

## 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



## 第九章、档案与文件系统的压缩与打包

最近更新日期: 2008/12/18

在 Linux 底下有相当多的压缩指令可以运作喔! 这些压缩指令可以让我们更方便从网络上下载大型的档案呢! 此外, 我们知道在 Linux 底下的扩展名是没有什么很特殊的意义的, 不过, 针对这些压缩指令所做出来的压缩文件, 为了方便记忆, 还是会有一些特殊的命名方式啦! 就让我们来看看吧!

1. 压缩文件案的用途与技术
2. Linux 系统常见的压缩指令
  - 2.1 `compress`
  - 2.2 `gzip`, `zcat`
  - 2.3 `bzip2`, `bzcat`
3. 打包指令: `tar`
4. 完整备份工具: `dump`, `restore`
5. 光盘写入工具
  - 5.1 `mkisofs`: 建立映像档
  - 5.2 `cdrecord`: 光盘刻录工具
6. 其他常见的压缩与备份工具
  - 6.1 `dd`
  - 6.2 `cpio`
7. 重点回顾
8. 本章习题
9. 参考数据与延伸阅读
10. 针对本文的建议: <http://phorum.vbird.org/viewtopic.php?t=23882>



### 压缩文件案的用途与技术

你是否有过文件档案太大, 导致无法以一片软盘将他复制完成的困扰? 又, 你是否有过, 发现一个软件里面有好多档案, 这些档案要将他复制与携带都很不方便的问题? 还有, 你是否有过要备份某些重要数据, 偏偏这些数据量太大了, 耗掉了你很多的硬盘与磁盘空间呢? 这个时候, 那个好用的『文件压缩』技术可就派的上用场了!

因为这些比较大型的档案透过所谓的文件压缩技术之后, 可以将他的磁盘使用量降低, 可以达到减低档案容量的效果, 此外, 有的压缩程序还可以进行容量限制, 使一个大型档案可以分割成为数个小型档案, 以方便软盘片携带呢!

那么什么是『文件压缩』呢？我们来稍微谈一谈他的原理好了。目前我们使用的计算机系统中都是使用所谓的 bytes 单位来计量的！不过，事实上，计算机最小的计量单位应该是 bits 才对啊，此外，我们也知道  $1 \text{ byte} = 8 \text{ bits}$ 。但是如果今天我们只是记忆一个数字，亦即是 1 这个数字呢？他会如何记录？假设一个 byte 可以看成底下的模样：

□□□□□□□□

**Tips:**

由于  $1 \text{ byte} = 8 \text{ bits}$ ，所以每个 byte 当中会有 8 个空格，而每个空格可以是 0, 1，这里仅是做为一个约略的介绍，更多的详细资料请参考第零章的计算器概论吧！



由于我们记录数字是 1，考虑计算机所谓的二进制喔，如此一来，1 会在最右边占据 1 个 bit，而其他的 7 个 bits 将会自动的被填上 0 啰！你看看，其实在这样的例子中，那 7 个 bits 应该是『空的』才对！不过，为了要满足目前我们的操作系统数据的存取，所以就会将该数据转为 byte 的型态来记录了！而一些聪明的计算机工程师就利用一些复杂的计算方式，将这些没有使用到的空间『丢』出来，以让档案占用的空间变小！这就是压缩的技术啦！

另外一种压缩技术也很有趣，他是将重复的数据进行统计记录的。举例来说，如果你的数据为『111...』共有 100 个 1 时，那么压缩技术会记录为『100 个 1』而不是真的有 100 个 1 的位存在！这样也能够精简档案记录的容量呢！非常有趣吧！

简单的说，你可以将他想成，其实档案里面有相当多的『空间』存在，并不是完全填满的，而『压缩』的技术就是将这些『空间』填满，以让整个档案占用的容量下降！不过，这些『压缩过的档案』并无法直接被我们的操作系统所使用的，因此，若要使用这些被压缩过的档案数据，则必须将他『还原』回来未压缩前的模样，那就是所谓的『解压缩』啰！而至于压缩前与压缩后的档案所占用的磁盘空间大小，就可以被称为是『压缩比』啰！更多的技术文件或许你可以参考一下：

- RFC 1952 文件: <http://www.ietf.org/rfc/rfc1952.txt>
- 鸟哥站上的备份:  
[http://linux.vbird.org/linux\\_basic/0240tarcompress/0240tarcompress\\_gzip.php](http://linux.vbird.org/linux_basic/0240tarcompress/0240tarcompress_gzip.php)

这个『压缩』与『解压缩』的动作有什么好处呢？最大的好处就是压缩过的档案容量变小了，所以你的硬盘容量无形之中就可以容纳更多的资料。此外，在一些网络数据的传输中，也会由于数据量的降低，好让网络带宽可以用来作更多的工作！而不是老是卡在一些大型的文件传输上面呢！目前很多的 WWW 网站也是利用文件压缩的技术来进行数据的传送，好让网站带宽的可利用率上升喔！

### Tips:

上述的 WWW 网站压缩技术蛮有趣的！他让你网站上面『看的到的数据』在经过网络传输时，使用的是『压缩过的数据』，等到这些压缩过的数据到达你的计算机主机时，再进行解压缩，由于目前的计算机指令周期相当的快速，因此其实在网页浏览的时候，时间都是花在『数据的传输』上面，而不是 CPU 的运算啦！如此一来，由于压缩过的数据量降低了，自然传送的速度就会增快不少！



若你是一位软件工程师，那么相信你也会喜欢将你自己的软件压缩之后提供大家下载来使用，毕竟没有人喜欢自己的网站天天都是带宽满载的吧？举个例子来说，Linux 2.6.27.4 完整的核心大小约有 300 MB 左右，而由于核心主要多是 ASCII code 的纯文本型态档案，这种档案的『多余空间』最多了。而一个提供下载的压缩过的 2.6.27.4 核心大约仅有 60MB 左右，差了几倍呢？你可以自己算一算喔！



Linux 系统常见的压缩指令：

在 Linux 的环境中，压缩文件案的扩展名大多是：『\*.tar, \*.tar.gz, \*.tgz, \*.gz, \*.Z, \*.bz2』，为什么会有这样的扩展名呢？不是说 Linux 的扩展名没有什么作用吗？

这是因为 Linux 支持的压缩指令非常多，且不同的指令所用的压缩技术并不相同，当然彼此之间可能就无法互通压缩/解压缩文件案啰。所以，当你下载到某个压缩文件时，自然就需要知道该档案是由哪种压缩指令所制作出来的，好用来对照着解压缩啊！也就是说，虽然 Linux 档案的属性基本上是与文件名没有绝对关系的，但是为了帮助我们人类小小的脑袋瓜子，所以适当的扩展名还是必要的！底下我们就列出几个常见的压缩文件案扩展名吧：

*.Z	compress 程序压缩的档案；
*.gz	gzip 程序压缩的档案；
*.bz2	bzip2 程序压缩的档案；
*.tar	tar 程序打包的数据，并没有压缩过；
*.tar.gz	tar 程序打包的档案，其中并且经过 gzip 的压缩
*.tar.bz2	tar 程序打包的档案，其中并且经过 bzip2 的压缩

Linux 上常见的压缩指令就是 gzip 与 bzip2，至于 compress 已经退流行了。gzip 是由 GNU 计划所开发出来的压缩指令，该指令已经取代了 compress。后

来 GNU 又开发出 bzip2 这个压缩比更好的压缩指令！不过，这些指令通常仅能针对一个档案来压缩与解压缩，如此一来，每次压缩与解压缩都要一大堆档案，岂不烦人？此时，那个所谓的『打包软件，tar』就显的很重要啦！

这个 tar 可以将很多档案『打包』成为一个档案！甚至是目录也可以这么玩。不过，单纯的 tar 功能仅是『打包』而已，亦即是将很多档案集结成为一个档案，事实上，他并没有提供压缩的功能，后来，[GNU 计划](#)中，将整个 tar 与压缩的功能结合在一起，如此一来提供使用者更方便并且更强大的压缩与打包功能！底下我们就来谈一谈这些在 Linux 底下基本的压缩指令吧！

---

## compress

compress 这个压缩指令是非常老旧的一款，大概只有在非常旧的 Unix 机器上面还会找到这个软件。我们的 CentOS 预设并没有安装这个软件到系统当中，所以想要了解这个软件的使用时，请先安装 ncompress 这个软件套件。不过，由于 gzip 已经可以解开使用 compress 压缩的档案，因此，compress 可以不用学习啦！但是，如果你所在的环境还是有老旧的系统，那么还是得要学一学就是了。好了，如果你有网络的话，那么安装其实很简单喔！

```
[root@www ~]# yum install ncompress
base      100% |=====| 1.1 kB
00:00
updates   100% |=====| 951 B
00:00
addons     100% |=====| 951 B
00:00
extras    100% |=====| 1.1 kB
00:00
Setting up Install Process
Parsing package install arguments
Resolving Dependencies          <==开始分析相依性
--> Running transaction check
--> Package ncompress.i386 0:4.2.4-47 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package      Arch   Version      Repository    Size
=====
Installing:
ncompress    i386   4.2.4-47     base          23 k
```

```

Transaction Summary
=====
Install      1 Package(s)  <==最后分析所要安装的软件数
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 23 k
Is this ok [y/N]: y    <==这里请按下 y 来确认安装
Downloading Packages:
(1/1): ncompress-4.2.4-47 100%
|=====| 23 kB    00:00
warning: rpmts_HdrFromFdno: Header V3 DSA signature:
NOKEY, key ID e8562897
Importing GPG key 0xE8562897 "CentOS-5 Key (CentOS 5
Official Signing Key)
<centos-5-key@centos.org>" from
http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
Is this ok [y/N]: y    <==这里则是与数字签名有关
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: ncompress
##### [1/1]

Installed: ncompress.i386 0:4.2.4-47
Complete!

