

# Compiler Optimization Notes

Mac Radigan

## Compiler Optimization

### Variable Declaration

declared in base scope

```
1 // variable_stack_1.cc
2
3 #include <stdio.h>
4
5 int main(void) {
6     int value;
7     for(int k=0; k<10; ++k)
8     {
9         value = k;
10        fprintf(stdout, "value: %d\n", value);
11    }
12    return 0;
13 }
14
15 // *EOF*
```

  

```
1
2 ./bin/variable_stack_1:      file format elf64-x86-64
3
4
5 Disassembly of section .init:
6
7 00000000004003b8 <_init>:
8   4003b8: 48 83 ec 08          sub    $0x8,%rsp
9   4003bc: 48 8b 05 1d 05 20 00 mov     0x20051d(%rip),%rax      # 6008e0 <_DYNAMIC+0x1d0>
10  4003c3: 48 85 c0             test   %rax,%rax
11  4003c6: 74 05              je     4003cd <_init+0x15>
12  4003c8: e8 43 00 00 00     callq 400410 <fprintf@plt+0x10>
13  4003cd: 48 83 c4 08          add    $0x8,%rsp
14  4003d1: c3                 retq
```

```

15
16 Disassembly of section .plt:
17
18 0000000004003e0 <__libc_start_main@plt-0x10>:
19 4003e0: ff 35 0a 05 20 00    pushq 0x20050a(%rip)      # 6008f0 <_GLOBAL_OFFSET_TABLE_+0x8>
20 4003e6: ff 25 0c 05 20 00    jmpq  *0x20050c(%rip)     # 6008f8 <_GLOBAL_OFFSET_TABLE_+0x10>
21 4003ec: 0f 1f 40 00          nopl 0x0(%rax)
22
23 0000000004003f0 <__libc_start_main@plt>:
24 4003f0: ff 25 0a 05 20 00    jmpq  *0x20050a(%rip)     # 600900 <_GLOBAL_OFFSET_TABLE_+0x18>
25 4003f6: 68 00 00 00 00      pushq $0x0
26 4003fb: e9 e0 ff ff ff      jmpq  4003e0 <_init+0x28>
27
28 000000000400400 <fprintf@plt>:
29 400400: ff 25 02 05 20 00    jmpq  *0x200502(%rip)     # 600908 <_GLOBAL_OFFSET_TABLE_+0x20>
30 400406: 68 01 00 00 00      pushq $0x1
31 40040b: e9 d0 ff ff ff      jmpq  4003e0 <_init+0x28>
32
33 Disassembly of section .plt.got:
34
35 000000000400410 <.plt.got>:
36 400410: ff 25 ca 04 20 00    jmpq  *0x2004ca(%rip)     # 6008e0 <_DYNAMIC+0x1d0>
37 400416: 66 90              xchg  %ax,%ax
38
39 Disassembly of section .text:
40
41 000000000400420 <_start>:
42 400420: 31 ed              xor    %ebp,%ebp
43 400422: 49 89 d1           mov    %rdx,%r9
44 400425: 5e                pop    %rsi
45 400426: 48 89 e2           mov    %rsp,%rdx
46 400429: 48 83 e4 f0        and    $0xfffffffffffff0,%rsp
47 40042d: 50                push   %rax
48 40042e: 54                push   %rsp
49 40042f: 49 c7 c0 b0 05 40 00 mov    $0x4005b0,%r8
50 400436: 48 c7 c1 40 05 40 00 mov    $0x400540,%rcx
51 40043d: 48 c7 c7 f7 04 40 00 mov    $0x4004f7,%rdi
52 400444: e8 a7 ff ff ff     callq 4003f0 <__libc_start_main@plt>
53 400449: f4                hlt
54 40044a: 66 0f 1f 44 00 00    nopw  0x0(%rax,%rax,1)
55
56 000000000400450 <deregister_tm_clones>:
57 400450: 55                push   %rbp
58 400451: b8 20 09 60 00     mov    $0x600920,%eax
59 400456: 48 3d 20 09 60 00   cmp    $0x600920,%rax
60 40045c: 48 89 e5           mov    %rsp,%rbp
61 40045f: 74 17             je     400478 <deregister_tm_clones+0x28>
62 400461: b8 00 00 00 00     mov    $0x0,%eax
63 400466: 48 85 c0           test   %rax,%rax
64 400469: 74 0d             je     400478 <deregister_tm_clones+0x28>
65 40046b: 5d                pop    %rbp
66 40046c: bf 20 09 60 00     mov    $0x600920,%edi
67 400471: ff e0             jmpq   *%rax
68 400473: 0f 1f 44 00 00     nopl   0x0(%rax,%rax,1)
69 400478: 5d                pop    %rbp
70 400479: c3                retq
71 40047a: 66 0f 1f 44 00 00    nopw  0x0(%rax,%rax,1)

```

```

72
73 0000000000400480 <register_tm_clones>:
74   400480:  be 20 09 60 00      mov     $0x600920,%esi
75   400485:  55                  push    %rbp
76   400486:  48 81 ee 20 09 60 00 sub     $0x600920,%rsi
77   40048d:  48 89 e5            mov     %rsp,%rbp
78   400490:  48 c1 fe 03         sar     $0x3,%rsi
79   400494:  48 89 f0            mov     %rsi,%rax
80   400497:  48 c1 e8 3f         shr     $0x3f,%rax
81   40049b:  48 01 c6            add     %rax,%rsi
82   40049e:  48 d1 fe            sar     %rsi
83   4004a1:  74 15              je      4004b8 <register_tm_clones+0x38>
84   4004a3:  b8 00 00 00 00      mov     $0x0,%eax
85   4004a8:  48 85 c0            test    %rax,%rax
86   4004ab:  74 0b              je      4004b8 <register_tm_clones+0x38>
87   4004ad:  5d                  pop     %rbp
88   4004ae:  bf 20 09 60 00      mov     $0x600920,%edi
89   4004b3:  ff e0              jmpq    *%rax
90   4004b5:  0f 1f 00            nopl    (%rax)
91   4004b8:  5d                  pop     %rbp
92   4004b9:  c3                  retq
93   4004ba:  66 0f 1f 44 00 00   nopw    0x0(%rax,%rax,1)
94
95 00000000004004c0 <__do_global_dtors_aux>:
96   4004c0:  80 3d 61 04 20 00 00 cmpb     $0x0,0x200461(%rip)      # 600928 <completed.6981>
97   4004c7:  75 17              jne     4004e0 <__do_global_dtors_aux+0x20>
98   4004c9:  55                  push    %rbp
99   4004ca:  48 89 e5            mov     %rsp,%rbp
100  4004cd:  e8 7e ff ff ff      callq   400450 <deregister_tm_clones>
101  4004d2:  c6 05 4f 04 20 00 01 movb     $0x1,0x20044f(%rip)      # 600928 <completed.6981>
102  4004d9:  5d                  pop     %rbp
103  4004da:  c3                  retq
104  4004db:  0f 1f 44 00 00      nopl    0x0(%rax,%rax,1)
105  4004e0:  f3 c3              repz    retq
106  4004e2:  0f 1f 40 00         nopl    0x0(%rax)
107  4004e6:  66 2e 0f 1f 84 00 00 nopw     %cs:0x0(%rax,%rax,1)
108  4004ed:  00 00 00
109
110 00000000004004f0 <frame_dummy>:
111  4004f0:  55                  push    %rbp
112  4004f1:  48 89 e5            mov     %rsp,%rbp
113  4004f4:  5d                  pop     %rbp
114  4004f5:  eb 89              jmp     400480 <register_tm_clones>
115
116 00000000004004f7 <main>:
117  4004f7:  55                  push    %rbp
118  4004f8:  48 89 e5            mov     %rsp,%rbp
119  4004fb:  48 83 ec 10         sub     $0x10,%rsp
120  4004ff:  c7 45 fc 00 00 00 00 movl     $0x0,-0x4(%rbp)
121  400506:  83 7d fc 09         cmpl    $0x9,-0x4(%rbp)
122  40050a:  7f 28              jg      400534 <main+0x3d>
123  40050c:  8b 45 fc            mov     -0x4(%rbp),%eax
124  40050f:  89 45 f8            mov     %eax,-0x8(%rbp)
125  400512:  48 8b 05 07 04 20 00 mov     0x200407(%rip),%rax      # 600920 <__TMC_END__>
126  400519:  8b 55 f8            mov     -0x8(%rbp),%edx
127  40051c:  be c4 05 40 00      mov     $0x4005c4,%esi
128  400521:  48 89 c7            mov     %rax,%rdi

