Monkey测试的调研与实践

姓名：郝裕玮

班级：计科1班

学号：18329015

目录

[1 Monkey测试基础知识 3](#_Toc90680484)

[1.1 Monkey简介 3](#_Toc90680485)

[1.2 Monkey参数 4](#_Toc90680486)

[2 实验环境搭建 7](#_Toc90680487)

[2.1 硬件环境 7](#_Toc90680488)

[2.2 软件配置 7](#_Toc90680489)

[3 Monkey测试过程 14](#_Toc90680490)

[3.1 测试步骤 14](#_Toc90680491)

[3.2 结果分析 16](#_Toc90680492)

[4 Monkey测试优缺点 20](#_Toc90680493)

[5 智能Monkey测试 21](#_Toc90680494)

[5.1 Maxim & UICrawler 21](#_Toc90680495)

[5.2 Maxim环境配置 23](#_Toc90680496)

[5.3 命令行运行Maxim 23](#_Toc90680497)

[5.4 分析日志 24](#_Toc90680498)

[5.5 UICrawler环境配置 29](#_Toc90680499)

[5.6 UICrawler运行结果 31](#_Toc90680500)

[6 实验感想 32](#_Toc90680501)

# 1 Monkey测试基础知识

## 1.1 Monkey简介

Monkey 测试是 Android 自动化测试APP的手段之一。monkey译为猴子，所以从原理上来说，它的自动化测试就是像一只什么都不懂的猴子一样在软件上乱敲按键。

Monkey测试是一个命令行工具，它可以运行在模拟器或实际设备中。它通过向系统发送伪随机的用户事件流（如按键输入、触摸屏输入、滑动Trackball、手势输入等操作），实现对全系统或某个应用程序进行压力测试。

Monkey工具存在Android系统中，使用Java语言写成。其

jar包在Android文件系统中的存放路径是：/system/framework/monkey.jar。

Monkey.jar程序是由一个名为“monkey”的Shell脚本来启动执行，shell脚本在Android文件系统中的存放路径是：/system/bin/monkey。

monkey需要通过adb工具来唤醒，即在cmd窗口中执行: adb shell monkey ｛+命令参数｝来进行Monkey测试。

## 1.2 Monkey参数

（1）-p：

用于指定的一个或多个应用包，当指定包后，monkey将只允许系统启动指定的app，如果没有指定，monkey将允许系统启动设备中的所有app。

#指定1个应用

#com.htc.Weather为包名，100是事件计数（即让Monkey程序模拟100次随机用户事件）。

adb shell monkey -p com.htc.Weather 100

#指定多个应用

adb shell monkey -p com.htc.Weather –p com.htc.pdfreader -p com.htc.photo.widgets 100

#不指定应用

#Monkey随机启动APP并发送100个随机事件。

adb shell monkey 100

（2）-v：

用于指定反馈信息级别，即日志的详细程度，总共有3个级别，一般测试的时候选择最高级别获取最详细的日志。

#Level0：仅提供启动提示、测试完成和最终结果等少量信息

adb shell monkey -p com.htc.Weather –v 100

#Level1：提供较为详细的日志，包括每个发送到 Activity 的事件信息

adb shell monkey -p com.htc.Weather –v -v 100

#Level2：最详细的日志，包括了测试中选中/未选中的 Activity 信息

adb shell monkey -p com.htc.Weather –v -v -v 100

（3）-s：

用于指定monkey生成器的seed值，一个seed值对应一个事件序列，当你使用进行monkey测试出现异常后，在日志中提取seed值并进行指定，可生成相同的事件序列，有利于bug的重现。

#测试1

adb shell monkey -p com.htc.Weather –s 10 100

#测试2

adb shell monkey -p com.htc.Weather –s 10 100

#两次测试的效果是相同的。

#因为模拟的用户操作序列（每次操作按照一定的先后顺序所组成的一系列操作，即一个序列）是一样的。

#操作序列虽然是随机生成的，但是只要我们指定了相同的seed值，就可以保证两次测试产生的随机操作序列是完全相同的。

#所以这个操作序列是伪随机的。

（4）--throttle：

用于指定用户操作事件的延迟时间，单位为毫秒，若不指定该参数，则Monkey会尽可能快地发送事件。参数缺省值为300ms。

adb shell monkey -p com.htc.Weather –-throttle 3000 100

（5）--ignore-crashes：

用于指定当应用程序崩溃时（Force & Close错误），Monkey是否停止运行。如果使用此参数，即使应用程序崩溃，Monkey依然会发送事件，直到事件计数完成。

#测试过程中即使Weather程序崩溃，Monkey依然会继续发送事件直到事件数目达到

1000为止

adb shell monkey -p com.htc.Weather --ignore-crashes 1000

#测试过程中，如果Weather程序崩溃，Monkey将会停止运行

adb shell monkey -p com.htc.Weather 1000

（6）--ignore-timeouts

用于指定当应用程序发生ANR（Application No Responding）错误时，Monkey是否停止运行。如果使用此参数，即使应用程序发生ANR错误，Monkey依然会发送事件，直到事件计数完成。

（7）--ignore-security-exceptions

用于指定当应用程序发生许可错误时（如证书许可，网络许可等），Monkey是否停止运行。如果使用此参数，即使应用程序发生许可错误，Monkey依然会发送事件，直到事件计数完成。

（8）--pct-｛+事件类别｝ ｛+事件类别百分比｝

用于指定每种类别事件的数目百分比（在Monkey事件序列中，该类事件数目占总事件数目的百分比）

① --pct-touch ｛+百分比｝

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)

adb shell monkey -p com.htc.Weather --pct-touch 10 1000

② --pct-motion ｛+百分比｝

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)

adb shell monkey -p com.htc.Weather --pct-motion 20 1000

③ --pct-trackball ｛+百分比｝

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)

adb shell monkey -p com.htc.Weather --pct-trackball 30 1000

④ --pct-nav ｛+百分比｝

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)

以上仅举例4条，其余指令在后续报告中会补充说明。

# 2 实验环境搭建

## 2.1 硬件环境

Windows系统 **+** 安卓手机

## 2.2 软件配置

adb介绍：

Android Debug Bridge（安卓调试桥）tools。它就是一个命令行窗口，用于通过电脑端与模拟器或者是设备之间的交互。

adb是一个C/S架构的应用程序，由三部分组成：

① 运行在PC端的adb client：

命令行程序adb用于从shell或脚本中运行adb命令。首先，adb程序尝试定位主机上的adb服务器，如果找不到adb服务器，adb程序自动启动一个adb服务器。接下来，当设备的adbd和PC端的adb server建立连接后，adb client就可以向adb server发送服务请求。

