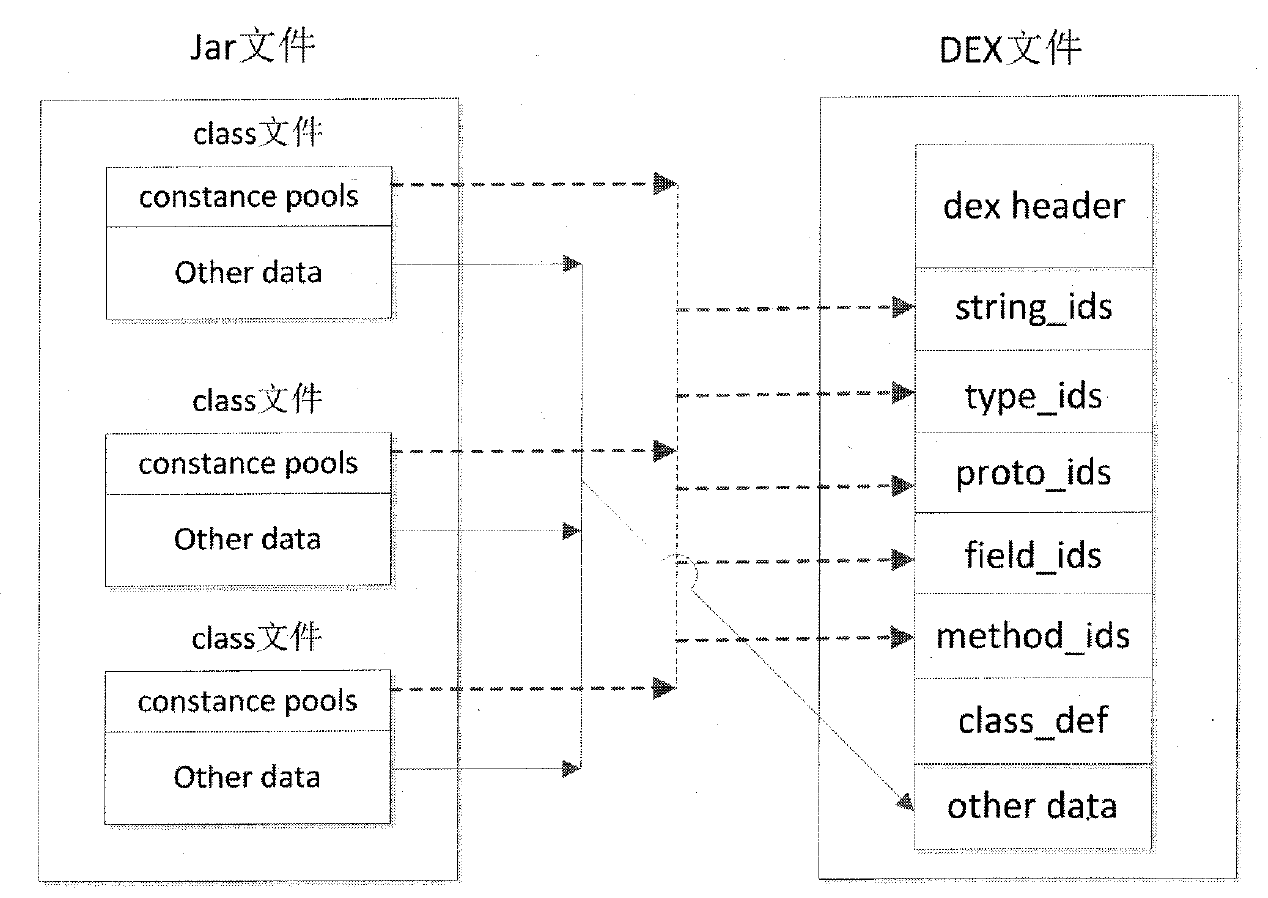
Android原生程序:动态链接库和普通可执行程序，其中普通可执行程序可以直接调试，而动态链接库需要Android程序加载后再远程附加。

# Dalvik虚拟机与Jvm虚拟机

## 二者字节码不同

## Dalivk可执行文件体积更小



设置共享常量池，不同calss中的相同常量数值被合并，所以dex体积更小

## 架构不同

Jvm基于栈，Dalvik基于寄存器

示例代码:

|  |
| --- |
| public class Hello {  public int foo(int a, int b){  return (a + b) \* (a - b);  }  public static void main(String arg[]) {  Hello hello = new Hello();  System.out.println(hello.foo(5,3));  }  }  // 编译命令  // javac Hello.java  // dx –dex –output=Hello.dex Hello.class  // 编译过程可能产生的问题：unsupported class file version 52.0  // 原因：java版本与android sdk版本不适配  // 我的java版本是1.8，android sdk的版本29即可  // 反编译.class文件  // javap –c –classpath . Hello  // -c 反编译 –classpath 指定要反编译的class所在的目录 |