```

```

129 400524: b8 00 00 00 00      mov     $0x0,%eax
130 400529: e8 d2 fe ff ff      callq   400400 <fprintf@plt>
131 40052e: 83 45 fc 01          addl    $0x1,-0x4(%rbp)
132 400532: eb d2               jmp     400506 <main+0xf>
133 400534: b8 00 00 00 00      mov     $0x0,%eax
134 400539: c9                 leaveq  %eax
135 40053a: c3                 retq
136 40053b: 0f 1f 44 00 00      nopl    0x0(%rax,%rax,1)
137
138 0000000000400540 <__libc_csu_init>:
139 400540: 41 57              push    %r15
140 400542: 41 56              push    %r14
141 400544: 41 89 ff           mov     %edi,%r15d
142 400547: 41 55              push    %r13
143 400549: 41 54              push    %r12
144 40054b: 4c 8d 25 ae 01 20 00 lea     0x2001ae(%rip),%r12      # 600700 <__frame_dummy_init_array_entry>
145 400552: 55                push    %rbp
146 400553: 48 8d 2d ae 01 20 00 lea     0x2001ae(%rip),%rbp      # 600708 <__init_array_end>
147 40055a: 53                push    %rbx
148 40055b: 49 89 f6           mov     %rsi,%r14
149 40055e: 49 89 d5           mov     %rdx,%r13
150 400561: 4c 29 e5           sub     %r12,%rbp
151 400564: 48 83 ec 08        sub     $0x8,%rsp
152 400568: 48 c1 fd 03        sar     $0x3,%rbp
153 40056c: e8 47 fe ff ff      callq   4003b8 <_init>
154 400571: 48 85 ed           test    %rbp,%rbp
155 400574: 74 20              je      400596 <__libc_csu_init+0x56>
156 400576: 31 db              xor     %ebx,%ebx
157 400578: 0f 1f 84 00 00 00 00 nopl    0x0(%rax,%rax,1)
158 40057f: 00
159 400580: 4c 89 ea           mov     %r13,%rdx
160 400583: 4c 89 f6           mov     %r14,%rsi
161 400586: 44 89 ff           mov     %r15d,%edi
162 400589: 41 ff 14 dc        callq   *(%r12,%rbx,8)
163 40058d: 48 83 c3 01        add     $0x1,%rbx
164 400591: 48 39 eb           cmp     %rbp,%rbx
165 400594: 75 ea              jne     400580 <__libc_csu_init+0x40>
166 400596: 48 83 c4 08        add     $0x8,%rsp
167 40059a: 5b                pop     %rbx
168 40059b: 5d                pop     %rbp
169 40059c: 41 5c              pop     %r12
170 40059e: 41 5d              pop     %r13
171 4005a0: 41 5e              pop     %r14
172 4005a2: 41 5f              pop     %r15
173 4005a4: c3                 retq
174 4005a5: 90                 nop
175 4005a6: 66 2e 0f 1f 84 00 00 nopw    %cs:0x0(%rax,%rax,1)
176 4005ad: 00 00 00
177
178 00000000004005b0 <__libc_csu_fini>:
179 4005b0: f3 c3              repz retq
180
181 Disassembly of section .fini:
182
183 00000000004005b4 <_fini>:
184 4005b4: 48 83 ec 08        sub     $0x8,%rsp
185 4005b8: 48 83 c4 08        add     $0x8,%rsp

```

```
186 4005bc: c3          retq
```

declared within statement block

```
1 // variable_stack_1.cc
2
3 #include <stdio.h>
4
5 int main(void) {
6     for(int k=0; k<10; ++k)
7     {
8         int value = k;
9         fprintf(stdout, "value: %d\n", value);
10    }
11    return 0;
12 }
13
14 // *EOF*
```

```
1
2 ./bin/variable_stack_2:      file format elf64-x86-64
3
4
5 Disassembly of section .init:
6
7 00000000004003b8 <_init>:
8 4003b8: 48 83 ec 08          sub    $0x8,%rsp
9 4003bc: 48 8b 05 1d 05 20 00 mov     0x20051d(%rip),%rax      # 6008e0 <_DYNAMIC+0x1d0>
10 4003c3: 48 85 c0             test   %rax,%rax
11 4003c6: 74 05              je     4003cd <_init+0x15>
12 4003c8: e8 43 00 00 00      callq 400410 <fprintf@plt+0x10>
13 4003cd: 48 83 c4 08          add    $0x8,%rsp
14 4003d1: c3                retq
15
16 Disassembly of section .plt:
17
18 00000000004003e0 <__libc_start_main@plt-0x10>:
19 4003e0: ff 35 0a 05 20 00    pushq 0x20050a(%rip)          # 6008f0 <_GLOBAL_OFFSET_TABLE_+0x8>
20 4003e6: ff 25 0c 05 20 00    jmpq  *0x20050c(%rip)        # 6008f8 <_GLOBAL_OFFSET_TABLE_+0x10>
21 4003ec: 0f 1f 40 00          nopl  0x0(%rax)
22
23 00000000004003f0 <__libc_start_main@plt>:
24 4003f0: ff 25 0a 05 20 00    jmpq  *0x20050a(%rip)          # 600900 <_GLOBAL_OFFSET_TABLE_+0x18>
25 4003f6: 68 00 00 00 00      pushq $0x0
26 4003fb: e9 e0 ff ff ff      jmpq  4003e0 <_init+0x28>
27
28 0000000000400400 <fprintf@plt>:
29 400400: ff 25 02 05 20 00    jmpq  *0x200502(%rip)          # 600908 <_GLOBAL_OFFSET_TABLE_+0x20>
30 400406: 68 01 00 00 00      pushq $0x1
31 40040b: e9 d0 ff ff ff      jmpq  4003e0 <_init+0x28>
```

```

32
33 Disassembly of section .plt.got:
34
35 000000000400410 <.plt.got>:
36 400410: ff 25 ca 04 20 00      jmpq    *0x2004ca(%rip)      # 6008e0 <_DYNAMIC+0x1d0>
37 400416: 66 90                  xchg    %ax,%ax
38
39 Disassembly of section .text:
40
41 000000000400420 <_start>:
42 400420: 31 ed                  xor     %ebp,%ebp
43 400422: 49 89 d1               mov     %rdx,%r9
44 400425: 5e                     pop     %rsi
45 400426: 48 89 e2               mov     %rsp,%rdx
46 400429: 48 83 e4 f0            and     $0xfffffffffffff0,%rsp
47 40042d: 50                     push    %rax
48 40042e: 54                     push    %rsp
49 40042f: 49 c7 c0 b0 05 40 00   mov     $0x4005b0,%r8
50 400436: 48 c7 c1 40 05 40 00   mov     $0x400540,%rcx
51 40043d: 48 c7 c7 f7 04 40 00   mov     $0x4004f7,%rdi
52 400444: e8 a7 ff ff ff        callq   4003f0 <__libc_start_main@plt>
53 400449: f4                     hlt
54 40044a: 66 0f 1f 44 00 00     nopw    0x0(%rax,%rax,1)
55
56 000000000400450 <deregister_tm_clones>:
57 400450: 55                     push    %rbp
58 400451: b8 20 09 60 00        mov     $0x600920,%eax
59 400456: 48 3d 20 09 60 00     cmp     $0x600920,%rax
60 40045c: 48 89 e5               mov     %rsp,%rbp
61 40045f: 74 17                  je      400478 <deregister_tm_clones+0x28>
62 400461: b8 00 00 00 00        mov     $0x0,%eax
63 400466: 48 85 c0               test    %rax,%rax
64 400469: 74 0d                  je      400478 <deregister_tm_clones+0x28>
65 40046b: 5d                     pop     %rbp
66 40046c: bf 20 09 60 00        mov     $0x600920,%edi
67 400471: ff e0                  jmpq    *%rax
68 400473: 0f 1f 44 00 00        nopl    0x0(%rax,%rax,1)
69 400478: 5d                     pop     %rbp
70 400479: c3                     retq
71 40047a: 66 0f 1f 44 00 00     nopw    0x0(%rax,%rax,1)
72
73 000000000400480 <register_tm_clones>:
74 400480: be 20 09 60 00        mov     $0x600920,%esi
75 400485: 55                     push    %rbp
76 400486: 48 81 ee 20 09 60 00   sub     $0x600920,%rsi
77 40048d: 48 89 e5               mov     %rsp,%rbp
78 400490: 48 c1 fe 03            sar     $0x3,%rsi
79 400494: 48 89 f0               mov     %rsi,%rax
80 400497: 48 c1 e8 3f            shr     $0x3f,%rax
81 40049b: 48 01 c6               add     %rax,%rsi
82 40049e: 48 d1 fe               sar     %rsi
83 4004a1: 74 15                  je      4004b8 <register_tm_clones+0x38>
84 4004a3: b8 00 00 00 00        mov     $0x0,%eax
85 4004a8: 48 85 c0               test    %rax,%rax
86 4004ab: 74 0b                  je      4004b8 <register_tm_clones+0x38>
87 4004ad: 5d                     pop     %rbp
88 4004ae: bf 20 09 60 00        mov     $0x600920,%edi

