

tar

将许多文件一起保存至一个单独的磁带或磁盘归档，并能从归档中单独还原所需文件。

补充说明

tar命令可以为linux的文件和目录创建档案。利用tar[]可以为某一特定文件创建档案（备份文件），也可以在档案中改变文件，或者向档案中加入新的文件[]tar最初被用来在磁带上创建档案，现在，用户可以在任何设备上创建档案。利用tar命令，可以把一大堆的文件和目录全部打包成一个文件，这对于备份文件或几个文件组合成为一个文件以便于网络传输是非常有用的。

首先要弄清两个概念：打包和压缩。打包是指将一大堆文件或目录变成一个总的文件；压缩则是将一个大的文件通过一些压缩算法变成一个小文件。

为什么要区分这两个概念呢？这源于Linux中很多压缩程序只能针对一个文件进行压缩，这样当你想要压缩一大堆文件时，你得先将这一大堆文件先打成一个包[]tar命令），然后再用压缩程序进行压缩[]gzip bzip2命令）。

语法

```
tar [选项...] [FILE]...
```

选项

```
-A, --catenate, --concatenate 追加 tar 文件至归档
-c, --create                    创建一个新归档
-d, --diff, --compare          找出归档和文件系统的差异
--delete                       从归档(非磁带!)中删除
-r, --append                   追加文件至归档结尾
-t, --list                     列出归档内容
--test-label                   测试归档卷标并退出
-u, --update                   仅追加比归档中副本更新的文件
-x, --extract, --get          从归档中解出文件
```

操作修饰符:

```
--check-device                当创建增量归档时检查设备号(默认)
-g, --listed-incremental=FILE 处理新式的 GNU 格式的增量备份
-G, --incremental            处理老式的 GNU 格式的增量备份
--ignore-failed-read         当遇上不可读文件时不要以非零值退出
--level=NUMBER              所创建的增量列表归档的输出级别
-n, --seek                   归档可检索
--no-check-device            当创建增量归档时不要检查设备号
--no-seek                   归档不可检索
--occurrence[=NUMBER]      仅处理归档中每个文件的第 NUMBER
                             个事件；仅当与以下子命令 --delete,
                             --diff, --extract 或是 --list
                             中的一个联合使用时，此选项才有效。而且不管文件列表是以命令行形式给出或是通过
                             -T 选项指定的[]NUMBER 值默认为 1
--sparse-version=MAJOR[.MINOR]
                             设置所用的离散格式版本(隐含)
```



```

--numeric-owner      总是以数字代表用户/组的名称
--owner=名称      强制将 NAME
                   作为所添加的文件的所有者
-p, --preserve-permissions, --same-permissions
                   解压文件权限信息(默认只为超级用户服务)
--preserve          与 -p 和 -s 一样
--same-owner        尝试解压时保持所有者关系一致(超级用户默认此项)
-s, --preserve-order, --same-order
                   member arguments are listed in the same order
as
                   the files in the archive

```

Handling of extended file attributes:

```

--acls              Enable the POSIX ACLs support
--no-acls           Disable the POSIX ACLs support
--no-selinux        Disable the SELinux context support
--no-xattrs         Disable extended attributes support
--selinux           Enable the SELinux context support
--xattrs            Enable extended attributes support
--xattrs-exclude=MASK specify the exclude pattern for xattr keys
--xattrs-include=MASK specify the include pattern for xattr keys

```

设备选择和切换:

```

-f, --file=ARCHIVE      使用归档文件或 ARCHIVE 设备
--force-local           即使归档文件存在副本还是把它认为是本地归档
-F, --info-script=名称, --new-volume-script=名称
                       在每卷磁带最后运行脚本(隐含 -M)
-L, --tape-length=NUMBER 写入 NUMBER × 1024 字节后更换磁带
-M, --multi-volume      创建/列出/解压多卷归档文件
--rmt-command=COMMAND  使用指定的 rmt COMMAND 代替 rmt
--rsh-command=COMMAND  使用远程 COMMAND 代替 rsh
--volno-file=FILE      使用/更新 FILE 中的卷数

```

设备分块:

```

-b, --blocking-factor=BLOCKS 每个记录 BLOCKS × 512 字节
-B, --read-full-records      读取时重新分块(只对 4.2BSD 管道有效)
-i, --ignore-zeros          忽略归档中的零字节块(即文件结尾)
--record-size=NUMBER        每个记录的字节数 NUMBER 乘以 512

```

选择归档格式:

```

-H, --format=FORMAT      创建指定格式的归档

```

FORMAT 是以下格式中的一种:

```

gnu              GNU tar 1.13.x 格式
oldgnu           GNU 格式 as per tar <= 1.12

```

pax	POSIX 1003.1-2001 (pax) 格式
posix	等同于 pax
ustar	POSIX 1003.1-1988 (ustar) 格式
v7	old V7 tar 格式

--old-archive, --portability

等同于 --format=v7

--pax-option=关键字[:=值][,关键字[:=值]]...

