

有道云链接：： [http://note.youdao.com/noteshare?](http://note.youdao.com/noteshare?id=04f6a605d8a2d6de40aa2a968788cf23&sub=BD1325CB132543689893A432DD79587D)

[id=04f6a605d8a2d6de40aa2a968788cf23&sub=BD1325CB132543689893A432DD79587D](http://note.youdao.com/noteshare?id=04f6a605d8a2d6de40aa2a968788cf23&sub=BD1325CB132543689893A432DD79587D)

书、编译环境：链接：[https://pan.baidu.com/s/1tR3gMy\\_59YRekqX915AjsQ](https://pan.baidu.com/s/1tR3gMy_59YRekqX915AjsQ) 提取码：ziya

手写JVM代码：<https://gitee.com/luban-ziya/ziya-jvm-open>

【2.16】一、Hotspot初探：用Java实现JVM框架

【2.18】二、Hotspot初探：深入理解内存模型与GC

【2.20】三、Hotspot初探：深入理解多线程

一、Hotspot初探：用Java实现JVM框架

1、对Hotspot、计算机有个整体上的认识

2、可以用java手写JVM

1、如何搭建单步调试openjdk环境

2、openjdk目录结构

oops

研究Hotspot源码的节奏

1、语言基础：c、c++，jit 汇编、硬编码、编译原理 劝退

2、类加载器

2、面向对象

类加载器加载

ClassFileParser::parseClassFile

生成两个klass

创建对象 oop

InstanceKlass::allocate\_instance

执行引擎

实现思路

基于栈运行的

栈的实现思路

内存模型

gc

3、Java的类与对象在JVM中的存在形式

klass pointer JVM层面

class java

根基

面向对象

HSDB

查看JVM内部

静态属性

非静态属性

JVM的堆栈

局部变量的存储

证明

C++的顶峰

4、JVM是如何运行Java程序的

5、如何实现虚拟机栈

6、如何实现执行引擎

7、单步调试演示手写JVM框架

clas文件

字节码文件

字节码增强

agent

jvm

Hotspot

java虚拟机

Hotspot java虚拟机

jdk 基于Hotspot实现的java语言的基础

openjdk = jdk + Hotspot

clion 主要用这个

netbeans

Ubuntu 图像界面

centos 服务器

unix

-----  
Windows

mac

>11

编译

make clion不认识 无法解析

netbeans 认识make

cmake 是这样

make=>cmake

main

main

启动流程

jdk开始启动的 来启动Hotspot

Hotspot

找到native方法对应的源码

new Thread().start start0

面向对象

类、对象

类 class Stu

元空间 类的元信息 klass实例 new klass

klass模型：java的类在jvm中的存在形式

是不是数组

instanceKlass 类的元信息

Class对象是在堆区

java类

生成两个klass

instanceKlass 存储类的元信息：public、属性、方法、继承

instanceMirrorKlass java.lang.Class对象

new Object

Hotspot内部 new Oop

内存编织

填充对齐

padding

精通属性到底存储在哪里

证明

ArrayKlass

基本类型 int[] TypeArrayKlass

引用类型 String[] ObjArrayKlass

oop模型：java的对象在JVM中的存在形式

markOop mark word

设计很怪异

写法也很怪异

普通对象指针 ordinary object pointer

double占两个slot

实现上要考虑哪些东西

double 10 正常的情况 在内存中如何存储

jvm中如何存储

做个对比

加深理解

人为拆的

很影响性能

java虚拟机规范的要求

64 8 32位有限

32bit

历史遗留

32bit 栈的每个slot 4B

64bit 8B

double 10

8B 16B

拆

合

内存同步

线程安全

三大特性

HSDB

查看klass、oop、静态属性、非静态属性、栈

如何用java实现jvm

栈的实现方式

执行的实现方向

知道

不知道

记忆型

理解型

刻意练习

如何查看klass

直接引用

内存地址

间接引用

静态常量池项的index

静态常量池 klass结构

在堆区

0x00000007c0060830 Main类对应的klass实例

研究Hotspot源码的节奏

C、C++

编译器的不同

C语言 gcc

c++ g++

hotspot

c++

.h

.cpp

.hpp .h + .cpp

定义与实现分开

c

硬编码

.h

.c

.h

.c

.c++

.hpp

.cpp

手写JVM

Java的多线程是如何基于OS的线程实现的

new Thread oop

JavaThread

OSThread

pthread\_create

知道

不知道

JVM是如何运行Java程序的

类加载器

SPI、类加载机制 pass

- 1、把硬盘上的字节码文件读入内存
- 2、解析生成元信息klass (Class对象) instanceKlass
- 3、找到main方法
- 4、执行字节码指令

栈

执行引擎

执行引擎 执行字节码指令的

字节码解释器

c++实现的

hotspot\src\share\vm\interpreter\bytecodeInterpreter.cpp

```
while() {  
char c =...  
switch() {  
case ldc:  
break;  
case new:  
break;  
}  
}
```

模板解释器

硬编码实现的

Hotspot

jit 代码块 优化单位

基于栈去运行的

cpu有很多寄存器

OS的栈

栈顶 esp

栈顶

Stack stack

mov r8, esp

mov esp, [stack] 切到自实现的栈

....

mov esp, r8

核心：切esp寄存器

-----0x22 esp = 0x22

add

-----0x11 ebp = 0x11

-----0x10 esp = 0x10

main

-----0x1 ebp = 0x1

栈的实现思路 听不懂没关系

移花接木的实现方式（我自己实现的jvm是这种方式 简单、好理解）

1、如何关联

2、在你自己实现的栈用能用CPU的指令

push

pop

一种方式

切栈

iload 10

new

c++层面

java去实现，栈实现起来很简单

把栈跟线程绑定起来

拓展式（难，不好理解）原生程序是如何基于系统栈去运行的

研究jvm的运行时，两个非常重要的流程分支

call\_stub

entry\_point

java -> jvm <- java 桥梁

java | jvm

os跑一个程序，栈长啥样子

OD 逆向分析工具

内存模型

java jvm自己的

myjvm

Hotspot虚拟机上的

java自举

jvm自举