```

关于 yum 更详细的用法我们会在后续的章节介绍，这里仅是提供一个大概的用法而已。等你安装好这个软件后，接下来让我们看看如何使用 compress 吧！

```

[root@www ~]# compress [-rcv] 档案或目录  <==这里是
压缩
[root@www ~]# uncompress 档案.Z           <==这里是
解压缩
选项与参数:
-r  : 可以连同目录下的档案也同时给予压缩呢！
-c  : 将压缩数据输出成为 standard output (输出到屏幕)
-v  : 可以秀出压缩后的档案信息以及压缩过程中的一些档
名变化。

```

```
范例一：将 /etc/man.config 复制到 /tmp，并加以压缩
[root@www ~]# cd /tmp
[root@www tmp]# cp /etc/man.config .
[root@www tmp]# compress -v man.config
man.config:  -- replaced with man.config.Z
Compression: 41.86%
[root@www tmp]# ls -l /etc/man.config /tmp/man*
-rw-r--r-- 1 root root 4617 Jan  6  2007
/etc/man.config  <==原有档案
-rw-r--r-- 1 root root 2684 Nov 10 17:14
/tmp/man.config.Z <==经过压缩的档案！
```

不知道你有没有发现，复制到 /tmp 的 man.config 不见了！因为被压缩成为 man.config.Z 啰 也就是说，在预设的情况下，被 compress 压缩的源文件会不见，而压缩文件案会被建立起来，而且扩展名会是 \*.Z。仔细看一下，档案由原本的 4617bytes 降低到 2684bytes 左右，确实有减少一点啦！那么如何解压缩呢？

```
范例二：将刚刚的压缩文件解开
[root@www tmp]# uncompress man.config.Z
[root@www tmp]# ll man*
-rw-r--r-- 1 root root 4617 Nov 10 17:14 man.config
```

解压缩直接用 uncompress 即可！解压缩完毕后该档案就自动的变回来了！不过，那个压缩文件却又不存在啰～ 这样可以理解用法了吗？那如果我想要保留源文件且又要建立压缩文件呢？可以使用 -c 的语法！

```
范例三：将 man.config 压缩成另外一个档案来备份
[root@www tmp]# compress -c man.config >
man.config.back.Z
[root@www tmp]# ll man*
-rw-r--r-- 1 root root 4617 Nov 10 17:14 man.config
-rw-r--r-- 1 root root 2684 Nov 10 17:24
man.config.back.Z
# 这个 -c 的选项比较有趣！他会将压缩过程的数据输出到
屏幕上，而不是写入成为
# *.Z 的压缩文件。所以，我们可以透过数据流重导向的方法
将数据输出成为另一个档名。
# 关于数据流重导向，我们会在 bash shell 当中详细谈论
的啦！
```

再次强调，compress 已经很少人在使用了，因为这支程序无法解开 \*.gz 的档

案，而 gzip 则可以解开 \*.Z 的档案，所以，如果你的 distribution 上面没有 compress 的话，那就不要进行上面的练习啰！^\_^

---

## gzip, zcat

gzip 可以说是应用度最广的压缩指令了！目前 gzip 可以解开 compress, zip 与 gzip 等软件所压缩的档案。至于 gzip 所建立的压缩文件为 \*.gz 的檔名喔！让我们来看看这个指令的语法吧：

```
[root@www ~]# gzip [-cdtv#] 檔名
[root@www ~]# zcat 檔名.gz
选项与参数：
-c  : 将压缩的数据输出到屏幕上，可透过数据流重导向来处理；
-d  : 解压缩的参数；
-t  : 可以用来检验一个压缩文件的一致性～看看档案有无错误；
-v  : 可以显示出原档案/压缩文件案的压缩比等信息；
-#  : 压缩等级，-1 最快，但是压缩比最差、-9 最慢，但是压缩比最好！预设是 -6

范例一：将 /etc/man.config 复制到 /tmp，并且以 gzip 压缩
[root@www ~]# cd /tmp
[root@www tmp]# cp /etc/man.config .
[root@www tmp]# gzip -v man.config
man.config:      56.1% -- replaced with man.config.gz
[root@www tmp]# ll /etc/man.config /tmp/man*
-rw-r--r-- 1 root root 4617 Jan  6  2007
/etc/man.config
-rw-r--r-- 1 root root 2684 Nov 10 17:24
/tmp/man.config.back.Z
-rw-r--r-- 1 root root 2057 Nov 10 17:14
/tmp/man.config.gz  <==gzip 压缩比较佳
```

与 compress 类似的，当你使用 gzip 进行压缩时，在预设的状态下原本的档案会被压缩成为 .gz 的檔名，源文件就不再存在了。您也可以发现，由于 gzip 的压缩比要比 compress 好的多，所以当然建议使用 gzip 啦！此外，使用 gzip 压缩的档案在 Windows 系统中，竟然可以被 WinRAR 这个软件解压缩呢！很好用吧！至于其他的用法如下：



范例二：由于 man.config 是文本文件，请将范例一的压缩文件的内容读出来！

```
[root@www tmp]# zcat man.config.gz
# 由于 man.config 这个原本的档案是文本文件，因此我们可以尝试使用 zcat 去读取！
# 此时屏幕上会显示 man.config.gz 解压缩之后的档案内容！
```

范例三：将范例一的档案解压缩

```
[root@www tmp]# gzip -d man.config.gz
# 不要使用 gunzip 这个指令，不好背！使用 gzip -d 来进行解压缩！
# 与 gzip 相反， gzip -d 会将原本的 .gz 删除，产生原本的 man.config 档案。
```

范例四：将范例三解开的 man.config 用最佳的压缩比压缩，并保留原本的档案

```
[root@www tmp]# gzip -9 -c man.config > man.config.gz
```

其实 gzip 的压缩已经优化过了，所以虽然 gzip 提供 1~9 的压缩等级，不过使用默认的 6 就非常好用了！因此上述的范例四可以不要加入那个 -9 的选项。范例四的重点在那个 -c 与 > 的使用啰！

cat 可以读取纯文本档，那个 zcat 则可以读取纯文本档被压缩后的压缩文件！由于 gzip 这个压缩指令主要想要用来取代 compress 的，所以不但 compress 的压缩文件案可以使用 gzip 来解开，同时 zcat 这个指令可以同时读取 compress 与 gzip 的压缩文件哟！

---

## bzip2, bzipcat

若说 gzip 是为了取代 compress 并提供更好的压缩比而成立的，那么 bzip2 则是为了取代 gzip 并提供更佳的压缩比而来的。bzip2 真是很不错用的东西～这玩意的压缩比竟然比 gzip 还要好～至于 bzip2 的用法几乎与 gzip 相同！看看底下的用法吧！

```
[root@www ~]# bzip2 [-cdkzv#] 檔名
[root@www ~]# bzipcat 檔名.bz2
选项与参数：
-c  : 将压缩的过程产生的数据输出到屏幕上！
-d  : 解压缩的参数
-k  : 保留源文件，而不会删除原始的档案喔！
```



```
-z : 压缩的参数  
-v : 可以显示出原档案/压缩文件案的压缩比等信息;  
-# : 与 gzip 同样的, 都是在计算压缩比的参数, -9 最佳, -1 最快!
```

范例一: 将刚刚的 /tmp/man.config 以 bzip2 压缩

```
[root@www tmp]# bzip2 -z man.config  
# 此时 man.config 会变成 man.config.bz2 !
```

范例二: 将范例一的档案内容读出来!

```
[root@www tmp]# bzcat man.config.bz2  
# 此时屏幕上会显示 man.config.bz2 解压缩之后的档案内容!!
```

范例三: 将范例一的档案解压缩

```
[root@www tmp]# bzip2 -d man.config.bz2
```

范例四: 将范例三解开的 man.config 用最佳的压缩比压缩, 并保留原本的档案

```
[root@www tmp]# bzip2 -9 -c man.config > man.config.bz2
```

使用 compress 扩展名自动建立为 .Z , 使用 gzip 扩展名自动建立为 .gz 。这里的 bzip2 则是自动的将扩展名建置为 .bz2 啰! 所以当我们使用具有压缩功能的 bzip2 -z 时, 那么刚刚的 man.config 就会自动的变成了 man.config.bz2 这个档名啰!

好了, 那么如果我想要读取这个档案的内容呢? 是否一定要解开? 当然不需要啰! 可以使用简便的 bzcat 这个指令来读取内容即可! 例如上面的例子中, 我们可以使用 bzcat man.config.bz2 来读取数据而不需要解开! 此外, 当你要解开一个压缩文件时, 这个档案的名称为 .bz, .bz2, .tbz, .tbz2 等等, 那么就可以尝试使用 bzip2 来解看看啦! 当然啰, 也可以使用 bunzip2 这个指令来取代 bzip2 -d 啰。



打包指令: tar

前一小节谈到的指令大多仅能针对单一档案来进行压缩, 虽然 gzip 与 bzip2 也能够针对目录来进行压缩, 不过, 这两个指令对目录的压缩指的是『将目录内的所有档案“分别”进行压缩』的动作! 而不像在 Windows 的系统, 可以使用类似 WinRAR 这一类的压缩软件来将好多数据『包成一个档案』的样式。

这种将多个档案或目录包成一个大档案的指令功能, 我们可以称呼他是一种『打包指令』啦! 那 Linux 有没有这种打包指令呢? 是有的! 那就是鼎鼎大名的 tar 这个玩意儿了! tar 可以将多个目录或档案打包成一个大档案, 同时还可以透

过 gzip/bzip2 的支持，将该档案同时进行压缩！更有趣的是，由于 tar 的使用太广泛了，目前 Windows 的 WinRAR 也支持 .tar.gz 档名的解压缩呢！很不错吧！所以底下我们就来玩一玩这个咚咚！

---

## tar

tar 的选项与参数非常的多！我们只讲几个常用的选项，更多选项您可以自行 man tar 查询啰！

```
[root@www ~]# tar [-j|-z] [cv] [-f 建立的檔名]
filename... <==打包与压缩
[root@www ~]# tar [-j|-z] [tv] [-f 建立的檔名]
<==察看檔名
[root@www ~]# tar [-j|-z] [xv] [-f 建立的檔名] [-C 目录]
<==解压缩
```

选项与参数：

- c : 建立打包档案，可搭配 -v 来察看过程中被打包的档名(filename)
- t : 察看打包档案的内容含有哪些档名，重点在察看『档名』就是了；
- x : 解打包或解压缩的功能，可以搭配 -C (大写) 在特定目录解开

特别留意的是，-c, -t, -x 不可同时出现在一串指令列中。

- j : 透过 bzip2 的支持进行压缩/解压缩：此时档名最好为 \*.tar.bz2
- z : 透过 gzip 的支持进行压缩/解压缩：此时档名最好为 \*.tar.gz
- v : 在压缩/解压缩的过程中，将正在处理的文件名显示出来！

-f filename: -f 后面要立刻接要被处理的档名！建议 -f 单独写一个选项啰！

-C 目录 : 这个选项用在解压缩，若要在特定目录解压缩，可以使用这个选项。

其他后续练习会使用到的选项介绍：

- p : 保留备份数据的原本权限与属性，常用于备份(-c)重要的配置文件
- P : 保留绝对路径，亦即允许备份数据中含有根目录存在之意；
- exclude=FILE: 在压缩的过程中，不要将 FILE 打包！

其实最简单的使用 tar 就只要记忆底下的方式即可：

- 压 缩：tar -jcv -f filename.tar.bz2 要被压缩的档案或目录名称
- 查 询：tar -jtv -f filename.tar.bz2
- 解压缩：tar -jxv -f filename.tar.bz2 -C 欲解压缩的目录

那个 filename.tar.bz2 是我们自己取的档名，tar 并不会主动的产生建立的档名喔！我们要自定义啦！所以扩展名就显的很重要了！如果不加 [-j|-z] 的话，档名最好取为 \*.tar 即可。如果是 -j 选项，代表有 bzip2 的支持，因此档名最好就取为 \*.tar.bz2，因为 bzip2 会产生 .bz2 的扩展名之故！至于如果是加上了 -z 的 gzip 的支持，那档名最好取为 \*.tar.gz 喔！了解乎？

另外，由于『-f filename』是紧接在一起的，过去很多文章常会写成『-jcvf filename』，这样是对的，但由于选项的顺序理论上是可以变换的，所以很多读者会误认为『-jvfc filename』也可以～事实上这样会导致产生的档名变成 c！因为 -fc 嘛！所以啰，建议您在学习 tar 时，将『-f filename』与其他选项独立出来，会比较不容易发生问题。