```

```

89 4004b3: ff e0          jmpq    *%rax
90 4004b5: 0f 1f 00      nopl   (%rax)
91 4004b8: 5d           pop    %rbp
92 4004b9: c3          retq
93 4004ba: 66 0f 1f 44 00 00 nopw   0x0(%rax,%rax,1)
94
95 00000000004004c0 <__do_global_dtors_aux>:
96 4004c0: 80 3d 61 04 20 00 00 cmpb   $0x0,0x200461(%rip)      # 600928 <completed.6981>
97 4004c7: 75 17      jne    4004e0 <__do_global_dtors_aux+0x20>
98 4004c9: 55      push   %rbp
99 4004ca: 48 89 e5    mov    %rsp,%rbp
100 4004cd: e8 7e ff ff callq  400450 <deregister_tm_clones>
101 4004d2: c6 05 4f 04 20 00 01 movb   $0x1,0x20044f(%rip)      # 600928 <completed.6981>
102 4004d9: 5d      pop    %rbp
103 4004da: c3      retq
104 4004db: 0f 1f 44 00 00 nopl   0x0(%rax,%rax,1)
105 4004e0: f3 c3      repz retq
106 4004e2: 0f 1f 40 00 nopl   0x0(%rax)
107 4004e6: 66 2e 0f 1f 84 00 00 nopw   %cs:0x0(%rax,%rax,1)
108 4004ed: 00 00 00
109
110 00000000004004f0 <frame_dummy>:
111 4004f0: 55      push   %rbp
112 4004f1: 48 89 e5    mov    %rsp,%rbp
113 4004f4: 5d      pop    %rbp
114 4004f5: eb 89      jmp    400480 <register_tm_clones>
115
116 00000000004004f7 <main>:
117 4004f7: 55      push   %rbp
118 4004f8: 48 89 e5    mov    %rsp,%rbp
119 4004fb: 48 83 ec 10 sub    $0x10,%rsp
120 4004ff: c7 45 fc 00 00 00 00 movl   $0x0,-0x4(%rbp)
121 400506: 83 7d fc 09 cmpl   $0x9,-0x4(%rbp)
122 40050a: 7f 28      jg     400534 <main+0x3d>
123 40050c: 8b 45 fc    mov    -0x4(%rbp),%eax
124 40050f: 89 45 f8    mov    %eax,-0x8(%rbp)
125 400512: 48 8b 05 07 04 20 00 mov    0x200407(%rip),%rax      # 600920 <__TMC_END__>
126 400519: 8b 55 f8    mov    -0x8(%rbp),%edx
127 40051c: be c4 05 40 00 mov    $0x4005c4,%esi
128 400521: 48 89 c7    mov    %rax,%rdi
129 400524: b8 00 00 00 00 mov    $0x0,%eax
130 400529: e8 d2 fe ff callq  400400 <fprintf@plt>
131 40052e: 83 45 fc 01 addl   $0x1,-0x4(%rbp)
132 400532: eb d2      jmp    400506 <main+0xf>
133 400534: b8 00 00 00 00 mov    $0x0,%eax
134 400539: c9      leaveq %eax
135 40053a: c3      retq
136 40053b: 0f 1f 44 00 00 nopl   0x0(%rax,%rax,1)
137
138 0000000000400540 <__libc_csu_init>:
139 400540: 41 57      push   %r15
140 400542: 41 56      push   %r14
141 400544: 41 89 ff    mov    %edi,%r15d
142 400547: 41 55      push   %r13
143 400549: 41 54      push   %r12
144 40054b: 4c 8d 25 ae 01 20 00 lea     0x2001ae(%rip),%r12      # 600700 <__frame_dummy_init_array_entry>
145 400552: 55      push   %rbp

```

```

146 400553: 48 8d 2d ae 01 20 00 lea 0x2001ae(%rip),%rbp # 600708 <__init_array_end>
147 40055a: 53 push %rbx
148 40055b: 49 89 f6 mov %rsi,%r14
149 40055e: 49 89 d5 mov %rdx,%r13
150 400561: 4c 29 e5 sub %r12,%rbp
151 400564: 48 83 ec 08 sub $0x8,%rsp
152 400568: 48 c1 fd 03 sar $0x3,%rbp
153 40056c: e8 47 fe ff ff callq 4003b8 <_init>
154 400571: 48 85 ed test %rbp,%rbp
155 400574: 74 20 je 400596 <__libc_csu_init+0x56>
156 400576: 31 db xor %ebx,%ebx
157 400578: 0f 1f 84 00 00 00 00 nopl 0x0(%rax,%rax,1)
158 40057f: 00
159 400580: 4c 89 ea mov %r13,%rdx
160 400583: 4c 89 f6 mov %r14,%rsi
161 400586: 44 89 ff mov %r15d,%edi
162 400589: 41 ff 14 dc callq *(%r12,%rbx,8)
163 40058d: 48 83 c3 01 add $0x1,%rbx
164 400591: 48 39 eb cmp %rbp,%rbx
165 400594: 75 ea jne 400580 <__libc_csu_init+0x40>
166 400596: 48 83 c4 08 add $0x8,%rsp
167 40059a: 5b pop %rbx
168 40059b: 5d pop %rbp
169 40059c: 41 5c pop %r12
170 40059e: 41 5d pop %r13
171 4005a0: 41 5e pop %r14
172 4005a2: 41 5f pop %r15
173 4005a4: c3 retq
174 4005a5: 90 nop
175 4005a6: 66 2e 0f 1f 84 00 00 nopw %cs:0x0(%rax,%rax,1)
176 4005ad: 00 00 00
177
178 00000000004005b0 <__libc_csu_fini>:
179 4005b0: f3 c3 repz retq
180
181 Disassembly of section .fini:
182
183 00000000004005b4 <_fini>:
184 4005b4: 48 83 ec 08 sub $0x8,%rsp
185 4005b8: 48 83 c4 08 add $0x8,%rsp
186 4005bc: c3 retq

```

### implementation difference (side-by-side diff, no suppression)

```

1
2
3 Disassembly of section .init: Disassembly of section .init:
4
5 00000000004003b8 <_init>: 00000000004003b8 <_init>:
6 4003b8: 48 83 ec 08 sub $0x8,%rsp 4003b8: 48 83 ec 08 sub $0x8,%rsp
7 4003bc: 48 8b 05 1d 05 20 00 mov 0x20051d(%rip),%rax 4003bc: 48 8b 05 1d 05 20 00 mov 0x20051d(%rip),%rax
8 4003c3: 48 85 c0 test %rax,%rax 4003c3: 48 85 c0 test %rax,%rax
9 4003c6: 74 05 je 4003cd <_init+0x15> 4003c6: 74 05 je 4003cd <_init+0x15>
10 4003c8: e8 43 00 00 00 callq 400410 <fprintf@plt+0x1 4003c8: e8 43 00 00 00 callq 400410 <fprintf@plt+0x1
11 4003cd: 48 83 c4 08 add $0x8,%rsp 4003cd: 48 83 c4 08 add $0x8,%rsp
12 4003d1: c3 retq 4003d1: c3 retq
13
14 Disassembly of section .plt: Disassembly of section .plt:
15

