

文件查找和压缩

讲师：王晓春

文件查找和打包压缩

内容概述

1 文件查找

1.1 locate

1.2 find

1.2.1 指定搜索目录层级

1.2.2 对每个目录先处理目录内的文件，再处理目录本身

1.2.3 根据文件名和inode查找

1.2.4 根据属主、属组查找

1.2.5 根据文件类型查找

1.2.6 空文件或目录

1.2.7 组合条件

1.2.8 排除目录

1.2.9 根据文件大小来查找

1.2.10 根据时间戳

1.2.11 根据权限查找

1.2.12 正则表达式

1.2.13 处理动作

1.3 参数替换 xargs

2 压缩和解压缩

2.1 compress 和 uncompress

2.2 gzip和gunzip

2.3 bzip2和bunzip2

2.4 xz 和 unxz

2.5 zip 和 unzip

3 打包和解包

3.1 tar

3.2 split

3.3 cpio

文件查找和打包压缩

内容概述

- locate
- find
- xargs
- compress和uncompress
- gzip和gunzip
- bzip2和bunzip2
- xz和unxz
- zip和unzip
- tar
- cpio

1 文件查找

在文件系统上查找符合条件的文件

文件查找：

- 非实时查找(数据库查找)：locate
- 实时查找：find

1.1 locate

- locate 查询系统上预建的文件索引数据库 /var/lib/mlocate/mlocate.db
- 索引的构建是在系统较为空闲时自动进行(周期性任务)，执行updatedb可以更新数据库
- 索引构建过程需要遍历整个根文件系统，很消耗资源
- locate和updatedb命令来自于mlocate包

工作特点：

- 查找速度快
- 模糊查找
- 非实时查找
- 搜索的是文件的全路径，不仅仅是文件名
- 可能只搜索用户具备读取和执行权限的目录

格式：

```
locate [OPTION]... [PATTERN]...
```

常用选项

- i 不区分大小写的搜索
- n N 只列举前N个匹配项目
- r 使用基本正则表达式

范例:

```
#搜索名称或路径中包含“conf”的文件
locate conf
#使用Regex来搜索以“.conf”结尾的文件
locate -r '\.conf$'
```

范例: locatedb创建数据库

```
[root@centos8 ~]#dnf -y install mlocate
[root@centos8 ~]#locate conf
locate: can not stat () `/var/lib/mlocate/mlocate.db': No such file or directory
[root@centos8 ~]#updatedb
[root@centos8 ~]#ll /var/lib/mlocate/mlocate.db
-rw-r----- 1 root slocate 1041065 Jun 11 20:08 /var/lib/mlocate/mlocate.db
[root@centos8 ~]#locate -n 3 conf
/boot/config-4.18.0-147.el8.x86_64
/boot/grub2/i386-pc/configfile.mod
/boot/loader/entries/5b85fc7444b240a992c42ce2a9f65db5-0-rescue.conf
```

范例: 文件新创建和删除,无法马上更新locate数据库

```
[root@centos8 ~]#touch test.log
[root@centos8 ~]#locate test.log
locate: can not stat () `/var/lib/mlocate/mlocate.db': No such file or directory
[root@centos8 ~]#updatedb
[root@centos8 ~]#locate test.log
/root/test.log
[root@centos8 ~]#touch test2.log
[root@centos8 ~]#locate test2.log
[root@centos8 ~]#updatedb
[root@centos8 ~]#locate test2.log
/root/test2.log
[root@centos8 ~]#rm -f test2.log
[root@centos8 ~]#locate test2.log
/root/test2.log
[root@centos8 ~]#
```

范例:

```
[root@centos8 ~]#locate -n 10 -ir '\.CONF$'
/boot/loader/entries/5b85fc7444b240a992c42ce2a9f65db5-0-rescue.conf
/boot/loader/entries/5b85fc7444b240a992c42ce2a9f65db5-4.18.0-147.el8.x86_64.conf
/etc/autofs.conf
/etc/autofs_ldap_auth.conf
/etc/dracut.conf
/etc/fuse.conf
/etc/host.conf
/etc/idmapd.conf
/etc/kdump.conf
/etc/krb5.conf
```

1.2 find

find 是实时查找工具，通过遍历指定路径完成文件查找

工作特点：

- 查找速度略慢
- 精确查找
- 实时查找
- 查找条件丰富
- 可能只搜索用户具备读取和执行权限的目录

格式：

```
find [OPTION]... [查找路径] [查找条件] [处理动作]
```

查找路径：指定具体目标路径；默认为当前目录

查找条件：指定的查找标准，可以文件名、大小、类型、权限等标准进行；默认为找出指定路径下的所有文件

处理动作：对符合条件的文件做操作，默认输出至屏幕

1.2.1 指定搜索目录层级

```
-maxdepth level 最大搜索目录深度，指定目录下的文件为第1级
-mindepth level 最小搜索目录深度
```

