  
  權限分別是(1)user (2)group (3)others三種身份！可藉由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！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 |

* 假如要『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@www ~]# 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@www ~]# ls -al .bashrc  -rwxr-xr-x 1 root root 395 Jul 4 11:45 .bashrc |

假如是『-rwxr-xr--』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@www ~]# ls -al .bashrc  -rwxr-xr-x 1 root root 395 Jul 4 11:45 .bashrc  [root@www ~]# chmod a+w .bashrc  [root@www ~]# ls -al .bashrc  -rwxrwxrwx 1 root root 395 Jul 4 11:45 .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@www ~]# chmod a-x .bashrc  [root@www ~]# ls -al .bashrc  -rw-rw-rw- 1 root root 395 Jul 4 11:45 .bashrc |

+ 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』。

目錄與檔案之權限意義：



檔案權限對於一般檔案與目錄檔案有大大的不同啊！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)；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。
* x (eXecute)：該檔案具有被系統執行的權限。Windows下檔案是否具有執行能力是藉由『副檔名』來判斷，例如：.exe, .bat, .com 等，但Linux下檔案是否能被執行，則由是否具有『x』權限決定！跟檔名沒有絕對的關係。

對檔案的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！因為檔案記錄的是實際的資料嘛！

* 權限對目錄的重要性

目錄主要的內容在記錄檔名清單！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)：具有讀取目錄結構清單的權限，表示可以查詢該目錄下的檔名資料。可利用 ls將該目錄的內容列表顯示出來！
* w (modify contents of directory)：表示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)：目錄只是記錄檔名，不能執行！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！工作目錄(work directory)就是目前所在的目錄！而變換目錄的指令是『cd』(change directory)！

|  |
| --- |
| 例題：有個目錄的權限如下所示：  drwxr--r-- 3 root root 4096 Jun 25 08:35 .ssh  系統有個帳號為vbird，該帳號並沒有支援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對此目錄僅有r權限，因此vbird可以查詢目錄下的檔名列表。因為vbird不具有x的權限， 因此不能切換到此目錄內！如果在某目錄下不具有x的權限，那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，即使具有該目錄的r的權限。 |

在架設網站時會卡在權限的設定上，只開放r的權限的目錄資料給網路的人來瀏覽，結果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)，用戶無法正確的查閱到檔案的內容(顯示權限不足)。要注意：要開放目錄給人瀏覽時，應該至少也要給予r及x的權限，但w權限不可隨便給！

* 先用root的身份建立所需要的檔案與目錄環境

用root的身份在所有人都可工作的/tmp目錄中建立一testing的目錄，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，建立空檔案可用[touch](http://linux.vbird.org/linux_basic/0220filemanager.php#touch)來處理。過程為：

|  |
| --- |
| [root@www ~]# cd /tmp <==切換工作目錄到/tmp  [root@www tmp]# mkdir testing <==建立新目錄  [root@www tmp]# chmod 744 testing <==變更權限  [root@www tmp]# touch testing/testing <==建立空的檔案  [root@www tmp]# chmod 600 testing/testing <==變更權限  [root@www tmp]# ls -ald testing testing/testing  drwxr--r-- 2 root root 4096 Sep 19 16:01 testing  -rw------- 1 root root 0 Sep 19 16:01 testing/testing  # 目錄的權限是744，所屬群組與使用者均是root ！一般使用者對這個目錄/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？假設鳥哥的系統中含有一個帳號名為vbird的，透過『su - vbird』指令來變換身份喔！看看底下的操作先！

|  |
| --- |
| [root@www tmp]# su - vbird <==切換身份成為 vbird 囉！  [vbird@www ~]$ cd /tmp <==看一下，身份變了喔！提示字元也變成 $ 了！  [vbird@www tmp]$ ls -l testing/  ?--------- ? ? ? ? ? testing  # 因為具有 r 的權限可以查詢檔名。不過權限不足(沒有x)，所以會有一堆問號。  [vbird@www tmp]$ cd testing/  -bash: cd: testing/: Permission denied  # 因為不具有x，所以沒有進入的權限！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？

上面的練習知道只有r可讓使用者讀取目錄的檔名列表，詳細資訊卻讀不到，同時也不能將該目錄變成工作目錄(用cd進入該目錄之意)。x在目錄當中是與『能否進入該目錄』有關， w則可讓使用者刪除、更新、新建檔案或目錄！

