#### 2021/05/26 x64汇编与逆向 第3课 流程控制语句的识别

**笔记本:** x64汇编与逆向

创建时间: 2021/5/26 星期三 10:05

作者: ileemi

• 课前会议

• 流程控制语句的识别

• <u>if</u>

• if else

• if else-if

• <u>switch-case</u>

- do while
- while
- for

# 课前会议

64位Release程序的预留空间可能被用来保存寄存器环境(非rcx、rdx、r8、r9寄存器)或者局部变量。

# 流程控制语句的识别

**图形识别法**:通过IDA和x64dbg的跳转线条判断流程控制语句,虚线表示条件跳转(jxx),实线表示无条件跳转(jmp),中间包含的每个蓝色圆点表示一条汇编语句,线的开头表示判断开始。

#### x64dbug定位main函数:

00007557600003150	EQ GO EE EE EE	call tost 755760081145	
		The state of the s	
	THE CONTRACT OF THE PERSON OF		rax:EntryPoin
00007FF7699B21F1			rax:EntryPoin
00007FF7699B21F4			
00007FF7699B21F9	E8 22 01 00 00	call test.7FF7699B2320	mair
00007FF7699B21FE	89 44 24 28	mov dword ptrss:[rsp+28],eax	
00007FF7699B2202	E8 60 F0 FF FF	<pre>call test.7FF7699B1267</pre>	
00007FF7699B2320	48 83 EC 48	sub rsp,48	
00007FF7699B2324	E8 68 ED FF FF	call test.7FF7699B1091	
00007FF7699B2329	48 89 44 24 28	mov qword ptrss:[rsp+28],rax	rax:EntryPoin
00007FF7699B232E	E8 B1 EF FF FF	<pre>call test.7FF7699B12E4</pre>	
00007FF7699B2333	48 8B 00	mov rax, qword ptrds: [rax]	rax:EntryPoin
00007FF7699B2336	48 89 44 24 30	mov qword ptrss:[rsp+30],rax	rax:EntryPoin
00007FF7699B233B	E8 72 EF FF FF	call test.7FF7699B12B2	
00007FF7699B2340	8B 00	mov eax, dword ptrds: [rax]	rax:EntryPoin
00007FF7699B2342	89 44 24 20	mov dword ptrss:[rsp+20],eax	
00007FF7699B2346	4C 8B 44 24 28	mov r8,qword ptrss:[rsp+28]	
00007FF7699B234B	48 8B 54 24 30	mov rdx, qword ptrss:[rsp+30]	rdx:EntryPoin
00007FF7699B2350	8B 4C 24 20	mov ecx,dword ptrss:[rsp+20]	No. of the State o
00007FF7699B2354	E8 04 EF FF FF	call test.7FF7699B125D	mair
00007FF7699B2359	48 83 C4 48	add rsp,48	
00007FF7699B235D	C3	ret	
	00007FF7699B21F9 00007FF7699B21FE 00007FF7699B2202 00007FF7699B2320 00007FF7699B2324 00007FF7699B2325 00007FF7699B2336 00007FF7699B2336 00007FF7699B2336 00007FF7699B2340 00007FF7699B2340 00007FF7699B2340 00007FF7699B2346 00007FF7699B2348	00007FF7699B21E5         0F B6 C0           00007FF7699B21EA         85 C0           00007FF7699B21EA         74 00           00007FF7699B21EC         48 8B 44 24 38           00007FF7699B21F1         48 8B 98           00007FF7699B21F2         E8 45 F1 FF FF           00007FF7699B21F2         89 44 24 28           00007FF7699B21F3         89 44 24 28           00007FF7699B2202         E8 60 F0 FF FF           00007FF7699B2324         E8 68 ED FF FF           00007FF7699B2324         E8 8B 1 EF FF FF           00007FF7699B2335         48 8B 00           00007FF7699B2336         48 89 44 24 30           00007FF7699B2340         8B 00           00007FF7699B2342         89 44 24 20           00007FF7699B2344         48 8B 44 24 20           00007FF769B2345         48 8B 44 24 20           00007FF769B2346         48 8B 44 24 20           00007FF769B2345         48 8B 44 24 20           00007FF769B2350         8B 4C 24 20           00007FF769B2354         E8 04 EF FF FF           00007FF769B2359         48 83 C4 48	00007FF7699821E5         0F B6 C0         movzx eax,al           00007FF7699821EA         74 0D         je test.7FF7699821F9           00007FF7699821EC         48 88 44 24 38         mov rax,qword ptrss: [rsp+38]           00007FF7699821F1         48 88 68         mov rax,qword ptrds: [rax]           00007FF7699821F2         E8 45 F1 FF FF         call test.7FF76998133E           00007FF7699821F9         E8 22 01 00 00         call test.7FF769982320           00007FF7699821FE         89 44 24 28         mov dword ptrss: [rsp+28],eax           00007FF769982320         48 83 EC 48         sub rsp,48           00007FF769982324         E8 68 ED FF FF         call test.7FF769981091           00007FF769982329         48 89 44 24 28         mov qword ptrss: [rsp+28],rax           00007FF769982332         E8 81 EF FF FF         call test.7FF76998126           00007FF769982333         48 89 44 24 30         mov qword ptrss: [rsp+28],rax           00007FF769982336         48 89 44 24 30         mov rax,qword ptrds: [rax]           00007FF769982340         8B 00         mov ax,dword ptrss: [rsp+20],eax           00007FF769982345         89 44 24 20         mov dword ptrss: [rsp+20],eax           00007FF769982346         48 85 42 4 30         mov rdx,qword ptrss: [rsp+20]           00007FF769982350

