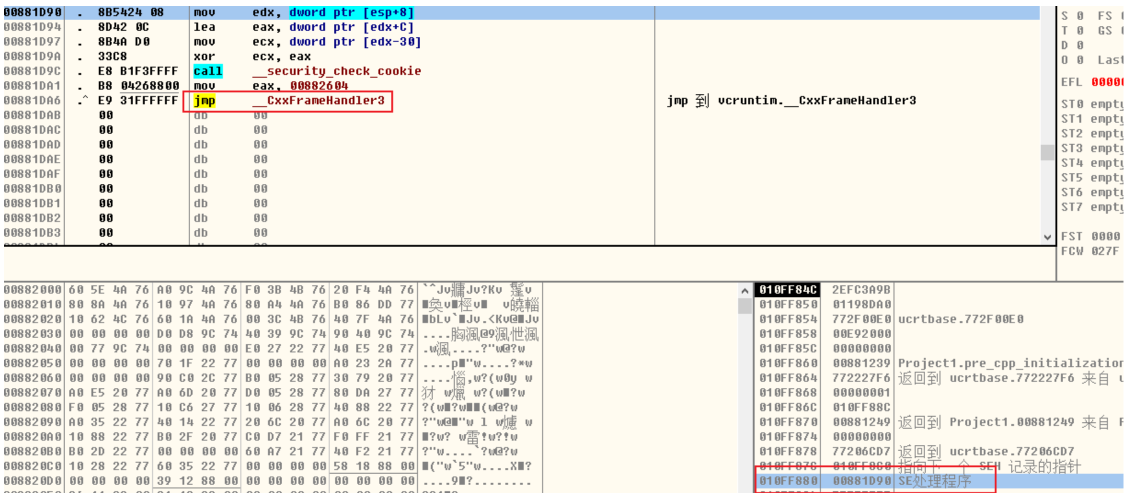
异常

`RaiseException`用于抛出异常，C++的`throw`就是用的它

快速定位catch块处理过程

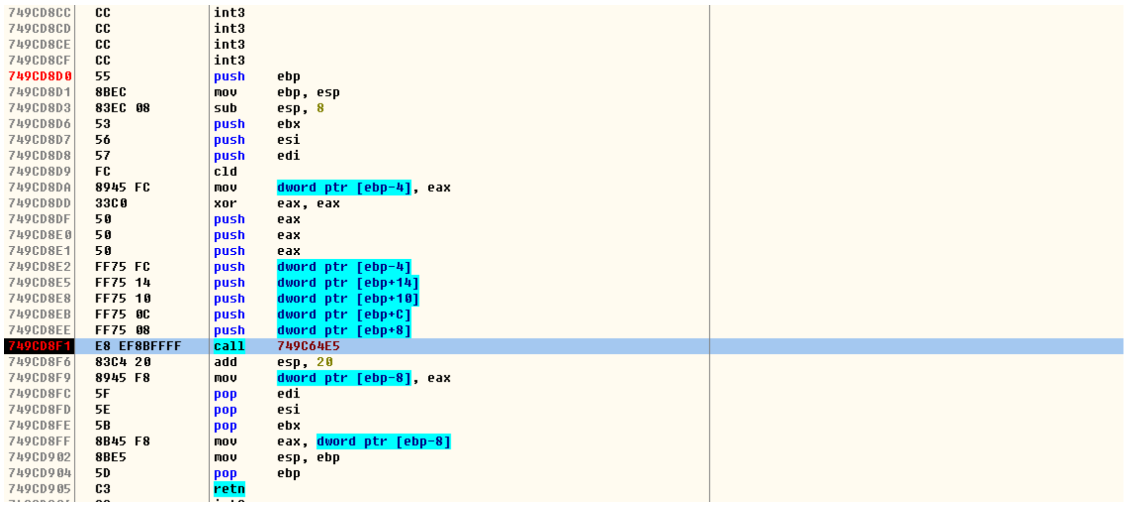
锁定SEH链，找参数多的跟进去，最后定位到`call reg`

1. 跟进SEH链



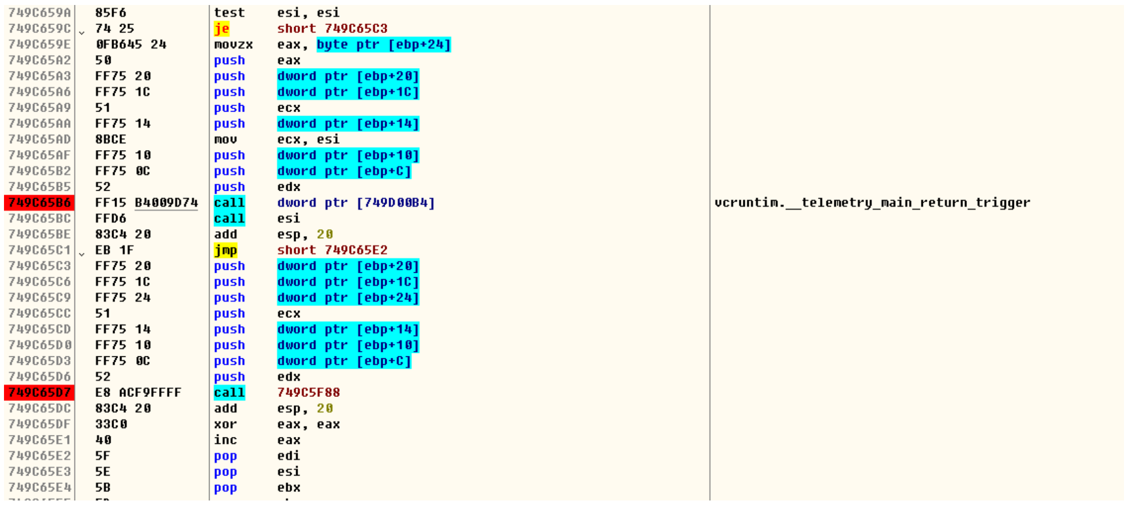
```
00881D90  .  8B5424 08    mov     edx, dword ptr [esp+8]
00881D94  .  8D42 0C      lea     eax, dword ptr [edx+C]
00881D97  .  8B4A D0      mov     ecx, dword ptr [edx-30]
00881D9A  .  33C8         xor     ecx, eax
00881D9C  .  E8 B1F3FFFF  call    __security_check_cookie
00881DA1  .  B8 04268800  mov     eax, 00882604
00881DA6  .^ E9 31FFFFFF  jmp     __CxxFrameHandler3          jmp 到 vcruntim.__CxxFrameHandler3
00881DAB     00           db      00
00881DAC     00           db      00
00881DAD     00           db      00
00881DAE     00           db      00
00881DAF     00           db      00
00881DB0     00           db      00
00881DB1     00           db      00
00881DB2     00           db      00
00881DB3     00           db      00

S 0  FS
T 0  GS
D 0
O 0  Last
EFL  0000
ST0 empty
ST1 empty
ST2 empty
ST3 empty
ST4 empty
ST5 empty
ST6 empty
ST7 empty
FST  0000
FCW  027F

00882000 60 5E 4A 76 A0 9C 4A 76 F0 3B 4B 76 20 F4 4A 76 `^Jv鏞Jv?Kv 鬘v
00882010 80 8A 4A 76 10 97 4A 76 80 A4 4A 76 B0 86 DD 77 ■奐v■桱v■  v皢軡
00882020 10 62 4C 76 60 1A 4A 76 00 3C 4B 76 40 7F 4A 76 ■bLv`■Jv.<Kv@■Jv
00882030 00 00 00 00 D0 D8 9C 74 40 39 9C 74 90 40 9C 74 ....胸渢@9渢怈渢
00882040 00 77 9C 74 00 00 00 00 E0 27 22 77 40 E5 20 77 .w渢....?"w@?w
00882050 00 00 00 00 70 1F 22 77 00 00 00 00 A0 23 2A 77 ....p■"w....?*w
00882060 00 00 00 00 90 C0 2C 77 B0 05 28 77 30 79 20 77 ....惱,w?(w0y w
00882070 A0 E5 20 77 A0 6D 20 77 D0 05 28 77 80 DA 27 77 犲 w燖 w?(w■?w
00882080 F0 05 28 77 10 C6 27 77 10 06 28 77 40 88 22 77 ?(w■?w■■(w@?w
00882090 A0 35 22 77 40 14 22 77 20 6C 20 77 A0 6C 20 77 ?"w@■"w l w爈 w
008820A0 10 88 22 77 B0 2F 20 77 C0 D7 21 77 F0 FF 21 77 ■?w? w雷!w?!w
008820B0 B0 2D 22 77 00 00 00 00 60 A7 21 77 40 F2 21 77 ?"w....`?w@?w
008820C0 10 28 22 77 60 35 22 77 00 00 00 00 58 18 88 00 ■("w`5"w....X■?
008820D0 00 00 00 00 39 12 88 00 00 00 00 00 00 00 00 00 ....9■?........

010FF84C  2EFC3A9B
010FF850  01198DA0
010FF854  772F00E0  ucrtbase.772F00E0
010FF858  00E92000
010FF85C  00000000
010FF860  00881239  Project1.pre_cpp_initialization
010FF864  772227F6  返回到 ucrtbase.772227F6 来自 u
010FF868  00000001
010FF86C  010FF88C
010FF870  00881249  返回到 Project1.00881249 来自 P
010FF874  00000000
010FF878  77206CD7  返回到 ucrtbase.77206CD7
010FF87C  010FF8C0  指向下一个 SEH 记录的指针
010FF880  00881D90  SE处理程序
```

