

鸟哥的 Linux 私房菜

为取得较佳浏览结果, 请爱用 [firefox](#) 浏览本网页



第六章、Linux 的档案权限与目录配置

最近更新日期: 2008/10/21

Linux 最优秀的地方之一, 就在于他的多人多任务的环境。而为了让各个使用者具有较保密的档案数据, 因此档案的权限管理就变的很重要了。Linux 一般将档案可存取的身份分为三个类别, 分别是 owner/group/other, 且三种身份各有 read/write/execute 等权限。若管理不当, 你的 Linux 主机将会变的很『不苏服! @_@』。另外, 你如果首次接触 Linux 的话, 那么, 在 Linux 底下这么多的目录/档案, 到底每个目录/档案代表什么意义呢? 底下我们就来一一介绍呢!

1. 使用者与群组
2. Linux 档案权限概念
 - 2.1 Linux 文件属性
 - 2.2 如何改变文件属性与权限: `chgrp`, `chown`, `chmod`
 - 2.3 目录与档案之权限意义
 - 2.4 Linux 档案种类与扩展名
3. Linux 目录配置
 - 3.1 Linux 目录配置的依据—FHS: `/`, `/usr`, `/var`
 - 3.2 目录树(directory tree)
 - 3.3 绝对路径与相对路径
4. 重点回顾
5. 本章练习
6. 参考数据与延伸阅读
7. 针对本文的建议: <http://phorum.vbird.org/viewtopic.php?t=23878>



使用者与群组

经过[第五章](#)的洗礼之后, 你应该可以在 Linux 的指令列模式底下输入指令了吧? 接下来, 当然是要让你好好的浏览一下 Linux 系统里面有哪些重要的档案啰。不过, 每个档案都有相当多的属性与权限, 其中最重要的可能就是档案的拥有者的概念了。所以, 在开始档案相关信息的介绍前, 鸟哥先就简单的(1)使用者及(2)群组与(3)非本群组外的其他人等概念作个说明吧~ 好让你快点进入状况的哩!
^_^
—

1. 档案拥有者

初次接触 Linux 的朋友大概会觉得很怪异, 怎么『Linux 有这么多使用者, 还分什么群组, 有什么用?』。这个『用户与群组』的功能可是相当健全

而好用的一个安全防护呢！怎么说呢？由于 Linux 是个多人多任务的系统,因此可能常常会有多人同时使用这部主机来进行工作的情况发生,为了考虑每个人的隐私权以及每个人的喜好的工作环境,因此,这个『档案拥有者』的角色就显的相当的重要了！

例如当你将你的 e-mail 情书转存成档案之后,放在你自己的家目录,你总不希望被其他人看见自己的情书吧？这个时候,你就把该档案设定成『只有档案拥有者,就是我,才能看与修改这个档案的内容』,那么即使其他人知道你有这个相当『有趣』的档案,不过由于你有设定适当的权限,所以其他人自然也就无法知道该档案的内容啰！

2. 群组概念

那么群组呢？为何要配置文件案还有所属的群组？其实,群组最有用的功能之一,就是当你在团队开发资源的时候啦！举例来说,假设有两组专题生在我的主机里面,第一个专题组别为 projecta,里面的成员有 class1, class2, class3 三个;第二个专题组别为 projectb,里面的成员有 class4, class5, class6。这两个专题之间是有竞争性质的,但却要缴交同一份报告。每组的组员之间必须要能够互相修改对方的数据,但是其他组的组员则不能看到本组自己的档案内容,此时该如何是好？

在 Linux 底下这样的限制是很简单啦！我可以经由简易的档案权限设定,就能限制非自己团队(亦即是群组啰)的其他人不能够阅览内容啰！而且亦可以让自己的团队成员可以修改我所建立的档案！同时,如果我自己还有私人隐密的文件,仍然可以设定成让自己的团队成员也看不到我的档案数据。很方便吧！

另外,如果 teacher 这个账号是 projecta 与 projectb 这两个专题的老师,他想要同时观察两者的进度,因此需要能够进入这两个群组的权限时,你可以设定 teacher 这个账号,『同时支持 projecta 与 projectb 这两个群组!』,也就是说:每个账号都可以有多个群组的支持呢！

这样说或许你还不容易理解这个使用者与群组的关系吧？没关系,我们可以使用目前『家庭』的观念来进行解说喔！假设有一家人,家里只有三兄弟,分别是王大毛、王二毛与王三毛三个人,而这个家庭是登记在王大毛的名下的!所以,『王大毛家有三个人,分别是王大毛、王二毛与王三毛』,而且这三个人都有自己的房间,并且共同拥有一个客厅喔！

- 使用者的意义:由于王家三人各自拥有自己的房间,所以,王二毛虽然可以进入王三毛的房间,但是二毛不能翻三毛的抽屉喔!那样会被三毛 K 的!因为抽屉里面可能有三毛自己私人的东西,例如情书啦,日记啦等等的,这是『私人的空间』,所以当然不能让二毛拿啰!
- 群组的概念:由于共同拥有客厅,所以王家三兄弟可以在客厅打开

电视机啦、翻阅报纸啦、坐在沙发上面发呆啦等等的！反正，只要是在客厅的玩意儿，三兄弟都可以使用喔！因为大家都是一家人嘛！

这样说来应该有点晓得了喔！那个『王大毛家』就是所谓的『群组』啰，至于三兄弟就是分别为三个『使用者』，而这三个使用者是在同一个群组里面的喔！而三个使用者虽然在同一群组内，但是我们可以设定『权限』，好让某些用户个人的信息不被群组的拥有者查询，以保有个人『私人的空间』啦！而设定群组共享，则可让大家共同分享喔！

3. 其他人的概念

好了，那么今天又有个人，叫做张小猪，他是张小猪家的人，与王家没有关系啦！这个时候，除非王家认识张小猪，然后开门让张小猪进来王家，否则张小猪永远没有办法进入王家，更不要说进到王三毛的房间啦！不过，如果张小猪透过关系认识了三毛，并且跟王三毛成为好朋友，那么张小猪就可以透过三毛进入王家啦！呵呵！没错！那个张小猪就是所谓的『其他人，Others』啰！

因此，我们就可以知道啦，在 Linux 里面，任何一个档案都具有『User, Group 及 Others』三种身份的个别权限，我们可以将上面的说明以底下的图示来解释：

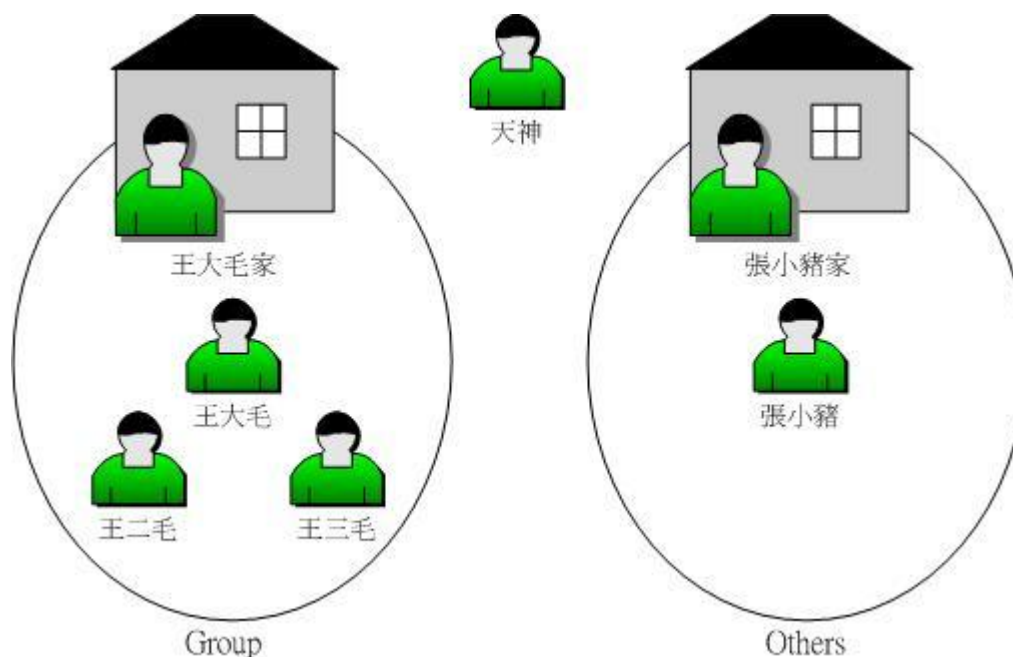


图 1.1、每个档案的拥有者、群组与其他人的示意图

我们以王三毛为例，王三毛这个『档案』的拥有者为王三毛，他属于王大毛这个群组，而张小猪相对于王三毛，则只是一个『其他人(others)』

而已。

不过，这里有个特殊的人物要来介绍的，那就是『万能的天神』！这个天神具有无限的神力，所以他可以到达任何他想要去的地方，呵呵！那个人在 Linux 系统中的身份代号是『root』啦！所以要小心喔！那个 root 可是『万能的天神』喔！

无论如何，『使用者身份』，与该使用者所支持的『群组』概念，在 Linux 的世界里面是相当的重要的，他可以帮助你让你的多任务 Linux 环境变的更容易管理！更详细的『身份与群组』设定，我们将在[账号管理](#)再进行解说。底下我们将针对文件系统与档案权限来进行说明。

- Linux 用户身份与群组记录的档案

在我们 Linux 系统当中，默认的情况下，所有的系统上的账号与一般身份使用者，还有那个 root 的相关信息，都是记录在/etc/passwd 这个档案内的。至于个人的密码则是记录在/etc/shadow 这个档案下。此外，Linux 所有的组名都纪录在/etc/group 内！这三个档案可以说是 Linux 系统里面账号、密码、群组信息的集中地啰！不要随便删除这三个档案啊！ ^_^

至于更多的与账号群组有关的设定，还有这三个档案的格式，不要急，我们在[第四篇讲到账号](#)时，会再跟大家详细的介绍的！这里先有概念即可。



Linux 档案权限概念

大致了解了 Linux 的使用者与群组之后，接着下来，我们要来谈一谈，这个档案的权限要如何针对这些所谓的『使用者』与『群组』来设定该档案的权限呢？这个部分是相当的重要的，尤其对于初学者来说，因为档案的权限与属性是学习 Linux 的一个相当重要的关卡，如果没有这部份的概念，那么你将老是听不懂别人在讲什么呢！尤其是当你在你的屏幕前面出现了『Permission deny』的时候，不要担心，『肯定是权限设定错误』啦！呵呵！好了，闲话不多聊，赶快来瞧一瞧先：



