

ar

建立或修改备存文件，或是从备存文件中抽取文件

补充说明

ar命令是一个建立或修改备存文件，或是从备存文件中抽取文件的工具。ar可让您集合许多文件，成为单一的备存文件。在备存文件中，所有成员文件皆保有原来的属性与权限

语法

```
ar [-]{dmpqrtx}[abcfilNoPsSuvV] [membername] [count] archive files...
Usage: ar [emulation options] [-]{dmpqrstx}[abcDfilMNoPsSTuvV] [--plugin
<name>] [member-name] [count] archive-file file...
       ar -M [<mri-脚本>]
```

选项

ar 允许你在第一个命令行参数中以任意顺序混合指定操作码p 和修饰符mod 。

只要你愿意,也可以用破折号作为命令行第一个参数的开始.

p 关键字指明要执行的操作,只能指明为如下之一:

d 从档案中删除模块.通过files 指明要删除的模块的名称;如果没有指出要删除的文件名称,档案不会改变任何内容.

如果给出了'v' 修饰符,ar 会例出它删除的每一个模块.

m 用此操作在档案中移动成员.

如果某个符号名在档案的多个成员中有定义,那么程序怎样连接档案文件得到的结果可能是不同的.

如果没有为m 指定修饰符,由files 指出的成员将移动到档案的末尾;可以通过`a`,`b` 或`i` 等修饰符,指定成员

移动的具体位置.

p 在标准输出上打印档案中指定的成员.如果给出了`v` 修饰符,在打印成员内容之前,先打印成员的名字.

如果没有指明files 参数,档案中所有的成员都会被打印出来.

q 快速追加;增加 files 到 archive 的末尾,不进行替换检查.

修饰符`a`,`b` 和`i`不影响此操作,新成员始终追加到档案的末尾处.

修饰符`v` 可以使 ar 列出它追加的所有文件.

由于本功能是用于快速操作,即使档案中有符号表索引也不进行更新;可以使用`ar s` 或 ranlib 明确要求更新

这些索引.

在为快速追加重建索引时,由于有太多不同的系统,所以 GNU ar 采用`q` 作为`r`的一个同义字.

r 把文件 files 插入 archive (替换).本操作与`q` 是不同的,如果档案中已有的某

个成员与插入文件的名称相同, 此成员将被删除.

如果不存在名称为 files 的文件, ar 显示一个错误消息, 并且保留档案中已有的同名成员. 缺省情况下, 新成员增加到档案的末尾; 可以通过使用 `a` `b` 或 `i` 等修饰符指定相对于已有成员的位置.

通过使用 `v` 修饰符会为每个插入的文件产生一行输出, 根据输出中的字符 `a` 或 `r` 可以表明该文件是追加的 (没有删除以前的成员) 还是替换的.

t 显示一个 archive 档案所包含内容的列表, 或档案中的由 files 指出的文件列表. 通常只显示成员的名称, 如果使用 `v` 修饰符, 可以得到成员的权限, 时间属性, 属主, 组和大小.

如果没有指出 files, 档案中的所有文件都会列出.

如果档案中 (称为 `b.a`) 有多个同名成员 (称为 `fie`), `ar t b.a fie` 仅仅列出第一个; 要看到它们的全部, 必须要求完整的列表—在本例中是 `ar t b.a`.

x 从档案中抽取成员 (名称为 files) . 如果使用 `v` 修饰符, ar 会列出它抽取的每一个文件的名字. 如果没有给出 files, 抽取档案中所有的文件.

可以在操作符 p 后紧随一定数量的修饰符 mod 以指明操作的各种行为.

a 增加文件到档案中已有成员之后, 如果使用了修饰符 a, 必须在档案名称 archive 之前以 membername 参数的形式给出档案中已有成员的名字.

b 增加文件到档案中已有成员之前, 如果使用了修饰符 b, 必须在档案名称 archive 之前以 membername 参数的形式给出档案中已有成员的名字. (和修饰符 `i` 相同).

c 建立档案. 指定的档案 archive 始终会被建立, 如果你要求执行的是更新, 通过此修饰符建立档案时会给出一个警告.

f 截短档案成员的名字. ar 通常允许任意长度的文件名, 但这会导致与某些系统上的 ar 出现兼容性问题, 使用 f 修饰符可以在往档案中追加文件时把名字截短.

i 插入文件到档案中已有成员之前, 如果使用了修饰符 i, 必须在档案名称 archive 之前以 membername 参数的形式给出档案中已有成员的名字. (与修饰符 `b` 相同).

l 接受此修饰符, 但不起作用.

N 使用 count 参数. 本修饰符用于在档案中有多个同名成员的情况. 删除或抽取档案中给定名字的第 count 个实例.

o 抽取成员时保留他们原始的时间属性. 如果没有此修饰符, 文件以抽取的时间作为它的时间属性.

P 匹配档案中的名字时使用完整的路径名。 ar 不能建立使用完整路径名的档案 (这不符合 POSIX 标准), 但其它的档案工具能够建立, 本选项会使 ar 在抽取由其它工具建立的档案文件时, 使用完整的路径名去匹配档案中成员的名字。

s 即使没有对档案进行改变, 用本修饰符也可以往档案中写一个目标文件的索引或更新已经存在的索引。 可以与其它操作一起使用本修饰符, 也可以单独使用。 对一个档案执行 `ar s` 与执行 `ranlib` 等价。

S 不生成档案的符号表。 这可以加速建立大的档案文件的过程, 但这样建立的档案不能被连接器使用, 为建立符号表, 在最后执行 `ar` 时应该不用 `S` 修饰符, 或者对档案执行一次 `ranlib` 。

u 通常 ar r... 把所有列出的文件插入到档案中, 如果希望仅仅插入比档案中已有成员更新的文件时, 就应该使用此修饰符。 `u` 修饰符仅允许与 `r` (替换) 操作一起使用。 某些情况下, 由于用 `q` 操作比较文件的时间属性会失去速度上的优势, 所以不允许执行 `qu` 组合操作。

v 使用本修饰符可以进行冗余的操作。 附加了此修饰符时, 很多操作会显示更多的消息, 如处理的文件名等。

V 显示 ar 的版本号。

选项参数

```
--plugin <p> - load the specified plugin
```

ar 支持的目标 `elf64-x86-64 elf32-i386 elf32-x86-64 a.out-i386-linux pei-i386 pei-x86-64 elf64-l1om elf64-k1om elf64-little elf64-big elf32-little elf32-big plugin srec symbolsrec verilog tekhex binary ihex`

实例

打包文件

```
[root@localhost ~]# ls # 显示当前目录文件
a.c b.c d.c install.log qte
anaconda-ks.cfg c.c Desktop

[root@localhost ~]# ar rv one.bak a.c b.c # 打包 a.c b.c文件
ar: 正在创建 one.bak
a - a.c
a - b.c
```

打包多个文件

```
[root@localhost ~]# ar rv two.bak *.c // 打包以.c结尾的文件
ar: 正在创建 two.bak
a - a.c
a - b.c
a - c.c
a - d.c
```

显示打包文件的内容

```
[root@localhost ~]# ar t two.bak  
a.c  
b.c  
c.c  
d.c
```

删除打包文件的成员文件

```
[root@localhost ~]# ar d two.bak a.c b.c c.c  
[root@localhost ~]# ar t two.bak  
d.c
```

From:

<https://rd.irust.top/> - 学习笔记

Permanent link:

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

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

