

鳥哥的 Linux 私房菜

為取得較佳瀏覽結果，請愛用 [firefox](#) 瀏覽本網頁

| [鳥站首頁](#) | [基礎學習](#) | [伺服器架設](#) | [企業應用](#) | [桌面應用](#) | [安全管理](#) | [鳥站討論板](#) | [酷學園](#) | [書籍截誤](#) | [關於鳥哥](#) | [崑山資傳](#) |

第九章、檔案與檔案系統的壓縮與打包

切換解析度為 800x600

最近更新日期：2009/08/20

在 Linux 底下有相當多的壓縮指令可以運作喔！這些壓縮指令可以讓我們更方便從網路上面下載大型的檔案呢！此外，我們知道在 Linux 底下的副檔名是沒有什麼很特殊的意義的，不過，針對這些壓縮指令所做出來的壓縮檔，為了方便記憶，還是會有一些特殊的命名方式啦！就讓我們來看看吧！

1. 壓縮檔案的用途與技術
2. Linux 系統常見的壓縮指令
 - 2.1 compress
 - 2.2 gzip, zcat
 - 2.3 bzip2, bzip2, bzip2
3. 打包指令：tar
4. 完整備份工具：dump, restore
5. 光碟寫入工具
 - 5.1 mkisofs：建立映像檔
 - 5.2 cdrecord：光碟燒錄工具
6. 其他常見的壓縮與備份工具
 - 6.1 dd
 - 6.2 cpio
7. 重點回顧
8. 本章習題
9. 參考資料與延伸閱讀
10. 針對本文的建議：<http://phorum.vbird.org/viewtopic.php?t=23882>



壓縮檔案的用途與技術

你是否有過文件檔案太大，導致無法以一片軟碟將他複製完成的困擾？又，你是否有過，發現一個軟體裡面有好多檔案，這些檔案要將他複製與攜帶都很不方便的問題？還有，你是否有過要備份某些重要資料，偏偏這些資料量太大了，耗掉了你很多的磁碟空間呢？這個時候，那個好用的『檔案壓縮』技術可就派的上用場了！

因為這些比較大型的檔案透過所謂的檔案壓縮技術之後，可以將他的磁碟使用量降低，可以達到減低檔案容量的效果，此外，有的壓縮程式還可以進行容量限制，使一個大型檔案可以分割成為數個小型檔案，以方便軟碟片攜帶呢！

那麼什麼是『檔案壓縮』呢？我們來稍微談一談他的原理好了。目前我們使用的電腦系統中都是使用所謂的 bytes 單位來計量的！不過，事實上，電腦最小的計量單位應該是 bits 才對啊，此外，我們也知道 1

byte = 8 bits。但是如果今天我們只是記憶一個數字，亦即是 1 這個數字呢？他會如何記錄？假設一個 byte 可以看成底下的模樣：

□□□□□□□□

Tips:

由於 1 byte = 8 bits，所以每個 byte 當中會有 8 個空格，而每個空格可以是 0, 1，這裡僅是做為一個約略的介紹，更多的詳細資料請參考第零章的計算機概論吧！



由於我們記錄數字是 1，考慮電腦所謂的二進位喔，如此一來，1 會在最右邊佔據 1 個 bit，而其他的 7 個 bits 將會自動的被填上 0 囉！你看看，其實在這樣的例子中，那 7 個 bits 應該是『空的』才對！不過，為了要滿足目前我們的作業系統資料的存取，所以就將該資料轉為 byte 的型態來記錄了！而一些聰明的電腦工程師就利用一些複雜的計算方式，將這些沒有使用到的空間『丟』出來，以讓檔案佔用的空間變小！這就是壓縮的技術啦！

另外一種壓縮技術也很有趣，他是將重複的資料進行統計記錄的。舉例來說，如果你的資料為『1111....』共有 100 個 1 時，那麼壓縮技術會記錄為『100 個 1』而不是真的有 100 個 1 的位元存在！這樣也能夠精簡檔案記錄的容量呢！非常有趣吧！

簡單的說，你可以將他想成，其實檔案裡面有相當多的『空間』存在，並不是完全填滿的，而『壓縮』的技術就是將這些『空間』填滿，以讓整個檔案佔用的容量下降！不過，這些『壓縮過的檔案』並無法直接被我們的作業系統所使用的，因此，若要使用這些被壓縮過的檔案資料，則必須將他『還原』回來未壓縮前的模樣，那就是所謂的『解壓縮』囉！而至於壓縮前與壓縮後的檔案所佔用的磁碟空間大小，就可以被稱為是『壓縮比』囉！更多的技術文件或許你可以參考一下：

- RFC 1952 文件：<http://www.ietf.org/rfc/rfc1952.txt>
- 鳥哥站上的備份：http://linux.vbird.org/linux_basic/0240tarcompress/0240tarcompress_gzip.php

這個『壓縮』與『解壓縮』的動作有什麼好處呢？最大的好處就是壓縮過的檔案容量變小了，所以你的硬碟容量無形之中就可以容納更多的資料。此外，在一些網路資料的傳輸中，也會由於資料量的降低，好讓網路頻寬可以用來作更多的工作！而不是老是卡在一些大型的檔案傳輸上面呢！目前很多的 WWW 網站也是利用檔案壓縮的技術來進行資料的傳送，好讓網站頻寬的可利用率上升喔！

Tips:

上述的 WWW 網站壓縮技術蠻有趣的！他讓你網站上面『看的到的資料』在經過網路傳輸時，使用的是『壓縮過的資料』，等到這些壓縮過的資料到達你的電腦主機時，再進行解壓縮，由於目前的電腦運算速度相當的快速，因此其實在網頁瀏覽的時候，時間都是花在『資料的傳輸』上面，而不是 CPU 的運算啦！如此一來，由於壓縮過的資料量降低了，自然傳送的速度就會增快不少！



若你是一位軟體工程師，那麼相信你也會喜歡將你自己的軟體壓縮之後提供大家下載來使用，畢竟沒有人喜歡自己的網站天天都是頻寬滿載的吧？舉個例子來說，Linux 2.6.27.4 完整的核心大小約有 300 MB 左右，而由於核心主要多是 ASCII code 的純文字型態檔案，這種檔案的『多餘空間』最多了。而一個提供下載的壓縮過的 2.6.27.4 核心大約僅有 60MB 左右，差了幾倍呢？你可以自己算一算喔！

Linux 系統常見的壓縮指令：

在Linux的環境中，壓縮檔案的副檔名大多是：『*.tar, *.tar.gz, *.tgz, *.gz, *.Z, *.bz2』，為什麼會有這樣的副檔名呢？不是說 Linux 的副檔名沒有什麼作用嗎？

這是因為 Linux 支援的壓縮指令非常多，且不同的指令所用的壓縮技術並不相同，當然彼此之間可能就無法互通壓縮/解壓縮檔案囉。所以，當你下載到某個壓縮檔時，自然就需要知道該檔案是由哪種壓縮指令所製作出來的，好用來對照著解壓縮啊！也就是說，雖然 Linux 檔案的屬性基本上是與檔名沒有絕對關係的，但是為了幫助我們人類小小的腦袋瓜子，所以適當的副檔名還是必要的！底下我們就列出幾個常見的壓縮檔案副檔名吧：

```
*.Z      compress 程式壓縮的檔案；
*.gz     gzip  程式壓縮的檔案；
*.bz2    bzip2 程式壓縮的檔案；
*.tar     tar  程式打包的資料，並沒有壓縮過；
*.tar.gz  tar  程式打包的檔案，其中並且經過 gzip 的壓縮
*.tar.bz2 tar  程式打包的檔案，其中並且經過 bzip2 的壓縮
```

Linux上常見的壓縮指令就是 gzip 與 bzip2，至於 compress 已經退流行了。gzip 是由 [GNU 計畫](#)所開發出來的壓縮指令，該指令已經取代了 compress。後來 GNU 又開發出 bzip2 這個壓縮比更好的壓縮指令！不過，這些指令通常僅能針對一個檔案來壓縮與解壓縮，如此一來，每次壓縮與解壓縮都要一大堆檔案，豈不煩人？此時，那個所謂的『打包軟體, tar』就顯的很重要啦！

這個 tar 可以將很多檔案『打包』成為一個檔案！甚至是目錄也可以這麼玩。不過，單純的 tar 功能僅是『打包』而已，亦即是將很多檔案集結成為一個檔案，事實上，他並沒有提供壓縮的功能，後來，[GNU 計畫](#)中，將整個 tar 與壓縮的功能結合在一起，如此一來提供使用者更方便並且更強大的壓縮與打包功能！底下我們就來談一談這些在 Linux 底下基本的壓縮指令吧！

compress

compress這個壓縮指令是非常老舊的一款，大概只有在非常舊的 Unix 機器上面還會找到這個軟體。我們的 CentOS 預設並沒有安裝這個軟體到系統當中，所以想要瞭解這個軟體的使用時，請先安裝 [ncompress](#) 這個軟體。不過，由於 gzip 已經可以解開使用 compress 壓縮的檔案，因此，compress 可以不用學習啦！但是，如果你所在的環境還是有老舊的系統，那麼還是得要學一學就是了。好了，如果你有網路的話，那麼安裝其實很簡單喔！

```
[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          <==開始分析相依性
--> 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) <==最後分析所要安裝的軟體數
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 23 k
Is this ok [y/N]: y          <==這裡請按下 y 來確認安裝
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          <==這裡則是與數位簽章有關
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!

