

## 16.NULL,0,nullptr的区别分析

在编写C程序的时候只看到过NULL，而在C++的编程中，我们可以看到NULL和nullptr两种关键字，其实nullptr是C++11版本中新加入的，它的出现是为了解决NULL表示空指针在C++中具有二义性的问题，为了弄明白这个问题，我查找了一些资料，总结如下。

### 一、C程序中的NULL

在C语言中，NULL通常被定义为：`#define NULL ((void *)0)`

所以说NULL实际上是一个空指针，如果在C语言中写入以下代码，编译是没有问题的，因为在C语言中把空指针赋给int和char指针的时候，发生了隐式类型转换，把void指针转换成了相应类型的指针。

```
int *pi = NULL;
char *pc = NULL;
```

### 二、C++程序中的NULL

但是问题来了，以上代码如果使用C++编译器来编译则是会出错的，因为C++是强类型语言，void\*是不能隐式转换成其他类型的指针的，所以实际上编译器提供的头文件做了相应的处理：

```
#ifdef __cplusplus
#define NULL 0
#else
#define NULL ((void *)0)
#endif
```

可见，在C++中，NULL实际上是0，而不是(void\*)0，因为C++中不能把void\*类型的指针隐式转换成其他类型的指针，所以为了结果空指针的表示问题，C++引入了0来表示空指针，这样就有了上述代码中的NULL宏定义。但是实际上，用NULL代替0表示空指针在函数重载时会出现问题，程序执行的结果会与我们的想法不同，举例如下：


```
#include <iostream>
using namespace std;

void func(void* i)
{
    cout << "func1" << endl;
}

void func(int i)
{
    cout << "func2" << endl;
}

void main(int argc, char* argv[])
{
```

```
func(NULL);  
func(nullptr);  
getchar();  
}
```



```
func2  
func1
```

@稀土掘金技术社区

在这段代码中，我们对函数func进行可重载，参数分别是void\*类型和int类型，但是运行结果却与我们使用NULL的初衷是相违背的，因为我们本来是想用NULL来代替空指针，但是在将NULL输入到函数中时，它却选择了int形参这个函数版本(因为NULL定义为0)，所以是有问题的，这就是用NULL代替空指针在C++程序中的二义性。

### 三、C++中的nullptr

为解决NULL代指空指针存在的二义性问题，在C++11版本(2011年发布)中特意引入了nullptr这一新的关键字来代指空指针，从上面的例子中我们可以看到，使用nullptr作为实参，确实选择了正确的以void\*作为形参的函数版本。

总结：

1. NULL在C++中就是0，这是因为在C++中void\* 类型是不允许隐式转换成其他类型的，所以之前C++中用0来代表空指针，但是在重载整形的情况下，会出现上述的问题。
2. 所以，C++11加入了nullptr，可以保证在任何情况下都代表空指针，而不会出现上述的情况，因此，建议以后还是都用nullptr替代NULL吧，而NULL就当做0使用。参考资料：
3. [史上最明白的 NULL、0、nullptr 区别分析（老师讲N篇都没讲明白的东东），今天终于明白了，如果和我一样以前不明白的可以好好的看看... - porter\_代码工作者 - 博客园 (cnblogs.com)]  
[<https://www.cnblogs.com/porter/p/3611718.html>]
4. [C++中NULL和nullptr的区别\_天涯明月刀的博客-CSDN博客\_nullptr和null区别]  
[[https://blog.csdn.net/qz\\_18108083/article/details/84346655](https://blog.csdn.net/qz_18108083/article/details/84346655)]