

分支结构

if-else

条件跳转还原为反条件

识别

1. 单分支 if:

- 条件跳转 `label1`，且 `label1` 标号处上方没有跳转

```
00181040 55          push     ebp
00181041 8B EC       mov     ebp,esp
00181043 83 7D 08 64  cmp     dword ptr [argc],64h
00181047 76 0D       jbe     main+16h (0181056h)
00181049 68 F8 20 18 00 push    offset string "argc > 100" (01820F8h)
0018104E E8 BD FF FF FF call    printf (0181010h)
00181053 83 C4 04     add     esp,4
00181056 33 C0       xor     eax,eax
00181058 5D         pop     ebp
00181059 C3         ret
```

2. 双分支 if-else

- 条件跳转 `label1`，且 `label1` 标号处上方有跳转 `label2`
- 条件跳转与标号 `label1` 处为 if 块，`label1` 与 `label2` 为 else 块

```
00C21838 83 7D 08 64  cmp     dword ptr [argc],64h
00C2183C 76 0F       jbe     main+3Dh (0C2184Dh)
00C2183E 68 30 7B C2 00 push    offset string "argc > 100" (0C27B30h)
00C21843 E8 03 F8 FF FF call    _printf (0C2104Bh)      if
00C21848 83 C4 04     add     esp,4
00C2184B EB 0D       jmp     main+4Ah (0C2185Ah)
00C2184D 68 40 7B C2 00 push    offset string "argc <= 100" (0C27B40h)
00C21852 E8 F4 F7 FF FF call    _printf (0C2104Bh)
00C21857 83 C4 04     add     esp,4      else
00C2185A 8B F4       mov     esi,esp
```

3. 多分支 if-elseif-else

- 同双分支，但是有多处跳转到同一目标的语句，且跳转地址没有骑跨

```
0052183C 75 0F       jne     main+3Dh (052184Dh)
0052183E 68 30 7B 52 00 push    offset string "argc == 100" (0527B30h)
00521843 E8 03 F8 FF FF call    _printf (052104Bh)
00521848 83 C4 04     add     esp,4
0052184B EB 3D       jmp     main+7Ah (052188Ah)
0052184D 81 7D 08 C8 00 00 00 00 cmp     dword ptr [argc],0C8h
00521854 75 0F       jne     main+55h (0521865h)
00521856 68 40 7B 52 00 push    offset string "argc == 200" (0527B40h)
0052185B E8 EB F7 FF FF call    _printf (052104Bh)
00521860 83 C4 04     add     esp,4
00521863 EB 25       jmp     main+7Ah (052188Ah)
00521865 81 7D 08 2C 01 00 00 00 cmp     dword ptr [argc],12Ch
0052186C 75 0F       jne     main+6Dh (052187Dh)
0052186E 68 50 7B 52 00 push    offset string "argc == 300" (0527B50h)
00521873 E8 D3 F7 FF FF call    _printf (052104Bh)
00521878 83 C4 04     add     esp,4
0052187B EB 0D       jmp     main+7Ah (052188Ah)
0052187D 68 60 7B 52 00 push    offset string "final" (0527B60h)
00521882 E8 C4 F7 FF FF call    _printf (052104Bh)
00521887 83 C4 04     add     esp,4
0052188A 8B F4       mov     esi,esp
0052188C 68 68 7B 52 00 push    offset string "pause" (0527B68h)
```

4. 嵌套 if-else

- 特征综合前三种

```

009241E8 83 7D 08 64      cmp     dword ptr [argc],64h
009241EC 76 23              jbe     main+51h (0924211h)
009241EE 81 7D 08 C8 00 00 00 cmp     dword ptr [argc],0C8h
009241F5 76 09              jbe     main+40h (0924200h)
009241F7 6B 45 08 03      imul    eax,dword ptr [argc],3
009241FB 89 45 08          mov     dword ptr [argc],eax
009241FE 8B 0F          jmp     main+4Fh (092420Fh)
00924200 8B 45 08          mov     eax,dword ptr [argc]
00924203 33 D2          xor     edx,edx
00924205 B9 03 00 00 00    mov     ecx,3
0092420A F7 F1          div     eax,ecx
0092420C 89 45 08          mov     dword ptr [argc],eax
0092420F EB 1E          jmp     main+6Fh (092422Fh)
00924211 83 7D 08 32      cmp     dword ptr [argc],32h
00924215 76 09              jbe     main+60h (0924220h)
00924217 C7 45 08 32 00 00 00 mov     dword ptr [argc],32h
0092421E EB 0F          jmp     main+6Fh (092422Fh)
00924220 8B 45 08          mov     eax,dword ptr [argc]
00924223 33 D2          xor     edx,edx
00924225 B9 03 00 00 00    mov     ecx,3
0092422A F7 F1          div     eax,ecx
0092422C 89 55 08          mov     dword ptr [argc],edx
0092422F 8B 45 08          mov     eax,dword ptr [argc]
00924232 50          push    eax
00924233 68 30 7B 92 00    push    offset string "%d\n" (0927B30h)
00924238 E8 38 D1 FF FF    call    _printf (0921375h)
0092423D 83 C4 08          add     esp,8

