linux打包压缩命令汇总 - 风生水起 - 博客园

linux打包压缩命令汇总

tar命令

[root@linux ~]# tar [-cxtzjvfpPN] 文件与目录 ....

参数：

-c ：建立一个压缩文件的参数指令(create 的意思)；

-x ：解开一个压缩文件的参数指令！

-t ：查看 tarfile 里面的文件！

特别注意，在参数的下达中， c/x/t 仅能存在一个！不可同时存在！

因为不可能同时压缩与解压缩。

-z ：是否同时具有 gzip 的属性？亦即是否需要用 gzip 压缩？

-j ：是否同时具有 bzip2 的属性？亦即是否需要用 bzip2 压缩？

-v ：压缩的过程中显示文件！这个常用，但不建议用在背景执行过程！

-f ：使用档名，请留意，在 f 之后要立即接档名喔！不要再加参数！

例如使用『 tar -zcvfP tfile sfile』就是错误的写法，要写成

『 tar -zcvPf tfile sfile』才对喔！

-p ：使用原文件的原来属性（属性不会依据使用者而变）

-P ：可以使用绝对路径来压缩！

-N ：比后面接的日期(yyyy/mm/dd)还要新的才会被打包进新建的文件中！

--exclude FILE：在压缩的过程中，不要将 FILE 打包！

范例：

范例一：将整个 /etc 目录下的文件全部打包成为 /tmp/etc.tar

[root@linux ~]# tar -cvf /tmp/etc.tar /etc <==仅打包，不压缩！

[root@linux ~]# tar -zcvf /tmp/etc.tar.gz /etc <==打包后，以 gzip 压缩

[root@linux ~]# tar -jcvf /tmp/etc.tar.bz2 /etc <==打包后，以 bzip2 压缩

特别注意，在参数 f 之后的文件档名是自己取的，我们习惯上都用 .tar 来作为辨识。

如果加 z 参数，则以 .tar.gz 或 .tgz 来代表 gzip 压缩过的 tar file ～

如果加 j 参数，则以 .tar.bz2 来作为附档名啊～

上述指令在执行的时候，会显示一个警告讯息：

『tar: Removing leading `/' from member names』那是关於绝对路径的特殊设定。

范例二：查阅上述 /tmp/etc.tar.gz 文件内有哪些文件？

[root@linux ~]# tar -ztvf /tmp/etc.tar.gz

由於我们使用 gzip 压缩，所以要查阅该 tar file 内的文件时，

就得要加上 z 这个参数了！这很重要的！

范例三：将 /tmp/etc.tar.gz 文件解压缩在 /usr/local/src 底下

[root@linux ~]# cd /usr/local/src

[root@linux src]# tar -zxvf /tmp/etc.tar.gz

在预设的情况下，我们可以将压缩档在任何地方解开的！以这个范例来说，

我先将工作目录变换到 /usr/local/src 底下，并且解开 /tmp/etc.tar.gz ，

则解开的目录会在 /usr/local/src/etc 呢！另外，如果您进入 /usr/local/src/etc

则会发现，该目录下的文件属性与 /etc/ 可能会有所不同喔！

范例四：在 /tmp 底下，我只想要将 /tmp/etc.tar.gz 内的 etc/passwd 解开而已
[root@linux ~]# cd /tmp
[root@linux tmp]# tar -zxvf /tmp/etc.tar.gz etc/passwd
我可以透过 tar -ztvf 来查阅 tarfile 内的文件名称，如果单只要一个文件，
就可以透过这个方式来下达！注意到！ etc.tar.gz 内的根目录 / 是被拿掉了！

范例五：将 /etc/ 内的所有文件备份下来，并且保存其权限！
[root@linux ~]# tar -zxvpf /tmp/etc.tar.gz /etc
这个 -p 的属性是很重要的，尤其是当您要保留原本文件的属性时！