Linux 文件属性

嗯！既然要让你了解 Linux 的文件属性，那么有个重要的也是常用的指令就必须要先跟你说啰！那一个？！就是『ls』这一个察看档案的指令啰！在你以 root 的身份登入 Linux 之后，下达『ls -al』看看，会看到底下的几个咚咚：

```
[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 ..
-rw-----  1  root  root    1474  Sep  4 18:27
anaconda-ks.cfg
-rw-----  1  root  root     199  Sep  8 17:14 .bash_history
-rw-r--r--  1  root  root      24  Jan  6  2007 .bash_logout
-rw-r--r--  1  root  root     191  Jan  6  2007 .bash_profile
-rw-r--r--  1  root  root     176  Jan  6  2007 .bashrc
-rw-r--r--  1  root  root     100  Jan  6  2007 .cshrc
drwx-----  3  root  root    4096  Sep  5 10:37 .gconf          <=
范例说明处
drwx-----  2  root  root    4096  Sep  5 14:09 .gconfd
-rw-r--r--  1  root  root   42304  Sep  4 18:26 install.log <=
范例说明处
-rw-r--r--  1  root  root    5661  Sep  4 18:25
install.log.syslog
[  1  ][  2  ][  3  ][  4  ][  5  ][  6  ][  7
  ]
[  权限  ][连结][拥有者][群组][档案容量][  修改日期  ][  檔
名      ]

```

Tips:

由于本章后续的 `chgrp`, `chown` 等指令可能都需要使用 `root` 的身份才能够处理，所以这里建议您以 `root` 的身份登入 Linux 来学习本章。



`ls` 是『list』的意思，重点在显示档案的文件名与相关属性。而选项『`-al`』则表示列出所有的档案详细的权限与属性（包含隐藏文件，就是文件名第一个字符为『`.`』的档案）。如上所示，在你第一次以 `root` 身份登入 Linux 时，如果你输入上述指令后，应该有上列的几个东西，先解释一下上面七个字段个别的意思：

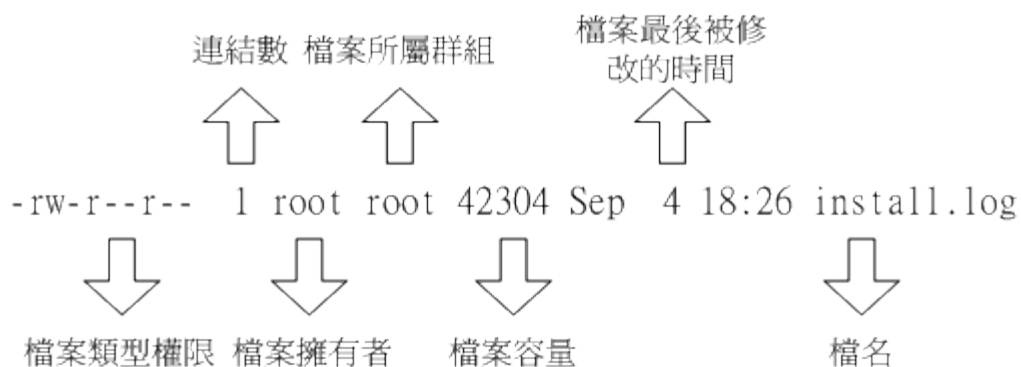


图 2.1.1、文件属性的示意图

- 第一栏代表这个档案的类型与权限(permission):

这个地方最需要注意了!仔细看的话,你应该可以发现这一栏其实共有十个字符:
(图 2.1.1 及图 2.1.2 并无关系)

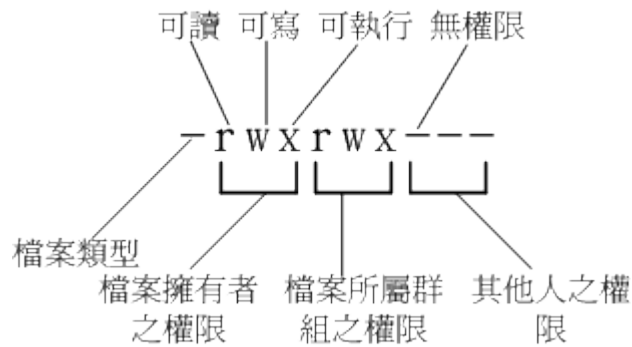


图 2.1.2、档案的类型与权限之内容

- 第一个字符代表这个档案是『目录、档案或链接文件等等』：
 - 当为[d]则是目录,例如上表档名为『.gconf』的那一行;
 - 当为[-]则是档案,例如上表檔名为『install.log』那一行;
 - 若是[l]则表示为连结档(link file);
 - 若是[b]则表示为装置文件里面的可供储存的接口设备(可随机存取装置);
 - 若是[c]则表示为装置文件里面的串行端口设备,例如键盘、鼠标(一次性读取装置)。
- 接下来的字符中,以三个为一组,且均为『rwx』的三个参数的组合。其中,[r]代表可读(read)、[w]代表可写(write)、[x]代表可执行(execute)。要注意的是,这三个权限的位置不会改变,如果没有权限,就会出现减号[-]而已。
 - 第一组为『档案拥有者的权限』,以『install.log』那个档案为例,该档案的拥有者可以读写,但不可执行;
 - 第二组为『同群组的权限』;
 - 第三组为『其他非本群组的权限』。

例题:

若有一个档案的类型与权限数据为『-rwxr-xr--』,请说明其意义为何?

答:

先将整个类型与权限数据分开查阅，并将十个字符整理成为如下所示：

`[-][rwx][r-x][r--]`

1 234 567 890

1 为：代表这个文件名为目录或档案，本例中为档案(-)；

234 为：拥有者的权限，本例中为可读、可写、可执行(rwx)；

567 为：同群组用户权力，本例中为可读可执行(rx)；

890 为：其他用户权力，本例中为可读(r)

同时注意到，rwx 所在的位置是不会改变的，有该权限就会显示字符，没有该权限就变成减号(-)就是了。

另外，目录与档案的权限意义并不相同，这是因为目录与档案所记录的数据内容不相同所致。由于目录与档案的权限意义非常的重要，所以鸟哥将他独立到 [2.3 节目录与档案之权限意义](#) 中再来谈。

- 第二栏表示有多少档名连结到此节点(i-node)：

每个档案都会将他的权限与属性记录到文件系统的 i-node 中，不过，我们使用的目录树却是使用文件名来记录，因此每个档名就会连结到一个 i-node 啰！这个属性记录的，就是有多少不同的档名连结到相同的一个 i-node 号码去就是了。关于 i-node 的相关资料我们会在第八章谈到文件系统时再加强介绍的。

- 第三栏表示这个档案(或目录)的『拥有者账号』
- 第四栏表示这个档案的所属群组

在 Linux 系统下，你的账号会附属于一个或多个的群组中。举刚刚我们提到的例子，class1, class2, class3 均属于 projecta 这个群组，假设某个档案所属的群组为 projecta，且该档案的权限如图 2.1.2 所示(-rwxrwx---)，则 class1, class2, class3 三人对于该档案都具有可读、可写、可执行的权限(看群组权限)。但如果是不属于 projecta 的其他账号，对于此档案就不具有任何权限了。

- 第五栏为这个档案的容量大小，默认单位为 bytes；
- 第六栏为这个档案的建档日期或者是最近的修改日期：

这一栏的内容分别为日期(月/日)及时间。如果这个档案被修改的时间距离现在

太久了，那么时间部分会仅显示年份而已。 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 为 2007 年所修改过的档案，
  离现在太远之故；
# 至于 install.log 是今年（2008）所建立的，所以就显示完整的时间了。
```

如果想要显示完整的时间格式，可以利用 `ls` 的选项，亦即：『`ls -l --full-time`』就能够显示出完整的时间格式了！包括年、月、日、时间喔。另外，如果你当初是以繁体中文安装你的 Linux 系统，那么日期字段将会以中文来显示。可惜的是，中文并没有办法在纯文本的终端机模式中正确的显示，所以此栏会变成乱码。那你就得要使用『`LANG=en_US`』来修改语系喔！

如果想要让系统默认的语系变成英文的话，那么你可以修改系统配置文件『`/etc/sysconfig/i18n`』，利用第五章谈到的 `nano` 来修改该档案的内容，使 `LANG` 这个变量成为上述的内容即可。

- 第七栏为这个档案的档名

这个字段就是档名了。比较特殊的是：如果档名之前多一个『`.`』，则代表这个档案为『隐藏档』，例如上表中的 `.gconf` 那一行，该档案就是隐藏档。你可以使用『`ls`』及『`ls -a`』这两个指令去感受一下什么是隐藏档啰！

Tips:

对于更详细的 `ls` 用法，还记得怎么查询吗？对啦！使用 `man ls` 或 `info ls` 去看看他的基础用法去！自我进修是很重要的，因为『师傅带进门，修行看个人！』，自古只有天才学生，没有天才老师呦！加油吧！^_^



这七个字段的意义是很重要的！务必清楚的知道各个字段代表的意义！尤其是第一个字段的九个权限，那是整个 Linux 档案权限的重点之一。底下我们来做个简单的练习，你就会比较清楚啰！

例题：

假设 `test1`，`test2`，`test3` 同属于 `testgroup` 这个群组，如果有下面的两个档案，请说明两个档案的拥有者与其相关的权限为何？

```
-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：

- 档案 test.txt 的拥有者为 root，所属群组为 root。至于权限方面则只有 root 这个账号可以存取此档案，其他人则仅能读此档案；
- 另一个档案 ping_tsai 的拥有者为 test1，而所属群组为 testgroup。其中：
 - test1 可以针对此档案具有可读可写可执行的权力；
 - 而同群组的 test2, test3 两个人与 test1 同样是 testgroup 的群组账号，则仅可读可执行但不能写 (亦即不能修改)；
 - 至于非 testgoup 这一个群组的人则仅可以读，不能写也不能执行！

例题：

如果我的目录为底下的样式，请问 testgroup 这个群组的成员与其他人(others)是否可以进入本目录？

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

答：

- 档案拥有者 test1[rwx]可以在本目录中进行任何工作；
- 而 testgroup 这个群组[r-x]的账号，例如 test2, test3 亦可以进入本目录进行工作，但是不能在本目录下进行写入的动作；
- 至于 other 的权限中[r--]虽然有 r，但是由于没有 x 的权限，因此 others 的使用者，并不能进入此目录！