闲话少说，让我们来测试几个常用的 tar 方法吧！

- 
- 使用 tar 加入 -j 或 -z 的参数备份 /etc/ 目录

有事没事备份一下 /etc 这个目录是件好事！备份 /etc 最简单的方法就是使用 tar 啰！让我们来玩玩先：

```
[root@www ~]# tar -zpcv -f /root/etc.tar.gz /etc
tar: Removing leading `/' from member names <==注意
这个警告讯息
/etc/
.... 中间省略....
/etc/esd.conf
/etc/crontab
# 由于加上 -v 这个选项，因此正在作用中的文件名就会显示在屏幕上。
# 如果你可以翻到第一页，会发现出现上面的错误讯息！底下会讲解。
# 至于 -p 的选项，重点在于『保留原本档案的权限与属性』之意。
```

```
[root@www ~]# tar -jpcv -f /root/etc.tar.bz2 /etc
# 显示的讯息会跟上面一模一样啰！

[root@www ~]# ll /root/etc*
-rw-r--r-- 1 root root 8740252 Nov 15 23:07
/root/etc.tar.bz2
-rw-r--r-- 1 root root 13010999 Nov 15 23:01
/root/etc.tar.gz
[root@www ~]# du -sm /etc
118      /etc
# 为什么建议您使用 -j 这个选项？从上面的数值你可以
知道了吧？ ^_^
```

由上述的练习，我们知道使用 bzip2 亦即 -j 这个选项来制作备份时，能够得到比较好的压缩比！如上表所示，由原本的 /etc/ (118MBytes) 下降到 8.7Mbytes 左右！至于加上『-p』这个选项的原因是为了保存原本档案的权限与属性！我们曾在[第七章的 cp 指令介绍](#)时谈到权限与文件类型(例如连结档)对复制的不同影响。同样的，在备份重要的系统数据时，这些原本档案的权限需要做完整的备份比较好。此时 -p 这个选项就派的上用场了。接下来让我们看看打包档案内有什么数据存在？

- 
- 查阅 tar 档案的数据内容(可察看文件名)，与备份文件名有否根目录的意义

要察看档名非常的简单！可以这样做：

```
[root@www ~]# tar -jtv -f /root/etc.tar.bz2
.... 前面省略....
-rw-r--r-- root/root 1016 2008-05-25 14:06:20
etc/dbus-1/session.conf
-rw-r--r-- root/root 153 2007-01-07 19:20:54
etc/esd.conf
-rw-r--r-- root/root 255 2007-01-06 21:13:33
etc/crontab
```

如果加上 -v 这个选项时，详细的档案权限/属性都会被列出来！如果只是想要知道档名而已，那么就将 -v 拿掉即可。从上面的数据我们可以发现一件很有趣的事情，那就是每个文件名都没了根目录了！这也是上一个练习中出现的那个警告讯息『tar: Removing leading `/' from member names(移除了档名开头的`/')』所告知的情况！

那为什么要拿掉根目录呢？主要是为了安全！我们使用 tar 备份的数据可能会需要解压缩回来使用，在 tar 所记录的文件名（就是我们刚刚使用 tar -jtvf 所察看到的档名）那就是解压缩后的实际档名。如果拿掉了根目录，假设你将备份数据在 /tmp 解开，那么解压缩的档名就会变成『/tmp/etc/xxx』。但『如果没有拿掉根目录，解压缩后的档名就会是绝对路径，亦即解压缩后的数据一定会被放置到 /etc/xxx 去！』如此一来，你的原本的 /etc/ 底下的数据，就会被备份数据所覆盖过去了！

#### Tips:

你会说：『既然是备份数据，那么还原回来也没有什么问题吧？』想象一个状况，你备份的资料是一年前的旧版 CentOS 4.x，你只是想要了解一下过去的备份内容究竟有哪些数据而已，结果一解开该档案，却发现你目前新版的 CentOS 5.x 底下的 /etc 被旧版的备份数据覆盖了！此时你该如何是好？所以啰，当然是拿掉根目录比较安全一些的。



如果你确定你就是需要备份根目录到 tar 的档案中，那可以使用 -P (大写) 这个选项，请看底下的例子分析：

```
范例：将文件名中的(根)目录也备份下来，并察看一下备份档的内容档名
[root@www ~]# tar -jpPcv -f /root/etc.and.root.tar.bz2 /etc
.... 中间过程省略....
[root@www ~]# tar -jtf /root/etc.and.root.tar.bz2
/etc/dbus-1/session.conf
/etc/esd.conf
/etc/crontab
# 这次查阅文件名不含 -v 选项，所以仅有文件名而已！没有详细属性/权限等参数。
```

有发现不同点了吧？如果加上 -P 选项，那么文件名内的根目录就会存在喔！不过，鸟哥个人建议，还是不要加上 -P 这个选项来备份！毕竟很多时候，我们备份是为了要未来追踪问题用的，倒不一定需要还原回原本的系统中！所以拿掉根目录后，备份数据的应用会比较有弹性！也比较安全呢！

- 
- 将备份的数据解压缩，并考虑特定目录的解压缩动作（-C 选项的应用）

那如果想要解打包呢？很简单的动作就是直接进行解打包嘛！

```
[root@www ~]# tar -jxv -f /root/etc.tar.bz2
[root@www ~]# ll
.... (前面省略)....
drwxr-xr-x 105 root root    12288 Nov 11 04:02 etc
.... (后面省略)....
```

此时该打包档案会在『本目录下进行解压缩』的动作！所以，你等一下就会在家目录底下发现一个名为 `etc` 的目录啰！所以啰，如果你想要将该档案在 `/tmp` 底下解开，可以 `cd /tmp` 后，再下达上述的指令即可。不过，这样好像很麻烦呢～有没有更简单的方法可以『指定欲解开的目录』呢？有的，可以使用 `-C` 这个选项喔！举例来说：

```
[root@www ~]# tar -jxv -f /root/etc.tar.bz2 -C /tmp
[root@www ~]# ll /tmp
.... (前面省略)....
drwxr-xr-x 105 root root    12288 Nov 11 04:02 etc
.... (后面省略)....
```

这样一来，你就能够将该档案在不同的目录解开啰！鸟哥个人是认为，这个 `-C` 的选项务必要记忆一下的！好了，处理完毕后，请记得将这两个目录删除一下呢！

```
[root@www ~]# rm -rf /root/etc /tmp/etc
```

再次强调，这个『`rm -rf`』是很危险的指令！下达时请务必确认一下后面接的档名。我们要删除的是 `/root/etc` 与 `/tmp/etc`，您可不要将 `/etc/` 删除掉了！系统会死掉的～ ^\_^

- 
- 仅解开单一档案的方法

刚刚上头我们解压缩都是将整个打包档案的内容全部解开！想象一个情况，如果我只想要解开打包档案内的其中一个档案而已，那该如何做呢？很简单的，你只要使用 `-jtv` 找到你要的档名，然后将该档名解开即可。我们用底下的例子来说明一下：

```
# 1. 先找到我们要的档名，假设解开 shadow 档案好了：
[root@www ~]# tar -jtv -f /root/etc.tar.bz2 | grep
'shadow'
-r----- root/root  1230 2008-09-29 02:21:20
etc/shadow-
-r----- root/root   622 2008-09-29 02:21:20
```

```

etc/gshadow-
-r----- root/root    636 2008-09-29 02:21:25
etc/gshadow
-r----- root/root    1257 2008-09-29 02:21:25
etc/shadow <==这是我们要的！
# 先搜寻重要的档名！其中那个 grep 是『撷取』关键词的功能！我们会在第三篇说明！
# 这里您先有个概念即可！那个管线 | 配合 grep 可以撷取关键词的意思！

# 2. 将该档案解开！语法与实际作法如下：
[root@www ~]# tar -jxv -f 打包档.tar.bz2 待解开档名
[root@www ~]# tar -jxv -f /root/etc.tar.bz2 etc/shadow
etc/shadow
[root@www ~]# ll etc
total 8
-r----- 1 root root 1257 Sep 29 02:21 shadow <==
哟喝！只有一个档案啦！
# 很有趣！此时只会解开一个档案而已！不过，重点是那个档名！你要找到正确的档名。
# 在本例中，你不能写成 /etc/shadow ！因为记录在 etc.tar.bz2 内的档名之故！

```

- 打包某目录，但不含该目录下的某些档案之作法

假设我们想要打包 /etc/ /root 这几个重要的目录，但却不想要打包 /root/etc\* 开头的档案，因为该档案都是刚刚我们才建立的备份档嘛！而且假设这个新的打包档案要放置成为 /root/system.tar.bz2 ，当然这个档案自己不要打包自己（因为这个档案放置在 /root 底下啊！），此时我们可以透过 --exclude 的帮忙！那个 exclude 就是不包含的意思！所以你可以这样做：

```

[root@www ~]# tar -jcv -f /root/system.tar.bz2
--exclude=/root/etc* \
> --exclude=/root/system.tar.bz2 /etc /root

```

上面的指令是一整列的～其实你可以打成：『tar -jcv -f /root/system.tar.bz2 --exclude=/root/etc\* --exclude=/root/system.tar.bz2 /etc /root』，如果想要两行输入时，最后面加上反斜杠 (\) 并立刻按下 [enter] ，就能够到第二行继续输入了。这个指令下达的方式我们会在第三章再详细说明。透过这个 --exclude="file" 的动作，我们可以将几个特殊的档案或目录移除在打包之列，让打包的动作变的更简便喔！^\_^



另外，在新版的 tar 指令中，鸟哥发现原本的『 --exclude file 』似乎无法实际运作了！使用 man tar 明明有看到这个选项的说，但使用 info tar 才发现，选项功能已经变成了『 --exclude=file 』的模式！这个地方得要特别留意呢！

---

- 仅备份比某个时刻还要新的档案

某些情况下你会想要备份新的档案而已，并不想要备份旧档案！此时 --newer-mtime 这个选项就粉重要啦！其实有两个选项啦，一个是『 --newer 』另一个就是『 --newer-mtime 』，这两个选项有何不同呢？我们在 [第七章的 touch](#) 介绍中谈到过三种不同的时间参数，当使用 --newer 时，表示后续的日期包含『 mtime 与 ctime 』，而 --newer-mtime 则仅是 mtime 而已！这样知道了吧！^\_^。那就让我们来尝试处理一下啰！

```
# 1. 先由 find 找出比 /etc/passwd 还要新的档案
[root@www ~]# find /etc -newer /etc/passwd
.... (过程省略)....
# 此时会显示出比 /etc/passwd 这个档案的 mtime 还要
新的档名，
# 这个结果在每部主机都不相同！您先自行查阅自己的主机
即可，不会跟鸟哥一样！

[root@www ~]# ll /etc/passwd
-rw-r--r-- 1 root root 1945 Sep 29 02:21 /etc/passwd

# 2. 好了，那么使用 tar 来进行打包吧！日期为上面看到的
的 2008/09/29
[root@www ~]# tar -jcv -f
/root/etc.newer.then.passwd.tar.bz2 \
> --newer-mtime="2008/09/29" /etc/*
.... (中间省略)....
/etc/smartd.conf    <==真的有备份的档案
.... (中间省略)....
/etc/yum.repos.d/   <==目录都会被记录下来！
tar: /etc/yum.repos.d/CentOS-Base.repo: file is
unchanged; not dumped
# 最后行显示的是『没有被备份的』，亦即 not dumped 的
意思！

# 3. 显示出档案即可
[root@www ~]# tar -jtv -f
/root/etc.newer.then.passwd.tar.bz2 | \
```

```
> grep -v '/'$'  
# 透过这个指令可以呼叫出 tar.bz2 内的结尾非 / 的檔  
名！就是我们要的啦！
```