控制 pax 关键字

--posix 等同于 --format=posix

-V, --label=TEXT 创建带有卷名 TEXT

的归档；在列出/解压时，使用 TEXT

作为卷名的模式串

压缩选项:

-a, --auto-compress 使用归档后缀名来决定压缩程序

-I, --use-compress-program=PROG

通过 PROG 过滤(必须是能接受 -d
选项的程序)

-j, --bzip2 通过 bzip2 过滤归档

-J, --xz 通过 xz 过滤归档

--lzip 通过 lzip 过滤归档

--lzma 通过 lzma 过滤归档

--lzop

--no-auto-compress 不使用归档后缀名来决定压缩程序

-z, --gzip, --gunzip, --ungzip 通过 gzip 过滤归档

-Z, --compress, --uncompress 通过 compress 过滤归档

本地文件选择:

--add-file=FILE 添加指定的 FILE 至归档(如果名字以 -
开始会很有用的)

--backup[=CONTROL] 在删除前备份，选择 CONTROL 版本

-C, --directory=DIR 改变至目录 DIR

--exclude=PATTERN 排除以 PATTERN 指定的文件

--exclude-backups 排除备份和锁文件

--exclude-caches 除标识文件本身外，排除包含

CACHEDIR.TAG 的目录中的内容

--exclude-caches-all 排除包含 CACHEDIR.TAG 的目录

--exclude-caches-under 排除包含 CACHEDIR.TAG 的目录中所有内容

--exclude-tag=FILE 除 FILE 自身外，排除包含 FILE
的目录中的内容

--exclude-tag-all=FILE 排除包含 FILE 的目录

--exclude-tag-under=FILE 排除包含 FILE 的目录中的所有内容

--exclude-vcs 排除版本控制系统目录

-h, --dereference

跟踪符号链接；将它们所指向的文件归档并输出

--hard-dereference

跟踪硬链接；将它们所指向的文件归档并输出

-K, --starting-file=MEMBER-NAME

```

begin at member MEMBER-NAME when reading the
archive
--newer-mtime=DATE    当只有数据改变时比较数据和时间
--no-null             禁用上一次的效果 --null 选项
--no-recursion        避免目录中的自动降级
--no-unquote          不以 -T 读取的文件名作为引用结束
--null               -T 读取以空终止的名字 -C 禁用
-N, --newer=DATE-OR-FILE, --after-date=DATE-OR-FILE
    只保存比 DATE-OR-FILE 更新的文件
--one-file-system     创建归档时保存在本地文件系统中
-P, --absolute-names  不要从文件名中清除引导符 '/'
--recursion           目录递归(默认)
--suffix=STRING       在删除前备份, 除非被环境变量
SIMPLE_BACKUP_SUFFIX
    覆盖, 否则覆盖常用后缀(' ')
-T, --files-from=FILE  从 FILE
    中获取文件名来解压或创建文件
--unquote             以 -T
    读取的文件名作为引用结束(默认)
-X, --exclude-from=FILE  排除 FILE 中列出的模式串

```

文件名变换:

```

--strip-components=NUMBER  解压时从文件名中清除 NUMBER
    个引导部分
--transform=EXPRESSION, --xform=EXPRESSION
    使用 sed 代替 EXPRESSION
    来进行文件名变换

```

文件名匹配选项(同时影响排除和包括模式串):

```

--anchored             模式串匹配文件名头部
--ignore-case          忽略大小写
--no-anchored          模式串匹配任意 '/' 后字符(默认对
exclusion 有效)
--no-ignore-case       匹配大小写(默认)
--no-wildcards         逐字匹配字符串
--no-wildcards-match-slash  通配符不匹配 '/'
--wildcards            use wildcards (default)
--wildcards-match-slash
    通配符匹配 '/' (默认对排除操作有效)

```

提示性输出:

```

--checkpoint[=NUMBER]  每隔 NUMBER
    个记录显示进度信息(默认为 10 个)
--checkpoint-action=ACTION  在每个检查点上执行 ACTION
--full-time             print file time to its full resolution
--index-file=FILE      将详细输出发送至 FILE
-l, --check-links      只要不是所有链接都被输出就打印信息
--no-quote-chars=STRING  禁用来自 STRING 的字符引用
--quote-chars=STRING   来自 STRING 的额外的引用字符