- Linux 档案权限的重要性：

与 Windows 系统不一样的是，在 Linux 系统当中，每一个档案都多加了很多的属性进来，尤其是群组的概念，这样有什么用途呢？其实，最大的用途是在『数据安全性』上面的。

- 系统保护的功能：
举个简单的例子，在你的系统中，关于系统服务的档案通常只有 root 才能读写或者是执行，例如/etc/shadow 这一个账号管理的档案，由于该档案记录了系统中所有账号的数据，因此是很重要的一个配置文件，当然不能让任何人读取 (否则密码会被窃取啊)，只有 root 才能够来读取啰！

所以该档案的权限就会成为[`-rw-----`]啰！

- 团队开发软件或数据共享的功能：
此外，如果你有一个软件开发团队，在你的团队中，你希望每个人都可以使用某一些目录下的档案，而非你的团队的其他人则不予以开放呢？以上面的例子来说，testgroup 的团队共有三个人，分别是 test1, test2, test3，那么我就可以将团队所需的档案权限订为[`-rwxrwx---`]来提供给 testgroup 的工作团队使用啰！
- 未将权限设定妥当的危害：
再举个例子来说，如果你的目录权限没有作好的话，可能造成其他人都可以在你的系统上面乱搞啰！例如本来只有 root 才能做的开关机、ADSL 的拨接程序、新增或删除用户等等的指令，若被你改成任何人都可以执行的话，那么如果使用者不小心给你重新启动啦！重新拨接啦！等等的！那么你的系统不就会常常莫名其妙的挂掉啰！而且万一你的用户的密码被其他不明人士取得的话，只要他登入你的系统就可以轻而易举的执行一些 root 的工作！

可怕吧！因此，在你修改你的 linux 档案与目录的属性之前，一定要先搞清楚，什么数据是可变的，什么是不可变的！千万注意啰！接下来我们来处理一下文件属性与权限的变更吧！

如何改变文件属性与权限

我们现在知道档案权限对于一个系统的安全性重要了，也知道档案的权限对于使用者与群组的相关性，那么如何修改一个档案的属性与权限呢？又！有多少档案的权限我们可以修改呢？其实一个档案的属性与权限有很多！我们先介绍几个常用于群组、拥有者、各种身份的权限之修改的指令，如下所示：

- `chgrp` : 改变档案所属群组
- `chown` : 改变档案拥有者
- `chmod` : 改变档案的权限, SUID, SGID, SBIT 等等的特性

-
- 改变所属群组, `chgrp`

改变一个档案的群组真是很简单的，直接以 `chgrp` 来改变即可，咦！这个指令就

是 change group 的缩写嘛！这样就很好记了吧！^_^。不过，请记住，要被改变的组名必须要在/etc/group 档案内存在才行，否则就会显示错误！

假设你是以 root 的身份登入 Linux 系统的，那么在你的家目录内有一个 install.log 的档案，如何将该档案的群组改变一下呢？假设你已经知道在 /etc/group 里面已经存在一个名为 users 的群组，但是 testing 这个群组名字就不存在/etc/group 当中了，此时改变群组成为 users 与 testing 分别会有什么现象发生呢？

```
[root@www ~]# chgrp [-R] dirname/filename ...
选项：
-R : 进行递归(recursive)的持续变更，亦即连同次目录下的所有档案、目录
      都更新成为这个群组之意。常常用在变更某一目录内所有的档案之情况。
范例：
[root@www ~]# chgrp users install.log
[root@www ~]# ls -l
-rw-r--r--  1 root users 68495 Jun 25 08:53 install.log
[root@www ~]# chgrp testing install.log
chgrp: invalid group name `testing' <== 发生错误讯息
啰~找不到这个群组名~
```

发现了吗？档案的群组被改成 users 了，但是要改成 testing 的时候，就会发生错误~注意喔！发生错误讯息还是要努力的查一查错误讯息的内容才好！将他英文翻译成为中文，就知道问题出在哪里了。

-
- 改变档案拥有者，chown

如何改变一个档案的拥有者呢？很简单呀！既然改变群组是 change group，那么改变拥有者就是 change owner 啰！BINGO！那就是 chown 这个指令的用途，要注意的是，用户必须是已经存在系统中的账号，也就是在/etc/passwd 这个档案中有纪录的用户名称才能改变。

chown 的用途还满多的，他还可以顺便直接修改群组的名称呢！此外，如果要连目录下的所有次目录或档案同时更改档案拥有者的话，直接加上 -R 的选项即可！我们来看看语法与范例：

```
[root@www ~]# chown [-R] 账号名称 档案或目录
[root@www ~]# chown [-R] 账号名称:组名 档案或目录
选项：
```

-R : 进行递归(recursive)的持续变更, 亦即连同次目录下的所有档案都变更

范例: 将 install.log 的拥有者改为 bin 这个账号:

```
[root@www ~]# chown bin install.log
[root@www ~]# ls -l
-rw-r--r-- 1 bin users 68495 Jun 25 08:53
install.log
```

范例: 将 install.log 的拥有者与群组改回为 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
```

Tips:

事实上, chown 也可以使用『chown user.group file』, 亦即在拥有者与群组间加上小数点『.』也行! 不过很多朋友设定账号时, 喜欢在账号当中加入小数点(例如 vbird.tsai 这样的账号格式), 这就会造成系统的误判了! 所以我们比较建议使用冒号『:』来隔开拥有者与群组啦! 此外, chown 也能单纯的修改所属群组呢! 例如

『chown .sshd install.log』就是修改群组~看到了吗? 就是那个小数点的用途!



知道如何改变档案的群组与拥有者了, 那么什么时候要使用 chown 或 chgrp 呢? 或许你会觉得奇怪吧? 是的, 确实有时候需要变更档案的拥有者的, 最常见的例子就是在复制档案给你之外的其他人时, 我们使用最简单的 cp 指令来说明好了:

```
[root@www ~]# cp 来源档案 目标文件
```

假设你今天要将 .bashrc 这个档案拷贝成为 .bashrc_test 档名, 且是要给 bin 这个人, 你可以这样做:

```
[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
<==新档案的属性没变
```

由于复制行为(cp)会复制执行者的属性与权限, 所以! 怎么办? .bashrc_test 还是属于 root 所拥有, 如此一来, 即使你将档案拿给 bin 这个使用者了, 那他仍然无法修改的(看属性/权限就知道了吧), 所以你就必须要将这个档案的拥有

者与群组修改一下啰！知道如何修改了吧？

- 改变权限, `chmod`

档案权限的改变使用的是 `chmod` 这个指令，但是，权限的设定方法有两种，分别可以使用数字或者是符号来进行权限的变更。我们就来谈一谈：

- 数字类型改变档案权限

Linux 档案的基本权限就有九个，分别是 owner/group/others 三种身份各有自己的 read/write/execute 权限，先复习一下刚刚上面提到的数据：档案的权限字符为：『-rwxrwxrwx』，这九个权限是三个三个一组的！其中，我们可以使用数字来代表各个权限，各权限的分数对照表如下：

r:4

w:2

x:1

每种身份(owner/group/others)各自的三个权限(r/w/x)分数是需要累加的，例如当权限为：[-rwxrwx---] 分数则是：

owner = rwx = 4+2+1 = 7

group = rwx = 4+2+1 = 7

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

所以等一下我们设定权限的变更时，该档案的权限数字就是 770 啦！变更权限的指令 `chmod` 的语法是这样的：

```
[root@www ~]# chmod [-R] xyz 档案或目录
选项：
xyz : 就是刚刚提到的数字类型的权限属性，为 rwx 属性
      数值的相加。
-R : 进行递归(recursive)的持续变更，亦即连同次目录下
      的所有档案都会变更
```

举例来说，如果要将 `.bashrc` 这个档案所有的权限都设定启用，那么就下

达：

```
[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
```

那如果要将权限变成『 -rwxr-xr-- 』呢？那么权限的分数就成为
[4+2+1][4+0+1][4+0+0]=754 啰！所以你需要下达『 chmod 754
filename』。另外，在实际的系统运作中最常发生的一个问题就是，常
常我们以 vim 编辑一个 shell 的文字批处理文件后，他的权限通常是
-rw-rw-r-- 也就是 664， 如果要将该档案变成可执行文件，并且不要让
其他人修改此一档案的话， 那么就需要-rwxr-xr-x 这样的权限，此时就
得要下达：『 chmod 755 test.sh 』的指令啰！

另外，如果有些档案你不希望被其他人看到，那么应该将档案的权限设定
为例如：『-rwxr-----』，那就下达『 chmod 740 filename 』吧！

例题：
将刚刚你的.bashrc 这个档案的权限修改回-rw-r--r--的情况吧！
答：
-rw-r--r--的分数是 644，所以指令为：
chmod 644 .bashrc

• 符号类型改变档案权限

还有一个改变权限的方法呦！从之前的介绍中我们可以发现，基本上就九个权限分别是(1)user (2)group (3)others 三种身份啦！那么我们就可以藉由 u, g, o 来代表三种身份的权限！此外， a 则代表 all 亦即全部的身份！那么读写的权限就可以写成 r, w, x 啰！也就是可以使用底下的方式来看：

chmod	u	+(加入)	r	档案或目录
	g	-(除去)	w	
	o	=(设定)	x	
	a			

来实作一下吧！假如我们要『设定』一个档案的权限成为『-rwxr-xr-x』时，基本上就是：

- user (u)：具有可读、可写、可执行的权限；
- group 与 others (g/o)：具有可读与执行的权限。

所以就是：

```
[root@www ~]# chmod u=rwx,go=rx .bashrc
# 注意喔！那个 u=rwx,go=rx 是连在一起的，中间并没有任何空格符！
[root@www ~]# ls -al .bashrc
-rwxr-xr-x 1 root root 395 Jul  4 11:45 .bashrc
```

那么假如是『-rwxr-xr--』这样的权限呢？可以使用『chmod u=rwx,g=rx,o=r filename』来设定。此外，如果我不知道原先的文件属性，而我只想要增加.bashrc 这个档案的每个人均可写入的权限，那么我就可以使用：

```
[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
```

而如果是要将权限去掉而不更动其他已存在的权限呢？例如要拿掉全部人的可执行权限，则：