现在你知道这个指令的好用了吧！甚至可以进行差异档案的记录与备份呢～ 这样子的备份就会显的更容易啰！你可以这样想象，如果我在一个月前才进行过一次完整的数据备份，那么这个月想要备份时，当然可以仅备份上个月进行备份的那个时间点之后的更新的档案即可！为什么呢？因为原本的档案已经有备份了嘛！干嘛还要进行一次？只要备份新数据即可。这样可以降低备份的容量啊！

---

- 基本名称： tarfile, tarball ？

另外值得一提的是，tar 打包出来的档案有没有进行压缩所得到的档案称呼不同喔！如果仅是打包而已，就是『 tar -cv -f file.tar 』而已，这个档案我们称呼为 tarfile 。如果还有进行压缩的支持，例如『 tar -jcv -f file.tar.bz2 』时，我们就称呼为 tarball (tar 球?)！这只是一个基本的称谓而已，不过很多书籍与网络都会使用到这个 tarball 的名称！所以得要跟您介绍介绍。

此外，tar 除了可以将资料打包成为档案之外，还能够将档案打包到某些特别的装置去，举例来说， 磁带机 (tape) 就是一个常见的例子。磁带机由于是一次性读取/写入的装置，因此我们不能够使用类似 cp 等指令来复制的！那如果想要将 /home, /root, /etc 备份到磁带机 (/dev/st0) 时，就可以使用：『tar -cv -f /dev/st0 /home /root /etc』，很简单容易吧！磁带机用在备份（尤其是企业应用）是很常见的工作喔！

---

- 特殊应用：利用管线命令与数据流

在 tar 的使用中，有一种方式最特殊，那就是透过标准输入输出的数据流重导向(standard input/standard output)，以及管线命令 (pipe) 的方式，将待处理的档案一边打包一边解压缩到目标目录去。关于数据流重导向与管线命令更详细的数据我们会在 [bash](#) 再跟大家介绍，底下先来看一个例子吧！

```
# 1. 将 /etc 整个目录一边打包一边在 /tmp 解开  
[root@www ~]# cd /tmp  
[root@www ~]# tar -cvf - /etc | tar -xvf -  
# 这个动作有点像是 cp -r /etc /tmp 啦～依旧是有其有  
用途的！  
# 要注意的地方在于输出档变成 - 而输入档也变成 - ，又  
有一个 | 存在～
```

```
# 这分别代表 standard output, standard input 与管线命令啦！
# 简单的想法中，你可以将 - 想成是在内存中的一个装置（缓冲区）。
# 更详细的数据流与管线命令，请翻到 bash 章节啰！
```

在上面的例子中，我们想要『将 /etc 底下的资料直接 copy 到目前所在的路径，也就是 /tmp 底下』，但是又觉得使用 `cp -r` 有点麻烦，那么就直接以这个打包的方式来打包，其中，指令里面的 `-` 就是表示那个被打包的档案啦！由于我们不想让中间档案存在，所以就以这一个方式来进行复制的行为啦！

---

- 例题：系统备份范例

系统上有非常多的重要目录需要进行备份，而且其实我们也不建议你将备份数据放置到 /root 目录下！假设目前你已经知道重要的目录有底下这几个：

- /etc/（配置文件）
- /home/（用户的家目录）
- /var/spool/mail/（系统中，所有账号的邮件信箱）
- /var/spool/cron/（所有账号的工作排成配置文件）
- /root（系统管理员的家目录）

然后我们也知道，由于[第八章](#)曾经做过的练习的关系，/home/loop\* 不需要备份，而且 /root 底下的压缩文件也不需要备份，另外假设你要将备份的数据放置到 /backups，并且该目录仅有 root 有权限进入！此外，每次备份的档名都希望不相同，例如使用：backup-system-20081130.tar.bz2 之类的档名来处理。那你该如何处理这个备份数据呢？（请先动手作看看，再来察看一下底下的参考解答！）

```
# 1. 先处理要放置备份数据的目录与权限：
[root@www ~]# mkdir /backups
[root@www ~]# chmod 700 /backups
[root@www ~]# ll -d /backups
drwx----- 2 root root 4096 Nov 30 16:35 /backups

# 2. 假设今天是 2008/11/30，则建立备份的方式如下：
[root@www ~]# tar -jcv -f
/backups/backup-system-20081130.tar.bz2 \
> --exclude=/root/*.bz2 --exclude=/root/*.gz
--exclude=/home/loop* \
> /etc /home /var/spool/mail /var/spool/cron /root
```

```
.... (过程省略)....
```

```
[root@www ~]# ll -h /backups/  
-rw-r--r-- 1 root root 8.4M Nov 30 16:43  
backup-system-20081130.tar.bz2
```



完整备份工具：dump

某些时刻你想要针对文件系统进行备份或者是储存的功能时，不能不谈到这个 dump 指令！这玩意儿我们曾在前一章的 [/etc/fstab](#) 里面稍微谈过。其实这个指令除了能够针对整个 filesystem 备份之外，也能够仅针对目录来备份喔！底下就让我们来谈一谈这个指令的用法吧！



dump

其实 dump 的功能颇强，他除了可以备份整个文件系统之外，还可以制定等级喔！什么意思啊！假设你的 /home 是独立的一个文件系统，那你第一次进行过 dump 后，再进行第二次 dump 时，你可以指定不同的备份等级，假如指定等级为 1 时，此时新备份的数据只会记录与第一次备份所有差异的档案而已。看不懂吗？没关系！我们用一张简图来说明。

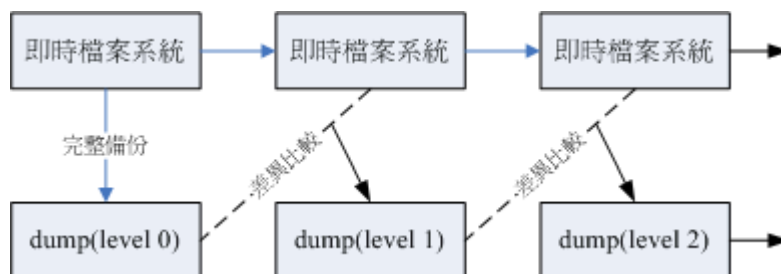


图 4.1.1、dump 运作的等级 (level)

如上图所示，上方的『实时文件系统』是一直随着时间而变化的数据，例如在 /home 里面的档案数据会一直变化一样。而底下的方块则是 dump 备份起来的数据，第一次备份时使用的是 level 0，这个等级也是完整的备份啦！等到第二次备份时，实时文件系统内的数据已经与 level 0 不一样了，而 level 1 仅只是比较目前的文件系统与 level 0 之间的差异后，备份有变化过的档案而已。至于 level 2 则是与 level 1 进行比较啦！这样了解呼？

虽然 dump 支持整个文件系统或者是单一各别目录，但是对于目录的支持是比较不足的，这也是 dump 的限制所在。简单的说，如果想要备份的数据如下时，则有不同的限制情况：

- 当待备份的资料为单一文件系统：  
如果是单一文件系统（filesystem），那么该文件系统可以使用完整的 dump 功能，包括利用 0~9 的数个 level 来备份，同时，备份时可以使用挂载点或者是装置文件名（例如 /dev/sda5 之类的装置文件名）来进行备份！
- 待备份的数据只是目录，并非单一文件系统：  
例如你仅想要备份 /home/someone/，但是该目录并非独立的文件系统时。此时备份就有限制啦！包括：
  - 所有的备份数据都必须要在该目录（本例为：/home/someone/）底下；
  - 且仅能使用 level 0，亦即仅支持完整备份而已；
  - 不支持 -u 选项，亦即无法建立 /etc/dumpdates 这个各别 level 备份的时间记录文件；

dump 的选项虽然非常的繁复，不过如果只是想要简单的操作时，您只要记得底下的几个选项就很够用了！

```
[root@www ~]# dump [-Suvj] [-level] [-f 备份档] 待备份资料
[root@www ~]# dump -W
```

选项与参数：

- S：仅列出后面的待备份数据需要多少磁盘空间才能够备份完毕；
- u：将这次 dump 的时间记录到 /etc/dumpdates 档案中；
- v：将 dump 的档案过程显示出来；
- j：加入 bzip2 的支持！将数据进行压缩，默认 bzip2 压缩等级为 2
- level：就是我们谈到的等级，从 -0 ~ -9 共十个等级；
- f：有点类似 tar 啦！后面接产生的档案，亦可接例如 /dev/st0 装置文件名等
- W：列出在 /etc/fstab 里面的具有 dump 设定的 partition 是否有备份过？

- 用 dump 备份完整的文件系统

现在就让我们来做几个范例吧！假如你要将系统的最小的文件系统提出来进行备份，那该如何进行呢？

```
# 1. 先找出系统中最小的那个文件系统，如下所示：
```

```

[root@www ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/hdc2                  9.5G     3.7G   5.3G   42% /
/dev/hdc3                  4.8G     651M   3.9G   15% /home
/dev/hdc1                   99M       11M    83M   12% /boot
<==看起来最小的就是他啦!
tmpfs                     363M          0   363M    0% /dev/shm

# 2. 先测试一下，如果要备份此文件系统，需多少容量？
[root@www ~]# dump -S /dev/hdc1
5630976    <==注意一下，这个单位是 bytes ，所以差不多是 5.6MBytes。

# 3. 将完整备份的文件名记录成为 /root/boot.dump ，同时更新记录文件：
[root@www ~]# dump -0u -f /root/boot.dump /boot
DUMP: Date of this level 0 dump: Tue Dec  2 02:53:45 2008 <==记录等级与备份时间
DUMP: Dumping /dev/hdc1 (/boot) to /root/boot.dump
<==dump 的来源与目标
DUMP: Label: /boot
<==文件系统的 label
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files]
<==开始进行档案对应
DUMP: mapping (Pass II) [directories]
DUMP: estimated 5499 blocks.
<==评估整体 block 数量
DUMP: Volume 1 started with block 1 at: Tue Dec  2 02:53:46 2008
DUMP: dumping (Pass III) [directories]
<==开始 dump 工作
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /root/boot.dump
<==结束写入备份档
DUMP: Volume 1 completed at: Tue Dec  2 02:53:47 2008
DUMP: Volume 1 5550 blocks (5.42MB)
<==最终备份数据容量
DUMP: Volume 1 took 0:00:01
DUMP: Volume 1 transfer rate: 5550 kB/s
DUMP: 5550 blocks (5.42MB) on 1 volume(s)
DUMP: finished in 1 seconds, throughput 5550

