链滴

# 一个有趣的 Bug

作者：localvar

原文链接：https://ld246.com/article/1522221733181

来源网站：链滴

许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

今早一到公司，就看到隔壁组一堆人在讨论一个Bug，足足半个多小时没有结果。人散了以后，一位事找我帮忙看一下，发现是因为一个函数被递归调用导致栈溢出并崩溃，但奇怪的是代码中并没有递调用这个函数。看到这个现象，我立刻想到了以前遇到的这个Bug，经过排查，果然是类似的原因，是十分钟不到就解决了问题，引来一片佩服的目光。把这个Bug的代码整理简化后如下：

```
struct Base1
{
    virtual void Foo1()
    {
        printf("Base1::Foo1\n");
    }
};

struct Base2
{
    virtual void Foo2()
    {
        printf("Base2::Foo2\n");
    }
};

void Bar(void* p)
{
    Base2* p2 = (Base2*)(p);
    p2->Foo2(); // 这一句实际调用的是 Derived::Foo1，导致递归
}

struct Derived : public Base1, public Base2
{
    virtual void Foo1()  // 这个函数会被递归调用
    {
        printf("Derived::Foo1\n");
        Bar(this);
    }
};

int main()
{
    Derived d;
    d.Foo1();
    return 0;
}
```

这里有意思的是，原始代码中的Derived::Foo1相当复杂，指针乱了以后没有直接Access Violation而等到多次递归后栈溢出才崩掉，也算是小概率事件了。如果直接崩了，我应该不会马上想到正确的排方向，也就没法秀一把自己的技术能力了。