2. 参数多的`call`跟进



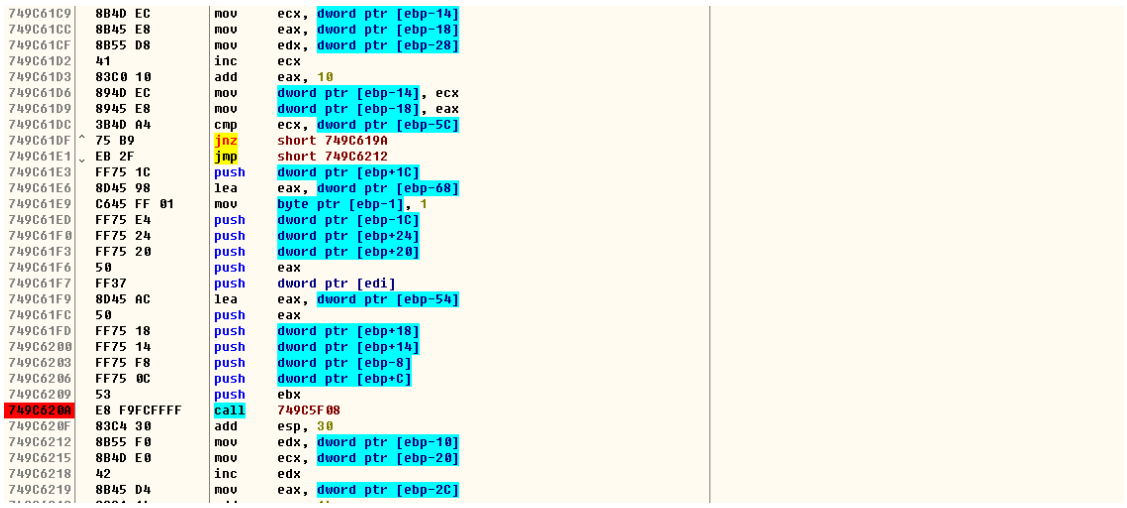
```
749CD8CC   CC          int3
749CD8CD   CC          int3
749CD8CE   CC          int3
749CD8CF   CC          int3
749CD8D0   55          push    ebp
749CD8D1   8BEC        mov     ebp, esp
749CD8D3   83EC 08     sub     esp, 8
749CD8D6   53          push    ebx
749CD8D7   56          push    esi
749CD8D8   57          push    edi
749CD8D9   FC          cld
749CD8DA   8945 FC     mov     dword ptr [ebp-4], eax
749CD8DD   33C0        xor     eax, eax
749CD8DF   50          push    eax
749CD8E0   50          push    eax
749CD8E1   50          push    eax
749CD8E2   FF75 FC     push    dword ptr [ebp-4]
749CD8E5   FF75 14     push    dword ptr [ebp+14]
749CD8E8   FF75 10     push    dword ptr [ebp+10]
749CD8EB   FF75 0C     push    dword ptr [ebp+C]
749CD8EE   FF75 08     push    dword ptr [ebp+8]
749CD8F1   E8 EF8BFFFF call    749C64E5
749CD8F6   83C4 20     add     esp, 20
749CD8F9   8945 F8     mov     dword ptr [ebp-8], eax
749CD8FC   5F          pop     edi
749CD8FD   5E          pop     esi
749CD8FE   5B          pop     ebx
749CD8FF   8B45 F8     mov     eax, dword ptr [ebp-8]
749CD902   8BE5        mov     esp, ebp
749CD904   5D          pop     ebp
749CD905   C3          retn
```

