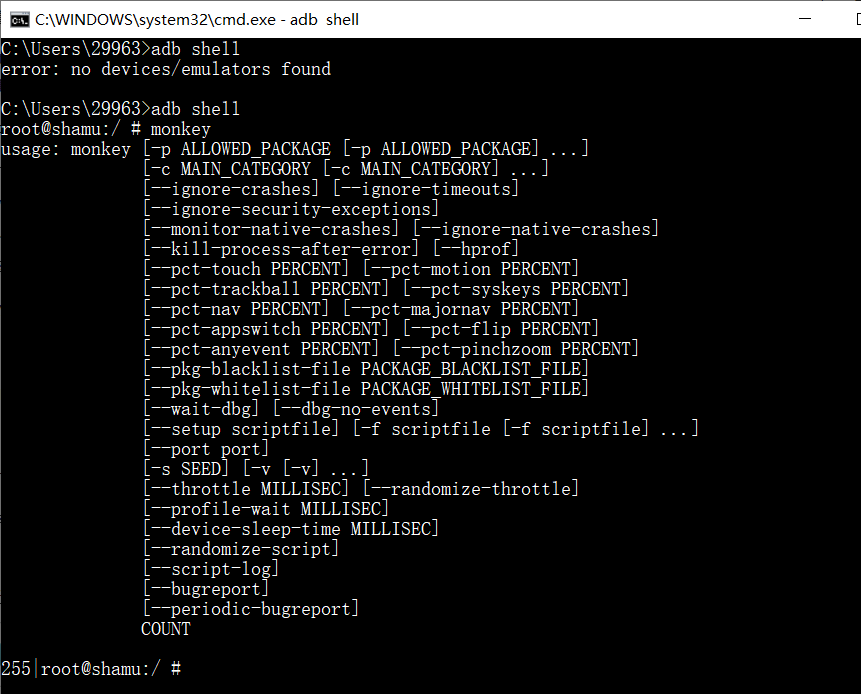
Monkey的使用

monkey是一个黑盒压力测试工具，采用随机算法，对被测应用随机发送伪随机的用户事件流，如点击，拖拽，滑动，输入等，来确定应用是否会发生异常，并且会产生日志来进行分析，但是测试数据测试事件都是随机的，有一定局限性。适合作为压力或者稳定性测试工具来测试应用的崩溃率，稳定性。但是可以通过二次开发来进行业务流程控制或一定程度上控制测试覆盖

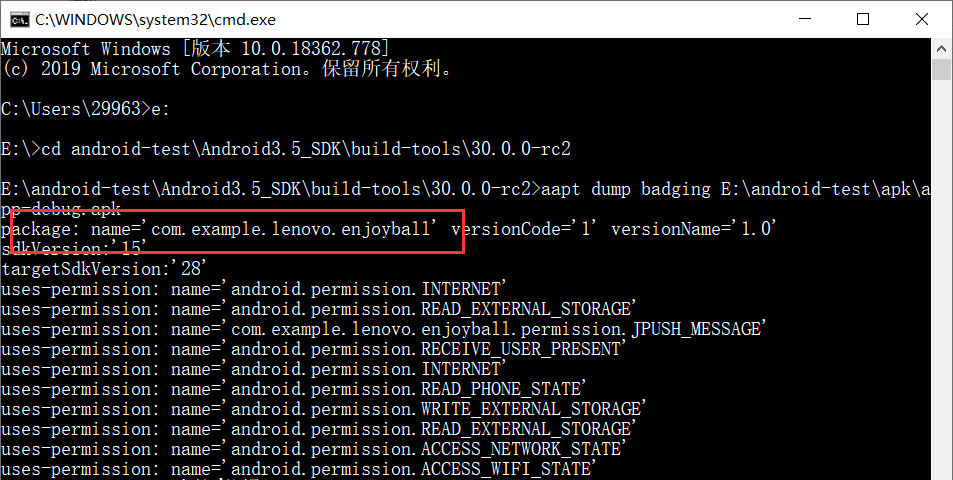
使用命令



到aapt路径下：E:\>cd android-test\Android3.5\_SDK\build-tools\30.0.0-rc2

获取包名： >aapt dump badging E:\android-test\apk\app-debug.apk

'com.example.lenovo.enjoyball



* 向系统发送1000 次随机事件，各个随机事件的时间间隔为500 毫秒，测试过程忽略安全、超时、崩溃等异常。

adb shell monkey -p com.example.lenovo.enjoyball --throttle 500 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore-native-crashes --monitor-native-crashes -v -v -v 1000 > E:\monkey.log

