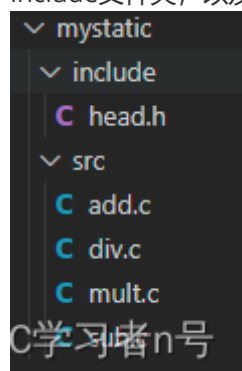


库的介绍

- 库是特殊的一种程序，编写库的程序和编写一般的程序区别不大，只是库不能单独运行。
- 库文件有两种，静态库和动态库（共享库），区别是：静态库在程序的链接阶段被复制到了程序中；动态库在链接阶段没有被复制到程序中，而是程序在运行时由系统动态加载到内存中供程序调用。所以调用了静态库的程序可以将库文件剪切到其他地方也不会影响程序的执行；而exe程序在执行的时候必须要在指定的位置能够找到动态库才能执行。
- 库的好处：1.代码保密 2.方便部署和分发

静态库的制作

比如设计一个算数（+*/）的动态库。首先在mystatic文件夹里面有两个文件夹，分别存放了头文件的include文件夹，以及存放了函数实现的src文件夹。



静态库制作流程

1、命名规则：

Linux : libxxx.a

lib : 前缀（固定）

xxx : 库的名字，自己起

.a : 后缀（固定）

2、制作

- gcc编译生成得 .o 文件

```
gcc -c add.c sub.c mult.c div.c -I../include
```

首先将当前所在目录改变到src中，通过gcc的-c命令生成对应的.o文件，此时因为这些实现函数的文件所依赖的头文件在上一级的include文件夹中，所以要通过-I（大写i）指定头文件所在的目录，否则无法编译通过。

此时在当前的src目录中已经存在了四个.o文件。

- 将 .o 文件打包，使用 ar 工具（archive）

ar rcs libxxx.a xxx.o xxx.o

r - 将文件插入备存文件中

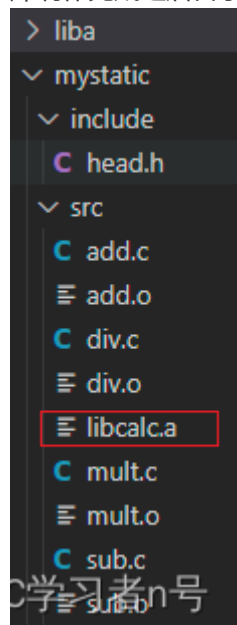
c - 建立备存文件

s - 索引

此时还是在当前的src目录中，执行一下命令，将当前目录中的.o文件进行打包，库名为calc。执行完成之后在当前目录就会有libcalc.a的文件。

```
ar rcs libcalc.a add.o sub.o mult.o div.o
```

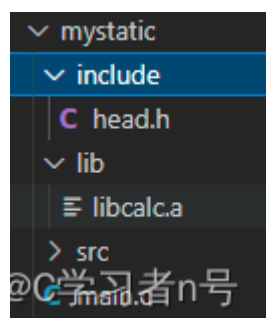
库制作完成之后目录中的文件：



静态库的使用

为了更符合项目的目录习惯，现在mystatic的目录中新建一个lib目录，并且将刚才生成的.a静态库拷贝到这个目录中。

此外在mystatic中直接创建一个main.c文件，用于使用静态库。



此时在mystatic目录中执行一下命令：因为静态库头文件和main都不在同一个目录所以要通过-L和-I指定。

此时在当前mystatic目录中生成了app可执行文件，它在编译过程中就已经将库赋值到了main.c的编译文件中，所以app可执行文件的执行不会依赖库，可以将app文件拷贝到任何地方都能执行。

```
gcc -o app main.c -lcalc -L./lib/ -I./include
```

总结

静态库的制作使用到了-I -l 和-L三个gcc参数。