```

關於 yum 更詳細的用法我們會在後續的章節介紹，這裡僅是提供一個大概的用法而已。等你安裝好這個軟體後，接下來讓我們看看如何使用 compress 吧！

```

[root@www ~]# compress [-rcv] 檔案或目錄 <==這裡是壓縮
[root@www ~]# uncompress 檔案.Z          <==這裡是解壓縮
選項與參數：
-r  ：可以連同目錄下的檔案也同時給予壓縮呢！
-c  ：將壓縮資料輸出成為 standard output (輸出到螢幕)
-v  ：可以秀出壓縮後的檔案資訊以及壓縮過程中的一些檔名變化。

範例一：將 /etc/man.config 複製到 /tmp，並加以壓縮
[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 <==原有檔案
-rw-r--r-- 1 root root 2684 Nov 10 17:14 /tmp/man.config.Z <==經過壓縮的檔案！
```

不知道你有沒有發現，複製到 /tmp 的 man.config 不見了！因為被壓縮成為 man.config.Z 囉 也就是說，在預設的情況中，被 `compress` 壓縮的原始檔案會不見，而壓縮檔案會被建立起來，而且副檔名會是 *.Z。仔細看一下，檔案由原本的 4617bytes 降低到 2684bytes 左右，確實有減少一點啦！那麼如何解壓縮呢？

```
範例二：將剛剛的壓縮檔解開
[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
```

解壓縮直接用 `uncompress` 即可！解壓縮完畢後該檔案就自動的變回來了！不過，那個壓縮檔卻又不存在囉～ 這樣可以理解用法了嗎？那如果我想要保留原始檔案且又要建立壓縮檔呢？可以使用 -c 的語法！

```
範例三：將 man.config 壓縮成另外一個檔案來備份
[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
# 這個 -c 的選項比較有趣！他會將壓縮過程的資料輸出到螢幕上，而不是寫入成為
# *.Z 的壓縮檔。所以，我們可以透過資料流重導向的方法將資料輸出成為另一個檔名。
# 關於資料流重導向，我們會在第十一章 bash 詳細談論的啦！
```

再次強調，`compress` 已經很少人在使用了，因為這支程式無法解開 *.gz 的檔案，而 `gzip` 則可以解開 *.Z 的檔案，所以，如果你的 distribution 上面沒有 `compress` 的話，那就不要進行上面的練習囉！ ^_^

gzip, zcat

`gzip` 可以說是應用度最廣的壓縮指令了！目前 `gzip` 可以解開 `compress`, `zip` 與 `gzip` 等軟體所壓縮的檔案。至於 `gzip` 所建立的壓縮檔為 *.gz 的檔名喔！讓我們來看看這個指令的語法吧：

```
[root@www ~]# gzip [-cdtv#] 檔名
[root@www ~]# zcat 檔名.gz
選項與參數：
-c : 將壓縮的資料輸出到螢幕上，可透過資料流重導向來處理；
```

```
-d : 解壓縮的參數；
-t : 可以用來檢驗一個壓縮檔的一致性～看看檔案有無錯誤；
-v : 可以顯示出原檔案/壓縮檔案的壓縮比等資訊；
-# : 壓縮等級，-1 最快，但是壓縮比最差、-9 最慢，但是壓縮比最好！預設是 -6

範例一：將 /etc/man.config 複製到 /tmp，並且以 gzip 壓縮
[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壓縮比較佳
```

與 compress 類似的，當你使用 gzip 進行壓縮時，在預設的狀態下原本的檔案會被壓縮成為 .gz 的檔名，原始檔案就不再存在了。您也可以發現，由於 gzip 的壓縮比要比 compress 好的多，所以當然建議使用 gzip 啦！此外，使用 gzip 壓縮的檔案在 Windows 系統中，竟然可以被 WinRAR 這個軟體解壓縮呢！很好用吧！至於其他的用法如下：

```
範例二：由於 man.config 是文字檔，請將範例一的壓縮檔的內容讀出來！
[root@www tmp]# zcat man.config.gz
# 由於 man.config 這個原本的檔案是文字檔，因此我們可以嘗試使用 zcat 去讀取！
# 此時螢幕上會顯示 man.config.gz 解壓縮之後的檔案內容！

範例三：將範例一的檔案解壓縮
[root@www tmp]# gzip -d man.config.gz
# 不要使用 gunzip 這個指令，不好背！使用 gzip -d 來進行解壓縮！
# 與 gzip 相反，gzip -d 會將原本的 .gz 刪除，產生原本的 man.config 檔案。

範例四：將範例三解開的 man.config 用最佳的壓縮比壓縮，並保留原本的檔案
[root@www tmp]# gzip -9 -c man.config > man.config.gz
```

其實 gzip 的壓縮已經最佳化過了，所以雖然 gzip 提供 1~9 的壓縮等級，不過使用預設的 6 就非常好用了！因此上述的範例四可以不要加入那個 -9 的選項。範例四的重點在那個 -c 與 > 的使用囉！

cat 可以讀取純文字檔，那個 zcat 則可以讀取純文字檔被壓縮後的壓縮檔！由於 gzip 這個壓縮指令主要想要用來取代 compress 的，所以不但 compress 的壓縮檔案可以使用 gzip 來解開，同時 zcat 這個指令可以同時讀取 compress 與 gzip 的壓縮檔呦！

bzip2, bzipcat

若說 gzip 是為了取代 compress 並提供更好的壓縮比而成立的，那麼 bzip2 則是為了取代 gzip 並提供更佳的壓縮比而來的。bzip2 真是很不錯用的東西～這玩意的壓縮比竟然比 gzip 還要好～至於 bzip2 的用法幾乎與 gzip 相同！看看底下的用法吧！


```
[root@www ~]# bzip2 [-cdkzv#] 檔名
[root@www ~]# bzipcat 檔名.bz2
選項與參數：
-c : 將壓縮的過程產生的資料輸出到螢幕上！
-d : 解壓縮的參數
-k : 保留原始檔案，而不會刪除原始的檔案喔！
-z : 壓縮的參數
-v : 可以顯示出原檔案/壓縮檔案的壓縮比等資訊；
-# : 與 gzip 同樣的，都是在計算壓縮比的參數，-9 最佳，-1 最快！
```

範例一：將剛剛的 /tmp/man.config 以 bzip2 壓縮

```
[root@www tmp]# bzip2 -z man.config
# 此時 man.config 會變成 man.config.bz2 !
```

範例二：將範例一的檔案內容讀出來！

```
[root@www tmp]# bzipcat man.config.bz2
# 此時螢幕上會顯示 man.config.bz2 解壓縮之後的檔案內容！！
```

範例三：將範例一的檔案解壓縮

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

範例四：將範例三解開的 man.config 用最佳的壓縮比壓縮，並保留原本的檔案

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

使用 compress 副檔名自動建立為 .Z，使用 gzip 副檔名自動建立為 .gz。這裡的 bzip2 則是自動的將副檔名建置為 .bz2 囉！所以當我們使用具有壓縮功能的 bzip2 -z 時，那麼剛剛的 man.config 就會自動的變成了 man.config.bz2 這個檔名囉！

好了，那麼如果我想要讀取這個檔案的內容呢？是否一定要解開？當然不需要囉！可以使用簡便的 bzipcat 這個指令來讀取內容即可！例如上面的例子中，我們可以使用 bzipcat man.config.bz2 來讀取資料而不需要解開！此外，當你要解開一個壓縮檔時，這個檔案的名稱為 .bz, .bz2, .tbz, .tbz2 等等，那麼就可以嘗試使用 bzip2 來解看看啦！當然囉，也可以使用 bunzip2 這個指令來取代 bzip2 -d 囉。



打包指令：tar

前一小節談到的指令大多僅能針對單一檔案來進行壓縮，雖然 gzip 與 bzip2 也能夠針對目錄來進行壓縮，不過，這兩個指令對目錄的壓縮指的是『將目錄內的所有檔案 "分別" 進行壓縮』的動作！而不像在 Windows 的系統，可以使用類似 WinRAR 這一類的壓縮軟體來將好多資料『包成一個檔案』的樣式。

這種將多個檔案或目錄包成一個大檔案的指令功能，我們可以稱呼他是一種『打包指令』啦！那 Linux 有沒有這種打包指令呢？是有的！那就是鼎鼎大名的 tar 這個玩意兒了！tar 可以將多個目錄或檔案打包成一個大檔案，同時還可以透過 gzip/bzip2 的支援，將該檔案同時進行壓縮！更有趣的是，由於 tar 的使用太廣泛了，目前 Windows 的 WinRAR 也支援 .tar.gz 檔名的解壓縮呢！很不錯吧！所以底下我們就來玩一玩這個咚咚！



tar 的選項與參數非常的多！我們只講幾個常用的選項，更多選項您可以自行 `man tar` 查詢囉！

```
[root@www ~]# tar [-j|-z] [cv] [-f 建立的檔名] filename... <==打包與壓縮
[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 目錄] <==解壓縮
```

選項與參數：

- c : 建立打包檔案，可搭配 -v 來察看過程中被打包的檔名(filename)
- t : 察看打包檔案的內容含有哪些檔名，重點在察看『檔名』就是了；
- x : 解打包或解壓縮的功能，可以搭配 -C (大寫) 在特定目錄解開
特別留意的是，-c, -t, -x 不可同時出現在一串指令列中。
- j : 透過 bzip2 的支援進行壓縮/解壓縮：此時檔名最好為 *.tar.bz2
- z : 透過 gzip 的支援進行壓縮/解壓縮：此時檔名最好為 *.tar.gz
- v : 在壓縮/解壓縮的過程中，將正在處理的檔名顯示出來！
- f filename : -f 後面要立刻接要被處理的檔名！建議 -f 單獨寫一個選項囉！
- C 目錄 : 這個選項用在解壓縮，若要在特定目錄解壓縮，可以使用這個選項。

其他後續練習會使用到的選項介紹：

- p : 保留備份資料的原本權限與屬性，常用於備份(-c)重要的設定檔
- P : 保留絕對路徑，亦即允許備份資料中含有根目錄存在之意；
- exclude=FILE : 在壓縮的過程中，不要將 FILE 打包！

其實最簡單的使用 tar 就只要記憶底下的方式即可：

- 壓 縮：tar -jcv -f filename.tar.bz2 要被壓縮的檔案或目錄名稱
- 查 詢：tar -jtv -f filename.tar.bz2
- 解壓縮：tar -jxv -f filename.tar.bz2 -C 欲解壓縮的目錄