```

```

kBytes/sec
  DUMP: Date of this level 0 dump: Tue Dec  2 02:53:45
2008
  DUMP: Date this dump completed:  Tue Dec  2 02:53:47
2008
  DUMP: Average transfer rate: 5550 kB/s
  DUMP: DUMP IS DONE
# 在指令的下达方面，dump 后面接 /boot 或 /dev/hdc1
都可以的！
# 而执行 dump 的过程中会出现如上的一些讯息，您可以自
行仔细的观察！

[root@www ~]# ll /root/boot.dump /etc/dumpdates
-rw-rw-r-- 1 root disk      43 Dec  2 02:53
/etc/dumpdates
-rw-r--r-- 1 root root 5683200 Dec  2 02:53
/root/boot.dump
# 由于加上 -u 的选项，因此 /etc/dumpdates 该档案的内
容会被更新！注意，
# 这个档案仅有在 dump 完整的文件系统时才有支持主动
更新的功能。

# 4. 观察一下系统主动建立的记录文件：
[root@www ~]# cat /etc/dumpdates
/dev/hdc1      0   Tue Dec  2 02:53:47 2008 +0800
[文件系统] [等级] [      ctime 的时间      ]

```

这样很简单的就建立起来 /root/boot.dump 档案，该档案将整个 /boot/ 文件系统都备份下来了！并且将备份的时间写入 /etc/dumpdates 档案中，准备让下次备份时可以作为参考依据。现在让我们来进行一个测试，检查看看能否真的建立 level 1 的备份呢？

```

# 0. 看一下有没有任何文件系统被 dump 过的资料？
[root@www ~]# dump -W
Last dump(s) done (Dump '>' file systems):
> /dev/hdc2      (      /) Last dump: never
> /dev/hdc3      ( /home) Last dump: never
  /dev/hdc1      ( /boot) Last dump: Level 0, Date Tue
Dec  2 02:53:47 2008
# 如上列的结果，该结果会提出 /etc/fstab 里面第五字段
设定有需要 dump 的
# partition，然后与 /etc/dumpdates 进行比对，可以得
到上面的结果啦！

```



```

# 尤其是第三行，可以显示我们曾经对 /dev/hdc1 进行过
dump 的备份动作喔！

# 1. 先恶搞一下，建立一个大约 10 MB 的档案在 /boot
内：
[root@www ~]# dd if=/dev/zero of=/boot/testing.img
bs=1M count=10
10+0 records in
10+0 records out
10485760 bytes (10 MB) copied, 0.166128 seconds, 63.1
MB/s

# 2. 开始建立差异备份档，此时我们使用 level 1 吧：
[root@www ~]# dump -lu -f /root/boot.dump.1 /boot
.... (中间省略)....

[root@www ~]# ll /root/boot*
-rw-r--r-- 1 root root 5683200 Dec  2 02:53
/root/boot.dump
-rw-r--r-- 1 root root 10547200 Dec  2 02:56
/root/boot.dump.1
# 看看档案大小，岂不是就是刚刚我们所建立的那个档案
的容量吗？ ^_^

# 3. 最后再看一下是否有记录 level 1 备份的时间点呢？
[root@www ~]# dump -W
Last dump(s) done (Dump '>' file systems):
> /dev/hdc2      (    /) Last dump: never
> /dev/hdc3      ( /home) Last dump: never
> /dev/hdc1      ( /boot) Last dump: Level 1, Date Tue
Dec  2 02:56:33 2008
.... (中间省略)....

```

透过这个简单的方式，我们就能够仅备份差异档案的部分啰！底下再来看看针对单一目录的 dump 用途！

- 
- 用 dump 备份非文件系统，亦即单一目录的方法

现在让我们来处理一下 /etc 的 dump 备份吧！因为 /etc 并非单一文件系统，他只是个目录而已。所以依据限制的说明，-u, level 1~9 都是不适用的。我们只能使用 level 0 的完整备份将 /etc 给他 dump 下来。因此作法就变的很简单了！

```

# 让我们将 /etc 整个目录透过 dump 进行备份，且含压缩
功能
[root@www ~]# dump -0j -f /root/etc.dump.bz2 /etc
  DUMP: Date of this level 0 dump: Tue Dec  2 12:08:22
2008
  DUMP: Dumping /dev/hdc2 (/ (dir etc)) to
/root/etc.dump.bz2
  DUMP: Label: /1
  DUMP: Writing 10 Kilobyte records
  DUMP: Compressing output at compression level 2
(bzlib)
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 115343 blocks.
  DUMP: Volume 1 started with block 1 at: Tue Dec  2
12:08:23 2008
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /root/etc.dump.bz2
  DUMP: Volume 1 completed at: Tue Dec  2 12:09:49 2008
  DUMP: Volume 1 took 0:01:26
  DUMP: Volume 1 transfer rate: 218 kB/s
  DUMP: Volume 1 124680kB uncompressed, 18752kB
compressed, 6.649:1
  DUMP: 124680 blocks (121.76MB) on 1 volume(s)
  DUMP: finished in 86 seconds, throughput 1449
kBytes/sec
  DUMP: Date of this level 0 dump: Tue Dec  2 12:08:22
2008
  DUMP: Date this dump completed:  Tue Dec  2 12:09:49
2008
  DUMP: Average transfer rate: 218 kB/s
  DUMP: Wrote 124680kB uncompressed, 18752kB
compressed, 6.649:1
  DUMP: DUMP IS DONE
# 上面特殊字体的部分显示：原本有 124680kb 的容量，
被压缩成为 18752kb，
# 整个压缩比为 6.649:1 ，还可以的压缩情况啦！

```

一般来说 dump 不会使用包含压缩的功能，不过如果你想要将备份的空间降低的话，那个 `-j` 的选项是可以使用的。加上 `-j` 之后你的 dump 成果会使用较少的硬盘容量啦！如上述的情况来看，档案容量由原本的 128MB 左右下滑到 18MB 左右，当然可以节省备份空间啰！

---

## restore

备份文件就是在急用时可以回复系统的重要数据，所以有备份当然就得要学学如何复原了！ dump 的复原使用的是 restore 这个指令！这个指令的选项也非常的多～您可以自行 man restore 瞧瞧！ 鸟哥在这里仅作个简单的介绍啰！

```
[root@www ~]# restore -t [-f dumpfile] [-h] <==  
用来察看 dump 档  
[root@www ~]# restore -C [-f dumpfile] [-D 挂载点] <==  
比较 dump 与实际档案  
[root@www ~]# restore -i [-f dumpfile] <==  
进入互动模式  
[root@www ~]# restore -r [-f dumpfile] <==  
还原整个文件系统  
选项与参数：  
相关的各种模式，各种模式无法混用喔！例如不可以写 -tC  
啦！  
-t  ：此模式用在察看 dump 起来的备份文件中含有什么重要数据！类似 tar -t 功能；  
-C  ：此模式可以将 dump 内的数据拿出来跟实际的文件系统做比较，  
        最终会列出『在 dump 档案内有记录的，且目前文件系统不一样』的档案；  
-i  ：进入互动模式，可以仅还原部分档案，用在 dump 目录时的还原！  
-r  ：将整个 filesystem 还原的一种模式，用在还原针对文件系统的 dump 备份；  
其他较常用到的选项功能：  
-h  ：察看完整备份数据中的 inode 与文件系统 label 等信息  
-f  ：后面就接你要处理的那个 dump 档案啰！  
-D  ：与 -C 进行搭配，可以查出后面接的挂载点与 dump 内有不同的档案！
```

- 用 restore 观察 dump 后的备份数据内容

要找出 dump 的内容就使用 restore -t 来查阅即可！例如我们将 boot.dump 的档案内容捉出来看看！

```
[root@www ~]# restore -t -f /root/boot.dump
```

```

Dump   date: Tue Dec  2 02:53:45 2008
<==说明备份的日期
Dumped from: the epoch
Level 0 dump of /boot on www.vbird.tsai:/dev/hdc1  <==
说明 level 状态
Label: /boot
<==说明该 filesystem 的表头!
      2      .
     11      ./lost+found
    2009      ./grub
    2011      ./grub/grub.conf
....底下省略....

[root@www ~]# restore -t -f /root/etc.dump
Dump tape is compressed.           <==
加注说明数据有压缩
Dump   date: Tue Dec  2 12:08:22 2008
Dumped from: the epoch
Level 0 dump of / (dir etc) on www.vbird.tsai:/dev/hdc2
<==是目录!
Label: /1
      2      .
   1912545    ./etc
   1912549    ./etc/rpm
   1912550    ./etc/rpm/platform
....底下省略....

```

这个查阅的数据其实显示出的是文件名与源文件的 inode 状态，所以我们可以说，dump 会参考 inode 的记录哩！透过这个查询我们也能知道 dump 的内容为何呢！再来查一查如何还原吧！

- 
- 比较差异并且还原整个文件系统

为什么 dump 可以进行累积备份呢？就是因为他具有可以查询文件系统与备份文件之间的差异，并且将分析到的差异数据进行备份的缘故。所以我们先来看看，如何查询有变动过的信息呢？你可以使用如下的方法检验：

```

# 0. 先尝试变更文件系统的内容：
[root@www ~]# cd /boot
[root@www boot]# mv config-2.6.18-92.el5
config-2.6.18-92.el5-back

```

```

# 1. 看使进行文件系统与备份文件之间的差异！
[root@www boot]# restore -C -f /root/boot.dump
Dump   date: Tue Dec  2 02:53:45 2008
Dumped from: the epoch
Level 0 dump of /boot on www.vbird.tsai:/dev/hdc1
Label: /boot
filesys = /boot
restore: unable to stat ./config-2.6.18-92.el5: No
such file or directory
Some files were modified!  1 compare errors
# 看到上面的特殊字体了吧！ 那就是有差异的部分！ 总共有
一个档案被变更！
# 我们刚刚确实有更动过该档案，嘿嘿！ 这样是否能了解？！

# 2. 将文件系统改回来啊！
[root@www boot]# mv config-2.6.18-92.el5-back
config-2.6.18-92.el5
[root@www boot]# cd /root

```

如同上面的动作，透过曾经备份过的信息，也可以找到与目前实际文件系统中  
有差异的数据呢！ 如果你不想要进行累积备份，但也能透过这个动作找出最近这  
一阵子有变动过的档案说！ 了解乎？ 那如何还原呢？ 由于 dump 是记录整个文  
件系统的，因此还原时你也应该要给予一个全新的文件系统才行。 因此底下我  
我们先建立一个文件系统，然后再来还原吧！

```

# 1. 先建立一个新的 partition 来使用，假设我们需要的是 150M 的容量
[root@www ~]# fdisk /dev/hdc
Command (m for help): n
First cylinder (2335-5005, default 2335): <==这里按
Enter
Using default value 2335
Last cylinder or +size or +sizeM or +sizeK (2335-5005,
default 5005): +150M
Command (m for help): p
.... 中间省略....
/dev/hdc8           2335           2353           152586
83  Linux

Command (m for help): w

[root@www ~]# partprobe  <==很重要的动作！ 别忘记！

```

```
# 这样就能够建立一个 /dev/hdc8 的 partition , 然后继续格式化吧!
```

```
[root@www ~]# mkfs -t ext3 /dev/hdc8  
[root@www ~]# mount /dev/hdc8 /mnt
```

```
# 2. 开始进行还原的动作! 请您务必到新文件系统的挂载点底下去!
```

```
[root@www ~]# cd /mnt  
[root@www mnt]# restore -r -f /root/boot.dump  
restore: ./lost+found: File exists
```