识别特征:判断往下跳,上面没有jmp指令

- 单分支,虚线
- 多分支,虚线嵌套

Debug:

```
75 6F | ine test.7FF7699B193 | 48 8D 00 51 lea rex, qword ptr ds | 00007FF7699B9C28: "argc == 1 begin" | 88 B8 F8 F1 call test.7FF7699B10 | 48 BD 00 50 lea rex, qword ptr ds | 00007FF7699B9C40: "argc == 0 begin" | 83 BD E0 00 cmp dword ptr ss: [rb | 16 B8 A3 F8 F1 call test.7FF7699B15 | 18 B8 B E0 00 cmp dword ptr ss: [rb | 16 B8 B E0 00 cmp dword ptr ss: [rb | 16 B8 B E0 00 cmp dword ptr ss: [rb | 16 B8 B E0 00 cmp dword ptr ds | 00007FF7699B9C40: "argc == 0 begin" | 18 B8 B E0 00 cmp dword ptr ds | 00007FF7699B9C40: "argc == 0 begin" | 18 B8 B E0 00 cmp dword ptr ds | 00007FF7699B9C40: "argc == 0 begin" | 18 B8 B E0 00 cmp dword ptr ds | 00007FF7699B9C70: "argc == 1 mid" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C70: "argc == 1 mid" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 4" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 4" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 1 end" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 1 end" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 1 end" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 1 end" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 00007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 000007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 000007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 000007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 000007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 000007FF7699B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 000007FF769B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 000007FF769B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 000007FF769B9C58: "argc == 5" | 18 B0 E0 00 cmp dword ptr ds | 000007FF769B9C58: "argc == 5" | 18 B0 E0 00 cm
                                                                                                                                                                                                                                                                                                                                                                                                                          @00007FF7699B1
if分支识别
```