那個 filename.tar.bz2 是我們自己取的檔名，tar 並不會主動的產生建立的檔名喔！我們要自訂啦！所以副檔名就顯的很重要了！如果不加 [-j|-z] 的話，檔名最好取為 *.tar 即可。如果是 -j 選項，代表有 bzip2 的支援，因此檔名最好就取為 *.tar.bz2，因為 bzip2 會產生 .bz2 的副檔名之故！至於如果是加上了 -z 的 gzip 的支援，那檔名最好取為 *.tar.gz 喔！瞭解乎？

另外，由於『-f filename』是緊接在一起的，過去很多文章常會寫成『-jcvf filename』，這樣是對的，但由於選項的順序理論上是可以變換的，所以很多讀者會誤認為『-jvfc filename』也可以～事實上這樣會導致產生的檔名變成 c！因為 -fc 嘛！所以囉，建議您在學習 tar 時，將『-f filename』與其他選項獨立出來，會比較不容易發生問題。

閒話少說，讓我們來測試幾個常用的 tar 方法吧！

■ 使用 tar 加入 -j 或 -z 的參數備份 /etc/ 目錄

有事沒事備份一下 /etc 這個目錄是件好事！備份 /etc 最簡單的方法就是使用 tar 囉！讓我們來玩玩先：


```
[root@www ~]# tar -zpcv -f /root/etc.tar.gz /etc
tar: Removing leading '/' from member names <==注意這個警告訊息
/etc/
....中間省略....
/etc/esd.conf
/etc/crontab
# 由於加上 -v 這個選項，因此正在作用中的檔名就會顯示在螢幕上。
# 如果你可以翻到第一頁，會發現出現上面的錯誤訊息！底下會講解。
# 至於 -p 的選項，重點在於『保留原本檔案的權限與屬性』之意。

[root@www ~]# tar -jpcv -f /root/etc.tar.bz2 /etc
# 顯示的訊息會跟上面一模一樣囉！

[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
# 為什麼建議您使用 -j 這個選項？從上面的數值你可以知道了吧？^_^
```

由上述的練習，我們知道使用 bzip2 亦即 -j 這個選項來製作備份時，能夠得到比較好的壓縮比！如上表所示，由原本的 /etc/ (118MBytes) 下降到 8.7Mbytes 左右！至於加上『-p』這個選項的原因是為了保存原本檔案的權限與屬性！我們曾在第七章的 cp 指令介紹時談到權限與檔案類型(例如連結檔)對複製的不同影響。同樣的，在備份重要的系統資料時，這些原本檔案的權限需要做完整的備份比較好。此時 -p 這個選項就派的上用場了。接下來讓我們看看打包檔案內有什麼資料存在？

■ 查閱 tar 檔案的資料內容(可察看檔名)，與備份檔名有否根目錄的意義

要察看檔名非常的簡單！可以這樣做：

```
[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 這個選項時，詳細的檔案權限/屬性都會被列出來！如果只是想要知道檔名而已，那麼就將 -v 拿掉即可。從上面的資料我們可以發現一件很有趣的事情，那就是每個檔名都沒了根目錄了！這也是上一個練習中出現的那個警告訊息『tar: Removing leading '/' from member names(移除了檔名開頭的 '/')』所告知的情況！

那為什麼要拿掉根目錄呢？主要是為了安全！我們使用 tar 備份的資料可能會需要解壓縮回來使用，在 tar 所記錄的檔名(就是我們剛剛使用 tar -jtvf 所察看到的檔名)那就是解壓縮後的實際檔名。如果拿掉了根目錄，假設你將備份資料在 /tmp 解開，那麼解壓縮的檔名就會變成『/tmp/etc/xxx』。但『如果沒有拿掉根目錄，解壓縮後的檔名就會是絕對路徑，亦即解壓縮後的資料一定會被放置到 /etc/xxx 去！』如此一來，你的原本的 /etc/ 底下的資料，就會被備份資料所覆蓋過去了！

Tips:

你會說：『既然是備份資料，那麼還原回來也沒有什麼問題吧？』想像一個狀況，你備份的資料是一年前的舊版 CentOS 4.x，你只是想要瞭解一下過去的備份內容究竟有哪些資料而已，結果一解開該檔案，卻發現你目前新版的 CentOS 5.x 底下的 /etc 被舊版的備份資料覆蓋了！此時你該如何是好？所以囉，當然是拿掉根目錄比較安全一些的。



如果你確定你就是需要備份根目錄到 tar 的檔案中，那可以使用 -P (大寫) 這個選項，請看底下的例子分析：

```
範例：將檔名中的(根)目錄也備份下來，並察看一下備份檔的內容檔名
[root@www ~]# tar -jpcv -f /root/etc.and.root.tar.bz2 /etc
....中間過程省略....
[root@www ~]# tar -jtf /root/etc.and.root.tar.bz2
/etc/dbus-1/session.conf
/etc/esd.conf
/etc/crontab
# 這次查閱檔名不含 -v 選項，所以僅有檔名而已！沒有詳細屬性/權限等參數。
```

有發現不同點了吧？如果加上 -P 選項，那麼檔名內的根目錄就會存在喔！不過，鳥哥個人建議，還是不要加上 -P 這個選項來備份！畢竟很多時候，我們備份是為了要未來追蹤問題用的，倒不一定需要還原回原本的系統中！所以拿掉根目錄後，備份資料的應用會比較有彈性！也比較安全呢！

■ 將備份的資料解壓縮，並考慮特定目錄的解壓縮動作 (-C 選項的應用)

那如果想要解打包呢？很簡單的動作就是直接進行解打包嘛！

```
[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
....(後面省略)....
```

此時該打包檔案會在『本目錄下進行解壓縮』的動作！所以，你等一下就會在家目錄底下發現一個名為 etc 的目錄囉！所以囉，如果你想要將該檔案在 /tmp 底下解開，可以 cd /tmp 後，再下達上述的指令即可。不過，這樣好像很麻煩呢～有沒有更簡單的方法可以『指定欲解開的目錄』呢？有的，可以使用 -C 這個選項喔！舉例來說：

```
[root@www ~]# tar -jxv -f /root/etc.tar.bz2 -C /tmp
這樣一來，你就能夠將該檔案在不同的目錄解開囉！鳥哥個人是認為，這個 -C 的選項務必要記憶一下的(前面省略)，處理完畢後，請記得將這兩個目錄刪除一下呢！
drwxr-xr-x 105 root root    12288 Nov 11 04:02 etc
....(後面省略)....
```

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

再次強調，這個『rm -rf』是很危險的指令！下達時請務必確認一下後面接的檔名。我們要刪除的是 /root/etc 與 /tmp/etc，您可不要將 /etc/ 刪除掉了！系統會死掉的～ ^_^

■ 僅解開單一檔案的方法

剛剛上頭我們解壓縮都是將整個打包檔案的內容全部解開！想像一個情況，如果我只想要解開打包檔案內的其中一個檔案而已，那該如何做呢？很簡單的，你只要使用 -jtv 找到你要的檔名，然後將該檔名解開即可。我們用底下的例子來說明一下：

```
# 1. 先找到我們要的檔名，假設解開 shadow 檔案好了：
[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 <==這是我們要的！
# 先搜尋重要的檔名！其中那個 grep 是『擷取』關鍵字的功能！我們會在第三篇說明！
# 這裡您先有個概念即可！那個管線 | 配合 grep 可以擷取關鍵字的意思！

# 2. 將該檔案解開！語法與實際作法如下：
[root@www ~]# tar -jxv -f 打包檔.tar.bz2 待解開檔名
[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 <==呦喝！只有一個檔案啦！
# 很有趣！此時只會解開一個檔案而已！不過，重點是那個檔名！你要找到正確的檔名。
# 在本例中，你不能寫成 /etc/shadow！因為記錄在 etc.tar.bz2 內的檔名之故！
```

■ 打包某目錄，但不含該目錄下的某些檔案之作法

假設我們想要打包 /etc/ /root 這幾個重要的目錄，但卻不想要打包 /root/etc* 開頭的檔案，因為該檔案都是剛剛我們才建立的備份檔嘛！而且假設這個新的打包檔案要放置成為 /root/system.tar.bz2，當然這個檔案自己不要打包自己（因為這個檔案放置在 /root 底下啊！），此時我們可以透過 --exclude 的幫忙！那個 exclude 就是不包含的意思！所以你可以這樣做：

```
[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～其實你可以打成：『tar -jcv -f /root/system.tar.bz2 --exclude=/root/etc* --exclude=/root/system.tar.bz2 /etc /root』，如果想要兩行輸入時，最後面加上反斜線 (\) 並立刻

按下 [enter]，就能夠到第二行繼續輸入了。這個指令下達的方式我們會在第三章再仔細說明。透過這個 `--exclude="file"` 的動作，我們可以將幾個特殊的檔案或目錄移除在打包之列，讓打包的動作變的更簡便喔！^_^

另外，在新版的 tar 指令中，鳥哥發現原本的『`--exclude file`』似乎無法實際運作了！使用 `man tar` 明明有看到這個選項的說，但使用 `info tar` 才發現，選項功能已經變成了『`--exclude=file`』的模式！這個地方得要特別留意呢！

■ 僅備份比某個時刻還要新的檔案

某些情況下你會想要備份新的檔案而已，並不要備份舊檔案！此時 `--newer-mtime` 這個選項就粉重要啦！其實有兩個選項啦，一個是『`--newer`』另一個就是『`--newer-mtime`』，這兩個選項有何不同呢？我們在第七章的 touch 介紹中談到過三種不同的時間參數，當使用 `--newer` 時，表示後續的日期包含『`mtime` 與 `ctime`』，而 `--newer-mtime` 則僅是 `mtime` 而已！這樣知道了吧！^_^。那就讓我們來嘗試處理一下囉！

```
# 1. 先由 find 找出比 /etc/passwd 還要新的檔案
[root@www ~]# find /etc/passwd
....(過程省略)....
# 此時會顯示出比 /etc/passwd 這個檔案的 mtime 還要新的檔名，
# 這個結果在每部主機都不相同！您自行查閱自己的主機即可，不會跟鳥哥一樣！

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

# 2. 好了，那麼使用 tar 來進行打包吧！日期為上面看到的 2008/09/29
[root@www ~]# tar -jcv -f /root/etc.newer.then.passwd.tar.bz2 \
> --newer-mtime="2008/09/29" /etc/*
....(中間省略)....
/etc/smardt.conf      <==真的有的有備份的檔案
....(中間省略)....
/etc/yum.repos.d/     <==目錄都會被記錄下來！
tar: /etc/yum.repos.d/CentOS-Base.repo: file is unchanged; not dumped
# 最後行顯示的是『沒有被備份的』，亦即 not dumped 的意思！

# 3. 顯示出檔案即可
[root@www ~]# tar -jtv -f /root/etc.newer.then.passwd.tar.bz2 | \
> grep -v '/$'
# 透過這個指令可以呼叫出 tar.bz2 內的結尾非 / 的檔名！就是我們要的啦！
```

