

Permisson&directory

# 本章介紹

- **Linux**中的檔案都有各自的權限,如果沒有權限的觀念,是無法管理系統的,另外**Linux**目錄樹也與權限息息相關,因此本章為管理系統的重要基礎!

# 前情提要

- 身分切換成root:
  - su -
  - sudo -i
- 
- 兩種方法皆可

- 上一章介紹了基本的操作，現在來看一個重要的觀念：權限
- 不過在開始看權限之前，首先得先了解檔案與使用者之間的關係

# 檔案的 擁有者/群組/其他人

- 使用者與檔案的關係,可分為:
  - 擁有者(owner):檔案擁有者
  - 群組(group):檔案所屬的群組,群組由多個使用者組成
  - 其他人(others):不屬於上述兩者

## 問題：使用者分群組，有什麼用？

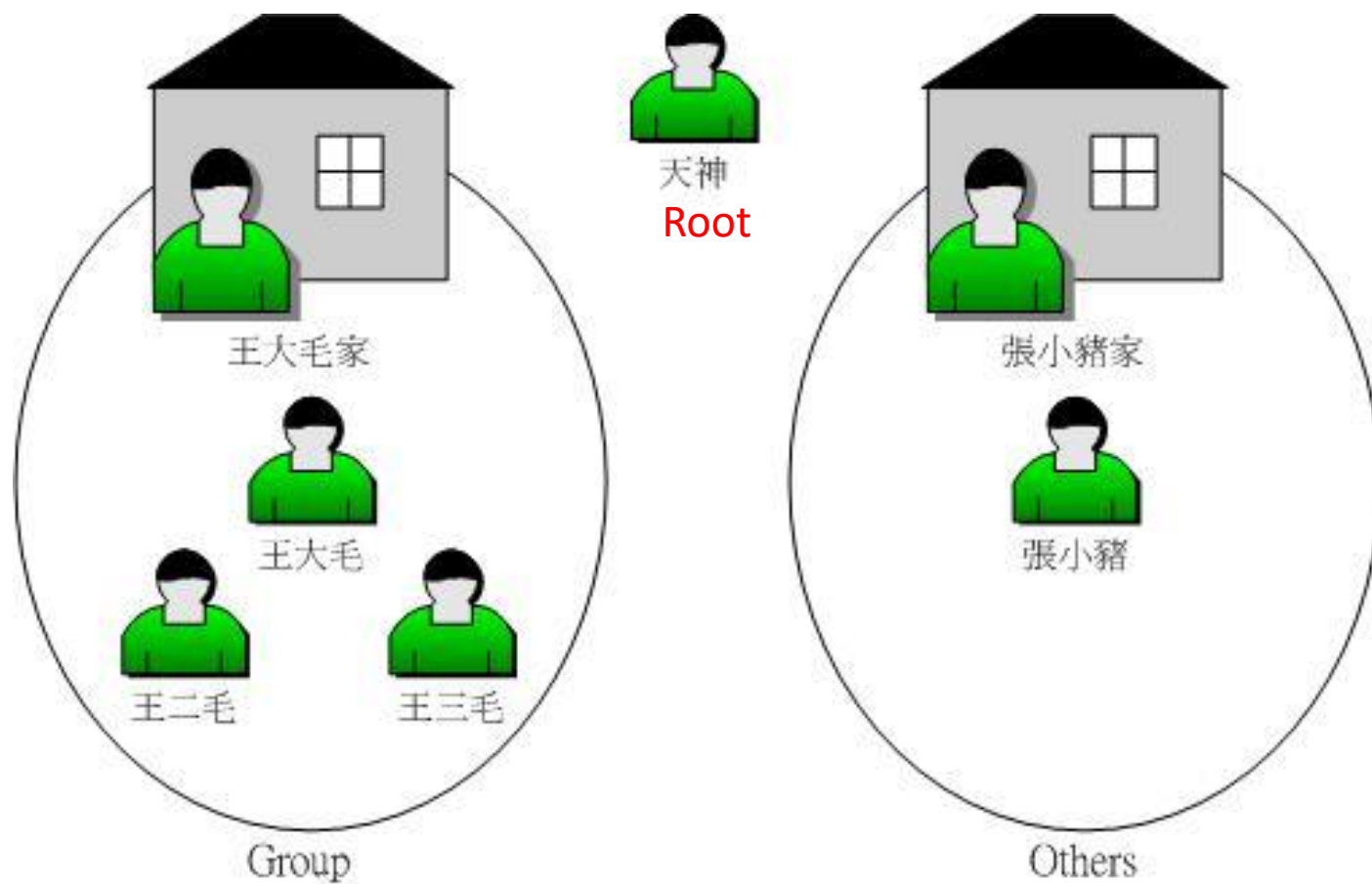
- Linux是個多人多工的系統,必須保護每個使用者的隱私
- 群組可想像為一家人,家庭就是群組,而家庭成員即是使用者,但家庭成員應互相尊重隱私,也會彼此分享資源