② 运行在PC端的adb server：

adb server是运行在主机上的一个后台进程。它的作用在于检测USB端口感知设备的连接和拔除，以及模拟器实例的启动或停止，adb server还需要将adb client的请求通过USB或者TCP的方式发送到对应的adbd上；

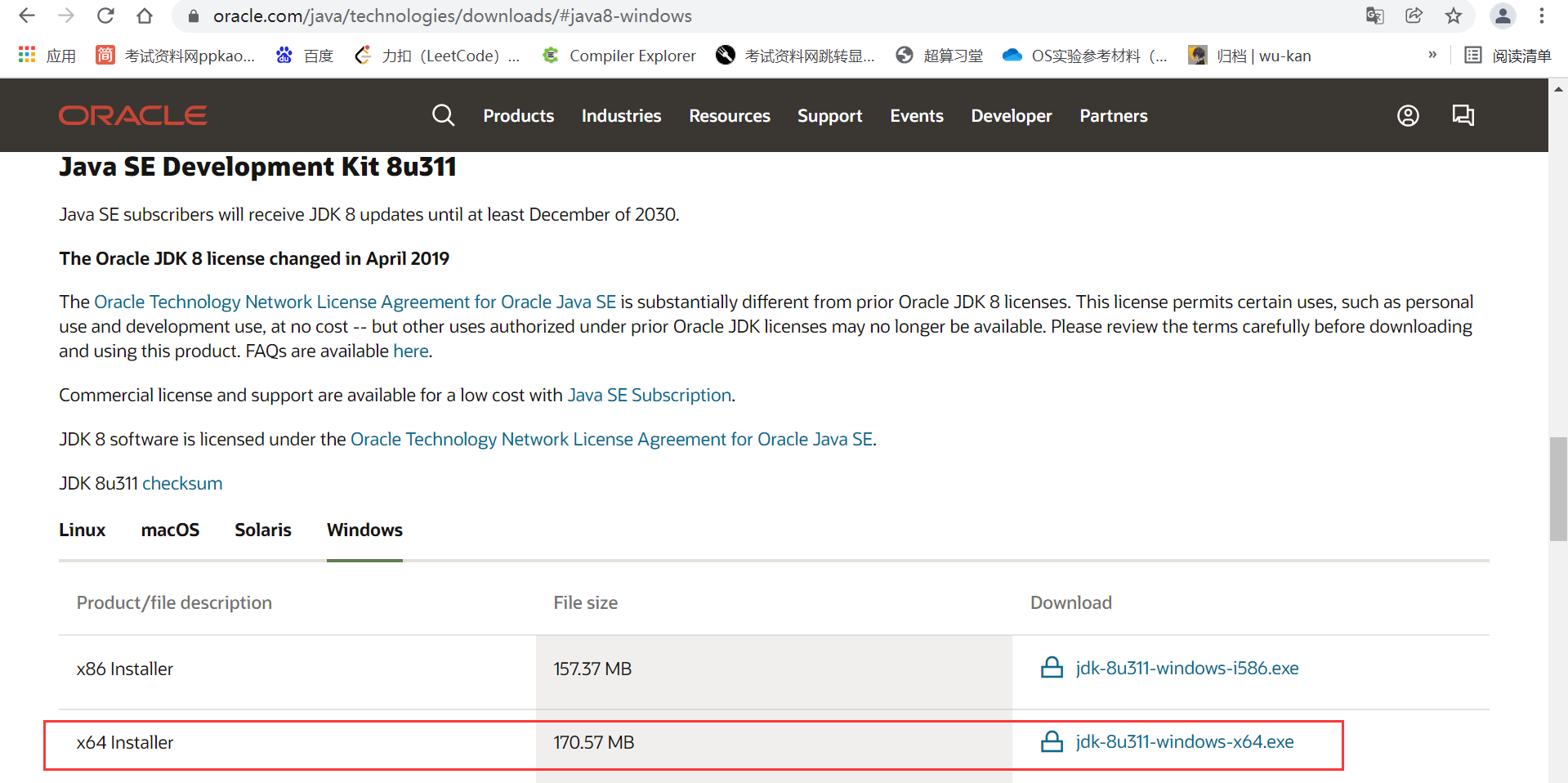
③ 运行在设备端的常驻进程adb demon (adbd)：

程序adbd作为一个后台进程在Android设备或模拟器系统中运行。它的作用是连接adb服务器，并且为运行在主机上的客户端提供一些服务。

