

C语言中条件编译#ifdef的妙用详解

前言

文档简介

本文主要介绍c语言中条件编译相关的预编译指令，包括 #define、#undef、#ifdef、#ifndef、#if、#elif、#else、#endif、defined。

关键词解释

关键词	含义
#define	定义一个预处理宏
#undef	取消宏的定义
#if	编译预处理中的条件命令，相当于C语法中的if语句
#ifdef	判断某个宏是否被定义，若已定义，执行随后的语句
#ifndef	与#ifdef相反，判断某个宏是否未被定义
#elif	若#if, #ifdef, #ifndef或前面的#elif条件不满足，则执行#elif之后的语句，相当于C语法中的else-if
#else	与#if, #ifdef, #ifndef对应, 若这些条件不满足，则执行#else之后的语句，相当于C语法中的else
#endif	#if, #ifdef, #ifndef这些条件命令的结束标志.
defined	与#if, #elif配合使用，判断某个宏是否被定义

条件编译

条件编译是根据实际定义宏（某类条件）进行代码静态编译的手段。可根据表达式的值或某个特定宏是否被定义来确定编译条件。

防止重复包含头文件的宏

最常见的条件编译是防止重复包含头文件的宏，形式跟下面代码类似：

```
#ifndef ABCD_H
#define ABCD_H

// ... some declaration codes

#endif // #ifndef ABCD_H
```

debug宏控

在实现文件中通常有如下类似的定义：

```
#ifdef _DEBUG

// ... do some operations

#endif

#ifdef _WIN32

// ... use Win32 API

#endif
```

预编译指令应用举例

#define、#undef

#define命令定义一个宏：

#define MACRO_NAME[(args)] [tokens[(opt)]]

之后出现的MACRO_NAME将被替代为所定义的标记(tokens)。宏可带参数，而后面的

标记也是可选的。

宏定义，按照是否带参数通常分为对象宏、函数宏两种。

对象宏

对象宏: 不带参数的宏被称为"对象宏(objectlike macro)"。对象宏多用于定义常量、通用标识。例如：

```
// 常量定义
#define MAX_LENGTH 100
// 通用标识，日志输出宏
#define SLog printf
// 预编译宏
#define _DEBUG
```

函数宏

函数宏：带参数的宏。利用宏可以提高代码的运行效率: 子程序的调用需要压栈出栈, 这一过程如果过于频繁会耗费掉大量的CPU运算资源。 所以一些代码量小但运行频繁的代码如果采用带参数宏来实现会提高代码的运行效率。但多数c++程序不推荐使用函数宏，调试上有一定难度，可考虑使用c++的inline代替之。例如：

```
// 最小值函数
#define MIN(a,b) ((a)>(b)? (a):(b))
// 安全释放内存函数
#define SAFE_DELETE(p) {if(NULL!=p){delete p; p = NULL;}}
#undef可以取消宏定义，与#define对应。
```

defined

defined用来测试某个宏是否被定义。defined(name): 若宏被定义，则返回1，否则返回0。它与#if、#elif、#else结合使用来判断宏是否被定义，乍一看好像它显得多余, 因为有了#ifdef和#ifndef。defined可用于在一条判断语句中声明多个判别条件；#ifdef和#ifndef则仅支持判断一个宏是否定义。

#if defined(VAX) && defined(UNIX) && !defined(DEBUG) 和#if、#elif、#else不同，#ifdef、#ifndef、defined测试的宏可以是对象宏，也可以是函数宏。

#ifdef、#ifndef、#else、#endif

条件编译中相对常用的预编译指令。模式如下：

```
#ifdef ABC
// ... codes while defined ABC
#elif (CODE_VERSION > 2)
// ... codes while CODE_VERSION > 2
#else
// ... remained cases
#endif // #ifdef ABC
```

#ifdef用于判断某个宏是否定义，和#ifndef功能正好相反，二者仅支持判断单个宏是否已经定义，上面例子中二者可以互换。如果不需要多条件预编译的话，上面例子中的#elif和#else均可以不写。

#if、#elif、#else、#endif

#if可支持同时判断多个宏的存在，与常量表达式配合使用。常用格式如下：

```
#if 常量表达式1
// ... some codes
#elif 常量表达式2
// ... other codes
#elif 常量表达式3
// ...
...
#else
// ... statement
#endif
```

常量表达式可以是包含宏、算术运算、逻辑运算等等的合法C常量表达式，如果常量表达式为一个未定义的宏，那么它的值被视为0。

```
#if MACRO_NON_DEFINED // 等价于

#if 0
```

在判断某个宏是否被定义时，应当避免使用`#if`，因为该宏的值可能就是被定义为0。而应当使用`#ifdef`或`#ifndef`。注意: `#if`、`#elif`之后的宏只能是对象宏。如果宏未定义，或者该宏是函数宏，则编译器可能会有对应宏未定义的警告。