2015/12/29 ldy

StrictMode介绍与使用

# 简单介绍

StrictMode意思为严格模式，是用来检测程序中违例情况的开发者工具。

分为两类，

## ThreadPolicy

可以检测指定线程（一般为主线程）的耗时操作，网络操作，io操作等。

## VmPolicy

主要检测内存泄露相关，指定实例数量，如果超过即出现泄露，因为activity在内存中正常只有一个，如果超过即为泄露。还可以检测数据库操作，流操作等是否close()。

# 具体函数

## ThreadPolicy

ThreadPolicy:线程策略检测的内容有

自定义的耗时调用 使用 **detectCustomSlowCalls()** 开启

磁盘读取操作 使用 **detectDiskReads()** 开启

磁盘写入操作 使用 **detectDiskWrites()** 开启

网络操作 使用 **detectNetwork()** 开启

## vmpolicy

VmPolicy: 虚拟机策略检测的内容有

Activity泄露 使用 **detectActivityLeaks()** 开启

未关闭的Closable对象泄露 使用 **detectLeakedClosableObjects()** 开启

泄露的Sqlite对象 使用 **detectLeakedSqlLiteObjects()** 开启

检测实例数量 使用 **setClassInstanceLimit()** 开启

# 程序展示

StrictMode.*setThreadPolicy*(**new** StrictMode.ThreadPolicy.Builder()  
 .detectAll()  
 .penaltyLog()  
 .build());  
StrictMode.*setVmPolicy*(**new** StrictMode.VmPolicy.Builder()  
 .detectActivityLeaks()  
 .penaltyLog()  
 .build());

Detect用于指定监视种类，上面介绍过。

Penalty用于指定发现后的惩罚，可以是日志打印，弹出对话框亦或者崩溃等，这里用的是日志。

# 注意

其日志打印出的耗时时间可能不准确，以实际测试出的为准，且开启严格模式会对性能造成一定影响，且其引入是从API9开始，部分方法是从API11开始的。

# 进阶知识

## 自己指定耗时操作，指定实例数量

### 指定耗时操作

private static long SLOW\_CALL\_THRESHOLD = 500;

public void executeTask(Runnable task) {

long startTime = SystemClock.uptimeMillis();

task.run();

long cost = SystemClock.uptimeMillis() - startTime;

if (cost > SLOW\_CALL\_THRESHOLD) {

StrictMode.noteSlowCall("slowCall cost=" + cost);

}

}

其中如果run时间大于500，则发出提示，此时如果调用detectCustomSlowCalls即可收到惩罚。

### 指定实例数量

StrictMode.setVmPolicy(new VmPolicy.Builder().setClassInstanceLimit(SearchBox.class, 1).penaltyLog().build());

即为SearchBox只能有一个实例

## 推荐使用方式

if (IS\_DEBUG && Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.GINGERBREAD) {

StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder().detectAll().penaltyLog().build());

StrictMode.setVmPolicy(new VmPolicy.Builder().detectAll().penaltyLog().build());

}

只在debug模式下使用，不再release版本中使用，同时，严格模式自API 9 开始引入，某些API方法也从 API 11 引入。使用时应该注意 API 级别。

# 参考

## <http://droidyue.com/blog/2015/09/26/android-tuning-tool-strictmode/?hmsr=toutiao.io&utm_medium=toutiao.io&utm_source=toutiao.io&utm_source=tuicool&utm_medium=referral>