# 群組的概念

- 多個使用者雖然在同一群組內，但可以設定『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』，而設定群組共享，則可讓大家共同分享

# 其他人的概念

(非擁有者,也不是群組成員,例如下圖中張小豬一家對王大毛一家的成員來說就是其他人)





- 任何一個檔案都具有『User, Group及Others』三種身份的個別權限

也就是說，你跟這個檔案的關係(你對這個檔案是擁有者/群組/其他人)決定了你對這檔案擁有的權限

# Linux 使用者身份與群組記錄的檔案

- 一般身份使用者和root(管理員)的相關資訊，記錄在/etc/passwd這個檔案內
- 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
- 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！  
不要隨便刪除這三個檔案

觀察檔案權限

先來看看 `ls -l` 的結果 (此時身為root)

```
[root@localhost ~]# ls -l
```

```
total 8
```

```
-rw-----. 1 root root 1727 Jul  7  2021 anaconda-ks.cfg
```

```
-rw-r--r--. 1 root root   0 Jul  7  2021 a.out
```

```
-rw-r--r--. 1 root root 1755 Jul  7  2021 initial-setup-ks.cfg
```

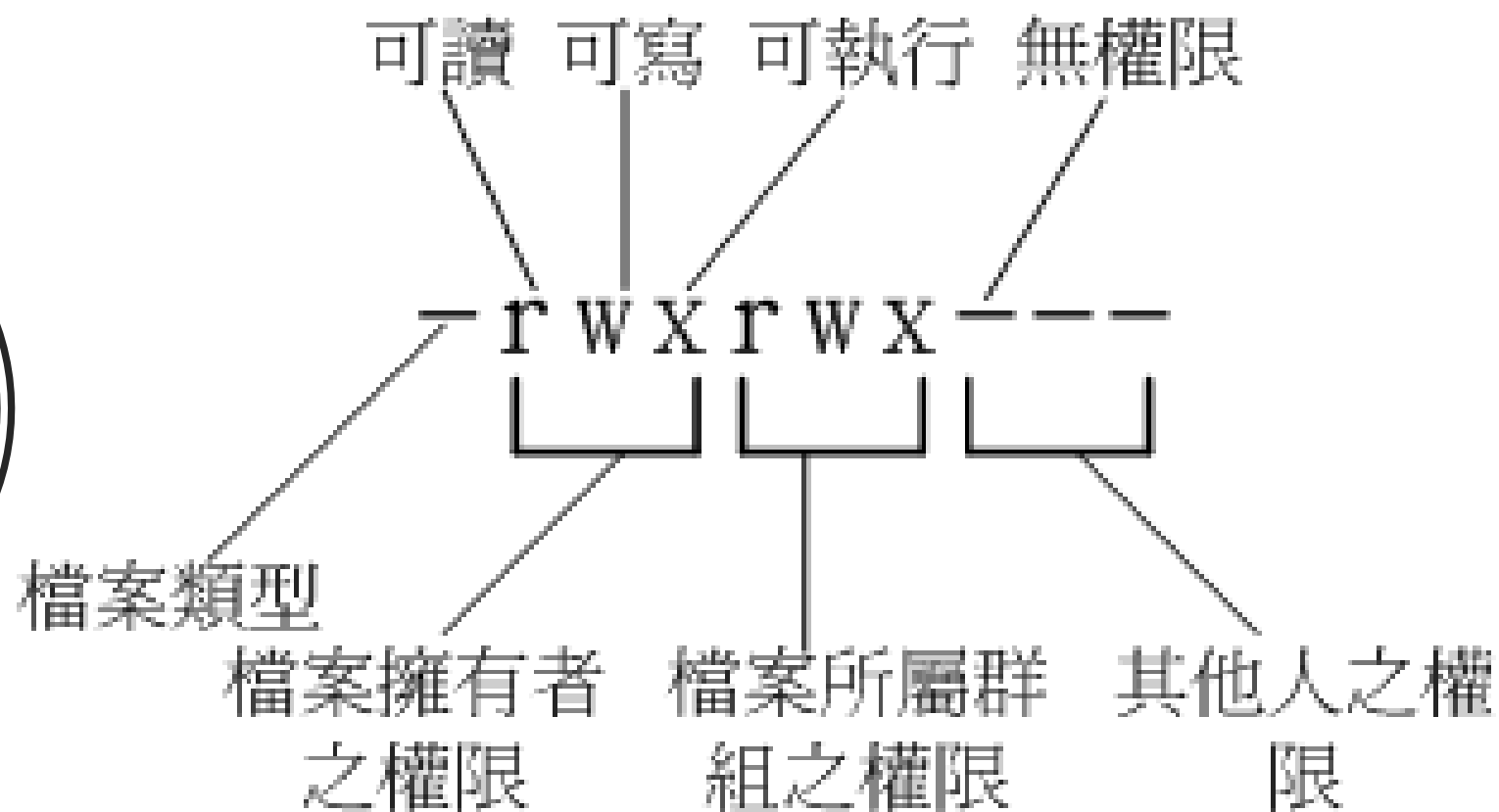
-rw-r--r--. 1 root root 1864 May 4 18:01 initial-setup-ks.cfg