```

```

--quoting-style=STYLE  设置名称引用风格；有效的 STYLE
                        值请参阅以下说明
-R, --block-number      每个信息都显示归档内的块数
--show-defaults         显示 tar 默认选项
--show-omitted-dirs     列表或解压时，列出每个不匹配查找标准的目录
--show-transformed-names, --show-stored-names
                        显示变换后的文件名或归档名
--totals[=SIGNAL]       处理归档后打印出总字节数；当此
                        SIGNAL 被触发时带参数 -
                        打印总字节数；允许的信号为：
                        SIGHUP SIGQUIT SIGINT SIGUSR1 和
                        SIGUSR2 同时也接受不带 SIG
                        前缀的信号名称
--utc                   以 UTC 格式打印文件修改时间
-v, --verbose           详细地列出处理的文件
--warning=KEYWORD       警告控制：
-w, --interactive, --confirmation
                        每次操作都要求确认

```

兼容性选项:

```

-o                       创建归档时，相当于
                        --old-archive 展开归档时，相当于
                        --no-same-owner

```

其它选项:

```

-?, --help              显示此帮助列表
--restrict              禁用某些潜在的有危险的选项
--usage                 显示简短的用法说明
--version               打印程序版本

```

长选项和相应短选项具有相同的强制参数或可选参数。

除非以 --suffix 或 SIMPLE_BACKUP_SUFFIX

设置备份后缀，否则备份后缀就是“~”。

可以用 --backup 或 VERSION_CONTROL 设置版本控制，可能的值为：

```

none, off               从不做备份
t, numbered             进行编号备份
nil, existing

```

如果编号备份存在则进行编号备份，否则进行简单备份

```

never, simple          总是使用简单备份

```

--quoting-style 选项的有效参数为:

```

literal
shell
shell-always
c
c-maybe
escape

```

```
locale
clocale
```

此 tar 默认为:

```
--format=gnu -f- -b20 --quoting-style=escape --rmt-command=/etc/rmt
--rsh-command=/usr/bin/ssh
```

参数

文件或目录：指定要打包的文件或目录列表。

实例

- z[]有gzip属性的
- j[]有bz2属性的
- Z[]有compress属性的
- v[]显示所有过程
- O[]将文件解开到标准输出

```
tar -cf archive.tar foo bar # 从文件 foo 和 bar 创建归档文件 archive.tar[]
tar -tvf archive.tar       # 详细列举归档文件 archive.tar 中的所有文件。
tar -xf archive.tar        # 展开归档文件 archive.tar 中的所有文件。
```

下面的参数-f是必须的

-f: 使用档案名字，切记，这个参数是最后一个参数，后面只能接档案名。

```
tar -cf all.tar *.jpg
# 这条命令是将所有.jpg的文件打成一个名为all.tar的包[] -c是表示产生新的包[] -f指定包的文件名。
```

```
tar -rf all.tar *.gif
# 这条命令是将所有.gif的文件增加到all.tar的包里面去[] -r是表示增加文件的意思。
```

```
tar -uf all.tar logo.gif
# 这条命令是更新原来tar包all.tar中logo.gif文件[] -u是表示更新文件的意思。
```

```
tar -tf all.tar
# 这条命令是列出all.tar包中所有文件[] -t是列出文件的意思
```

```
tar -cvf archive.tar foo bar # 从文件foo和bar创建archive.tar[]
tar -tvf archive.tar        # 详细列出archive.tar中的所有文件。
tar -xf archive.tar         # 从archive.tar提取所有文件。
```

zip格式

压缩[] zip -r [目标文件名].zip [原文件/目录名]

解压[] unzip [原文件名].zip

注：-r参数代表递归

tar格式（该格式仅仅打包，不压缩）

打包[] tar -cvf [目标文件名].tar [原文件名/目录名]