3. 双分支下断点，看来哪一个



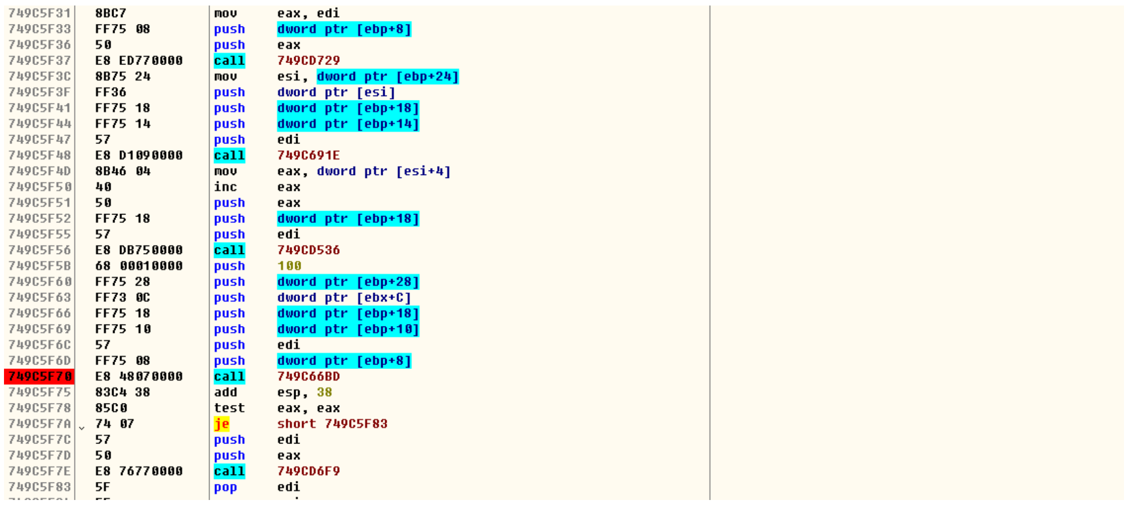
```
749C659A   85F6           test   esi, esi
749C659C ⌄ 74 25          je     short 749C65C3
749C659E   0FB645 24      movzx  eax, byte ptr [ebp+24]
749C65A2   50             push   eax
749C65A3   FF75 20        push   dword ptr [ebp+20]
749C65A6   FF75 1C        push   dword ptr [ebp+1C]
749C65A9   51             push   ecx
749C65AA   FF75 14        push   dword ptr [ebp+14]
749C65AD   8BCE           mov    ecx, esi
749C65AF   FF75 10        push   dword ptr [ebp+10]
749C65B2   FF75 0C        push   dword ptr [ebp+C]
749C65B5   52             push   edx
749C65B6   FF15 B4009D74  call   dword ptr [749D00B4]        vcruntim.__telemetry_main_return_trigger
749C65BC   FFD6           call   esi
749C65BE   83C4 20        add    esp, 20
749C65C1 ⌄ EB 1F          jmp    short 749C65E2
749C65C3   FF75 20        push   dword ptr [ebp+20]
749C65C6   FF75 1C        push   dword ptr [ebp+1C]
749C65C9   FF75 24        push   dword ptr [ebp+24]
749C65CC   51             push   ecx
749C65CD   FF75 14        push   dword ptr [ebp+14]
749C65D0   FF75 10        push   dword ptr [ebp+10]
749C65D3   FF75 0C        push   dword ptr [ebp+C]
749C65D6   52             push   edx
749C65D7   E8 ACF9FFFF    call   749C5F88
749C65DC   83C4 20        add    esp, 20
749C65DF   33C0           xor    eax, eax
749C65E1   40             inc    eax
749C65E2   5F             pop    edi
749C65E3   5E             pop    esi
749C65E4   5B             pop    ebx
```

4. 跟进参数多的`call`



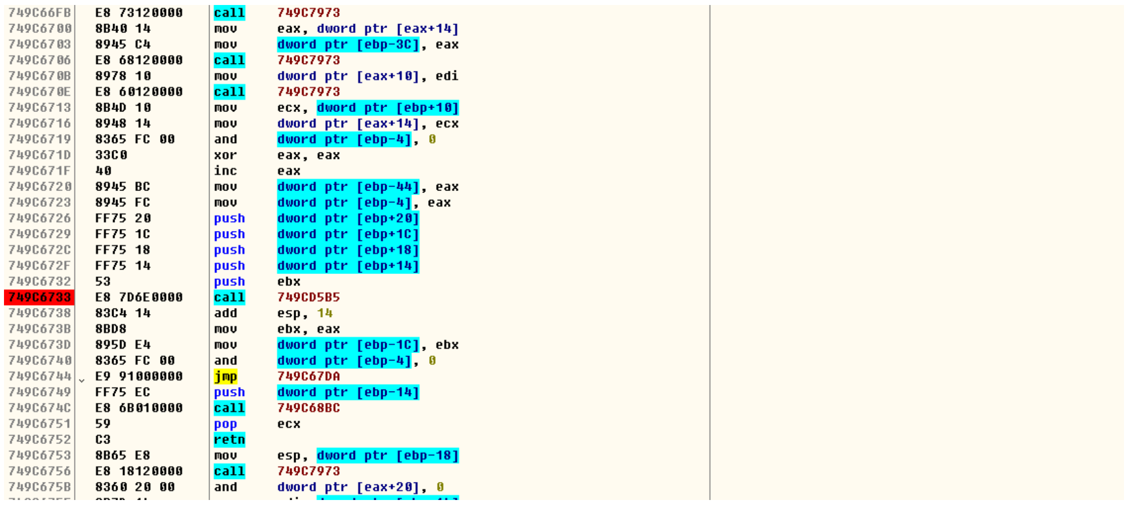
```
749C61C9   8B4D EC        mov    ecx, dword ptr [ebp-14]
749C61CC   8B45 E8        mov    eax, dword ptr [ebp-18]
749C61CF   8B55 D8        mov    edx, dword ptr [ebp-28]
749C61D2   41             inc    ecx
749C61D3   83C0 10        add    eax, 10
749C61D6   894D EC        mov    dword ptr [ebp-14], ecx
749C61D9   8945 E8        mov    dword ptr [ebp-18], eax
749C61DC   3B4D A4        cmp    ecx, dword ptr [ebp-5C]
749C61DF ^ 75 B9          jnz    short 749C619A
749C61E1 ⌄ EB 2F          jmp    short 749C6212
749C61E3   FF75 1C        push   dword ptr [ebp+1C]
749C61E6   8D45 98        lea    eax, dword ptr [ebp-68]
749C61E9   C645 FF 01     mov    byte ptr [ebp-1], 1
749C61ED   FF75 E4        push   dword ptr [ebp-1C]
749C61F0   FF75 24        push   dword ptr [ebp+24]
749C61F3   FF75 20        push   dword ptr [ebp+20]
749C61F6   50             push   eax
749C61F7   FF37           push   dword ptr [edi]
749C61F9   8D45 AC        lea    eax, dword ptr [ebp-54]
749C61FC   50             push   eax
749C61FD   FF75 18        push   dword ptr [ebp+18]
749C6200   FF75 14        push   dword ptr [ebp+14]
749C6203   FF75 F8        push   dword ptr [ebp-8]
749C6206   FF75 0C        push   dword ptr [ebp+C]
749C6209   53             push   ebx
749C620A   E8 F9FCFFFF    call   749C5F08
749C620F   83C4 30        add    esp, 30
749C6212   8B55 F0        mov    edx, dword ptr [ebp-10]
749C6215   8B4D E0        mov    ecx, dword ptr [ebp-20]
749C6218   42             inc    edx
749C6219   8B45 D4        mov    eax, dword ptr [ebp-2C]
```