Linux檔案種類與副檔名



任何裝置在Linux下都是檔案，連資料溝通的介面也有專屬的檔案在負責～故Linux的檔案種類很多～有哪些種類的檔案呢？

* 檔案種類：

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。有哪些種類的檔案類型呢？

* 正規檔案(regular file )：一般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中最多的一種檔案類型，稱為純文字檔，因為內容為人類可以直接讀到的資料，例如數字、字母等等。幾乎只要可以用來做為設定的檔案都屬於這一種檔案類型。
  + 二進位檔(binary)：系統僅認識且可以執行二進位檔案(binary file)？Linux當中的可執行檔(scripts, 文字型批次檔不算)就是這種格式～ 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中會讀取某些特定格式的檔案，被稱為資料檔(data file)。舉例來說，Linux在使用者登入時，都會將登錄的資料記錄在/var/log/wtmp內，該檔案是一個data file，能透過last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。
* 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！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，如硬碟與軟碟等！可隨機的在硬碟的不同區塊讀寫，這種裝置就是區塊裝置！查一下/dev/sda會發現第一個屬性為[ b ]！
  + 字元(character)設備檔：即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，滑鼠『跳到』不能而是『滑動』到另一個地方！第一個屬性為[ c ]。
* 資料接口檔(sockets)：稱為資料接口檔，這類型的檔案常被用在網路上的資料承接。可啟動一個程式來監聽用戶端的要求，而用戶端可透過這socket進行資料溝通。第一個屬性為[s]，最常在/var/run目錄中看到這種檔案類型。
* 資料輸送檔(FIFO, pipe)：FIFO(first-in-first-out)是特殊的檔案類型，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。
* Linux檔案副檔名：

Linux沒所謂的『副檔名』，檔案能否被執行，與第一欄的十個屬性有關，與檔名無關。Windows能被執行的檔案副檔名通常是.com .exe .bat等，Linux只要權限中有x，即表這檔案可被執行！不過可被執行跟可執行成功是不一樣的～舉例來說，純文字檔修改權限為-rwxrwxrwx後，當然不能執行成功～因為內容根本沒可執行的資料。所以x表檔案有可執行的能力，但能不能執行成功，得要看該檔案的內容～不過仍希望可藉由副檔名來瞭解該檔案，通常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)，批次檔為使用shell寫成的，所以副檔名就編成 .sh；
* \*Z, \*.tar, \*.tar.gz, \*.zip, \*.tgz：經過打包的壓縮檔。壓縮軟體分別為gunzip, tar等，不同的壓縮軟體，而取其相關的副檔名！
* \*.html, \*.php：網頁相關檔案，分別代表HTML語法與PHP語法的網頁檔案！.html的檔案可使用網頁瀏覽器來直接開啟，至於.php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！

Linux系統上的檔名只是瞭解該檔案可能的用途，真正的執行與否需要權限的規範！例如雖然有一個檔案為可執行檔，如常見的/bin/ls顯示檔案屬性的指令，如果這個檔案的權限被修改成無法執行時，ls就變成不能執行囉！

上述的問題最常發生在檔案傳送的過程中。例如在網路上下載一個可執行檔，但是在Linux系統中無法執行！可能是檔案的屬性被改變了！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！

* Linux檔案名稱的限制：

Linux在文字介面下指令操作關係，一般設定Linux檔案名稱時，最好避免特殊字元！如：\* ? > < ; & ! [ ] | \ ' " ` ( ){ }。

因為這些符號在文字介面下，是有特殊意義的！檔案名稱的開頭為小數點『.』表『隱藏檔』！同時由於指令下達中，常會用到 -option之類的選項，所以最好也避免將檔案檔名的開頭以-或+來命名！

Linux目錄配置的依據--FHS



Linux開發產品或distributions太多，如果每個人都各自配置檔案放置的目錄，可能造成管理上的困擾。所以就有Filesystem Hierarchy Standard (FHS)標準！

FHS的主要目的是希望讓使用者瞭解到已安裝軟體通常放置於那個目錄下，所以希望獨立的軟體開發商、作業系統製作者及想要維護系統的使用者，能遵循FHS標準。FHS重點在規範每個特定目錄下該要放置什麼樣的資料。FHS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), /opt (第三方協力軟體) | /etc (設定檔), /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/lock (程序相關) |

* 可分享的：可分享給其他系統掛載使用的目錄，包括執行檔與使用者的郵件等資料，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。