Release:存在一定的优化,按照结构进行代码还原即可

```
.text:00000001400010A7
.text:00000001400010A9
.text:00000001400010B0
                                                                              short loc 1400010D4
                                                                 lea
call
                                                                              rcx, aArgc1Begin ; "argc == 1 begin' sub_140001060
 .text:00000001400010B5
.text:00000001400010BC
.text:00000001400010C1
                                                                              rcx, aArgc1Mid ; "argc == 1 mid" sub_140001060
                                                                              rcx, aArgclEnd ; "argc == 1 end" sub_140001060 eax, eax
                                                                 lea
                                                                 call
xor
add
 .text:00000001400010C8
.text:00000001400010CD
                                                                              rsp, 28h
 text:00000001400010CF
 text:00000001400010D3
 text:00000001400010D3
.text:00000001400010D4
.text:00000001400010D4
 .text:00000001400010D4 loc_1400010D4:
.text:00000001400010D4
                                                                                                         ; CODE XREF: main+71j
                                                                 cmp
                                                                              short loc_1400010E5
rcx, aArgc5 ; "a
sub_140001060
 text:00000001400010D7
 text:0000001400010D7
text:00000001400010D9
text:00000001400010E0
                                                                                                         ; "argc == 5"
                                                                 call
 text:00000001400010E5
text:00000001400010E5 loc_1400010E5:
.text:00000001400010E5
                                                                                                        ; CODE XREF: main+37↑j
                                                                 xor
 text:00000001400010F7
                                                                 add
                                                                              rsp, 28h
 text:00000001400010EB
                                                                 retn
```

• if 条件为 && 时, ida分析Debug版存在 if 嵌套, 还原成 if 嵌套即可

```
.text:00000001400118C5
.text:00000001400118C7
.text:00000001400118CE
                                                                        short loc_1400118E5
[rbp+0D0h+arg_0], 2
short loc_1400118E5
                                                              cmp
                                                              iz
                                                                        [rbp+0D0h+arg_0], 3
short loc_1400118E5
         .text:00000001400118D0
&& : .text:00000001400118D7
                                                              jz
         .text:00000001400118D9
.text:00000001400118E0
                                                                        rcx, Format
printf
                                                                                                 "argc != 1 && argc != 2 && argc != 3"
         text:00000001400118F5
         text:00000001400118E5 loc_1400118E5:
                                                                                              ; CODE XREF: sub_140011880+451j
                                                                                               ; sub 140011880+4Efj ...
         text:00000001400118E5
     .text:00000001400118E5
```

if 条件为 || 时,ida分析Debug版存在两个虚线交叉,**还原成 ||,获取 if 嵌套** 

```
.text:00000001400118C5
                                             short loc_1400118D0
                                      jz
  .text:00000001400118C7
                                             [rbp+0D0h+arg_0],
                                                                      还原为 if 即可
                                             short loc_1400118DC
  .text:00000001400118CF
                                      inz
  .text:00000001400118D0
  ; CODE XREF: sub_140011880+451j
  .text:00000001400118D0
                                      lea
                                             rcx. Format
                                                            ; "argc == 1 && argc == 2
  text:00000001400118D7
                                      call
                                             printf
  text:00000001400118DC
  .text:00000001400118DC loc_1400118DC:
                                                            ; CODE XREF: sub 140011880+4Efi
.text:00000001400118DC
                                      xor
                                             eax, eax
```

# if else

虚线 if + 实线 else ,两部分有重叠(重叠部分没有代码,有代码就是goto语 句)。

### if else-if

 else-if, IDA会将其识别为 else 中嵌套 if。还原代码时, if else-if 结构可按 if else 结构进行还原, 也可根据需求将其还原成 if else-if 结构。

```
short loc_1400118D8
if .text:0000000140011807
                                                          rcx, aArgc0
sub_14001118B
                                                 lea
                                                                               "argc == 0"
                                                          loc_14001196C
    .text:00000001400118D3
                                                 jmp
    text:00000001400118D8
    .text:00000001400118D8
                                                          ; CODE XREF: sub_140011880+45↑j [rbp+0D0h+arg_0], 1 short loc_14001191C
    text:00000001400118D8 loc_1400118D8:
    .text:00000001400118D8
    .text:00000001400118DF
                                                 jnz
                                                          rcx, aArgc1
sub_14001118B
   lea
call
                                                                            ; "argc == 1"
                                                          rcx, aArgc0Mid ; "argc != 0 mid" sub_14001118B
    .text:00000001400118ED
                                                                                                         Debug
    .text:00000001400118F4
                                                 call
                                                          [rbp+0D0h+arg_0], 2
short loc_14001190E
    .text:00000001400118F9
    .text:0000000140011900
                                                 jnz
    .text:0000000140011902
.text:0000000140011909
                                                          rcx, aArgc2
sub_14001118B
                                                                            ; "argc == 2"
                                                 call
    .text:000000014001190E
.text:000000014001190E loc 14001190E:
                                                          ; CODE XREF: sub_140011880+801j
rcx, aArgc0End ; "argc != 0 end"
sub_14001187
    text:000000014001190E
                                                 lea
    text:0000000140011915
                                                 call
                                                          sub 14001118B
    text:000000014001191A
                                                          short loc_14001196C
                                                 jmp
    .text:000000014001191C ; -----
    text:000000014001191C
                                                                             ; CODE XREF: sub_140011880+5F1j
    text:000000014001191C loc 14001191C:
                                                          [rbp+0D0h+arg_0], 2
    text:000000014001191C
   000000CCE 00000001400118CE: sub_140011880+4E
                                                   (Synchronized with Hex View-1)
```