配置步骤如下：

（1）配置Java环境，安装JDK

访问<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> 安装Windows安装包



下载完执行安装.exe文件默认安装即可。

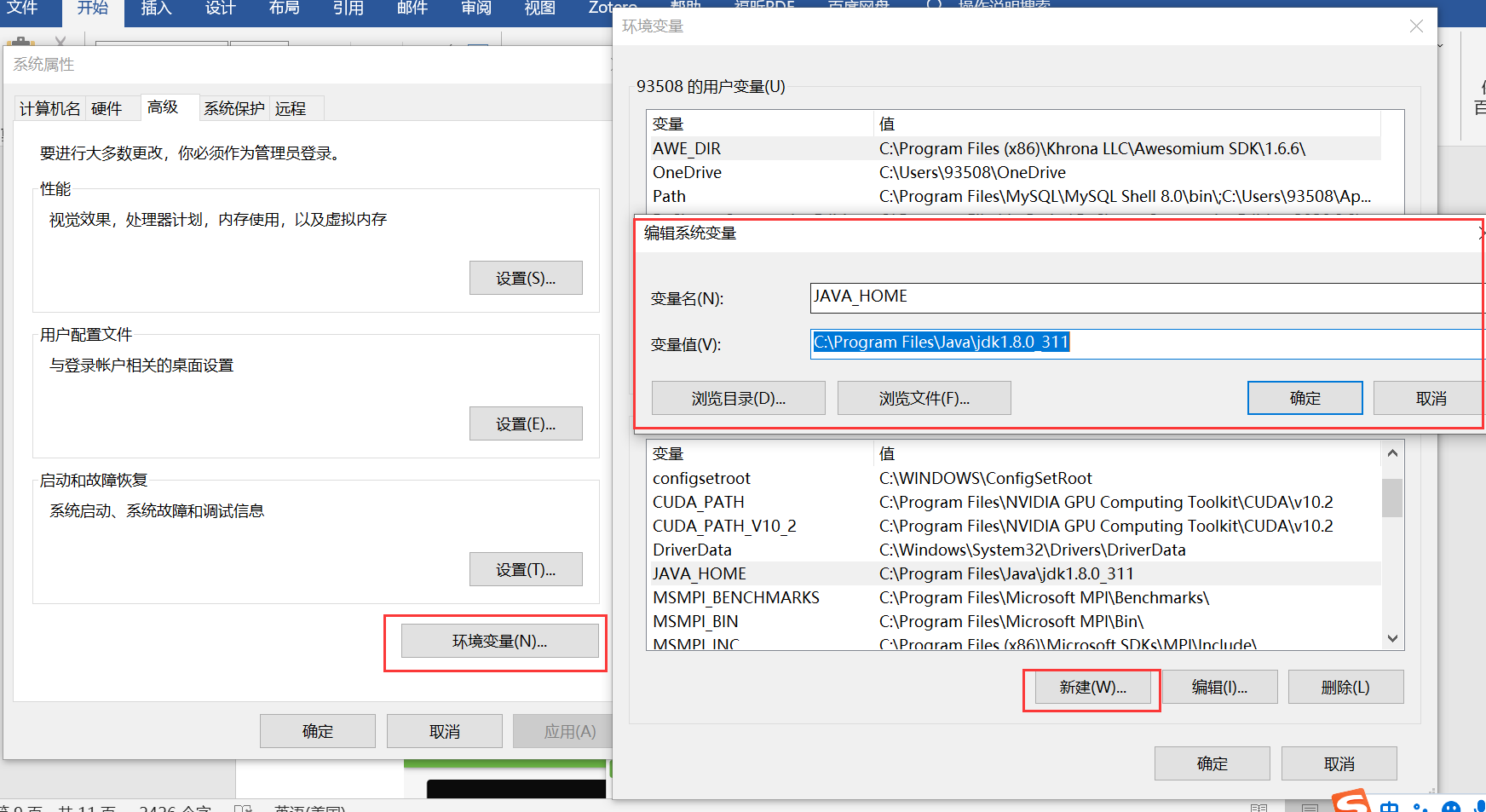
（2）JDK 环境变量配置

安装完成后，需要配置环境变量。

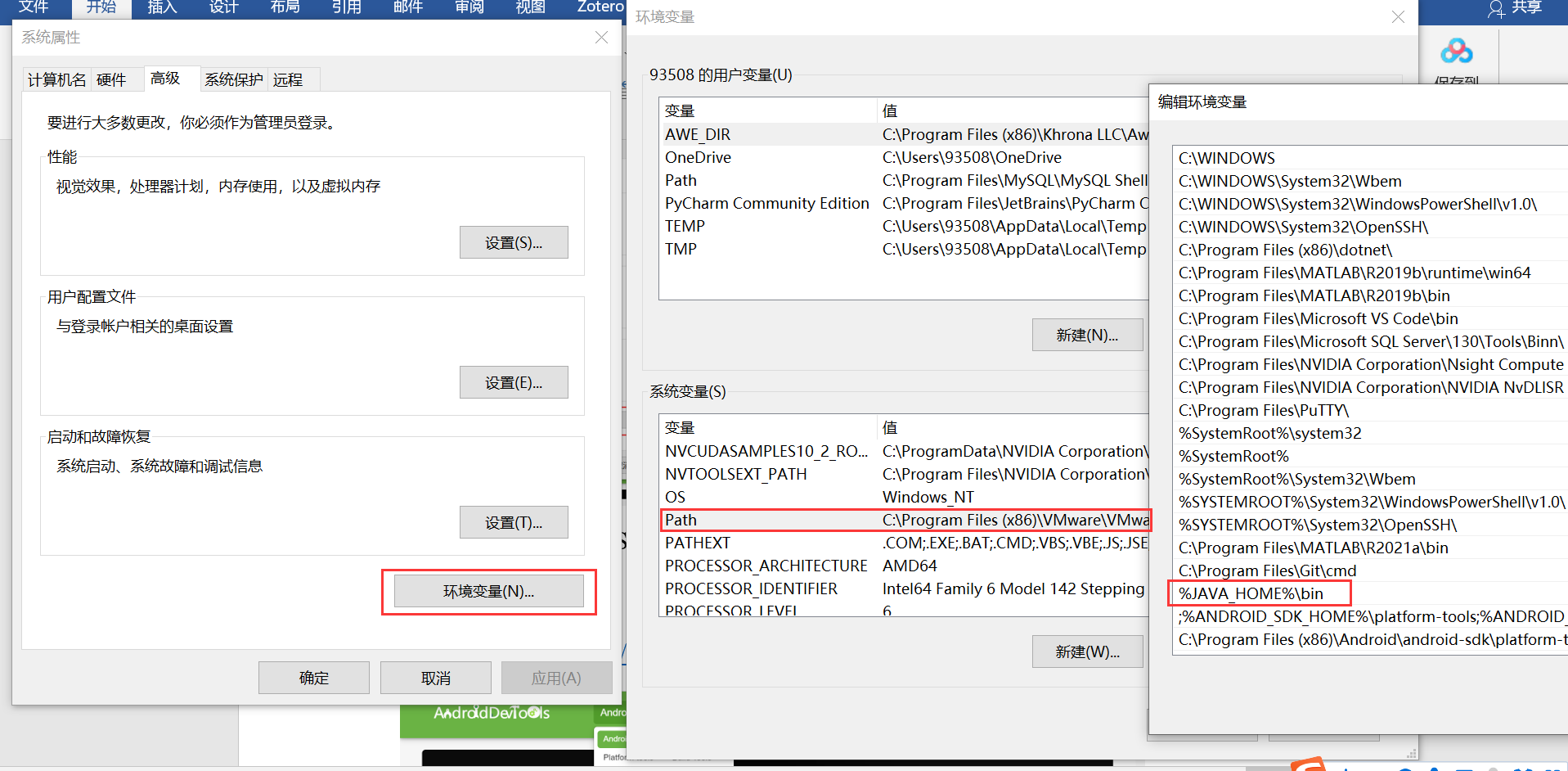
