

APP Monkey 压力测试

1、环境搭建

- 去Java官网下载**JDK1.8** (推荐这个), 安装
- 配置java 环境变量, 配置完再终端输入**java -version** 或者 **java** 不报错即代表配置成功
- 去官网下载 **Android SDK**, 安装并配置好android sdk环境变量, 配置完在终端输入adb 或 adb shell monkey 不报错即代表配置成功

2、什么是Monkey

顾名思义, Monkey就是猴子, Monkey测试, 就像一只猴子, 在电脑面前, 乱敲屏幕在测试。

我们可以通过Monkey程序模拟用户触摸屏幕、滑动Trackball、按键等操作来对设备上的程序进行压力测试, 检测程序多久的时间会发生异常

3、Monkey用来做什么

Monkey 主要用于Android 的压力测试 自动的一个压力测试小工具, 主要目的就是为了测试app 是否会Crash.

4、Monkey 命令启动方式

- 可以通过PC机CMD窗口中执行: adb shell monkey {+命令参数} 来进行Monkey测试
- 在PC上adb shell 进入Android系统, 通过执行 monkey {+命令参数} 来进行Monkey 测试
- 在Android机或者模拟器上直接执行monkey 命令, 可以在Android机上安装Android终端模拟器 (Terminal Emulator for Android)

5、获取APP应用包名方式

- 获取哪个app包名, 就打开哪个app
- 在终端命令行输入以下命令

```
1.Mac/Linux: adb shell dumpsys window windows | grep mFocusedApp
2.Windows :   adb shell dumpsys window windows | findstr mFocusedApp
```

```
示例: mFocusedApp=AppWindowToken{2194784 token=Token{723a5d8
ActivityRecord{fdd5cbb u0 com.tingrongyi.easytjdz/.activity.MainActivity
t7170}}}
包名: com.tingrongyi.easytjdz
```

6、Monkey输出日志

```
adb shell monkey -p cn.goapk.market 100 > 路径/log.txt
```

7、Monkey基本参数介绍

- -p 参数 <允许的包名列表>

用此参数指定一个或多个包。指定包之后，monkey将只允许系统启动指定的app。如果指定包，monkey将允许系统启动设备中的所有app。

指定一个包：adb shell monkey -p cn.goapk.market 100

指定多个包：adb shell monkey -p fishjoy.control.menu -p cn.goapk.market 100

- -v 参数

用于指定输出日志级别（级别就是日志的详细程度），总共分3个级别，分别对应的参数如下表所示：

Level 0：adb shell monkey -p cn.goapk.market -v 100 // 缺省值，仅提供启动提示、测试完成和最终结果等少量信息

Level 1：adb shell monkey -p cn.goapk.market -v -v 100 // 提供较为详细的日志，包括每个发送到Activity的事件信息

Level 2：adb shell monkey -p cn.goapk.market -v -v -v 100 // 最详细的日志，包括了测试中选中/未选中的Activity信息

- -s（随机数种子）

用于指定伪随机数生成器的seed值，如果seed相同，则两次Monkey测试所产生的事件序列也相同的。示例：

monkey测试1：adb shell monkey -p cn.goapk.market -s 10 100

monkey测试2：adb shell monkey -p cn.goapk.market -s 10 100

- --throttle <毫秒>

用于指定用户操作（即事件）时间的时延，单位是毫秒；如果指定这个参数，monkey会尽可能快的生成和发送消息。示例：adb shell monkey -p cn.goapk.market --throttle 3000 100

- --ignore-timeouts

通常用于应用程序发生任何超时错误（如“Application Not responding”对话框）Monkey将停止运行，设置此项，Monkey将继续发送事件给系统，直到事件计数完成。示例：adb shell monkey -p cn.goapk.market --ignore-timeouts 100

