

第八章 文件与文件系统的压缩、打包与备份

一、Linux 系统常见的压缩命令

在 Linux 的环境中,压缩文件的扩展名大多是“*.tar, *.tar.gz, *.tgz, *.gz, *.Z, *.bz2, *.xz”。但是在 linux 中扩展名不是没什么作用吗?这是因为, Linux 支持的压缩命令非常多,人们要想解压该文档,就必须知道该文档是用什么命令来进行压缩的,所以拓展名就是为了告诉用户它是使用什么方式压缩的,那么我们就可以用对应的命令进行解压缩。

tar 是一种将很多文件(目录)打包成一个文件的命令,单纯的 tar 命令只是打包,并不提供压缩,但后来的 GNU 计划中,将整个 tar 与压缩功能结合在一起了。

1.gzip, zcat/zmore/zless/zgrep

gzip 所建立的压缩文件后缀名为.gz。该命令也可以解压缩 compress、zip 和 gzip 等软件所压缩的文件。其使用格式如下:

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

```
[dmtsai@study ~]$ zcat 文件名.gz
```

选项与参数:

-c: 将压缩的数据输出到屏幕上,可通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~看看文件有无错误;

-v: 可以显示出原文件/压缩文件案的压缩比等信息;

-#: #为数字的意思,代表压缩等级,-1 最快,但是压缩比最差、-9 最慢,但是压缩比最好!预设是-6。

注意:在预设状态下,使用 gzip 命令会压缩文件为.gz 的文件,但是**源文件就会被删除**。

zcat 用于不真正解压缩文件,就能显示压缩包中文件的内容的场合。

```
zcat [选项] 文件名
```

-S: 指定 gzip 格式的压缩包的后缀。当后缀不是标准压缩包后缀时使用此选项;

-c: 将文件内容写到标注输出;

-d: 执行解压缩操作;

-l: 显示压缩包中文件的列表;

-L: 显示软件许可信息;

-q: 禁用警告信息;

-r: 在目录上执行递归操作;

-t: 测试压缩文件的完整性;

-V: 显示命令的版本信息;

-l: 更快的压缩速度;

-9: 更高的压缩比。

zcat/zmore/zless 则可以对应于 cat/more/less 的方式来读取纯文本档被压缩后的压缩文件。由于 gzip 这个压缩命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 来解压缩,同时 zcat 这个命令可以同时读取 compress 与 gzip 的压缩文件。

另外,如果想从压缩文本文档中查找数据,可以使用 **zgrep** 来搜索关键词,而无需将压缩文本解压缩。如果使用 compress 建立的.z 文件,可以使用 znew 将文件更新成 gzip 的格式。

2.bzzip2, bzip2/bzcat/bzmore/bzless/bzgrep

bzip2 则是为了取代 gzip 并提供更佳的压缩比。其使用方法如下:

```
[dmtsai@study ~]$ bzip2 [-cdkzv#] 文件名
```

```
[dmtsai@study ~]$ bzipcat 文件名.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，而不会删除原始的文件喔；
- z: 压缩的参数（默认值，可以不加）；
- v: 可以显示出原文件/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。

3. xz, xzcat/xzmore/xzless/xzgrep

xz 压缩比相比 bzip2 来说更高，用法也基本上跟 gzip/bzip2 一致。

```
[dmtsai@study ~]$ xz [-dtlkc#] 文件名
```

```
[dmtsai@study ~]$ xzcat 文件名.xz
```

选项与参数：

- d: 解压缩；
- t: 测试压缩文件的完整性；
- l: 列出压缩文件的相关信息；
- k: 保留原本的文件不删除；
- c: 将数据由屏幕上输出；
- #: 同上，压缩比参数。

综上所述，可以对这三个命令的性能和时间性能进行总结如下。

类型	压缩比	压缩所需时间
gzip	最低	最少
bzip2	中等	中等
xz	最高	最长

二、打包命令：tar、解压缩后的 SELinux 问题

虽然 gzip、bzip2、xz 也能够针对目录来进行压缩，不过它们对目录的压缩指的是“将目录内的所有文件**分别**进行压缩”，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打包为一个文件。

在 Linux 下，可以使用 tar 指令。它可以将多个目录或文件打包成一个大文件，并通过 gzip/bzip2/xz 的支持，将该文件进行压缩。

tar 的参数非常多，这里仅列举几个常用的。

