

勘误一：

-s 没加上，少 500，修正请见下文，涉及内容如下。

### 4.3.1 -s 参数的示例讲解

问题：我们在使用 monkey 工具在执行测试的时候，很有可能我们在 monkey 命令执行完成后，发现了一些问题，这时候，咱们的程序同事可能就会说：“hi，哥们帮我复现一下那个问题，我好定位下是那块的问题。”，这是非常普遍的一种情况，那么有没有办法可以使 monkey 完全重复一下上次的操作呢？比如：上一次，最开始单击的是 x 轴为 200，y 轴为 300 的坐标点，而后又从该点执行划屏操作，划到另一个坐标点 x 轴为 500，y 轴为 600，再后来又执行一系列的输入、划屏和单击事件。那么我们有没有办法保证每次的执行是完全一致的呢？在这里我可以很肯定的告诉大家，不能。但是，我们能保证每次的执行事件、序列是一致的，也就是说上次执行的是先单击再划屏事件，这次它执行的也是先单击再划屏事件。那么在 monkey 中加入那个参数就可以干这件事了呢？

解答：monkey 提供的“-s”参数，用于指定伪随机数生成器的 seed（种子）值，如果 seed 值相同，则两次 Monkey 测试所产生的事件序列也相同的。

比如，这里我们分别使用“monkey”执行了两次测试，即：

第一次测试输入的 monkey 命令为：“adb shell monkey -v -v -v -p com.android.calendar -s 100 500”；

第二次测试输入的 monkey 命令为：“adb shell monkey -v -v -v -p com.android.calendar -s 100 500”；

从上述两次输入的 monkey 命令来看，它们的随机种子都是 100，是一致的。这样就能够使得两次测试的效果是相同的，因为模拟的用户操作序列是一样的，就可以保证两次测试产生的随机操作序列是完全相同的。

#### 【重点提示】

- (1) 重现问题是测试人员经常会面对的一件事情，所以大家在应用 monkey 时一定要掌握好“-s”参数的应用，每次建议大家执行测试时都应该记录使用的命令及用管道命令保存输出结果到文件中，使得命令和执行结果一一对应。
- (2) 这里给大家举一个例子，比如，我们执行 monkey 命令“adb shell monkey -v -v -v -p

com.android.calendar -s 100 500 > C:\Monkey\_Results\calendar\_TC01\_S\_01.txt”，执行完成上述命令后，您就会在“C:\Monkey\_Results”文件夹下看到有一个“calendar\_TC01\_S\_01.txt”的文本文件，该文件的内容就是 monkey 的执行输出结果信息。接下来，您可以设计一个 Excel 表格对其进行管理，这里给大家一个文档格式内容（当然最好还是依据于您的实际情况和需求进行设定），如图 4-5 所示。

**日历应用稳定性测试（使用工具：Monkey）**

序列	执行人	执行时间	命令行	结果存放位置
1	悟空	2015.3.10 16:10	adb shell monkey -v -v -v -p com.android.calendar -s 100 500 > C:\Monkey_Results\calendar_TC01_S_01.txt	C:\Monkey_Results\calendar_TC01_S_01.txt
2	悟空	2015.3.10 18:11	adb shell monkey -v -v -v -p com.android.calendar -s 100 500 > C:\Monkey_Results\calendar_TC01_S_02.txt	C:\Monkey_Results\calendar_TC01_S_02.txt
3	悟空	2015.3.11 10:00	adb shell monkey -v -v -v -p com.android.calendar -s 100 500 > C:\Monkey_Results\calendar_TC01_S_03.txt	C:\Monkey_Results\calendar_TC01_S_03.txt
4	悟空	2015.3.11 16:15	adb shell monkey -v -v -v -p com.android.calendar -s 100 500 > C:\Monkey_Results\calendar_TC01_S_04.txt	C:\Monkey_Results\calendar_TC01_S_04.txt
5	悟空	2015.3.11 20:20	adb shell monkey -v -v -v -p com.android.calendar -s 100 500 > C:\Monkey_Results\calendar_TC01_S_05.txt	C:\Monkey_Results\calendar_TC01_S_05.txt

图 4-5 monkey 命令之行后的输出

勘误二：

104 页，?? 应为 106

即：

从图 3-106 中，我们能够清楚的看到我们刚才通过命令行创建的模拟器，显示在模拟器列表中。

**【重点提示】**

勘误三：

90 页，图注应为 3-77，如下所示：

```
ca. 管理员: C:\Windows\system32\cmd.exe
^C^C
C:\Users\Administrator>adb bugreport
=====
== dumpstate: 2015-06-22 11:45:08
=====

Build: JZ054K.N719KEBMF3
Build fingerprint: 'samsung/t03gctc/t03gctc:4.1.2/JZ054K/N719KEBMF3:user/release-keys'
Bootloader: N719KEBMC1
Radio: mdm
Network: China Telecom
Kernel: Linux version 3.0.31-999893 (se.infra@SEP-90) (gcc version 4.4.3 (GCC) )
#1 SMP PREEMPT Fri Jun 21 10:47:45 KST 2013
Command line: console=ram loglevel=4 androidboot.baseband=mdm sec_debug.level=0
sec_watchdog.sec_pet=5 androidboot.debug_level=0x4f4c sec_log=0x20000000x46000000
0 s3cfb.bootloaderfb=0x5ec00000 sysscope=0xee000000 lcdtype=1 consoleblank=0 lpc
harge=0 lpj=3981312 vmalloc=176m oops=panic pmic_info=65 cordon=ffe2d070c05edcb6
09239c576b7b3070 androidboot.emmc_checksum=3 androidboot.odin_download=0 android
boot.bootloader=N719KEBMC1 androidboot.serialno=4df7b6be03f2302b snd_soc_core.pm
down_time=1000

----- UPTIME (uptime) -----
up time: 02:48:32, idle time: 09:17:04, sleep time: 00:05:25
[uptime: 0.1s elapsed]

----- MEMORY INFO (/proc/meminfo) -----
MemTotal:      1833572 kB
MemFree:       47648 kB
Buffers:       121896 kB
Cached:        657960 kB
SwapCached:    0 kB
Active:        943980 kB
Inactive:      536852 kB
Active(anon):  702240 kB
Inactive(anon): 5028 kB
Active(file):  241740 kB
Inactive(file): 531824 kB
Unevictable:   564 kB
```

图 3-77 查看日志消息命令及其相关输出内容