查看高级系统设置—环境变量—系统变量—新建

① 添加JAVA\_HOME 环境变量

路径为：C:\Program Files\Java\jdk1.8.0\_311

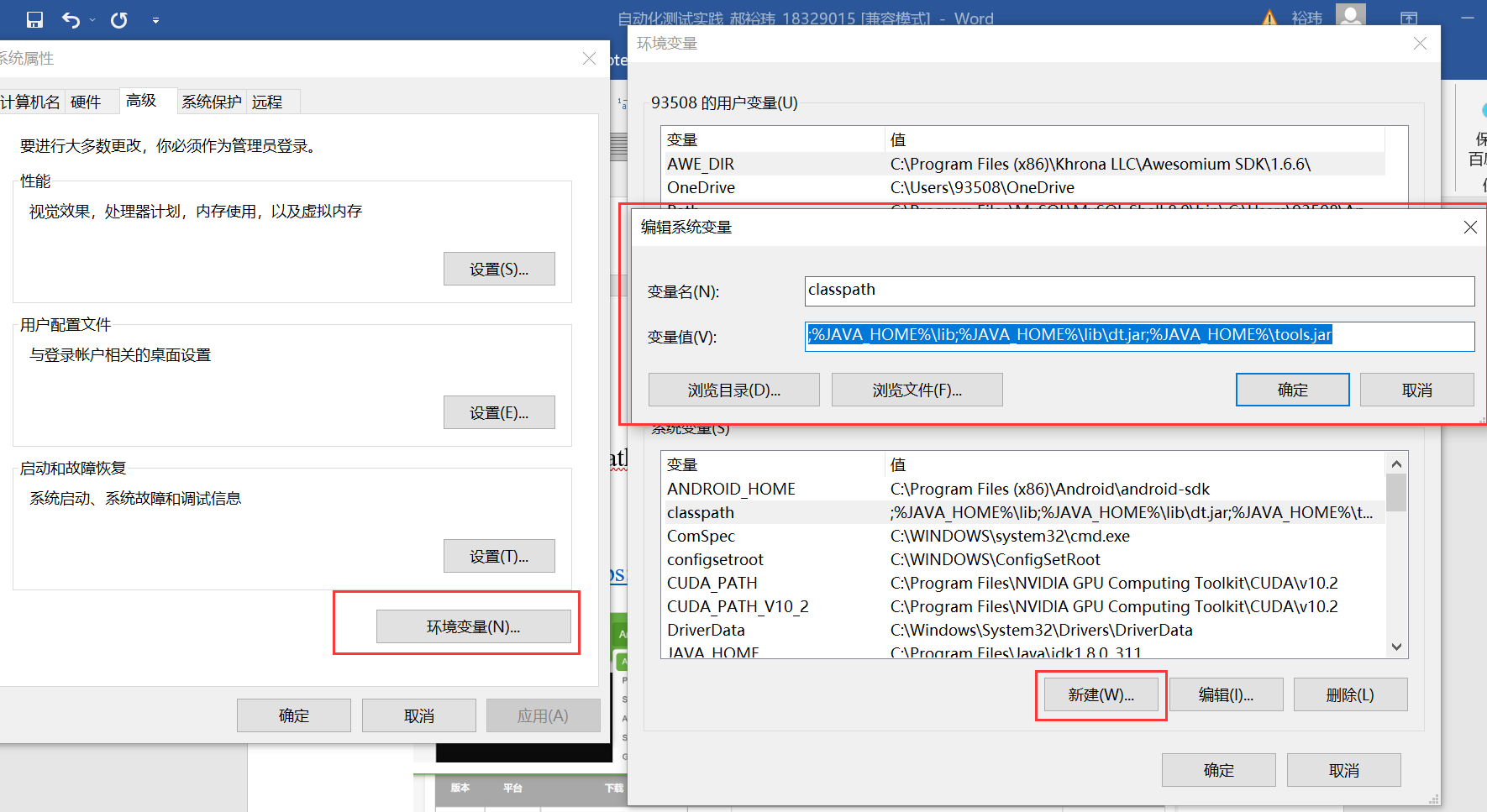


② 修改 Path 环境变量，在Path中添加%JAVA\_HOME%\bin

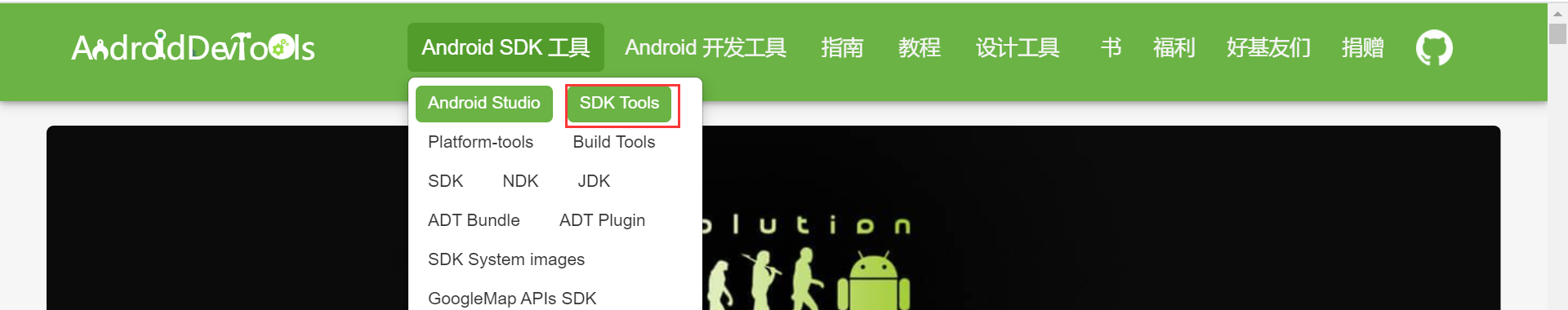


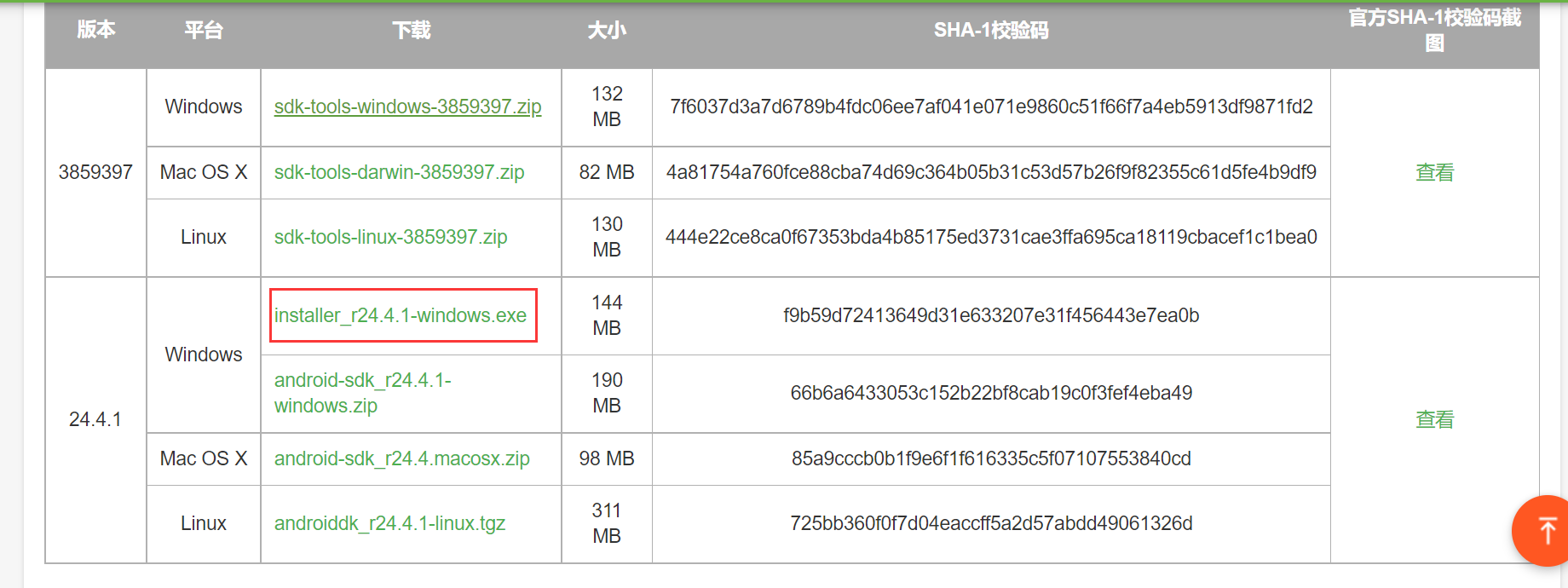
