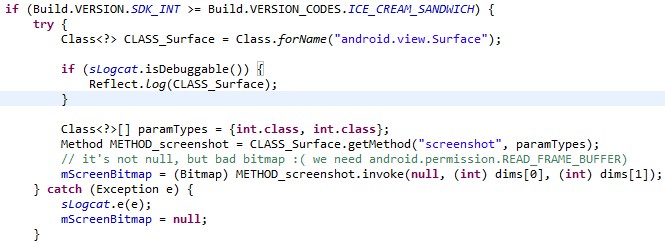
Android 系统权限获取

为了实现ScreenCaptureSerivce，经过查阅android官方API发现android 4.0以上版本提供了截屏功能的调用接口。不过，要获得这一接口，我们必须获得系统权限：android.permission.READ\_FRAME\_BUFFER。



在讨论如何获得系统权限之前，笔者认为有必要先了解一下Android的权限规则。

**Android的权限规则**

（1)Android中的apk必须签名

这种签名不是基于某一权威证书的，它是一种自签名证书。需要注意的是，有的android系统权限是基于签名的。比如：system等级的权限有专门对应的签名，如果签名不对，权限就获取不到。默认生成的APK文件是debug签名的。

（2）基于UserID的进程级别的安全机制

进程有自己独立的地址空间，进程与进程之间默认是不能互相访问的，Android通过为每一个apk分配唯一的linux userID来实现这一约束（众所周知，Android系统是运行在Linux内核上的），其名称为"app\_"加一个数字。

Android提供了如下所述的一种机制，可以使两个apk打破前面讲的这种约束。

在AndroidManifest.xml中，我们可以利用sharedUserId属性给不同的package分配相同的userID，这样，两个package会被当做同一个程序，从而系统会给两个程序分配相同的UserID。当然，基于安全考虑，两个apk需要具有相同的签名，否则，没有验证也就没有意义了。

**如何获得Android系统权限**

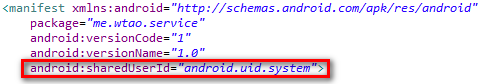
一般情况下，android APP开发者不允许获得系统权限，从而不能得到具有系统权限的android apk。但是，为了功能的实现，笔者不得不去探索解决这一问题的方法。

研究发现，从系统签名的方向做一些操作，或许能够使得android apk获得系统权限。

具体来说，我们要获得系统权限的话，使用系统签名的方法，有两种做法：

方法一：

* 在程序的AndroidManifest.Xml中的manifest节点中加入android:sharedUserId="android.u id.system"这个属性。



* 修改Android.mk文件，加入LOCAL\_CERTIFICATE := platform。
* 在Android系统源码的环境下，使用mm命令来编译，生成的apk就有系统权限了。

方法二：

* 加入android:sharedUserId="android.uid.system"这个属性。
* 使用eclipse编译出apk文件。
* 使用目标系统的platform密钥来重新给apk文件签名。具体做法如下：
  + 首先找到密钥文件。先进入 Android 源码目录 F:\android\_source\build\target\product\security，找到 platform.pk8和platform.x509.pem两个文件。
  + 然后用Android提供的Signapk工具来签名。signapk的源代码是在"build/tools/signapk"下，编译后在“out/host/linux-x86/framework"下，用法为"java -jar signapk.jar platform.x509.pem platform.pk8 input.apk output.apk"。其中input.apk是指我们要签名的原始文件apk，后面output.apk是指我们签名后输出的apk文件的名称。

加入android:sharedUserId="android.uid.system"这个属性，通过共享User id,拥有同一个User id的多个APK可以配置运行在同一个进程中。那么把程序的UID配成android.uid.system，也就是要让程序运行在系统进程中，这样就拥有系统权限了吗？答案是否定的，因为加入UID还远远不够。如果这时候安装APK的话，会发现无法安装，提示签名不符，原因是程序想要运行在系统进程中还要有目标系统的platform key，就是上面第二个方法提到的platform.pk8和platform.x509.pem两个文件。用这两个key签名后，apk才真正可以放入系统进程中。第一个方法中加入LOCAL\_CERTIFICATE := platform其实也就是用这两个key来签名。

但不幸的是，最终发现了一个致命的问题，就是这样生成的程序只有在原始的Android系统或者是自己编译的系统中才可以运行。因为这样的系统才可以拿到platform.pk8和platform.x509.pem两个文件，要是在别的Android系统上连安装都安装不了。试试原始的Android中的key来签名，程序在模拟器上运行OK，只不过放到G3上安装的话，直接提示"Package ... has no signatures that match those in shared user android.uid.system"。其实，这样也是为了保护了系统的安全。

所以，最终来说，通过这种方法，我们不能解决，或者说是不能够完美的解决实现截屏功能这一问题。

Reference：http://m.blog.csdn.net/blog/xpp1993/11247173