<https://blog.csdn.net/u013571243/article/details/49619067>

当我们的程序中有经常使用的模块，而且这种模块在其他程序中也会用到，这时按照软件重用的思想，我们应该将它们生成库，使得以后编程可以减少开发代码量。这里介绍命令ar，用来对库操作。  
1.ar基本用法  
　　ar命令可以用来创建、修改库，也可以从库中提出单个模块。库是一单独的文件，里面包含了按照特定的结构组织起来的其它的一些文件（称做此库文件的member）。原始文件的内容、模式、时间戳、属主、组等属性都保留在库文件中。  
　　下面是ar命令的格式：  
　　ar [-]{dmpqrtx}[abcfilNoPsSuvV] [membername] [count] archive files...  
　　例如我们可以用ar rv libtest.a hello.o hello1.o来  
生成一个库，库名字是test，链接时可以用-ltest链接。该库中存放了两个模块hello.o和hello1.o。选项前可以有‘-'字符，也可以  
没有。下面我们来看看命令的操作选项和任选项。现在我们把{dmpqrtx}部分称为操作选项，而[abcfilNoPsSuvV]部分称为任选项。  
  
补充说明：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。  
  
参　　数：  
指令参数  
-d 　删除库文件中的成员文件。  
-m 　变更成员文件在库文件中的次序。  
-p 　显示库文件中的成员文件内容。  
-q 　将问家附加在库文件末端。  
-r 　将文件插入库文件中。  
-t 　显示库文件中所包含的文件。  
-x 　自库文件中取出成员文件。  
选项参数  
a<成员文件> 　将文件插入库文件中指定的成员文件之后。  
b<成员文件> 　将文件插入库文件中指定的成员文件之前。  
c 　建立库文件。  
f 　为避免过长的文件名不兼容于其他系统的ar指令指令，因此可利用此参数，截掉要放入库文件中过长的成员文件名称。  
i<成员文件> 　将问家插入库文件中指定的成员文件之前。  
o 　保留库文件中文件的日期。  
s 　若库文件中包含了对象模式，可利用此参数建立备存文件的符号表。  
S 　不产生符号表。  
u 　只将日期较新文件插入库文件中。  
v 　程序执行时显示详细的信息。  
V 　显示版本信息。

ar用来管理一种文档。这种文档中可以包含多个其他任意类别的文件。这些被包含的文件叫做这个文档的成员。ar用来向这种文档中添加、删除、解出成员。成员的原有属性（权限、属主、日期等）不会丢失。

实际上通常只有在开发中的目标连接库是这种格式的，所以尽管不是，我们基本可以认为ar是用来操作这种目标链接库（.a文件）的。

ar的常用用法见正文。

1、创建库文件

我 不知道怎么创建一个空的库文件。好在这个功能好像不是很需要。通常人们使用“ar cru liba.a a.o"这样的命令来创建一个库并把a.o添加进去。"c"关键字告诉ar需要创建一个新库文件，如果没有指定这个标志则ar会创建一个文件，同时会给出 一个提示信息，"u"用来告诉ar如果a.o比库中的同名成员要新，则用新的a.o替换原来的。但是我发现这个参数也是可有可无的，可能是不同版本的ar 行为不一样吧。实际上用"ar -r liba.a a.o"在freebsd5上面始终可以成功。

2、加入新成员

使用"ar -r liba.a b.o"即可以将b.o加入到liba.a中。默认的加入方式为append，即加在库的末尾。"r"关键字可以有三个修饰符"a", "b"和"i"。

* "a"表示after，即将新成员加在指定成员之后。例如"ar -ra a.c liba.a b.c"表示将b.c加入liba.a并放在已有成员a.c之后；
* "b"表示before，即将新成员加在指定成员之前。例如"ar -rb a.c liba.a b.c";
* "i"表示insert，跟"b"作用相同。

3、列出库中已有成员

"ar -t liba.a"即可。如果加上"v"修饰符则会一并列出成员的日期等属性。

4、删除库中成员

"ar -d liba.a a.c"表示从库中删除a.c成员。如果库中没有这个成员ar也不会给出提示。如果需要列出被删除的成员或者成员不存在的信息，就加上"v"修饰符。

5、从库中解出成员

"ar -x liba.a b.c"

6、调整库中成员的顺序

使用"m"关键字。与"r"关键字一样，它也有3个修饰符"a","b", "i"。如果要将b.c移动到a.c之前，则使用"ar -mb a.c liba.a b.c"