

C 头文件

头文件是扩展名为 `.h` 的文件，包含了 C 函数声明和宏定义，被多个源文件中引用共享。有两种类型的头文件：程序员编写的头文件和编译器自带的头文件。

在程序中要使用头文件，需要使用 C 预处理指令 **#include** 来引用它。前面我们已经看过 **stdio.h** 头文件，它是编译器自带的头文件。

引用头文件相当于复制头文件的内容，但是我们不会直接在源文件中复制头文件的内容，因为这么做很容易出错，特别在程序是由多个源文件组成的时候。

A simple practice in C 或 C++ 程序中，建议把所有的常量、宏、系统全局变量和函数原型写在头文件中，在需要的时候随时引用这些头文件。

引用头文件的语法

使用预处理指令 **#include** 可以引用用户和系统头文件。它的形式有以下两种：

```
#include <file>
```

这种形式用于引用系统头文件。它在系统目录的标准列表中搜索名为 `file` 的文件。在编译源代码时，您可以通过 `-I` 选项把目录前置在该列表前。

```
#include "file"
```

这种形式用于引用用户头文件。它在包含当前文件的目录中搜索名为 `file` 的文件。在编译源代码时，您可以通过 `-I` 选项把目录前置在该列表前。

引用头文件的操作

#include 指令会指示 C 预处理器浏览指定的文件作为输入。预处理器的输出包含了已经生成的输出，被引用文件生成的输出以及 **#include** 指令之后的文本输出。例如，如果您有一个头文件 `header.h`，如下：

```
char *test (void);
```

和一个使用了头文件的主程序 `program.c`，如下：

```
int x;
#include "header.h"

int main (void)
{
```

```
puts (test ());  
}
```

编译器会看到如下的代码信息：

```
int x;  
char *test (void);  
  
int main (void)  
{  
    puts (test ());  
}
```

只引用一次头文件

如果一个头文件被引用两次，编译器会处理两次头文件的内容，这将产生错误。为了防止这种情况，标准的做法是把文件的整个内容放在条件编译语句中，如下：

```
#ifndef HEADER_FILE  
#define HEADER_FILE  
  
the entire header file file  
  
#endif
```

这种结构就是通常所说的包装器 **#ifndef**。当再次引用头文件时，条件为假，因为 `HEADER_FILE` 已定义。此时，预处理器会跳过文件的整个内容，编译器会忽略它。

有条件引用

有时需要从多个不同的头文件中选择一个引用到程序中。例如，需要指定在不同的操作系统上使用的配置参数。您可以通过一系列条件来实现这点，如下：

```
#if SYSTEM_1  
    # include "system_1.h"  
#elif SYSTEM_2  
    # include "system_2.h"  
#elif SYSTEM_3  
    ...  
#endif
```

但是如果头文件比较多的时候，这么做是很不妥当的，预处理器使用宏来定义头文件的名称。这就是所谓的**有条件引用**。它不是用头文件的名称作为 **#include** 的直接参数，您只需要使用宏名称代替即可：

```
#define SYSTEM_H "system_1.h"

...

#include SYSTEM_H
```

SYSTEM_H 会扩展，预处理器会查找 system_1.h，就像 **#include** 最初编写的那样。SYSTEM_H 可通过 -D 选项被您的 Makefile 定义。

[← C 预处理器](#)[C 强制类型转换 →](#)

2 篇笔记

[写笔记](#)

在有多个 .h 文件和多个 .c 文件的时候，往往会用一个 **global.h** 的头文件来包括所有的 .h 文件，然后在除 **global.h** 文件外的头文件中包含 **global.h** 就可以实现所有头文件的包含，同时不会乱。方便在各个文件里面调用其他文件的函数或者变量。

```
#ifndef _GLOBAL_H
#define _GLOBAL_H
#include <fstream>
#include <iostream>
#include <math.h>
#include <Config.h>
```

PPBgogogo 2年前 (2017-09-04)



C 语言中 include <> 与 include "" 的区别？

#include < > 引用的是编译器的类库路径里面的头文件。

#include " " 引用的是你程序目录的相对路径中的头文件，如果在程序目录没有找到引用的头文件则到编译器的类库路径的目录下找该头文件。

熙熙 5天前