```
[root@www ~]# chmod a-x .bashrc
[root@www ~]# ls -al .bashrc
-rw-rw-rw- 1 root root 395 Jul  4 11:45 .bashrc
```

知道 +, -, = 的不同点了吗？对啦！+ 与 - 的状态下，只要是没有指定到的项目，则该权限『不会被变动』，例如上面的例子中，由于仅以 - 拿掉 x 则其他两个保持当时的值不变！多多实作一下，你就会知道如何改变权限啰！这在某些情况底下很好用的～举例来说，你想要教一个朋友如何让一个程序可以拥有执行的权限，但你又不知道该档案原本的权限为何，此时，利用『chmod a+x filename』，就可以让该程序拥有执行的权限了。是否很方便？

🐦 目录与档案之权限意义:

现在我们知道了 Linux 系统内档案的三种身份(拥有者、群组与其他人), 知道每种身份都有三种权限(rwx), 已知道能够使用 `chown`, `chgrp`, `chmod` 去修改这些权限与属性, 当然, 利用 `ls -l` 去观察档案也没问题。前两小节也谈到了这些档案权限对于数据安全的重要性。那么, 这些档案权限对于一般档案与目录档案有何不同呢? 有大大不同啊! 底下就让鸟哥来说清楚, 讲明白!

- 权限对档案的重要性

档案是实际含有数据的地方, 包括一般文本文件、数据库内容文件、二进制可执行文件(binary program)等等。因此, 权限对于档案来说, 他的意义是这样的:

- `r` (read): 可读取此一档案的实际内容, 如读取文本文件的文字内容等;
- `w` (write): 可以编辑、新增或者是修改该档案的内容(但不含删除该档案);
- `x` (eXecute): 该档案具有可以被系统执行的权限。

那个可读(`r`)代表读取档案内容是还好了解, 那么可执行(`x`)呢? 这里你就必须要小心啦! 因为在 Windows 底下一个档案是否具有执行的能力是藉由『扩展名』来判断的, 例如: `.exe`, `.bat`, `.com` 等等, 但是在 Linux 底下, 我们的档案是否能被执行, 则是藉由是否具有『`x`』这个权限来决定的! 跟档名是没有绝对的关系的!

至于最后一个 `w` 这个权限呢? 当你对一个档案具有 `w` 权限时, 你可以具有写入/编辑/新增/修改档案的内容的权限, 但并不具备有删除该档案本身的权限! 对于档案的 `rwx` 来说, 主要都是针对『档案的内容』而言, 与档案档名的存在与否没有关系喔! 因为档案记录的是实际的数据嘛!

- 权限对目录的重要性

档案是存放实际数据的所在, 那么目录主要是储存啥玩意啊? 目录主要的内容在记录文件名列表, 文件名与目录有强烈的关连啦! 所以如果是针对目录时, 那个 `r`, `w`, `x` 对目录是什么意思呢?

- `r` (read contents in directory):

表示具有读取目录结构列表的权限, 所以当你具有读取(`r`)一个目录的权限时, 表示你可以查询该目录下的文件名数据。所以你就可以利用 `ls` 这个指令将该目录的内容列表显示出来!

- `w` (modify contents of directory):

这个可写入的权限对目录来说，是很了不起的！ 因为他表示你具有异动该目录结构列表的权限，也就是底下这些权限：

- 建立新的档案与目录；
- 删除已经存在的档案与目录(不论该档案的权限为何！)
- 将已存在的档案或目录进行更名；
- 搬移该目录内的档案、目录位置。

总之，目录的 w 权限就与该目录底下的文件名异动有关就对了啦！

- x (access directory):

咦！目录的执行权限有啥用途啊？目录只是记录文件名而已，总不能拿来执行吧？没错！目录不可以被执行，目录的 x 代表的是用户能否进入该目录成为工作目录的用途！ 所谓的工作目录(work directory)就是你目前所在的目录啦！举例来说，当你登入 Linux 时， 你所在的家目录就是你当下的工作目录。而变换目录的指令是『cd』(change directory)啰！

大致的目录权限概念是这样，底下我们来看几个范例，让你了解一下啥是目录的权限啰！

例题：

有个目录的权限如下所示：

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

系统有个账号名称为 vbird，这个账号并没有支持 root 群组，请问 vbird 对这个目录有何权限？是否可切换到此目录中？

答：

vbird 对此目录仅具有 r 的权限，因此 vbird 可以查询此目录下的文件名列表。因为 vbird 不具有 x 的权限，因此 vbird 并不能切换到此目录内！（相当重要的概念！）

上面这个例题中因为 vbird 具有 r 的权限，因为是 r 乍看之下好像就具有可以进入此目录的权限，其实那是错的。 能不能进入某一个目录，只与该目录的 x 权限有关啦！此外， 工作目录对于指令的执行是非常重要的，如果你在某目录下不具有 x 的权限， 那么你就无法切换到该目录下，也就无法执行该目录下的任何指令，即使你具有该目录的 r 的权限。

很多朋友在架设网站的时候都会卡在一些权限的设定上，他们开放目录数据给因特网的任何人来浏览， 却只开放 r 的权限，如上面的范例所示那样，那样的结果就是导致网站服务器软件无法到该目录下读取档案(最多只能看到文件名)，最终用户总是无法正确的查阅到档案的内容(显示权限不足啊！)。要注意：要开

放目录给任何人浏览时,应该至少也要给予 r 及 x 的权限,但 w 权限不可随便给!为什么 w 不能随便给,我们来看下一个例子:

例题:

假设有个账号名称为 dmtsai, 他的家目录在 /home/dmtsai/, dmtsai 对此目录具有 [rwx] 的权限。若在此目录下有个名为 the_root.data 的档案, 该档案的权限如下:

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

请问 dmtsai 对此档案的权限为何? 可否删除此档案?

答:

如上所示, 由于 dmtsai 对此档案来说是『others』的身份, 因此这个档案他无法读、无法编辑也无法执行, 也就是说, 他无法变动这个档案的内容就是了。

但是由于这个档案在他的家目录下, 他在此目录下具有 rwx 的完整权限, 因此对于 the_root.data 这个『档名』来说, 他是能够『删除』的! 结论就是, dmtsai 这个用户能够删除 the_root.data 这个档案!

还是看不太懂? 有听没有懂喔! 没关系~我们底下就来设计一个练习, 让你实际玩玩看, 应该就能够比较近入状况啦! 不过, 由于很多指令我们还没有教, 所以底下的指令有的先了解即可, 详细的指令用法我们会在后面继续介绍的。

- 先用 root 的身份建立所需要的档案与目录环境

我们用 root 的身份在所有人都可以工作的 /tmp 目录中建立一个名为 testing 的目录, 该目录的权限为 744 且目录拥有者为 root。另外, 在 testing 目录下在建立一个空的档案, 档名亦为 testing。建立目录可用 mkdir(make directory), 建立空档案可用 touch(下一章会说明)来处理。所以过程如下所示:

```
[root@www ~]# cd /tmp <==切换工作目录到/tmp
[root@www tmp]# mkdir testing <==建立新目录
[root@www tmp]# chmod 744 testing <==变更权限
[root@www tmp]# touch testing/testing <==建立空的档案
[root@www tmp]# chmod 600 testing/testing <==变更权限
[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
# 仔细看一下，目录的权限是 744，且所属群组与使用者
均是 root 喔！
# 那么在这样的情况下，一般身份用户对这个目录/档案
的权限为何？
```

- 一般用户的读写权限为何？观察中

在上面的例子中，虽然目录是 744 的权限设定，一般用户应该能有 r 的权限，但这样的权限使用者能做啥事呢？假设鸟哥的系统中含有一个账号名为 vbird 的，我们可以透过『su - vbird』这个指令来变换身份喔！看看底下的操作先！

```
[root@www tmp]# su - vbird <==切换身份成为 vbird 啰！
[vbird@www ~]$ cd /tmp      <==看一下，身份变了喔！提示
字符也变成 $ 了！
[vbird@www tmp]$ ls -l testing/
?----- ? ? ? ?           ? testing
# 因为具有 r 的权限可以查询档名。不过权限不足(没有 x)，
所以会有一堆问号。
[vbird@www tmp]$ cd testing/
-bash: cd: testing/: Permission denied
# 因为不具有 x，所以当然没有进入的权限啦！有没有呼
应前面的权限说明啊！
```

- 如果该目录属于用户本身，会有什么状况？

上面的练习我们知道了只有 r 确实可以让用户读取目录的文件名列表，不过详细的信息却还是读不到的，同时也不能将该目录变成工作目录(用 cd 进入该目录之意)。那如果我们让该目录变成用户的，那么用户在这个目录底下是否能够删除档案呢？底下的练习做看看：

```
[vbird@www tmp]$ exit          <==让 vbird 变回
原本的 root 身份喔！
[root@www tmp]# chown vbird testing <==修改权限，让
vbird 拥有此目录
[root@www tmp]# su - vbird      <==再次变成 vbird
来操作
[vbird@www ~]$ cd /tmp/testing  <==可以进入目录
了呢！
[vbird@www testing]$ ls -l
```

