

```

1  StubRoutines::call_stub [0x00007f8ab9000564, 0x00007f8ab900079b] (567 bytes)
2  ;; enter
3  0x00007f8ab9000564: push    %rbp
4  0x00007f8ab9000565: mov     %rsp,%rbp
5  ;; subptr
6  0x00007f8ab9000568: sub     $0x60,%rsp
7  0x00007f8ab900056c: mov     %r9,-0x8(%rbp)
8  0x00007f8ab9000570: mov     %r8,-0x10(%rbp)
9  ;; c_rarg3
10 0x00007f8ab9000574: mov     %rcx,-0x18(%rbp)
11 ;; c_rarg2
12 0x00007f8ab9000578: mov     %edx,-0x20(%rbp)
13 ;; c_rarg1
14 0x00007f8ab900057b: mov     %rsi,-0x28(%rbp)
15 ;; c_rarg0
16 0x00007f8ab900057f: mov     %rdi,-0x30(%rbp)
17 ;; rbx
18 0x00007f8ab9000583: mov     %rbx,-0x38(%rbp)
19 ;; r12
20 0x00007f8ab9000587: mov     %r12,-0x40(%rbp)
21 0x00007f8ab900058b: mov     %r13,-0x48(%rbp)
22 0x00007f8ab900058f: mov     %r14,-0x50(%rbp)
23 0x00007f8ab9000593: mov     %r15,-0x58(%rbp)
24 ;; stmxcsr
25 0x00007f8ab9000597: stmxcsr -0x60(%rbp)
26 ;; movl
27 0x00007f8ab900059b: mov     -0x60(%rbp),%eax
28 ;; andl
29 0x00007f8ab900059e: and     $0xffc0,%eax
30 ;; cmp32
31 0x00007f8ab90005a4: cmp     0x1698e19e(%rip),%eax      # 0x00007f8acf98e748
32 ;; jcc
33 0x00007f8ab90005aa: je      0x00007f8ab90005b7
34 ;; ldmxcsr
35 0x00007f8ab90005b0: ldmxcsr 0x1698e191(%rip)          # 0x00007f8acf98e748
36 ;; bind
37 ;; Load up thread register
38 0x00007f8ab90005b7: mov     0x18(%rbp),%r15
39 ;; reinit_heapbase
40 0x00007f8ab90005bb: mov     0x169adc1e(%rip),%r12     # 0x00007f8acf9ae1e0
41 ;; cmpptr
42 0x00007f8ab90005c2: cmpq    $0x0,0x8(%r15)
43 ;; jcc
44 0x00007f8ab90005ca: je      0x00007f8ab9000647
45 ;; stop
46 0x00007f8ab90005d0: mov     %rsp,-0x28(%rsp)
47 0x00007f8ab90005d5: sub     $0x80,%rsp
48 0x00007f8ab90005dc: mov     %rax,0x78(%rsp)
49 0x00007f8ab90005e1: mov     %rcx,0x70(%rsp)
50 0x00007f8ab90005e6: mov     %rdx,0x68(%rsp)
51 0x00007f8ab90005eb: mov     %rbx,0x60(%rsp)
52 0x00007f8ab90005f0: mov     %rbp,0x50(%rsp)
53 0x00007f8ab90005f5: mov     %rsi,0x48(%rsp)
54 0x00007f8ab90005fa: mov     %rdi,0x40(%rsp)
55 0x00007f8ab90005ff: mov     %r8,0x38(%rsp)
56 0x00007f8ab9000604: mov     %r9,0x30(%rsp)
57 0x00007f8ab9000609: mov     %r10,0x28(%rsp)
58 0x00007f8ab900060e: mov     %r11,0x20(%rsp)
59 0x00007f8ab9000613: mov     %r12,0x18(%rsp)
60 0x00007f8ab9000618: mov     %r13,0x10(%rsp)
61 0x00007f8ab900061d: mov     %r14,0x8(%rsp)
62 0x00007f8ab9000622: mov     %r15,(%rsp)
63 0x00007f8ab9000626: movabs  $0x7f8acf3a6170,%rdi
64 0x00007f8ab9000630: movabs  $0x7f8ab90005d0,%rsi
65 0x00007f8ab900063a: mov     %rsp,%rdx
66 0x00007f8ab900063d: and     $0xfffffffffffffffff0,%rsp
