简单来说：宏定义又称为宏代换、宏替换，简称“宏”。是C提供的三种预处理功能的其中一种。

复杂的请看下面，讲的很全。下面的带参宏定义，多行宏定义，在Linux内核源码中很多。另外sizeof也是一个宏定义。

宏定义

宏定义是C提供的三种预处理功能的其中一种，这三种预处理包括：宏定义、文件包含、条件编译

1. 不带参数的宏定义：

宏定义又称为宏代换、宏替换，简称“宏”。

格式： #define 标识符 字符串

其中的标识符就是所谓的符号常量，也称为“宏名”。

预处理（预编译）工作也叫做宏展开：将宏名替换为字符串。

掌握"宏"概念的关键是“换”。一切以换为前提、做任何事情之前先要换，准确理解之前就要“换”。

即在对相关命令或语句的含义和功能作具体分析之前就要换：

例： 　　#define PI 3.1415926 　　把程序中出现的PI全部换成3.1415926

说明：

（1）宏名一般用大写

（2）使用宏可提高程序的通用性和易读性，减少不一致性，减少输入错误和便于修改。例如：数组大小常用宏定义

（3）预处理是在编译之前的处理，而编译工作的任务之一就是语法检查，预处理不做语法检查。

（4）宏定义末尾不加分号；

（5）宏定义写在函数的花括号外边，作用域为其后的程序，通常在文件的最开头。

（6）可以用#undef命令终止宏定义的作用域

（7）宏定义可以嵌套

（8）字符串" "中永远不包含宏

（9）宏定义不分配内存，变量定义分配内存。

2. 带参数的宏定义：

除了一般的字符串替换，还要做参数代换

格式： 　　#define 宏名（参数表） 字符串

例如：#define S（a,b） a\*b

area=S（3,2）；第一步被换为area=a\*b; ，第二步被换为area=3\*2;

类似于函数调用，有一个哑实结合的过程：

（1）实参如果是表达式容易出问题

#define S（r） r\*r

area=S（a+b）；第一步换为area=r\*r;,第二步被换为area=a+b\*a+b;

正确的宏定义是#define S（r） （（r）\*（r））

（2）宏名和参数的括号间不能有空格

（3）宏替换只作替换，不做计算，不做表达式求解

（4）函数调用在编译后程序运行时进行，并且分配内存。宏替换在编译前进行，不分配内存

（5）宏的哑实结合不存在类型，也没有类型转换。

（6）函数只有一个返回值，利用宏则可以设法得到多个值

（7）宏展开使源程序变长，函数调用不会

（8）宏展开不占运行时间，只占编译时间，函数调用占运行时间（分配内存、保留现场、值传递、返回值）

3. 宏定义其他冷门、重点知识

#define用法

1、 用无参宏定义一个简单的常量

#define LEN 12

这个是最常见的用法，但也会出错。

比如下面几个知识点你会吗？可以看下：

（1） #define NAME "zhangyuncong"

程序中有"NAME"则，它会不会被替换呢？

（2） #define 0x abcd

可以吗？也就是说，可不可以用把标识符的字母替换成别的东西？

（3） #define NAME "zhang

这个可以吗？

（4） #define NAME "zhangyuncong"

程序中有上面的宏定义，并且，程序里有句：

NAMELIST这样，会不会被替换成"zhangyuncong"LIST

四个题答案都是否定的。

第一个，""内的东西不会被宏替换。这一点应该大都知道。

第二个，宏定义前面的那个必须是合法的用户标识符

第三个，宏定义也不是说后面东西随便写，不能把字符串的两个""拆开。

第四个：只替换标识符，不替换别的东西。NAMELIST整体是个标识符，而没有NAME标识符，所以不替换。

也就是说，这种情况下记住：#define 第一位置第二位置

（1） 不替换程序中字符串里的东西。

（2） 第一位置只能是合法的标识符（可以是关键字）

（3） 第二位置如果有字符串，必须把""配对。

（4） 只替换与第一位置完全相同的标识符

还有就是老生常谈的话：记住这是简单的替换而已，不要在中间计算结果，一定要替换出表达式之后再算。

2、 带参宏一般用法

比如#define MAX（a,b） （（a）>（b）？（a）：（b））

则遇到MAX（1+2,value）则会把它替换成：

（（1+2）>（value）？（1+2）：（value））

注意事项和无参宏差不多。

但还是应注意

#define FUN（a） "a"

则，输入FUN（345）会被替换成什么？

其实，如果这么写，无论宏的实参是什么，都不会影响其被替换成"a"的命运。

也就是说，""内的字符不被当成形参，即使它和一模一样。

那么，你会问了，我要是想让这里输入FUN（345）它就替换成"345"该怎么实现呢？

请看下面关于#的用法

3、 有参宏定义中#的用法

#define STR（str） #str

#用于把宏定义中的参数两端加上字符串的""

比如，这里STR（my#name）会被替换成"my#name"

一般由任意字符都可以做形参，但以下情况会出错：

STR（））这样，编译器不会把“）”当成STR（）的参数。

STR（，）同上，编译器不会把“，”当成STR的参数。

STR（A,B）如果实参过多，则编译器会把多余的参数舍去。（VC++2008为例）

STR（（A,B））会被解读为实参为：（A,B），而不是被解读为两个实参，第一个是（A第二个是B）。 　　 4、 有参宏定义中##的用法

#define WIDE（str） L##str

则会将形参str的前面加上L

比如：WIDE（"abc"）就会被替换成L"abc"

如果有#define FUN（a,b） vo##a##b（）

那么FUN（id ma,in）会被替换成void main（）

5、 多行宏定义：

#define doit（m,n） for（int i=0;i<（n）；++i）\

{\

m+=i;\

}