

## 头文件的作用

一般来说我们C语言程序需要用到的很多的.c 文件，当某一些公共的资源需要在各个源文件中使用的時候，就可以把它写在头文件中，被其它的.c文件包含，可以避免编写同样的代码。

头文件内部放：

- 1) 普通函数声明
- 2) 宏定义
- 3) 结构体、共用体模板定义（声明）
- 4) 枚举常量列表
- 5) static 函数和 inline 函数定义
- 6) 其他头文件

添加头文件的搜索路径

```
$ gcc demo.c -v
```

```
#include "...": search starts here:
#include <...> search starts here:
 /usr/lib/gcc/x86_64-linux-gnu/7/include
 /usr/local/include
 /usr/lib/gcc/x86_64-linux-gnu/7/include-fixed
 /usr/include/x86_64-linux-gnu
 /usr/include
End of search list.
```

默认搜索路径

```
$ gcc demo.c -I./ -v
```

```
#include "...": search starts here:
#include <...> search starts here:
 ./
 /usr/lib/gcc/x86_64-linux-gnu/7/include
 /usr/local/include
 /usr/lib/gcc/x86_64-linux-gnu/7/include-fixed
 /usr/include/x86_64-linux-gnu
 /usr/include
End of search list.
```

## 头文件格式：

```
1  #ifndef      __DEMO_H    // 判断是否有定义某个宏，用来防止头文件被多次包含出现的重定义问题
2  #define      __DEMO_H    //如果没有定义，说明前面没有包含过该头文件
3
4  //其它头文件
5  ....
6  ...
7
8  // 结构体声明
9  ..
10  ....
11
12 // 函数声明
13 ....
14 ....
15
16 // 静态函数定义
17 .....
```

```

18  ...
19
20  // 枚举
21
22
23
24
25
26  #endif // ifndef 的结束

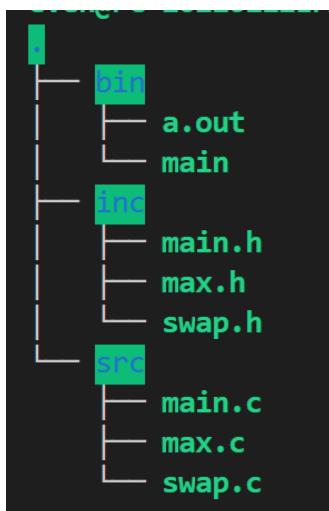
```

由于项目中各个文件比较多，建议使用不同的文件夹来存放不同类型的文件：

bin：二进制的可执行文件

inc：用户自己写的头文件

src：源文件



如何编译：

```

1  $ gcc src/*.c -o bin/Tiezhushu -I./inc
2
3  gcc          编译器
4  src/*.c      需要编译的源文件路径 + 文件名
5  -o           指定输出文件的名称 后边必须接 目标文件的路径+名字
6  bin/Tiezhushu  目标文件的路径 + 名字
7  -I./inc      -I 指定头文件路径 后面必须接头文件所在的路径

```

注意：