5. 同理



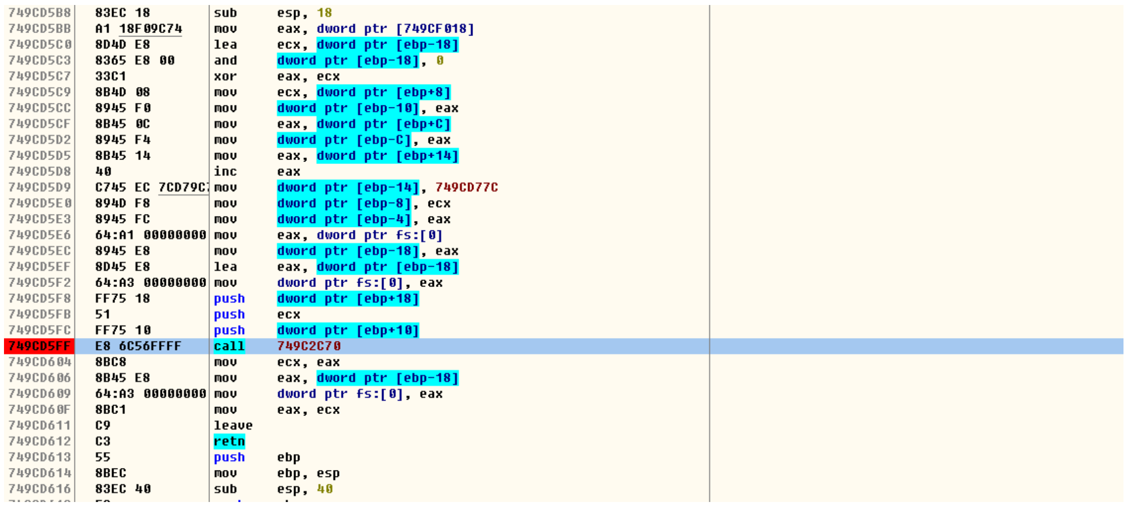
```
749C5F31   8BC7           mov    eax, edi
749C5F33   FF75 08        push   dword ptr [ebp+8]
749C5F36   50             push   eax
749C5F37   E8 ED770000    call   749CD729
749C5F3C   8B75 24        mov    esi, dword ptr [ebp+24]
749C5F3F   FF36           push   dword ptr [esi]
749C5F41   FF75 18        push   dword ptr [ebp+18]
749C5F44   FF75 14        push   dword ptr [ebp+14]
749C5F47   57             push   edi
749C5F48   E8 D1090000    call   749C691E
749C5F4D   8B46 04        mov    eax, dword ptr [esi+4]
749C5F50   40             inc    eax
749C5F51   50             push   eax
749C5F52   FF75 18        push   dword ptr [ebp+18]
749C5F55   57             push   edi
749C5F56   E8 DB750000    call   749CD536
749C5F5B   68 00010000    push   100
749C5F60   FF75 28        push   dword ptr [ebp+28]
749C5F63   FF73 0C        push   dword ptr [ebx+C]
749C5F66   FF75 18        push   dword ptr [ebp+18]
749C5F69   FF75 10        push   dword ptr [ebp+10]
749C5F6C   57             push   edi
749C5F6D   FF75 08        push   dword ptr [ebp+8]
749C5F70   E8 48070000    call   749C66BD
749C5F75   83C4 38        add    esp, 38
749C5F78   85C0           test   eax, eax
749C5F7A ⌄ 74 07          je     short 749C5F83
749C5F7C   57             push   edi
749C5F7D   50             push   eax
749C5F7E   E8 76770000    call   749CD6F9
749C5F83   5F             pop    edi
```

6. 同理



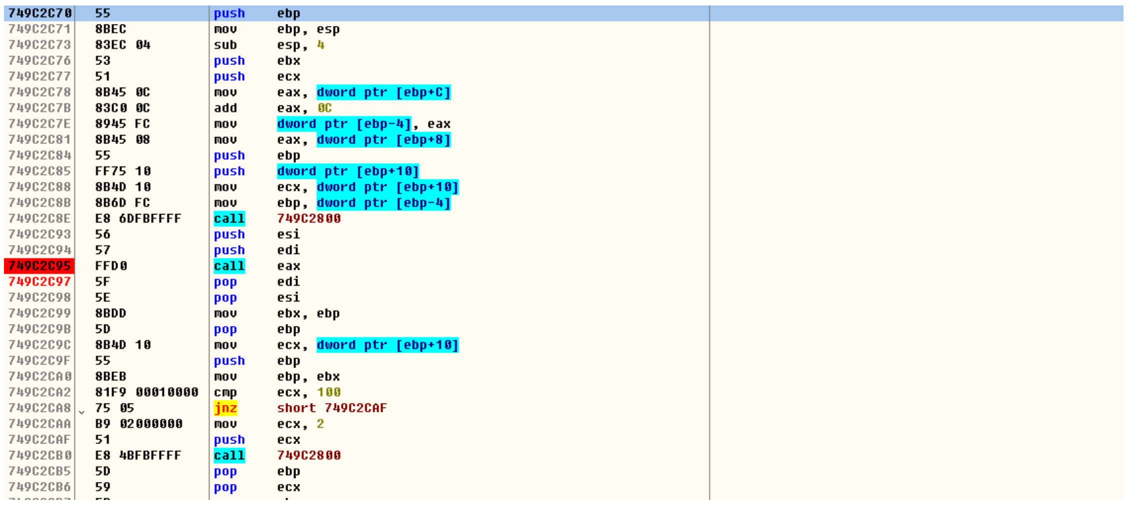
```
749C66FB   E8 73120000    call   749C7973
749C6700   8B40 14        mov    eax, dword ptr [eax+14]
749C6703   8945 C4        mov    dword ptr [ebp-3C], eax
749C6706   E8 68120000    call   749C7973
749C670B   8978 10        mov    dword ptr [eax+10], edi
749C670E   E8 60120000    call   749C7973
749C6713   8B4D 10        mov    ecx, dword ptr [ebp+10]
749C6716   8948 14        mov    dword ptr [eax+14], ecx
749C6719   8365 FC 00     and    dword ptr [ebp-4], 0
749C671D   33C0           xor    eax, eax
749C671F   40             inc    eax
749C6720   8945 BC        mov    dword ptr [ebp-44], eax
749C6723   8945 FC        mov    dword ptr [ebp-4], eax
749C6726   FF75 20        push   dword ptr [ebp+20]
749C6729   FF75 1C        push   dword ptr [ebp+1C]
749C672C   FF75 18        push   dword ptr [ebp+18]
749C672F   FF75 14        push   dword ptr [ebp+14]
749C6732   53             push   ebx
749C6733   E8 7D6E0000    call   749CD5B5
749C6738   83C4 14        add    esp, 14
749C673B   8BD8           mov    ebx, eax
749C673D   895D E4        mov    dword ptr [ebp-1C], ebx
749C6740   8365 FC 00     and    dword ptr [ebp-4], 0
749C6744 ⌄ E9 91000000    jmp    749C67DA
749C6749   FF75 EC        push   dword ptr [ebp-14]
749C674C   E8 6B010000    call   749C68BC
749C6751   59             pop    ecx
749C6752   C3             retn
749C6753   8B65 E8        mov    esp, dword ptr [ebp-18]
749C6756   E8 18120000    call   749C7973
749C675B   8360 20 00     and    dword ptr [eax+20], 0
```