現在你知道這個指令的好用了吧！甚至可以進行差異檔案的記錄與備份呢～這樣子的備份就會顯的更容易囉！你可以這樣想像，如果我在一個月前才進行過一次完整的資料備份，那麼這個月想要備份時，當然可以僅備份上個月進行備份的那個時間點之後的更新的檔案即可！為什麼呢？因為原本的檔案已經有備份了嘛！幹嘛還要進行一次？只要備份新資料即可。這樣可以降低備份的容量啊！

■ 基本名稱：tarfile, tarball ？

另外值得一提的是，tar 打包出來的檔案有沒有進行壓縮所得到檔案稱呼不同喔！如果僅是打包而已，就是『tar -cv -f file.tar』而已，這個檔案我們稱呼為 tarfile。如果還有進行壓縮的支援，例如『tar -jcv -f file.tar.bz2』時，我們就稱呼為 tarball (tar 球?)！這只是一個基本的稱謂而已，不過很多書籍與網路都會使用到這個 tarball 的名稱！所以得要跟您介紹介紹。

此外，tar 除了可以將資料打包成為檔案之外，還能夠將檔案打包到某些特別的裝置去，舉例來說，磁帶機 (tape) 就是一個常見的例子。磁帶機由於是一次性讀取/寫入的裝置，因此我們不能夠使用類似 cp 等指令來複製的！那如果想要將 /home, /root, /etc 備份到磁帶機 (/dev/st0) 時，就可以使用：『tar -cv -f /dev/st0 /home /root /etc』，很簡單容易吧！磁帶機用在備份 (尤其是企業應用) 是很常見的工作喔！

■ 特殊應用：利用管線命令與資料流

在 tar 的使用中，有一種方式最特殊，那就是透過標準輸入輸出的資料流重導向 (standard input/standard output)，以及管線命令 (pipe) 的方式，將待處理的檔案一邊打包一邊解壓縮到目標目錄去。關於資料流重導向與管線命令更詳細的資料我們會在第十一章 bash 再跟大家介紹，底下先來看一個例子吧！

```
# 1. 將 /etc 整個目錄一邊打包一邊在 /tmp 解開
[root@www ~]# cd /tmp
[root@www tmp]# tar -cvf - /etc | tar -xvf -
# 這個動作有點像是 cp -r /etc /tmp 啦~依舊是有其用途的！
# 要注意的地方在於輸出檔變成 - 而輸入檔也變成 -，又有一個 | 存在~
# 這分別代表 standard output, standard input 與管線命令啦！
# 簡單的想法中，你可以將 - 想成是在記憶體中的一個裝置(緩衝區)。
# 更詳細的資料流與管線命令，請翻到 bash 章節囉！
```

在上面的例子中，我們想要『將 /etc 底下的資料直接 copy 到目前所在的路徑，也就是 /tmp 底下』，但是又覺得使用 cp -r 有點麻煩，那麼就直接以這個打包的方式來打包，其中，指令裡面的 - 就是表示那個被打包的檔案啦！由於我們不想要讓中間檔案存在，所以就以這一個方式來進行複製的行為啦！

■ 例題：系統備份範例

系統上有非常多的重要目錄需要進行備份，而且其實我們也不建議你將備份資料放置到 /root 目錄下！假設目前你已經知道重要的目錄有底下這幾個：

- /etc/ (設定檔)
- /home/ (使用者的家目錄)
- /var/spool/mail/ (系統中，所有帳號的郵件信箱)
- /var/spool/cron/ (所有帳號的工作排成設定檔)
- /root (系統管理員的家目錄)

然後我們也知道，由於第八章曾經做過的練習的關係，/home/loop* 不需要備份，而且 /root 底下的壓縮檔也不需要備份，另外假設你要將備份的資料放置到 /backups，並且該目錄僅有 root 有權限

進入！此外，每次備份的檔名都希望不相同，例如使用：backup-system-20091130.tar.bz2 之類的檔名來處理。那你該如何處理這個備份資料呢？(請先動手作看看，再來察看一下底下的參考解答！)

```
# 1. 先處理要放置備份資料的目錄與權限：
[root@www ~]# mkdir /backups
[root@www ~]# chmod 700 /backups
[root@www ~]# ll -d /backups
dwx----- 2 root root 4096 Nov 30 16:35 /backups

# 2. 假設今天是 2009/11/30，則建立備份的方式如下：
[root@www ~]# tar -jcv -f /backups/backup-system-20091130.tar.bz2 \
> --exclude=/root/* .bz2 --exclude=/root/*.gz --exclude=/home/loop* \
> /etc /home /var/spool/mail /var/spool/cron /root
....(過程省略)....

[root@www ~]# ll -h /backups/
-rw-r--r-- 1 root root 8.4M Nov 30 16:43 backup-system-20091130.tar.bz2
```

完整備份工具：dump

某些時刻你想要針對檔案系統進行備份或者是儲存的功能時，不能不談到這個 dump 指令！這玩意兒我們曾在前一章的 `/etc/fstab` 裡面稍微談過。其實這個指令除了能夠針對整個 filesystem 備份之外，也能夠僅針對目錄來備份喔！底下就讓我們來談一談這個指令的用法吧！

dump

其實 dump 的功能頗強，他除了可以備份整個檔案系統之外，還可以制定等級喔！什麼意思啊！假設你的 `/home` 是獨立的一個檔案系統，那你第一次進行過 dump 後，再進行第二次 dump 時，你可以指定不同的備份等級，假如指定等級為 1 時，此時新備份的資料只會記錄與第一次備份所有差異的檔案而已。看不懂嗎？沒關係！我們用一張簡圖來說明。

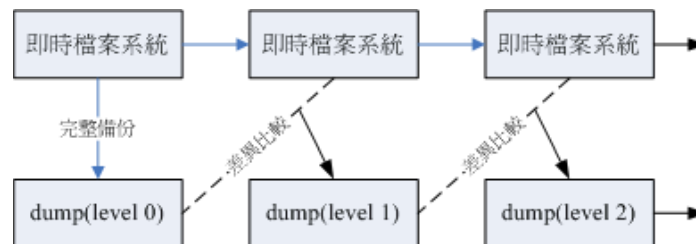


圖 4.1.1、dump 運作的等級 (level)

如上圖所示，上方的『即時檔案系統』是一直隨著時間而變化的資料，例如在 `/home` 裡面的檔案資料會一直變化一樣。而底下的方塊則是 dump 備份起來的資料，第一次備份時使用的是 level 0，這個等級也是完整的備份啦！等到第二次備份時，即時檔案系統內的資料已經與 level 0 不一樣了，而 level 1 僅只是比較目前的檔案系統與 level 0 之間的差異後，備份有變化過的檔案而已。至於 level 2 則是與 level 1 進行比較啦！這樣瞭解呼？

雖然 dump 支援整個檔案系統或者是單一各別目錄，但是對於目錄的支援是比較不足的，這也是 dump 的限制所在。簡單的說，如果想要備份的資料如下時，則有不同的限制情況：

- 當待備份的資料為單一檔案系統：

如果是單一檔案系統 (filesystem)，那麼該檔案系統可以使用完整的 dump 功能，包括利用 0~9 的數個 level 來備份，同時，備份時可以使用掛載點或者是裝置檔名 (例如 /dev/sda5 之類的裝置檔名) 來進行備份！
- 待備份的資料只是目錄，並非單一檔案系統：

例如你僅想要備份 /home/someone/，但是該目錄並非獨立的檔案系統時。此時備份就有限制啦！包括：

 - 所有的備份資料都必須要在該目錄 (本例為：/home/someone/) 底下；
 - 且僅能使用 level 0，亦即僅支援完整備份而已；
 - 不支援 -u 選項，亦即無法建立 /etc/dumpdates 這個各別 level 備份的時間記錄檔；

dump 的選項雖然非常的繁複，不過如果只是想要簡單的操作時，您只要記得底下的幾個選項就很夠用了！

```
[root@www ~]# dump [-Suvj] [-level] [-f 備份檔] 待備份資料
[root@www ~]# dump -W
選項與參數：
-S   : 僅列出後面的待備份資料需要多少磁碟空間才能夠備份完畢；
-u   : 將這次 dump 的時間記錄到 /etc/dumpdates 檔案中；
-v   : 將 dump 的檔案過程顯示出來；
-j   : 加入 bzip2 的支援！將資料進行壓縮，預設 bzip2 壓縮等級為 2
-level: 就是我們談到的等級，從 -0 ~ -9 共十個等級；
-f   : 有點類似 tar 啦！後面接產生的檔案，亦可接例如 /dev/st0 裝置檔名等
-W   : 列出在 /etc/fstab 裡面的具有 dump 設定的 partition 是否有備份過？
```