范例：

```
find /etc -maxdepth 2 -mindepth 2
```

1.2.2 对每个目录先处理目录内的文件，再处理目录本身

```
-depth
-d #warning: the -d option is deprecated; please use -depth instead, because the
latter is a POSIX-compliant feature
```

范例：

```
[root@centos8 data]#tree /data/test
```

```
/data/test
├─ f1.txt
├─ f2.txt
├─ test2
│   └─ test3
│       ├── f3.txt
│       └─ f4.txt
```

4 directories, 2 files

```
[root@centos8 data]#find /data/test
```

```
/data/test
/data/test/f1.txt
/data/test/f2.txt
/data/test/test2
/data/test/test2/test3
/data/test/test2/test3/f3.txt
/data/test/test2/test3/f4.txt
```

```
[root@centos8 data]#find /data/test -depth
```

```
/data/test/f1.txt
/data/test/f2.txt
/data/test/test2/test3/f3.txt
/data/test/test2/test3/f4.txt
/data/test/test2/test3
/data/test/test2
/data/test
```

1.2.3 根据文件名和inode查找

<code>-name "文件名称"</code>	#支持使用glob, 如: *, ?, [], [^], 通配符要加双引号引起来
<code>-iname "文件名称"</code>	#不区分字母大小写
<code>-inum n</code>	#按inode号查找
<code>-samefile name</code>	#相同inode号的文件
<code>-links n</code>	#链接数为n的文件
<code>-regex "PATTERN"</code>	#以PATTERN匹配整个文件路径, 而非文件名称

范例:

```
find -name snow.png
find -iname snow.png
find / -name ".txt"
find /var -name "log*"

[root@centos8 data]#find -regex ".*\.txt$"
./scripts/b.txt
./f1.txt
```

1.2.4 根据属主、属组查找

<code>-user USERNAME</code>	<code>#查找属主为指定用户(UID)的文件</code>
<code>-group GRPNAME</code>	<code>#查找属组为指定组(GID)的文件</code>
<code>-uid UserID</code>	<code>#查找属主为指定的UID号的文件</code>
<code>-gid GroupID</code>	<code>#查找属组为指定的GID号的文件</code>
<code>-nouser</code>	<code>#查找没有属主的文件</code>
<code>-nogroup</code>	<code>#查找没有属组的文件</code>

1.2.5 根据文件类型查找

`-type TYPE`

TYPE可以是以下形式:

f: 普通文件
d: 目录文件
l: 符号链接文件
s: 套接字文件
b: 块设备文件
c: 字符设备文件
p: 管道文件

范例:

`#查看/home的目录`
`find /home -type d -ls`

1.2.6 空文件或目录

`-empty`

范例:

`[root@centos8 ~]#find /app -type d -empty`

1.2.7 组合条件

与: `-a` , 默认多个条件是与关系
或: `-o`
非: `-not` !

范例:

```
[root@centos8 ~]#find /etc/ -type d -o -type l |wc -l
307
[root@centos8 ~]#find /etc/ -type d -o -type l -ls |wc -l
101
[root@centos8 ~]#find /etc/ \( -type d -o -type l \) -ls |wc -l
307
```

德·摩根定律:

- (非 A) 或 (非 B) = 非(A 且 B)
- (非 A) 且 (非 B) = 非(A 或 B)

示例:

```
!A -a !B = !(A -o B)
!A -o !B = !(A -a B)
```

范例:

```
find -user joe -group joe
find -user joe -not -group joe
find -user joe -o -user jane
find -not \( -user joe -o -user jane \)
find / -user joe -o -uid 500
```