```



```

16 0000000004003e0 <_libc_start_main@plt-0x10>: 00000000004003e0 <_libc_start_main@plt-0x10>:
17 4003e0: ff 35 0a 05 20 00 pushq 0x20050a(%rip) # 4003e0: ff 35 0a 05 20 00 pushq 0x20050a(%rip) #
18 4003e6: ff 25 0c 05 20 00 jmpq *0x20050c(%rip) 4003e6: ff 25 0c 05 20 00 jmpq *0x20050c(%rip)
19 4003ec: 0f 1f 40 00 nopl 0x0(%rax) 4003ec: 0f 1f 40 00 nopl 0x0(%rax)
20
21 0000000004003f0 <_libc_start_main@plt>: 00000000004003f0 <_libc_start_main@plt>:
22 4003f0: ff 25 0a 05 20 00 jmpq *0x20050a(%rip) 4003f0: ff 25 0a 05 20 00 jmpq *0x20050a(%rip)
23 4003f6: 68 00 00 00 00 pushq $0x0 4003f6: 68 00 00 00 00 pushq $0x0
24 4003fb: e9 e0 ff ff ff jmpq 4003e0 <_init+0x28> 4003fb: e9 e0 ff ff ff jmpq 4003e0 <_init+0x28>
25
26 000000000400400 <printf@plt>: 0000000000400400 <printf@plt>:
27 400400: ff 25 02 05 20 00 jmpq *0x200502(%rip) 400400: ff 25 02 05 20 00 jmpq *0x200502(%rip)
28 400406: 68 01 00 00 00 pushq $0x1 400406: 68 01 00 00 00 pushq $0x1
29 40040b: e9 d0 ff ff ff jmpq 4003e0 <_init+0x28> 40040b: e9 d0 ff ff ff jmpq 4003e0 <_init+0x28>
30
31 Disassembly of section .plt.got: Disassembly of section .plt.got:
32
33 000000000400410 <.plt.got>: 0000000000400410 <.plt.got>:
34 400410: ff 25 ca 04 20 00 jmpq *0x2004ca(%rip) 400410: ff 25 ca 04 20 00 jmpq *0x2004ca(%rip)
35 400416: 66 90 xchg %ax,%ax 400416: 66 90 xchg %ax,%ax
36
37 Disassembly of section .text: Disassembly of section .text:
38
39 000000000400420 <_start>: 0000000000400420 <_start>:
40 400420: 31 ed xor %ebp,%ebp 400420: 31 ed xor %ebp,%ebp
41 400422: 49 89 d1 mov %rdx,%r9 400422: 49 89 d1 mov %rdx,%r9
42 400425: 5e pop %rsi 400425: 5e pop %rsi
43 400426: 48 89 e2 mov %rsp,%rdx 400426: 48 89 e2 mov %rsp,%rdx
44 400429: 48 83 e4 f0 and $0xfffffffffffff0,%rsi 400429: 48 83 e4 f0 and $0xfffffffffffff0,%rsi
45 40042d: 50 push %rax 40042d: 50 push %rax
46 40042e: 54 push %rsp 40042e: 54 push %rsp
47 40042f: 49 c7 c0 b0 05 40 00 mov $0x4005b0,%r8 40042f: 49 c7 c0 b0 05 40 00 mov $0x4005b0,%r8
48 400436: 48 c7 c1 04 05 40 00 mov $0x400540,%rcx 400436: 48 c7 c1 04 05 40 00 mov $0x400540,%rcx
49 40043d: 48 c7 c7 f7 04 40 00 mov $0x4004f7,%rdi 40043d: 48 c7 c7 f7 04 40 00 mov $0x4004f7,%rdi
50 400444: e8 a7 ff ff ff callq 4003f0 <_libc_start_ma 400444: e8 a7 ff ff ff callq 4003f0 <_libc_start_ma
51 400449: f4 hlt 400449: f4 hlt
52 40044a: 66 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1) 40044a: 66 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1)
53
54 000000000400450 <deregister_tm_clones>: 0000000000400450 <deregister_tm_clones>:
55 400450: 55 push %rbp 400450: 55 push %rbp
56 400451: b8 20 09 60 00 mov $0x600920,%eax 400451: b8 20 09 60 00 mov $0x600920,%eax
57 400456: 48 3d 20 09 60 00 cmp $0x600920,%rax 400456: 48 3d 20 09 60 00 cmp $0x600920,%rax
58 40045c: 48 89 e5 mov %rsp,%rbp 40045c: 48 89 e5 mov %rsp,%rbp
59 40045f: 74 17 je 400478 <deregister_tm_c 40045f: 74 17 je 400478 <deregister_tm_c
60 400461: b8 00 00 00 00 mov $0x0,%eax 400461: b8 00 00 00 00 mov $0x0,%eax
61 400466: 48 85 c0 test %rax,%rax 400466: 48 85 c0 test %rax,%rax
62 400469: 74 0d je 400478 <deregister_tm_c 400469: 74 0d je 400478 <deregister_tm_c
63 40046b: 5d pop %rbp 40046b: 5d pop %rbp
64 40046c: bf 20 09 60 00 mov $0x600920,%edi 40046c: bf 20 09 60 00 mov $0x600920,%edi
65 400471: ff e0 jmpq *%rax 400471: ff e0 jmpq *%rax
66 400473: 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1) 400473: 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1)
67 400478: 5d pop %rbp 400478: 5d pop %rbp
68 400479: c3 retq 400479: c3 retq
69 40047a: 66 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1) 40047a: 66 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1)
70
71 000000000400480 <register_tm_clones>: 0000000000400480 <register_tm_clones>:
72 400480: be 20 09 60 00 mov $0x600920,%esi 400480: be 20 09 60 00 mov $0x600920,%esi
73 400485: 55 push %rbp 400485: 55 push %rbp
74 400486: 48 81 ee 20 09 60 00 sub $0x600920,%rsi 400486: 48 81 ee 20 09 60 00 sub $0x600920,%rsi
75 40048d: 48 89 e5 mov %rsp,%rbp 40048d: 48 89 e5 mov %rsp,%rbp
76 400490: 48 c1 fe 03 sar $0x3,%rsi 400490: 48 c1 fe 03 sar $0x3,%rsi
77 400494: 48 89 f0 mov %rsi,%rax 400494: 48 89 f0 mov %rsi,%rax
78 400497: 48 c1 e8 3f shr $0x3f,%rax 400497: 48 c1 e8 3f shr $0x3f,%rax
79 40049b: 48 01 c6 add %rax,%rsi 40049b: 48 01 c6 add %rax,%rsi
80 40049e: 48 d1 fe sar %rsi 40049e: 48 d1 fe sar %rsi
81 4004a1: 74 15 je 4004b8 <register_tm_clo 4004a1: 74 15 je 4004b8 <register_tm_clo
82 4004a3: b8 00 00 00 00 mov $0x0,%eax 4004a3: b8 00 00 00 00 mov $0x0,%eax
83 4004a8: 48 85 c0 test %rax,%rax 4004a8: 48 85 c0 test %rax,%rax
84 4004ab: 74 0b je 4004b8 <register_tm_clo 4004ab: 74 0b je 4004b8 <register_tm_clo
85 4004ad: 5d pop %rbp 4004ad: 5d pop %rbp
86 4004ae: bf 20 09 60 00 mov $0x600920,%edi 4004ae: bf 20 09 60 00 mov $0x600920,%edi
87 4004b3: ff e0 jmpq *%rax 4004b3: ff e0 jmpq *%rax
88 4004b5: 0f 1f 00 nopl (%rax) 4004b5: 0f 1f 00 nopl (%rax)
89 4004b8: 5d pop %rbp 4004b8: 5d pop %rbp
90 4004b9: c3 retq 4004b9: c3 retq
91 4004ba: 66 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1) 4004ba: 66 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1)
92
93 0000000004004c0 <__do_global_ctors_aux>: 00000000004004c0 <__do_global_ctors_aux>:
94 4004c0: 80 3d 61 04 20 00 00 cmpb $0x0,0x200461(%rip) 4004c0: 80 3d 61 04 20 00 00 cmpb $0x0,0x200461(%rip)
95 4004c7: 75 17 jne 4004e0 <__do_global_dto 4004c7: 75 17 jne 4004e0 <__do_global_dto
96 4004c9: 55 push %rbp 4004c9: 55 push %rbp
97 4004ca: 48 89 e5 mov %rsp,%rbp 4004ca: 48 89 e5 mov %rsp,%rbp
98 4004cd: e8 7e ff ff ff callq 400450 <deregister_tm_c 4004cd: e8 7e ff ff ff callq 400450 <deregister_tm_c
99 4004d2: c6 05 4f 04 20 00 01 movb $0x1,0x20044f(%rip) 4004d2: c6 05 4f 04 20 00 01 movb $0x1,0x20044f(%rip)
100 4004d9: 5d pop %rbp 4004d9: 5d pop %rbp
101 4004da: c3 retq 4004da: c3 retq
102 4004db: 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1) 4004db: 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1)
103 4004e0: f3 c3 repz retq 4004e0: f3 c3 repz retq
104 4004e2: 0f 1f 40 00 nopl 0x0(%rax) 4004e2: 0f 1f 40 00 nopl 0x0(%rax)
105 4004e6: 66 2e 0f 1f 84 00 00 nopw %cs:0x0(%rax,%rax,1) 4004e6: 66 2e 0f 1f 84 00 00 nopw %cs:0x0(%rax,%rax,1)