### switch-case

不适合通过 IDA和x64dbg 查看线结构, case越多, 线也就越多。适合通过特征 查表进行辨别。

64位中case表中记录的是偏移(4字节)而不是地址。根据case值之间的差值大小会有一个表、两个表、if判断三种方式。

#### do while

执行完判断后有向上跳转操作(继续执行相关语句,再判断条件)

```
text:00000001400010B1
    .text:00000001400010B1 loc_1400010B1:
                                                                    ; CODE XREF: main+3E↓j
    .text:00000001400010B1
                                            lea
                                                    rcx, aBegin
                                                                      "begin"
                                                    sub_140001060
  .text:00000001400010B8
                                            call
    .text:00000001400010BD
                                                    ebx, 0Ah
                                            cmp
   .text:00000001400010C0
                                                    short loc 1400010DA
                                            inz
                                                                    ; "if"
    .text:00000001400010C2
                                                    rcx, aIf
                                            lea
                                                    sub_140001060
   .text:00000001400010C9
                                            call
                                                   rcx, aEnd
sub_140001060
                                                                    ; "end"
    .text:00000001400010CE
                                            lea
   .text:00000001400010D5
                                            call
   .text:00000001400010DA
   .text:00000001400010DA loc_1400010DA:
                                                                    ; CODE XREF: main+201j
    .text:00000001400010DA
                                            dec
                                                    ehx
    text:00000001400010DC
                                            cmp
                                                    ebx, edi
                              do while
.text:00000001400010DE
                                                    short loc_1400010B1
                                            jle
```