```
-rw----- 1 root root 0 Sep 19 16:01 testing <==档案不是 vbird 的！  
[vbird@www testing]$ rm testing <==尝试杀掉这个档案看看！  
rm: remove write-protected regular empty file  
`testing'? y  
# 竟然可以删除！这样理解了吗？！
```

透过上面这个简单的步骤，你就可以清楚的知道，`x` 在目录当中是与『能否进入该目录』有关，至于那个 `w` 则具有相当重要的权限，因为他可以让使用者删除、更新、新建档案或目录，是个很重要的参数啊！这样可以理解了吗？！^{^_^}

Linux 档案种类与扩展名

我们在基础篇一直强调一个概念，那就是：任何装置在 Linux 底下都是档案，不仅如此，连数据沟通的接口也有专属的档案在负责～所以，你会了解到，Linux 的档案种类真的很多～除了前面提到的一般档案(-)与目录档案(d)之外，还有哪些种类的档案呢？

- 档案种类：

我们在刚刚提到使用『`ls -l`』观察到第一栏那十个字符中，第一个字符为档案的类型。除了常见的一般档案(-)与目录档案(d)之外，还有哪些种类的文件类型呢？

- 正规档案(regular file)：

就是一般我们在进行存取类型的档案，在由 `ls -al` 所显示出来的属性方面，第一个字符为 `[-]`，例如 `[-rwxrwxrwx]`。另外，依照档案的内容，又大略可以分为：

- 纯文本档(ASCII)：这是 Linux 系统中最多种的一种文件类型，称为纯文本档是因为内容为我们人类可以直接读到的数据，例如数字、字母等等。几乎只要我们可以用来做为设定的档案都属于这一种文件类型。举例来说，你可以下达『`cat ~/.bashrc`』就可以看到该档案的内容。(cat 是将一个档案内容读出来的指令)
- 二进制文件(binary)：还记得我们在『[第零章、计算器概论](#)』里面的[软件程序的运作](#)中提过，我们的系统其实仅认识且可以执行二进制文件(binary file)吧？没错～你的 Linux 当中的可执行文件(scripts，文字型批处理文件不算)就是这种格式的啦～举例来说，刚刚下达的指令 cat 就是一个 binary file。
- 数据格式文件(data)：有些程序在运作的过程当中会读取某些特

定格式的档案，那些特定格式的档案可以被称为数据文件 (data file)。举例来说，我们的 Linux 在使用者登入时，都会将登录的数据记录在 /var/log/wtmp 那个档案内，该档案是一个 data file，他能够透过 last 这个指令读出来！但是使用 cat 时，会读出乱码～因为他是属于一种特殊格式的档案。瞭乎？

- 目录(directory):
就是目录啰～第一个属性为 [d]，例如 [drwxrwxrwx]。
- 连结档(link):
就是类似 Windows 系统底下的快捷方式啦！第一个属性为 [l] (英文 L 的小写)，例如 [lrwxrwxrwx] ；
- 设备与装置文件(device):
与系统周边及储存等相关的一些档案，通常都集中在/dev 这个目录之下！通常又分为两种：
 - 区块(block)设备档：就是一些储存数据，以提供系统随机存取的接口设备，举例来说，硬盘与软盘等就是啦！你可以随机的在硬盘的不同区块读写，这种装置就是成组设备啰！你可以自行查一下/dev/sda 看看，会发现第一个属性为[b]喔！
 - 字符(character)设备文件：亦即是一些串行端口的接口设备，例如键盘、鼠标等等！这些设备的特色就是『一次性读取』的，不能够截断输出。举例来说，你不可能让鼠标『跳到』另一个画面，而是『滑动』到另一个地方啊！第一个属性为 [c]。
- 资料接口文件(sockets):
既然被称为数据接口文件，想当然尔，这种类型的档案通常被用在网络上的数据承接了。我们可以启动一个程序来监听客户端的要求，而客户端就可以透过这个 socket 来进行数据的沟通了。第一个属性为 [s]，最常在/var/run 这个目录中看到这种文件类型了。
- 数据输送文件(FIFO, pipe):
FIFO 也是一种特殊的文件类型，他主要的目的在解决多个程序同时存取一个档案所造成的错误问题。FIFO 是 first-in-first-out 的缩写。第一个属性为[p] 。

除了设备文件是我们系统中很重要的档案，最好不要随意修改之外(通常他也不会让你修改的啦!)，另一个比较有趣的档案就是连结档。如果你常常将应用程序捉到桌面来的话，你就应该知道在 Windows 底下有所谓的『快捷方式』。同样的，你可以将 linux 下的连结档简单的视为一个档案或目录的快捷方式。至于 socket 与 FIFO 档案比较难理解，因为这两个咚咚与程序(process)比较有关系，这个等到未来你了解 process 之后，再回来查阅吧！此外，你也可以透过

man fifo 及 man socket 来查阅系统上的说明！

- Linux 档案扩展名：

基本上，Linux 的档案是没有所谓的『扩展名』的，我们刚刚就谈过，一个 Linux 档案能不能被执行，与他的第一栏的十个属性有关，与文件名根本一点关系也没有。这个观念跟 Windows 的情况不相同喔！在 Windows 底下，能被执行的档案扩展名通常是 .com .exe .bat 等等，而在 Linux 底下，只要你的权限当中具有 x 的话，例如 [-rwx-r-xr-x] 即代表这个档案可以被执行喔！

不过，可以被执行跟可以执行成功是不一样的～举例来说，在 root 家目录下的 install.log 是一个纯文本档，如果经由修改权限成为 -rwxrwxrwx 后，这个档案能够真的执行成功吗？当然不行～因为他的内容根本就没有可以执行的数据。所以说，这个 x 代表这个档案具有可执行的能力，但是能不能执行成功，当然就得要看该档案的内容啰～

虽然如此，不过我们仍然希望可以藉由扩展名来了解该档案是什么东西，所以，通常我们还是会以适当的扩展名来表示该档案是什么种类的。底下有数种常用的扩展名：

- *.sh：批处理文件 (scripts)，因为批处理文件为使用 shell 写成的，所以扩展名就编成 .sh 啰；
- *.Z, *.tar, *.tar.gz, *.zip, *.tgz：经过打包的压缩文件。这是因为压缩软件分别为 gunzip, tar 等等的，由于不同的压缩软件，而取其相关的扩展名啰！
- *.html, *.php：网页相关档案，分别代表 HTML 语法与 PHP 语法的网页档案啰！.html 的档案可使用网页浏览器来直接开启，至于 .php 的档案，则可以透过 client 端的浏览器来 server 端浏览，以得到运算后的网页结果呢！

基本上，Linux 系统上的文件名真的只是让你了解该档案可能的用途而已，真正的执行与否仍然需要权限的规范才行！例如虽然有一个档案为可执行文件，如常见的 /bin/ls 这个显示文件属性的指令，不过，如果这个档案的权限被修改成无法执行时，那么 ls 就变成不能执行啰！

上述的这种问题最常发生在档案传送的过程中。例如你在网络上下载一个可执行文件，但是偏偏在你的 Linux 系统中就是无法执行！呵呵！那么就是可能档案的属性被改变了！不要怀疑，从网络上传送到你的 Linux 系统中，档案的属性与权限确实是会被改变的喔！

- Linux 档案长度限制：

在 Linux 底下，使用预设的 Ext2/Ext3 文件系统时，针对档案的档名长度限制为：

- 单一档案或目录的最大容许文件名为 255 个字符；
- 包含完整路径名称及目录 (/) 之完整档名为 4096 个字符。

是相当长的档名喔！我们希望 Linux 的文件名可以一看就知道该档案在干嘛的，所以档名通常是很长很长！而用惯了 Windows 的人可能会受不了，因为文件名通常真的都很长，对于用惯 Windows 而导致打字速度不快的朋友来说，嗯！真的是很困扰.....不过，只得劝你好好的加强打字的训练啰！

而由[第五章谈到的热键](#)你也会知道，其实可以透过[tab]按键来确认档案的文件名的！这很好用啊！当然啦，如果你已经读完了本书第三篇关于 [BASH](#) 的用法，那么你将会发现『[哇！变量真是一个相当好用的东西呐！](#)』嗯！看不懂，没关系，到第三篇谈到 bash 再说！

- Linux 文件名的限制：

由于 Linux 在文字接口下的一些指令操作关系，一般来说，你在设定 Linux 底下的文件名时，最好可以避免一些特殊字符比较好！例如底下这些：

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

因为这些符号在文字接口下，是有特殊意义的！另外，文件名的开头为小数点『.』时，代表这个档案为『隐藏档』喔！同时，由于指令下达当中，常常会使用到 -option 之类的选项，所以你最好也避免将档案档名的开头以 - 或 + 来命名啊！



Linux 目录配置

在了解了每个档案的相关种类与属性，以及了解了如何更改文件属性/权限的相关信息后，再来要了解的就是，为什么每套 Linux distributions 他们的配置文件啊、执行文件啊、每个目录内放置的咚咚啊，其实都差不多？原来是有一套标准依据的哩！我们底下就来瞧一瞧。



Linux 目录配置的依据--FHS