```

```

106 4004ed: 00 00 00          4004ed: 00 00 00
107
108 00000000004004f0 <frame_dummy>:          00000000004004f0 <frame_dummy>:
109 4004f0: 55          push    %rbp          4004f0: 55          push    %rbp
110 4004f1: 48 89 e5    mov     %rsp,%rbp          4004f1: 48 89 e5    mov     %rsp,%rbp
111 4004f4: 5d          pop     %rbp          4004f4: 5d          pop     %rbp
112 4004f5: eb 89       jmp     400480 <register_tm_clo>          4004f5: eb 89       jmp     400480 <register_tm_clo>
113
114 00000000004004f7 <main>:          00000000004004f7 <main>:
115 4004f7: 55          push    %rbp          4004f7: 55          push    %rbp
116 4004f8: 48 89 e5    mov     %rsp,%rbp          4004f8: 48 89 e5    mov     %rsp,%rbp
117 4004fb: 48 83 ec 10 sub     $0x10,%rsp          4004fb: 48 83 ec 10 sub     $0x10,%rsp
118 4004ff: c7 45 fc 00 00 00 00 movl    $0x0,-0x4(%rbp)          4004ff: c7 45 fc 00 00 00 00 movl    $0x0,-0x4(%rbp)
119 400506: 83 7d fc 09 cmpl    $0x9,-0x4(%rbp)          400506: 83 7d fc 09 cmpl    $0x9,-0x4(%rbp)
120 40050a: 7f 28       jg      400534 <main+0x3d>          40050a: 7f 28       jg      400534 <main+0x3d>
121 40050c: 8b 45 fc    mov     -0x4(%rbp),%eax          40050c: 8b 45 fc    mov     -0x4(%rbp),%eax
122 40050f: 89 45 f8    mov     %eax,-0x8(%rbp)          40050f: 89 45 f8    mov     %eax,-0x8(%rbp)
123 400512: 48 8b 05 07 04 20 00 mov     0x200407(%rip),%rax          400512: 48 8b 05 07 04 20 00 mov     0x200407(%rip),%rax
124 400519: 8b 55 f8    mov     -0x8(%rbp),%edx          400519: 8b 55 f8    mov     -0x8(%rbp),%edx
125 40051c: be c4 05 40 00 mov     $0x4005c4,%esi          40051c: be c4 05 40 00 mov     $0x4005c4,%esi
126 400521: 48 89 c7    mov     %rax,%rdi          400521: 48 89 c7    mov     %rax,%rdi
127 400524: b8 00 00 00 00 mov     $0x0,%eax          400524: b8 00 00 00 00 mov     $0x0,%eax
128 400529: e8 d2 fe ff ff callq   400400 <fprintf@plt>          400529: e8 d2 fe ff ff callq   400400 <fprintf@plt>
129 40052e: 83 45 fc 01 addl    $0x1,-0x4(%rbp)          40052e: 83 45 fc 01 addl    $0x1,-0x4(%rbp)
130 400532: eb d2       jmp     400506 <main+0xf>          400532: eb d2       jmp     400506 <main+0xf>
131 400534: b8 00 00 00 00 mov     $0x0,%eax          400534: b8 00 00 00 00 mov     $0x0,%eax
132 400539: c9          leaveq  %eax          400539: c9          leaveq  %eax
133 40053a: c3          retq                    40053a: c3          retq
134 40053b: 0f 1f 44 00 00 nopl    0x0(%rax,%rax,1)          40053b: 0f 1f 44 00 00 nopl    0x0(%rax,%rax,1)
135
136 0000000000400540 <__libc_csu_init>:          0000000000400540 <__libc_csu_init>:
137 400540: 41 57       push    %r15          400540: 41 57       push    %r15
138 400542: 41 56       push    %r14          400542: 41 56       push    %r14
139 400544: 41 89 ff    mov     %edi,%r15d          400544: 41 89 ff    mov     %edi,%r15d
140 400547: 41 55       push    %r13          400547: 41 55       push    %r13
141 400549: 41 54       push    %r12          400549: 41 54       push    %r12
142 40054b: 4c 8d 25 ae 01 20 00 lea     0x2001ae(%rip),%r12          40054b: 4c 8d 25 ae 01 20 00 lea     0x2001ae(%rip),%r12
143 400552: 55          push    %rbp          400552: 55          push    %rbp
144 400553: 48 8d 2d ae 01 20 00 lea     0x2001ae(%rip),%rbp          400553: 48 8d 2d ae 01 20 00 lea     0x2001ae(%rip),%rbp
145 40055a: 53          push    %rbx          40055a: 53          push    %rbx
146 40055b: 49 89 f6    mov     %rsi,%r14          40055b: 49 89 f6    mov     %rsi,%r14
147 40055e: 49 89 d5    mov     %rdx,%r13          40055e: 49 89 d5    mov     %rdx,%r13
148 400561: 4c 29 e5    sub     %r12,%rbp          400561: 4c 29 e5    sub     %r12,%rbp
149 400564: 48 83 ec 08 sub     $0x8,%rsp          400564: 48 83 ec 08 sub     $0x8,%rsp
150 400568: 48 c1 fd 03 sar     $0x3,%rbp          400568: 48 c1 fd 03 sar     $0x3,%rbp
151 40056c: e8 47 fe ff ff callq   4003b8 <_init>          40056c: e8 47 fe ff ff callq   4003b8 <_init>
152 400571: 48 85 ed    test    %rbp,%rbp          400571: 48 85 ed    test    %rbp,%rbp
153 400574: 74 20       je      400596 <__libc_csu_init>          400574: 74 20       je      400596 <__libc_csu_init>
154 400576: 31 db       xor     %ebx,%ebx          400576: 31 db       xor     %ebx,%ebx
155 400578: 0f 1f 84 00 00 00 00 xor     0x0(%rax,%rax,1)          400578: 0f 1f 84 00 00 00 00 xor     0x0(%rax,%rax,1)
156 40057f: 00          nopl    0x0(%rax,%rax,1)          40057f: 00          nopl    0x0(%rax,%rax,1)
157 400580: 4c 89 ea    mov     %r13,%rdx          400580: 4c 89 ea    mov     %r13,%rdx
158 400583: 4c 89 f6    mov     %r14,%rsi          400583: 4c 89 f6    mov     %r14,%rsi
159 400586: 44 89 ff    mov     %r15d,%edi          400586: 44 89 ff    mov     %r15d,%edi
160 400589: 41 ff 14 dc callq   *(%r12,%rbx,8)          400589: 41 ff 14 dc callq   *(%r12,%rbx,8)
161 40058d: 48 83 c3 01 add     $0x1,%rbx          40058d: 48 83 c3 01 add     $0x1,%rbx
162 400591: 48 39 eb    cmp     %rbp,%rbx          400591: 48 39 eb    cmp     %rbp,%rbx
163 400594: 75 ea       jne     400580 <__libc_csu_init>          400594: 75 ea       jne     400580 <__libc_csu_init>
164 400596: 48 83 c4 08 add     $0x8,%rsp          400596: 48 83 c4 08 add     $0x8,%rsp
165 40059a: 5b          pop     %rbx          40059a: 5b          pop     %rbx
166 40059b: 5d          pop     %rbp          40059b: 5d          pop     %rbp
167 40059c: 41 5c       pop     %r12          40059c: 41 5c       pop     %r12
168 40059e: 41 5d       pop     %r13          40059e: 41 5d       pop     %r13
169 4005a0: 41 5e       pop     %r14          4005a0: 41 5e       pop     %r14
170 4005a2: 41 5f       pop     %r15          4005a2: 41 5f       pop     %r15
171 4005a4: c3          retq                    4005a4: c3          retq
172 4005a5: 90          nop                    4005a5: 90          nop
173 4005a6: 66 2e 0f 1f 84 00 00 nopw    %cs:0x0(%rax,%rax,1)          4005a6: 66 2e 0f 1f 84 00 00 nopw    %cs:0x0(%rax,%rax,1)
174 4005ad: 00 00 00    nopw                    4005ad: 00 00 00    nopw
175
176 00000000004005b0 <__libc_csu_fini>:          00000000004005b0 <__libc_csu_fini>:
177 4005b0: f3 c3      repz retq          4005b0: f3 c3      repz retq
178
179 Disassembly of section .fini:          Disassembly of section .fini:
180
181 00000000004005b4 <_fini>:          00000000004005b4 <_fini>:
182 4005b4: 48 83 ec 08 sub     $0x8,%rsp          4005b4: 48 83 ec 08 sub     $0x8,%rsp
183 4005b8: 48 83 c4 08 add     $0x8,%rsp          4005b8: 48 83 c4 08 add     $0x8,%rsp
184 4005bc: c3          retq                    4005bc: c3          retq
185
186 == No Difference ==

```

## Memory Initialization

declared in .BSS segment

```

1 // segments_1.cc
2
3 #include <stdio.h>
4 #include <string.h>
5
6 #define N (1024)
7
8 int value[N]; // .bss
9
10 int main(void) {
11     memset(&value, 'x', N);
12     fprintf(stdout, "value: %d\n", value[0]);
13     return 0;
14 }
15
16 // *EOF*

```

```

1
2 ./bin/segments_1:      file format elf64-x86-64
3 ./bin/segments_1
4 architecture: i386:x86-64, flags 0x00000112:
5 EXEC_P, HAS_SYMS, D_PAGED
6 start address 0x000000000400470
7
8 Program Header:
9   PHDR off 0x0000000000000040 vaddr 0x000000000400040 paddr 0x000000000400040 align 2**3
10     filesz 0x00000000000001c0 memsz 0x00000000000001c0 flags r-x
11   INTERP off 0x0000000000000200 vaddr 0x000000000400200 paddr 0x000000000400200 align 2**0
12     filesz 0x000000000000001c memsz 0x000000000000001c flags r--
13   LOAD off 0x0000000000000000 vaddr 0x000000000400000 paddr 0x000000000400000 align 2**21
14     filesz 0x000000000000074c memsz 0x000000000000074c flags r-x
15   LOAD off 0x0000000000000750 vaddr 0x000000000600750 paddr 0x000000000600750 align 2**21
16     filesz 0x0000000000000228 memsz 0x0000000000001250 flags rw-
17   DYNAMIC off 0x0000000000000760 vaddr 0x000000000600760 paddr 0x000000000600760 align 2**3
18     filesz 0x00000000000001d0 memsz 0x00000000000001d0 flags rw-
19   NOTE off 0x000000000000021c vaddr 0x00000000040021c paddr 0x00000000040021c align 2**2
20     filesz 0x0000000000000020 memsz 0x0000000000000020 flags r--
21   EH_FRAME off 0x0000000000000620 vaddr 0x000000000400620 paddr 0x000000000400620 align 2**2
22     filesz 0x0000000000000034 memsz 0x0000000000000034 flags r--
23   STACK off 0x0000000000000000 vaddr 0x0000000000000000 paddr 0x0000000000000000 align 2**4
24     filesz 0x0000000000000000 memsz 0x0000000000000000 flags rw-
25
26 Dynamic Section:
27   NEEDED          libc.so.6
28   INIT            0x0000000004003f8
29   FINI            0x000000000400604
30   INIT_ARRAY      0x000000000600750