解包 `tar -xvf [原文件名].tar`

注：c参数代表create（创建）x参数代表extract（解包）v参数代表verbose（详细信息）f参数代表filename（文件名），所以f后必须接文件名。

tar.gz格式

方式一：利用前面已经打包好的tar文件，直接用压缩命令。

压缩 `gzip [原文件名].tar`

解压 `gunzip [原文件名].tar.gz`

方式二：一次性打包并压缩、解压并解包

打包并压缩 `tar -zcvf [目标文件名].tar.gz [原文件名/目录名]`

解压并解包 `tar -zxvf [原文件名].tar.gz`

注：z代表用gzip算法来压缩/解压。

tar.bz2格式

方式一：利用已经打包好的tar文件，直接执行压缩命令：

压缩 `bzip2 [原文件名].tar`

解压 `bunzip2 [原文件名].tar.bz2`

方式二：一次性打包并压缩、解压并解包

打包并压缩 `tar -jcvf [目标文件名].tar.bz2 [原文件名/目录名]`

解压并解包 `tar -jxvf [原文件名].tar.bz2`

注：小写j代表用bzip2算法来压缩/解压。

tar.xz格式

方式一：利用已经打包好的tar文件，直接用压缩命令：

压缩 `xz [原文件名].tar`

解压 `unxz [原文件名].tar.xz`

方式二：一次性打包并压缩、解压并解包

打包并压缩 `tar -Jcvf [目标文件名].tar.xz [原文件名/目录名]`

解压并解包 `tar -Jxvf [原文件名].tar.xz`

注：大写J代表用xz算法来压缩/解压。

tar.Z格式（已过时）

方式一：利用已经打包好的tar文件，直接用压缩命令：

压缩 `compress [原文件名].tar`

解压 `uncompress [原文件名].tar.Z`

方式二：一次性打包并压缩、解压并解包

打包并压缩 `tar -Zcvf [目标文件名].tar.Z [原文件名/目录名]`

解压并解包 `tar -Zxvf [原文件名].tar.Z`

注：大写Z代表用ncompress算法来压缩/解压。另ncompress是早期Unix系统的压缩格式，但由于ncompress的压缩率太低，现已过时。

jar格式

压缩 `jar -cvf [目标文件名].jar [原文件名/目录名]`

解压 `jar -xvf [原文件名].jar`

注：如果是打包的是Java类库，并且该类库中存在主类，那么需要写一个META-INF/MANIFEST.MF配置文件，内容如下：

```
Manifest-Version: 1.0
Created-By: 1.6.0_27 (Sun Microsystems Inc.)
Main-class: the_name_of_the_main_class_should_be_put_here
```

然后用如下命令打包：

`jar -cvfm [目标文件名].jar META-INF/MANIFEST.MF [原文件名/目录名]`

这样以后就能用“`java -jar [文件名].jar`”命令直接运行主类中的public static void main方法了。

7z格式

压缩 `7z a [目标文件名].7z [原文件名/目录名]`

解压 `7z x [原文件名].7z`

注：这个7z解压命令支持rar格式，即：

`7z x [原文件名].rar`

其它例子

将文件全部打包成tar包

```
tar -cvf log.tar log2012.log    仅打包，不压缩！
tar -zcvf log.tar.gz log2012.log  打包后，以 gzip 压缩
tar -jcvf log.tar.bz2 log2012.log  打包后，以 bzip2 压缩
```

在选项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选项，则以.tar.gz或.tar.gz来代表gzip压缩过的tar包；如果加j选项，则以.tar.bz2来作为tar包名。

解压目录

去掉第一层目录结构，要出除第二层 `--strip-components 2`

```
tar -xvf portal-web-v2.0.0.tar --strip-components 1 -C 指定目录
```

查阅上述tar包内有哪些文件

```
tar -ztvf log.tar.gz
```

由于我们使用 gzip 压缩的log.tar.gz，所以要查阅log.tar.gz包内的文件时，就得要加上z这个选项了。

将tar包解压缩

```
tar -zxvf /opt/soft/test/log.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

只将tar内的部分文件解压出来 □

```
tar -zxvf /opt/soft/test/log30.tar.gz log2013.log
```

我可以透过tar -ztvf来查阅 tar 包内的文件名称，如果单只要一个文件，就可以透过这个方式来解压部分文件！

文件备份下来，并且保存其权限 □

```
tar -zcvpf log31.tar.gz log2014.log log2015.log log2016.log
```

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。

在文件夹当中，比某个日期新的文件才备份 □

```
tar -N "2012/11/13" -zcvf log17.tar.gz test
```

备份文件夹内容是排除部分文件：

```
tar --exclude scf/service -zcvf scf.tar.gz scf/*
```

打包文件之后删除源文件：

```
tar -cvf test.tar test --remove-files
```

其实最简单的使用 tar 就只要记忆底下的方式即可：

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

查询□tar -jtv -f filename.tar.bz2

解压缩□tar -jxv -f filename.tar.bz2 -C 欲解压缩的目录

From:

<https://rd.irust.top/> - 学习笔记

Permanent link:

<https://rd.irust.top/doku.php?id=command:tar>

Last update: **2021/10/15 14:58**