Monkey向命令行打印输出的log信息级别:

**默认级别0：**-v 只打印启动信息，测试完成信息和最终结果信息

**级别1：**-v -v 提供有关测试在运行时（例如发送到您的 Activity 的各个事件）的更多详细信息

**级别2**：-v -v -v 打印最详细信息,例如已选择或未选择用于测试的 Activity

Monkey日志保存方法：

adb shell monkey -p 包名 -v -v 100 >e:\monkeylog.txt  
在log的最开始都会显示Monkey执行的seed值、执行次数和测试的包名

* --throttle 在事件之间插入固定延迟。通过这个选项可以减缓Monkey的执行速度。如果不指定该选项，Monkey将不会被延迟，事件将尽可能快地被产成。
* --pct-touch 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 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-trackball 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)。
* --pct-nav 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 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：5-way键盘的中间按键、回退按键、菜单按键)
* --pct-syskeys 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 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-anyevent 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。
* --wait-dbg 停止执行中的Monkey，直到有调试器和它相连接。
* --monitor-native-crashes 监视并报告Android系统中本地代码的崩溃事件。

|  |  |  |
| --- | --- | --- |
| **分类** | **选项** | **说明** |
| 调试 | --dbg-no-events | 指定后，Monkey 将初始启动到测试 Activity，但不会生成任何其他事件。 为了获得最佳结果，请结合使用 -v、一个或多个软件包约束条件以及非零限制，以使 Monkey 运行 30 秒或更长时间。这提供了一个环境，您可以在其中监控应用调用的软件包转换操作。 |
| --hprof | 设置此选项，将在Monkey事件序列之前和之后立即生成profiling报告。这将会在data/misc中生成大文件(~5Mb)，所以要小心使用它 |
| --ignore-crashes | 通常，当应用程序崩溃或发生任何失控异常时，Monkey将停止运行。如果设置此选项，Monkey将继续向系统发送事件，直到计数完成 |
| --ignore-timeouts | 应用程序发生任何超时错误(如“Application Not Responding”对话框)时，Monkey将停止运行。如果设置此选项，Monkey将继续向系统发送事件，直到计数完成 |
| -ignore-security-exceptions | 当应用程序发生权限许可错误时，Monkey将停止运行。如果设置了此选项，Monkey将继续向系统发送事件，直到计数完成 |
| --ignore-native-crashes | 当应用发生底层C/C++代码引起的奔溃事件时，Monkey将停止运行。如果设置了此项，Monkey将继续向系统发送事件，直到计数完成 |
| --monitor-native-crashes | 监视并报告Android系统中Android C/C++引起的崩溃事件。如果设置了--kill-process-after-error，系统将停止运行 |
| --kill-process-after-error | 当Monkey由于一个错误而停止时，出错的应用程序将继续处于运行状态。当设置了此选项时，将会通知系统停止发生错误的进程。注意，当Monkey正常执行完毕，它不会关闭所有启动的应用，设备依然保留Monkey结束时的状态 |
| --wait-dbg | 启动Monkey后，先中断其运行，等待调试器附加上来 |

#头文件、控制monkey发送消息的参数

type=raw events

count=10

speed=1.0

#以下为monkey命令

start data >>

DispatchPress(KEYCODE\_HOME)

LaunchActivity(com.example.apptest2,com.example.apptest2.MainActivity)

ProfileWait()

Tap(400,200)

DispatchPress(KEYCODE\_1)

ProfileWait()

Tap(400,320)

DispatchPress(KEYCODE\_2)

ProfileWait()

DispatchPress(66)

DispatchPress(66)

ProfileWait()

Tap(400,200)

DispatchPress(67)

DispatchPress(67)

Tap(400,320)

DispatchPress(67)

DispatchPress(67)

DispatchPress(KEYCODE\_HOME)

#Tap(100,400)

#ProfileWait()