```

优化方案

1. 代码外提
2. 流程归并

switch - case

识别

1. 分支数小于4时，编译器可能会模拟 if-else 结构
 - 与 if-else 的区别：所有的判断跳转都在一起，中间没有实质性代码

```

00FC10A2 8B 4D F8          mov     ecx,dword ptr [n]
00FC10A5 83 C4 08          add     esp,8
00FC10A8 8B C1          mov     eax,ecx
00FC10AA 83 E8 01      case 1:  sub     eax,1
00FC10AD 74 13          je      main+42h (0FC10C2h)
00FC10AF 83 E8 01      case 2:  sub     eax,1
00FC10B2 74 0A          je      main+3Eh (0FC10BEh)
00FC10B4 83 E8 62      case 100: sub     eax,62h
00FC10B7 75 0A          jne     main+43h (0FC10C3h)
00FC10B9 6B F6 64      imul    esi,esi,64h
00FC10BC EB 05          jmp     main+43h (0FC10C3h)
00FC10BE D1 EE          shr     esi,1
00FC10C0 EB 01          jmp     main+43h (0FC10C3h)
00FC10C2 46          inc     esi
00FC10C3 51          push    ecx
00FC10C4 56          push    esi
00FC10C5 68 0C 21 FC 00    push    offset string "%d, %u" (0FC210Ch)
00FC10CA E8 51 FF FF FF    call    printf (0FC1020h)
00FC10CF 68 14 21 FC 00    push    offset string "pause" (0FC2114h)
00FC10D4 FF 15 60 20 FC 00 call    dword ptr [__imp__system (0FC2060h)]

```

2. 分支数大于3，并且 case 的判定值存在明显的线性关系
 - 跳转表，存储跳转的地址，每个地址4个字节


```

0x003E1134 00 01 02 06 03 04 06 06 06 06 06 06 06 06 06 .....
0x003E1144 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E1154 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E1164 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E1174 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E1184 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E1194 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E11A4 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E11B4 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E11C4 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E11D4 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E11E4 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E11F4 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E1204 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E1214 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
0x003E1224 06 06 06 06 06 06 06 06 06 06 06 06 05 3b .....
0x003E1234 0d 04 30 3e 00 f2 75 02 f2 c3 f2 e9 79 02 00 00 ..0>..?u.???y...
0x003E1244 56 6a 01 e8 37 0b 00 00 e8 81 06 00 00 50 e8 62 Vj.??...??...P?b

```

- case 地址表

```

0x003E1118 003e10ba 003e10c1 003e10c8 003e10cf 003e10d6 ?.>.?.>.?.>.?.>.?.>.
0x003E112C 003e10dd 003e10ea 06020100 06060403 06060606 ?.>.?.>.....
0x003E1140 06060606 06060606 06060606 06060606 06060606 .....
0x003E1154 06060606 06060606 06060606 06060606 06060606 .....
0x003E1168 06060606 06060606 06060606 06060606 06060606 .....
0x003E117C 06060606 06060606 06060606 06060606 06060606 .....
0x003E1190 06060606 06060606 06060606 06060606 06060606 .....
0x003E11A4 06060606 06060606 06060606 06060606 06060606 .....
0x003E11B8 06060606 06060606 06060606 06060606 06060606 .....
0x003E11CC 06060606 06060606 06060606 06060606 06060606 .....
0x003E11E0 06060606 06060606 06060606 06060606 06060606 .....
0x003E11F4 06060606 06060606 06060606 06060606 06060606 .....
0x003E1208 06060606 06060606 06060606 06060606 06060606 .....
0x003E121C 06060606 06060606 06060606 06060606 06060606 .....
0x003E1230 3b050606 3e30040d 0275f200 e9f2c3f2 00000279 ...;.0>..?u.???y...
0x003E1244 68016256 00000b37 000681e8 62025000 0800000b Vj.??...??...P?b