8、Monkey日志分析

- 正常情况

- 如果Monkey测试顺利执行完成，在log的最后，会打印出当前执行事件的次数和所花费的时间；// Monkey finished 代表执行完成

- Monkey执行中断，在log的最后也能查看到当前已执行的次数。Monkey执行完成的log具体如下：

```
Events injected: 6000
:Dropped: keys=0 pointers=9 trackballs=0 flips=0
## Network stats: elapsed time=808384ms (0ms mobile, 808384ms wifi,
0msnot connected)
// Monkey finished
```

- 异常情况：在日志中搜索以下关键字
 - 无响应问题可以在日志中搜索“ANR”。
 - 崩溃问题搜索“CRASH”。
 - 内存泄露问题搜索“GC”（需进一步分析）
 - 异常问题搜索“Exception”（如果出现空指针，NullPointerException，需格外重视）。

```
// event type          // Event percentages:

--pct-touch <percent>    // 0: 15.0%
调整触摸事件的百分比(触摸事件是一个down-up事件，它发生在屏幕上的某单一 位置)。

--pct-motion <percent>   // 1: 10.0%
调整动作事件的百分比(动作事件由屏幕上某处的一个down事件、一系列的伪随机事 件和一个up事件组
成)。

--pct-pinchzoom <percent> // 2: 2.0%
缩放事件百分比

--pct-trackball <percent> // 3: 15.0%
调整轨迹事件的百分比(轨迹事件由一个或几个随机的移动组成，有时还伴随有点击)。

屏幕旋转事件百分比      // 4: -0.0%

--pct-nav <percent>      // 5: 25.0%
调整“基本”导航事件的百分比(导航事件由来自方向输入 设备的up/down/left/right组成)。

--pct-majornav <percent> // 6: 15.0%
调整“主要”导航事件的百分比(这些导航事件通常引发图 形界面中的动作，如：回退按键、菜单按键)

--pct-syskeys <percent>  // 7: 2.0%
调整“系统”按键事件的百分比(这些按键通常被保留，由 系统使用，如Home、Back、Start Call、End
Call及音量控制键)。

--pct-appswitch <percent> // 8: 2.0%
调整启动Activity的百分比。在随机间隔里，Monkey将执行一个startActivity()调 用，作为最大程
度覆盖包中全部Activity的一种方法。

--pct-flip <percent>     // 9: 1.0%
键盘翻转事件百分比
```

```
--pct-anyevent <percent> // 10: 13.0%
```

调整其它类型事件的百分比。它包罗了所有其它类型的事件，如：按键、其它不常用的设备按钮、等等。

附录1

- 可能会遇到的坑
 - 针对windows电脑，配置好环境变量，手机已连接好USB，并打开了开发者选项，在终端输入adb devices 查看设备，只显示 List of devices attached，无设备列表
 - 排查
 - 查看一下是不是电脑上没有Android手机驱动程序
 - 若是，下载一个PC端的豌豆荚，数据线连接上手机，就会自动安装驱动程序了。
- Mac 电脑虽然已经在.bash_profile 和.zshrc文件中配置好环境变量了，但是在终端敲adb 命令时依然报错 command not found
 - 解决方式:在.zshrc文件 中"# User configuration" 位置下面加入 "source ~/.bash_profile(注意：前提是bash_profile文件中的环境变量 你已经各种配置好了)
 - 还有一种，以上配置都OK， adb 命令还是无法识别，解决：

- 在./bash_profile 中添加 adb 的路径

```
export PATH=$PATH:$ANDROID_HOME/platform-tools
export PATH=$PATH:$ANDROID_HOME/platform-tools/adb
```

- 然后需要在./bash_profile 环境变量中添加 zsh 的环境变量配置

```
export PATH=/bin:/usr/bin:/usr/local/bin:$PATH
```

- 保存退出,并运行source ./bash_profile 使环境变量生效

```
source ./bash_profile
```

- 在.zshrc 文件中末尾添加source ./bash_profile

```
vi .zshrc

// 文件末尾添加
source ./bash_profile
```

附录2

1. windows monkey 测试环境搭建参考链接: <https://blog.csdn.net/zha6476003/article/details/79372684>
2. Mac monkey 测试环境搭建参考链接:https://blog.csdn.net/yuki_xiaobaomin/article/details/78236222