■ 用 dump 備份完整的檔案系統

現在就讓我們來做幾個範例吧！假如你要將系統的最小的檔案系統提出來進行備份，那該如何進行呢？

```
# 1. 先找出系統中最小的那個檔案系統，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 <==看起來最小的就是他啦！
tmpfs            363M    0  363M   0% /dev/shm

# 2. 先測試一下，如果要備份此檔案系統，需多少容量？
[root@www ~]# dump -S /dev/hdc1
5630976 <==注意一下，這個單位是 bytes，所以差不多是 5.6MBytes。
```

```
# 3. 將完整備份的檔名記錄成為 /root/boot.dump，同時更新記錄檔：
[root@www ~]# dump -Ou -f /root/boot.dump /boot
DUMP: Date of this level 0 dump: Tue Dec 2 02:53:45 2008 <==記錄等級與備份時間
DUMP: Dumping /dev/hdc1 (/boot) to /root/boot.dump <==dump的來源與目標
DUMP: Label: /boot <==檔案系統的 label
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files] <==開始進行檔案對應
DUMP: mapping (Pass II) [directories]
DUMP: estimated 5499 blocks. <==評估整體block數量
DUMP: Volume 1 started with block 1 at: Tue Dec 2 02:53:46 2008
DUMP: dumping (Pass III) [directories] <==開始 dump 工作
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /root/boot.dump <==結束寫入備份檔
DUMP: Volume 1 completed at: Tue Dec 2 02:53:47 2008
DUMP: Volume 1 5550 blocks (5.42MB) <==最終備份資料容量
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

# 在指令的下達方面，dump 後面接 /boot 或 /dev/hdc1 都可以的！
# 而執行 dump 的過程中會出現如上的一些訊息，您可以自行仔細的觀察！

[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
# 由於加上 -u 的選項，因此 /etc/dumpdates 該檔案的內容會被更新！注意，
# 這個檔案僅有在 dump 完整的檔案系統時才有支援主動更新的功能。

# 4. 觀察一下系統主動建立的記錄檔：
[root@www ~]# cat /etc/dumpdates
/dev/hdc1 0 Tue Dec 2 02:53:47 2008 +0800
[檔案系統][等級][ ctime 的時間 ]
```

這樣很簡單的就建立起來 /root/boot.dump 檔案，該檔案將整個 /boot/ 檔案系統都備份下來了！並且將備份的時間寫入 /etc/dumpdates 檔案中，準備讓下次備份時可以作為一個參考依據。現在讓我們來進行一個測試，檢查看看能否真的建立 level 1 的備份呢？

```
# 0. 看一下有沒有任何檔案系統被 dump 過的資料？
[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
# 如上列的結果，該結果會提出 /etc/fstab 裡面第五欄位設定有需要 dump 的
# partition，然後與 /etc/dumpdates 進行比對，可以得到上面的結果啦！
# 尤其是第三行，可以顯示我們曾經對 /dev/hdc1 進行過 dump 的備份動作喔！
```

```
# 1. 先惡搞一下，建立一個大約 10 MB 的檔案在 /boot 內：
[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. 開始建立差異備份檔，此時我們使用 level 1 吧：
[root@www ~]# dump -1u -f /root/boot.dump.1 /boot
....(中間省略)....

[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
# 看看檔案大小，豈不是就是剛剛我們所建立的那個大檔案的容量嗎？ ^_^

# 3. 最後再看一下是否有記錄 level 1 備份的時間點呢？
[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
....(中間省略)....
```

透過這個簡單的方式，我們就能夠僅備份差異檔案的部分囉！底下再來看看針對單一目錄的 dump 用途！

■ 用 dump 備份非檔案系統，亦即單一目錄的方法

現在讓我們來處理一下 /etc 的 dump 備份吧！因為 /etc 並非單一檔案系統，他只是個目錄而已。所以依據限制的說明，-u, level 1~9 都是不適用的。我們只能夠使用 level 0 的完整備份將 /etc 給他 dump 下來。因此作法就變的很簡單了！

```
# 讓我們將 /etc 整個目錄透過 dump 進行備份，且含壓縮功能
[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 (bzip)
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
```

上面特殊字體的部分顯示：原本有 124680kb 的容量，被壓縮成為 18752kb，
整個壓縮比為 6.649:1，還可以的壓縮情況啦！

一般來說 dump 不會使用包含壓縮的功能，不過如果你想要將備份的空間降低的話，那個 -j 的選項是可以使用的。加上 -j 之後你的 dump 成果會使用較少的硬碟容量啦！如上述的情況來看，檔案容量由原本的 128MB 左右下滑到 18MB 左右，當然可以節省備份空間囉！

restore

備份檔就是在急用時可以回復系統的重要資料，所以有備份當然就得要學學如何復原了！dump 的復原使用的是 restore 這個指令！這個指令的選項也非常的多～您可以自行 man restore 瞧瞧！鳥哥在這裡僅作個簡單的介紹囉！

```
[root@www ~]# restore -t [-f dumpfile] [-h]          <==用來察看 dump 檔
[root@www ~]# restore -C [-f dumpfile] [-D 掛載點] <==比較dump與實際檔案
[root@www ~]# restore -i [-f dumpfile]              <==進入互動模式
[root@www ~]# restore -r [-f dumpfile]              <==還原整個檔案系統
```

選項與參數：

相關的各種模式，各種模式無法混用喔！例如不可以寫 -tC 啦！

-t：此模式用在察看 dump 起來的備份檔中含有什麼重要資料！類似 tar -t 功能；

-C：此模式可以將 dump 內的資料拿出來跟實際的檔案系統做比較，
最終會列出『在 dump 檔案內有記錄的，且目前檔案系統不一樣』的檔案；

-i：進入互動模式，可以僅還原部分檔案，用在 dump 目錄時的還原！

-r：將整個 filesystem 還原的一種模式，用在還原針對檔案系統的 dump 備份；

其他較常用到的選項功能：

-h：察看完整備份資料中的 inode 與檔案系統 label 等資訊

-f：後面就接你要處理的那個 dump 檔案囉！

-D：與 -C 進行搭配，可以查出後面接的掛載點與 dump 內有不同的檔案！

■ 用 restore 觀察 dump 後的備份資料內容

要找出 dump 的內容就使用 restore -t 來查閱即可！例如我們將 boot.dump 的檔案內容捉出來看看！

```
[root@www ~]# restore -t -f /root/boot.dump
Dump date: Tue Dec 2 02:53:45 2008          <==說明備份的日期
Dumped from: the epoch
Level 0 dump of /boot on www.vbird.tsai:/dev/hdc1 <==說明 level 狀態
```

```

Label: /boot                                     <==說明該 filesystem 的表頭！
    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.                       <==加註說明資料有壓縮
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 <==是目錄！
Label: /1
    2      .
  1912545  ./etc
  1912549  ./etc/rpm
  1912550  ./etc/rpm/platform
....底下省略....

```

這個查閱的資料其實顯示出的是檔名與原始檔案的 inode 狀態，所以我們可以說，dump 會參考 inode 的記錄哩！透過這個查詢我們也能知道 dump 的內容為何呢！再來查一查如何還原吧！

■ 比較差異並且還原整個檔案系統

為什麼 dump 可以進行累積備份呢？就是因為它具有可以查詢檔案系統與備份檔案之間的差異，並且將分析到的差異資料進行備份的緣故。所以我們先來看看，如何查詢有變動過的資訊呢？你可以使用如下的方法檢驗：

```

# 0. 先嘗試變更檔案系統的內容：
[root@www ~]# cd /boot
[root@www boot]# mv config-2.6.18-128.el5 config-2.6.18-128.el5-back

# 1. 看使進行檔案系統與備份檔案之間的差異！
[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-2.6.18-128.el5: No such file or directory
Some files were modified! 1 compare errors
# 看到上面的特殊字體了吧！那就是有差異的部分！總共有一個檔案被變更！
# 我們剛剛確實有更動過該檔案，嘿嘿！這樣是否能瞭解？！

# 2. 將檔案系統改回來啊！
[root@www boot]# mv config-2.6.18-128.el5-back config-2.6.18-128.el5
[root@www boot]# cd /root

```

如同上面的動作，透過曾經備份過的資訊，也可以找到與目前實際檔案系統中有差異的資料呢！如果你不想要進行累積備份，但也能透過這個動作找出最近這一陣子有變動過的檔案說！瞭解乎？那如何還原呢？由於 dump 是記錄整個檔案系統的，因此還原時你也應該要給予一個全新的檔案系統才行。因此底下我們先建立一個檔案系統，然後再來還原吧！

```
# 1. 先建立一個新的 partition 來使用，假設我們需要的是 150M 的容量
[root@www ~]# fdisk /dev/hdc
Command (m for help): n
First cylinder (2335-5005, default 2335): <==這裡按 Enter
Using default value 2335
Last cylinder or +size or +sizeM or +sizeK (2335-5005, default 5005): +150M
Command (m for help): p
....中間省略....
/dev/hdc8      2335      2353      152586  83  Linux

Command (m for help): w

[root@www ~]# partprobe <==很重要的動作！別忘記！
# 這樣就能夠建立一個 /dev/hdc8 的 partition，然後繼續格式化吧！

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

# 2. 開始進行還原的動作！請您務必到新檔案系統的掛載點底下去！
[root@www ~]# cd /mnt
[root@www mnt]# restore -r -f /root/boot.dump
restore: ./lost+found: File exists
```

由於我們是備份整個檔案系統，因此你也可以建置一個全新的檔案系統 (partition) 來進行還原的動作！整個還原的動作也不難，如上表最後一個指令，就是將[備份檔案中的資料還原到本目錄下](#)。你必須要變更目錄到掛載點所在的那個目錄才行啊！這樣還原的檔案才不會跑錯地方！如果你還想要將 level 1 的 /root/boot.dump.1 那個檔案的內容也還原的話，那就繼續使用『restore -r -f /root/boot.dump.1』去還原吧！