7. 跟进剩下一个`call`



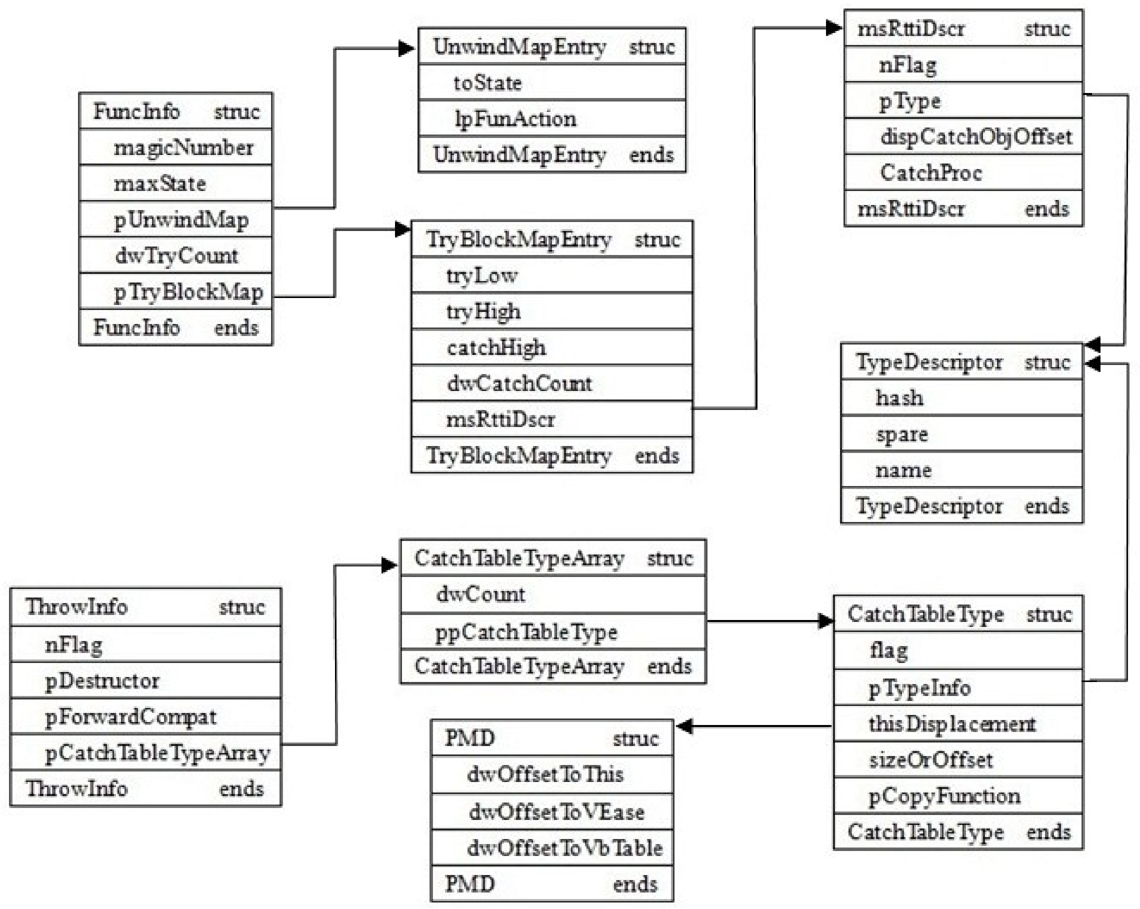
```
749CD5B8   83EC 18           sub    esp, 18
749CD5BB   A1 18F09C74       mov    eax, dword ptr [749CF018]
749CD5C0   8D4D E8           lea    ecx, dword ptr [ebp-18]
749CD5C3   8365 E8 00        and    dword ptr [ebp-18], 0
749CD5C7   33C1              xor    eax, ecx
749CD5C9   8B4D 08           mov    ecx, dword ptr [ebp+8]
749CD5CC   8945 F0           mov    dword ptr [ebp-10], eax
749CD5CF   8B45 0C           mov    eax, dword ptr [ebp+C]
749CD5D2   8945 F4           mov    dword ptr [ebp-C], eax
749CD5D5   8B45 14           mov    eax, dword ptr [ebp+14]
749CD5D8   40                inc    eax
749CD5D9   C745 EC 7CD79C7   mov    dword ptr [ebp-14], 749CD77C
749CD5E0   894D F8           mov    dword ptr [ebp-8], ecx
749CD5E3   8945 FC           mov    dword ptr [ebp-4], eax
749CD5E6   64:A1 00000000    mov    eax, dword ptr fs:[0]
749CD5EC   8945 E8           mov    dword ptr [ebp-18], eax
749CD5EF   8D45 E8           lea    eax, dword ptr [ebp-18]
749CD5F2   64:A3 00000000    mov    dword ptr fs:[0], eax
749CD5F8   FF75 18           push   dword ptr [ebp+18]
749CD5FB   51                push   ecx
749CD5FC   FF75 10           push   dword ptr [ebp+10]
749CD5FF   E8 6C56FFFF       call   749C2C70
749CD604   8BC8              mov    ecx, eax
749CD606   8B45 E8           mov    eax, dword ptr [ebp-18]
749CD609   64:A3 00000000    mov    dword ptr fs:[0], eax
749CD60F   8BC1              mov    eax, ecx
749CD611   C9                leave
749CD612   C3                retn
749CD613   55                push   ebp
749CD614   8BEC              mov    ebp, esp
749CD616   83EC 40           sub    esp, 40
```

8. 最后定位到`call reg`，即`call`的目标就是`catch`过程



```
749C2C70   55                push   ebp
749C2C71   8BEC              mov    ebp, esp
749C2C73   83EC 04           sub    esp, 4
749C2C76   53                push   ebx
749C2C77   51                push   ecx
749C2C78   8B45 0C           mov    eax, dword ptr [ebp+C]
749C2C7B   83C0 0C           add    eax, 0C
749C2C7E   8945 FC           mov    dword ptr [ebp-4], eax
749C2C81   8B45 08           mov    eax, dword ptr [ebp+8]
749C2C84   55                push   ebp
749C2C85   FF75 10           push   dword ptr [ebp+10]
749C2C88   8B4D 10           mov    ecx, dword ptr [ebp+10]
749C2C8B   8B6D FC           mov    ebp, dword ptr [ebp-4]
749C2C8E   E8 6DFBFFFF       call   749C2800
749C2C93   56                push   esi
749C2C94   57                push   edi
749C2C95   FFD0              call   eax
749C2C97   5F                pop    edi
749C2C98   5E                pop    esi
749C2C99   8BDD              mov    ebx, ebp
749C2C9B   5D                pop    ebp
749C2C9C   8B4D 10           mov    ecx, dword ptr [ebp+10]
749C2C9F   55                push   ebp
749C2CA0   8BEB              mov    ebp, ebx
749C2CA2   81F9 00010000     cmp    ecx, 100
749C2CA8 ⌄ 75 05             jnz    short 749C2CAF
749C2CAA   B9 02000000       mov    ecx, 2
749C2CAF   51                push   ecx
749C2CB0   E8 4BFBFFFF       call   749C2800
749C2CB5   5D                pop    ebp
749C2CB6   59                pop    ecx
```

异常回调与抛出异常



重要结构字段说明

FuncInfo

```
FuncInfo            struc              ; (sizeof=0x14)
   magicNumber      dd       ?         ; 编译器生成标记固定数字 0x19930520
   maxState         dd       ?         ; 最大栈展开数的下标值
   pUnwindMap       dd       ?         ; 指向栈展开函数表的指针，指向 UnwindMapEntry 表结构
   dwTryCount       dd       ?         ;  try 块数量
   pTryBlockMap     dd       ?         ;  try 块列表，指向 TryBlockMapEntry 表结构
FuncInfo            ends
```

TryBlockMapEntry

```
TryBlockMapEntry             struc          ; (sizeof=0x14)
   tryLow                    dd ?           ; try 块的最小状态索引，用于范围检查
   tryHigh                   dd ?           ; try 块的最大状态索引，用于范围检查
   catchHigh                 dd ?           ; catch 块的最高状态索引，用于范围检查
   dwCatchCount              dd ?           ; catch 块个数
   pCatchHandlerArray        dd ?           ; catch 块描述，指向 _msRttiDscr 表结构
TryBlockMapEntry             ends
```

_msRttiDscr

```
_msRttiDscr              struc    ; (sizeof=0x10)
    nFlag                dd ?     ; 用于 catch 块的匹配检查
    pType                dd ?     ; catch 块要捕捉的类型 ，指向 TypeDescriptor 表结构
    dispCatchObjOffset   dd ?     ; 用于定位异常对象在当前 EBP 中的偏移位置
    CatchProc            dd ?     ; catch 块的首地址
_msRttiDscr              ends
```