范例:

```
[root@centos8 data]#find ! \( -type d -a -empty \)| wc -l
56
[root@centos8 data]#find ! -type d -o ! -empty |wc -l
56

[mage@centos8 data]$find ! -user wang ! -user mage
.
./script40
./script40/backup
./script40/backup/args.sh
./script40/backup/chook_rabbit.sh
./script40/backup/disk_check.sh
./script40/backup/ping.sh
./script40/backup/systeminfo.sh
./script40/backup/test_read.sh
./script40/backup/test.sh
./script40/if_bmi.sh
./script40/case_yesorno.sh
./script40/case_yesorno2.sh
./script40/b.txt
./f1.txt
./test
./test/f1.txt.link
./f1.txtlink
./test2

[root@centos8 home]#ll
total 0
drwx-----. 2 mage      mage      62 Jan 16 17:53 mage
drwx-----. 2 wang      wang       62 Jan 16 10:43 wang
drwx-----. 2 xiaoming  xiaoming 62 Apr  6 09:51 xiaoming
[root@centos8 home]#find ! \( -user wang -o -user mage \)
.
./xiaoming
./xiaoming/.bash_logout
./xiaoming/.bash_profile
./xiaoming/.bashrc
[root@centos8 home]#find ! -user wang -a ! -user mage
```

```
.
./xiaoming
./xiaoming/.bash_logout
./xiaoming/.bash_profile
./xiaoming/.bashrc
```

#找出/tmp目录下, 属主不是root, 且文件名不以f开头的文件

```
find /tmp \( -not -user root -a -not -name 'f*' \) -ls
find /tmp -not \( -user root -o -name 'f*' \) -ls
```

1.2.8 排除目录

范例:

#查找/etc/下, 除/etc/security目录的其它所有.conf后缀的文件

```
find /etc -path '/etc/security' -a -prune -o -name "*.conf"
```

#查找/etc/下, 除/etc/security和/etc/systemd,/etc/dbus-1三个目录的所有.conf后缀的文件

```
find /etc \( -path "/etc/security" -o -path "/etc/systemd" -o -path "/etc/dbus-1" \) -a -prune -o -name "*.conf"
```

#排除/proc和/sys目录

```
find / \( -path "/sys" -o -path "/proc" \) -a -prune -o -type f -a -mmin -1
```

1.2.9 根据文件大小来查找

-size [+ -]#UNIT	#常用单位: k, M, G, c (byte), 注意大小写敏感
#UNIT:	#表示[#-1, #], 如: 6k 表示(5k, 6k]
-#UNIT	#表示[0, #-1], 如: -6k 表示[0, 5k]
+#UNIT	#表示[# , ∞), 如: +6k 表示(6k, ∞)

范例:

```
find / -size +10G
```

```
[root@centos8 ~]#find / -size +10G
/proc/kcore
```

```
find: '/proc/25229/task/25229/fd/6': No such file or directory
```

```
find: '/proc/25229/task/25229/fdinfo/6': No such file or directory
```

```
find: '/proc/25229/fd/5': No such file or directory
```

```
find: '/proc/25229/fdinfo/5': No such file or directory
```

```
[root@centos8 ~]#ll -h /proc/kcore
```

```
-r----- 1 root root 128T Dec 14 2020 /proc/kcore
```

```
[root@centos8 ~]#du -sh /proc/kcore
```

```
0 /proc/kcore
```

1.2.10 根据时间戳

#以“天”为单位

```
-atime [+|-]#
```

#表示[# , #+1)

+# #表示[#+1, ∞]

-# #表示[0,#)

