

```

//下面是call_stub执行到call *%eax(包含此条汇编)的堆栈状态
//return_from_Java 是紧跟在call *%eax后面的那条指令的地址
// 偏移(相对于rbp)
//-----
//      [ return_from_Java ] <--- rsp
//      [ argument word n   ] <--- rsi
//      ...
// -N   [ argument word 1   ] //被调用的java方法的第一个参数, 如果参数的类型不是oop, 则直接存放参数值, 否则是oop地址
// -7 -1C [ Possible padding for stack alignment ]
// -6 -18 [ Possible padding for stack alignment ]
// -5 -14 [ Possible padding for stack alignment ]

// -4 -10 [ mxcsr save      ]
// -3 -C  [ saved rbx,      ]
// -2 -8  [ saved rsi       ]
// -1 -4  [ saved rdi       ]

// 0 0 [ saved rbp,        ] <--- rbp

// 1 4 [ return address    ]
// 2 8 [ ptr. to call wrapper ]
// 3 C [ result            ]
// 4 10 [ result_type       ]
// 5 14 [ method            ] <--- rbx
// 6 18 [ entry_point       ]
// 7 1C [ parameters        ]
// 8 20 [ parameter_size    ]
// 9 24 [ thread            ]

//每次method_entry_point_zerolocals都生成如下的堆栈
//begin-----
//下面是method_entry_point_zerolocals中由generate_fixed_frame生成的堆栈frame
//-----
// -8 -0x20 [ (%esp)        ] //reserve word for pointer to expression stack bottom
// -7 -0x1C [ 第一个字节码内存地址 ] //如果是invokeXXX指令, 那么就是这条invokeXXX指令在内存中的地址
// -6 -0x18 [ argument word 1 所在堆栈位置 ] <--- rdi
// -5 -0x14 [ ConstantPoolCache ]
// -4 -0x10 [ 0              ]
// -3 -0xC  [ method         ]
// -2 -0x8  [ 0              ] //如果是invokeXXX指令, 并有除方法参数之外的本地变量, 则这里存放本地变量的堆栈位置
// -1 -0x4  [ argument word n 所在堆栈位置 ]
// 0 0 [ saved rbp,          ] <--- rbp,
//-----

//      [ return_from_Java ] <--- rsp
//      [ 0                 ] 总共_max_locals - _size_of_parameters个0
//      ...
//      [ 0                 ]
//      [ 0                 ]
//end-----

//return_from_Java被移到上面了
//-----
//      [ argument word n   ] <--- rsi
//      ...
// -N [ argument word 1     ]
// -7 [ Possible padding for stack alignment ]
// -6 [ Possible padding for stack alignment ]
// -5 [ Possible padding for stack alignment ]
// -4 [ mxcsr save          ]
// -3 [ saved rbx,          ]
// -2 [ saved rsi           ]
// -1 [ saved rdi           ]
// 0 [ saved rbp,          ] <--- rbp
// 1 [ return address       ]
// 2 [ ptr. to call wrapper ]
// 3 [ result               ]
// 4 [ result_type          ]
// 5 [ method               ] <--- rbx
// 6 [ entry_point          ]
// 7 [ parameters           ]
// 8 [ parameter_size       ]
// 9 [ thread               ]

```