其中，`nFlag`用于检查catch块类型的匹配

- `nFlag = 1`，常量
- `nFlag = 2`，变量
- `nFlag = 4`，未知
- `nFlag = 8`，引用

TypeDescriptor

```
TypeDescriptor           struc
    hash                 dd ?     ; 类型名称的 Hash 数值
    spare                dd ?     ; 保留，可能用于 RTTI 名称记录
    name                 db ?     ; 类型名称
TypeDescriptor           ends
```

ThrowInfo

```
ThrowInfo                struc    ; (sizeof=0x10)
    nFlag                dd ?     ; 抛出异常类型标记
    pDestructor          dd ?     ; 异常对象的析构函数地址
    pForwardCompat       dd ?     ; 未知
    pCatchTableTypeArray dd ?     ; catch 块类型表 ，指向 CatchTableTypeArray 表结构
ThrowInfo                ends
```

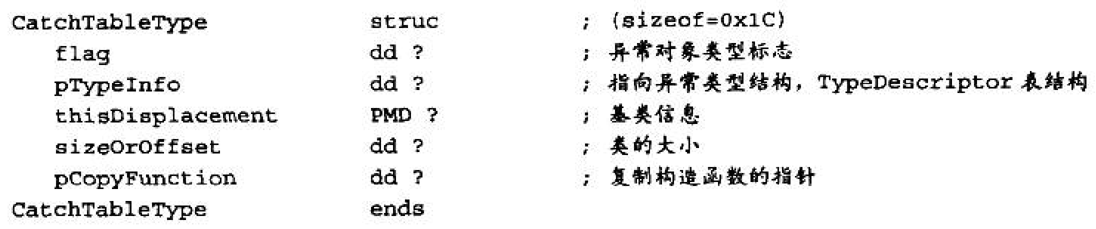
其中，`nFlag`用于检查catch块类型的匹配

- `nFlag = 1`，常量
- `nFlag = 2`，变量

CatchTableTypeArray

```
CatchTableTypeArray      struc    ; (sizeof=0x8)
    dwCount              dd ?     ; CatchTableType 数组包含的元素个数
    ppCatchTableType     dd ?     ; catch 块的类型信息，类型为 CatchTableType**
CatchTableTypeArray      ends
```

CatchTableType

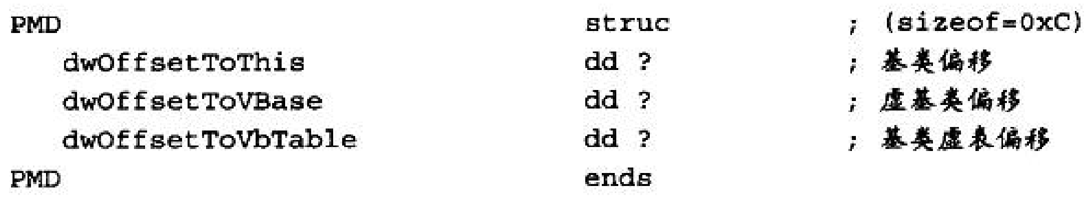


```
CatchTableType          struc           ; (sizeof=0x1C)
    flag                dd ?            ; 异常对象类型标志
    pTypeInfo           dd ?            ; 指向异常类型结构, TypeDescriptor 表结构
    thisDisplacement    PMD ?           ; 基类信息
    sizeOrOffset        dd ?            ; 类的大小
    pCopyFunction       dd ?            ; 复制构造函数的指针
CatchTableType          ends
```

其中，`flag`标记用于判断异常对象的类型

- `0x1`：简单类型赋值
- `0x2`：已被捕获
- `0x4`：有虚表基类赋值
- `0x8`：指针和引用类型赋值

当异常类型为对象时，由于对象存在基类等相关信息，则利用`PMD thisDisplacement`成员保存基类信息



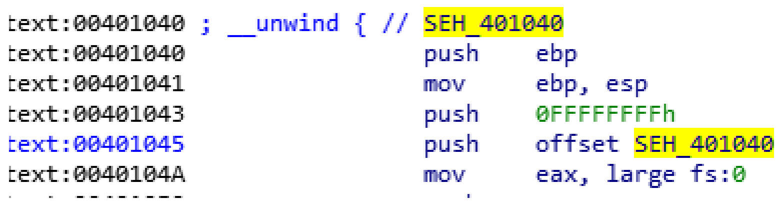
```
PMD                         struc       ; (sizeof=0xC)
    dwOffsetToThis          dd ?        ; 基类偏移
    dwOffsetToVBase         dd ?        ; 虚基类偏移
    dwOffsetToVbTable       dd ?        ; 基类虚表偏移
PMD                         ends
```