### public、protected、private、default的区别

public：任何类都可使用

protected：保护派生类，其子类可以使用

default：针对本包

private：私有类，只有在类中才可以使用

### static关键词

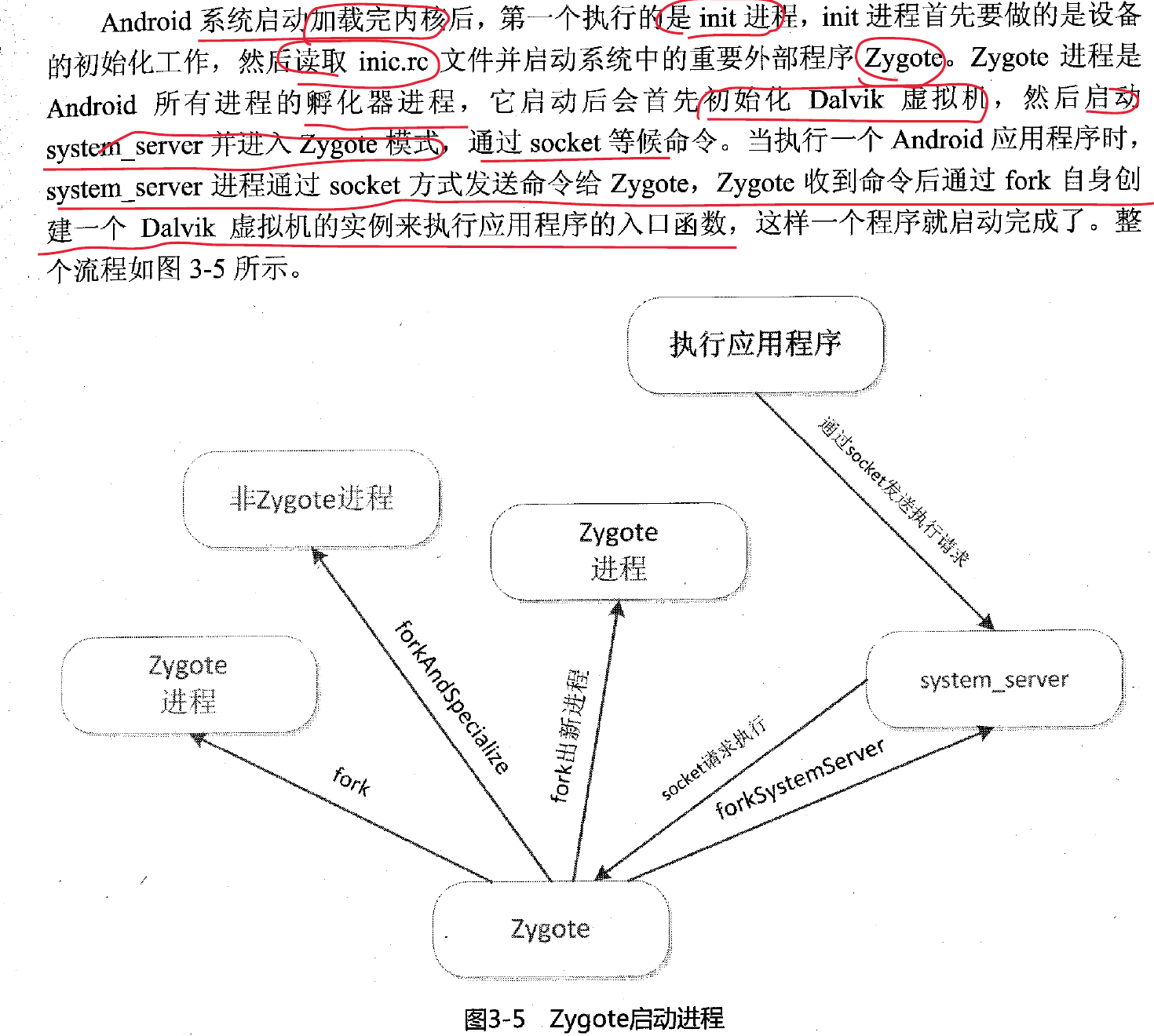
方便在没有创建对象的情况下调用方法/变量；

也就是说，静态成员方法/变量不存放在对象的内存中，

对应的C++中，它也不存放在对象的虚表中？那么静态成员方法/变量存放在哪里？

静态成员方法一样存放在text段，但是在调用它之前，没有this指针入参，所以静态成员方法无法访问类的非静态成员方法(非静态成员方法中可能会访问非静态成员变量)和非静态成员变量。

### Android系统启动机制



因为Xposed是通过Hook Zygote进程来达成目标的，所以Xposed需要热启动。

而Frida使用二进制动态插桩技术，对可执行程序没有永久改变。对Dalvik的Hook与Xposed类似，将java函数变成native函数。

Firda的两种模式：

1. attach，利用ptrace
2. spwan，新启动1个进程，在应用进程启动前Hook