[ 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 ]

- 註：上圖中有的欄位[1] [2] ... 就是下面講解的順序！

## [1]權限的意義



# [1]第一欄之一：常見的檔案類型

- [ **d** ]:目錄
- [ **-** ]:檔案，例如[上表](#)檔名為『initial-setup-ks.cfg』那一行；
- [ **l** ]:連結檔(link file)；
- [ **b** ]:裝置檔裡面的可供儲存的周邊設備(可隨機存取裝置)；
- [ **c** ]:裝置檔裡面的序列埠設備，例如鍵盤、滑鼠(一次性讀取裝置)。

# 第一欄之二：rwx參數意義

- [ r ]代表可讀(read)
  - [ w ]代表可寫(write)
  - [ x ]代表可執行(execute)
- 
- 注意:順序一定是rwx,不可能是wrx , rxw ,rwx ....



- 綜合以上，現在來思考以下權限代表甚麼意義？

**d r w x r - - - -**

答：代表這是一個目錄(**d**)，該目錄的使用者權限為**rw****x**，群組權限只有**r**，而其他人沒有任何權限

- 問題：

- 今天有一個使用者叫做小明，小明是群組A的成員；現在有一個檔案的擁有者為小明，檔案的所屬群組為A，且該檔案的權限如下：`rw xr - - r - -`，那小明對該檔案是屬於 `rw x` 權限還是 `r - -` 權限呢？

- 答：

- 權限的身分比對是由左往右，若身分符合就不再繼續比對；因此小明的身分比對會停在使用者權限，不會繼續往右比對，所以小明對該檔案的權限為`rw x`

# 特別注意

- 檔案的權限和目錄權限代表的意義完全不同，之後會有更詳細的說明，這裡先留意即可

## [2]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，因此每個檔名就會連結到一個i-node (04章會再詳細介紹inode)
- 第二欄記錄的，就是有多少不同的檔名連結到相同的一個i-node號碼

- [3]第三欄：表示這個檔案(或目錄)的『擁有者帳號』
- [4]第四欄：表示這個檔案的所屬群組
- [5]第五欄：為這個檔案的容量大小，預設單位為bytes
- [6]第六欄：為這個檔案的建檔日期或者是最近的修改日期  
(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。)
- [7]第七欄：為這個檔案的檔名  
(如果檔名之前多一個『.』，則代表這個檔案為『隱藏檔』)