* 根目錄 (/) 的意義與內容：

所有的目錄都是由根目錄衍生出來的，根目錄同時也與開機/還原/系統修復有關。由於系統開機需要特定的開機軟體、核心檔案、開機所需程式、函式庫等檔案資料，若系統出現錯誤時，根目錄也必須要包含能夠修復檔案系統的程式。因為根目錄很重要，所以FHS希望根目錄不要放在大的分割槽內，因為越大的分割槽會放入越多的資料，如此根目錄所在分割槽就可能會有較多發生錯誤的機會。因此FHS建議根目錄(/)所在分割槽應越小越好，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。如此效能較佳，根目錄所在的檔案系統也較不易發生問題。FHS定義根目錄下的次目錄：

|  |  |
| --- | --- |
| 目錄 | 應放置檔案內容 |
| /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 | 主要放置開機會用到的檔案，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grub開機管理程式，則還會存在/boot/grub/目錄。 |
| /dev | Linux任何裝置與周邊設備都以檔案的型態存在於這目錄中。透過存取這個目錄下的某個檔案，就等於存取某個裝置～較重要的檔案有/dev/null, /dev/zero, /dev/tty, /dev/lp\*, /dev/hd\*, /dev/sd\*等。 |
| /etc | 系統主要的設定檔幾乎都在這個目錄內，例如人員的帳號密碼檔、 各種服務的啟始檔等。一般來說，該目錄下的各檔案屬性是可讓一般使用者查閱，但只有root有權力修改。FHS建議不要放置可執行檔(binary)在該目錄中。較重要的檔案有：/etc/inittab, /etc/init.d/, /etc/modprobe.conf, /etc/X11/, /etc/fstab, /etc/sysconfig/等等。另外，其下重要的目錄有：   * /etc/init.d/：所有服務的預設啟動script都是放在這裡的，例如要啟動或者關閉iptables的話：『/etc/init.d/iptables start』、『/etc/init.d/iptables stop』。 * /etc/xinetd.d/：這就是所謂的super daemon管理的各項服務的設定檔目錄。 * /etc/X11/：與X Window有關的各設定檔都在這，尤其是xorg.conf這個X Server的設定檔。 |
| /home | 系統預設的使用者家目錄(home directory)。在新增一般使用者帳號時，預設的使用者家目錄都在此目錄。~：表目前這個使用者的家目錄，而~dmtsai：則代表 dmtsai 的家目錄！ |
| /lib | 放置開機時會用到的函式庫，及在/bin或/sbin下指令會呼叫的函式庫。函式庫呢可想成是『外掛』，某些指令要有『外掛』才能執行程式。尤其是/lib/modules/目錄，會放置核心相關的模組(驅動程式)！ |
| /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！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 | 如果要暫時掛載某些額外的裝置，一般可放置到這個目錄中。 |
| /opt | 第三方協力軟體放置的目錄。什麼是第三方協力軟體？如KDE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。另外如果想要自行安裝額外的軟體(非原本distribution提供)，也能將軟體安裝到這來。不過以前的Linux還是習慣放在/usr/local目錄下！ |
| /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，所以會希望root的家目錄與根目錄放置在同一個分割槽中。 |
| /sbin | Linux有非常多指令是用來設定系統環境的，只有root才能『設定』系統，其他使用者最多只能『查詢』。放在/sbin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。某些伺服器軟體程式，一般則放置到/usr/sbin/當中。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, init, mkfs等等。 |
| /srv | srv為『service』的縮寫，是一些網路服務啟動後，服務所需取用的資料目錄。常見的服務例如WWW, FTP等。舉例來說，WWW伺服器需要的網頁資料就可放置在/srv/www/裡面。 |
| /tmp | 一般使用者或正在執行的程序暫時放置檔案的地方。這個目錄任何人都能夠存取，所以要定期的清理。重要資料不可放置在此目錄！因為FHS甚至建議在開機時，應該要將/tmp下的資料都刪除！ |

FHS所定義根目錄的標準就僅有上面，不過Linux還有許多目錄需要瞭解一下。底下是Linux中的幾個重要目錄：

|  |  |
| --- | --- |
| 目錄 | 應放置檔案內容 |
| /lost+found | 標準的ext2/ext3檔案系統才會產生的目錄，目的在當檔案系統發生錯誤時，將一些遺失的片段放到這目錄下。這目錄常在分割槽的最頂層，如加裝一顆硬碟於/disk，系統會自動產生『/disk/lost+found』。 |
| /proc | 是一個『虛擬檔案系統(virtual filesystem)』！放置的資料都是在記憶體當中，如系統核心、行程資訊(process)、周邊裝置的狀態及網路狀態等。這目錄下的資料都在記憶體中，本身不佔任何硬碟空間。 |
| /sys | 跟/proc非常類似，也是一個虛擬的檔案系統，主要也是記錄與核心相關的資訊。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！ |

