

C 语言预处理概述以及文件包含命令

前面各章中，已经多次使用过 `#include` 命令。使用库函数之前，应该用 `#include` 引入对应的头文件。这种以 `#` 号开头的命令称为预处理命令。

C 语言源文件要经过编译、链接才能生成可执行程序

1) 编译 (Compile) 会将源文件 (.c 文件) 转换为目标文件。对于 VC/VS，目标文件后缀为 .obj；对于 GCC，目标文件后缀为 .o。

编译是针对单个源文件的，一次编译操作只能编译一个源文件，如果程序中有多个源文件，就需要多次编译操作。

2) 链接 (Link) 是针对多个文件的，它会将编译生成的多个目标文件以及系统中的库、组件等合并成一个可执行程序。

不过，在编译之前有时候还需要对源文件进行简单的处理。例如，我们希望自己的程序在 Windows 和 Linux 下都能够运行，那么就要在 Windows 下使用 VS 编译一遍，然后在 Linux 下使用 GCC 编译一遍。但是现在有个问题，程序中要实现的某个功能在 VS 和 GCC 下使用的函数不同 (假设 VS 下使用 `a()`，GCC 下使用 `b()`)，VS 下的函数在 GCC 下不能编译通过，GCC 下的函数在 VS 下也不能编译通过，怎么办呢？

这就需要在编译之前先对源文件进行处理 如果检测到是 VS 那么就保留 `a()` 删除 `b()`；如果检测到是 GCC，那么就保留 `b()` 删除 `a()`。

这些在编译之前对源文件进行的简单处理，就称为预处理 (即预先处理、提前处理)。

预处理主要是处理以 `#` 开头的命令，例如 `#include <stdio.h>` 等。预处理命令要放在所有函数之外，而且一般都放在源文件的前面。

预处理是 C 语言的一个重要功能，由预处理程序完成。当对一个源文件进行编译时，系统将自动调用预处理程序对源程序中的预处理部分作处理，处理完毕自动进入对源程序的

编译。

编译器会将预处理的结果保存到和源文件同名的*.i* 文件中，例如 main.c 的预处理结果在 main.i 中。和.c 一样，*.i* 也是文本文件，可以用编辑器打开直接查看内容。

C 语言提供了多种预处理功能，如宏定义、文件包含、条件编译等，合理地使用它们会使编写的程序便于阅读、修改、移植和调试，也有利于模块化程序设计。

#include 命令

`#include` 是文件包含命令，主要用来引入对应的头文件。`#include` 的处理过程很简单，就是将头文件的内容插入到该命令所在的位置，从而把头文件和当前源文件连接成一个源文件，这与复制粘贴的效果相同。

`#include` 有两种使用方式：

```
#include <stdio.h>
#include "myHeader.h"
```

使用尖括号 `< >` 和双引号 `" "` 的区别在于头文件的搜索路径不同，我们将在《[C 语言头文件的路径](#)》一节中深入探讨，请大家先记住：包含标准库的头文件一般用尖括号，包含自定义的头文件一般用双引号。

说明：

一个 `#include` 命令只能包含一个头文件，多个头文件需要多个 `#include` 命令。文件包含允许嵌套，也就是说在一个被包含的文件中又可以包含另一个文件。