-mtime

-ctime

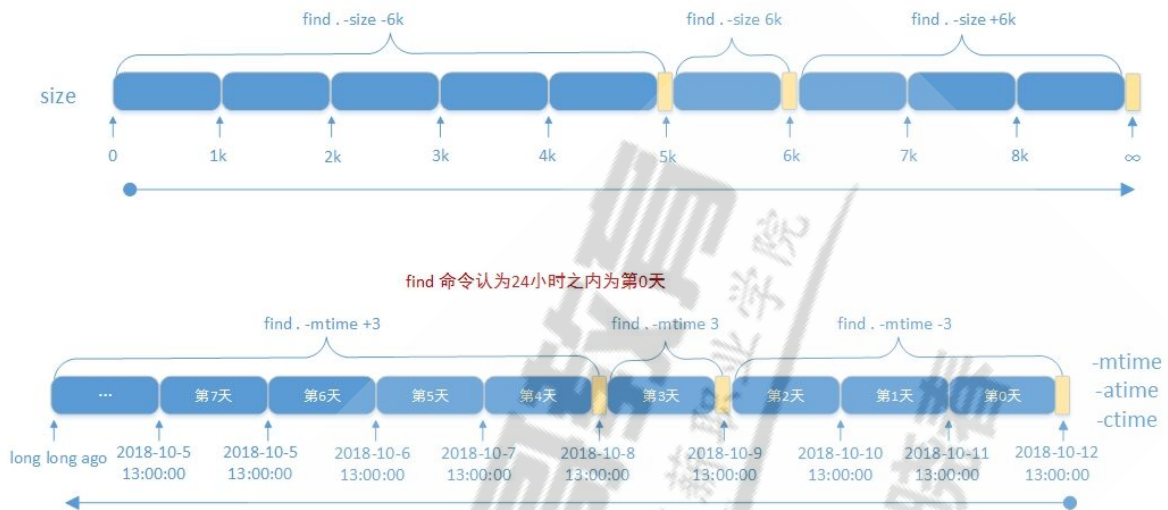
#以“分钟”为单位

-amin

-mmin

-cmin

Find 查找条件助记



1.2.11 根据权限查找

-perm [/|-]MODE

MODE #精确权限匹配

/MODE #任何一类(u,g,o)对象的权限中只要能一位匹配即可，或关系，+ 从CentOS 7开始淘汰

-MODE #每一类对象都必须同时拥有指定权限，与关系

0 表示不关注

说明：

`find -perm 755` 会匹配权限模式恰好是755的文件

只要当任意人有写权限时，`find -perm /222`就会匹配

只有当每个人都有写权限时，`find -perm -222`才会匹配

只有当其它人 (other) 有写权限时，`find -perm -002`才会匹配

1.2.12 正则表达式

-regextype type

Changes the regular expression syntax understood by **-regex** and **-iregex** tests which occur later on the command line. Currently-implemented types are emacs (this is the default), posix-awk, posix-basic, posix-egrep and posix-extended.

-regex pattern

File name matches regular expression pattern. This is a match on the whole path, not a search. For example, to match a file named `./fubar3`, you can use the regular expression `.*bar.` or `.*b.*3`, but not `f.*r3`. The regular expressions understood by `find` are by default Emacs Regular Expressions, but this can be changed with the **-regextype** option.

范例:

```
find /you/find/dir -regextype posix-extended -regex "regex"
```

1.2.13 处理动作

-print: 默认的处理动作, 显示至屏幕

-ls: 类似于对查找到的文件执行 `"ls -dils"` 命令格式输出

-fls file: 查找到的所有文件的长格式信息保存至指定文件中, 相当于 `-ls > file`

-delete: 删除查找到的文件, 慎用!

-ok COMMAND {} \; 对查找到的每个文件执行由COMMAND指定的命令, 对于每个文件执行命令之前, 都会交互式要求用户确认

-exec COMMAND {} \; 对查找到的每个文件执行由COMMAND指定的命令

{ }: 用于引用查找到的文件名称自身

范例:

```
#备份配置文件, 添加.orig这个扩展名
find -name ".conf" -exec cp {} {}.orig \;
#提示删除存在时间超过3天以上的joe的临时文件
find /tmp -ctime +3 -user joe -ok rm {} \;
#在主目录中寻找可被其它用户写入的文件
find ~ -perm -002 -exec chmod o-w {} \;
#查找/data下的权限为644, 后缀为sh的普通文件, 增加执行权限
find /data -type f -perm 644 -name "*.sh" -exec chmod 755 {} \;
```

1.3 参数替换 xargs

由于很多命令不支持管道|来传递参数, xargs用于产生某个命令的参数, xargs可以读入stdin的数据, 并且以空格符或回车符将stdin的数据分隔成为参数

另外, 许多命令不能接受过多参数, 命令执行可能会失败, xargs可以解决

注意: 文件名或者是其他意义的名词内含有空格符的情况

find 经常和 xargs 命令进行组合, 形式如下:

```
find | xargs COMMAND
```

范例: :

#显示10个数字

```
[root@centos8 ~]#seq 10 | xargs
```

```
1 2 3 4 5 6 7 8 9 10
```

#删除当前目录下的大量文件

```
ls | xargs rm
```

#

```
find -name "*.sh" | xargs ls -l
```

```
[root@centos8 data]#echo {1..10} |xargs
```

```
1 2 3 4 5 6 7 8 9 10
```

```
[root@centos8 data]#echo {1..10} |xargs -n1
```

```
1
2
3
4
5
6
7
8
9
10
```

```
[root@centos8 data]#echo {1..10} |xargs -n2
```

```
1 2
3 4
5 6
7 8
9 10
```

#批量创建和删除用户

```
echo user{1..10} |xargs -n1 useradd
```

```
echo user{1..100} | xargs -n1 userdel -r
```