由于我们是备份整个文件系统，因此你也可以建置一个全新的文件系统 (partition) 来进行还原的动作！整个还原的动作也不难，如上表最后一个指令，就是将备份文件中的数据还原到本目录下。你必须要变更目录到挂载点所在的那个目录才行啊！这样还原的档案才不会跑错地方！如果你还想要将 level 1 的 /root/boot.dump.1 那个档案的内容也还原的话，那就继续使用『restore -r -f /root/boot.dump.1』去还原吧！

- 
- 仅还原部分档案的 restore 互动模式

某些时候你只是要将备份档的某个内容捉出来而已，并不想要全部解开，那该如何是好？此时你可以进入 restore 的互动模式 (interactive mode)。在底下我们使用 etc.dump 来进行范例说明。假如你要将 etc.dump 内的 passwd 与 shadow 捉出来而已，该如何进行呢？

```
[root@www ~]# cd /mnt  
[root@www mnt]# restore -i -f /root/etc.dump  
restore >  
# 此时你就已经进入 restore 的互动模式画面中! 要注意的是:  
# 你目前正在 etc.dump 这个档案内了! 所有的动作都是在 etc.dump 内!  
  
restore > help  
Available commands are:  
      ls [arg] - list directory          <==列出  
etc.dump 内的档案或目录  
      cd arg - change directory          <==在  
etc.dump 内变更目录  
      pwd - print current directory      <==列出在
```

```

etc.dump 内的路径文件名
    add [arg] - add `arg' to list of files to be
extracted
    delete [arg] - delete `arg' from list of files
to be extracted
    extract - extract requested files
# 上面三个指令是重点！各指令的功能为：
# add file      : 将 file 加入等一下要解压缩的档案列表
中
# delete file   : 将 file 移除出解压缩的列表，并非删除
etc.dump 内的档案！别误会！^_^
# extract       : 开始将刚刚选择的档案列表解压缩了去！
    setmodes - set modes of requested directories
    quit - immediately exit program
    what - list dump header information
    verbose - toggle verbose flag (useful with
`ls'`)
    prompt - toggle the prompt display
    help or `?' - print this list

restore > ls
.:
etc/ <==会显示出在 etc.dump 内主要的目录，因为我们
备份 /etc ，所以档名为此！

restore > cd etc                <==在 etc.dump 内
变换路径到 etc 目录下
restore > pwd                  <==列出本目录的文件
名称为？
/etc
restore > ls passwd shadow group <==看看，真的有这
三个档案喔！
passwd
shadow
group
restore > add passwd shadow group <==加入解压缩列表
restore > delete group          <==加错了！将 group
移除解压缩列表
restore > ls passwd shadow group
*passwd <==有要被解压缩的，档名之前会出现 * 的符号
呢！
*shadow
group

```



```

restore > extract                                <==开始进行解压缩
去!
You have not read any volumes yet.    <==这里会询问你
需要的 volume
Unless you know which volume your file(s) are on you
should start
with the last volume and work towards the first.
Specify next volume # (none if no more volumes): 1 <==
只有一个 volume
set owner/mode for '.'? [yn] n <==不需要修改权限

restore > quit                                    <==离开 restore 的
功能

[root@www ~]# ll -d etc
drwxr-xr-x 2 root root 1024 Dec 15 17:49 etc <==解
压缩后，所建立出来的目录啦!
[root@www ~]# ll etc
total 6
-rw-r--r-- 1 root root 1945 Sep 29 02:21 passwd
-r----- 1 root root 1257 Sep 29 02:21 shadow

```

透过交互式的 restore 功能，可以让你将备份的数据取出一部份，而不必全部都得解压缩才能够取得你想要的档案数据。而 restore 内的 add 除了可以增加档案外，也能够增加整个备份的『目录』喔！还不错玩吧！赶紧测试看看先！

^^  
—



## 光盘写入工具

某些时刻你可能会希望将系统上最重要的数据给他备份出来，虽然目前随身碟已经有够便宜，你可以使用这玩意儿来备份。不过某些重要的、需要重复备份的数据(可能具有时间特性)，你可能会需要使用类似 DVD 之类的储存媒体来备份出来！举例来说，你的系统配置文件或者是讨论区的数据库档案(变动性非常的频繁)。虽然 Linux 图形接口已经有不少的刻录软件可用，但有时如果你希望系统自动在某些时刻帮你主动的进行刻录时，那么文字接口的刻录行为就有帮助啦！

那么文本模式的刻录行为要怎么处理呢？通常的作法是这样的：

- 先将所需要备份的数据建置成为一个映像档(iso)，利用 mkisofs 指令来处理；
- 将该映像文件刻录至光盘或 DVD 当中，利用 cdrecord 指令来处理。

底下我们就分别来谈谈这两个指令的用法吧！

---

## mkisofs: 建立映像档

我们从 FTP 站捉下来的 Linux 映像档（不管是 CD 还是 DVD）都得要继续刻录成为实体的光盘/DVD 后，才能够进一步的使用，包括安装或更新你的 Linux 啦！同样的道理，你想要利用刻录机将你的数据刻录到 DVD 时，也得要先将你的数据报成一个映像档，这样才能够写入 DVD 片中。而将你的数据报成一个映像档的方式就透过 mkisofs 这个指令即可。mkisofs 的使用方式如下：

```
[root@www ~]# mkisofs [-o 映像档] [-rv] [-m file] 待
备份文件.. [-V vol] \
> -graft-point isodir=systemdir ...
```

选项与参数：

- o : 后面接你想要产生的那个映像档档名。
- r : 透过 Rock Ridge 产生支持 Unix/Linux 的档案数据，可记录较多的信息；
- v : 显示建置 ISO 档案的过程
- m file : -m 为排除档案 (exclude) 的意思，后面的档案不备份到映像档中
- V vol : 建立 Volume，有点像 Windows 在档案总管内看到的 CD title 的东西
- graft-point: graft 有转嫁或移植的意思，相关资料在底下文章内说明。

其实 mkisofs 有非常多好用的选项可以选择，不过如果我们只是想要制作数据光盘时，上述的选项也就够用了。光盘的格式一般称为 iso9660，这种格式一般仅支持旧版的 DOS 档名，亦即档名只能以 8.3（文件名 8 个字符，扩展名 3 个字符）的方式存在。如果加上 -r 的选项之后，那么档案信息能够被记录的比较完整，可包括 UID/GID 与权限等等！所以，记得加这个 -r 的选项。

此外，一般预设的情况下，所有要被加到映像档中的档案都会被放置到映像文件中的根目录，如此一来可能会造成刻录后的档案分类不易的情况。所以，你可以使用 -graft-point 这个选项，当你使用这个选项之后，可以利用如下的方法来定义位于映像文件中的目录，例如：

- 映像文件中的目录所在=实际 Linux 文件系统的目录所在
- /movies/= /srv/movies/（在 Linux 的 /srv/movies 内的档案，加至映像文件中的 /movies/ 目录）
- /linux/etc=/etc（将 Linux 中的 /etc/ 内的所有数据备份到映像文件中的 /linux/etc/ 目录中）

我们透过一个简单的范例来说明一下吧。如果你想要将 /root, /home, /etc 等目录内的数据通通刻录起来的话，先得要处理一下映像档，我们先不使用 -graft-point 的选项来处理这个映像档试看看：

```
[root@www ~]# mkisofs -r -v -o /tmp/system.img /root
/home etc
INFO:   ISO-8859-1 character encoding detected by
locale settings.
        Assuming ISO-8859-1 encoded filenames on
source filesystem,
        use -input-charset to override.
mkisofs 2.01 (cpu-pc-linux-gnu)
Scanning /root
Scanning /root/test4
.... 中间省略....
 97.01% done, estimate finish Tue Dec 16 17:07:14 2008
<==显示百分比
 98.69% done, estimate finish Tue Dec 16 17:07:15 2008
Total translation table size: 0
Total rockridge attributes bytes: 9840    <==额外记录
属性所耗用之容量
Total directory bytes: 55296              <==目录占用
容量
Path table size(bytes): 406
Done with: The File(s)                    Block(s)
298728
Writing:   Ending Padblock                Start Block
298782
Done with: Ending Padblock                Block(s)    150
Max brk space used 0
298932 extents written (583 MB)

[root@www ~]# ll -h /tmp/system.img
-rw-r--r-- 1 root root 584M Dec 16 17:07
/tmp/system.img
[root@www ~]# mount -o loop /tmp/system.img /mnt
[root@www ~]# df -h
Filesystem                Size  Used Avail Use% Mounted
on
/tmp/system.img           584M  584M    0 100% /mnt
<==就是这玩意儿！
[root@www ~]# ls /mnt
alex                crontab2            etc. tar. gz
```

```

system.tar.bz2
anaconda-ks.cfg  etc                      install.log
test1
arod             etc.and.root.tar.bz2
install.log.syslog test2
boot.dump        etc.dump                  loopdev
test3
# 看吧！一堆数据都放置在一起！包括有的没有的目录与档案等等！

[root@www ~]# umount /mnt

```

由上面的范例我们可以看到，三个目录（/root, /home, /etc）的数据通通放置到了映像文件的最顶层目录中！真是不方便～尤其由于 /root/etc 的存在，导致那个 /etc 的数据似乎没有被包含进来的样子！真不合理～而且还有 lost+found 的目录存在！真是超不喜欢的！此时我们可以使用 -graft-point 来处理啰！

```

[root@www ~]# mkisofs -r -V 'linux_file' -o
/tmp/system.img \
> -m /home/lost+found -graft-point /root=/root
/home=/home /etc=/etc
[root@www ~]# ll -h /tmp/system.img
-rw-r--r-- 1 root root 689M Dec 17 11:41
/tmp/system.img
# 上面的指令会建立一个档案，期中 -graft-point 后面
接的就是我们要备份的数据。
# 必须要注意的是那个等号的两边，等号左边是在映像文件
内的目录，右侧则是实际的数据。

[root@www ~]# mount -o loop /tmp/system.img /mnt
[root@www ~]# ll /mnt
dr-xr-xr-x 105 root root 32768 Dec 17 11:40 etc
dr-xr-xr-x  5 root root  2048 Dec 17 11:40 home
dr-xr-xr-x  7 root root  4096 Dec 17 11:40 root
# 瞧！数据是分门别类的在各个目录中喔这样了解乎？最后
将数据卸除一下：

[root@www ~]# umount /mnt

```

其实鸟哥一直觉得很奇怪，怎么我的数据会这么大(600多MB)？原来是 /home 里面在第八章的时候，练习时多了一个 /home/loopdev 的大档案！所以在重新制作一次 iso 档，并多加一个『-m /home/loopdev』来排除该档案的备份，最

终的档案则仅有 176MB 啰！还好还好！ ^\_^！接下来让我们处理刻录的动作了吧！

---

## cdrecord: 光盘刻录工具

我们是透过 cdrecord 这个指令来进行文字接口的刻录行为, 这个指令常见的选项有底下数个:

```
[root@www ~]# cdrecord -scanbus dev=ATA
<==查询刻录机位置
[root@www ~]# cdrecord -v dev=ATA:x,y,z
blank=[fast|all] <==抹除重复读写片
[root@www ~]# cdrecord -v dev=ATA:x,y,z -format
<==格式化 DVD+RW
[root@www ~]# cdrecord -v dev=ATA:x,y,z [可用选项功能] file.iso
```

