



预定义符号

这些预定义符号都是语言内置的

```
__FILE__      //进行编译的源文件
__LINE__      //文件当前的行号
__DATE__      //文件被编译的日期
__TIME__      //文件被编译的时间
__STDC__      //如果编译器遵循ANSI C，其值为1，否则未定义
```

```
#include <stdio.h>

int main() {
    printf("%s\n", __FILE__);    //进行编译的源文件
    printf("%d\n", __LINE__);    //文件当前的行号
    printf("%s\n", __DATE__);    //文件被编译的日期
    printf("%s\n", __TIME__);    //文件被编译的时间
    //printf("%s\n", __STDC__);  //如果编译器遵循ANSI C，其值为1，否则未定义")
}
```

Microsoft Visual Studio 调试控制台

E:\Github_Gitee\C\Define\test.c
7
Oct 11 2022
20:33:30

E:\Github_Gitee\C\Define\x64\Debug\Define.exe (进程 17088) 已退出，代码为 0。
按任意键关闭此窗口。 . . _

CSDN @期邈云汉

VS环境下未定义STDC,说明Visual Studio并未完全遵循ANSI C。



#define



#define 定义标识符

```
#define name stuff //名称：内容
```

```
#define MAX 1000
#define reg register    //为 register这个关键字，创建一个简短的名字
#define do_forever for(;;)    //用更形象的符号来替换一种实现
#define CASE break;case    //在写case语句的时候自动把 break写上。
// 如果定义的 stuff过长，可以分成几行写，除了最后一行外，每行的后面都加一个反斜杠(续行符)。
#define DEBUG_PRINT printf("file:%s\tline:%d\t \
                                date:%s\ttime:%s\n" ,\
__FILE__,__LINE__ ,\
__DATE__,__TIME__ )

int main() {
    printf("%d\n", MAX);
}
```

```
printf("%s\n", STR);
do_forever; //执行到这里会死循环
return 0;
}
```

在define进行定义时，最好不要在后面加上分号 `;`，替换时也会将分号替换，容易导致问题。

```
#define 1000;
int max = 0;
if (3 > 1)
    max = MAX;
else
    max = 0;
//报错
```

★ #define 定义宏

#define 机制包括了一个规定，允许把参数替换到文本中，这种实现通常称为宏（macro）或定义宏（define macro）

```
#define name( parament-list ) stuff
```

其中的 parament-list 是一个由逗号隔开的符号表，它们可能出现在stuff中。

```
//int Max(int x, int y) {
//    return (x > y ? x : y);
//}
//若采用宏定义
#define Max(x,y) (x>y?x:y)
int main() {
    int a = 10;
    int b = 5;
    int max = Max(a, b);
    printf("%d\n", max);
    return 0;
}
```

最终得到较大值10。找到了宏的标识，直接进行符号替换。

- 参数列表的左括号必须与name紧邻。
- 如果两者之间有任何空白存在，参数列表就会被解释为stuff的一部分

例如：

```
#define SQUARE(x) x*x

int main() {
    printf("%d\n", SQUARE(5));
    printf("%d\n", SQUARE(5 + 1)); //宏的参数是完全替换的，相当于 5 + 1 * 5 + 1 结果为11
    return 0;
}
```

做改进：

```
#define SQUARE(x) (x)*(x) //这样就不容易出错
//最好最外层也加上括号
```

考虑如下计算：

```
#define Double(x) (x)+(x)
int main() {
    int a = 5;
    printf("%d\n", (10 * Double(a))); //这里相当于 10 * (5) + (5) 结果为55
    return 0;
}
```

在宏替换内容上加上括号以保证得到预期的结果。

```
#define Double(x) ((x)+(x))
```

```
//#define Double(x) (x)+(x)
#define Double(x) ((x)+(x))
int main() {
    int a = 5;
    printf("%d\n", (10 * Double(a))); //这里相当于 10 * (5) + (5) 结果为55
    return 0;
}
```

Microsoft Visual Studio 调试控制台

100

E:\Github_Gitee\C\Define\x64\Debug\Define.exe (进程 10924) 已退出，代码为 0。
按任意键关闭此窗口。 . . . _

CSDN @期邈云汉

所以用于对数值表达式进行求值的宏定义都应该用这种方式加上括号，避免在使用宏时由于参数中的操作符或邻近操作符之间不可预料的相互作用。

✳ 替换规则

在程序中扩展#define定义符号和宏时，需要涉及几个步骤。

1. 在调用宏时，首先对参数进行检查，看看是否包含任何由#define定义的符号。如果是，它们首先被替换。
2. 替换文本随后被插入到程序中原来文本的位置。对于宏，参数名被他们的值所替换。
3. 最后，再次对结果文件进行扫描，看看它是否包含任何由#define定义的符号。如果是，就重复上述处理过程。