细节

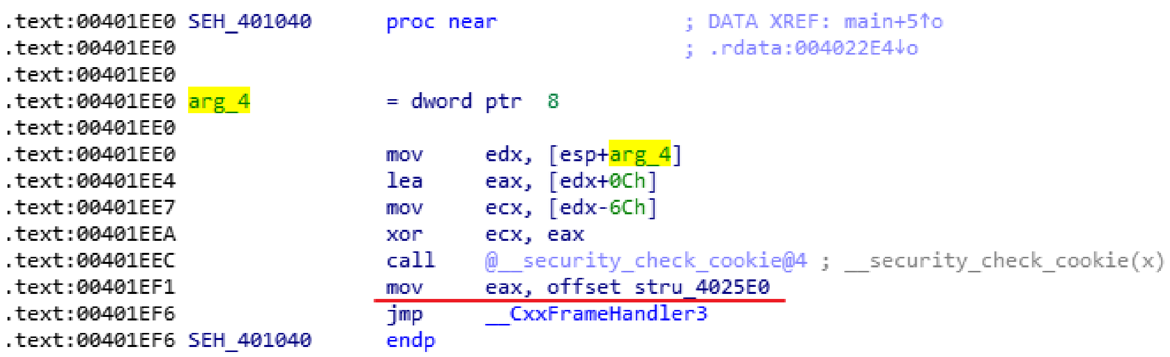
上方路线

FuncInfo --> TryBlockMapEntry --> _msRttiDscr --> TypeDescriptor

在具备异常处理功能的函数中，编译器会在函数入口处注册一个异常回调函数，当该函数中抛出异常时，此回调将被执行，即SEH

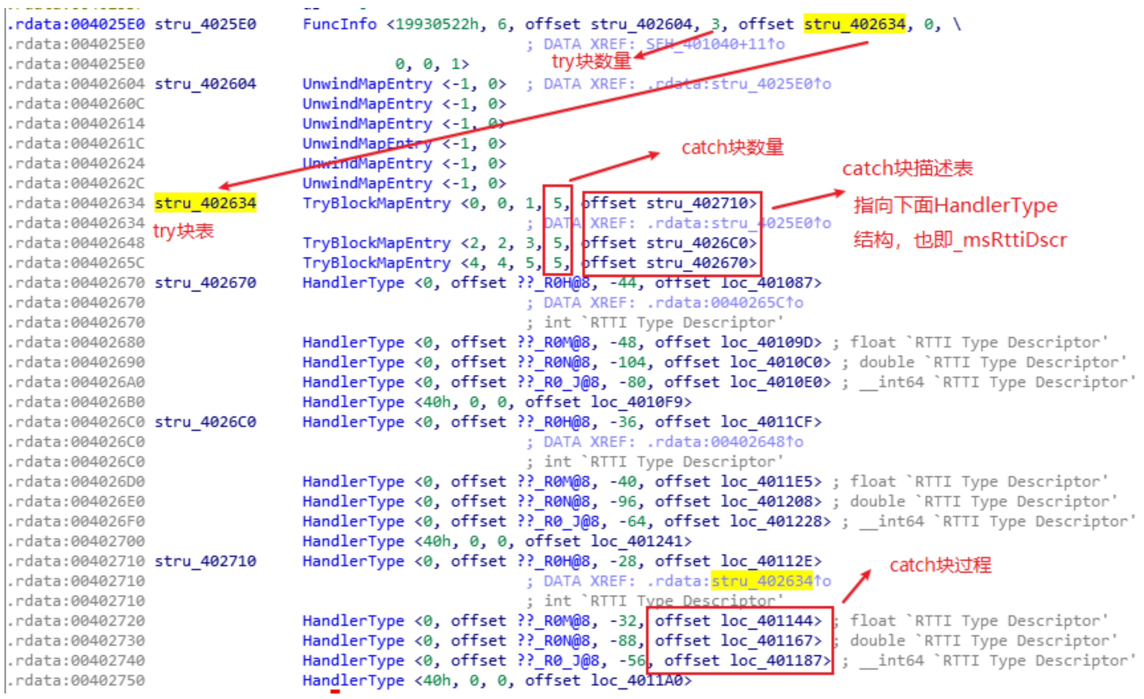


```
text:00401040 ; __unwind { // SEH_401040
text:00401040                 push    ebp
text:00401041                 mov     ebp, esp
text:00401043                 push    0FFFFFFFFh
text:00401045                 push    offset SEH_401040
text:0040104A                 mov     eax, large fs:0
```



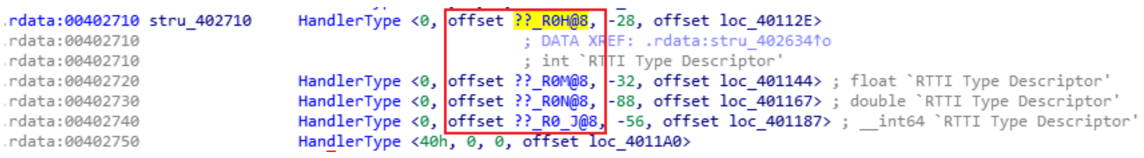
```
.text:00401EE0 SEH_401040      proc near               ; DATA XREF: main+5↑o
.text:00401EE0                                         ; .rdata:004022E4↓o
.text:00401EE0
.text:00401EE0 arg_4           = dword ptr  8
.text:00401EE0
.text:00401EE0                 mov     edx, [esp+arg_4]
.text:00401EE4                 lea     eax, [edx+0Ch]
.text:00401EE7                 mov     ecx, [edx-6Ch]
.text:00401EEA                 xor     ecx, eax
.text:00401EEC                 call    @__security_check_cookie@4 ; __security_check_cookie(x)
.text:00401EF1                 mov     eax, offset stru_4025E0
.text:00401EF6                 jmp     __CxxFrameHandler3
.text:00401EF6 SEH_401040      endp
```

此函数会传给`eax`一个全局变量给`__CxxFrameHandler3`作为参数，此全局变量则是`FuncInfo`表，根据此表可以找到所有的`try-catch`信息

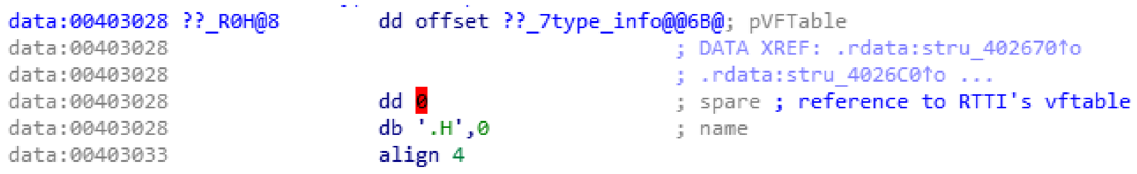


```
.rdata:004025E0 stru_4025E0     FuncInfo <19930522h, 6, offset stru_402604, 3, offset stru_402634, 0, \
.rdata:004025E0                                   ; DATA XREF: SEH_401040+11↑o
.rdata:004025E0                           0, 0, 1>
.rdata:00402604 stru_402604     UnwindMapEntry <-1, 0>  ; DATA XREF: .rdata:stru_4025E0↑o
.rdata:0040260C                 UnwindMapEntry <-1, 0>
.rdata:00402614                 UnwindMapEntry <-1, 0>
.rdata:0040261C                 UnwindMapEntry <-1, 0>
.rdata:00402624                 UnwindMapEntry <-1, 0>
.rdata:0040262C                 UnwindMapEntry <-1, 0>
.rdata:00402634 stru_402634     TryBlockMapEntry <0, 0, 1, 5, offset stru_402710>
.rdata:00402634                                   ; DATA XREF: .rdata:stru_4025E0↑o
.rdata:00402648                 TryBlockMapEntry <2, 2, 3, 5, offset stru_4026C0>
.rdata:0040265C                 TryBlockMapEntry <4, 4, 5, 5, offset stru_402670>
.rdata:00402670 stru_402670     HandlerType <0, offset ??_R0H@8, -44, offset loc_401087>
.rdata:00402670                                   ; DATA XREF: .rdata:0040265C↑o
.rdata:00402670                                   ; int `RTTI Type Descriptor'
.rdata:00402680                 HandlerType <0, offset ??_R0M@8, -48, offset loc_40109D> ; float `RTTI Type Descriptor'
.rdata:00402690                 HandlerType <0, offset ??_R0N@8, -104, offset loc_4010C0> ; double `RTTI Type Descriptor'
.rdata:004026A0                 HandlerType <0, offset ??_R0_J@8, -80, offset loc_4010E0> ; __int64 `RTTI Type Descriptor'
.rdata:004026B0                 HandlerType <40h, 0, 0, offset loc_4010F9>
.rdata:004026C0 stru_4026C0     HandlerType <0, offset ??_R0H@8, -36, offset loc_4011CF>
.rdata:004026C0                                   ; DATA XREF: .rdata:00402648↑o
.rdata:004026C0                                   ; int `RTTI Type Descriptor'
.rdata:004026D0                 HandlerType <0, offset ??_R0M@8, -40, offset loc_4011E5> ; float `RTTI Type Descriptor'
.rdata:004026E0                 HandlerType <0, offset ??_R0N@8, -96, offset loc_401208> ; double `RTTI Type Descriptor'
.rdata:004026F0                 HandlerType <0, offset ??_R0_J@8, -64, offset loc_401228> ; __int64 `RTTI Type Descriptor'
.rdata:00402700                 HandlerType <40h, 0, 0, offset loc_401241>
.rdata:00402710 stru_402710     HandlerType <0, offset ??_R0H@8, -28, offset loc_40112E>
.rdata:00402710                                   ; DATA XREF: .rdata:stru_402634↑o
.rdata:00402710                                   ; int `RTTI Type Descriptor'
.rdata:00402720                 HandlerType <0, offset ??_R0M@8, -32, offset loc_401144> ; float `RTTI Type Descriptor'
.rdata:00402730                 HandlerType <0, offset ??_R0N@8, -88, offset loc_401167> ; double `RTTI Type Descriptor'
.rdata:00402740                 HandlerType <0, offset ??_R0_J@8, -56, offset loc_401187> ; __int64 `RTTI Type Descriptor'
.rdata:00402750                 HandlerType <40h, 0, 0, offset loc_4011A0>
```