#这个命令是错误的

```
find /sbin/ -perm /700 | ls -l
```

#查找有特殊权限的文件，并排序

```
find /bin/ -perm /7000 | xargs ls -l
```

#此命令和上面有何区别？

```
find /bin/ -perm -7000 | xargs ls -l
```

#以字符nul分隔

```
find -type f -name "*.txt" -print0 | xargs -0 rm
```

#并发执行多个进程

```
seq 100 |xargs -i -P10 wget -P /data http://10.0.0.8/{ }.html
```

#并行下载bilibili视频

```
yum install python3-pip -y
```

```
pip3 install you-get
```

```
seq 10 | xargs -i -P3 you-get https://www.bilibili.com/video/BV1HZ4y1p7Bf?p={}
```

练习

- 1、查找/var目录下属主为root，且属组为mail的所有文件
- 2、查找/var目录下不属于root、lp、gdm的所有文件
- 3、查找/var目录下最近一周内其内容修改过，同时属主不为root，也不是postfix的文件
- 4、查找当前系统上没有属主或属组，且最近一个周内曾被访问过的文件
- 5、查找/etc目录下大于1M且类型为普通文件的所有文件
- 6、查找/etc目录下所有用户都没有写权限的文件
- 7、查找/etc目录下至少有一类用户没有执行权限的文件
- 8、查找/etc/init.d目录下，所有用户都有执行权限，且其它用户有写权限的文件

2 压缩和解压缩

主要针对单个文件压缩,而非目录

2.1 compress 和 uncompress

此工具来自于ncompress包,此工具目前已经很少使用

对应的文件是 .Z 后缀

格式

```
compress Options [file ...]  
uncompress file.Z
```

#解压缩

常用选项

```
-d 解压缩，相当于uncompress  
-c 结果输出至标准输出,不删除原文件  
-v 显示详情
```

zcat file.Z 不显式解压缩的前提下查看文本文件内容

范例：

```
zcat file.Z >file
```

2.2 gzip和gunzip

来自于 gzip 包

对应的文件是 .gz 后缀

格式：

```
gzip [OPTION]... FILE ...
```

常用选项：

- k keep, 保留原文件,CentOS 8 新特性
- d 解压缩, 相当于gunzip
- c 结果输出至标准输出, 保留原文件不改变
- # 指定压缩比, #取值为1-9, 值越大压缩比越大

范例:

```
#解压缩
gunzip file.gz

#不显式解压缩的前提下查看文本文件内容
zcat file.gz
```

范例:

```
gzip -c messages >messages.gz
gzip -c -d messages.gz > messages
zcat messages.gz > messages
cat messages | gzip > m.gz
```

2.3 bzip2和bunzip2

来自于 bzip2 包

对应的文件是 .bz2 后缀

格式:

```
bzip2 [OPTION]... FILE ...
```

常用选项

- k keep, 保留原文件
- d 解压缩
- c 结果输出至标准输出, 保留原文件不改变
- # 1-9, 压缩比, 默认为9

范例:

```
bunzip2 file.bz2    解压缩
bzipcat file.bz2    不显式解压缩的前提下查看文本文件内容
```

2.4 xz 和 unxz

来自于 xz 包

对应的文件是 .xz 后缀

格式

```
xz [OPTION]... FILE ...
```

常用选项

- k keep, 保留原文件
- d 解压缩
- c 结果输出至标准输出, 保留原文件不改变
- # 压缩比, 取值1-9, 默认为6

范例:

```
unxz file.xz      #解压缩
xzcat file.xz     #不显式解压缩的前提下查看文本文件内容
```

2.5 zip 和 unzip

zip 可以实现打包目录和多个文件成一个文件并压缩, 但可能会丢失文件属性信息, 如: 所有者和组信息, 一般建议使用 tar 代替

分别来自于 zip 和 unzip 包

对应的文件是 .zip 后缀

范例: zip帮助

```
[root@centos8 ~]#zip
Copyright (c) 1990-2008 Info-ZIP - Type 'zip "-L"' for software license.
Zip 3.0 (July 5th 2008). Usage:
zip [-options] [-b path] [-t mmdyyy] [-n suffixes] [zipfile list] [-xi list]
  The default action is to add or replace zipfile entries from list, which
  can include the special name - to compress standard input.
  If zipfile and list are omitted, zip compresses stdin to stdout.
  -f  freshen: only changed files  -u  update: only changed or new files
  -d  delete entries in zipfile    -m  move into zipfile (delete OS files)
  -r  recurse into directories     -j  junk (don't record) directory names
  -0  store only                   -l  convert LF to CR LF (-ll CR LF to LF)
  -1  compress faster              -9  compress better
  -q  quiet operation              -v  verbose operation/print version info
  -c  add one-line comments        -Z  add zipfile comment
  -@  read names from stdin         -o  make zipfile as old as latest entry
  -x  exclude the following names  -i  include only the following names
  -F  fix zipfile (-FF try harder) -D  do not add directory entries
  -A  adjust self-extracting exe    -J  junk zipfile prefix (unzipsfx)
  -T  test zipfile integrity        -X  exclude extra file attributes
  -y  store symbolic links as the link instead of the referenced file
  -e  encrypt                      -n  don't compress these suffixes
  -h2 show more help
```

```
[root@centos8 ~]#zip -h2
```

Extended Help for Zip

See the Zip Manual for more detailed help

zip stores files in zip archives. The default action is to add or replace zipfile entries.