因为利用 Linux 来开发产品或 distributions 的社群/公司与个人实在太多了，如果每个人都用自己的想法来配置档案放置的目录，那么将可能造成很多管理上的困扰。你能想象，你进入一个企业之后，所接触到的 Linux 目录配置方法竟然跟你以前学的完全不同吗？很难想象吧～所以，后来就有所谓的 Filesystem Hierarchy Standard (FHS) 标准的出炉了！

根据 FHS (<http://www.pathname.com/fhs/>) 的官方文件指出，他们的主要目的是希望让使用者可以了解到已安装软件通常放置于那个目录下，所以他们希望独立的软件开发商、操作系统制作者、以及想要维护系统的用户，都能够遵循 FHS 的标准。也就是说，FHS 的重点在于规范每个特定的目录下应该要放置什么样子的数据而已。这样做好处非常多，因为 Linux 操作系统就能够在既有的面貌下(目录架构不变)发展出开发者想要的独特风格。

事实上，FHS 是根据过去的经验一直再持续的改版的，FHS 依据文件系统使用的频繁与否与是否允许使用者随意更动，而将目录定义成为四种交互作用的形态，用表格来说有点像底下这样：

	可分享的(shareable)	不可分享的(unshareable)
不变的(static)	/usr (软件放置处)	/etc (配置文件)
	/opt (第三方协力软件)	/boot (开机与核心档)
可变动的(variable)	/var/mail (使用者邮件信箱)	/var/run (程序相关)
	/var/spool/news (新闻组)	/var/lock (程序相关)

上表中的目录就是一些代表性的目录，该目录下所放置的数据在底下会谈到，这里先略过不谈。我们要了解的是，什么是那四个类型？

- 可分享的：可以分享给其他系统挂载使用的目录，所以包括执行文件与用户的邮件等数据，是能够分享给网络上其他主机挂载用的目录；
- 不可分享的：自己机器上面运作的装置档案或者是与程序有关的 socket 档案等，由于仅与自身有关，所以当然就不适合分享给其他主机了。
- 不变的：有些数据是不会经常变动的，跟随着 distribution 而不变动。例如函式库、文件说明文件、系统管理员所管理的主机服务配置文件等等；
- 可变动的：经常改变的数据，例如登录文件、一般用户可自行收受的新闻组等。

事实上，FHS 针对目录树架构仅定义出三层目录下应该放置什么数据而已，分别是底下这三个目录的定义：

- / (root, 根目录)：与开机系统有关；
- /usr (unix software resource)：与软件安装/执行有关；
- /var (variable)：与系统运作过程有关。

为什么要定义出这三层目录呢？其实是有意义的喔！每层目录下所应该要放置

的目录也都又特定的规定喔！由于我们尚未介绍完整的 Linux 系统，所以底下的介绍你可能会看不懂！没关系，先由个概念即可，等到妳将基础篇全部看完后，就重头将基础篇再看一遍！到时候你就会豁然开朗啦！^_^

Tips:

这个 root 在 Linux 里面的意义真的很多很多～多到让人搞不懂那是啥玩意儿。如果以『账号』的角度来看，所谓的 root 指的是『系统管理员！』的身份，如果以『目录』的角度来看，所谓的 root 意即指的是根目录，就是 / 啦～要特别注意喔！



• 根目录 (/) 的意义与内容:

根目录是整个系统最重要的一个目录，因为不但所有的目录都是由根目录衍生出来的，同时根目录也与开机/还原/系统修复等动作有关。由于系统开机时需要特定的开机软件、核心档案、开机所需程序、函式库等等档案数据，若系统出现错误时，根目录也必须包含有能够修复文件系统的程序才行。因为根目录是这么的重要，所以在 FHS 的要求方面，他希望根目录不要放在非常大的分割槽内，因为越大的分割槽妳会放入越多的数据，如此一来根目录所在分割槽就可能会有较多发生错误的机会。

因此 FHS 标准建议：根目录 (/) 所在分割槽应该越小越好，且应用程序所安装的软件最好不要与根目录放在同一个分割槽内，保持根目录越小越好。如此不但效能较佳，根目录所在的文件系统也较不容易发生问题。

有鉴于上述的说明，因此 FHS 定义出根目录 (/) 底下应该要有底下这些次目录的存在才好：

目录	应放置档案内容
/bin	系统有很多放置执行文件的目录，但/bin 比较特殊。因为/bin 放置的是在单人维护模式下还能够被操作的指令。在/bin 底下的指令可以被 root 与一般账号所使用，主要有：cat, chmod, chown, date, mv, mkdir, cp, bash 等等常用的指令。
/boot	这个目录主要在放置开机会使用到的档案，包括 Linux 核心档案以及开机选单与开机所需配置文件等等。Linux kernel 常用的档名为：vmlinuz, 如果使用的是 grub 这个开机管理程序，则还会存在/boot/grub/这个目录喔！
/dev	在 Linux 系统上，任何装置与接口设备都是以档案的型态存在于这个目录当中的。你只要透过存取这个目录底下的某个档案，就等于存取某个装置啰～ 比要重要的档案有/dev/null, /dev/zero, /dev/tty, /dev/lp*, /dev/hd*, /dev/sd*等等
/etc	系统主要的配置文件几乎都放置在这个目录内，例如人员的账号密码文件、各种服务的启始档等等。一般来说，这个目录下

	<p>的各文件属性是可以让一般使用者查阅的，但是只有 root 有权力修改。FHS 建议不要放置可执行文件(binary)在这个目录中喔。比较重要的档案有： /etc/inittab, /etc/init.d/, /etc/modprobe.conf, /etc/X11/, /etc/fstab, /etc/sysconfig/ 等等。另外，其下重要的目录有：</p> <ul style="list-style-type: none"> • /etc/init.d/：所有服务的预设启动 script 都是放在这里的，例如要启动或者关闭 iptables 的话： 『 /etc/init.d/iptables start』、 『/etc/init.d/iptables stop』 • /etc/xinetd.d/：这就是所谓的 super daemon 管理的各项服务的配置文件目录。 • /etc/X11/：与 X Window 有关的各种配置文件都在这里，尤其是 xorg.conf 或 XF86Config 这两个 X Server 的配置文件。
/home	<p>这是系统默认的用户家目录(home directory)。在你新增一个一般使用者账号时，默认的用户家目录都会规范到这里来。比较重要的是，家目录有两种代号喔： ~：代表目前这个用户的家目录，而 ~dmtsai：则代表 dmtsai 的家目录！</p>
/lib	<p>系统的函式库非常的多，而/lib 放置的则是在开机时会用到的函式库，以及在/bin或/sbin底下的指令会呼叫的函式库而已。什么是函式库呢？妳可以将他想成是『外挂』，某些指令必须要有这些『外挂』才能够顺利完成程序的执行之意。尤其重要的是/lib/modules/这个目录，因为该目录会放置核心相关的模块(驱动程序)喔！</p>
/media	<p>media 是『媒体』的英文，顾名思义，这个/media 底下放置的就是可移除的装置啦！包括软盘、光盘、DVD 等等装置都暂时挂载于此。常见的档名有：/media/floppy, /media/cdrom 等等。</p>
/mnt	<p>如果妳想要暂时挂载某些额外的装置，一般建议妳可以放置到这个目录中。在古早时候，这个目录的用途与/media 相同啦！只是有了/media 之后，这个目录就用来暂时挂载用了。</p>
/opt	<p>这个是给第三方协力软件放置的目录。什么是第三方协力软件啊？举例来说，KDE 这个桌面管理系统是一个独立的计划，不过他可以安装到 Linux 系统中，因此 KDE 的软件就建议放置到此目录下了。另外，如果妳想要自行安装额外的软件(非原本的 distribution 提供的)，那么也能够将你的软件安装到这里来。不过，以前的 Linux 系统中，我们还是习惯放置在 /usr/local 目录下呢！</p>
/root	<p>系统管理员(root)的家目录。之所以放在这里，是因为如果进入单人维护模式而仅挂载根目录时，该目录就能够拥有 root</p>

	的家目录，所以我们会希望 root 的家目录与根目录放置在同一个分割槽中。
/sbin	Linux有非常多指令是用来设定系统环境的,这些指令只有root才能够利用来『设定』系统,其他用户最多只能用来『查询』而已。 放在/sbin底下的为开机过程中所需要的,里面包括了开机、修复、还原系统所需要的指令。至于某些服务器软件程序,一般则放置到/usr/sbin/当中。至于本机自行安装的软件所产生的系统执行文件(system binary),则放置到/usr/local/sbin/当中了。常见的指令包括:fdisk, fsck, ifconfig, init, mkfs 等等。
/srv	srv 可以视为『service』的缩写,是一些网络服务启动之后,这些服务所需要取用的数据目录。常见的服务例如 WWW, FTP 等等。举例来说,WWW 服务器需要的网页资料就可以放置在 /srv/www/里面。
/tmp	这是让一般用户或者是正在执行的程序暂时放置档案的地方。这个目录是任何人都能够存取的,所以你需要定期的清理一下。当然,重要数据不可放置在此目录啊! 因为 FHS 甚至建议在开机时,应该要将/tmp 下的数据都删除唷!

