

# 檔案與檔案系統的壓縮與打包

陳建良



# 內容

- 壓縮檔案的用途與技術
- Linux 系統常見的壓縮指令
- 打包指令：`tar`
- 完整備份工具：`dump`
- 光碟寫入工具
- 其他常見的壓縮與備份工具



# 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它複製完成的困擾？又，你是否有過，發現一個軟體裡面有好多檔案，這些檔案要將它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，可以將它的磁碟使用量降低，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？我們來稍微談一談它的原理好了。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  $1 \text{ byte} = 8 \text{ bits}$ 。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？它會如何記錄？假設一個 **byte** 可以看成底下的模樣：

□□□□□□□□



# LINUX 系統常見的壓縮指令

- \*.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



# compress

```
[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！
```



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
```

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！

# gzip, zcat

```
[dmtsai@study ~]$ gzip [-cdtv#] 檔名
```

```
[dmtsai@study ~]$ 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；

-# : # 為數字的意思，代表壓縮等級，-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：找出 /etc 底下（不含子目錄）容量最大的檔案，並將它複製到 /tmp，然後以 gzip 壓縮

```
[dmtsai@study ~]$ ls -ldSr /etc/* # 忘記選項意義？請自行 man 囉！
```

.....(前面省略).....

```
-rw-r--r--. 1 root root 25213 Jun 10 2014 /etc/dnsmasq.conf
```

```
-rw-r--r--. 1 root root 69768 May 4 17:55 /etc/ld.so.cache
```

```
-rw-r--r--. 1 root root 670293 Jun 7 2013 /etc/services
```

```
[dmtsai@study ~]$ cd /tmp
```

```
[dmtsai@study tmp]$ cp /etc/services .
```

```
[dmtsai@study tmp]$ gzip -v services
```

```
services: 79.7% -- replaced with services.gz
```

```
[dmtsai@study tmp]$ ll /etc/services /tmp/services*
```

```
-rw-r--r--. 1 root root 670293 Jun 7 2013 /etc/services
```

```
-rw-r--r--. 1 dmtsai dmtsai 136088 Jun 30 18:40 /tmp/services.gz
```



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
```



# bzip2, bzipcat

```
[root@www ~]# bzip2 [-cdkzv#] 檔名
```

```
[root@www ~]# bzcat 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]# bzcat 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
```



# 打包指令：tar

```
[dmtsai@study ~]$ tar [-z|-j|-J] [cv] [-f 待建立的新檔名] filename... <==打包與壓縮  
[dmtsai@study ~]$ tar [-z|-j|-J] [tv] [-f 既有的 tar檔名] <==察看檔名  
[dmtsai@study ~]$ tar [-z|-j|-J] [xv] [-f 既有的 tar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。
- z : 透過 gzip 的支援進行壓縮/解壓縮：此時檔名最好為 \*.tar.gz
- j : 透過 bzip2 的支援進行壓縮/解壓縮：此時檔名最好為 \*.tar.bz2
- J : 透過 xz 的支援進行壓縮/解壓縮：此時檔名最好為 \*.tar.xz  
特別留意，-z, -j, -J 不可以同時出現在一串指令列中
- 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！(比較不會忘記)
- 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(小寫) : 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(大寫) : 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！



# 打包指令：tar

- 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



## ■ 使用 tar 加入 -j 或 -z 的參數備份 /etc/ 目錄

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



## ■ 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

```
[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 -jpPcv -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。
```



- 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)

```
[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)....
```





## ■ 僅解開單一檔案的方法

```
# 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！
# 這裡你先有個概念即可！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

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



## ■ 僅備份比某個時刻還要新的檔案

```
# 1. 先由 find 找出比 /etc/passwd 還要新的檔案
[root@www ~]# find /etc -newer /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！你先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/smartd.conf      <==真的有備份的檔案
....(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！
```



## ■ 特殊應用：利用管線命令與資料流

```
# 1. 將 /etc 整個目錄一邊打包一邊在 /tmp 解開
[root@www ~]# cd /tmp
[root@www ~]# tar -cvf - /etc | tar -xvf -
```

# 這個動作有點像是 cp -r /etc /tmp 啦~依舊是有其有用途的！  
# 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！



# 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)
- **/home/loop\*** 不需要備份，而且 **/root** 底下的壓縮檔也不需要備份，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？



# 1. 先處理要放置備份資料的目錄與權限：

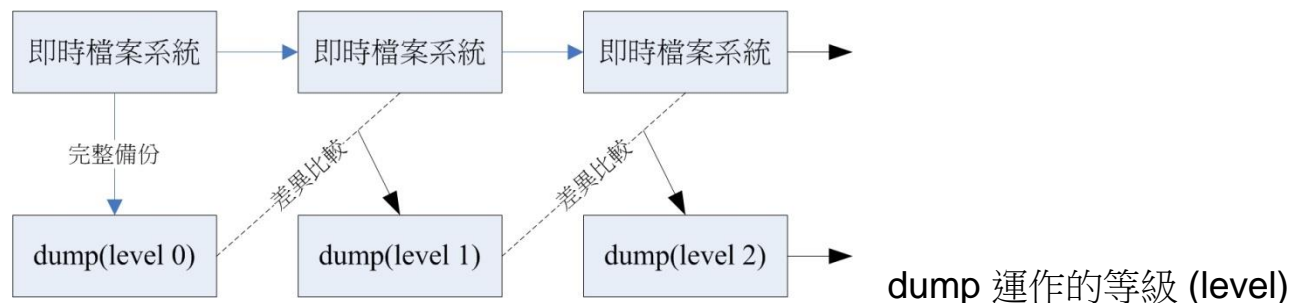
```
[root@www ~]# mkdir /backups
[root@www ~]# chmod 700 /backups
[root@www ~]# ll -d /backups
drwx----- 2 root root 4096 Nov 30 16:35 /backups
```

# 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



```
[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？





- 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；



# 用 dump 備份完整的檔案系統

# 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
<b>/dev/hdc1</b>	<b>99M</b>	<b>11M</b>	<b>83M</b>	<b>12%</b>	<b>/boot</b> <==看起來最小的就是它啦！
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 -0u -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，你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          ]
```

只備份差異的資料

```
[root@free ~]# dump -1u -f /root/boot.dump.1 /boot
```

# 用 dump 備份非檔案系統，亦即單一目錄的方法

```
# 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！
```



# restore

```
[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

```
[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....

# 比較差異

- 為什麼 **dump** 可以進行累積備份呢？就是因為他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
```

```
filesystem = /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
```



# 還原整個檔案系統

# 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
```

# mkisofs：建立映像檔

```
[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。



- 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
```

```
[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
```

# cdrecord：光碟燒錄工具

```
[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

```
[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) *
```



# 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
```





# 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
```



# dd

```
[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。



# cpio

```
[root@www ~]# cpio -ovcB > [file|device] <==備份  
[root@www ~]# cpio -ivcdu < [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