因為根目錄與開機有關，開機過程中僅有根目錄會被掛載，其他分割槽是在開機完後才會進行掛載。因此根目錄下與開機過程有關的目錄，不能與根目錄放到不同的分割槽！那哪些目錄不可與根目錄分開呢？有底下這些：

* /etc：設定檔
* /bin：重要執行檔
* /dev：所需要的裝置檔案
* /lib：執行檔所需的函式庫與核心所需的模組
* /sbin：重要的系統執行檔

這五個目錄千萬不可與根目錄分開在不同的分割槽！請背下來啊！

* /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)，如果知道如何透過網路進行分割槽的掛載，那麼/usr可以分享給區域網路內的其他主機來使用！

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。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。因為所有系統預設的軟體(distribution發佈者提供的軟體)都會放置到/usr下，因此這目錄類似Windows系統的『C:\Windows\(當中的一部份)+C:\Program files\』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 |
| /usr/bin/ | 大部分的使用者可使用指令都放在這裡！請注意他與/bin的不同之處。(是否與開機過程有關) |
| /usr/include/ | c/c++等程式語言的檔頭(header)與包含檔(include)放置處，當以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/lib/ | 包含各應用軟體的函式庫、目標檔案(object file)，以及不被一般使用者慣用的執行檔或腳本(script)。 某些軟體會提供一些特殊的指令來進行伺服器的設定，這些指令也不會經常被系統管理員操作，就會放到這個目錄下。如果使用的是X86\_64的Linux系統，可能會有/usr/lib64/目錄產生！ |
| /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)。舉例來說，distribution提供的軟體較舊，想安裝較新的軟體但又不想移除舊版，此時可以將新版軟體安裝於/usr/local/目錄下，與原先的舊版軟體分別！/usr/local下也有bin, etc, include, lib...的次目錄！ |
| /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！ |
| /usr/share/ | 放置共享文件的地方，這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！在此目錄下常見的還有這些次目錄：   * /usr/share/man：線上說明文件 * /usr/share/doc：軟體雜項的文件說明   /usr/share/zoneinfo：與時區有關的時區檔案 |
| /usr/src/ | 一般原始碼建議放置到這裡，src有source的意思。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 |
| /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，會不會有兩個人同時在使用一個燒錄機燒片？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。 |
| /var/log/ | 登錄檔放置的目錄！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/ | 個人電子郵件信箱的目錄，這個目錄也被放置到/var/spool/mail/目錄中！通常兩個目錄是互為連結檔！ |
| /var/run/ | 某些程式或者是服務啟動後，會將他們的PID放置在這目錄下。 |
| /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！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！ |

* 針對FHS，各家distributions的異同

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！不過大致的FHS標準，差異性有限！

目錄樹(directory tree)



Linux所有的檔案與目錄都是由根目錄開始的！是所有目錄與檔案的源頭～然後再分支下來，因此稱這種目錄配置方式為：『目錄樹(directory tree)』主要的特性有：

* 目錄樹的啟始點為根目錄 (/, root)；
* 每個目錄能用本地端的partition或網路上的filesystem。如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。

絕對路徑與相對路徑



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。如./home/dmtsai。開頭不是/就屬於相對路徑的寫法

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，想要進入/var/log 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。

|  |
| --- |
| 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。 |

|  |
| --- |
| 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！ |

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：

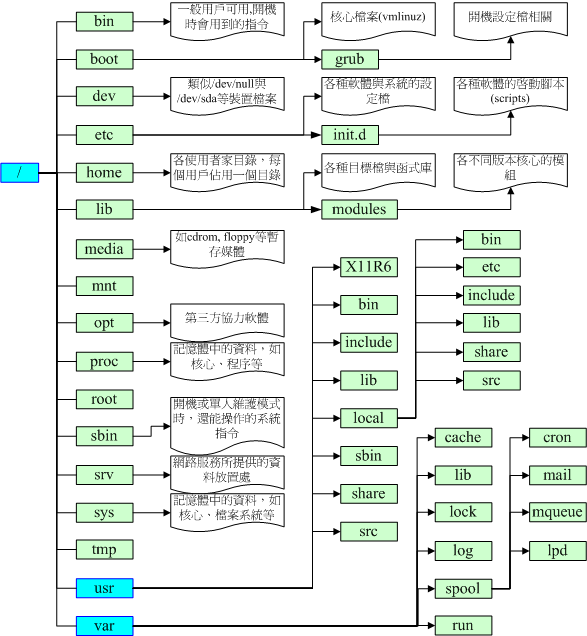


圖3.2.1、目錄樹架構示意圖