# 如何改變檔案屬性與權限？

- `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
- 註 ： SUID,SGID,SBIT為特殊權限，第三章會詳談

chgrp

# chgrp

```
[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～



chown

# chown

```
[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
```

# chown

- 事實上，chown也可以使用『chown user.group file』來一次修改user 和 group，(在擁有者與群組間加上小數點『.』)
- 不過建議使用冒號『:』來隔開擁有者與群組 (怕有.開頭的user或group)
- 此外，chown也能單純的修改所屬群組！例如『chown :sshd initial-setup-ks.cfg』就是修改群組

# chown範例

```
[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？

chmod

# 改變權限, chmod

權限的設定方法有兩種:

- 1.數字法(8進位)
- 2.符號法

chmod : 數字法更改權限

# chmod(數字法)

權限對應的分數(以8進位代表):

r :4 (8進位4轉二進位為100)

w :2 (8進位2轉二進位為010)

x :1(8進位1轉二進位為001)

有沒有發現，權限對應到的分數(八進位)轉成2進位後的數字1就是rwx他們各自所在的位置!!

100	010	001
r - -	-w-	- - x

範例

[-rwxrwx---] 分數則是：

owner = rwx = 4+2+1 = 7

group = rwx = 4+2+1 = 7

others= --- = 0+0+0 = 0



# chmod(數字法範例)

```
[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

```
[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
```

chmod: 符號法更改權限

# Chmod(符號類型改變檔案權限)

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

# Chmod(符號類型改變檔案權限)

- 範例1:設定一個檔案的權限成為『-rwxr-xr-x』

```
[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
```

# Chmod(符號類型改變檔案權限)

- 範例2:不知原先的檔案屬性，只想要增加每個人均可寫入的權限

```
[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
```

# Chmod(符號類型改變檔案權限)

- 範例3: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