范例六：在 /home 当中，比 2005/06/01 新的文件才备份
[root@linux ~]# tar -N '2005/06/01' -zcvf home.tar.gz /home

范例七：我要备份 /home, /etc ，但不要 /home/dmtsai
[root@linux ~]# tar --exclude /home/dmtsai -zcvf myfile.tar.gz /home/* /etc

范例八：将 /etc/ 打包后直接解开在 /tmp 底下，而不产生文件！
[root@linux ~]# cd /tmp
[root@linux tmp]# 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！
要注意的地方在於输出档变成 - 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！
这部分我们会在 Bash shell 时，再次提到这个指令跟大家再解释啰！

gzip, zcat 命令
[root@linux ~]# gzip [-cdt#] 档名
[root@linux ~]# zcat 档名.gz
参数：
-c ：将压缩的资料输出到萤幕上，可透过资料流重导向来处理；
-d ：解压缩的参数；
-t ：可以用来检验一个压缩档的一致性～看看文件有无错误；
-# ：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：
范例一：将 /etc/man.config 複制到 /tmp ，并且以 gzip 压缩
[root@linux ~]# cd /tmp
[root@linux tmp]# cp /etc/man.config .
[root@linux tmp]# gzip man.config
此时 man.config 会变成 man.config.gz ！
范例二：将范例一的文件内容读出来！
[root@linux tmp]# zcat man.config.gz
此时萤幕上会显示 man.config.gz 解压缩之后的文件内容！！

范例三：将范例一的文件解压缩
[root@linux tmp]# gzip -d man.config.gz

范例四：将范例三解开的 man.config 用最佳的压缩比压缩，并保留原本的文件
[root@linux tmp]# gzip -9 -c man.config > man.config.gz

bzip2, bzcat 命令
[root@linux ~]# bzip2 [-cdz] 档名
[root@linux ~]# bzcat 档名.bz2
参数:
-c ：将压缩的过程产生的资料输出到萤幕上!
-d ：解压缩的参数
-z ：压缩的参数
-# ：与 gzip 同样的，都是在计算压缩比的参数， -9 最佳， -1 最快!
范例:
范例一：将刚刚的 /tmp/man.config 以 bzip2 压缩
[root@linux tmp]# bzip2 -z man.config
此时 man.config 会变成 man.config.bz2 !
范例二：将范例一的文件内容读出来!
[root@linux tmp]# bzcat man.config.bz2
此时萤幕上会显示 man.config.bz2 解压缩之后的文件内容!!

范例三：将范例一的文件解压缩
[root@linux 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，并保留原本的文件
[root@linux tmp]# bzip2 -9 -c man.config > man.config.bz2

compress 命令
[root@linux ~]# compress [-dcr] 文件或目录
参数:
-d ：用来解压缩的参数
-r ：可以连同目录下的文件也同时给予压缩呢!
-c ：将压缩资料输出成为 standard output (输出到萤幕)
范例:
范例一：将 /etc/man.config 複制到 /tmp ，并加以压缩
[root@linux ~]# cd /tmp
[root@linux tmp]# cp /etc/man.config .
[root@linux tmp]# compress man.config
[root@linux tmp]# ls -l
-rw-r--r-- 1 root root 2605 Jul 27 11:43 man.config.Z
范例二：将刚刚的压缩档解开
[root@linux tmp]# compress -d man.config.Z

范例三：将 man.config 压缩成另外一个文件来备份
[root@linux tmp]# compress -c man.config > man.config.back.Z
[root@linux tmp]# ll man.config*
-rw-r--r-- 1 root root 4506 Jul 27 11:43 man.config
-rw-r--r-- 1 root root 2605 Jul 27 11:46 man.config.back.Z
这个 -c 的参数比较有趣!他会将压缩过程的资料输出到萤幕上，而不是写入成为
file.Z 文件。所以，我们可以透过资料流重导向的方法将资料输出成为另一个档名。
关於资料流重导向，我们会在 bash shell 当中详细谈论的啦!