事实上 FHS 针对根目录所定义的标准就仅有上面的咚咚,不过我们的 Linux 底下还有许多目录你也需要了解一下的。底下是几个在 Linux 当中也是非常重要的目录喔:

目录	应放置档案内容
/lost+found	这个目录是使用标准的 ext2/ext3 文件系统格式才会产生的一个目录,目的在于当文件系统发生错误时,将一些遗失的片段放置到这个目录下。这个目录通常会在分割槽的最顶层存在,例如你加装一颗硬盘于/disk 中,那在这个系统下就会自动产生一个这样的目录『/disk/lost+found』
/proc	这个目录本身是一个『虚拟文件系统(virtual filesystem)』喔!他放置的数据都是在内存当中,例如系统核心、行程信息(process)、周边装置的状态及网络状态等等。因为这个目录下的数据都是在内存当中,所以本身不占任何硬盘空间啊!比较重要的档案例如: /proc/cpuinfo, /proc/dma, /proc/interrupts, /proc/ioports, /proc/net/* 等等。
/sys	这个目录其实跟/proc 非常类似,也是一个虚拟的文件系统,主要也是记录与核心相关的信息。包括目前已加载的核心模块与核心侦测到的硬件装置信息等等。这个目录同样不占硬盘容量喔!

除了这些目录的内容之外,另外要注意的是,因为根目录与开机有关,开机过程

中仅有根目录会被挂载，其他分割槽则是在开机完成之后才会持续的进行挂载的行为。就是因为如此，因此根目录下与开机过程有关的目录，就不能够与根目录放到不同的分割槽去！那哪些目录不可与根目录分开呢？有底下这些：

- /etc: 配置文件
- /bin: 重要执行档
- /dev: 所需要的装置档案
- /lib: 执行档所需的函式库与核心所需的模块
- /sbin: 重要的系统执行文件

这五个目录千万不可与根目录分开在不同的分割槽！请背下来啊！好了，谈完了根目录，接下来我们就来谈谈/usr 以及/var 啰！先看/usr 里面有些什么东西：

- /usr 的意义与内容：

依据 FHS 的基本定义，/usr 里面放置的数据属于可分享的与不可变动的 (shareable, static)，如果你知道如何透过网络进行分割槽的挂载 (例如在服务器篇会谈到的 [NFS 服务器](#))，那么/usr 确实可以分享给局域网络内的其他主机来使用喔！

很多读者都会误会/usr 为 user 的缩写，其实 usr 是 Unix Software Resource 的缩写，也就是『Unix 操作系统软件资源』所放置的目录，而不是用户的数据啦！这点要注意。FHS 建议所有软件开发者，应该将他们的数据合理的分别放置到这个目录下的次目录，而不要自行建立该软件自己独立的目录。

因为是所有系统默认的软件 (distribution 发布者提供的软件) 都会放置到/usr 底下，因此这个目录有点类似 Windows 系统的『C:\Windows\ + C:\Program files\』这两个目录的综合体，系统刚安装完毕时，这个目录会占用最多的硬盘容量。一般来说，/usr 的次目录建议有底下这些：

目录	应放置档案内容
/usr/X11R6/	为 X Window System 重要数据所放置的目录，之所以取名为 X11R6 是因为最后的 X 版本为第 11 版，且该版的第 6 次释出之意。
/usr/bin/	绝大部分的用户可使用指令都放在这里！请注意到他与 /bin 的不同之处。（是否与开机过程有关）
/usr/include/	c/c++等程序语言的档头(header)与包含档(include)放置处，当我们以 tarball 方式 (*.tar.gz 的方式安装软件) 安装某些数据时，会使用到里头的许多包含档喔！
/usr/lib/	包含各应用软件的函式库、目标档案(object file)，以及不被一般使用者惯用的执行档或脚本(script)。某些软件会提供一些特殊的指令来进行服务器的设定，这些指令也不会经常被系统管理员操作，那就会被摆放到这个目录下

	啦。要注意的是，如果你使用的是 X86_64 的 Linux 系统，那可能会有 /usr/lib64/ 目录产生喔！
/usr/local/	系统管理员在本机自行安装自己下载的软件(非 distribution 默认提供者)，建议安装到此目录，这样会比较便于管理。举例来说，你的 distribution 提供的软件较旧，你想安装较新的软件但又不想移除旧版，此时你可以将新版软件安装于 /usr/local/ 目录下，可与原先的旧版软件有分别啦！你可以自行到 /usr/local 去看看，该目录下也是具有 bin, etc, include, lib... 的次目录喔！
/usr/sbin/	非系统正常运作所需要的系统指令。最常见的就是某些网络服务器软件的服务指令(daemon)啰！
/usr/share/	放置共享文件的地方，在这个目录下放置的数据几乎是不分硬件架构均可读取的数据，因为几乎都是文本文件嘛！在此目录下常见的还有这些次目录： <ul style="list-style-type: none"> • /usr/share/man: 联机帮助文件 • /usr/share/doc: 软件杂项的文件说明 • /usr/share/zoneinfo: 与时区有关的时区档案
/usr/src/	一般原始码建议放置到这里，src 有 source 的意思。至于核心原始码则建议放置到 /usr/src/linux/ 目录下。

- /var 的意义与内容：

如果 /usr 是安装时会占用较大硬盘容量的目录，那么 /var 就是在系统运作后才会渐渐占用硬盘容量的目录。因为 /var 目录主要针对常态性变动的档案，包括快取(cache)、登录档(log file)以及某些软件运作所产生的档案，包括程序档案(lock file, run file)，或者例如 MySQL 数据库的档案等等。常见的次目录有：

目录	应放置档案内容
/var/cache/	应用程序本身运作过程中会产生的一些暂存档；
/var/lib/	程序本身执行的过程中，需要使用到的数据文件放置的目录。在此目录下各自的软件应该要有各自的目录。举例来说，MySQL 的数据库放置到 /var/lib/mysql/ 而 rpm 的数据库则放到 /var/lib/rpm 去！
/var/lock/	某些装置或者是档案资源一次只能被一个应用程序所使用，如果同时有两个程序使用该装置时，就可能产生一些错误的状况，因此就得要将该装置上锁(lock)，以确保该装置只会给单一软件所使用。举例来说，刻录机正在刻录一块光盘，你想一下，会不会有两个人同时在使用一个刻录机烧片？如

	果两个人同时刻录，那片子写入的是谁的资料？所以当第一个人在刻录时该刻录机就会被上锁，第二个人就得要该装置被解除锁定(就是前一个人用完了)才能够继续使用啰。
/var/log/	重要到不行！这是登录文件放置的目录！里面比较重要的档案如/var/log/messages, /var/log/wtmp(记录登入者的信息)等。
/var/mail/	放置个人电子邮件信箱的目录，不过这个目录也被放置到/var/spool/mail/目录中！通常这两个目录是互为链接文件啦！
/var/run/	某些程序或者是服务启动后，会将他们的 PID 放置在这个目录下喔！至于 PID 的意义我们会在后续章节提到的。
/var/spool/	这个目录通常放置一些队列数据，所谓的『队列』就是排队等待其他程序使用的数据啦！这些数据被使用后通常都会被删除。举例来说，系统收到新信会放置到/var/spool/mail/中，但使用者收下该信件后该封信原则上就会被删除。信件如果暂时寄不出去会被放到/var/spool/mqueue/中，等到被送出后就被删除。如果是工作排程数据(crontab)，就会被放置到/var/spool/cron/目录中！

建议在你读完整个基础篇之后，可以挑战 FHS 官方英文文件(参考本章[参考数据](#))，相信会让你对于 Linux 操作系统的目录有更深入的了解喔！

- 针对 FHS，各家 distributions 的异同

由于 FHS 仅是定义出最上层(/)及次层(/usr, /var)的目录内容应该要放置的档案或目录数据，因此，在其他次目录层级内，就可以随开发者自行来配置了。举例来说，CentOS 的网络设定数据放在 /etc/sysconfig/network-scripts/ 目录下，但是 SuSE 则是将网络放置在 /etc/sysconfig/network/ 目录下，目录名称可是不同的呢！不过只要记住大致的 FHS 标准，差异性其实有限啦！

目录树(directory tree)

另外，在 Linux 底下，所有的档案与目录都是由根目录开始的！那是所有目录与档案的源头～然后再一个一个的分支下来，有点像是树枝状啊～因此，我们也称这种目录配置方式为：『目录树(directory tree)』这个目录树有什么特性呢？他主要的特性有：

- 目录树的起始点为根目录 (/，root)；
- 每一个目录不止能使用本地端的 partition 的文件系统，也可以使用网络上的 filesystem。举例来说，可以利用 Network File System (NFS) 服务器挂载某特定目录等。

- 每一个档案在此目录树中的文件名(包含完整路径)都是独一无二的。

好，谈完了 FHS 的标准之后，实际来看看 CentOS 在根目录下会有什么样子的数据吧！我们可以下达以下的指令来查询：