```

31	INIT_ARRAYSZ	0x0000000000000008				
32	FINI_ARRAY	0x0000000000600758				
33	FINI_ARRAYSZ	0x0000000000000008				
34	HASH	0x0000000000400240				
35	STRTAB	0x0000000000400300				
36	SYMTAB	0x0000000000400270				
37	STRSZ	0x000000000000004e				
38	SYMENT	0x0000000000000018				
39	DEBUG	0x0000000000000000				
40	PLTGOT	0x0000000000600938				
41	PLTRELSZ	0x0000000000000048				
42	PLTREL	0x0000000000000007				
43	JMPREL	0x00000000004003b0				
44	RELA	0x0000000000400380				
45	RELASZ	0x0000000000000030				
46	RELAENT	0x0000000000000018				
47	VERNEED	0x0000000000400360				
48	VERNEEDNUM	0x0000000000000001				
49	VERSYM	0x000000000040034e				
50						
51	Version References:					
52	required from libc.so.6:					
53	0x09691a75 0x00 02 GLIBC_2.2.5					
54						
55	Sections:					
Idx	Name	Size	VMA	LMA	File off	Algn
57	0 .interp	0000001c	0000000000400200	0000000000400200	00000200	2**0
58		CONTENTS, ALLOC, LOAD, READONLY, DATA				
59	1 .note.ABI-tag	00000020	000000000040021c	000000000040021c	0000021c	2**2
60		CONTENTS, ALLOC, LOAD, READONLY, DATA				
61	2 .hash	0000002c	0000000000400240	0000000000400240	00000240	2**3
62		CONTENTS, ALLOC, LOAD, READONLY, DATA				
63	3 .dynsym	00000090	0000000000400270	0000000000400270	00000270	2**3
64		CONTENTS, ALLOC, LOAD, READONLY, DATA				
65	4 .dynstr	0000004e	0000000000400300	0000000000400300	00000300	2**0
66		CONTENTS, ALLOC, LOAD, READONLY, DATA				
67	5 .gnu.version	0000000c	000000000040034e	000000000040034e	0000034e	2**1
68		CONTENTS, ALLOC, LOAD, READONLY, DATA				
69	6 .gnu.version_r	00000020	0000000000400360	0000000000400360	00000360	2**3
70		CONTENTS, ALLOC, LOAD, READONLY, DATA				
71	7 .rela.dyn	00000030	0000000000400380	0000000000400380	00000380	2**3
72		CONTENTS, ALLOC, LOAD, READONLY, DATA				
73	8 .rela.plt	00000048	00000000004003b0	00000000004003b0	000003b0	2**3
74		CONTENTS, ALLOC, LOAD, READONLY, DATA				
75	9 .init	0000001a	00000000004003f8	00000000004003f8	000003f8	2**2
76		CONTENTS, ALLOC, LOAD, READONLY, CODE				
77	10 .plt	00000040	0000000000400420	0000000000400420	00000420	2**4
78		CONTENTS, ALLOC, LOAD, READONLY, CODE				
79	11 .plt.got	00000008	0000000000400460	0000000000400460	00000460	2**3
80		CONTENTS, ALLOC, LOAD, READONLY, CODE				
81	12 .text	00000192	0000000000400470	0000000000400470	00000470	2**4
82		CONTENTS, ALLOC, LOAD, READONLY, CODE				
83	13 .fini	00000009	0000000000400604	0000000000400604	00000604	2**2
84		CONTENTS, ALLOC, LOAD, READONLY, CODE				
85	14 .rodata	0000000f	0000000000400610	0000000000400610	00000610	2**2
86		CONTENTS, ALLOC, LOAD, READONLY, DATA				
87	15 .eh_frame_hdr	00000034	0000000000400620	0000000000400620	00000620	2**2

88		CONTENTS, ALLOC, LOAD, READONLY, DATA				
89	16 .eh_frame	000000f4 0000000000400658 0000000000400658 00000658 2**3				
90		CONTENTS, ALLOC, LOAD, READONLY, DATA				
91	17 .init_array	00000008 0000000000600750 0000000000600750 00000750 2**3				
92		CONTENTS, ALLOC, LOAD, DATA				
93	18 .fini_array	00000008 0000000000600758 0000000000600758 00000758 2**3				
94		CONTENTS, ALLOC, LOAD, DATA				
95	19 .dynamic	000001d0 0000000000600760 0000000000600760 00000760 2**3				
96		CONTENTS, ALLOC, LOAD, DATA				
97	20 .got	00000008 0000000000600930 0000000000600930 00000930 2**3				
98		CONTENTS, ALLOC, LOAD, DATA				
99	21 .got.plt	00000030 0000000000600938 0000000000600938 00000938 2**3				
100		CONTENTS, ALLOC, LOAD, DATA				
101	22 .data	00000010 0000000000600968 0000000000600968 00000968 2**3				
102		CONTENTS, ALLOC, LOAD, DATA				
103	23 .bss	00001020 0000000000600980 0000000000600980 00000978 2**5				
104		ALLOC				
105	24 .comment	00000029 0000000000000000 0000000000000000 00000978 2**0				
106		CONTENTS, READONLY				
107	25 .debug_aranges	00000030 0000000000000000 0000000000000000 000009a1 2**0				
108		CONTENTS, READONLY, DEBUGGING				
109	26 .debug_info	00000351 0000000000000000 0000000000000000 000009d1 2**0				
110		CONTENTS, READONLY, DEBUGGING				
111	27 .debug_abbrev	000000fa 0000000000000000 0000000000000000 00000d22 2**0				
112		CONTENTS, READONLY, DEBUGGING				
113	28 .debug_line	000000f0 0000000000000000 0000000000000000 00000e1c 2**0				
114		CONTENTS, READONLY, DEBUGGING				
115	29 .debug_str	000002b8 0000000000000000 0000000000000000 00000f0c 2**0				
116		CONTENTS, READONLY, DEBUGGING				
117	SYMBOL TABLE:					
118	0000000000400200 1	d .interp 0000000000000000		.interp		
119	000000000040021c 1	d .note.ABI-tag 0000000000000000		.note.ABI-tag		
120	0000000000400240 1	d .hash 0000000000000000		.hash		
121	0000000000400270 1	d .dynsym 0000000000000000		.dynsym		
122	0000000000400300 1	d .dynstr 0000000000000000		.dynstr		
123	000000000040034e 1	d .gnu.version 0000000000000000		.gnu.version		
124	0000000000400360 1	d .gnu.version_r 0000000000000000		.gnu.version_r		
125	0000000000400380 1	d .rela.dyn 0000000000000000		.rela.dyn		
126	00000000004003b0 1	d .rela.plt 0000000000000000		.rela.plt		
127	00000000004003f8 1	d .init 0000000000000000		.init		
128	0000000000400420 1	d .plt 0000000000000000		.plt		
129	0000000000400460 1	d .plt.got 0000000000000000		.plt.got		
130	0000000000400470 1	d .text 0000000000000000		.text		
131	0000000000400604 1	d .fini 0000000000000000		.fini		
132	0000000000400610 1	d .rodata 0000000000000000		.rodata		
133	0000000000400620 1	d .eh_frame_hdr 0000000000000000		.eh_frame_hdr		
134	0000000000400658 1	d .eh_frame 0000000000000000		.eh_frame		
135	0000000000600750 1	d .init_array 0000000000000000		.init_array		
136	0000000000600758 1	d .fini_array 0000000000000000		.fini_array		
137	0000000000600760 1	d .dynamic 0000000000000000		.dynamic		
138	0000000000600930 1	d .got 0000000000000000		.got		
139	0000000000600938 1	d .got.plt 0000000000000000		.got.plt		
140	0000000000600968 1	d .data 0000000000000000		.data		
141	0000000000600980 1	d .bss 0000000000000000		.bss		
142	0000000000000000 1	d .comment 0000000000000000		.comment		
143	0000000000000000 1	d .debug_aranges 0000000000000000		.debug_aranges		
144	0000000000000000 1	d .debug_info 0000000000000000		.debug_info		

```

145 0000000000000000 1 d .debug_abbrev 0000000000000000 .debug_abbrev
146 0000000000000000 1 d .debug_line 0000000000000000 .debug_line
147 0000000000000000 1 d .debug_str 0000000000000000 .debug_str
148 0000000000000000 1 df *ABS* 0000000000000000 crtstuff.c
149 00000000004004a0 1 F .text 0000000000000000 deregister_tm_clones
150 00000000004004d0 1 F .text 0000000000000000 register_tm_clones
151 0000000000400510 1 F .text 0000000000000000 __do_global_dtors_aux
152 0000000000600988 1 O .bss 0000000000000001 completed.6981
153 0000000000600758 1 O .fini_array 0000000000000000 __do_global_dtors_aux_fini_array_entry
154 0000000000400540 1 F .text 0000000000000000 frame_dummy
155 0000000000600750 1 O .init_array 0000000000000000 __frame_dummy_init_array_entry
156 0000000000000000 1 df *ABS* 0000000000000000 segments_1.cc
157 0000000000000000 1 df *ABS* 0000000000000000 crtstuff.c
158 0000000000400748 1 O .eh_frame 0000000000000000 __FRAME_END__
159 0000000000000000 1 df *ABS* 0000000000000000
160 0000000000600758 1 .init_array 0000000000000000 __init_array_end
161 0000000000600760 1 O .dynamic 0000000000000000 _DYNAMIC
162 0000000000600750 1 .init_array 0000000000000000 __init_array_start
163 0000000000400620 1 .eh_frame_hdr 0000000000000000 __GNU_EH_FRAME_HDR
164 0000000000600938 1 O .got.plt 0000000000000000 _GLOBAL_OFFSET_TABLE_
165 0000000000400600 g F .text 0000000000000002 __libc_csu_fini
166 0000000000000000 w *UND* 0000000000000000 _ITM_deregisterTMCloneTable
167 0000000000600980 g O .bss 0000000000000008 stdout@@GLIBC_2.2.5
168 0000000000600968 w .data 0000000000000000 data_start
169 0000000000600978 g .data 0000000000000000 _edata
170 0000000000400604 g F .fini 0000000000000000 _fini
171 0000000000000000 F *UND* 0000000000000000 memset@@GLIBC_2.2.5
172 0000000000000000 F *UND* 0000000000000000 __libc_start_main@@GLIBC_2.2.5
173 0000000000600968 g .data 0000000000000000 __data_start
174 0000000000000000 F *UND* 0000000000000000 fprintf@@GLIBC_2.2.5
175 0000000000000000 w *UND* 0000000000000000 __gmon_start__
176 00000000006009a0 g O .bss 0000000000000100 value
177 0000000000600970 g O .data 0000000000000000 .hidden __dso_handle
178 0000000000400610 g O .rodata 0000000000000004 _IO_stdin_used
179 0000000000400590 g F .text 0000000000000065 __libc_csu_init
180 00000000006019a0 g .bss 0000000000000000 _end
181 0000000000400470 g F .text 000000000000002a _start
182 0000000000600978 g .bss 0000000000000000 __bss_start
183 0000000000400547 g F .text 000000000000003e main
184 0000000000600978 g O .data 0000000000000000 .hidden __TMC_END__
185 0000000000000000 w *UND* 0000000000000000 _ITM_registerTMCloneTable
186 00000000004003f8 g F .init 0000000000000000 _init
187

