预处理命令详解

作为一枚C/C++程序猿，在我们编写和查看C/C++源代码的过程中会遇到各种编译指令，这些指令称为预处理命令。预处理命令虽然不是C/C+的一部分，但却扩展了C程序的设计环境，下面传智播客C/C+培训专家将向大家介绍如何应用预处理程序和注释简化程序开发过程，并提高程序的可读性。

ANSI标准定义的C语言所有预处理命令均以符号#开头，比如我写程序时常用的：

#define，#undef,＃include，#if，#else，#elif，#endif，#ifdef，#ifndef, #error

1.  #define 和 #undef

宏定义命令的一般形式为：

#define [MacroName][MacroValue]，示例如下：

#define ITHEIMA 传智播客黑马程序员

在源程序中每次遇到ITHEIMA时，均以定义的值传智播客黑马程序员代换它。

l  在使用该宏时，有以下几点注意事项：

n  该语句没有分号。在标识符和串之间可以有任意个空格。

n  定义宏的时候，可以使用之前已经定义好的宏。

n  如果串长于一行，可以在该行末尾用一反斜杠' \'续行。

#define LONG\_STRING " good good study，\

day day up！"

n  在定义宏标识符时，字母一般需要大写。

l  预处理运算符的使用：

n  # -- 该符号是“字符串化”的意思，出现在宏定义中的#是把跟在后面的参数转换成一个字符串

#define ERROR\_LOG(module)   fprintf(stderr, "error: "#module"\n")

ERROR\_LOG("add"); 转换为 fprintf(stderr,"error:"add"\n");

ERROR\_LOG(devied =0); 转换为 fprintf(stderr,"error:devied=0\n");

n  ## -- 是连接符号, 将多个串连接到一起。

char\*szStr = "传播播客\_黑马程序员";

#defineITCAST (exp)  cout <<sz##exp<< endl;

ITCAST(Str)  // cout<< szStr <<endl;   // 拼变量名

l  #undef –取消其后那个前面已定义过有宏名定义

#undef [MacroName],例如:

#undef ITHEIMA

2.   #include

命令＃i nclude使编译程序将另一源文件嵌入带有＃include的源文件，被读入的源文件必须用双引号或尖括号括起来。例如：

＃include"stdio.h"或者＃include

这两行代码均使用C编译程序读入并编译用于处理磁盘文件库的子程序。将文件嵌入＃i nclude命令中的文件内是可行的，这种方式称为嵌套的嵌入文件，嵌套层次依赖于具体实现。

l  如果显式路径名为文件标识符的一部分，则仅在那些子目录中搜索被嵌入文件。

例如: #include “../include/head.h”

l  如果文件名用双引号括起来，则首先检索当前工作目录。如果未发现文件，则在命令行中说明的所有目录中搜索。如果仍未发现文件，则搜索实现时定义的标准目录。

例如: include “head.h”

l  如果文件名被尖括号括起来，则首先在编译命令行中的目录内检索。如果文件没找到，则检索标准目录，不检索当前工作目录。

例如: include <head.h>

3.   条件编译命令： #if、#else，#elif及#endif

**#define MAX 80**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**#if MAX > 99**

**cout<<"MAX is bigger than 99"<**

**#elif MAX > 90**

**cout<<"MAX is bigger than 90"<**

**#else**

**cout<<"MAXis smaller than 90"<**

**#endif**

**return 0;**

**}**

l  #if的一般含义是如果#if后面的常量表达式为true，则编译它与#endif之间的代码，否则跳过这些代码。命令#endif标识一个#if块的结束。

跟在#if后面的表达式在编译时求值，因此它必须仅含常量及已定义过的标识符，不可使用变量。表达式不许含有操作符sizeof（sizeof也是编译时求值）。

l  #else命令的功能有点象C语言中的else；#else建立另一选择（在#if失败的情况下）。注意，#else属于#if块。

l  #elif命令意义与ELSE IF 相同，它形成一个if else-if阶梯状语句，可进行多种编译选择。#elif 后跟一个常量表达式。如果表达式为true，则编译其后的代码块，不对其它#elif表达式进行测试。否则，顺序测试下一块。

l

4.   #error

命令#error强迫编译程序停止编译，主要用于程序调试。该指令使预处理器发出一条错误消息，该消息包含指令中的文本.这条指令的目的就是在程序崩溃之前能够给出一定的信息。

5.  #ifdef 和 #ifndef

条件编译的另一种方法是用#ifdef与#ifndef命令，它们分别表示"如果有定义"及"如果无定义"。# ifdef的一般形式是：

                     # ifdef macroname

statement sequence

#endif

#ifdef与#ifndef可以用于#if、#else，#elif语句中，但必须与一个#endif。

**define MAX 91**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**#ifdef MAX**

**cout<<"hello,MAX!"<**

**#else**

**cout<<"where is MAX?"<**

**#endif**

**#ifndef LEO**

**cout<<"LEO is not defined"<**

**#endif**

**return 0;**

**}**

今天关于预处理命令的知识点就为大家接受到这里, 欢迎大家留言交流.