③ 添加 classpath 环境变量：

路径为：;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\tools.jar

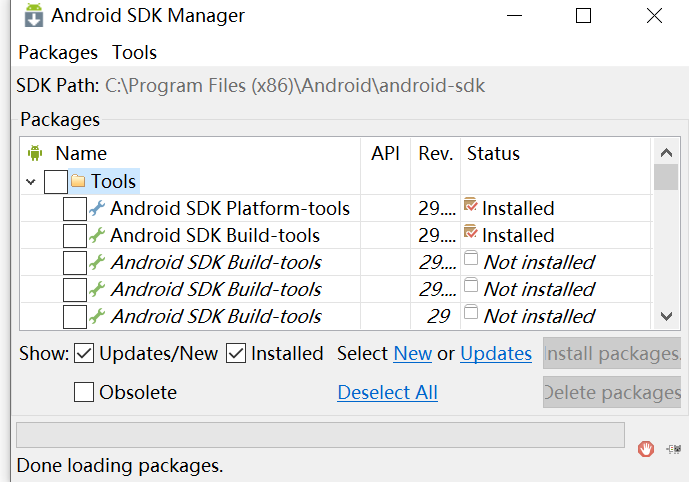


（3）进入<https://www.androiddevtools.cn/>下载SDK安装包





解压后，双击SDK Manager.exe打开SDK管理器，并按照默认配置安装相关安装包即可。