```
[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！
```

# Chmod(符號類型改變檔案權限)

注意:    +,-沒有指定的權限不變

=沒有指定的權限直接消失

- 可自行試驗看看

# 目錄與檔案之權限意義



# 權限對檔案的重要性

- 首先，何謂檔案？
- 檔案: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

# 權限對檔案的重要性

權限對於檔案來說，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。

# 權限對檔案的重要性

- 對於檔案的rwx來說，主要都是針對『檔案的內容』，與檔案檔名的存在與否沒有關係

# 權限對目錄的重要性

- 何謂目錄？
- 目錄:主要的內容在記錄檔名清單，檔名與目錄強烈的關連

# 權限對目錄的代表意義

- r (read contents in directory) :
  - 表示具有讀取目錄結構清單的權限
  - 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。所以就可以利用 ls 這個指令將該目錄的內容列表顯示出來
  - 注意：具有r權限能用ls看到檔名清單，但不一定能用ls -l 看到檔案權限，能否看到檔按權限是由下面介紹的x權限來決定

- w (modify contents of directory) :

- 這個可寫入的權限對目錄來說，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

- x (access directory) :

- 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)
- 沒有x的話用 `ls -l` 看不到權限(假設只有r的情況下)

# 權限對目錄&檔案的重要性

- 整理:

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)



- 要開放目錄給任何人瀏覽時，應該至少也要給予r(可看到檔名列表,但看不到權限)及x(進入此目錄)的權限，但w權限不可隨便給(這樣別人就可以隨意刪除檔案)！

例題：

假設有個帳號名稱為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！

# 複製檔案該有的最低權限

- 至少對欲複製的檔案有r權限

# Linux檔案種類與副檔名

# 一句名言：

- 在介紹檔案的種類前，先來看Linux裡的一句名言：

Everything is file, file is everything

- 意思就是字面上翻成中文的意思，也就是說，不論是文字檔，目錄，硬碟，滑鼠，鍵盤...對Linux來說都是檔案
- 既然所有東西都是檔案，那所有東西都有各自屬性與權限，也都可以透過檔案系統(04章會介紹)進行管理

# 檔案的種類

# 正規檔案(regular file)：

一般在進行存取的類型的檔案，在 `ls -al` 顯示出來的屬性方面，為 `[-]`，。另外，依照檔案的內容，又大略可以分為：

- 純文字檔(ASCII)：最多的一種檔案類型，內容為人類可直接讀到的資料。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
- 二進位檔(binary)：Linux當中的可執行檔(scripts, 文字型批次檔不算)就是這種，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，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

# 目錄(directory)

- 第一個屬性為 [ d ]
- 可理解為檔名清單

# 連結檔(link)

- 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，(**硬碟與軟碟**)，第一個屬性為[ **b** ]

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！第一個屬性為[ **c** ]。

# 資料接口檔(sockets)

- 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
- 這種類型的檔案通常被用在網路上的資料承接
- 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

# 資料輸送檔(FIFO, pipe)

- FIFO也是一種特殊的檔案類型，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
- FIFO是first-in-first-out的縮寫。第一個屬性為[p]

Linux的檔名

# Linux檔案副檔名

- 基本上，Linux的檔案是沒有所謂的『副檔名』，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)
- 在Linux底下，只要權限當中具有x的話，即代表這個檔案具有可以被執行的能力
- 雖然如此，我們仍然希望可以藉由副檔名來瞭解該檔案是什麼東西

註:x代表這個檔案具有可執行的能力，但是能不能執行成功，就得要看該檔案的內容

# 常見副檔名

\*.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，由於不同的壓縮軟體，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 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

- 基本上，Linux系統上的檔名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
- 上述的這種問題最常發生在檔案傳送的过程中
  - 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，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

- 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



# 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！例如底下這些：

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