67 0x00007f8ab9000641: callq   0x00007f8aceddb9fa
68 0x00007f8ab9000646: hlt
69 ;; bind
70 ;; pass parameters if any
71 ;; movl
72 0x00007f8ab9000647: mov     0x10(%rbp),%ecx
73 ;; testl
74 0x00007f8ab900064a: test    %ecx,%ecx
75 ;; jcc
76 0x00007f8ab900064c: je      0x00007f8ab9000664
77 ;; movptr
78 0x00007f8ab9000652: mov     -0x8(%rbp),%rdx
79 ;; movl
80 0x00007f8ab9000656: mov     %ecx,%esi
81 ;; BIND
82 ;; loop:
83 ;; movptr
84 0x00007f8ab9000658: mov     (%rdx),%rax
85 ;; addptr
86 0x00007f8ab900065b: add     $0x8,%rdx

```

```
87     ;; decrementl
88     0x00007f8ab900065f: dec     %esi
89     ;; push
90     0x00007f8ab9000661: push   %rax
91     ;; jcc
92     0x00007f8ab9000662: jne     0x00007f8ab9000658
93     ;; BIND
94     ;; parameters_done:
95     ;; movptr
96     0x00007f8ab9000664: mov     -0x18(%rbp),%rbx
97     ;; movptr
98     0x00007f8ab9000668: mov     -0x10(%rbp),%rsi
99     ;; mov
100    0x00007f8ab900066c: mov     %rsp,%r13
101    ;; call Java function
102    0x00007f8ab900066f: callq   %rsi
103    ;; call_stub_return_address:
104    ;; movptr
105    0x00007f8ab9000671: mov     -0x28(%rbp),%rdi
106    ;; movl
107    0x00007f8ab9000675: mov     -0x20(%rbp),%esi
108    ;; cmpl
109    0x00007f8ab9000678: cmp     $0xc,%esi
110    0x00007f8ab900067b: je      0x00007f8ab9000781
111    0x00007f8ab9000681: cmp     $0xb,%esi
112    0x00007f8ab9000684: je      0x00007f8ab9000781
113    0x00007f8ab900068a: cmp     $0x6,%esi
114    0x00007f8ab900068d: je      0x00007f8ab9000789
115    0x00007f8ab9000693: cmp     $0x7,%esi
116    0x00007f8ab9000696: je      0x00007f8ab9000792
117    ;; movl
118    0x00007f8ab900069c: mov     %eax,(%rdi)
119    ;; BIND
120    ;; exit:
121    ;; lea
122    0x00007f8ab900069e: lea     -0x60(%rbp),%rsp
123    ;; cmpptr
124    0x00007f8ab90006a2: cmp     0x18(%rbp),%r15
125    ;; jcc
126    0x00007f8ab90006a6: jne     0x00007f8ab90006e6
127    ;; get_thread
128    0x00007f8ab90006ac: push    %rax
129    0x00007f8ab90006ad: push    %rdi
130    0x00007f8ab90006ae: push    %rsi
131    0x00007f8ab90006af: push    %rdx
132    0x00007f8ab90006b0: push    %rcx
133    0x00007f8ab90006b1: push    %r8
134    0x00007f8ab90006b3: push    %r9
135    0x00007f8ab90006b5: push    %r10
136    0x00007f8ab90006b7: mov     %rsp,%r10
137    0x00007f8ab90006ba: and     $0xfffffffffffffff0,%rsp
138    0x00007f8ab90006be: push    %r10
139    0x00007f8ab90006c0: push    %r11
140    0x00007f8ab90006c2: mov     $0x1,%edi
141    0x00007f8ab90006c7: callq   0x00007f8ab90006c2
142    0x00007f8ab90006cc: pop     %r11
