# 经验分享

## 背景

前一份工作从事的是手机测试工作，主要涉及手机系统的功能测试，包括系统设置、预置应用、定制项、Google TA检查项等。涉及功能测试、压力测试、和自动化测试（Monkey和功能自动化测试）。现作简单分享.

## 常用内部命令的使用

* 1. 熟悉adb命令，能够使用adb对手机进行简单控制，比如push、pull等进行文件的操作。forward对端口进行转发，logcat信息的打印。对存储进行挂载（remount）等功能，将手机只读分区挂载为可读写的方法 mount –o remount rw /system/ 即可实现对system分区进行读写操作，可以将只读分区的内容正常进行修改和删除等操作。
  2. 熟悉手机内部命令，am、pm、mv、rm等常用命令，常用的一个方法，为了填充手机内存空间，可以通过dd命令进行内存填充，将手机内部存储空间填充满的方法，命令形如adb shell dd if=/dev/zero of=/mnt/sdcard/mem.txt bs=1024 count=100000000000 使用此命令可以快速对内存进行填充，直到内存完全填充完毕。使用am命令可以快速启动手机应用。
  3. 列出手机中安装的所有的应用程序的包名，可以使用命令adb shell pm list packages,使用此命令可以快速地将手机中所安装的所有的应用程序的包名显示出来。还可以在后面加入对应的关键字进行搜索。例如pm list package tencent,返回所有包名中包含tencent的应用程序包，像com.tencent.android.qqdownloader、com.tencent.bugly、com.tencent.mtt、com.tencent.qlauncher、com.tencent.qlauncher.onekeylocker、com.tencent.qqmusic、com.tencent.test.qqdriver.android.player、com.tencent.testagent等应用包名都会显示在列表中。
  4. 根据程序包名，可以找到apk所存放的位置。同样使用到pm命令，有一个参数为，帮助中显示为pm path: print the path to the .apk of the given PACKAGE.可以方便的使用此命令，比如pm path com.tencent.qlauncher 直接返回package:/data/app/com.tencent.qlauncher-1.apk，可以找到apk存放的位置。
  5. 通过命令安装应用程序;并携带相关参数

pm install: installs a package to the system. Options:

-l: install the package with FORWARD\_LOCK.

-r: reinstall an exisiting app, keeping its data.

-t: allow test .apks to be installed.

-i: specify the installer package name.

-s: install package on sdcard.

-f: install package on internal flash.

例如pm install –lrtis /sdcard/tuya.apk 可以快速将tuya.apk安装到外置SD卡上，并且限制其进行分享。

## Monkey命令使用方法

* 1. 熟悉monkey命令,并知道黑白名单的设置，可以通过批处理进行批量化的操作。在此分享一下monkey的使用方法，附使用帮助文档：

usage: monkey [-p ALLOWED\_PACKAGE [-p ALLOWED\_PACKAGE] ...]

[-c MAIN\_CATEGORY [-c MAIN\_CATEGORY] ...]

[--ignore-crashes] [--ignore-timeouts]

[--ignore-security-exceptions]

[--monitor-native-crashes] [--ignore-native-crashes]

[--kill-process-after-error] [--hprof]

[--pct-touch PERCENT] [--pct-motion PERCENT]

[--pct-trackball PERCENT] [--pct-syskeys PERCENT]

[--pct-nav PERCENT] [--pct-majornav PERCENT]

[--pct-appswitch PERCENT] [--pct-flip PERCENT]

[--pct-anyevent PERCENT] [--pct-pinchzoom PERCENT]

[--pkg-blacklist-file PACKAGE\_BLACKLIST\_FILE]

[--pkg-whitelist-file PACKAGE\_WHITELIST\_FILE]

[--wait-dbg] [--dbg-no-events]

[--setup scriptfile] [-f scriptfile [-f scriptfile] ...]

[--port port]

[-s SEED] [-v [-v] ...]

[--throttle MILLISEC] [--randomize-throttle]

[--profile-wait MILLISEC]

[--device-sleep-time MILLISEC]

[--randomize-script]

[--script-log]

[--bugreport]

[--periodic-bugreport]

COUNT

-p参数可以指定运行的应用程序的包名，可以方便进行白名单测试，例如-p com.android.browser 可以单独进行浏览器测试。

[--ignore-crashes] [--ignore-timeouts]

[--ignore-security-exceptions]

[--monitor-native-crashes] [--ignore-native-crashes]

这几个主要是对错误的处理方式，通常情况下选择“遇错不停”即所有的错误都忽略。直到次数跑完。

[--kill-process-after-error] [--hprof] 遇到错误进程后，将进程杀死并导出错误信息。

[--pct-touch PERCENT] [--pct-motion PERCENT]

[--pct-trackball PERCENT] [--pct-syskeys PERCENT]

[--pct-nav PERCENT] [--pct-majornav PERCENT]

[--pct-appswitch PERCENT] [--pct-flip PERCENT]

[--pct-anyevent PERCENT] [--pct-pinchzoom PERCENT]

分配各个事件所占的比例，比如“点击”、“滑动”、“方向键”、“程序切换”、“任意事件”等。

[--pkg-blacklist-file PACKAGE\_BLACKLIST\_FILE]

[--pkg-whitelist-file PACKAGE\_WHITELIST\_FILE]

指定黑白名单的文件路径和文件名。

[-s SEED] [-v [-v] ...] 指定本次monkey测试的种子数。帮助生成伪随机序列。

[--throttle MILLISEC] [--randomize-throttle] 设置每次点击的延时时间，通常设置成1000ms.后面一个选项为随机进行延时。