- 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『.』時，代表這個檔案為『隱藏檔』
- 由於指令下達當中，常常會使用到 -option 之類的選項，最好也避免將檔案檔名的開頭以 - 或 + 來命名

# Linux目錄配置

註：這裡有點複雜，理解即可，不過目錄樹的底下各目錄代表的意義需稍微記憶一下

# Linux目錄配置的依據--FHS

# Filesystem Hierarchy Standard (FHS)

	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)

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。

- 註：以上理解即可，不須死記

- 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(不是user!!!!)
- /var (variable)：與系統運作過程有關。

# 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
- 所有的目錄都是由根目錄衍生出來的
- 由於系統開機時需要特定檔案資料，若系統出現錯誤時，根目錄也必須要包含有能夠**修復檔案系統的程式**才行
- 因為根目錄的重要，所以在**FHS**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，就可能會有多發生錯誤的機會

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	<p>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：</p> <ul style="list-style-type: none"><li>• /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</li><li>• /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。</li><li>• /etc/sgml/(建議)：與 SGML 格式有關的各項設定檔</li><li>• /etc/xml/(建議)：與 XML 格式有關的各項設定檔</li></ul>
/lib	<p>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：</p> <ul style="list-style-type: none"><li>• /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！</li></ul>

/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	如果你想要暫時掛載某些額外的裝置，一般建議你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，如果你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	<p>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：</p> <ul style="list-style-type: none"><li>• ~：代表目前這個使用者的家目錄</li><li>• ~dmtsai：則代表 dmtsai 的家目錄！</li></ul>
/lib<qual>	<p>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</p>
/root	<p>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。</p>

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： <a href="#">/proc/cpuinfo</a> , <a href="#">/proc/dma</a> , <a href="#">/proc/interrupts</a> , <a href="#">/proc/ioports</a> , <a href="#">/proc/net/*</a> 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：**/etc, /bin, /dev, /lib, /sbin** 這五個重要目錄
- 現在許多的 **Linux distributions** 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**
- 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！

/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`)
  - 若知道如何透過網路進行分割槽的掛載，`/usr`可以分享給區域網路內的其他主機來使用
- 很多人誤會`/usr`為`user`的縮寫，其實`usr`是Unix Software Resource的縮寫，而不是使用者的資料 (這點要注意)
- FHS建議所有軟體開發者，應該資料合理的放置到`/usr`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，`/usr`會佔用最多的硬碟容量
- 一般來說，`/usr`的次目錄建議有下頁這些：

/usr/bin/	所有 <b>一般用戶能夠使用的指令都放在這裡</b> ！目前新的 CentOS 7 已經將全部的使用者指令放置於此，而使用 <b>連結檔的方式將 /bin 連結至此</b> ！也就是說， <b>/usr/bin 與 /bin 是一模一樣了</b> ！另外，FHS 要求在此目錄下不應該有子目錄！
/usr/lib/	基本上，與 /lib 功能相同，所以 <b>/lib 就是連結到此目錄中的</b> ！
/usr/local/	<b>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)</b> ，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， <b>該目錄下也是具有bin, etc, include, lib...的次目錄喔</b> ！
