[讨论] 请教：Java 字节码如何执行的
JianLeiXing 2014-04-22
JVM在使用解释器执行过程中，执行Java的某个方法最终会调用如下的函数:

```
StubRoutines::call_stub()(
        (address)&link,
        // (intptr_t*)&(result->_value), // see NOTE above (compiler problem)
        result_val_address,          // see NOTE above (compiler problem)
        result_type,
        method(),
        entry_point,
        args->parameters(),
        args->size_of_parameters(),
        CHECK
      );
```

其中StubRoutines::call_stub() 会返回一个CallStub类型的函数指针(如下),该指针的值实际是StubRoutines::_call_stub_entry的值

```
typedef void (*CallStub)(
    address   link,
    intptr_t* result,
    BasicType result_type,
    methodOopDesc* method,
    address   entry_point,
    intptr_t* parameters,
    int       size_of_parameters,
    TRAPS
  );
```

StubRoutines：：_call_stub_entry 的初始化在如下：

```
**************************************************************************
init_globals
↓
stubRoutines_init1
↓
......
↓
generate_initial(){
......
StubRoutines::_call_stub_entry = generate_call_stub(StubRoutines::_call_stub_return_address);
......
}
**************************************************************************
address generate_call_stub(address& return_address) {
    StubCodeMark mark(this, "StubRoutines", "call_stub");
    address start = __ pc();
...
    // stub code
    __ enter();
    __ movptr(rcx, parameter_size);              // parameter counter
    __ shlptr(rcx, Interpreter::logStackElementSize); // convert parameter count to bytes
    __ addptr(rcx, locals_count_in_bytes);       // reserve space for register saves
    __ subptr(rsp, rcx);
    __ andptr(rsp, -(StackAlignmentInBytes));    // Align
...
  return start;
}
```

问题1：generate_call_stub 方法实现的一堆汇编指令是如何作为CallStub类型的方法的实现？
问题2：CallStub方法中的entry_point的地址是不是method的字节码起始地址。

RednaxelaFX 2014-04-23
问题1：啥都不用做。把那段机器码的起始地址当作指针，强制转换为CallStub类型就完事了。
参见这个例子：http://rednaxelafx.iteye.com/blog/428721
问题2：传入CallStub的entry_point是method->from_interpreted_entry()。这不是“字节码”起始地址，而是解释器的方法入口处理函数。
对多数还在解释执行的普通Java方法来说，这会指向解释器的zerolocals_entry。
请参考HLLVM群组之前两帖，
http://hllvm.group.iteye.com/group/topic/39806 (JIT编译以及执行native code的流程)
http://hllvm.group.iteye.com/group/topic/37707 (java_main的汇编入口在哪里)