**三、Monkey测试的日志分析**

完成Monkey测试之后，如何开展Monkey测试中日志分析和内存泄露分析？

**1. 一般的场景是这样的：**

1）启动Monkey之前，打开APP，使用Andorid device monitor dump 一个内存“快照”（hprof 文件）出来；

2）根据测试策略，运行monkey测试；

3）再次使用Andorid device monitor dump 一个hprof 文件出来；

4）使用MAT（Memory Analyze Tool）打开两个hprof文件，比较两个的内存占用情况，如果后一个内存占用更多，区分PSS 及 USS的内存占用情况，如果持续增长，肯定是有泄露的。或封装autoMonkey记录内存情况，结束时自动生成曲线图；如图5所示：

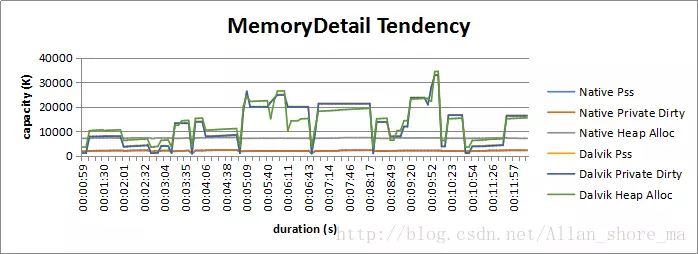


图5 内存趋势详情

5）针对hprof文件，看下到底是哪些对象更多，占用的内存更大。

**2. Monkey日志分析**

**--------------------打赏⬇小编--------------------------------------打赏⬆小编------------------**

Monkey日志分析是Monkey测试中非常重要的一个环节。通过日志分析，可以获取APP在测试过程中是否发生异常，以及发生的频率。同时，还可以获取对应的错误信息，帮助开发定位和解决问题。Monkey运行时输出的日志（monkey\_test.txt文件）一般包括四类信息，如图6所示：

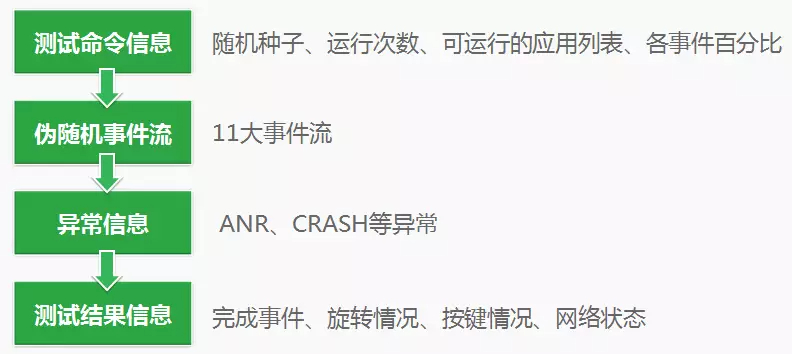


图6 Monkey运行log输出顺序

**1）测试命令信息**

Monkey启动后会输出当前执行命令的各种参数信息，其中包括随机种子（Seed）信息、事件数量、可运行的应用列表以及各事件百分比等。这些信息由Monkey命令参数指定。Monkey日志中测试命令信息如下：

// 随机种子值和执行事件数量

:Monkey: seed=4007 count=92665

// 可运行的应用列表

:AllowPackage: com.xxx.xxx

:IncludeCategory: android.intent.category.LAUNCHER

:IncludeCategory: android.intent.category.MONKEY

// Selecting main activities from category android.intent.category.LAUNCHER

......

//   - NOT USING main activity com.jrdcom.care.launcher.CareLauncher (from package com.jrdcom.care.launcher)

// Seeded: 4007

// 各事件百分比

// Event percentages:

//   0: 40.0%

//   1: 25.0%

//   2: 0.1724138%

//   3: 1.2931035%

//   4: 10.0%

//   5: 2.1551723%

//   6: 10.0%

//   7: 0.1724138%

//   8: 10.0%

//   9: 0.0862069%

//   10: 1.1206896%

**2）伪随机事件流信息**

当Monkey开始执行测试后，会顺序输出执行的事件流信息，主要是前面日志中的11大事件。**Monkey日志中伪随机事件流信息如下：**

:Switch: #Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.xxx.xxx/.PageStartActivity;end

    // Allowing start of Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.xxx.xxx/.PageStartActivity } in package com.xxx.xxx

Sleeping for 500 milliseconds

:Sending Touch (ACTION\_DOWN): 0:(9.0,893.0)

    // Injection Failed

:Sending Touch (ACTION\_UP): 0:(9.040771,884.45905)

    // Injection Failed

Sleeping for 500 milliseconds

:Sending Key (ACTION\_DOWN): 23    // KEYCODE\_DPAD\_CENTER

:Sending Key (ACTION\_UP): 23    // KEYCODE\_DPAD\_CENTER

......

**3）异常信息**

当Monkey执行过程中遇到错误时，会输出相应异常信息。**Monkey日志中异常信息（ANR情况，需获取traces.txt文件，命令adb pull /data/anr/traces.txt）如下：**

// NOT RESPONDING: com.xxx.xxx(pid 6143)

ANR in com.xxx.xxx(com.xxx.xxx/.ui.CourseDetailsActivity), time=5339522

// ANR的详细信息

Reason: Input dispatching timed out (Waiting to send non-key event because the touched window has not finished processing certain input events that were delivered to it over 500.0ms ago.  Wait queue length: 2.  Wait queue head age: 9513.3ms.)

**4）Monkey执行结果信息**

当Monkey执行完事件后，会输出执行结果信息，其中包括执行的事件数量、旋转的角度、丢失的事件数量、网络状态以及Monkey最终的执行结果。**Monkey日志中执行结果信息如下：**

// 执行的事件数量

Events injected: 808

// 旋转的角度

:Sending rotation degree=0, persist=false

// Generated profiling reports in /data/misc

// 丢失的事件数量

:Dropped: keys=0 pointers=10 trackballs=0 flips=0 rotations=0

// 网络状态

## Network stats: elapsed time=138721ms (0ms mobile, 0ms wifi, 138721ms not connected)

// Monkey最终的执行结果

// Monkey finished

如果执行过程中出现异常导致执行失败，会输出失败的原因，第几个事件执行失败以及所使用的随机种子值。**Monkey日志中执行结果异常信息如下：**

// 显示Monkey执行失败

\*\* Monkey aborted due to error.

Events injected: 35

:Sending rotation degree=0, persist=false

// Generated profiling reports in /data/misc

:Dropped: keys=1 pointers=8 trackballs=0 flips=0 rotations=0

## Network stats: elapsed time=34388ms (0ms mobile, 0ms wifi, 34388ms not connected)

// 提示在执行到第35个事件时出现Crash，以及所使用的随机种子值是4007。

\*\* System appears to have crashed at event 35 of 92665 using seed 4007

**3. 搜索Monkey日志并统计测试结果：**

**1）程序无响应的情况：搜索关键字“ANR in”**

**2）崩溃的情况1：搜索关键字“CRASH”**

**3）崩溃的情况2：搜索关键字“Exception”**

**4）内存溢出的情况：搜索关键字“OOM”**

**5）测试成功的情况：搜索关键字“Monkey finished”**

**6）其他情况：测试执行中断**

根据统计结果，可以得到各种异常情况出现的次数，已经出现在哪个日志文件中，出现该错误的包名。如果需要更详细的错误信息，打开对应的Monkey日志文件查询。通过详细日志信息，测试可以定位到引起异常的原因，已经出现异常的代码行信息。常见的Java异常信息，如表1：