```
[dmtsai@study ~]$ tar [-z|-j|-J] [cv] [-f 待建立的新档名] filename... <==打包与压缩
```

```
[dmtsai@study ~]$ tar [-z|-j|-J] [tv] [-f 既有的 tar 档名] <==查看文件名
```

```
[dmtsai@study ~]$ tar [-z|-j|-J] [xv] [-f 既有的 tar 档名] [-C 目录] <==解压缩
```

选项与参数：

- c: 新建打包文件，可搭配-v 来查看过程中被打包的文件名(filename)；
- t: 查看打包文件的内容含有哪些文件名；
- x: 解打包或解压缩的功能，可以搭配-C(大写)在特定目录解压缩/解开；

【注意：-c,-t,-x 不可同时出现在一串指令列中。】

-z: 通过 gzip 的支持进行压缩/解压缩：此时文件名最好为*.tar.gz；

-j: 通过 bzip2 的支持进行压缩/解压缩：此时文件名最好为*.tar.bz2；

-J: 通过 xz 的支持进行压缩/解压缩：此时文件名最好为*.tar.xz；

【注意：-z,-j,-J 不可以同时出现在一串命令列中。】

-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文件名；
-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；
-p(小写): 保留备份数据的原本权限与属性，常用于备份(-c)重要的配置文本；
-P(大写): 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。

最简单只需要记住以下三个指令：

①压缩：tar -jcv -f filename.tar.bz2 要被压缩的文件或目录名称；

②查询：tar -jtv -f filename.tar.bz2；

③解压缩：tar -jxv -f filename.tar.bz2 -C 解压缩的目录。

1.使用 tar 加入-z,-j 或-J 的参数备份/etc/目录

使用格式如下：

```
tar -zpcv -f /root/etc.tar.gz /etc
```

```
tar -jpcv -f /root/etc.tar.bz2 /etc
```

```
tar -Jpcv -f /root/etc.tar.xz /etc
```

加入 p 的原因是为了保存原本文件的权限和属性。备份的时候一定要把原本文件的权限都备份好。

2.查阅 tar 文件的数据内容，与备份文件名是否有根目录的意义

```
tar -jtv -f /root/etc.tar.bz2
```

如果加上-v，那么详细的文件权限/属性都会被列出来，如果只想知道文件名，那么就不需要加-v。我们还可以发现，每个文件名都没有了根目录。这是为了安全，我们使用 tar 备份可能会需要解压缩，在 tar 所记录的文件名就是解压缩后的实际文件名，**如果没有拿掉根目录，解压缩后的文件名就是绝对路径，就会在解压缩后覆盖原来/etc/xxx 的目录中去。**

如果确认使用绝对路径，那么就将选项里面的-p 修改为-P 即可。

3.将备份的数据解压缩，被考虑特定目录的解压缩（-C 的应用）

默认情况下，解压缩目录是解压缩到当前目录。如果需要修改解压缩的目录，可以使用 -C 选项。例如

```
tar -jxv -f /root/etc.tar.bz2 -C /tmp
```

即将 etc.tar.bz2 解压到/tmp 目录中。

4.仅解开单一文件的方法

我们假设以解压缩 shadow 文件为例。

步骤：

STEP 1: 先找到我们要的文件名。

```
tar -jtv -f /root/etc.tar.bz2 | grep 'shadow'
```

STEP 2: 将该文件解开。用法如下：

```
tar -jxv -f 打包文件.tar.bz2 待解开文件名
```

实际运行命令为：

```
tar -jxv -f /root/etc.tar.bz2 etc/shadow
```

5.打包某目录，但不包含该目录下某些文件

```
tar -jcv -f /root/system.tar.bz2 --exclude=/root/etc*
```

6.仅备份比某个时刻新的文件（其实就是增量备份）

某些情况下，我们需要备份新文件而不是旧文件。此时--newer-mtime 这个选项就极其重要。当我们仅使用--newer 的时候，表示后续日期包括 mtime 和 ctime，如果是--newer-mtime，

显然只是 mtime。

7.其它知识: tarfile 和 tarball

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。

8.特殊应用: 利用管道命令与数据流

在 tar 的使用中,有一种方式最特殊,那就是通过标准输入输出的数据流重定向(standard input/standard output)以及管道命令(pipe)的方式,将待处理的文件一边打包一边解压缩到目标目录去。

例如将/etc 整个目录一边打包一边在/tmp 解压缩的命令如下:

