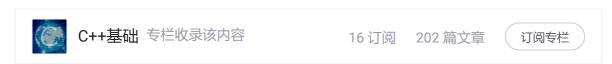
C++中定位new的使用方法



C++中的new关键字可以在内存中动态的分配空间。自定义MyClass类

```
class MyClass
1
 2
 3
    {
 4
    public:
 5
 6
    MyClass()
 7
 8
9
    {
10
    cout << "Construction MyClass." << this << endl;</pre>
11
12
13
    }
14
    ~MyClass()
15
16
17
    {
18
19
    imember = 0;
20
    cout << "Destruction MyClass." << this << endl;</pre>
21
22
23
    }
24
25
    int imember;
26
```

之后,通过new关键字动态分配内存

```
MyClass* fPtr1 = new MyClass();
```

此时,变量fPtr1中保存了动态分配内存的起始地址。但是该起始地址是编译器在内存中随机分配的。

通过定位new可以实现在指定位置上的内存分配。

1 格式

27

};

定位new的格式为

new(args) T

其中, args是一个对象的指针, 即指定的起始地址; T是分配内存的类型; 返回值也是指定的起始地址。

2 使用

```
1 char x[sizeof(MyClass)];
2
3 MyClass* fPtr2 = new(&x[0])MyClass;
```

以上代码的作用是在数组x的起始位置处分配内存。此时,x和fPtr2的值如图1所示。

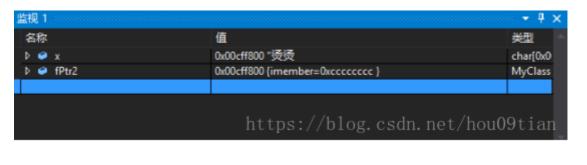


图1 x和fPtr2的值

从图1可以看出,通过定位new分配的内存就是在数组x的起始地址上。

3 释放内存

由于定位new分配的内存并不是"新分配"的,而是原先就有的。所以不能使用delete来释放该内存,而是要通过析构函数来释放资源。

```
fPtr2->~MyClass();
```

公安备案号11010502030143 京ICP备19004658号 京网文 [2020] 1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 ©1999-2022北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照



举拍