Basic command line:

```
zip options archive_name file file ...
```

Some examples:

```
Add file.txt to z.zip (create z if needed):    zip z file.txt
zip all files in current dir:                  zip z *
zip files in current dir and subdirs also:     zip -r z .
```

范例:

#打包并压缩

```
zip -r /backup/sysconfig.zip /etc/sysconfig/
```

#不包括目录本身, 只打包目录内的文件和子目录

```
cd /etc/sysconfig; zip -r /root/sysconfig.zip *
```

#默认解压缩至当前目录

```
unzip /backup/sysconfig.zip
```

#解压缩至指定目录, 如果指定目录不存在, 会在其父目录 (必须事先存在) 下自动生成

```
unzip /backup/sysconfig.zip -d /tmp/config
```

```
cat /var/log/messages | zip messages -
```

#-p 表示管道

```
unzip -p message.zip > message
```

范例: 交互式加密和解密

```
[root@centos8 magedu]#zip -e magedu.zip *
Enter password:
Verify password:
  adding: ca.crt (deflated 25%)
  adding: client.ovpn (deflated 27%)
  adding: dh.pem (deflated 19%)
  adding: magedu.crt (deflated 45%)
  adding: magedu.key (deflated 24%)
  adding: magedu.tar (deflated 72%)
  adding: ta.key (deflated 40%)
[root@centos8 magedu]#mv magedu.zip /root
[root@centos8 magedu]#cd
[root@centos8 ~]#unzip magedu.zip
Archive:  magedu.zip
[magedu.zip] ca.crt password:
password incorrect--reenter: [root@centos8 ~]#

[root@centos8 ~]#unzip magedu.zip
Archive:  magedu.zip
[magedu.zip] ca.crt password:
  inflating: ca.crt
  inflating: client.ovpn
  inflating: dh.pem
  inflating: magedu.crt
  inflating: magedu.key
  inflating: magedu.tar
```



```
inflating: ta.key
```

范例: 非交互式加密和解密

```
[root@centos8 test]#zip -P 123456 magedu.zip *
adding: ca.crt (deflated 25%)
adding: client.ovpn (deflated 27%)
adding: dh.pem (deflated 19%)
adding: ta.key (deflated 40%)
[root@centos8 test]#mv magedu.zip /opt
[root@centos8 test]#cd /opt
[root@centos8 opt]#unzip -P 123456 magedu.zip
Archive: magedu.zip
[magedu.zip] ca.crt password:
inflating: ca.crt
inflating: client.ovpn
inflating: dh.pem
inflating: ta.key
```

3 打包和解包

3.1 tar

tar 即 Tape ARchive 磁带归档, 可以对目录和多个文件打包成一个文件, 并且可以压缩, 保留文件属性不丢失, 常用于备份功能, 推荐使用

对应的文件是 .tar 后缀

格式

```
tar [-ABcdgGhiklmMoOpPrRsStuUvwXzZ] [-b <区块数目>] [-C <目的目录>] [-f <备份文件>] [-F
<Script文件>] [-K <文件>] [-L <媒体容量>] [-N <日期时间>] [-T <范本文件>] [-V <卷册名称>] [-X
<范本文件>] [-<设备编号>] [-<存储密度>] [--after-date=<日期时间>] [--atime-preserve] [--
backup=<备份方式>] [--checkpoint] [--concatenate] [--confirmation] [--delete] [--
exclude=<范本样式>] [--force-local] [--group=<群组名称>] [--help] [--ignore-failed-
read] [--new-volume-script=<Script文件>] [--newer-mtime] [--no-recursion] [--null] [--
numeric-owner] [--owner=<用户名称>] [--posix] [--serve] [--preserve-order] [--preserve-
permissions] [--record-size=<区块数目>] [--recursive-unlink] [--remove-files] [--rsh-
command=<执行指令>] [--same-owner] [--suffix=<备份字尾字符串>] [--totals] [--use-
compress-program=<执行指令>] [--version] [--volno-file=<编号文件>] [文件或目录...]
```