```
[root@study ~]# cd /tmp
```

```
[root@study tmp]# tar -cvf - /etc | tar -xvf -
```

这个命令有点像 cp -r /etc /tmp,但是要注意,输出文件和输入文件都变成了-,并且有一个|存在,者分别代表着 standard output、standard input 和管道命令。你可以把-想象为内存中的一个缓冲区,这就好理解了。

9.拓展: SELinux 问题

SELinux 的权限问题可能会让系统无法存取某些配置文件内容,导致影响系统的政策使用权。解决方法如下:

①通过各种可行的救援方式登入系统,然后修改/etc/selinux/config 档案,将 SELinux 改成 permissive 模式,重新启动后系统就正常了;

②在第一次恢复系统后,不要立即重新启动,先使用 restorecon-Rv/etc 自动修复一下 SELinux 的类型即可;

③通过各种可行的方式登入系统,建立/.autorelabel 档案,重新启动后系统会自动修复 SELinux 的类型,并且又会再次重新启动,之后就正常了。

三、XFS 文件系统的备份与还原

使用 tar 是针对目录树系统进行备份,如果想要针对整个文件系统进行备份和还原,就要使用 xfsdump 和 xfsrestore 两个命令。**注意 xfsdump 不支持目录备份。**

1.XFS 文件系统备份: xfsdump

xfsdump 除了可以进行文件系统的完整备份 (full backup),还可以进行增量备份 (Incremental backup)。

使用 xfsdump 时,请注意以下限制:

①xfsdump 不支持没有挂载的文件系统备份!所以只能备份已挂载的;

②xfsdump 必须使用 root 的权限才能操作 (涉及文件系统的关系);

③xfsdump 只能备份 XFS 文件系统;

④xfsdump 备份下来的数据只能让 xfsrestore 解析;

⑤xfsdump 是透过文件系统的 UUID 来分辨各个备份文件的,因此不能备份两个具有相同 UUID 的文件系统。

```
[root@study ~]# xfsdump [-L S_label] [-M M_label] [-l #] [-f 备份文件名] 待备份数据
```

```
[root@study ~]# xfsdump -l
```

选项与参数:

-L: xfsdump 会记录每次备份的 session 标头,这里可以填写针对此文件系统的简易说明;

-M: xfsdump 可以纪录储存媒体的标头,这里可以填写此媒体的简易说明;

-l: 是 L 的小写,就是指定等级~有 0~9 共 10 个等级 (默认为 0,即完整备份);

-f: 后面接产生的文件,亦可接例如/dev/st0 装置文件名或其他一般文件的文件名等;

-l: 从/var/lib/xfsdump/inventory 列出目前备份的信息状态。

(1) 用 xfsdump 备份完整文件系统。

```
#1.先确定/boot 是独立的文件系统！
[root@study ~]# df -h /boot
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda2        1014M  131M  884M  13% /boot      #挂载/boot 的是/dev/vda 装置！
# 看！确实是独立的文件系统喔！ /boot 是挂载点！

# 2. 将完整备份的文件名记录成为 /srv/boot.dump :
[root@study ~]# xfsdump -l 0 -L boot_all -M boot_all -f /srv/boot.dump /boot
```

(2) 用 xfsdump 进行增量备份

注意：一定要进行过完整备份之后，才能进行增量备份。

```
# 0.看一下有没有任何文件系统被 xfsdump 过的资料
[root@study ~]# xfsdump -l
#1.开始建立增量备份文件，此时我们使用 level1 吧：
[root@study~]#xfsdump -l 1 -Lboot_2 -Mboot_2 -f /srv/boot.dump1 /boot
#2.最后再看一下是否有记录 level1 备份的时间点呢
[root@study~]#xfsdump -l
```

2.XFS 文件系统还原：xfsrestore