```
[root@www ~]# ls -l /  
drwxr-xr-x  2 root root  4096 Sep  5 12:34 bin  
drwxr-xr-x  4 root root 1024 Sep  4 18:06 boot  
drwxr-xr-x 12 root root 4320 Sep 22 12:10 dev  
drwxr-xr-x 105 root root 12288 Sep 22 12:10 etc  
drwxr-xr-x  4 root root  4096 Sep  5 14:08 home  
drwxr-xr-x 14 root root  4096 Sep  5 12:12 lib  
drwx----- 2 root root 16384 Sep  5 01:49 lost+found  
drwxr-xr-x  2 root root  4096 Mar 30  2007 media  
drwxr-xr-x  2 root root    0 Sep 22 12:09 misc  
drwxr-xr-x  2 root root  4096 Mar 30  2007 mnt  
drwxr-xr-x  2 root root    0 Sep 22 12:09 net  
drwxr-xr-x  2 root root  4096 Mar 30  2007 opt  
dr-xr-xr-x 95 root root    0 Sep 22  2008 proc  
drwxr-xr-x  4 root root  4096 Sep  8 14:06 root  
drwxr-xr-x  2 root root 12288 Sep  5 12:33/sbin  
drwxr-xr-x  4 root root    0 Sep 22  2008 selinux  
drwxr-xr-x  2 root root  4096 Mar 30  2007 srv  
drwxr-xr-x 11 root root    0 Sep 22  2008 sys  
drwxrwxrwt  6 root root  4096 Sep 22 12:10 tmp  
drwxr-xr-x 14 root root  4096 Sep  4 18:00 usr  
drwxr-xr-x 26 root root  4096 Sep  4 18:19 var
```

上面表格中比较特殊的应该是/selinux 这个目录了，这个目录的内容数据也是在内存中的信息，同样的不会占用任何的硬盘容量。这个/selinux 是 Secure Enhance Linux(SELinux)的执行目录，而 SELinux 是 Linux 核心的重要外挂功能之一，他可以用来作为细部权限的控管，主要针对程序(尤其是网络程序)的访问权限来限制。关于 SELinux 我们会在后续的章节继续做介绍的喔！

如果我们将整个目录树以图标的方法来显示，并且将较为重要的档案数据列出来的话，那么目录树架构有点像这样：

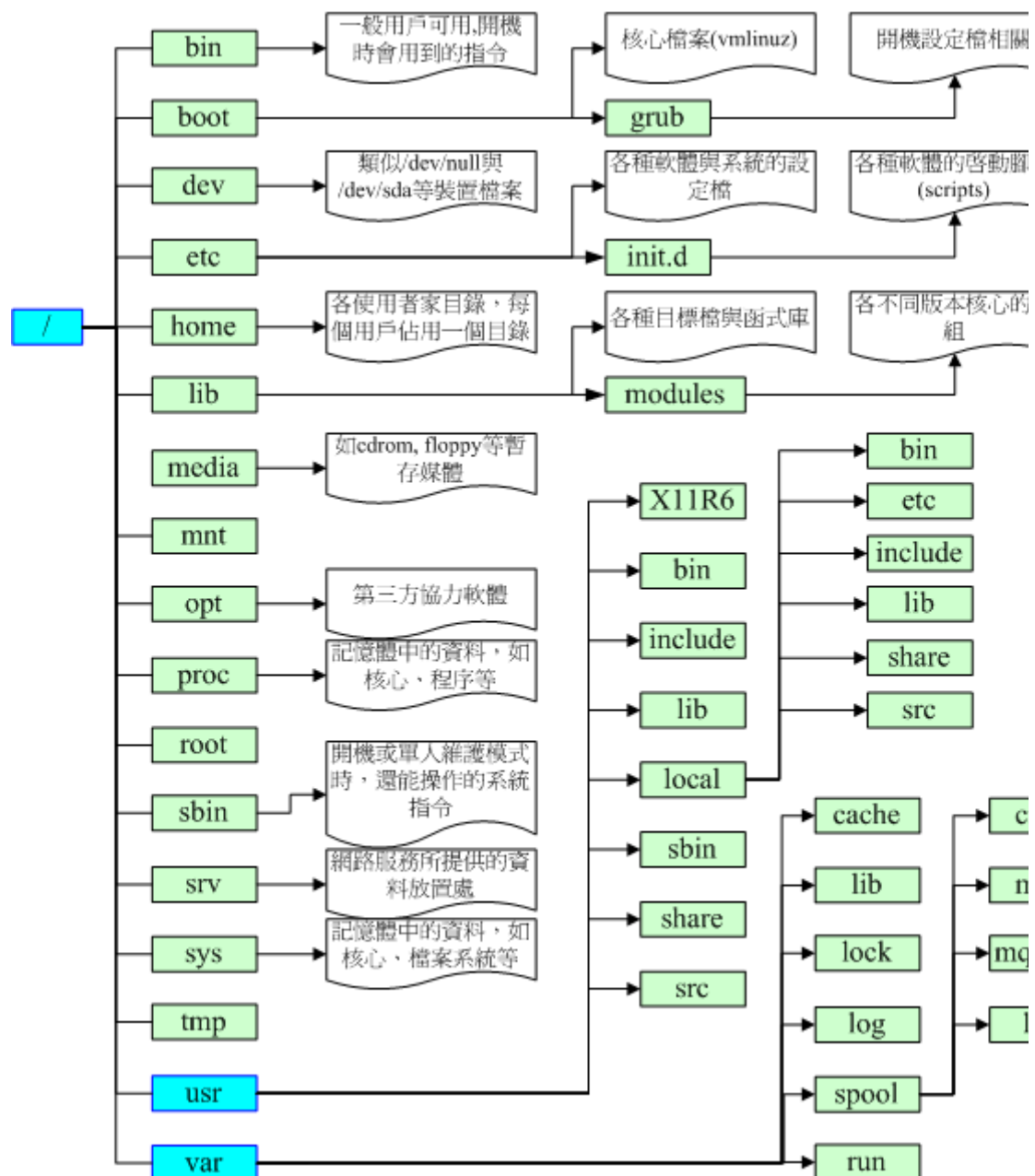


图 3.2.1、目录树架构示意图

鸟哥只有就各目录进行简单的解释，看看就好，详细的解释请回到刚刚说明的表格中去查阅喔！看完了 FHS 标准之后，现在回到[第三章里面去看看安装前 Linux 规划的分割情况](#)，对于当初为何需要分割为这样的情况，有点想法了吗？^^。根据 FHS 的定义，妳最好能够将/var 独立出来，这样对于系统的数据还有一些安全性的保护呢！因为至少/var 死掉时，你的根目录还会活着嘛！还能够进入救援模式啊！

💡绝对路径与相对路径

除了需要特别注意的 FHS 目录配置外，在文件名部分我们也要特别注意喔！因为根据档名写法的不同，也可将所谓的路径(path)定义为绝对路径(absolute)与相对路径(relative)。这两种文件名/路径的写法依据是这样的：

- 绝对路径：由根目录(/)开始写起的文件名或目录名称，例如
/home/dmtsai/.bashrc;
- 相对路径：相对于目前路径的文件名写法。例如 ./home/dmtsai
或 ../../home/dmtsai/ 等等。反正开头不是 / 就属于相对路径的写法

而你必须要了解，相对路径是以『你当前所在路径的相对位置』来表示的。举例来说，你目前在 /home 这个目录下，如果想要进入 /var/log 这个目录时，可以怎么写呢？

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

因为你在 /home 底下，所以要回到上一层 (../) 之后，才能继续往 /var 来移动的！特别注意这两个特殊的目录：

- . ：代表当前的目录，也可以使用 ./ 来表示；
- .. ：代表上一层目录，也可以 ../ 来代表。

这个 . 与 .. 目录概念是很重要的，你常常会看到 cd .. 或 ./command 之类的指令下达方式，就是代表上一层与目前所在目录的工作状态喔！很重要的呐！

例题：

如何先进入/var/spool/mail/目录，再进入到/var/spool/cron/目录内？

答：

由于/var/spool/mail 与/var/spool/cron 是同样在/var/spool/目录中，因此最简单的指令下达方法为：

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

如此就不需要在由根目录开始写起了。这个相对路径是非常有帮助的！尤其对于某些软件开发商来说。一般来说，软件开发商会将数据放置到/usr/local/里面的各相对目录，妳可以参考图 3.2.1 的相对位置。

但如果用户想要安装到不同目录呢？就得要使用相对路径啰！^_^

例题：

网络文件常常提到类似『./run.sh』之类的数据，这个指令的意义为何？

答：

由于指令的执行需要变量(bash 章节才会提到)的支持，若你的执行文件放置在本目录，并且本目录并非正规的执行文件目录(/bin, /usr/bin 等为正规)，此时要执行指令就得要严格指定该执行档。『./』代表『本目录』的意思，所以『./run.sh』代表『执行本目录下，名为 run.sh 的档案』啰！



重点回顾

- Linux 的每个档案中，依据权限分为使用者、群组与其他人三种身份；
- 群组最有用的功能之一，就是当你在团队开发资源的时候，且每个账号都可以有多个群组的支持；
- 利用 `ls -l` 显示的文件属性中，第一个字段是档案的权限，共有十个位，第一个位是文件类型，接下来三个为一组共三组，为使用者、群组、其他人的权限，权限有 `r, w, x` 三种；
- 如果档名之前多一个『.』，则代表这个档案为『隐藏档』；
- 更改档案的群组支持可用 `chgrp`，修改档案的拥有者可用 `chown`，修改档案的权限可用 `chmod`
- `chmod` 修改权限的方法有两种，分别是符号法与数字法，数字法中 `r, w, x` 分数为 4, 2, 1；
- 对档案来讲，权限的效能为：
 - `r`：可读取此一档案的实际内容，如读取文本文件的文字内容等；
 - `w`：可以编辑、新增或者是修改该档案的内容(但不含删除该档案)；
 - `x`：该档案具有可以被系统执行的权限。
- 对目录来说，权限的效能为：
 - `r` (read contents in directory)
 - `w` (modify contents of directory)
 - `x` (access directory)
- 要开放目录给任何人浏览时，应该至少也要给予 `r` 及 `x` 的权限，但 `w` 权限不可随便给；
- Linux 档名的限制为：单一档案或目录的最大容许文件名为 255 个字符；包含完整路径名称及目录 (/) 之完整档名为 4096 个字符
- 根据 FHS 的官方文件指出，他们的主要目的是希望让使用者可以了解到已安装软件通常放置于那个目录下
- FHS 订定出来的四种目录特色为：shareable, unshareable, static, variable 等四类；

- FHS 所定义的三层主目录为：/，/var，/usr 三层而已；
- 有五个目录不可与根目录放在不同的 partition，分别为/etc，/bin，/lib，/dev，/sbin 五个。



本章练习

(要看答案请将鼠标移动到『答：』底下的空白处，按下左键圈选空白处即可察看)

- 请说明/bin 与/usr/bin 目录所放置的执行文件有何不同之处？

/bin 主要放置在开机时，以及进入单人维护模式后还能够被使用的指令，至于/usr/bin 则是大部分软件提供的指令放置处。

- 请说明/bin 与/sbin 目录所放置的执行文件有何不同之处？

/bin 放置的是一般用户惯用的指令，至于/sbin 则是系统管理员才会使用到的指令。不过/bin 与/sbin 都与开机、单人维护模式有关。更多的执行档会被放置到/usr/bin 及/usr/sbin 底下。

- 哪几个目录不能够与根目录(/)放置到不同的 partition 中？并请说明该目录所放置的数据为何？

/etc(配置文件)，/bin(一般身份可用执行文件)，/dev(装置档案)，/lib(执行档的函式库或核心模块等)，/sbin(系统管理员可用指令)

- 试说明为何根目录要小一点比较好？另外在分割时，为什么/home，/usr，/var，/tmp 最好与根目录放到不同的分割槽？试说明可能的原因为何(由目录放置数据的内容谈起)？

根据 FHS 的说明，越小的/可以放置的较为集中且读取频率较不频繁，可避免较多的错误。至于/home(用户家目录)，/usr(软件资源)，/var(变动幅度较大的数据)，/tmp(系统暂存，数据莫名)中，因为数据量较大或者是读取频率较高，或者是不明的使用情况较多，因此建议不要与根目录放在一起，也会有助于系统安全。

- 早期的 Unix 系统文件名最多允许 14 个字符，而新的 Unix 与 Linux 系统中，文件名最多可以容许几个字符？

由于使用 Ext2/Ext3 文件系统，单一档名可达 255 字符，完整文件名（包含路径）可达 4096 个字符

- 当一个一般档案权限为 -rwxrwxrwx 则表示这个档案的意义为？

任何人皆可读取、修改或编辑、可以执行，但不一定能删除。

- 我需要将一个档案的权限改为 `-rwxr-xr--` 请问该如何下达指令？

`chmod 754 filename` 或 `chmod u=rwx,g=rx,o=r filename`

- 若我需要更改一个档案的拥有者与群组，该用什么指令？

`chown, chgrp`

- Linux 传统的文件系统为何？此外，常用的 Journaling 文件格式有哪些？

传统文件格式为：ext2,
Journaling 有 ext3 及 Reiserfs 等

- 请问底下的目录与主要放置什么数据：
/etc/, /etc/init.d, /boot, /usr/bin, /bin, /usr/sbin, /sbin, /dev, /var/log
 - /etc/：几乎系统的所有配置文件案均在此，尤其 `passwd, shadow`
 - /etc/init.d：系统开机的时候加载服务的 `scripts` 的摆放地点
 - /boot：开机配置文件，也是预设摆放核心 `vmlinuz` 的地方
 - /usr/bin, /bin：一般执行档摆放的地方
 - /usr/sbin, /sbin：系统管理员常用指令集
 - /dev：摆放所有系统装置档案的目录
 - /var/log：摆放系统注册表档案的地方
- 若一个档案的档名开头为『.』，例如 `.bashrc` 这个档案，代表什么？另外，如何显示出这个文件名与他的相关属性？

有『.』为开头的为隐藏档，需要使用 `ls -a` 这个 `-a` 的选项才能显示出隐藏档案的内容，而使用 `ls -al` 才能显示出属性。



参考数据与延伸阅读

- FHS 的标准官方文件：<http://proton.pathname.com/fhs/>，非常值得参考的文献！
- 关于 Journaling 日志式文章的相关说明
<http://www.linuxplanet.com/linuxplanet/reports/3726/1/>

2002/07/18：第一次完成

2003/02/06：重新编排与加入 FAQ

2005/06/28：将旧的数据移动到 [这里](#)

2005/07/15：呼呼～终于改完成了～这次的修订当中，加入了 FHS 的说明，希望大家能够比较清楚 Linux 的目录配置！

2005/08/05: 修订了最大档名字元, 应该是 255 才对! 另外, 加入了『檔名限制』的部分!

2005/09/03: 修订了目录权限相关的说明, 将原本仅具有 r 却写成无法使用 ls 浏览的说明数据移除!

2008/09/08: 旧的针对 FC4 所写的文章移动到[此处](#)

2008/09/20: 针对 FHS 加强说明了一下, 分为/, /usr, /var 三层来个别说明! 并非抄袭官网的数据而已喔!

2008/09/23: 经过一场大感冒, 停工了四、五天, 终于还是给他完工了! ^_^

2008/10/21: 原本的第四小节 Linux 的文件系统, 因为与第八章重复性太高, 将他移除了!

2002/02/18 以来统计人数

828455



本网页主要以 [firefox](#) 配合分辨率 1024x768 作为设计依据

<http://linux.vbird.org> is designed by [VBird](#) during 2001-2009. [Aerosol Lab](#).