COUNT 设置随机事件执行的测试次数，即点击次数。

## MonkeyRunner使用

* 1. 大致了解monkeyrunner模块的启动方式，能够利用monkeyrunner模块实现对手机的控制。大致方法如下
  2. 先通过android sdk文件夹目录，找到monkeyrunner.bat可以将monekyrunner运行起来，然后将常用的包导入from com.android.monkeyrunner import MonkeyDevice,MonekyRunner,MonkeyImage将常用的包进来。
  3. 对手机进行实例化控制，DUT=MonkeyRunner.waitForConnection(),可以实现对设备的连接和控制。
  4. 对手机进行常用的操作，附上方法列表

|  |  |
| --- | --- |
| Methods | |
| void | [broadcastIntent](http://www.android-doc.com/tools/help/MonkeyDevice.html#broadcastIntent) (*string* uri, *string* action, *string* data, *string* mimetype, *iterable* categories *dictionary* extras, *component* component, *iterable* flags)  Broadcasts an Intent to this device, as if the Intent were coming from an application. |  |
| void | [drag](http://www.android-doc.com/tools/help/MonkeyDevice.html#drag) (*tuple* start, *tuple* end, *float* duration, *integer* steps)  Simulates a drag gesture (touch, hold, and move) on this device's screen. |  |
| *object* | [getProperty](http://www.android-doc.com/tools/help/MonkeyDevice.html#getProperty) (*string* key)  Given the name of a system environment variable, returns its value for this device. The available variable names are listed in the [detailed description](http://www.android-doc.com/tools/help/MonkeyDevice.html#getProperty) of this method. |  |
| *object* | [getSystemProperty](http://www.android-doc.com/tools/help/MonkeyDevice.html#getSystemProperty) (*string* key)  . The API equivalent of **adb shell getprop <key>. This is provided for use by platform developers.** |  |
| void | [installPackage](http://www.android-doc.com/tools/help/MonkeyDevice.html#installPackage) (*string* path)  Installs the Android application or test package contained in packageFile onto this device. If the application or test package is already installed, it is replaced. |  |
| *dictionary* | [instrument](http://www.android-doc.com/tools/help/MonkeyDevice.html#instrument) (*string* className, *dictionary* args)  Runs the specified component under Android instrumentation, and returns the results in a dictionary whose exact format is dictated by the component being run. The component must already be present on this device. |  |
| void | [press](http://www.android-doc.com/tools/help/MonkeyDevice.html#press) (*string* name, *dictionary* type)  Sends the key event specified by type to the key specified by keycode. |  |
| void | [reboot](http://www.android-doc.com/tools/help/MonkeyDevice.html#reboot) (*string* into)  Reboots this device into the bootloader specified by bootloadType. |  |
| void | [removePackage](http://www.android-doc.com/tools/help/MonkeyDevice.html#removePackage) (*string* package)  Deletes the specified package from this device, including its data and cache. |  |
| *object* | [shell](http://www.android-doc.com/tools/help/MonkeyDevice.html#shell) (*string* cmd)  Executes an **adb** shell command and returns the result, if any. |  |
| void | [startActivity](http://www.android-doc.com/tools/help/MonkeyDevice.html#startActivity) (*string* uri, *string* action, *string* data, *string* mimetype, *iterable* categories *dictionary* extras, *component* component, *flags*)  Starts an Activity on this device by sending an Intent constructed from the supplied arguments. |  |
| [**MonkeyImage**](http://www.android-doc.com/tools/help/MonkeyImage.html) | [takeSnapshot](http://www.android-doc.com/tools/help/MonkeyDevice.html#takeSnapshot)()  Captures the entire screen buffer of this device, yielding a [**MonkeyImage**](http://www.android-doc.com/tools/help/MonkeyImage.html)object containing a screen capture of the current display. |  |
| void | [touch](http://www.android-doc.com/tools/help/MonkeyDevice.html#touch) (*integer* x, *integer* y, *integer* type)  Sends a touch event specified by type to the screen location specified by x and y. |  |
| void | [type](http://www.android-doc.com/tools/help/MonkeyDevice.html#touch) (*string* message)  Sends the characters contained in message to this device, as if they had been typed on the device's keyboard. This is equivalent to calling[**press()**](http://www.android-doc.com/tools/help/MonkeyDevice.html#press) for each keycode in **message** using the key event type **DOWN\_AND\_UP**. |  |
| void | [wake](http://www.android-doc.com/tools/help/MonkeyDevice.html#touch) ()  Wakes the screen of this device. |  |

## 批处理结合常用命令使用

有windows的批处理编写基础，结合ADB命令可以快速对手机进行各项操作。也可以方便对手机的命令进行批量化使用。简单举出两个例子..

### 4.1快速安装应用程序

大致编写如下：

@echo off

for /r %%i in (\*.apk) do (

echo install %%i

adb install -r %%i

)

echo install finished &pause>nul

使用此命令可以快速将当前批处理命令所在文件夹以及其所有子文件夹下面的apk文件进行安装。

附件

### 4.2 快速杀死进程

大致编写如下：

@echo off

title killProcess

echo please input the process name you want to kill?

set /p pid=

if "%pid%" neq "" (

for /f "tokens=2 " %%i in ('adb shell ps^|findstr /i "%pid%"') do (

echo kill %%i

adb shell kill %%i

echo kill %pid% successfully~~ &pause >nul

)

) else (

echo have not find the process

goto :eof

)

提示用户输入进程的大致名称即可找到对应的进程，然后将其杀死，不需要通过apk进行大量操作，简化了用户交互过程中浪费的时间。

附件：