```

declared in .DATA segment

```

1 // segments_2.cc
2
3 #include <stdio.h>
4
5 #define N (1024)
6

```

```

7   int value[N]; // .data
8
9   int main(void) {
10      value[0] = 0xdeadbeef;
11      value[N] = 0xdeadbeef;
12      fprintf(stdout, "value: %d\n", value[0]);
13      return 0;
14  }
15
16  // *EOF*

```

```

1
2  ./bin/segments_2:      file format elf64-x86-64
3  ./bin/segments_2
4  architecture: i386:x86-64, flags 0x00000112:
5  EXEC_P, HAS_SYMS, D_PAGED
6  start address 0x000000000400420
7
8  Program Header:
9      PHDR off 0x0000000000000040 vaddr 0x0000000004000040 paddr 0x0000000004000040 align 2**3
10      filesz 0x00000000000001c0 memsz 0x00000000000001c0 flags r-x
11      INTERP off 0x0000000000000200 vaddr 0x0000000004000200 paddr 0x0000000004000200 align 2**0
12      filesz 0x000000000000001c memsz 0x000000000000001c flags r--
13      LOAD off 0x0000000000000000 vaddr 0x0000000004000000 paddr 0x0000000004000000 align 2**21
14      filesz 0x00000000000006ec memsz 0x00000000000006ec flags r-x
15      LOAD off 0x00000000000006f0 vaddr 0x0000000006006f0 paddr 0x0000000006006f0 align 2**21
16      filesz 0x0000000000000220 memsz 0x00000000000001250 flags rw-
17      DYNAMIC off 0x0000000000000700 vaddr 0x000000000600700 paddr 0x000000000600700 align 2**3
18      filesz 0x00000000000001d0 memsz 0x00000000000001d0 flags rw-
19      NOTE off 0x000000000000021c vaddr 0x00000000040021c paddr 0x00000000040021c align 2**2
20      filesz 0x0000000000000020 memsz 0x0000000000000020 flags r--
21      EH_FRAME off 0x00000000000005c0 vaddr 0x0000000004005c0 paddr 0x0000000004005c0 align 2**2
22      filesz 0x0000000000000034 memsz 0x0000000000000034 flags r--
23      STACK off 0x0000000000000000 vaddr 0x0000000000000000 paddr 0x0000000000000000 align 2**4
24      filesz 0x0000000000000000 memsz 0x0000000000000000 flags rw-
25
26  Dynamic Section:
27      NEEDED          libc.so.6
28      INIT             0x0000000004003b8
29      FINI             0x0000000004005a4
30      INIT_ARRAY       0x0000000006006f0
31      INIT_ARRAYSZ     0x0000000000000008
32      FINI_ARRAY       0x0000000006006f8
33      FINI_ARRAYSZ     0x0000000000000008
34      HASH             0x000000000400240
35      STRTAB           0x0000000004002e0
36      SYMTAB           0x000000000400268
37      STRSZ            0x0000000000000047
38      SYMENT           0x0000000000000018
39      DEBUG            0x0000000000000000
40      PLTGOT           0x0000000006008d8
41      PLTRELSZ         0x0000000000000030
42      PLTREL           0x0000000000000007
43      JMPREL           0x000000000400388

```

44	RELA	0x0000000000400358				
45	RELASZ	0x0000000000000030				
46	RELAENT	0x0000000000000018				
47	VERNEED	0x0000000000400338				
48	VERNEEDNUM	0x0000000000000001				
49	VERSYM	0x0000000000400328				
50						
51	Version References:					
52	required from libc.so.6:					
53	0x09691a75 0x00 02 GLIBC_2.2.5					
54						
55	Sections:					
56	Idx Name	Size	VMA	LMA	File off	Algn
57	0 .interp	0000001c	0000000000400200	0000000000400200	00000200	2**0
58			CONTENTS, ALLOC, LOAD, READONLY, DATA			
59	1 .note.ABI-tag	00000020	000000000040021c	000000000040021c	0000021c	2**2
60			CONTENTS, ALLOC, LOAD, READONLY, DATA			
61	2 .hash	00000028	0000000000400240	0000000000400240	00000240	2**3
62			CONTENTS, ALLOC, LOAD, READONLY, DATA			
63	3 .dynsym	00000078	0000000000400268	0000000000400268	00000268	2**3
64			CONTENTS, ALLOC, LOAD, READONLY, DATA			
65	4 .dynstr	00000047	00000000004002e0	00000000004002e0	000002e0	2**0
66			CONTENTS, ALLOC, LOAD, READONLY, DATA			
67	5 .gnu.version	0000000a	0000000000400328	0000000000400328	00000328	2**1
68			CONTENTS, ALLOC, LOAD, READONLY, DATA			
69	6 .gnu.version_r	00000020	0000000000400338	0000000000400338	00000338	2**3
70			CONTENTS, ALLOC, LOAD, READONLY, DATA			
71	7 .rela.dyn	00000030	0000000000400358	0000000000400358	00000358	2**3
72			CONTENTS, ALLOC, LOAD, READONLY, DATA			
73	8 .rela.plt	00000030	0000000000400388	0000000000400388	00000388	2**3
74			CONTENTS, ALLOC, LOAD, READONLY, DATA			
75	9 .init	0000001a	00000000004003b8	00000000004003b8	000003b8	2**2
76			CONTENTS, ALLOC, LOAD, READONLY, CODE			
77	10 .plt	00000030	00000000004003e0	00000000004003e0	000003e0	2**4
78			CONTENTS, ALLOC, LOAD, READONLY, CODE			
79	11 .plt.got	00000008	0000000000400410	0000000000400410	00000410	2**3
80			CONTENTS, ALLOC, LOAD, READONLY, CODE			
81	12 .text	00000182	0000000000400420	0000000000400420	00000420	2**4
82			CONTENTS, ALLOC, LOAD, READONLY, CODE			
83	13 .fini	00000009	00000000004005a4	00000000004005a4	000005a4	2**2
84			CONTENTS, ALLOC, LOAD, READONLY, CODE			
85	14 .rodata	0000000f	00000000004005b0	00000000004005b0	000005b0	2**2
86			CONTENTS, ALLOC, LOAD, READONLY, DATA			
87	15 .eh_frame_hdr	00000034	00000000004005c0	00000000004005c0	000005c0	2**2
88			CONTENTS, ALLOC, LOAD, READONLY, DATA			
89	16 .eh_frame	000000f4	00000000004005f8	00000000004005f8	000005f8	2**3
90			CONTENTS, ALLOC, LOAD, READONLY, DATA			
91	17 .init_array	00000008	00000000006006f0	00000000006006f0	000006f0	2**3
92			CONTENTS, ALLOC, LOAD, DATA			
93	18 .fini_array	00000008	00000000006006f8	00000000006006f8	000006f8	2**3
94			CONTENTS, ALLOC, LOAD, DATA			
95	19 .dynamic	000001d0	0000000000600700	0000000000600700	00000700	2**3
96			CONTENTS, ALLOC, LOAD, DATA			
97	20 .got	00000008	00000000006008d0	00000000006008d0	000008d0	2**3
98			CONTENTS, ALLOC, LOAD, DATA			
99	21 .got.plt	00000028	00000000006008d8	00000000006008d8	000008d8	2**3
100			CONTENTS, ALLOC, LOAD, DATA			