```

- 通过 case 索引表拿到索引值 index，利用 index 取得 case 地址表中的地址进行跳转

```

003E109E 8B 45 F8      mov     eax,dword ptr [n]
003E10A1 83 C4 08      add     esp,8
003E10A4 48           dec     eax
003E10A5 3D FE 00 00 00  cmp     eax,0FEh
003E10AA 77 3E       ja      $LN9+0Dh (03E10EAh)
003E10AC 0F B6 80 34 11 3E 00 movzx   eax,byte ptr [eax+3E1134h]
003E10B3 FF 24 85 18 11 3E 00 jmp     dword ptr [eax*4+3E1118h]
003E10BA 68 0C 21 3E 00      push    offset string "n = 1\n" (03E210Ch)
003E10BF EB 21       jmp     $LN9+5h (03E10E2h)
$LN4:
003E10C1 68 14 21 3E 00      push    offset string "n = 2\n" (03E2114h)
003E10C6 EB 1A       jmp     $LN9+5h (03E10E2h)
$LN6:
003E10C8 68 1C 21 3E 00      push    offset string "n = 3\n" (03E211Ch)
003E10CD EB 13       jmp     $LN9+5h (03E10E2h)
$LN7:
003E10CF 68 24 21 3E 00      push    offset string "n = 5\n" (03E2124h)
003E10D4 EB 0C       jmp     $LN9+5h (03E10E2h)
$LN8:
003E10D6 68 2C 21 3E 00      push    offset string "n = 6\n" (03E212Ch)
003E10DB EB 05       jmp     $LN9+5h (03E10E2h)
$LN9:
003E10DD 68 34 21 3E 00      push    offset string "n = 255\n" (03E2134h)
003E10E2 E8 39 FF FF FF      call    printf (03E1020h)
003E10E7 83 C4 04      add     esp,4
003E10EA FF 75 F8      push    dword ptr [n]
003E10ED FF 75 08      push    dword ptr [argc]
003E10F0 68 40 21 3E 00      push    offset string "%d, %u" (03E2140h)
003E10F5 E8 26 FF FF FF      call    printf (03E1020h)
003E10FA 68 48 21 3E 00      push    offset string "pause" (03E2148h)
003E10FF FF 15 60 20 3E 00      call    dword ptr [__imp__system (03E2060h)]
003E1105 8B 4D FC      mov     ecx,dword ptr [ebp-4]
003E1108 83 C4 10      add     esp,10h

```

case索引表

case地址表

4. 当 case 间隔较大，且 最大case - 最小case > 255 即超过1字节，则会生成判定树

- 判定树：将每个 case 值作为一个节点，从这些节点中找到一个中间值作为跟节点，以此形成一个平衡二叉树
- jg和jz 或 ja和je 的组合就是此判定树的形式

- 直接找 `jz` 和 `jnz` 或 `je` 和 `jne` 即可找到节点
- 此模式下, 可能会存在跟前面几种模式的混合应用

```

009E109E 8B 45 F8      mov     eax,dword ptr [n]
009E10A1 83 C4 08      add     esp,8
009E10A4 83 F8 0A      cmp     eax,0Ah           =? 10
009E10A7 77 2D      ja      main+56h (09E10D6h)
009E10A9 74 24      je      main+4Fh (09E10CFh)
009E10AB 83 E8 02      sub     eax,2
009E10AE 74 18      je      main+48h (09E10C8h)   =? 2
009E10B0 83 E8 01      sub     eax,1
009E10B3 74 0C      je      main+41h (09E10C1h)   =? 3
009E10B5 83 E8 05      sub     eax,5
009E10B8 75 4B      jne     main+85h (09E1105h)   =? 8
009E10BA 68 1C 21 9E 00  push   offset string "n = 8\n" (09E211Ch)
009E10BF EB 3C      jmp     main+7Dh (09E10FDh)
009E10C1 68 14 21 9E 00  push   offset string "n = 3\n" (09E2114h)
009E10C6 EB 35      jmp     main+7Dh (09E10FDh)
009E10C8 68 0C 21 9E 00  push   offset string "n = 2\n" (09E210Ch)
009E10CD EB 2E      jmp     main+7Dh (09E10FDh)
009E10CF 68 24 21 9E 00  push   offset string "n = 10\n" (09E2124h)
009E10D4 EB 27      jmp     main+7Dh (09E10FDh)
009E10D6 8B 45 F8      mov     eax,dword ptr [n]
009E10D9 83 E8 23      sub     eax,23h           =? 35
009E10DC 74 1A      je      main+78h (09E10F8h)
009E10DE 83 E8 02      sub     eax,2
009E10E1 74 0E      je      main+71h (09E10F1h)   =? 37
009E10E3 2D 75 02 00 00  sub     eax,275h
009E10E8 75 1B      jne     main+85h (09E1105h)   =? 666
009E10EA 68 3C 21 9E 00  push   offset string "n = 666\n" (09E213Ch)
009E10EF EB 0C      jmp     main+7Dh (09E10FDh)
009E10F1 68 34 21 9E 00  push   offset string "n = 37\n" (09E2134h)
009E10F6 EB 05      jmp     main+7Dh (09E10FDh)
009E10F8 68 2C 21 9E 00  push   offset string "n = 35\n" (09E212Ch)
009E10FD E8 1E FF FF FF  call    printf (09E1020h)
009E1102 83 C4 04      add     esp,4
009E1105 FF 75 F8      push    dword ptr [n]
009E1108 FF 75 08      push    dword ptr [argc]

```

优化方案

1. 代码外提
2. 流程归并