143    0x00007f8ab90006ce: pop     %rsp
144    0x00007f8ab90006cf: pop     %r10
145    0x00007f8ab90006d1: pop     %r9
146    0x00007f8ab90006d3: pop     %r8
147    0x00007f8ab90006d5: pop     %rcx
148    0x00007f8ab90006d6: pop     %rdx
149    0x00007f8ab90006d7: pop     %rsi
150    0x00007f8ab90006d8: pop     %rdi
151    0x00007f8ab90006d9: mov     %rax,%rbx
152    0x00007f8ab90006dc: pop     %rax
153    ;; cmpptr
154    0x00007f8ab90006dd: cmp     %rbx,%r15
155    ;; jcc
156    0x00007f8ab90006e0: je      0x00007f8ab9000763
157    ;; bind
158    ;; jcc
159    0x00007f8ab90006e6: je      0x00007f8ab9000763
160    ;; stop
161    0x00007f8ab90006ec: mov     %rsp,-0x28(%rsp)
162    0x00007f8ab90006f1: sub     $0x80,%rsp
163    0x00007f8ab90006f8: mov     %rax,0x78(%rsp)
164    0x00007f8ab90006fd: mov     %rcx,0x70(%rsp)
165    0x00007f8ab9000702: mov     %rdx,0x68(%rsp)
166    0x00007f8ab9000707: mov     %rbx,0x60(%rsp)
167    0x00007f8ab900070c: mov     %rbp,0x50(%rsp)
168    0x00007f8ab9000711: mov     %rsi,0x48(%rsp)
169    0x00007f8ab9000716: mov     %rdi,0x40(%rsp)
170    0x00007f8ab900071b: mov     %r8,0x38(%rsp)
171    0x00007f8ab9000720: mov     %r9,0x30(%rsp)
172    0x00007f8ab9000725: mov     %r10,0x28(%rsp)
```

```
173 0x00007f8ab900072a: mov    %r11,0x20(%rsp)
174 0x00007f8ab900072f: mov    %r12,0x18(%rsp)
175 0x00007f8ab9000734: mov    %r13,0x10(%rsp)
176 0x00007f8ab9000739: mov    %r14,0x8(%rsp)
177 0x00007f8ab900073e: mov    %r15,(%rsp)
178 0x00007f8ab9000742: movabs $0x7f8acf3a6250,%rdi
179 0x00007f8ab900074c: movabs $0x7f8ab90006ec,%rsi
180 0x00007f8ab9000756: mov    %rsp,%rdx
181 0x00007f8ab9000759: and    $0xffffffffffffffff,%rsp
182 0x00007f8ab900075d: callq  0x00007f8aceddb9fa
183 0x00007f8ab9000762: hlt
184 ;; bind
185 ;; r15
186 0x00007f8ab9000763: mov    -0x58(%rbp),%r15
187 0x00007f8ab9000767: mov    -0x50(%rbp),%r14
188 0x00007f8ab900076b: mov    -0x48(%rbp),%r13
189 0x00007f8ab900076f: mov    -0x40(%rbp),%r12
190 0x00007f8ab9000773: mov    -0x38(%rbp),%rbx
191 ;; ldmxcsr
192 0x00007f8ab9000777: ldmxcsr -0x60(%rbp)
193 ;; addptr
194 0x00007f8ab900077b: add    $0x60,%rsp
195 ;; pop
196 0x00007f8ab900077f: pop    %rbp
197 ;; ret
198 0x00007f8ab9000780: retq
199 ;; is_long:
200 0x00007f8ab9000781: mov    %rax,(%rdi)
201 0x00007f8ab9000784: jmpq   0x00007f8ab900069e
202 ;; is_float:
203 0x00007f8ab9000789: vmovss %xmm0,(%rdi)
204 0x00007f8ab900078d: jmpq   0x00007f8ab900069e
205 ;; is_double:
206 0x00007f8ab9000792: vmovsd %xmm0,(%rdi)
207 0x00007f8ab9000796: jmpq   0x00007f8ab900069e
208
```