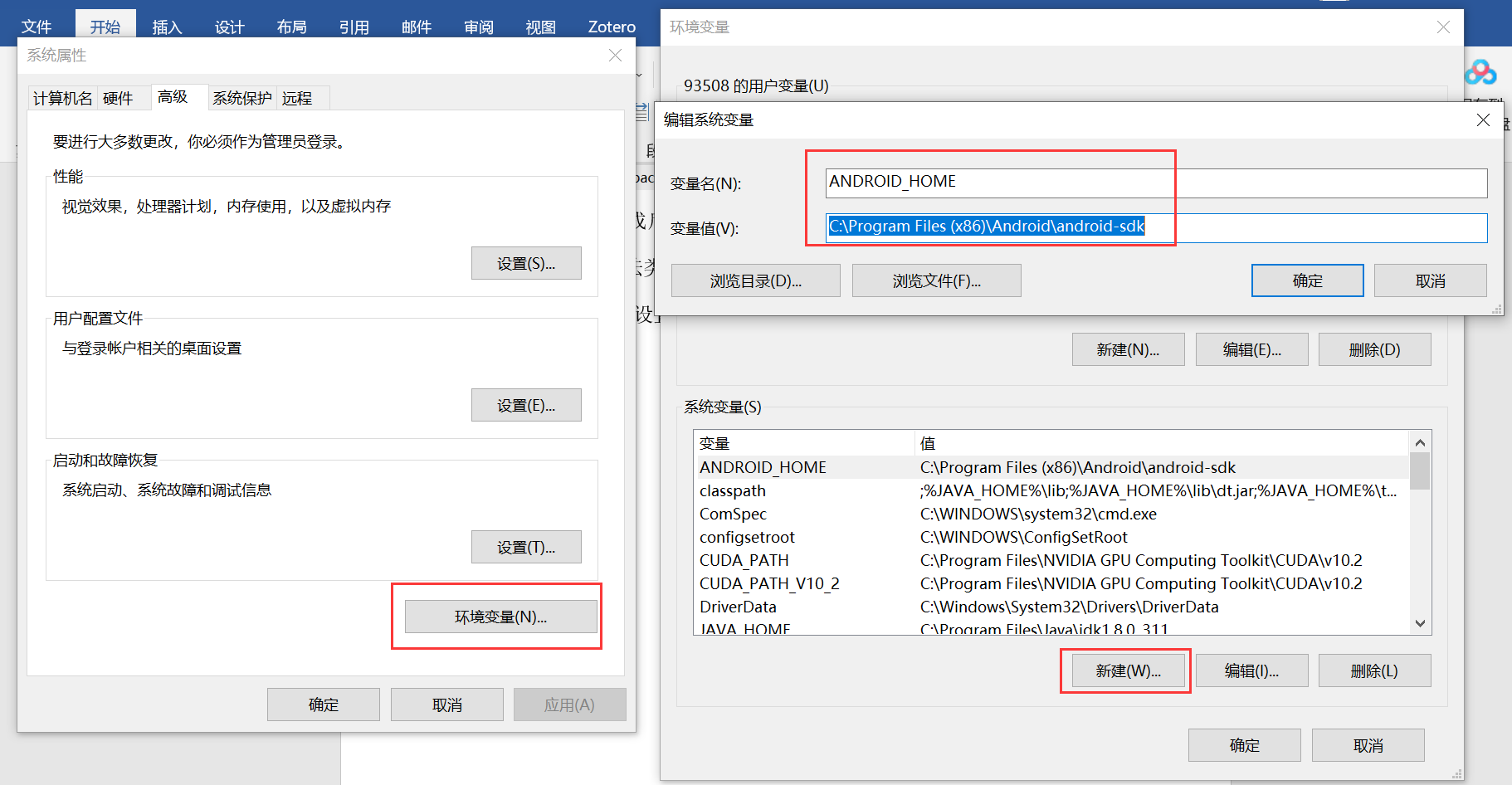


（4）安装完成后，需要配置环境变量。方法和设置JAVA JDK的环境变量的方法类似。

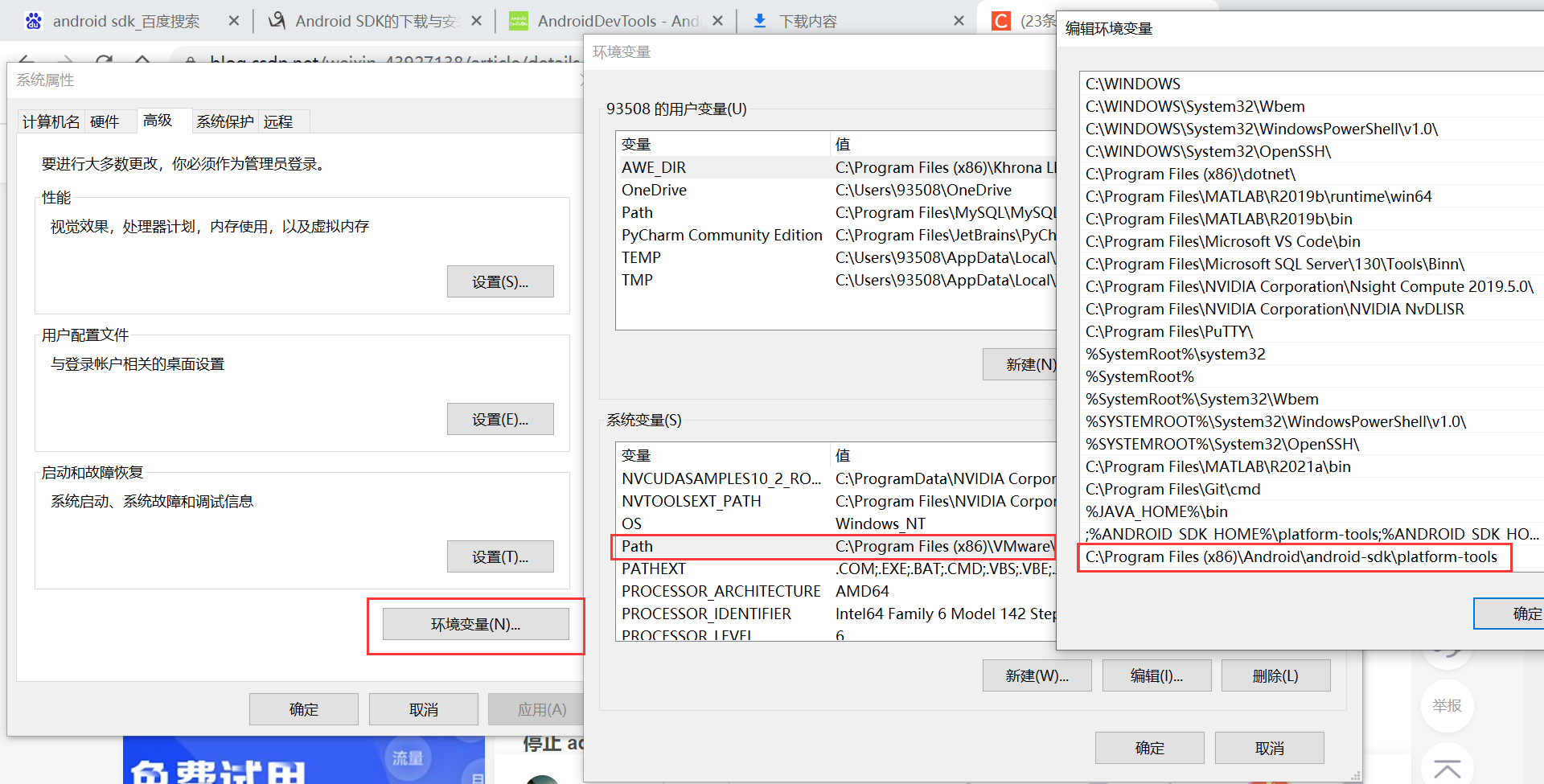
查看高级系统设置—环境变量—系统变量—新建

① 添加ANDROID\_HOME环境变量：

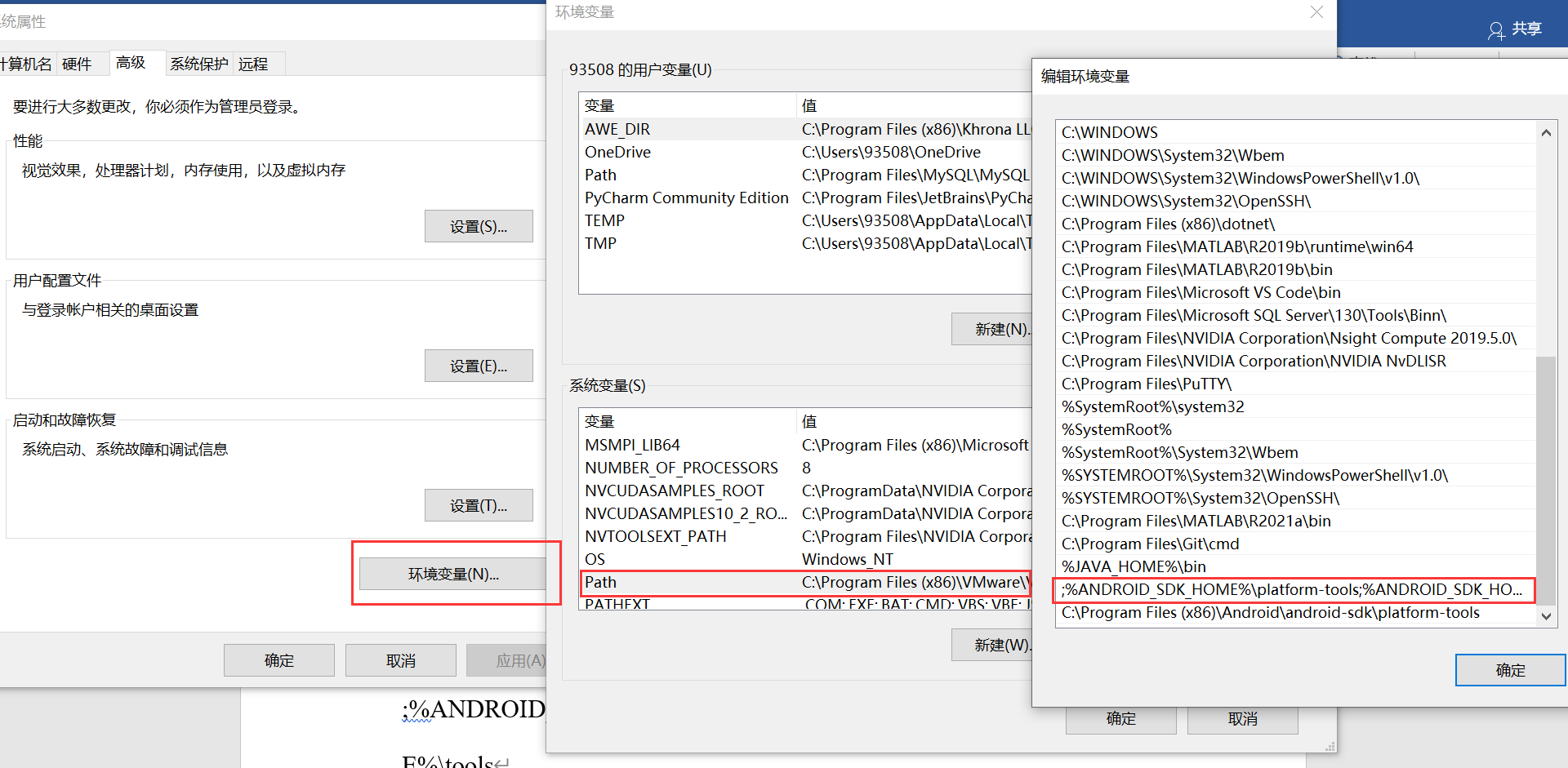
路径为：C:\Program Files (x86)\Android\android-sdk



