

# GCC命令详解

---

## 一、基本格式

---

```
1 | gcc [options] source_filename [-o target_name]
```

其中中括号表示可选项。

options是编译选项，可有可无，下面细说

source\_filename是待编译文件名

-o用来指定输出文件名，后面跟一个目标文件名target\_name

如果不指定target.name，则默认生成文件名为 `a.out`。

## 二、选项详解

---

### 1、无选项

#### 单文件编译

这个最简单，比如 `gcc main.c -o main`。表示完成编译连接全过程，生成可执行文件。

#### 多文件编译

多个文件可共同生成一个可执行文件，比如 `gcc main.c fibo.c -o main`。

#### 由.o文件生成可执行文件

不仅源文件可作为source\_filename，.o文件也可以。比如 `gcc main.o fibo.o -o main`。

### 2、-c选项

表示只进行编译、汇编，不进行连接。生成的文件叫做目标文件，或可重定向文件。

比如 `gcc -c main.c -o main.o`

这里，`source_filename`只能是源文件，不能是目标文件。并且，**一个.o文件必须和.c文件一一对应，不能由多个源文件生成一个目标文件。**

另外，无论用哪种选项，**头文件绝对禁止出现在`source_filename`中！**头文件是用来被源文件包含的，而不是直接参与编译的。

刚才说过，一个或多个.o文件也可共同生成可执行文件。

### 3、**-I 选项（大写字母I，不是小写字母l）**

用来添加一个头文件搜索目录。我们的头文件不仅仅包括标准库和当前目录下，还经常引用其它目录下的。这就需要添加头文件搜索路径。比如，当前目录下的 `main.c` 引用了当前目录下 `include/` 子目录的头文件 `header.h`，那么编译（生成.o文件，而不是可执行文件）命令就是 `gcc -c -I./include main.c -o main.o`。

**-I 选项后面只能是目录，而不能是文件！**

### 4、**-Wall 选项**

这个选项表示生成尽可能多的警告。有时标准的编译过程可能略过某些警告，但这个选项让它们都暴露出来。更多的警告可帮助我们快速查错。不要忽略警告，它可能会暴露出你程序的致命错误。注意，W是大写，不要写错。

### 5、**-O2选项**

这个选项表示优化，使得生成的目标代码在运行时间空间上更少，并且不保留调试信息。一般来说，-O2是在程序发布的时候才使用，开发阶段不要使用。

## 三、多文件编译

### 1、一步到位法

比如现在有源文件 `main.c fibo.c fact.c`，那么可以直接一步编译生成main可执行文件

```
1 | gcc main.c fibo.c fact.c -o main
```

如果头文件在include目录下，就是

```
1 | gcc -I./include main.c fibo.c fact.c -o main
```

## 2、循序渐进法

也可以先分别生成对应的目标文件，再生成相应可执行文件。这样更符合工程规范。

```
1 gcc -c main.c -o main.o
2 gcc -c fibo.c -o fibo.o
3 gcc -c fact.c -o fact.o
4 gcc main.o fibo.o fact.o -o main
```

如果main.c引用了在include目录下的头文件，那么就这样

```
1 gcc -I./include -c main.c -o main.o
2 gcc -c fibo.c -o fibo.o
3 gcc -c fact.c -o fact.o
4 gcc main.o fibo.o fact.o -o main
```