

cpio

用来建立、还原备份档的工具程序

补充说明

cpio命令主要是用来建立或者还原备份档的工具程序。cpio命令可以复制文件到归档包中，或者从归档包中复制文件。

语法

```
cpio(选项)
```

选项

- 0或--null 接受新增列控制字符，通常配合find指令的“-print0”参数使用；
- a或--rest-access-time 重新设置文件的存取时间；
- A或--append 附加到已存在的备份文档中，且这个备份文档必须存放在磁盘上，而不能放置于磁带机里；
- b或--awap 此参数的效果和同时指定“-ss”参数相同；
- B 将输入/输出的区块大小改成5210Bytes
- c 使用旧ASCII备份格式；
- C<区块大小>或--io-size=<区块大小>：设置输入/输出的区块大小，单位是Byte
- d或--make-directories 如有需要cpio会自行建立目录；
- E<范本文件>或--pattern-file=<范本文件>：指定范本文件，其内含有一个或多个范本样式，让cpio解开符合范本条件的文件，格式为每列一个范本样式；
- f或--nonmatching 让cpio解开所有不符合范本条件的文件；
- F<备份档>或--file=<备份档>：指定备份档的名称，用来取代标准输入或输出，也能借此通过网络使用另一台主机的保存设备存取备份档；
- H<备份格式>：指定备份时欲使用的文件格式；
- i或--extract 执行copy-in模式，还原备份档；
- l<备份档>：指定备份档的名称，用来取代标准输入，也能借此通过网络使用另一台主机的保存设备读取备份档；
- k 此参数将忽略不予处理，仅负责解决cpio不同版本间的兼容性问题；
- l或--link 以硬连接的方式取代复制文件，可在copy-pass模式下运用；
- L或--dereference 不建立符号连接，直接复制该连接所指向的原始文件；
- m或preserve-modification-time 不去更改文件的更改时间；
- M<回传信息>或--message=<回传信息>：设置更换保存媒体的信息；
- n或--numeric-uid-gid 使用“-tv”参数列出备份档的内容时，若再加上参数“-n”则会以用户识别和群组识别码替代拥有者和群组名称列出文件清单；
- o或--create 执行copy-out模式，建立备份档；
- O<备份档>：指定备份档的名称，用来取代标准输出，也能借此通过网络使用另一台主机的保存设备存放备份档；
- p或--pass-through 执行copy-pass模式，略过备份步骤，直接将文件复制到目的目录；
- r或--rename 当有文件名称需要更改时，采用互动模式；
- R<拥有者></.><所属群组>或---owner<拥有者></.><所属群组> 在copy-in模式还原备份档，或copy-pass模式复制文件时，可指定这些备份，复制的文件的拥有者与所属群组；
- s或--swap-bytes 交换每队字节的内容；
- S或--swap-halfwords 交换每半个字节的内容；
- t或--list 将输入的内容呈现出来；
- u或--unconditional 置换所有文件，不论日期时间的新旧与否，皆不予询问而直接覆盖；

```
-v或--verbose 详细显示指令的执行过程；
-V或--dot 执行指令时。在每个文件的执行程序前面加上“.”号；
--block-size=<区块大小>：设置输入/输出的区块大小，假如设置数值为5，则区块大小为2500，若设置成10，则区块大小为5120，以此类推；
--force-local 强制将备份档存放在本地主机；
--help 在线帮助；
--no-absolute-filenames 使用相对路径建立文件名称；
--no-preserve-owner 不保留文件的拥有者，谁解开了备份档，那些文件就归谁所有；
-only-verify-crc 当备份档采用CRC备份格式时，可使用这项参数检查备份档内的每个文件是否正确无误；
--quiet 不显示复制了多少区块；
--sparse 倘若一个文件内含有大量的连续0字节，则将此文件存在稀疏文件；
--version 显示版本信息。
```

实例

将/etc下的所有普通文件都备份到/opt/etc.cpio，使用以下命令：

```
find /etc -type f | cpio -ocvB >/opt/etc.cpio
```

将系统上所有资料备份到磁带机内，使用以下命令：

```
find / -print | cpio -covB > /dev/st0
```

这里的/dev/st0是磁带的设备名，代表SCSI磁带机。

查看上例磁带机上备份的文件，使用以下命令：

```
cpio -icdvt < /dev/st0 > /tmp/st_content
```

有时可能因为备份的文件过多，一个屏幕无法显示完毕，此时我们利用下面命令，让磁带机的文件信息输出到文件。

将示例1中的备份包还原到相应的位置，如果有相同文件进行覆盖，使用以下命令：

```
cpio -icdvt < /opt/etc.cpio
```

注意cpio恢复的路径，如果cpio在打包备份的时候用的是绝对路径，那么在恢复的时候会自动恢复到这些绝对路径下，本例就会将备份文件全部还原到/etc路径下对应的目录中。同理，如果在打包备份用的是相对路径，还原时也将恢复到相对路径下。

通过上面的示例，可以看出cpio无法直接读取文件，它需要每个文件或者目录的完整路径名才能识别读取，而find命令的输出刚好做到了这点，因此cpio命令一般和find命令配合使用。其实，上面的示例我们已经看到了它们的组合用法。

From:
<https://rd.irust.top/> - 学习笔记

Permanent link:
<https://rd.irust.top/doku.php?id=command:cpio>

Last update: 2021/10/15 14:58