• break: 其对应的线条会跳出循环判断条件线条

• continue: 线条跳转到语句判断位置

```
Debug:
```

```
.text:00000001400118CC
                                                         rcx, Format
                                                                         ; "begin'
        text:00000001400118D3
                                                 call
        text:00000001400118D8
                                                         [rbp+110h+var_EC], 0
                                                 cmp
       .text:00000001400118DC
                                                         short loc_1400118E0
short loc_140011908 ; continue
       .text:00000001400118DE
                                                 jmp
       .text:00000001400118E0 ; .text:00000001400118E0
       .text:00000001400118E0 loc_1400118E0: .text:00000001400118E0
                                                                          ; CODE XREF: sub_140011880+5Cfj
                                                         rcx, aMid
                                                                          ; "mid
       .text:00000001400118E7
                                                 call
                                                         printf
continutext:00000001400118EC
                                                         [rbp+110h+var_EC], 2
                                                 cmp
       .text:00000001400118F0 .text:00000001400118F2
                                                         short loc_1400118F4
short loc_140011913 ; break
                                                 jmp
       .text:00000001400118F4;
.text:00000001400118F4
       .text:00000001400118F4 loc_1400118F4: .text:00000001400118F4
                                                                          ; CODE XREF: sub_140011880+701j
                                                lea
                                                         rcx, aEnd
                                                                          ; "end
       .text:00000001400118FB
.text:0000000140011900
                                                 call
                                                         printf
                                                         eax, [rbp+110h+var_EC]
                                                 mov
       .text:0000000140011903
                                                 inc
       .text:0000000140011905
                                                         [rbp+110h+var_EC], eax
                                                mov
       .text:0000000140011908
.text:0000000140011908 loc_140011908:
                                                                          ; CODE XREF: sub_140011880+5Efj
    .text:0000000140011908
.text:000000014001190E
                                                         eax, [rbp+110h+arg_0]
[rbp+110h+var_EC], eax
                                                mov
       .text:0000000140011911
                                                jle
                                                         short loc_1400118CC
       .text:0000000140011913
       .text:0000000140011913 loc_140011913:
                                                                          ; CODE XREF: sub_140011880+721j
break 00000CF2 00000001400118F2: sub_140011880+72 (Synchronized with Hex View-1)
Release:
                                                                                    ; CODE XREF: main+41↓j
        .text:00000001400010B0 loc_1400010B0:
                                                                rcx, Format
        .text:00000001400010B0
                                                       lea
                                                                                    ; "begin"
                                                       call
        .text:00000001400010B7
                                                                printf
        .text:00000001400010BC
                                                                 ebx, ebx
                                                       test
        .text:00000001400010BE continue
                                                                 short loc_1400010DF; continue
                                                                                   ; "mid"
        .text:00000001400010C0
                                                       lea
                                                                 rcx, aMid
                                                                 printf
        .text:00000001400010C7
                                                       call
        .text:00000001400010CC
                                                                 ebx, 2
                                                       cmp
        .text:00000001400010CF break
                                                                short loc_1400010E3 ; break
        .text:00000001400010D1
                                                                rcx, aEnd
        .text:00000001400010D8
                                                       call
                                                                printf
        .text:00000001400010DD
                                                       inc
                                                                 ebx
        .text:00000001400010DF
        .text:00000001400010DF loc_1400010DF:
                                                                                   ; CODE XREF: main+1E↑j
        .text:00000001400010DF
        .text:00000001400010E1 do while
                                                                short loc_1400010B0
        .text:00000001400010E3
        .text:00000001400010E3 loc 1400010E3:
                                                                                    ; CODE XREF: main+2F1j
                                                                rbx, [rsp+28h+arg_0]
        .text:00000001400010E3
                                                       mov
        .text:00000001400010E8
                                                       xor
                                                                eax, eax
       .text:00000001400010EA
                                                                rsp, 20h
```

## while

Release版编译器会将其优化成 if + do while。

```
-- .text:00000001400010B0
                                             short loc_1400010C6 ; if
   text:00000001400010B2
   .text:00000001400010B2 loc_1400010B2:
                                                            ; CODE XREF: main+24↓j
  .text:00000001400010B2
                                      mov
                                              edx, ebx
  .text:00000001400010B4
                                              rcx, Format
                                                            : "%d\n"
                                      lea
  .text:00000001400010BB
                                      call
                                              printf
  .text:00000001400010C0
                                      inc
                                              ebx
                                              ebx, edi
  .text:00000001400010C2
                                      cmp
:: .text:00000001400010C4
                                              short loc_1400010B2; do-while
                                      ile
  .text:00000001400010C6
  ; CODE XREF: main+10↑j
  .text:00000001400010C6
                                      mov
                                             rbx, [rsp+28h+arg_0]
  .text:00000001400010CB
                                      xor
                                              eax, eax
```

#### for

Release版编译器会将其优化成 if + do while。

```
.text:00000001400010B0
                                                   jle
                                                            short loc_1400010C6 ; if
    .text:00000001400010B2
                                                                               ; CODE XREF: main+24↓j
    . text:000000140001082
. text:000000140001082
. text:000000140001084
. text:0000000140001088
. text:00000001400010C0
. text:00000001400010C2
. text:00000001400010C4
                                                            edx, ebx
rcx, Format
printf
                                                   mov
                                                                             ; "%d\n"
                                                   lea
                                                   call
                                                   inc
                                                            ebx
                                                             ebx, edi
                                                   cmp
                                                   jl
                                                             short loc_1400010B2; do-while
    .text:00000001400010C6
                                                                                ; CODE XREF: main+10↑j
    .text:00000001400010C6 loc_1400010C6:
                                                            rbx, [rsp+28h+arg_0]
.text:00000001400010C6
 .text:00000001400010CB
                                                   xor
                                                            eax, eax
```