② 将adb.exe在SDK里的路径添加到环境变量Path中



③ 将SDK的platform-tools和tool的路径添加到环境变量Path 中



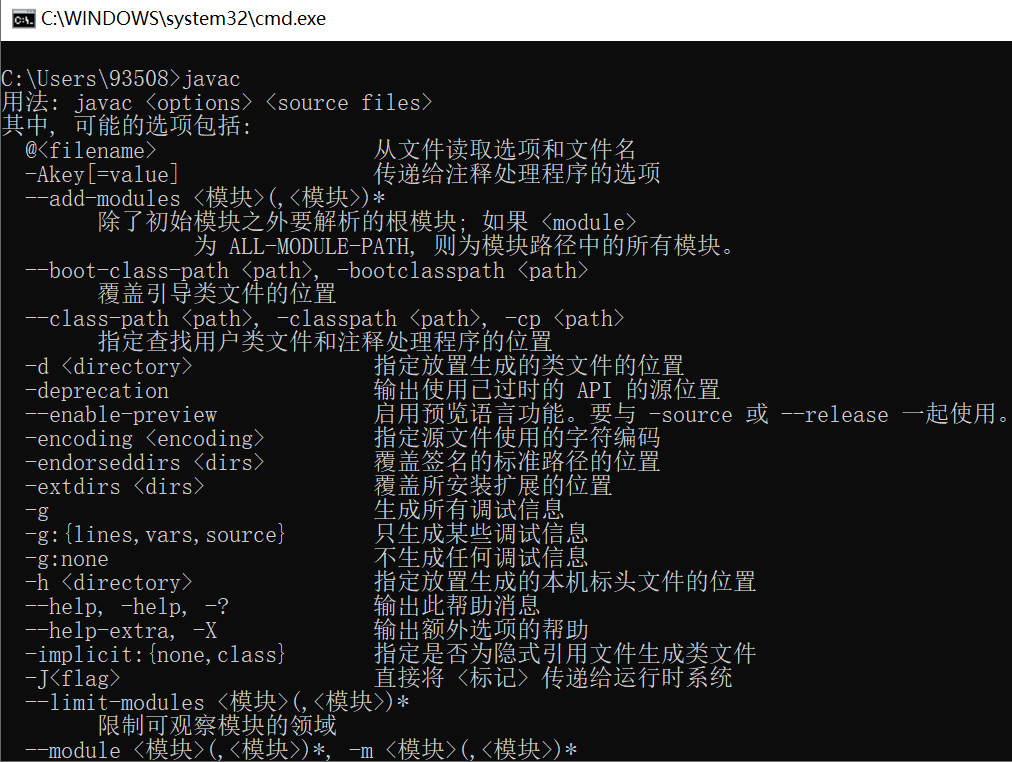
在Path变量中添加如下路径：

;%ANDROID\_SDK\_HOME%\platform-tools;%ANDROID\_SDK\_HOME%\tools

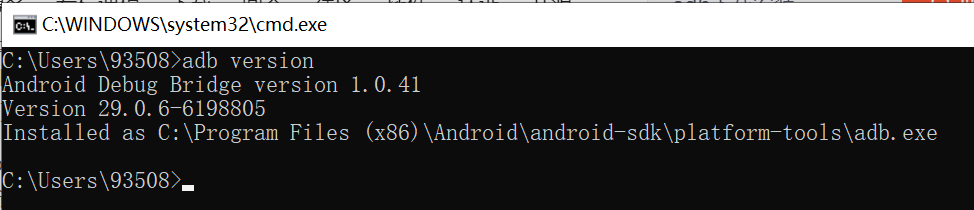
（5）检验JDK和adb安装是否成功

① 打开cmd，输入java和javac





② 打开cmd，输入adb version



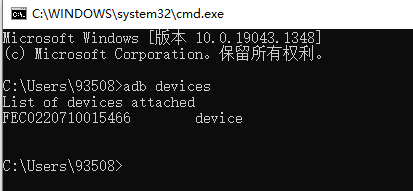
至此，所有配置均已完成。

# 3 Monkey测试过程

## 3.1 测试步骤

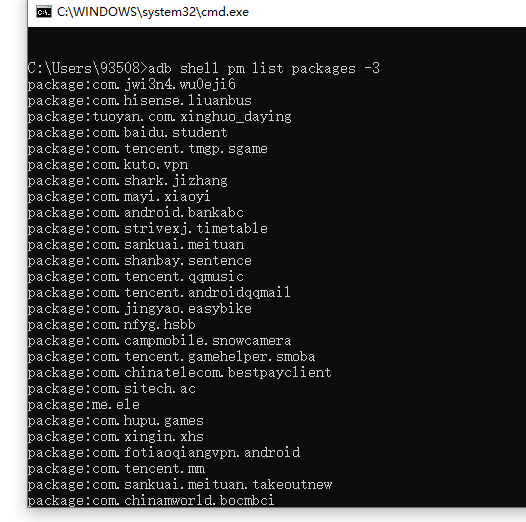
在手机中：设置—关于手机—连续点击“版本号”7 次，即可进入到开发者模式。然后可以到：设置—系统和更新—开发人员选项—调试里打开USB调试并授予adb相关权限。

