**AVMDBG说明**

Avmdbg的目标是打造一款轻量级的的安卓smali代码调试器，实现方式为Python扩展接口，可以通过编写python脚本实现安卓app的快速调试，目前测试版本主要提供以下API：

1. bool attach(*string*& processName);

说明: 附加到目标进程

debugger = AvmDebugger()  
**if not** debugger.attach("com.example.x0r.demo"):  
 **print** "[err]: process attach fail!"

1. void waitLoop();

说明: 循环等待调试事件

1. bool setBreakPoint(py::dict& breakPoint);

说明: 设置断点事件

参数: 字典类型

bpInfo1 = {  
 "class": "Lcom/example/x0r/demo/LoginActivity;",  
 "method": "testStackFrames1",  
 "sign": "(BIJLjava/lang/String;)V",  
 "index": 0x0f,  
 "registers": 14,  
 "callback": breakPointCallback  
}

class: 目标类名

method: 函数名称

sign: 函数标签 (参数与返回值类型说明)

index: 断点代码偏移 (使用baksmali的-f参数获取，JEB反汇编显示的偏移有问题)

registers: 函数使用的寄存器个数 (一般在反汇编代码的头部可以找到)

callback: 断点事件的回调函数 (可以在此回调函数中输出变量参数或者堆栈)

关于方法命名方式可参考: <https://github.com/JesusFreke/smali/wiki/TypesMethodsAndFields>

1. py::list getStackFrames(ObjectId threadId);

说明: 获取当前线程堆栈，参数为当前触发事件后暂停的线程ID，可在断点回调函数中使用

threadId = context["threadId"]  
stackFrames = debugger.getStackFrames(threadId)  
**for** frame **in** stackFrames:  
 **print** frame["class"] + " :: " + \  
 frame["method"] + " -> " + \  
 frame["sign"] + " - " + \  
 hex(frame["index"])

1. py::list getObjectFieldValues(ObjectId objectId);

说明: 获取对象的成员变量信息，参数为对象ID

objid = context["param"]["p0"]["id"]  
re = debugger.getObjectFieldValues(objid)

1. py::dict getArrayObjectValue(ObjectId objectId);

说明: 获取数组类型变量的信息

1. py::dict getRegisterValue(py::dict& Context, *string*& regName, u1 varType);

说明: 获取寄存器变量的值，需要指定寄存器名称(V命名或P命名法)以及变量类型

**print** debugger.getRegisterValue(context, "p0", JdwpTag.OBJECT)  
**print** debugger.getRegisterValue(context, "v1", JdwpTag.LONG)

附加: 可参考<https://github.com/JesusFreke/smali/wiki/Registers>

1. Dalvik虚拟机的寄存器都是32位的，64位类型(Long/Double)用2个相邻寄存器表示
2. 假设一个函数使用到M个寄存器，并且该函数有N个参数，根据Dalvik虚拟机参数传递方式中的规定：参数使用最后的N个寄存器，局部变量使用从v0开始的前(M-N)个寄存器。(注意参数占用2个寄存器的情况)
3. v命名法采用以小写字母“v”开头的方式表示函数中用到的局部变量与参数
4. p命令法对函数的局部变量寄存器命名没有影响，它的命名规则是：函数中引入的参数命名从p0开始，依次递增。非静态成员函数中，p0用于this指针。
5. py::str getStringValue(ObjectId objectId);

说明: 获取字符串String类型变量的值

val = context["param"]["p3"]["value"]  
**print** debugger.getStringValue(val["data"][0])  
**print** debugger.getStringValue(val["data"][1])