选项:

- A或--catenate 新增文件到已存在的备份文件。
- b<区块数目>或--blocking-factor=<区块数目> 设置每笔记录的区块数目, 每个区块大小为12Bytes。
- B或--read-full-records 读取数据时重设区块大小。
- c或--create 建立新的备份文件。
- C<目的目录>或--directory=<目的目录> 切换到指定的目录。
- d或--diff或--compare 对比备份文件内和文件系统上的文件的差异。
- f<备份文件>或--file=<备份文件> 指定备份文件。
- F<Script文件>或--info-script=<Script文件> 每次更换磁带时, 就执行指定的Script文件。
- g或--listed-incremental 处理GNU格式的大量备份。

-G或--incremental 处理旧的GNU格式的大量备份。

-h或--dereference 不建立符号连接，直接复制该连接所指向的原始文件。

-i或--ignore-zeros 忽略备份文件中的0 Byte区块，也就是EOF。

-k或--keep-old-files 解开备份文件时，不覆盖已有的文件。

-K<文件>或--starting-file=<文件> 从指定的文件开始还原。

-l或--one-file-system 复制的文件或目录存放的文件系统，必须与tar指令执行时所处的文件系统相同，否则不予复制。

-L<媒体容量>或--tape-length=<媒体容量> 设置存放每体的容量，单位以1024 Bytes计算。

-m或--modification-time 还原文件时，不变更文件的更改时间。

-M或--multi-volume 在建立，还原备份文件或列出其中的内容时，采用多卷册模式。

-N<日期格式>或--newer=<日期时间> 只将较指定日期更新的文件保存到备份文件里。

-o或--old-archive或--portability 将资料写入备份文件时使用v7格式。

-O或--stdout 把从备份文件里还原的文件输出到标准输出设备。

-p或--same-permissions 用原来的文件权限还原文件。

-P或--absolute-names 文件名使用绝对名称，不移除文件名称前的"/"号。

-r或--append 新增文件到已存在的备份文件的结尾部分。

-R或--block-number 列出每个信息在备份文件中的区块编号。

-s或--same-order 还原文件的顺序和备份文件内的存放顺序相同。

-S或--sparse 倘若一个文件内含大量的连续0字节，则将此文件存成稀疏文件。

-t或--list 列出备份文件的内容。

-T<范本文件>或--files-from=<范本文件> 指定范本文件，其内含有一个或多个范本样式，让tar解开或建立符合设置条件的文件。

-u或--update 仅置换较备份文件内的文件更新的文件。

-U或--unlink-first 解开压缩文件还原文件之前，先解除文件的连接。

-v或--verbose 显示指令执行过程。

-V<卷册名称>或--label=<卷册名称> 建立使用指定的卷册名称的备份文件。

-w或--interactive 遇到问题时先询问用户。

-W或--verify 写入备份文件后，确认文件正确无误。

-x或--extract或--get 从备份文件中还原文件。

-X<范本文件>或--exclude-from=<范本文件> 指定范本文件，其内含有一个或多个范本样式，让ar排除符合设置条件的文件。

-z或--gzip或--ungzip 通过gzip指令处理备份文件。

-Z或--compress或--uncompress 通过compress指令处理备份文件。

-<设备编号><存储密度> 设置备份用的外围设备编号及存放数据的密度。

--after-date=<日期时间> 此参数的效果和指定"-N"参数相同。

--atime-preserve 不变更文件的存取时间。

--backup=<备份方式>或--backup 移除文件前先进行备份。

--checkpoint 读取备份文件时列出目录名称。

--concatenate 此参数的效果和指定"-A"参数相同。

--confirmation 此参数的效果和指定"-w"参数相同。

--delete 从备份文件中删除指定的文件。

--exclude=<范本样式> 排除符合范本样式的文件。

--group=<群组名称> 把加入设备文件中的文件的所属群组设成指定的群组。

--help 在线帮助。

--ignore-failed-read 忽略数据读取错误，不中断程序的执行。

--new-volume-script=<Script文件> 此参数的效果和指定"-F"参数相同。

--newer-mtime 只保存更改过的文件。

--no-recursion 不做递归处理，也就是指定目录下的所有文件及子目录不予处理。

--null 从null设备读取文件名称。

--numeric-owner 以用户识别码及群组识别码取代用户名称和群组名称。

--owner=<用户名称> 把加入备份文件中的文件的拥有者设成指定的用户。

--posix 将数据写入备份文件时使用POSIX格式。

--preserve 此参数的效果和指定"-ps"参数相同。

--preserve-order 此参数的效果和指定"-A"参数相同。

--preserve-permissions 此参数的效果和指定"-p"参数相同。

