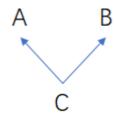
# 多重继承

### V字型继承



- 构造
  - 1. 构造父类,填父类虚表
  - 2. 子类多次填虚表, 覆盖父类虚表

```
C::C()
00F61060 55
00F61061 8B EC
 00F61063 51
 00F61064 56
00F61065 57
00F61066 8B F9
00F61068 89 7D FC
 00F6106B E8 90 FF FF FF
00F61070 8D 4F 08
                                                             lea
                                                                                      ecx,[edi+8]
00F61073 E8 B8 FF FF FF
       c = 30;
printf("C::C()\n");
00F61078 68 48 21 F6 00 push

00F6107D C7 07 3C 21 F6 00 mov

00F61083 C7 47 08 44 21 F6 00 mov

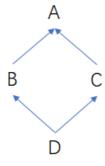
00F6108A C7 47 10 1E 00 00 00 mov

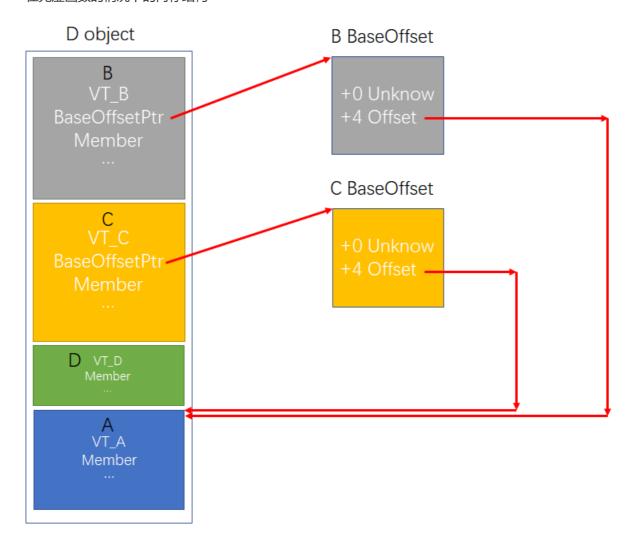
00F61091 E8 9A 01 00 00 call

00F61096 83 C4 04
                                                                                     offset string "C::C()\n" (0F62148h)
dword ptr [edi],offset C::`vftable' (0F6213Ch)
dword ptr [edi+8],offset C::`vftable' (0F62144h)
dword ptr [edi+10h],1Eh
printf (0F61230h)
再填虚表 (要盖父类虚表)
```

- 析构
  - 1. 子类填写虚表, 有几个父类就填几次
  - 2. 析构父类,存在双分支结构,用于判断父类的指针是否为空,然后各个父类开始填虚表

## 菱形继承





#### • 构造

。 构造函数会多一个参数, 1表示是否构造父类, 0表示不构造父类

```
TestD d;

007F1336 8D 4D E4 lea ecx,[d]

007F1339 6A 01 push 1

007F133B E8 90 FD FF FF call TestD::TestD (07F10D0h) 已用时间<=1ms
```

- · 构造时, 会先构造父类, 父类是虚继承自其父类的话, 则构造会多出一个参数 (上条)
- 在构造时,除了填写自己的虚表指针,还会覆盖其父类的虚表指针
- 析构
  - 。 析构与构造是相反的过程

#### 注意

菱形继承需要考虑不同情况下对象内存的布局样式,详情参考一阶段《继承中的对象内存分布》相关知识点