而在catch描述表中的第二项，保存着异常类型信息



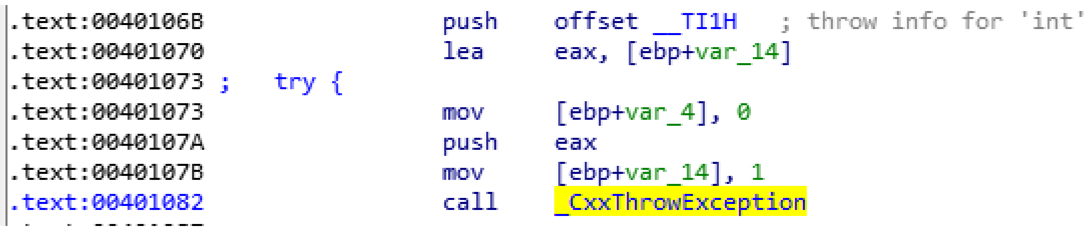
```
.rdata:00402710 stru_402710     HandlerType <0, offset ??_R0H@8, -28, offset loc_40112E>
.rdata:00402710                                   ; DATA XREF: .rdata:stru_402634↑o
.rdata:00402710                                   ; int `RTTI Type Descriptor'
.rdata:00402720                 HandlerType <0, offset ??_R0M@8, -32, offset loc_401144> ; float `RTTI Type Descriptor'
.rdata:00402730                 HandlerType <0, offset ??_R0N@8, -88, offset loc_401167> ; double `RTTI Type Descriptor'
.rdata:00402740                 HandlerType <0, offset ??_R0_J@8, -56, offset loc_401187> ; __int64 `RTTI Type Descriptor'
.rdata:00402750                 HandlerType <40h, 0, 0, offset loc_4011A0>
```



```
data:00403028 ??_R0H@8        dd offset ??_7type_info@@6B@ ; pVFTable
data:00403028                                         ; DATA XREF: .rdata:stru_402670↑o
data:00403028                                         ; .rdata:stru_4026C0↑o ...
data:00403028                 dd 0                    ; spare ; reference to RTTI's vftable
data:00403028                 db '.H',0               ; name
data:00403033                 align 4
```

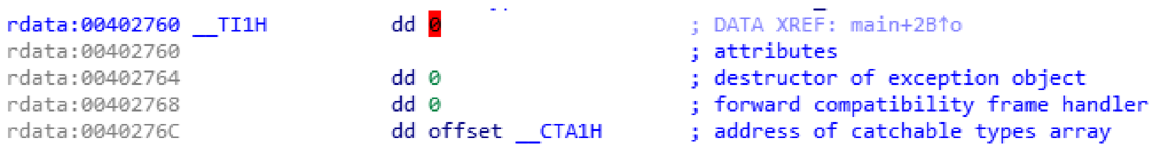
下方路线

在throw抛出异常时，会向异常函数传递一个全局的变量作为参数，此参数就是`ThrowInfo`表



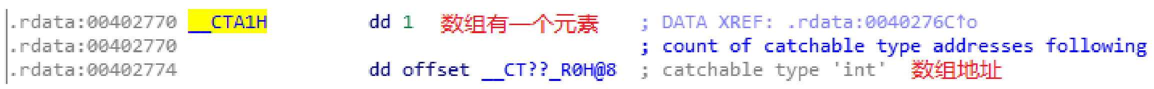
```
.text:0040106B                  push    offset __TI1H   ; throw info for 'int'
.text:00401070                  lea     eax, [ebp+var_14]
.text:00401073 ;    try {
.text:00401073                  mov     [ebp+var_4], 0
.text:0040107A                  push    eax
.text:0040107B                  mov     [ebp+var_14], 1
.text:00401082                  call    _CxxThrowException
```

第二个成员是异常对象的析构函数地址



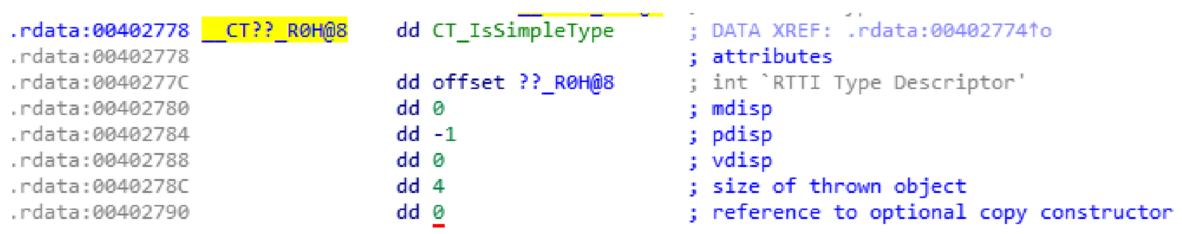
```
rdata:00402760 __TI1H          dd 0                    ; DATA XREF: main+2B↑o
rdata:00402760                                         ; attributes
rdata:00402764                 dd 0                    ; destructor of exception object
rdata:00402768                 dd 0                    ; forward compatibility frame handler
rdata:0040276C                 dd offset __CTA1H       ; address of catchable types array
```

第四个成员则是指向catch表类型数组

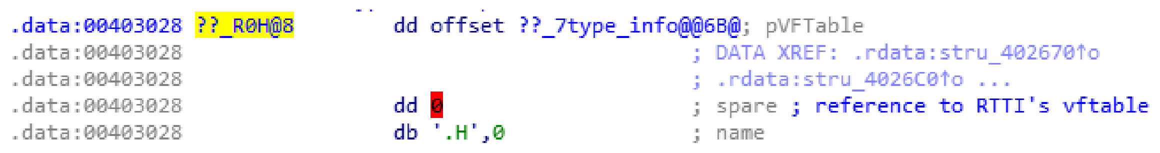


```
.rdata:00402770 __CTA1H         dd 1                    ; DATA XREF: .rdata:0040276C↑o
.rdata:00402770                                         ; count of catchable type addresses following
.rdata:00402774                 dd offset __CT??_R0H@8  ; catchable type 'int'
```

数组元素为`TypeDescriptor`类型



```
.rdata:00402778 __CT??_R0H@8     dd CT_IsSimpleType      ; DATA XREF: .rdata:00402774↑o
.rdata:00402778                                          ; attributes
.rdata:0040277C                  dd offset ??_R0H@8      ; int `RTTI Type Descriptor'
.rdata:00402780                  dd 0                    ; mdisp
.rdata:00402784                  dd -1                   ; pdisp
.rdata:00402788                  dd 0                    ; vdisp
.rdata:0040278C                  dd 4                    ; size of thrown object
.rdata:00402790                  dd 0                    ; reference to optional copy constructor
```



```
.data:00403028 ??_R0H@8          dd offset ??_7type_info@@6B@; pVFTable
.data:00403028                                            ; DATA XREF: .rdata:stru_402670↑o
.data:00403028                                            ; .rdata:stru_4026C0↑o ...
.data:00403028                   dd 0                     ; spare ; reference to RTTI's vftable
.data:00403028                   db '.H',0                ; name
```