--record-size=<区块数目> 此参数的效果和指定"-b"参数相同。

--recursive-unlink 解开压缩文件还原目录之前，先解除整个目录下所有文件的连接。

--remove-files 文件加入备份文件后，就将其删除。

--rsh-command=<执行指令> 设置要在远端主机上执行的指令，以取代rsh指令。
--same-owner 尝试以相同的文件拥有者还原文件。
--suffix=<备份字尾字符串> 移除文件前先行备份。
--totals 备份文件建立后，列出文件大小。
--use-compress-program=<执行指令> 通过指定的指令处理备份文件。
--version 显示版本信息。
--volno-file=<编号文件> 使用指定文件内的编号取代预设的卷册编号。

(1) 创建归档，保留权限

```
tar -cpvf /PATH/FILE.tar FILE...
```

(2) 追加文件至归档：注：不支持对压缩文件追加

```
tar -rf /PATH/FILE.tar FILE...
```

(3) 查看归档文件中的文件列表

```
tar -t -f /PATH/FILE.tar
```

(4) 展开归档

```
tar xf /PATH/FILE.tar  
tar xf /PATH/FILE.tar -C /PATH/
```

(5) 结合压缩工具实现：归档并压缩

-z 相当于gzip压缩工具
-j 相当于bzip2压缩工具
-J 相当于xz压缩工具

范例：

```
[root@centos8 ~]#tar zcvf etc.tar.gz /etc/  
[root@centos8 ~]#tar jcvf etc.tar.bz2 /etc/  
[root@centos8 ~]#tar Jcvf etc.tar.xz /etc/  
[root@centos8 ~]#ll etc.tar.*  
-rw-r--r-- 1 root root 3645926 Dec 20 22:00 etc.tar.bz2  
-rw-r--r-- 1 root root 5105347 Dec 20 21:59 etc.tar.gz  
-rw-r--r-- 1 root root 3101616 Dec 20 22:00 etc.tar.xz
```

#只打包目录内的文件，不所括目录本身

```
[root@centos8 ~]#cd /etc  
[root@centos8 etc]#tar zcvf /root/etc.tar.gz ./
```

#利用 tar 进行文件复制

```
[root@centos8 ~]#tar c /data/ | tar x -C /backup  
tar: Removing leading `/' from member names
```

--exclude 排除文件

范例：

```
tar zcvf /root/a.tgz --exclude=/app/host1 --exclude=/app/host2 /app
```

- T 选项指定输入文件
- X 选项指定包含要排除的文件列表

范例:

```
tar zcvf mybackup.tgz -T /root/includefilelist -X /root/excludefilelist
```

3.2 split

split 命令可以分割一个文件为多个文件

范例:

```
#分割大的 tar 文件为多份小文件  
split -b size -d tar-file-name prefix-name
```

示例:

```
split -b 1M mybackup.tgz mybackup-parts
```

#切换成的多个小分文件使用数字后缀

```
split -b 1M -d mybackup.tgz mybackup-parts
```

将多个切割的小文件合并成一个大文件

```
cat mybackup-parts* > mybackup.tar.gz
```

3.3 cpio

cpio 是历史悠久的打包和解包工具，不过目前也已较少使用

cpio 命令是通过重定向的方式将文件进行打包备份，还原恢复的工具，它可以解压以“.cpio”或者“.tar”结尾的文件

格式:

```
cpio [选项] > 文件名或者设备名  
cpio [选项] < 文件名或者设备名
```

常用选项

- o output模式，打包，将标准输入传入的文件名打包后发送到标准输出
- i input模式，解包，对标准输入传入的打包文件名解包到当前目录
- t 预览，查看标准输入传入的打包文件中包含的文件列表
- O filename 输出到指定的归档文件名
- A 向已存在的归档文件中追加文件
- I filename 对指定的归档文件名解压
- F filename 使用指定的文件名替代标准输入或输出
- d 解包生成目录，在cpio还原时，自动的建立目录
- v 显示打包过程中的文件名称

范例：

```
#将etc目录备份
find ./etc -print | cpio -ov > bak.cpio
#将/data内容追加bak.cpio
find /data | cpio -oA -F bak.cpio
#内容预览
cpio -tv < etc.cpio
#解包文件
cpio -idv < etc.cpio
```



祝大家学业有成

谢 谢

讲师：王晓春
邮箱：29308620@qq.com
电话：400-080-6560