```
[root@study ~]# xfsrestore -l          <==用来察看备份文件资料
[root@study ~]# xfsrestore [-f 备份文件] [-L S_label] [-s] 复原目录<==单一文件系统复原
[root@study ~]# xfsrestore [-f 备份文件] -r 待复原目录    <==透过增量备份文件复原系统
[root@study ~]# xfsrestore [-f 备份文件] -i 待复原目录    <==进入交互模式
选项与参数：
-l: 跟 xfsdump 相同的输出，可查询备份数据，包括 Label 名称与备份时间等；
-f: 后面接的就是备份文件；
-L: 就是 Session 的 Labelname，可用-l 查询到的数据；
-s: 需要接某特定目录，亦即仅复原某一个文件或目录之意；
-r: 如果是用文件来储存备份数据，那这个就不需要使用，如果是一个磁带内有多个文件，
    需要这东西来达成增量恢复；
-i: 进入交互模式，管理员使用的。
```

(1) 用 xfsrestore 观察 xfsdump 后的备份数据内容

xfsrestore -l

(2) 简单恢复 level0 的文件系统（完全备份文件）

xfsrestore -f /srv/boot.dump -L boot_all /boot

若仅复原备份文件内的 grub2 到/tmp/boot2/里面去的话，则为

xfsrestore -f /srv/boot.dump -L boot_all -s grub2 .tmp/boot2

(3) 仅恢复增量备份资料

恢复增量备份文件与恢复单一文件系统类似，备份数据是由 level0->level1->level2...进行，所以恢复也是类似。

xfsrestore -f /srv/boot.dump1 /tmp/boot

(4) 仅恢复部分文件的 xfsrestore 交互模式

如果我们想要部分数据还原，但是我们又不知道备份文件里面有什么，就需要交互界面来做尝试。

```
#1.先进入备份文件内，准备找出需要备份的文件名数据，同时恢复到/tmp/boot3 中
mkdir /tmp/boot3
xfsrestore -f /srv/boot.dump -i /tmp/boot3
```

```
===== subtree selection dialog =====

the following commands are available:

    pwd
    ls [ <path> ]
    cd [ <path> ]
    add [ <path> ]      # 可以加入復原檔案列表中
    delete [ <path> ]  # 從復原列表拿掉檔名！並非刪除喔！
    extract             # 開始復原動作！
    quit
    help

ls
#此处全部省略
-> add grub
-> add grub2
-> add config-3.10.0-229.el7.x86_64
-> extract
```

四、光盘写入工具

1.mkisofs: 建立镜像文件

```
[root@study ~]# mkisofs [-o 镜像文件] [-Jrv] [-V vol] [-m file] 待备份文件... \
> -graft-point isodir=systemdir ...

选项与参数:
-o: 后面接你想要产生的那个镜像文件的文件名;
-J: 产生较兼容于 windows 机器的文件名结构,可增加文件名长度到 64 个 unicode 字符;
-r: 透过 RockRidge 产生支持 Unix/Linux 的文件数据,可记录较多的信息(如 UID/GID 等);
-v: 显示构建 ISO 文件的过程;
-Vvol: 建立 Volume;
-mfile: -m 为排除文件(exclude)的意思,后面的文件不备份到镜像文件中;
-graft-point: graft 有转嫁或移植或移植的意思。
```

光盘的格式一般称为 iso9660, 这种格式一般仅支持旧版的 DOS 文件名, 亦即文件名只能以 8.3(文件名 8 个字符, 扩展名 3 个字符)的方式存在。所以要加上-r 这个选项。

一般情况下, 所有被添加到镜像文件中的文件都会被放置到镜像文件的根目录。为了解决这个问题, 可以使用-**graft-point** 选项, 利用如下方法来定义位于镜像文件中的目录:

镜像文件中的目录所在=实际 Linux 文件系统的目录所在

例如:/movies/=/stv/movies/(在 Linux 的/srv/movies 内的文件, 加至镜像文件中的/movies/目录)。

(1) 制作/修改开机启动光盘映像

```
mkisofs -o /custom.iso -b isolinux/isolinux.bin -c isolinux/boot.cat \
> -no-emul-boot -V 'CentOS 7 x86_64' -boot-load-size 4 -boot-info-table -R -J -v -T .
```

2. cdrecord: 光盘烧录工具

```
[root@study ~]# wodim --devices dev=/dev/sr0... <==查询刻录机的 BUS 位置
[root@study ~]# wodim -v dev=/dev/sr0 blank=[fast|all] <==擦除重复读写光盘
[root@study ~]# wodim -v dev=/dev/sr0 -format <==格式化 DVD+RW
[root@study ~]# wodim -v dev=/dev/sr0 [可用选项功能] file.iso
```

选项与参数：

- devices: 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；
- dev=/dev/sr0: 可以找出此光驱的 bus 地址，非常重要；
- blank=[fast|all]: blank 为擦除可重复写入的 CD/DVD-RW，使用 fast 较快，all 较完整；
- format: 对光盘片进行格式化，但是仅针对 DVD+RW 这种格式的 DVD；
- [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 BufferUnderrunFree 模式的写入功能
- sao: 支持 DVD-RW 的格式

(1) 检测你的刻录机位置

文本模式的刻录确实比较麻烦，需要首先找到刻录机才可以。查询刻录机的方式为：

