**如何引入leakcanary检测内存泄漏**

**---daniel.nie 20170525**

1. **下载最新的leakcanary源码：**[**https://github.com/square/leakcanary**](https://github.com/square/leakcanary)
2. **编写mk脚本，重新组合代码，保证编译通过，其中leakcanary依赖一个haha库，也需要下载下来：**[**https://github.com/square/haha**](https://github.com/square/haha)

****

1. **修改需要引入的工程的mk文件，将leakcanary作为一个lib库编进自己的工程**

**-**

**-LOCAL\_PROGUARD\_FLAG\_FILES := proguard.flags**

**+LOCAL\_PROGUARD\_ENABLED := disabled**

**+LOCAL\_DX\_FLAGS := --multi-dex**

**+LOCAL\_JACK\_FLAGS := --multi-dex native**

**+#LOCAL\_PROGUARD\_FLAG\_FILES := proguard.flags**

**LOCAL\_AAPT\_FLAGS := --auto-add-overlay \**

**- --extra-packages android.support.v7.preference:android.support.v14.preference:android.support.v17.preference:android.support.v7.appcompat:android.support.v7.recyclerview**

**+ --extra-packages android.support.v7.preference:android.support.v14.preference:android.support.v17.preference:android.support.v7.appcompat:android.support.v7.recyclerview:com.squareup.leakcanary**

**ifneq ($(INCREMENTAL\_BUILDS),)**

**LOCAL\_PROGUARD\_ENABLED := disabled**

**@@ -76,6 +81,7 @@ endif**

**include frameworks/opt/setupwizard/library/common-full-support.mk**

**include frameworks/base/packages/SettingsLib/common.mk**

**+include packages/apps/Settings/leakcanary/common.mk **

**其中要注意的事项是，6.0编译ok, 在7.1 jack编译，要把混淆关闭，或者编译报错。**

1. **编译通过后，恭喜你，可以修改代码，让leakcanary运行起来了，新增MyosApplication,修改BaseFragment, BaseActivity不用加，默认会自动检测。**

**，修改AndroidManifest.xml，引入application和声明需要的service及activity。**

1. **打包编译后，运行apk，那就可以自由操作，有leak 会有通知弹出。查看通知内容或者通过log，找到内存泄漏的元凶，修改代码。**