选项与参数:

- scanbus : 用在扫描磁盘总线并找出可用的刻录机, 后续的装置为 ATA 接口
- v : 在 cdrecord 运作的过程中, 显示过程而已。
- dev=ATA:x,y,z : 后续的 x, y, z 为你系统上刻录机所在的位置, 非常重要!
- blank=[fast|all]: blank 为抹除可重复写入的 CD/DVD-RW, 使用 fast 较快, all 较完整
- format : 仅针对 DVD+RW 这种格式的 DVD 而已;

[可用选项功能] 主要是写入 CD/DVD 时可使用的选项, 常见的选项包括有:

- data : 指定后面的档案以数据格式写入, 不是以 CD 音轨(-audio)方式写入!
- speed=X : 指定刻录速度, 例如 CD 可用 speed=40 为 40 倍数, DVD 则可用 speed=4 之类
- eject : 指定刻录完毕后自动退出光盘
- fs=Ym : 指定多少缓冲存储器, 可用在将映像档先暂存至缓冲存储器。预设 4m, 一般建议可增加到 8m, 不过, 还是得视你的刻录机而定。

针对 DVD 的选项功能:

- driveropts=burnfree : 打开 Buffer Underrun Free 模式的写入功能
- sao : 支持 DVD-RW 的格式

- 侦测你的刻录机所在位置:

文本模式的刻录确实是比较麻烦的，因为没有所见即所得的环境嘛！要刻录首先就得要找到刻录机才行！而由于早期的刻录机都是使用 SCSI 接口，因此查询刻录机的方法就得要配合着 SCSI 接口的认定来处理了。查询刻录机的方式为：

```
[root@www ~]# cdrecord -scanbus dev=ATA
Cdrecord-Clone 2.01 (cpu-pc-linux-gnu) Copyright (C)
1995-2004 J?rg Schilling
....中间省略....
scsibus1:
    1, 0, 0   100) *
    1, 1, 0   101) 'ASUS      ' 'DRW-2014S1      '
' 1.01' Removable CD-ROM
    1, 2, 0   102) *
    1, 3, 0   103) *
    1, 4, 0   104) *
    1, 5, 0   105) *
    1, 6, 0   106) *
    1, 7, 0   107) *
```

利用 `cdrecord -scanbus` 就能够找到正确的刻录机！由于目前个人计算机上最常使用的磁盘驱动器接口为 IDE 与 SATA，这两种接口都能够使用 `dev=ATA` 这种模式来查询，因此上述的指令得要背一下啦！另外，在查询的结果当中可以发现有一台刻录机，其中也显示出这台刻录机的型号，而最重要的就是上表中有底线的那三个数字！那三个数字就是代表这台刻录机的位置！以上表的例子中，这部刻录机的位置在『ATA:1,1,0』这个地方喔！

好了，那么现在要如何将 `/tmp/system.img` 刻录到 CD/DVD 里面去呢？鸟哥这里先以 CD 为例，鸟哥用的是 CD-RW（可重复读写）的光盘片，说实在话，虽然 CD-RW 或 DVD-RW 比较贵一点，不过至少可以重复利用，对环境的冲击比较小啦！建议大家使用可重复读写的片子。由于 CD-RW 可能要先进行抹除的工作（将原本里面的数据删除）然后才能写入，因此，底下我们先来看看如何抹除一片 CD/DVD 的方法，然后直接写入光盘吧！

#### Tips:

由于 CD/DVD 都是使用 `cdrecord` 这个指令，因此不论是 CD 还是 DVD 片，下达指令的方法都差不多！不过，DVD 的写入需要额外的 `driveropts=burnfree` 或 `-dao` 等选项的辅助才行。另外，CD 片有 CD-R（一次写入）与 CD-RW（重复写入），至于 DVD 则主要有两种格式，分别是 DVD-R 及 DVD+R 两种格式。如果是可重复读写的则为：DVD-RW，DVD+RW。除了 DVD+RW 的抹除方法可能不太一样



之外，其他写入的方式则是一样的。

---

- 进行 CD 的刻录动作：

```
# 0. 先抹除光盘的原始内容：（非可重复读写则可略过此步骤）
[root@www ~]# cdrecord -v dev=ATA:1,1,0 blank=fast
# 中间会跑出一堆讯息告诉你抹除的进度，而且会有 10 秒钟的时间等待你的取消！
# 可以避免『手滑』的情况！ ^_^

# 1. 开始刻录：
[root@www ~]# cdrecord -v dev=ATA:1,1,0 fs=8m -dummy -data \
> /tmp/system.img
.... 中间省略....
Track 01: 168 of 176 MB written (fifo 100%) [buf 100%] 10.5x. <==显示百分比
# 上面会显示进度，还有 10.5x 代表目前的刻录速度！
cdrecord: fifo had 2919 puts and 2919 gets.
cdrecord: fifo was 0 times empty and 2776 times full, min fill was 97%.

# 2. 刻录完毕后，测试挂载一下，检验内容：
[root@www ~]# mount -t iso9660 /dev/cdrom /mnt
[root@www ~]# df -h /mnt
Filesystem      Size  Used Avail Use% Mounted on
/dev/hdd        177M  177M    0 100% /mnt
<==瞧！确实是光盘内容！

[root@www ~]# ll /mnt
dr-xr-xr-x 105 root root 32768 Dec 17 11:54 etc
dr-xr-xr-x  5 root root  2048 Dec 17 11:54 home
dr-xr-xr-x  7 root root  4096 Dec 17 11:54 root

[root@www ~]# umount /mnt    <==不要忘了卸除
```

事实上如果你忘记抹除可写入光盘时，其实 `cdrecord` 很聪明的会主动的帮你抹除啦！因此上面的信息你只要记得刻录的功能即可。特别注意 `-data` 那个选项！因为如果没有加上 `-data` 的选项时，默认数据会以音轨格式写入光盘中，所以最好能够加上 `-data` 这个选项啰！上述的功能是针对 CD，底下我们使用一片



可重复读写的 DVD-RW 来测试一下写入的功能！

---

- 进行 DVD-RW 的刻录动作：

```
# 0. 同样的，先来抹除一下原本的内容：
[root@www ~]# cdrecord -v dev=ATA:1,1,0 blank=fast

# 1. 开始写入 DVD ，请注意，有些选项与 CD 并不相同了
喔！
[root@www ~]# cdrecord -v dev=ATA:1,1,0 fs=8m -data
-sao \
> driveropts=burnfree /tmp/system.img

# 2. 同样的，来给他测试测试！
[root@www ~]# mount /dev/cdrom /mnt
[root@www ~]# df -h /mnt
Filesystem            Size  Used Avail Use% Mounted
on
/dev/hdd                177M  177M    0 100% /mnt
[root@www ~]# umount /mnt
```

整体指令没有差很多啦！只是 CD-RW 会自动抹除，但 DVD-RW 似乎得要自己手动某除才行！并不会主动进入自动抹除的功能！害鸟哥重新测试过好几次～伤脑筋～^\_^！好啦！现在你就知道如何将你的数据刻录出来啦！

如果你的 Linux 是用来做为服务器之用的话，那么无时无刻的去想『如何备份重要数据』是相当重要的！关于备份我们会在第五篇再仔细的谈一谈，这里你要会使用这些工具即可！

---



其他常见的压缩与备份工具

还有一些很好用的工具得要跟大家介绍介绍，尤其是 dd 这个玩意儿呢！

---



我们在第八章当中的特殊 loop 装置挂载时使用过 dd 这个指令对吧？不过，这个指令可不只是制作一个档案而已喔～这个 dd 指令最大的功效，鸟哥认为，应该是在于『备份』啊！因为 dd 可以读取磁盘装置的内容(几乎是直接读取扇

区"sector"),然后将整个装置备份成一个档案呢!真的是相当的好用啊~ dd 的用途有很多啦~但是我们仅讲一些比较重要的选项,如下:

```
[root@www ~]# dd if="input_file" of="output_file"
bs="block_size" \
> count="number"
```

选项与参数:

- if : 就是 input file 啰~也可以是装置喔!
- of : 就是 output file 喔~也可以是装置;
- bs : 规划的一个 block 的大小,若未指定则预设是 512 bytes(一个 sector 的大小)
- count: 多少个 bs 的意思。

范例一: 将 /etc/passwd 备份到 /tmp/passwd.back 当中

```
[root@www ~]# dd if=/etc/passwd of=/tmp/passwd.back
3+1 records in
3+1 records out
1945 bytes (1.9 kB) copied, 0.000332893 seconds, 5.8
MB/s
```

```
[root@www ~]# ll /etc/passwd /tmp/passwd.back
-rw-r--r-- 1 root root 1945 Sep 29 02:21 /etc/passwd
-rw-r--r-- 1 root root 1945 Dec 17 18:09
/tmp/passwd.back
```

# 仔细的看一下,我的 /etc/passwd 档案大小为 1945 bytes, 因为我没有设定 bs ,

# 所以默认是 512 bytes 为一个单位,因此,上面那个 3+1 表示有 3 个完整的

# 512 bytes, 以及未满 512 bytes 的另一个 block 的意思啦!

# 事实上,感觉好像是 cp 这个指令啦~

范例二: 将自己的磁盘之第一个扇区备份下来

```
[root@www ~]# dd if=/dev/hdc of=/tmp/mbr.back bs=512
count=1
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.0104586 seconds, 49.0 kB/s
```

# 第一个扇区内含有 MBR 与 partition table , 透过这个动作,

# 我们可以一口气将这个磁盘的 MBR 与 partition table 进行备份哩!

范例三: 找出你系统最小的那个分割槽, 并且将他备份下来:

```
[root@www ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hdc2       9.5G  3.9G  5.1G  44% /
/dev/hdc3       4.8G  651M  3.9G  15% /home
/dev/hdc1       99M   21M   73M  23% /boot
<==就捉他好了！
[root@www ~]# dd if=/dev/hdc1 of=/tmp/boot.whole.disk
208782+0 records in
208782+0 records out
106896384 bytes (107 MB) copied, 6.24721 seconds, 17.1 MB/s
[root@www ~]# ll -h /tmp/boot.whole.disk
-rw-r--r-- 1 root root 102M Dec 17 18:14
/tmp/boot.whole.disk
# 等于是将整个 /dev/hdc1 通通捉下来的意思～如果要还原呢？就反向回去！
# dd if=/tmp/boot.whole.disk of=/dev/hdc1 即可！非常简单吧！
# 简单的说，如果想要整个硬盘备份的话，就类似 Norton 的 ghost 软件一般，
# 由 disk 到 disk，嘿嘿～利用 dd 就可以啦～厉害厉害！
```

你可以说，tar 可以用来备份关键数据，而 dd 则可以用来备份整颗 partition 或 整颗 disk，很不错啊～不过，如果要将数据填回到 filesystem 当中，可能需要考虑到原本的 filesystem 才能成功啊！让我们来完成底下的例题试试看：