■ 僅還原部分檔案的 restore 互動模式

某些時候你只是要將備份檔的某個內容捉出來而已，並不要全部解開，那該如何是好？此時你可以進入 restore 的互動模式 (interactive mode)。在底下我們使用 etc.dump 來進行範例說明。假如你要將 etc.dump 內的 passwd 與 shadow 捉出來而已，該如何進行呢？

```
[root@www ~]# cd /mnt
[root@www mnt]# restore -i -f /root/etc.dump
restore >
# 此時你就已經進入 restore 的互動模式畫面中！要注意的是：
# 你目前已經在 etc.dump 這個檔案內了！所有的動作都是在 etc.dump 內！
```



```
restore > help
Available commands are:
  ls [arg] - list directory          <==列出 etc.dump 內的檔案或目錄
  cd arg - change directory          <==在 etc.dump 內變更目錄
  pwd - print current directory      <==列出在 etc.dump 內的路徑檔名
  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

# 上面三個指令是重點！各指令的功能為：
# add file   : 將 file 加入等一下要解壓縮的檔案列表中
# delete file : 將 file 移除出解壓縮的列表，並非刪除 etc.dump 內的檔案！別誤會！^_^
# extract    : 開始將剛剛選擇的檔案列表解壓縮了去！
  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/ <==會顯示出在 etc.dump 內主要的目錄，因為我們備份 /etc，所以檔名為此！

restore > cd etc          <==在 etc.dump 內變換路徑到 etc 目錄下
restore > pwd             <==列出本目錄的檔名為？
/etc
restore > ls passwd shadow group <==看看，真的有這三個檔案喔！
passwd
shadow
group
restore > add passwd shadow group <==加入解壓縮列表
restore > delete group      <==加錯了！將 group 移除解壓縮列表
restore > ls passwd shadow group
*passwd <==有要被解壓縮的，檔名之前會出現 * 的符號呢！
*shadow
group
restore > extract          <==開始進行解壓縮去！
You have not read any volumes yet. <==這裡會詢問你需要的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 <==只有一個 volume
set owner/mode for '.'? [yn] n <==不需要修改權限

restore > quit             <==離開 restore 的功能

[root@www ~]# ll -d etc
drwxr-xr-x 2 root root 1024 Dec 15 17:49 etc <==解壓縮後，所建立出來的目錄啦！
[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
```

透過互動式的 restore 功能，可以讓你將備份的資料取出一部份，而不必全部都得解壓縮才能夠取得你想要的檔案資料。而 restore 內的 add 除了可以增加檔案外，也能夠增加整個備份的『目錄』喔！

還不錯玩吧！趕緊測試看看先！ ^_^

光碟寫入工具

某些時刻你可能會希望將系統上最重要的資料給他備份出來，雖然目前隨身碟已經有夠便宜，你可以使用這玩意兒來備份。不過某些重要的、需要重複備份的資料(可能具有時間特性)，你可能會需要使用類似 DVD 之類的儲存媒體來備份出來！舉例來說，你的系統設定檔或者是討論區的資料庫檔案(變動性非常的頻繁)。雖然 Linux 圖形介面已經有不少的燒錄軟體可用，但有時如果你希望系統自動在某些時刻幫你主動的進行燒錄時，那麼文字介面的燒錄行為就有幫助啦！

那麼文字模式的燒錄行為要怎麼處理呢？通常的作法是這樣的：

- 先將所需要備份的資料建置成為一個映像檔(iso)，利用 mkisofs 指令來處理；
- 將該映像檔燒錄至光碟或 DVD 當中，利用 cdrecord 指令來處理。

底下我們就分別來談談這兩個指令的用法吧！

mkisofs：建立映像檔

我們從 FTP 站捉下來的 Linux 映像檔 (不管是 CD 還是 DVD) 都得要繼續燒錄成為實體的光碟/DVD 後，才能夠進一步的使用，包括安裝或更新你的 Linux 啦！同樣的道理，你想要利用燒錄機將你的資料燒錄到 DVD 時，也得要先將你的資料包成一個映像檔，這樣才能夠寫入 DVD 片中。而將你的資料包成一個映像檔的方式就透過 mkisofs 這個指令即可。mkisofs 的使用方式如下：

```
[root@www ~]# mkisofs [-o 映像檔] [-rv] [-m file] 待備份檔案.. [-V vol] \  
> -graft-point isodir=systemdir ...  
選項與參數：  
-o：後面接你想要產生的那個映像檔檔名。  
-r：透過 Rock Ridge 產生支援 Unix/Linux 的檔案資料，可記錄較多的資訊；  
-v：顯示建置 ISO 檔案的過程  
-m file：-m 為排除檔案 (exclude) 的意思，後面的檔案不備份到映像檔中  
-V vol：建立 Volume，有點像 Windows 在檔案總管內看到的 CD title 的東西  
-graft-point：graft 有轉嫁或移植的意思，相關資料在底下文章內說明。
```

其實 mkisofs 有非常多好用的選項可以選擇，不過如果我們只是想要製作資料光碟時，上述的選項也就夠用了。光碟的格式一般稱為 iso9660，這種格式一般僅支援舊版的 DOS 檔名，亦即檔名只能以 8.3 (檔名8個字元，副檔名3個字元) 的方式存在。如果加上 -r 的選項之後，那麼檔案資訊能夠被記錄的比較完整，可包括UID/GID與權限等等！所以，記得加這個 -r 的選項。

此外，一般預設的情況下，所有要被加到映像檔中的檔案都會被放置到映像檔中的根目錄，如此一來可能會造成燒錄後的檔案分類不易的情況。所以，你可以使用 -graft-point 這個選項，當你使用這個選項之後，可以利用如下的方法來定義位於映像檔中的目錄，例如：

- 映像檔中的目錄所在=實際 Linux 檔案系統的目錄所在
- /movies/= /srv/movies/ (在 Linux 的 /srv/movies 內的檔案，加至映像檔中的 /movies/ 目

錄)

- `/linux/etc=/etc` (將 Linux 中的 `/etc/` 內的所有資料備份到映像檔中的 `/linux/etc/` 目錄中)

我們透過一個簡單的範例來說明一下吧。如果你想要將 `/root`, `/home`, `/etc` 等目錄內的資料通通燒錄起來的話，先得要處理一下映像檔，我們先不使用 `-graft-point` 的選項來處理這個映像檔試看看：

```
[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
....中間省略....
 97.01% done, estimate finish Tue Dec 16 17:07:14 2008 <==顯示百分比
 98.69% done, estimate finish Tue Dec 16 17:07:15 2008
Total translation table size: 0
Total rockridge attributes bytes: 9840 <==額外記錄屬性所耗用之容量
Total directory bytes: 55296 <==目錄佔用容量
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 <==就是這玩意兒！
[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！一堆資料都放置在一起！包括有的沒有的目錄與檔案等等！

[root@www ~]# umount /mnt
```

由上面的範例我們可以看到，三個目錄 (`/root`, `/home`, `/etc`) 的資料通通放置到了映像檔的最頂層目錄中！真是不方便～尤其由於 `/root/etc` 的存在，導致那個 `/etc` 的資料似乎沒有被包含進來的樣子！真不合理～而且還有 `lost+found` 的目錄存在！真是超不喜歡的！此時我們可以使用 `-graft-point` 來處理囉！

```
[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
```

```
# 上面的指令會建立一個大檔案，期中 -graft-point 後面接的就是我們要備份的資料。
# 必須要注意的是那個等號的兩邊，等號左邊是在映像檔內的目錄，右側則是實際的資料。
```

```
[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！資料是分門別類的在各個目錄中喔這樣瞭解乎？最後將資料卸載一下：

[root@www ~]# umount /mnt
```

其實鳥哥一直覺得很奇怪，怎麼我的資料會這麼大(600多MB)？原來是 /home 裡面在第八章的時候，練習時多了一個 /home/loopdev 的大檔案！所以在重新製作一次 iso 檔，並多加一個『-m /home/loopdev』來排除該檔案的備份，最終的檔案則僅有 176MB 囉！還好還好！^_^！接下來讓我們處理燒錄的動作了吧！

cdrecord : 光碟燒錄工具

我們是透過 cdrecord 這個指令來進行文字介面的燒錄行為，這個指令常見的選項有底下數個：

```
[root@www ~]# cdrecord -scanbus dev=ATA <==查詢燒錄機位置
[root@www ~]# cdrecord -v dev=ATA:x,y,z blank=[fast|all] <==抹除重複讀寫片
[root@www ~]# cdrecord -v dev=ATA:x,y,z -format <==格式化DVD+RW
[root@www ~]# cdrecord -v dev=ATA:x,y,z [可用選項功能] file.iso
```

選項與參數：

- scanbus : 用在掃描磁碟匯流排並找出可用的燒錄機，後續的裝置為 ATA 介面
- v : 在 cdrecord 運作的過程中，顯示過程而已。
- dev=ATA:x,y,z : 後續的 x, y, z 為你系統上燒錄機所在的位置，非常重要！
- blank=[fast|all] : blank 為抹除可重複寫入的CD/DVD-RW，使用fast較快，all較完整
- format : 僅針對 DVD+RW 這種格式的 DVD 而已；

[可用選項功能] 主要是寫入 CD/DVD 時可使用的選項，常見的選項包括有：

- data : 指定後面的檔案以資料格式寫入，不是以 CD 音軌(-audio)方式寫入！
- speed=X : 指定燒錄速度，例如CD可用 speed=40 為40倍數，DVD則可用 speed=4 之類
- eject : 指定燒錄完畢後自動退出光碟
- fs=Ym : 指定多少緩衝記憶體，可用在將映像檔先暫存至緩衝記憶體。預設為 4m，一般建議可增加到 8m，不過，還是得視你的燒錄機而定。