101	22	.data	00000010	00000000000600900	00000000000600900	00000900	2**3
102			CONTENTS, ALLOC, LOAD, DATA				
103	23	.bss	00001020	00000000000600920	00000000000600920	00000910	2**5
104			ALLOC				
105	24	.comment	00000029	00000000000000000	00000000000000000	00000910	2**0
106			CONTENTS, READONLY				
107	25	.debug_aranges	00000030	00000000000000000	00000000000000000	00000939	2**0
108			CONTENTS, READONLY, DEBUGGING				
109	26	.debug_info	00000351	00000000000000000	00000000000000000	00000969	2**0
110			CONTENTS, READONLY, DEBUGGING				
111	27	.debug_abbrev	000000fa	00000000000000000	00000000000000000	00000cba	2**0
112			CONTENTS, READONLY, DEBUGGING				
113	28	.debug_line	000000ee	00000000000000000	00000000000000000	00000db4	2**0
114			CONTENTS, READONLY, DEBUGGING				
115	29	.debug_str	000002b8	00000000000000000	00000000000000000	00000ea2	2**0
116			CONTENTS, READONLY, DEBUGGING				
117		SYMBOL TABLE:					
118	0000000000400200	1	d	.interp	00000000000000000	.interp	
119	000000000040021c	1	d	.note.ABI-tag	00000000000000000	.note.ABI-tag	
120	0000000000400240	1	d	.hash	00000000000000000	.hash	
121	0000000000400268	1	d	.dynsym	00000000000000000	.dynsym	
122	00000000004002e0	1	d	.dynstr	00000000000000000	.dynstr	
123	0000000000400328	1	d	.gnu.version	00000000000000000	.gnu.version	
124	0000000000400338	1	d	.gnu.version_r	00000000000000000	.gnu.version_r	
125	0000000000400358	1	d	.rela.dyn	00000000000000000	.rela.dyn	
126	0000000000400388	1	d	.rela.plt	00000000000000000	.rela.plt	
127	00000000004003b8	1	d	.init	00000000000000000	.init	
128	00000000004003e0	1	d	.plt	00000000000000000	.plt	
129	0000000000400410	1	d	.plt.got	00000000000000000	.plt.got	
130	0000000000400420	1	d	.text	00000000000000000	.text	
131	00000000004005a4	1	d	.fini	00000000000000000	.fini	
132	00000000004005b0	1	d	.rodata	00000000000000000	.rodata	
133	00000000004005c0	1	d	.eh_frame_hdr	00000000000000000	.eh_frame_hdr	
134	00000000004005f8	1	d	.eh_frame	00000000000000000	.eh_frame	
135	00000000006006f0	1	d	.init_array	00000000000000000	.init_array	
136	00000000006006f8	1	d	.fini_array	00000000000000000	.fini_array	
137	0000000000600700	1	d	.dynamic	00000000000000000	.dynamic	
138	00000000006008d0	1	d	.got	00000000000000000	.got	
139	00000000006008d8	1	d	.got.plt	00000000000000000	.got.plt	
140	0000000000600900	1	d	.data	00000000000000000	.data	
141	0000000000600920	1	d	.bss	00000000000000000	.bss	
142	0000000000000000	1	d	.comment	00000000000000000	.comment	
143	0000000000000000	1	d	.debug_aranges	00000000000000000	.debug_aranges	
144	0000000000000000	1	d	.debug_info	00000000000000000	.debug_info	
145	0000000000000000	1	d	.debug_abbrev	00000000000000000	.debug_abbrev	
146	0000000000000000	1	d	.debug_line	00000000000000000	.debug_line	
147	0000000000000000	1	d	.debug_str	00000000000000000	.debug_str	
148	0000000000000000	1	df	*ABS*	00000000000000000	crtstuff.c	
149	0000000000400450	1	F	.text	00000000000000000	deregister_tm_clones	
150	0000000000400480	1	F	.text	00000000000000000	register_tm_clones	
151	00000000004004c0	1	F	.text	00000000000000000	__do_global_dtors_aux	
152	0000000000600928	1	0	.bss	00000000000000001	completed.6981	
153	00000000006006f8	1	0	.fini_array	00000000000000000	__do_global_dtors_aux_fini_array_entry	
154	00000000004004f0	1	F	.text	00000000000000000	frame_dummy	
155	00000000006006f0	1	0	.init_array	00000000000000000	__frame_dummy_init_array_entry	
156	0000000000000000	1	df	*ABS*	00000000000000000	segments_2.cc	
157	0000000000000000	1	df	*ABS*	00000000000000000	crtstuff.c	

```

158 00000000004006e8 l 0 .eh_frame 0000000000000000 __FRAME_END__
159 0000000000000000 df *ABS* 0000000000000000
160 00000000006006f8 l .init_array 0000000000000000 __init_array_end
161 0000000000600700 l 0 .dynamic 0000000000000000 _DYNAMIC
162 00000000006006f0 l .init_array 0000000000000000 __init_array_start
163 00000000004005c0 l .eh_frame_hdr 0000000000000000 __GNU_EH_FRAME_HDR
164 00000000006008d8 l 0 .got.plt 0000000000000000 _GLOBAL_OFFSET_TABLE_
165 00000000004005a0 g F .text 0000000000000002 __libc_csu_fini
166 0000000000000000 w *UND* 0000000000000000 _ITM_deregisterTMCloneTable
167 0000000000600920 g 0 .bss 0000000000000008 stdout@@GLIBC_2.2.5
168 0000000000600900 w .data 0000000000000000 data_start
169 0000000000600910 g .data 0000000000000000 _edata
170 00000000004005a4 g F .fini 0000000000000000 _fini
171 0000000000000000 F *UND* 0000000000000000 __libc_start_main@@GLIBC_2.2.5
172 0000000000600900 g .data 0000000000000000 __data_start
173 0000000000000000 F *UND* 0000000000000000 fprintf@@GLIBC_2.2.5
174 0000000000000000 w *UND* 0000000000000000 __gmon_start__
175 0000000000600940 g 0 .bss 0000000000001000 value
176 0000000000600908 g 0 .data 0000000000000000 .hidden __dso_handle
177 00000000004005b0 g 0 .rodata 0000000000000004 _IO_stdin_used
178 0000000000400530 g F .text 0000000000000065 __libc_csu_init
179 0000000000601940 g .bss 0000000000000000 _end
180 0000000000400420 g F .text 000000000000002a _start
181 0000000000600910 g .bss 0000000000000000 __bss_start
182 00000000004004f7 g F .text 000000000000002a main
183 0000000000600910 g 0 .data 0000000000000000 .hidden __TMC_END__
184 0000000000000000 w *UND* 0000000000000000 _ITM_registerTMCloneTable
185 00000000004003b8 g F .init 0000000000000000 _init
186

```

## implementation difference (side-by-side diff, with suppression)

```

1  ./bin/segments_1 | ./bin/segments_2
2  start address 0x0000000000400470 | start address 0x0000000000400420
3  filesz 0x000000000000074c memsz 0x000000000000074c flags r-x | filesz 0x00000000000006ec memsz 0x00000000000006ec flags r-x
4  LOAD off 0x0000000000000750 vaddr 0x0000000000000750 paddr 0x00 | LOAD off 0x00000000000006f0 vaddr 0x00000000000006f0 paddr 0x00
5  filesz 0x0000000000000228 memsz 0x00000000000001250 flags rv- | filesz 0x0000000000000220 memsz 0x00000000000001250 flags rv-
6  DYNAMIC off 0x0000000000000760 vaddr 0x0000000000000760 paddr 0x00 | DYNAMIC off 0x0000000000000700 vaddr 0x0000000000000700 paddr 0x00
7  EH_FRAME off 0x0000000000000620 vaddr 0x0000000000000620 paddr 0x00 | EH_FRAME off 0x00000000000005c0 vaddr 0x00000000000005c0 paddr 0x00
8  INIT 0x000000000000003f8 | INIT 0x000000000000003b8
9  FINI 0x00000000000000604 | FINI 0x000000000000005a4
10 INIT_ARRAY 0x00000000000000750 | INIT_ARRAY 0x000000000000006f0
11 FINI_ARRAY 0x00000000000000758 | FINI_ARRAY 0x000000000000006f8
12 STRTAB 0x00000000000000300 | STRTAB 0x000000000000002e0
13 SYMTAB 0x00000000000000270 | SYMTAB 0x00000000000000268
14 STRSZ 0x0000000000000004e | STRSZ 0x0000000000000004f
15 PLTGOT 0x00000000000000938 | PLTGOT 0x000000000000008d8
16 PLTRELSZ 0x00000000000000048 | PLTRELSZ 0x00000000000000030
17 JMPREL 0x000000000000003b0 | JMPREL 0x00000000000000388
18 RELA 0x00000000000000380 | RELA 0x00000000000000358
19 VERNEED 0x00000000000000360 | VERNEED 0x00000000000000338
20 VERSYM 0x0000000000000034e | VERSYM 0x00000000000000328
21 2.hash 0000002c 000000000000400240 000000000000400240 000002 | 2.hash 00000028 000000000000400240 000000000000400240 000002
22 3.dynsym 00000090 000000000000400270 000000000000400270 000002 | 3.dynsym 00000078 000000000000400268 000000000000400268 000002
23 4.dynstr 0000004e 000000000000400300 000000000000400300 000003 | 4.dynstr 00000047 0000000000004002e0 0000000000004002e0 000002
24 5.gnu.version 0000000c 00000000000040034e 00000000000040034e 000003 | 5.gnu.version 0000000a 000000000000400328 000000000000400328 000003
25 6.gnu.version_r 00000020 000000000000400360 000000000000400360 00000 | 6.gnu.version_r 00000020 000000000000400338 000000000000400338 00000
26 7.rela.dyn 00000030 000000000000400380 000000000000400380 000003 | 7.rela.dyn 00000030 000000000000400358 000000000000400358 000003
27 8.rela.plt 00000048 0000000000004003b0 0000000000004003b0 000003 | 8.rela.plt 00000030 000000000000400388 000000000000400388 000003
28 9.init 0000001a 0000000000004003f8 0000000000004003f8 000003 | 9.init 0000001a 0000000000004003b8 0000000000004003b8 000003
29 10.plt 00000040 000000000000400420 000000000000400420 000004 | 10.plt 00000030 0000000000004003e0 0000000000004003e0 000003
30 11.plt.got 00000008 000000000000400460 000000000000400460 000004 | 11.plt.got 00000008 000000000000400410 000000000000400410 000004
31 12.text 00000192 000000000000400470 000000000000400470 000004 | 12.text 00000182 000000000000400420 000000000000400420 000004
32 13.fini 00000009 000000000000400604 000000000000400604 000006 | 13.fini 00000009 0000000000004005a4 0000000000004005a4 000005
33 14.rodata 0000000f 000000000000400610 000000000000400610 000006 | 14.rodata 0000000f 0000000000004005b0 0000000000004005b0 000005
34 15.eh_frame_hdr 00000034 000000000000400620 000000000000400620 000006 | 15.eh_frame_hdr 00000034 0000000000004005c0 0000000000004005c0 000005

```

v