例题：

你想要将你的 /dev/hdc1 进行完整的复制到另一个 partition 上，请使用你的系统上面未分割完毕的容量再建立一个与 /dev/hdc1 差不多大小的分割槽（只能比 /dev/hdc1 大，不能比他小！），然后将之进行完整的复制（包括需要复制 boot sector 的区块）。

答：

由于需要复制 boot sector 的区块，所以使用 cp 或者是 tar 这种指令是无法达成需求的！此时那个 dd 就派的上用场了。你可以这样做：

```
# 1. 先进行分割的动作
[root@www ~]# fdisk -l /dev/hdc
   Device Boot      Start         End      Blocks    Id  System
/dev/hdc1    *           1          13       104391    83   Linux
```

```

# 上面鸟哥仅撷取重要的数据而已！我们可以看到
/dev/hdc1 仅有 13 个 sector

[root@www ~]# fdisk /dev/hdc
Command (m for help): n
First cylinder (2354-5005, default 2354): 这里按
enter
Using default value 2354
Last cylinder or +size or +sizeM or +sizeK (2354-5005,
default 5005): 2366

Command (m for help): p
   Device Boot   Start    End  Blocks   Id  System
/dev/hdc9              2354   2366   104391   83   Linux

Command (m for help): w

[root@www ~]# partprobe

# 2. 不需要格式化，直接进行 sector 表面的复制！
[root@www ~]# dd if=/dev/hdc1 of=/dev/hdc9
208782+0 records in
208782+0 records out
106896384 bytes (107 MB) copied, 16.8797 seconds, 6.3
MB/s

[root@www ~]# mount /dev/hdc9 /mnt
[root@www ~]# df
Filesystem            1K-blocks      Used Available
Use% Mounted on
/dev/hdc1              101086    21408      74459
23% /boot
/dev/hdc9              101086    21408      74459
23% /mnt
# 这两个玩意儿会『一模一样』喔！
[root@www ~]# umount /mnt

```

非常有趣的范例吧！新分割出来的 partition 不需要经过格式化，因为 dd 可以将原本旧的 partition 上面，将 sector 表面的数据整个复制过来！当然连同 superblock, boot sector, meta data 等等通通也会复制过来！是否很有趣呢？未来你想要建置两颗一模一样的磁盘时，只要下达类似： dd if=/dev/sda of=/dev/sdb ，就能够让两颗磁盘一模一样，甚至 /dev/sdb 不需要分割与格式化，因为该指令可以将 /dev/sda 内的所有资料，包括 MBR 与 partition table 也

复制到 /dev/sdb 说! ^\_^

## cpio

这个指令挺有趣的，因为 cpio 可以备份任何东西，包括装置设备档案。不过 cpio 有个大问题，那就是 cpio 不会主动的去找档案来备份！啊！那怎么办？所以啰，一般来说，cpio 得要配合类似 [find](#) 等可以找到文件名的指令来告知 cpio 该被备份的数据在哪里啊！有点小麻烦啦～因为牵涉到我们在第三篇才会谈到的[数据流重导向](#)说～所以这里你就先背一下语法，等到第三篇讲完你就知道如何使用 cpio 啰！

```
[root@www ~]# cpio -ovcB > [file|device] <==备份
[root@www ~]# cpio -ivcdu < [file|device] <==还原
[root@www ~]# cpio -ivct < [file|device] <==察看
```

备份会使用到的选项与参数：

- o：将数据 copy 输出到档案或装置上
- B：让预设的 Blocks 可以增加至 5120 bytes，预设是 512 bytes！

这样的好处是可以让大档案的储存速度加快(请参考 i-nodes 的观念)

还原会使用到的选项与参数：

- i：将数据自档案或装置 copy 出来系统当中
- d：自动建立目录！使用 cpio 所备份的数据内容不见得会在同一层目录中，因此我们必须要让 cpio 在还原时可以建立新目录，此时就得要 -d 选项的帮助！
- u：自动的将较新的档案覆盖较旧的档案！
- t：需配合 -i 选项，可用在“察看”以 cpio 建立的档案或装置的内容

一些可共享的选项与参数：

- v：让储存的过程中文件名可以在屏幕上显示
- c：一种较新的 portable format 方式储存

你应该会发现一件事情，就是上述的选项与指令中怎么会没有指定需要备份的数据呢？还有那个大于 (>) 与小于 (<) 符号是怎么回事啊？因为 cpio 会将数据整个显示到屏幕上，因此我们可以透过将这些屏幕的数据重新导向 (>) 一个新的档案！至于还原呢？就是将备份文件读进来 cpio (<) 进行处理之意！我们来进行几个案例你就知道啥是啥了！

范例：找出 /boot 底下的所有档案，然后将他备份到

```

/tmp/boot.cpio 去!
[root@www ~]# find /boot -print
/boot
/boot/message
/boot/initrd-2.6.18-92.el5.img
.... 以下省略....
# 透过这个 find 我们可以找到 /boot 底下应该要存在的
  档名! 包括档案与目录

[root@www ~]# find /boot | cpio -ocvB > /tmp/boot.cpio
[root@www ~]# ll -h /tmp/boot.cpio
-rw-r--r-- 1 root root 16M Dec 17 23:30 /tmp/boot.cpio

```

我们使用 `find /boot` 可以找出档名，然后透过那条管线 (`|`，亦即键盘上的 `shift+\` 的组合)，就能将档名传给 `cpio` 来进行处理！最终会得到 `/tmp/boot.cpio` 那个档案喔！接下来让我们来进行解压缩看看。

```

范例：将刚刚的档案给他在 /root/ 目录下解开
[root@www ~]# cpio -idvc < /tmp/boot.cpio
[root@www ~]# ll /root/boot
# 你可以自行比较一下 /root/boot 与 /boot 的内容是否
  一模一样！

```

事实上 `cpio` 可以将系统的数据完整的备份到磁带上头去喔！如果你有磁带机的话！

- 备份： `find / | cpio -ocvB > /dev/st0`
- 还原： `cpio -idvc < /dev/st0`

这个 `cpio` 好像不怎么好用啦！但是，他可是备份的时候的一项利器呢！因为他可以备份任何的档案，包括 `/dev` 底下的任何装置档案！所以他可是相当重要的呢！而由于 `cpio` 必需要配合其他的程序，例如 `find` 来建立档名，所以 `cpio` 与管线命令及数据流重导向的相关性就相当的重要了！

其实系统里面已经含有一个使用 `cpio` 建立的档案喔！那就是 `/boot/initrd-xxx` 这个档案啦！现在让我们来将这个档案解压缩看看，看你能不能发现该档案的内容为何？

```

# 1. 我们先来看看该档案是属于什么文件格式，然后再加以处理：

```

```

[root@www ~]# file /boot/initrd-2.6.18-92.el5.img
/boot/initrd-2.6.18-92.el5.img: gzip compressed
data, ...
# 唔！看起来似乎是使用 gzip 进行压缩过~那如何处理
呢？

# 2. 透过更名，将该档案增加扩展名，然后予以解压缩看
看：
[root@www ~]# mkdir initrd
[root@www ~]# cd initrd
[root@www initrd]# cp /boot/initrd-2.6.18-92.el5.img
initrd.img.gz
[root@www initrd]# gzip -d initrd.img.gz
[root@www initrd]# ll
-rw----- 1 root root 5408768 Dec 17 23:53 initrd.img
[root@www initrd]# file initrd.img
initrd.img: ASCII cpio archive (SVR4 with no CRC)
# 嘿嘿！露出马脚了吧！确实是 cpio 的文件档喔！

# 3. 开始使用 cpio 解开此档案：
[root@www initrd]# cpio -iduv < initrd.img
sbin
init
sysroot
.... 以下省略....
# 瞧！这样就将这个档案解开啰！这样了解乎？

```

## 重点回顾

- 压缩指令为透过一些运算方法去将原本的档案进行压缩，以减少档案所占用的磁盘容量。压缩前与压缩后的档案所占用的磁盘空间大小，就可以被称为是『压缩比』
- 压缩的好处是可以减少磁盘容量的浪费，在 WWW 网站也可以利用文件压缩的技术来进行数据的传送，好让网站带宽的可利用率上升喔
- 压缩文件案的扩展名大多是：『\*.tar, \*.tar.gz, \*.tgz, \*.gz, \*.Z, \*.bz2』
- 常见的压缩指令有 gzip 与 bzip2，其中 bzip2 压缩比较之 gzip 还要更好，建议使用 bzip2！
- tar 可以用来进行档案打包，并可支持 gzip 或 bzip2 的压缩。
- 压 缩：tar -jcv -f filename.tar.bz2 要被压缩的档案或目录名称
- 查 询：tar -jtv -f filename.tar.bz2
- 解压缩：tar -jxv -f filename.tar.bz2 -C 欲解压缩的目录

- dump 指令可备份文件系统或单一目录
- dump 的备份若针对文件系统时，可进行 0-9 的 level 差异备份！其中 level 0 为完整备份；
- restore 指令可还原被 dump 建置的备份档；
- 要建立光盘刻录数据时，可透过 mkisofs 指令来建置；
- 可透过 cdrecord 来写入 CD 或 DVD 刻录机
- dd 可备份完整的 partition 或 disk，因为 dd 可读取磁盘的 sector 表面数据
- cpio 为相当优秀的备份指令，不过必须要搭配类似 find 指令来读入欲备份的文件名数据，方可进行备份动作。



### 本章习题

(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)

---



### 参考数据与延伸阅读

- 台湾学术网络管理文件：Backup Tools in UNIX(Linux):  
[http://nmc.nchu.edu.tw/tanet/backup\\_tools\\_in\\_unix.htm](http://nmc.nchu.edu.tw/tanet/backup_tools_in_unix.htm)
- 中文 How to 文件计划 (CLDP):  
<http://www.linux.org.tw/CLDP/HOWTO/hardware/CD-Writing-HOWTO/CD-Writing-HOWTO-3.html>
- 熊宝贝工作记录之：Linux 刻录实作：  
[http://csc.ocean-pioneer.com/docum/linux\\_burn.html](http://csc.ocean-pioneer.com/docum/linux_burn.html)
- PHP5 网管实验室：  
<http://www.php5.idv.tw/html.php?mod=article&do=show&shid=26>
- CentOS 5.x 之 man dump
- CentOS 5.x 之 man restore

---

2003/02/09：第一次完成

2003/05/05：修改 tar 的部分内容，尤其是 -P 这个参数的配合用法

2005/07/26：将旧有的文章移动到 [这里](#)

2005/07/27：大略修改了一些风格，另外，动作较大的是在范例的部分！

2005/08/29：加入了 dd 这个有趣的指令喔！

2006/05/02：将原本『tar -zxvpf /tmp/etc.tar.gz /etc』修改为『tar -zcvpf /tmp/etc.tar.gz /etc』感谢讨论区网友 chinu 提供的信息。

2008/10/31：将原本针对 FC4 的旧版本移动到[此处](#)

2008/12/18：这次的改版在这一章添加了不少东西！尤其是将 cpio 与 dd 的范例重新做个整理！并加入 dump/restore, mkisofs/cdrecord

---

2003/02/09 以来统计人数



518856



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](#).