針對 DVD 的選項功能：

- driveropts=burnfree : 打開 Buffer Underrun Free 模式的寫入功能
- sao : 支援 DVD-RW 的格式

■ 偵測你的燒錄機所在位置：

文字模式的燒錄確實是比較麻煩的，因為沒有所見即所得的環境嘛！要燒錄首先就得要找到燒錄機才行！而由於早期的燒錄機都是使用 SCSI 介面，因此查詢燒錄機的方法就得要配合著 SCSI 介面的認定來處理了。查詢燒錄機的方式為：

```
[root@www ~]# cdrecord -scanbus dev=ATA
Cdrecord-Clone 2.01 (cpu-pc-linux-gnu) Copyright (C) 1995-2004 J?rg Schilling
....中間省略....
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` 就能夠找到正確的燒錄機！由於目前個人電腦上最常使用的磁碟機介面為 IDE 與 SATA，這兩種介面都能夠使用 `dev=ATA` 這種模式來查詢，因此上述的指令得要背一下啦！另外，在查詢的結果當中可以發現有一台燒錄機，其中也顯示出這台燒錄機的型號，而最重要的就是上表中有底線的那三個數字！那三個數字就是代表這台燒錄機的位置！以上表的例子中，這部燒錄機的位置在『ATA:1,1,0』這個地方喔！

好了，那麼現在要如何將 `/tmp/system.img` 燒錄到 CD/DVD 裡面去呢？鳥哥這裡先以 CD 為例，鳥哥用的是 CD-RW (可重複讀寫) 的光碟片，說實在話，雖然 CD-RW 或 DVD-RW 比較貴一點，不過至少可以重複利用，對環境的衝擊比較小啦！建議大家使用可重複讀寫的片子。由於 CD-RW 可能要先進行抹除的工作(將原本裡面的資料刪除)然後才能寫入，因此，底下我們先來看看如何抹除一片 CD/DVD 的方法，然後直接寫入光碟吧！

Tips:

由於 CD/DVD 都是使用 `cdrecord` 這個指令，因此不論是 CD 還是 DVD 片，下達指令的方法都差不多！不過，DVD 的寫入需要額外的 `driveropts=burnfree` 或 `-dao` 等選項的輔助才行。另外，CD 片有 CD-R(一次寫入) 與 CD-RW(重複寫入)，至於 DVD 則主要有兩種格式，分別是 DVD-R 及 DVD+R 兩種格式。如果是可重複讀寫的則為：DVD-RW, DVD+RW。除了 DVD+RW 的抹除方法可能不太一樣之外，其他寫入的方式則是一樣的。



■ 進行 CD 的燒錄動作：

```
# 0. 先抹除光碟的原始內容：(非可重複讀寫則可略過此步驟)
[root@www ~]# cdrecord -v dev=ATA:1,1,0 blank=fast
# 中間會跑出一堆訊息告訴你抹除的進度，而且會有 10 秒鐘的時間等待你的取消！
# 可以避免『手滑』的情況！^_^

# 1. 開始燒錄：
[root@www ~]# cdrecord -v dev=ATA:1,1,0 fs=8m -dummy -data \
> /tmp/system.img
....中間省略....
Track 01: 168 of 176 MB written (fifo 100%) [buf 100%] 10.5x. <==顯示百分比
# 上面會顯示進度，還有 10.5x 代表目前的燒錄速度！
cdrecord: fifo had 2919 puts and 2919 gets.
```

```
cdrecord: fifo was 0 times empty and 2776 times full, min fill was 97%.

# 2. 燒錄完畢後，測試掛載一下，檢驗內容：
[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！確實是光碟內容！

[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    <==不要忘了卸載
```

事實上如果你忘記抹除可寫入光碟時，其實 cdrecord 很聰明的會主動的幫你抹除啦！因此上面的資訊你只要記得燒錄的功能即可。特別注意 -data 那個選項！因為如果沒有加上 -data 的選項時，預設資料會以音軌格式寫入光碟中，所以最好能夠加上 -data 這個選項囉！上述的功能是針對 CD，底下我們使用一片可重複讀寫的 DVD-RW 來測試一下寫入的功能！

■ 進行 DVD-RW 的燒錄動作：

```
# 0. 同樣的，先來抹除一下原本的內容：
[root@www ~]# cdrecord -v dev=ATA:1,1,0 blank=fast

# 1. 開始寫入 DVD，請注意，有些選項與 CD 並不相同了喔！
[root@www ~]# cdrecord -v dev=ATA:1,1,0 fs=8m -data -sao \
> driveropts=burnfree /tmp/system.img

# 2. 同樣的，來給他測試測試！
[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
```

整體指令沒有差很多啦！只是 CD-RW 會自動抹除，但 DVD-RW 似乎得要自己手動某除才行！並不會主動進入自動抹除的功能！害鳥哥重新測試過好幾次～傷腦筋～^_^！好啦！現在你就知道如何將你的資料燒錄出來啦！

如果你的 Linux 是用來做為伺服器之用的話，那麼無時無刻的去想『如何備份重要資料』是相當重要的！關於備份我們會在第五篇再仔細的談一談，這裡你要會使用這些工具即可！



其他常見的壓縮與備份工具

還有一些很好用的工具得要跟大家介紹介紹，尤其是 dd 這個玩意兒呢！



我們在第八章當中的特殊 loop 裝置掛載時使用過 dd 這個指令對吧？不過，這個指令可不只是製作一個檔案而已喔～這個 dd 指令最大的功效，鳥哥認為，應該是在於『備份』啊！因為 dd 可以讀取磁碟裝置的內容(幾乎是直接讀取磁區"sector")，然後將整個裝置備份成一個檔案呢！真的是相當的好用啊～ dd 的用途有很多啦～但是我們僅講一些比較重要的選項，如下：

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

選項與參數：

if ：就是 input file 囉～也可以是裝置喔！

of ：就是 output file 喔～也可以是裝置；

bs ：規劃的一個 block 的大小，若未指定則預設是 512 bytes(一個 sector 的大小)

count：多少個 bs 的意思。

範例一：將 /etc/passwd 備份到 /tmp/passwd.back 當中

```
[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
```

仔細的看一下，我的 /etc/passwd 檔案大小為 1945 bytes，因為我沒有設定 bs，

所以預設是 512 bytes 為一個單位，因此，上面那個 3+1 表示有 3 個完整的

512 bytes，以及未滿 512 bytes 的另一個 block 的意思啦！

事實上，感覺好像是 cp 這個指令啦～

範例二：將自己的磁碟之第一個磁區備份下來

```
[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

第一個磁區內含有 MBR 與 partition table，透過這個動作，

我們可以一口氣將這個磁碟的 MBR 與 partition table 進行備份哩！

範例三：找出你系統最小的那個分割槽，並且將他備份下來：

```
[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
# 等於是將整個 /dev/hdc1 通通捉下來的意思~如果要還原呢?就反向回去!
# dd if=/tmp/boot.whole.disk of=/dev/hdc1 即可!非常簡單吧!
# 簡單的說,如果想要整個硬碟備份的話,就類似 Norton 的 ghost 軟體一般,
# 由 disk 到 disk, 嘿嘿~利用 dd 就可以啦~厲害厲害!
```

你可以說, tar 可以用來備份關鍵資料,而 dd 則可以用來備份整顆 partition 或 整顆 disk, 很不錯啊~不過,如果要將資料填回到 filesystem 當中, 可能需要考慮到原本的 filesystem 才能成功啊!讓我們來完成底下的例題試看看:

例題:

你想要將你的 /dev/hdc1 進行完整的複製到另一個 partition 上,請使用你的系統上面未分割完畢的容量再建立一個與 /dev/hdc1 差不多大小的分割槽 (只能比 /dev/hdc1 大,不能比他小!),然後將之進行完整的複製 (包括需要複製 boot sector 的區塊)。

答:

由於需要複製 boot sector 的區塊,所以使用 cp 或者是 tar 這種指令是無法達成需求的!此時那個 dd 就派的上用場了。你可以這樣做:

```
# 1. 先進行分割的動作
[root@www ~]# fdisk -l /dev/hdc
   Device Boot    Start    End  Blocks  Id System
/dev/hdc1    *         1     13    104391  83  Linux
# 上面鳥哥僅擷取重要的資料而已!我們可以看到 /dev/hdc1 僅有 13 個磁柱

[root@www ~]# fdisk /dev/hdc
Command (m for help): n
First cylinder (2354-5005, default 2354): 這裡按 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
# 為什麼要使用 2366 呢?因為 /dev/hdc1 使用 13 個磁柱,因此新的 partition
# 我們也給她 13 個磁柱,因此 2354 + 13 - 1 = 2366 囉!

[root@www ~]# partprobe

# 2. 不需要格式化,直接進行 sector 表面的複製!
[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
# 這兩個玩意兒會『一模一樣』喔！
[root@www ~]# umount /mnt
```

非常有趣的範例吧！新分割出來的 partition 不需要經過格式化，因為 dd 可以將原本舊的 partition 上面，將 sector 表面的資料整個複製過來！當然連同 superblock, boot sector, meta data 等等通通也會複製過來！是否很有趣呢？未來你想要建置兩顆一模一樣的磁碟時，只要下達類似：`dd if=/dev/sda of=/dev/sdb`，就能夠讓兩顆磁碟一模一樣，甚至 /dev/sdb 不需要分割與格式化，因為該指令可以將 /dev/sda 內的所有資料，包括 MBR 與 partition table 也複製到 /dev/sdb 說！^_^



這個指令挺有趣的，因為 cpio 可以備份任何東西，包括裝置設備檔案。不過 cpio 有個大問題，那就是 cpio 不會主動的去找檔案來備份！啊！那怎辦？所以囉，一般來說，cpio 得要配合類似 `find` 等可以找到檔名的指令來告知 cpio 該被備份的資料在哪裡啊！有點小麻煩啦～因為牽涉到我們在第三篇才會談到的[資料流重導向](#)說～所以這裡你就先背一下語法，等到第三篇講完你就知道如何使用 cpio 囉！

```
[root@www ~]# cpio -ovcB > [file|device] <==備份
[root@www ~]# cpio -ivcd < [file|device] <==還原
[root@www ~]# cpio -ivct < [file|device] <==察看
```

備份會使用到的選項與參數：

- o：將資料 copy 輸出到檔案或裝置上
 - B：讓預設的 Blocks 可以增加至 5120 bytes，預設是 512 bytes！
- 這樣的好處是可以讓大檔案的儲存速度加快(請參考 i-nodes 的觀念)

還原會使用到的選項與參數：

- i：將資料自檔案或裝置 copy 出來系統當中
- d：自動建立目錄！使用 cpio 所備份的資料內容不見得會在同一層目錄中，因此我們必須要讓 cpio 在還原時可以建立新目錄，此時就得要 -d 選項的幫助！
- u：自動的將較新的檔案覆蓋較舊的檔案！
- t：需配合 -i 選項，可用在"察看"以 cpio 建立的檔案或裝置的內容