dd 命令
[root@linux ~]# dd if="input_file" of="outptu_file" bs="block_size" \
count="number"
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，如果没有设定时，预设是 512 bytes
count：多少个 bs 的意思。
范例：
范例一：将 /etc/passwd 备份到 /tmp/passwd.back 当中
[root@linux ~]# dd if=/etc/passwd of=/tmp/passwd.back
3+1 records in
3+1 records out
[root@linux ~]# ll /etc/passwd /tmp/passwd.back
-rw-r--r-- 1 root root 1746 Aug 25 14:16 /etc/passwd
-rw-r--r-- 1 root root 1746 Aug 29 16:57 /tmp/passwd.back
仔细的看一下，我的 /etc/passwd 文件大小为 1746 bytes，因为我没有设定 bs ，
所以预设是 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：备份 /dev/hda 的 MBR
[root@linux ~]# dd if=/dev/hda of=/tmp/mbr.back bs=512 count=1
1+0 records in
1+0 records out
这就得好好瞭解一下啰～我们知道整颗硬盘的 MBR 为 512 bytes，
就是放在硬盘的第一个 sector 啦，因此，我可以利用这个方式来将
MBR 内的所有资料都纪录下来，真的很厉害吧！ ^_^

范例三：将整个 /dev/hda1 partition 备份下来。
[root@linux ~]# dd if=/dev/hda1 of=/some/path/filenaem
这个指令很厉害啊！将整个 partition 的内容全部备份下来～
后面接的 of 必须要不是在 /dev/hda1 的目录内啊～否则，怎么读也读不完～
这个动作是很有效用的，如果改天你必须要完整的将整个 partition 的内容填回去，
则可以利用 dd if=/some/file of=/dev/hda1 来将资料写入到硬盘当中。
如果想要整个硬盘备份的话，就类似 Norton 的 ghost 软体一般，
由 disk 到 disk ，嘿嘿～利用 dd 就可以啦～厉害厉害!

cpio 命令
[root@linux ~]# cpio -covB > [file|device] <==备份
[root@linux ~]# cpio -icduv < [file|device] <==还原
参数：
-o ：将资料 copy 输出到文件或装置上
-i ：将资料自文件或装置 copy 出来系统当中
-t ：查看 cpio 建立的文件或装置的内容
-c ：一种较新的 portable format 方式储存
-v ：让储存的过程中文件名称可以在萤幕上显示
-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文件的储存速度加快(请参考 i-nodes 的观念)
-d ：自动建立目录！由於 cpio 的内容可能不是在同一个目录内，

如此的话在反备份的过程会有问题！ 这个时候加上 -d 的话，
就可以自动的将需要的目录建立起来了！
-u ：自动的将较新的文件覆盖较旧的文件！
范例：
范例一：将所有系统上的资料通通写入磁带机内！
[root@linux ~]# find / -print | cpio -covB > /dev/st0
一般来说，使用 SCSI 介面的磁带机，代号是 /dev/st0 喔！
范例二：检查磁带机上面有什么文件？
[root@linux ~]# cpio -icdvt < /dev/st0
[root@linux ~]# cpio -icdvt < /dev/st0 > /tmp/content
第一个动作当中，会将磁带机内的档名列出到萤幕上面，而我们可以透过第二个动作，
将所有的档名通通纪录到 /tmp/content 文件去！

范例三：将磁带上的资料还原回来～
[root@linux ~]# cpio -icduv < /dev/st0
一般来说，使用 SCSI 介面的磁带机，代号是 /dev/st0 喔！

范例四：将 /etc 底下的所有『文件』都备份到 /root/etc.cpio 中！
[root@linux ~]# find /etc -type f | cpio -o > /root/etc.cpio
这样就能够备份啰～您也可以将资料以 cpio -i < /root/etc.cpio
来将资料捉出来！！！！