Atitti.java android反编译解决方案-----虚拟机方案

哈哈，终极解决方案是虚拟机。。。c++也可以反编译为汇编代码，但无需担心，因为读懂汇编太麻烦。。只要不能拿到c++源码就可。。。

java可以使用虚拟机方式，使用汇编模式编写，真正的代码其实在 不外传的文档里面...如果对方反编译，只能得到类似于汇编的代码。。

作者::  ★(attilax)>>> 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

AtiVM vm = new AtiVM();

vm.getstatic(CRuntime.INSTANCE);// 获取指定类或者接口的静态域，并将其值压入栈顶

vm.push(null); // 将null推送至栈顶

vm.invokeinterface(CLibraryInterface.class, "time", 2);// 调用接口方法

// ，从栈顶取出俩个数作为参数，把返回值放入栈顶

vm.push(631123200);

vm.sub();// sub 将栈顶两int型数值相减并将结果压入栈顶

vm.push(850348800l);//

vm.cmp(); // cmp 比较栈顶两long型数值大小，并将结果（1，0，-1）压入栈顶

if (vm.ifle()) // ifle 当栈顶int型数值小于等于0时跳转

{

vm.getstatic(System.out);

vm.push("ok");

vm.invokevirtual(PrintStream.class, "println"); //调用实例方法

vm.ret(); // return

} else {

// vm.new0(RuntimeException.class);

vm.push("ex1");

vm.invokespecial(RuntimeException.class); // 调用超类构造方法，实例初始化方法，私有方法

vm.athrow(); // 将栈顶的异常抛出

}

System.out.println("---finish");

}

也能使用classloader做个bejyez ....中间容易的点儿..