一些可共用的選項與參數：

- v：讓儲存的過程中檔案名稱可以在螢幕上顯示
- c：一種較新的 portable format 方式儲存

你應該會發現一件事情，就是上述的選項與指令中怎麼會沒有指定需要備份的資料呢？還有那個大於(>)與小於(<)符號是怎麼回事啊？因為 cpio 會將資料整個顯示到螢幕上，因此我們可以透過將這些螢幕的資料重新導向(>)一個新的檔案！至於還原呢？就是將備份檔案讀進來 cpio (<) 進行處理之意！我們來進行幾個案例你就知道啥是啥了！

範例：找出 /boot 底下的所有檔案，然後將他備份到 /tmp/boot.cpio 去！

```
[root@www ~]# find /boot -print
/boot
/boot/message
/boot/initrd-2.6.18-128.el5.img
....以下省略....
# 透過這個 find 我們可以找到 /boot 底下應該要存在的檔名！包括檔案與目錄

[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
```

我們使用 `find /boot` 可以找出檔名，然後透過那條管線 (`|`，亦即鍵盤上的 `shift+\`` 的組合)，就能將檔名傳給 `cpio` 來進行處理！最終會得到 `/tmp/boot.cpio` 那個檔案喔！接下來讓我們來進行解壓縮看看。

```
範例：將剛剛的檔案給他在 /root/ 目錄下解開
[root@www ~]# cpio -idvc < /tmp/boot.cpio
[root@www ~]# ll /root/boot
# 你可以自行比較一下 /root/boot 與 /boot 的內容是否一模一樣！
```

事實上 `cpio` 可以將系統的資料完整的備份到磁帶機上頭去喔！如果你有磁帶機的話！

- 備份：`find / | cpio -ocvB > /dev/st0`
- 還原：`cpio -idvc < /dev/st0`

這個 `cpio` 好像不怎麼好用啦！但是，他可是備份的時候的一項利器呢！因為他可以備份任何的檔案，包括 `/dev` 底下的任何裝置檔案！所以他可是相當重要的呢！而由於 `cpio` 必需要配合其他的程式，例如 `find` 來建立檔名，所以 `cpio` 與管線命令及資料流重導向的相關性就相當的重要了！

其實系統裡面已經含有一個使用 `cpio` 建立的檔案喔！那就是 `/boot/initrd-xxx` 這個檔案啦！現在讓我們來將這個檔案解壓縮看看，看你能不能發現該檔案的內容為何？

```
# 1. 我們先來看看該檔案是屬於什麼檔案格式，然後再加以處理：
[root@www ~]# file /boot/initrd-2.6.18-128.el5.img
/boot/initrd-2.6.18-128.el5.img: gzip compressed data, ...
# 唔！看起來似乎是使用 gzip 進行壓縮過～那如何處理呢？

# 2. 透過更名，將該檔案增加副檔名，然後予以解壓縮看看：
[root@www ~]# mkdir initrd
[root@www ~]# cd initrd
[root@www initrd]# cp /boot/initrd-2.6.18-128.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！露出馬腳了吧！確實是 cpio 的文件檔喔！
```

```
# 3. 開始使用 cpio 解開此檔案：
[root@www initrd]# cpio -iduv < initrd.img
sbin
init
sysroot
....以下省略....
# 瞧！這樣就將這個檔案解開囉！這樣瞭解乎？
```



重點回顧

- 壓縮指令為透過一些運算方法去將原本的檔案進行壓縮，以減少檔案所佔用的磁碟容量。壓縮前與壓縮後的檔案所佔用的磁碟容量比值，就可以被稱為是『壓縮比』
- 壓縮的好處是可以減少磁碟容量的浪費，在 WWW 網站也可以利用檔案壓縮的技術來進行資料的傳送，好讓網站頻寬的可利用率上升喔
- 壓縮檔案的副檔名大多是：『*.tar, *.tar.gz, *.tgz, *.gz, *.Z, *.bz2』
- 常見的壓縮指令有 gzip 與 bzip2，其中 bzip2 壓縮比較之 gzip 還要更好，建議使用 bzip2！
- tar 可以用來進行檔案打包，並可支援 gzip 或 bzip2 的壓縮。
- 壓 縮：tar -jcv -f filename.tar.bz2 要被壓縮的檔案或目錄名稱
- 查 詢：tar -jtv -f filename.tar.bz2
- 解壓縮：tar -jxv -f filename.tar.bz2 -C 欲解壓縮的目錄
- dump 指令可備份檔案系統或單一目錄
- dump 的備份若針對檔案系統時，可進行 0-9 的 level 差異備份！其中 level 0 為完整備份；
- restore 指令可還原被 dump 建置的備份檔；
- 要建立光碟燒錄資料時，可透過 mkisofs 指令來建置；
- 可透過 cdrecord 來寫入 CD 或 DVD 燒錄機
- dd 可備份完整的 partition 或 disk，因為 dd 可讀取磁碟的 sector 表面資料
- cpio 為相當優秀的備份指令，不過必須要搭配類似 find 指令來讀入欲備份的檔名資料，方可進行備份動作。



本章習題

(要看答案請將滑鼠移動到『答：』底下的空白處，按下左鍵圈選空白處即可察看)

- 情境模擬題一：妳想要讓系統恢復到第八章情境模擬後的結果，亦即僅剩下 /dev/hdc6 以前的 partition，本章練習產生的 partition 都需要恢復原狀。因此 /dev/hdc8, /dev/hdc9 (在本章練習過程中產生的) 請將他刪除！刪除的方法同第八章的情境模擬題一所示。
- 情境模擬題二：妳想要逐時備份 /srv/myproject 這個目錄內的資料，又擔心每次備份的資訊太多，因此想要使用 dump 的方式來逐一備份資料到 /backups 這個目錄下。該如何處理？
 - 目標：瞭解到 dump 以及各個不同 level 的作用；
 - 前提：被備份的資料為單一 partition，亦即本例中的 /srv/myproject
 - 需求：/srv/myproject 為單一 filesystem，且在 /etc/fstab 內此掛載點的 dump 欄位為 1

實際處理的方法其實還挺簡單的！我們可以這樣做看看：

1. 先替該目錄製作一些資料，亦即複製一些東西過去吧！
`cp -a /etc /boot /srv/myproject`
 2. 開始進行 dump，記得，一開始是使用 level 0 的完整備份喔！
`mkdir /backups`
`dump -0u -j -f /backups/myproject.dump /srv/myproject`
上面多了個 -j 的選項，目的是為了要進行壓縮，減少備份的資料量。
 3. 嘗試將 /srv/myproject 這個檔案系統加大，將 /var/log/ 的資料複製進去吧！
`cp -a /var/log/ /srv/myproject`
此時原本的 /srv/myproject 已經被改變了！繼續進行備份看看！
 4. 將 /srv/myproject 以 level 1 來進行備份：
`dump -1u -j -f /backups/myproject.dump.1 /srv/myproject`
`ls -l /backups`
妳應該就會看到兩個檔案，其中第二個檔案 (myproject.dump.1) 會小的多！這樣就搞定囉備份資料！
- 情境模擬三：假設過了一段時間後，妳的 /srv/myproject 變的怪怪的，妳想要將該 filesystem 以剛剛的備份資料還原，此時該如何處理呢？妳可以這樣做的：
 1. 先將 /srv/myproject 卸載，並且將該 partition 重新格式化！
`umount /dev/hdc6`
`mkfs -t ext3 /dev/hdc6`
 2. 重新掛載原本的 partition，此時該目錄內容應該是空的！
`mount -a`
妳可以自行使用 df 以及 ls -l /srv/myproject 查閱一下該目錄的內容，是空的啦！
 3. 將完整備份的 level 0 的檔案 /backups/myproject.dump 還原回來：
`cd /srv/myproject`
`restore -r -f /backups/myproject.dump`
此時該目錄的內容為第一次備份的狀態！還需要進行後續的處理才行！
 4. 將後續的 level 1 的備份也還原回來：
`cd /srv/myproject`
`restore -r -f /backups/myproject.dump.1`
此時才是恢復到最後一次備份的階段！如果還有 level 2, level 3 時，就得要一個一個的依序還原才行！



參考資料與延伸閱讀

- 台灣學術網路管理文件：Backup Tools in UNIX(Linux):
http://nmc.nchu.edu.tw/tanet/backup_tools_in_unix.htm
- 中文 How to 文件計畫 (CLDP)：<http://www.linux.org.tw/CLDP/HOWTO/hardware/CD-Writing-HOWTO/CD-Writing-HOWTO-3.html>

- 熊寶貝工作記錄之：Linux 燒錄實作：http://csc.ocean-pioneer.com/docum/linux_burn.html
- PHP5 網管實驗室：<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 的部分內容，尤其是 -P 這個參數的配合用法

2005/07/26：將舊有的文章移動到 [這裡](#)

2005/07/27：大略修改了一些風格，另外，動作較大的是在範例的部分！

2005/08/29：加入了 dd 這個有趣的指令喔！

2006/05/02：將原本『tar -zxvpf /tmp/etc.tar.gz /etc』修改為『tar -zcvpf /tmp/etc.tar.gz /etc』感謝討論區網友 chinu 提供的資訊。

2008/10/31：將原本針對 FC4 的舊版本移動到[此處](#)

2008/12/18：這次的改版在這一章添加了不少東西！尤其是將 cpio 與 dd 的範例重新做個整理！並加入 dump/restore, mkisofs/cdrecord

2009/08/20：加入情境模擬的題目。

2003/02/09以來統計人數

772731

| [鳥站首頁](#) | [基礎學習](#) | [伺服器架設](#) | [企業應用](#) | [桌面應用](#) | [安全管理](#) | [鳥站討論板](#) | [酷學園](#) | [書籍戡誤](#) | [關於鳥哥](#) | [崑山資傳](#) |



本網頁主要以 [firefox](#) 配合解析度 1024x768 作為設計依據

<http://linux.vbird.org> is designed by [VBird](#) during 2001-2011. [ksu.edu](#)