/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 <b>基本功能與 /sbin 也差不多</b> ，因此目前 /sbin 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：

- /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

## FHS建議存在的目錄

/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。
- 常見的次目錄有以下這些：

/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。



<code>/var/mail/</code>	放置個人電子郵件信箱的目錄，不過這個目錄也被放置到 <code>/var/spool/mail/</code> 目錄中！通常這兩個目錄是互為連結檔啦！
<code>/var/run/</code>	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 <code>/run</code> 相同，這個目錄連結到 <code>/run</code> 去了！
<code>/var/spool/</code>	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 <code>/var/spool/mail/</code> 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 <code>/var/spool/mqueue/</code> 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 <code>/var/spool/cron/</code> 目錄中！

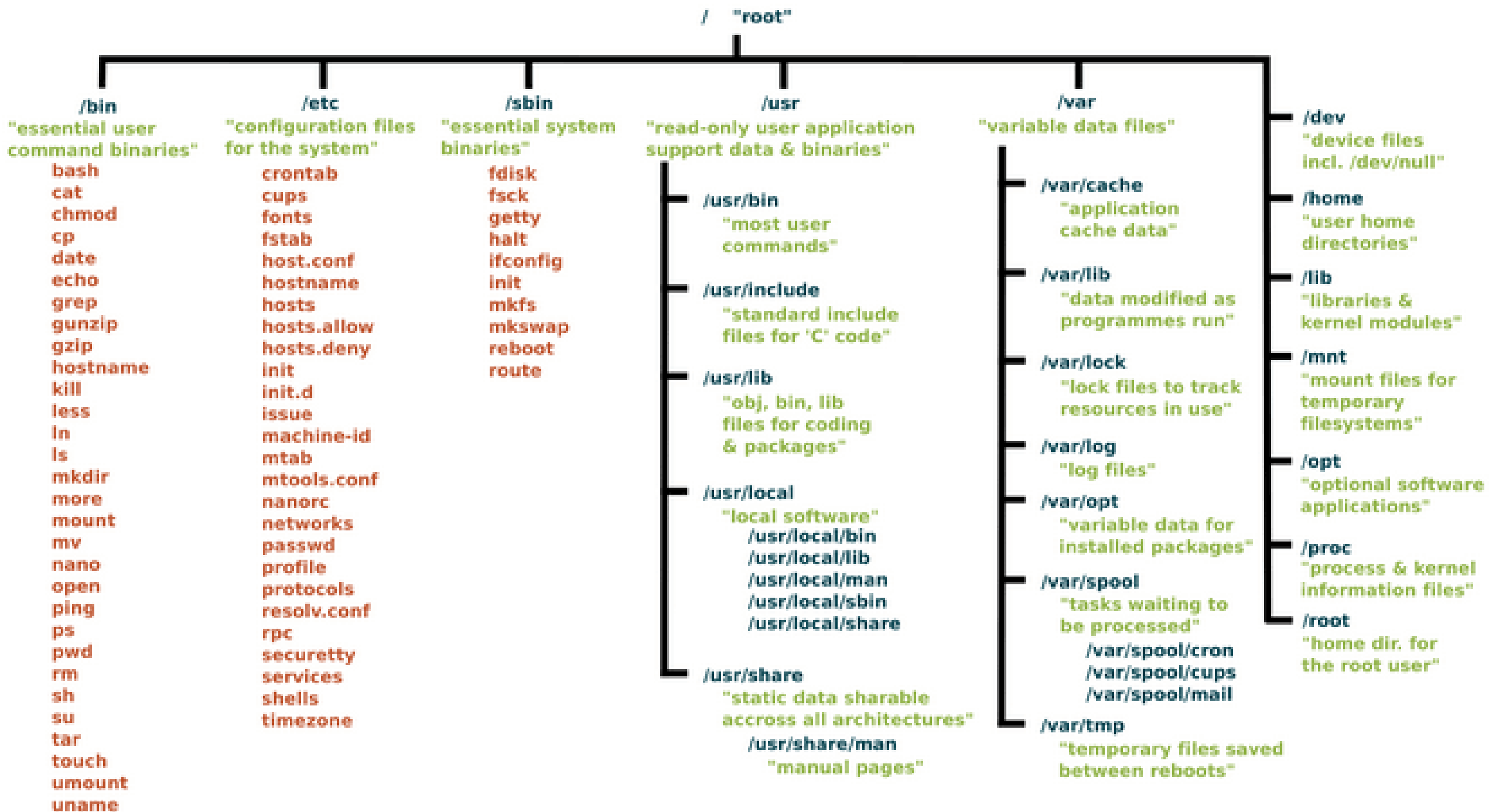


針對FHS，各家distributions的異同，與 CentOS7 的變化

- 由於FHS僅是定義出最上層(/)及次層(/usr, /var)的目錄，因此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，目錄名稱是不同的
  - 不過只要記住大致的FHS標準，差異性其實有限
- 此外，CentOS 7 在目錄的編排上與過去的版本不同。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

目錄樹(directory tree)

- 在Linux底下，**所有的檔案與目錄都是由根目錄開始的！**那『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**
  - 下圖是目錄樹的結構:



- 若忘記目錄結構與其內容，上面的圖非常有用
- 看似很複雜，其實需要記的就只有大分支(上圖深藍色的字)以及它底下可能有甚麼東西(上圖淺綠色的文字說明)
- 例如/**etc**底下放的是設定檔

目錄樹的記憶方式:

- 千萬不要硬背
- 由於目錄名稱都有其意思，因此先理解目錄地下放甚麼即可
- 等到日後熟悉linux後自然目錄樹結構就記起來了

路徑



- 大致了解目錄樹結構後，接下來要來看“路徑”
  - 在看路徑之前，先來認識甚麼是“工作目錄”？
    - 工作目錄就是你“當前”所在的目錄，可以用`pwd`指令查看
    - 你可以用“`cd <目錄名稱>`”切換工作目錄(第三章會詳談`cd`的用法)
- 路徑是系統用來識別檔案位置的方式，配合目錄樹結構的表示方法
- 例如我在`/tmp`目錄下有一個檔案A，而`tmp`隸屬於`/`目錄下，若我的工作目錄為`/home`此我可以這樣表達檔案A的路徑：
  1. `/tmp/A` (絕對路徑表示法，表示根目錄底下的`tmp`底下的A)
  2. `../tmp/A` (相對路徑表示法，表示`/home`的上一層目錄底下的`tmp`底下的A)

# 路徑表示方法

- **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**：相對於當前工作目錄的檔名寫法。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
- `.`：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。

# CentOS 的觀察

- 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
  - 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

### Linux 核心、LSB 的標準查閱:

- 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](https://www.linuxbase.org/lsb-cert/productdir.php?by_lsb)
- 若想要知道確切的核心與 LSB 所需求的幾種重要的標準的話，就要使用諸如 **uname**(之後章節介紹) 與 **lsb\_release** 等指令來查閱
- 不過 **lsb\_release** 指令已經**不是預設安裝**的軟體了，所以得要自己安裝該軟體才才行。(有網路的話可用yum安裝:19章)

常用(需記憶的)目錄彙整

- `/etc/`：幾乎系統的所有設定檔案均在此，尤其 `passwd,shadow`
- `/boot`：開機設定檔，也是預設擺放核心 `vmlinuz` 的地方
- `/usr/bin, /bin`：一般執行檔擺放的地方
- `/usr/sbin, /sbin`：系統管理員常用指令集
- `/dev`：擺放所有系統裝置檔案的目錄
- `/var/log`：擺放系統登錄檔案的地方
- `/run`：CentOS 7 以後才有，將經常變動的項目(每次開機都不同，如程序的PID)移動到記憶體暫存，所以 `/run` 並不佔實際磁碟容量

# 鳥站連結

- [https://linux.vbird.org/linux\\_basic/centos7/0210filepermission.php](https://linux.vbird.org/linux_basic/centos7/0210filepermission.php)

# 相關影片

- <https://www.youtube.com/watch?v=tnD7Ci8zwkc&list=PLmH0oRHkZPSUBmF5iXxtJb04e11MytQnu&index=9>(檔案權限:看到15分25秒即可)