注意：

1. 宏参数和#define 定义中可以出现其他#define定义的符号。但是对于宏，不能出现递归。
2. 当预处理器搜索#define定义的符号的时候，字符串常量的内容并不被搜索。

✳ # 和

如何把参数插入到字符串中?

```
#define PRINT(FORMAT, VALUE) printf("the value of " #VALUE " is "FORMAT "\n",  
VALUE)  
  
//这里#可以把其后面的参数替换为字符串嵌入  
  
int main() {  
    printf("hello world\n");  
    printf("hello ""world\n"); //天然合为一个字符串(发现字符串是有自动连接的特点的)  
    int a = 100;  
    printf("The value a is %d\n", a);  
    int b = 20;  
    printf("The value b is %d\n", b);  
  
    //那么考虑能否将同一个功能的打印操作合并  
    PRINT("%d", a); //这里只有当字符串作为宏参数的时候才可以把字符串放在字符串中  
    /*代码中的 #VALUE 会预处理器处理为:  
       "VALUE" .*/  
  
    return 0;  
}
```

把宏对应的参数替换为字符串

```
#define PRINT(FORMAT, VALUE) printf("the value of " #VALUE " is "FORMAT "\n", VALUE)  
//这里#可以把其后面的参数替换为字符串嵌入  
  
int main() {  
    printf("hello world\n");  
    printf("hello ""world\n"); //天然合为一个字符串(发现字符串是有自动连接的特点的)  
    int a = 100;  
    printf("The value a is %d\n", a);  
    int b = 20;  
    printf("The value b is %d\n", b);  
  
    //那么考虑能否将同一个功能的打印操作合并  
    PRINT("%d", a); //这里只有当字符串作为宏参数的时候才可以把字符串放在字符串中  
    /*代码中的 #VALUE 会预处理器处理为:  
       "VALUE" .*/  
  
    return 0;  
}
```

Microsoft Visual Studio 调试控制台

```
hello world  
hello world  
The value a is 100  
The value b is 20  
the value of a is 100
```

未 E:\Github_Gitee\C\Define\x64\Debug\Define.exe (进程 8364) 已退出, 代码为 0。
按任意键关闭此窗口. . .

CSDN @期邈云汉

##的作用

将分离的片段合成为一个符号

```
#define CAT(A,B) A##B
//把A和B组合成一个符号
int main() {
    int windows11 = 2022;

    printf("%d\n", CAT(windows, 11));

}
```

```
#define CAT(A,B) A##B
//把A和B组合成一个符号
int main() {
    int Windows11 = 2022;

    printf("%d\n", CAT(Windows, 11));

}
```

Microsoft Visual Studio 调试控制台

2022

E:\Github_Gitee\C\Define\x64\Debug\Define.exe (进程 16632) 已退出，代码为 0。
按任意键关闭此窗口。 . . _

CSDN @期邈云汉

★带副作用的宏参数

当宏参数在宏的定义中出现超过一次的时候，如果参数带有副作用，那么你在使用这个宏的时候就可能出现危险，导致不可预测的后果。副作用就是表达式求值的时候出现的永久性效果。

```
x+1; //不带副作用
x++; //带有副作用
```

考虑宏：

```
//考虑宏的副作用
#define MAX(a, b) ( (a) > (b) ? (a) : (b) )
int main() {
    int x = 5;
    int y = 8;
    int z = MAX(x++, y++);
    printf("x=%d y=%d z=%d\n", x, y, z);
    //z = ((x++) > (y++) ? (x++) : (y++)); x = 6 y = 10 z = 9
}
```

```
//考虑宏的副作用
#define MAX(a, b) ( (a) > (b) ? (a) : (b) )
int main() {
    int x = 5;
    int y = 8;
    int z = MAX(x++, y++);
    printf("x=%d y=%d z=%d\n", x, y, z);
    //z = ((x++) > (y++) ? (x++) : (y++)); x = 6 y = 10 z = 9
}
```

Microsoft Visual Studio 调试控制台

x=6 y=10 z=9

E:\Github_Gitee\C\Define\x64\Debug\Define.exe (进程 15172)已退出，代码为 0。
按任意键关闭此窗口。 . . .

CSDN @期邈云汉

后序更新....