# 预处理

**预处理是在编译前所做的工作，编译器自动调用预处理程序对源码中以’#’开头的预处理部分进行处理，处理完毕后，进入源码的编译阶段。**

**1.预定义符号:**

**常用的预定义符号:**

**\_\_FILE\_\_ : 当前编译的文件名.**

**\_\_FUNCTION\_\_ : 当前所在函数的函数名.**

**\_\_DATE\_\_ : 当前编译日期.**

**\_\_TIME\_\_ : 当前编译时间.**

**//以上格式占位符都用%s ，如：printf("%s",\_\_DATE\_\_);**

**\_\_LINE\_\_:当前行数. // 格式占位符用%d，如：printf("%d",\_\_LINE\_\_);**

**2.宏定义: 自定义一个宏(要符合标识符的命名规则),用于表示任意数据或者表达式.**

**无参宏定义: #define 宏名 宏替换**

**比如: #define A 35 //A就代表整型数据35.**

**#define TEST "abddgdsfdlfj" //TEST就代表字符串"abddgdsfdlfj"**

**#define TEST abddgdsfdlfj //TEST就代表abddgdsfdlfj**

**不能给宏定义的常量赋值**

**如：A = 66; // 错误，不能给宏定义常量赋值**

**带参宏定义: #define 宏名(参数表) 宏替换**

**比如: #define M(a,b) a+b-2**

**K = M(1,2) + 4;**

**---> K = 1+2-2+4**

**宏定义尽量用大写，使其在程序中容易辨别区分**

**常量的定义:**

**除了宏定义以外，还可以通过const关键字定义常量:**

**普通常量: <cosnt> <数据类型> <常量名> = <常量值>;**

**如：const int a = 30; //定义一个常量a，其值等于30**

**定义成常量后，值不可被改变**

**如：a=40; //错误，不能给常量赋值**

**3.文件包含:**

**#include <name.h>**

**//包含系统头文件用<>，只会在系统头文件中找**

**#include "name.h"**

**//包含自定义头文件用””，在自定义头文件中找不到就会在系统头文件中找**

**文件包含允许嵌套,即在一个被包含文件中可以包含其它文件.**

**#pragma once //不让文件被包含两次，在头文件最上面添加**

**vs独有的，有使用平台的限制，其他平台可能不存在**

1. **条件编译: 根据不同的条件编译不同代码段.**

**----------------------------------------------------------------------------------------------**

**#if 表达式**

**//判断表达式的逻辑值(真或假),若逻辑值为真,则编译代码段1,否则编译代码段2.**

**代码段1;**

**#else**

**代码段2;**

**#endif**

**----------------------------------------------------------------------------------------------**

**#ifdef 宏名**

**//如果定义了宏"宏名",则编译代码段.**

**代码段;**

**#endif**

**----------------------------------------------------------------------------------------------**

**#ifndef 宏名**

**//如果没有定义宏"宏名",则编译代码段**

**代码段;**

**#endif**

**----------------------------------------------------------------------------------------------**

**#ifdef 宏名**

**//如果定义了宏"宏名",则编译代码段1,否则编译代码段2;**

**代码段1;**

**#else**

**代码段2;**

**#endif**

**----------------------------------------------------------------------------------------------**

**#ifndef 宏名**

**//如果没有定义宏"宏名",则编译代码段1,否则编译代码段2;**

**代码段1;**

**#else**

**代码段2;**

**#endif**

**----------------------------------------------------------------------------------------------------**