

静态库

命名格式

- 1 lib开头
  - 2 静态库名
  - 3 .a结尾
- libsort.a

描述

- 优点
  - 寻址方便，速度快
  - 😄 库被打包到可执行程序中，直接发布可执行程序即可使用
- 缺点
  - 🗑️ 静态库的代码在编译过程中已经被载入可执行程序，因此体积较大。
  - 🗑️ 如果静态函数库改变了，那么你的程序必须重新编译。
- 使用场合
  - 在核心程序上使用，保证速度，可忽视空间
  - 主流应用于80、90年代，现在很少用

★ 制作

- 1 得到 \*.o gcc a.c b.c c.c -c
- 2 得到静态库 libmytest.a
  - ar rcs libmytest.a a.o b.o c.o  
(就是一个打包.o文件的过程)
  - ar 工具不包含在gcc中
  - r --> 将文件插入静态库中
  - c --> 创建静态库，不管库是否存在
  - s --> 写入一个目标文件索引到库中，或者更新一个存在的目标文件索引。
  - 查看库中的符号（函数、全局变量等）：nm libmytest.a
- 3 使用静态库
  - gcc + 源文件 + -L 静态库路径 + -l静态库名 + -I头文件目录 + -o 可执行文件名
  - gcc main.c -L./ -lmytest -I./ -o app
  - L --> 指定库所在的路径
  - l --> 指定库的名字
    - 👉 去掉前缀 lib
    - 👉 去掉后缀 .a
    - 👉 只留下中间部分
  - I --> 头文件目录位置
  - gcc + 源文件 + -I头文件 + libxxx.a
  - 生成的静态库需要跟对应的头文件同时发布
  - 头文件中存放的是函数接口（函数声明）