C 语言条件编译详解

预处理程序提供了条件编译的功能，可以按不同的条件去编译不同的程序部分，因而产生不同的目标代码文件。这对于程序的移植和调试是很有用的。条件编译有三种形式，下面分别介绍。

第一种形式

第一种形式的格式为：

#ifdef 标识符

程序段 1

#else

程序段 2

#endif

它的功能是，如果标识符已被 #define 命令定义过则对程序段 1 进行编译；否则对程序段 2 进行编译。如果没有程序段 2（它为空），本格式中的#else 可以没有，即可以写为：

#ifdef 标识符

程序段

#endif

请看下面的例子：

```
1. #include <stdio.h>
2. #define WIN16 true
3. int main(void){
```

```
4.      #ifdef WIN16
5.          printf("The value of sizeof(int) is 2.\n");
6.      #else
7.          printf("The value of sizeof(int) is 4.\n");
8.      #endif
9.      return 0;
10. }
```

运行结果：

The value of sizeof(int) is 2.

第 4 行插入了条件编译预处理命令，要根据 WIN16 是否被定义过来决定编译哪一个 printf 语句。而在程序的第 2 行已对 WIN16 作过宏定义，所以应对第一个 printf 语句进行编译。

程序第 2 行宏定义中，定义 WIN16 表示字符串 true，其实也可以为任何字符串，甚至不给出任何字符串，写为：

```
#define WIN16
```

也具有同样的意义。只有取消程序的第 2 行才会去编译第二个 printf 语句。

第二种形式

第二种形式的格式为：

#ifndef 标识符

    程序段 1

#else

    程序段 2

#endif

与第一种形式的区别是将 ifdef 改为 ifndef。它的功能是，如果标识符未被#define 命令定义过则对程序段 1 进行编译，否则对程序段 2 进行编译。这与第一种形式的功能正相反。

第三种形式

第三种形式的格式为：

#if 常量表达式

程序段 1

#else

程序段 2

#endif

它的功能是，如常量表达式的值为真（非 0），则对程序段 1 进行编译，否则对程序段 2 进行编译。因此可以使程序在不同条件下，完成不同的功能。

请看下面的例子：

```
#include <stdio.h>
#define R 1
int main(){
    float len, area_round, area_square;
    printf ("input a number: ");
    scanf("%f", &len);
    #if R
        area_round = 3.14159*len*len;
        printf("Area of round is: %f\n", area_round);
    #else
        area_square = len*len;
        printf("Area of square is: %f\n", area_square);
    #endif
    return 0;
```

```
15. }
```

运行结果：

```
input a number: 4
Area of round is: 50.265442
```

第 2 行宏定义中，定义 R 为 1，因此在条件编译时，常量表达式的值为真，所以计算并输出圆面积。

上面介绍的条件编译当然也可以用条件语句 if-else 来实现。 但是用条件语句将会对整个源程序进行编译，生成的目标代码程序较长，而采用条件编译，则根据条件只编译其中的程序段 1 或程序段 2，生成的目标程序较短。如果条件选择的程序段很长，采用条件编译的方法是十分必要的。