```
[root@study ~]# ll /dev/sr0
brw-rw----+ 1 root cdrom 11, 0 Jun 26 22:14 /dev/sr0 # 一般 Linux 光驱文件名
[root@demo ~]# wodim --devices dev=/dev/sr0
wodim: Overview of accessible drives (1 found) :
-----
0  dev='/dev/sr0'      rwrw-- : 'ASUS' 'DRW-24D1ST'
```

注意：一定要有 dev=/dev/xxx，否则系统会通知找不到光盘。

(2) 进行 CD/DVD 的刻录工作

①先擦除光盘原始内容（只读光盘无需此操作）

```
[root@demo ~]# wodim -v dev=/dev/sr0 blank=fast
```

中间会跑出一堆信息告诉你擦除的进度，而且会有 10 秒钟的时间等待你的取消。

②开始刻录

```
[root@demo ~]# wodim -v dev=/dev/sr0 speed=4 -dummy -eject /tmp/system.img
```

③刻录完成，测试挂载一下，检查内容。

五、其它常见的压缩与备份工具

1.dd

在上一章，我们使用 dd 来创建了一个大文件。但是实际上，dd 的关键作用在于备份，其可以读取硬盘设备的内容（几乎是直接读取扇区“sector”），然后将这个设备备份成一个文件。dd 有很多用途，下面选取比较重要的几个讲述如下：

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

选项与参数：

if: 输入文件 (input file), 也可以是设备;
of: 输出文件 (output file), 也可以是设备;
bs: 计划的一个 block 的大小, 若未指定则预设为 512bytes (一个 sector 的大小);
count: 多少个 bs 的意思。

dd 是一个一个扇区去读写, 即使没有用到的扇区也被写入到备份当, 所以产生的文件会跟原本硬盘一样大。不过 dd 就是因为不理睬文件系统, 单纯有啥记录啥, 因此无论磁盘内的文件系统是否认识, 它都可以备份和还原。

2.cpio

cpio 可以备份任何东西, 包括硬件设备文件。不过 cpio 有个大问题, 那就是 cpio 不会主动的去找文件来备份。一般来说, cpio 要配合类似 find 等可以找到文件名的命令来高速 cpio 应该被备份的数据在哪里。

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

备份会使用到的选项与参数:

- o: 将数据 copy 输出到文件或设备上;
- B: 让预设的 Blocks 可以增加至 5120bytes, 预设是 512bytes 这样的好处是可以让大文件的储存速度加快 (请参考 i-nodes 的概念)。

还原会使用到的选项与参数:

- i: 将数据自文件或设备 copy 出来到系统当中;
- 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;
- u: 自动的将较新的文件覆盖较旧的文件;
- t: 需配合 -i 选项, 可用在查看以 cpio 建立的文件或设备的内容。

一些可共享的选项与参数:

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

我们发现, 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(<)是怎么回事呢?

因为 cpio 会将数据整个显示到屏幕上, 因此我们可以通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。

例如: 找出/boot 下所有的文件, 并将其备份到/tmp/boot.cpio 中去。命令如下:

```
[root@study ~]# cd /
[root@study /]# find boot -print
[root@study /]# find boot | cpio -ocvB > /tmp/boot.cpio
[root@study /]# ll -h /tmp/boot.cpio
-rw-r--r--. 1 root root 108M Jul  3 00:05 /tmp/boot.cpio
[root@study ~]# file /tmp/boot.cpio
/tmp/boot.cpio: ASCII cpio archive (SVR4 with no CRC)
```

我们使用 find root 来找到文件名, 然后通过管道, 就能讲文件名传送给 cpio 进行处理, 最终得到/tmp/boot.cpio 文件。为何要先转换到根目录再寻找 boot 呢, 为何不直接寻找/boot 呢? 这是因为 cpio 很笨, **它不会理会你给出的是绝对路径还是相对路径的文件名, 所以即便你加上绝对路径的/开头, 那么未来解开的时候, 它也一定会覆盖掉原本的/boot。**