USB线连接安卓机和电脑，打开cmd，输入adb devices查看设备连接情况。

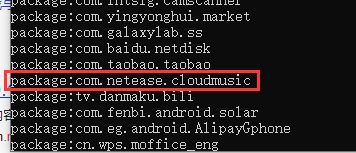


由上图可知，设备连接成功。

输入adb shell pm list packages -3 查看手机上所有的第三方安装包



最终我们选择网易云音乐APP进行测试。



文件安装名为：com.netease.cloudmusic

输入：adb shell monkey -p com.netease.cloudmusic -v -v -v 1000 > C:\Users\93508\Desktop\test.out

代表打开网易云音乐APP并进行随机操作1000次，同时将日志log命名为test.out文件存储到对应路径中。

具体日志结果和录制视频可见压缩包中文件夹Origin Monkey。

## 3.2 结果分析

我们对于操作日志进行分析：

最开始输出了相关参数bash arg以及Monkey执行的seed值、

执行次数count和测试包名AllowPackage。

:IncludeCategory: android.intent.category.LAUNCHER 和:IncludeCategory: android.intent.category.MONKEY是Monkey默认的参数，如果需要添加其它的Category，使用-c 参数添加。

  bash arg: -p

  bash arg: com.netease.cloudmusic

  bash arg: -v

  bash arg: -v

  bash arg: -v

  bash arg: 1000

:Monkey: seed=1639659218185 count=1000

:AllowPackage: com.netease.cloudmusic

:IncludeCategory: android.intent.category.LAUNCHER

:IncludeCategory: android.intent.category.MONKEY

带有NOT USING的均代表该安装包未被测试。

// Selecting main activities from category

android.intent.category.LAUNCHER

//   - NOT USING main activity com.android.deskclock.AlarmsMainActivity (from package com.android.deskclock)

//   - NOT USING main activity com.huawei.gallery.app.GalleryMain (from package com.android.gallery3d)

//   - NOT USING main activity com.android.mediacenter.PageActivity (from package com.android.mediacenter)

//   - NOT USING main activity com.android.mms.ui.ConversationList (from package com.android.mms)

//   - NOT USING main activity com.android.settings.HWSettings (from package com.android.settings)

//   - NOT USING main activity com.huawei.android.findmyphone.ui.findphone.FindMyPhoneActivity (from package com.huawei.android.findmyphone)

带有+ USING的代表该安装包被测试。

//   + Using main activity

com.netease.cloudmusic.activity.LoadingActivity (from package com.netease.cloudmusic)

包含了Seed种子序号以及各种操作的占比。

// Seeded: 1639659218185

// Event percentages:

//   0: 15.0%

//   1: 10.0%

//   2: 2.0%

//   3: 15.0%

//   4: -0.0%

//   5: -0.0%

//   6: 25.0%

//   7: 15.0%

//   8: 2.0%

//   9: 2.0%

//   10: 1.0%

//   11: 13.0%

查看源码可知各序号代表操作如下：

    public static final int FACTOR\_TOUCH        = 0;

    public static final int FACTOR\_MOTION       = 1;

    public static final int FACTOR\_PINCHZOOM    = 2;

    public static final int FACTOR\_TRACKBALL    = 3;

    public static final int FACTOR\_ROTATION     = 4;

    public static final int FACTOR\_PERMISSION   = 5;

    public static final int FACTOR\_NAV          = 6;

    public static final int FACTOR\_MAJORNAV     = 7;

    public static final int FACTOR\_SYSOPS       = 8;

    public static final int FACTOR\_APPSWITCH    = 9;

    public static final int FACTOR\_FLIP         = 10;

    public static final int FACTOR\_ANYTHING     = 11;

0：触摸事件百分比，参数--pct-touch

1：手势事件百分比，参数--pct-motion

2：二指缩放事件百分比，参数--pct-pinchzoom

3：轨迹球事件百分比，参数--pct-trackball

4：屏幕旋转事件百分比，参数--pct-rotation

5：指定（权限）事件百分比，参数--pct-permission

6：基本导航事件百分比，参数--pct-nav

7：主要导航事件百分比，参数--pct-majornav

8：系统导航事件百分比，参数--pct-syskeys

9： 启动activity事件百分比，参数--pct-appswitch

10：键盘翻转事件百分比，参数--pct-flip

11：其他事件百分比，参数--pct-anyevent

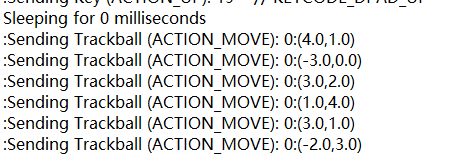
Switch表示当前执行的activity，注释中的Allowing则代表activity成功跳转（切换）。

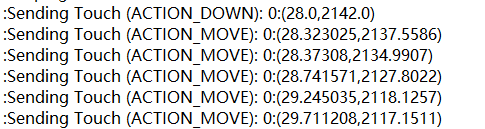
:Switch:

#Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.netease.cloudmusic/.activity.LoadingActivity;end

    // Allowing start of Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.netease.cloudmusic/.activity.LoadingActivity } in package com.netease.cloudmusic

接下来的内容就是各种操作的具体信息：





Monkey测试结束后，日志会在最后打印出当前执行事件的次数Events injected和执行时间elapsed time。Dropped代表执行失败的操作次数。

Events injected: 1000

:Sending rotation degree=0, persist=false

:Dropped: keys=0 pointers=0 trackballs=0 flips=5 rotations=0

## Network stats: elapsed time=10614ms (0ms mobile, 0ms wifi, 10614ms not connected)

// Monkey finished

同时对测试结果的分析还需要搜索以下几个关键词：

ANR、 Exception、 Null、 Error、 crash（Fatal）

　　① ANR问题：在日志中搜索“ANR”

　　ANR=Application Not Responding

　　在Android上，若应用程序有一段时间响应不够灵敏(5秒内没有输入响应事件)，则系统会向用户显示一个对话框，这个对话框称作应用程序无响应（ANR：Application Not Responding）对话框。

　　② 崩溃问题：在日志中搜索“Exception”

　　Null 指针异常：空指针异常主要是有 NullPointerException 异常提示；

　　Debug 异常：出现的是 IllegalStateException 异常；

　　低内存异常：出现的是 OutOfMemoryError 异常；

　　操作无响应异常：TimerOut（KeyDispatchingTimedOut 提示）；

　　RuntimeException 异常（操作无响应或应用的服务无法启动或连接）；

　　StaleDataException/readException 异常（android 数据库出现异常）；

　　IllegalArgumentException 异常（向函数传递了一个不正确或不合法的参数）；

# 4 Monkey测试优缺点

（1）优点：

Monkey的优点是可以对APP进行各种混乱操作，能有效的对

APP进行压力测试 ，观察APP是否有Crash或者异常日志。

（2）缺点：

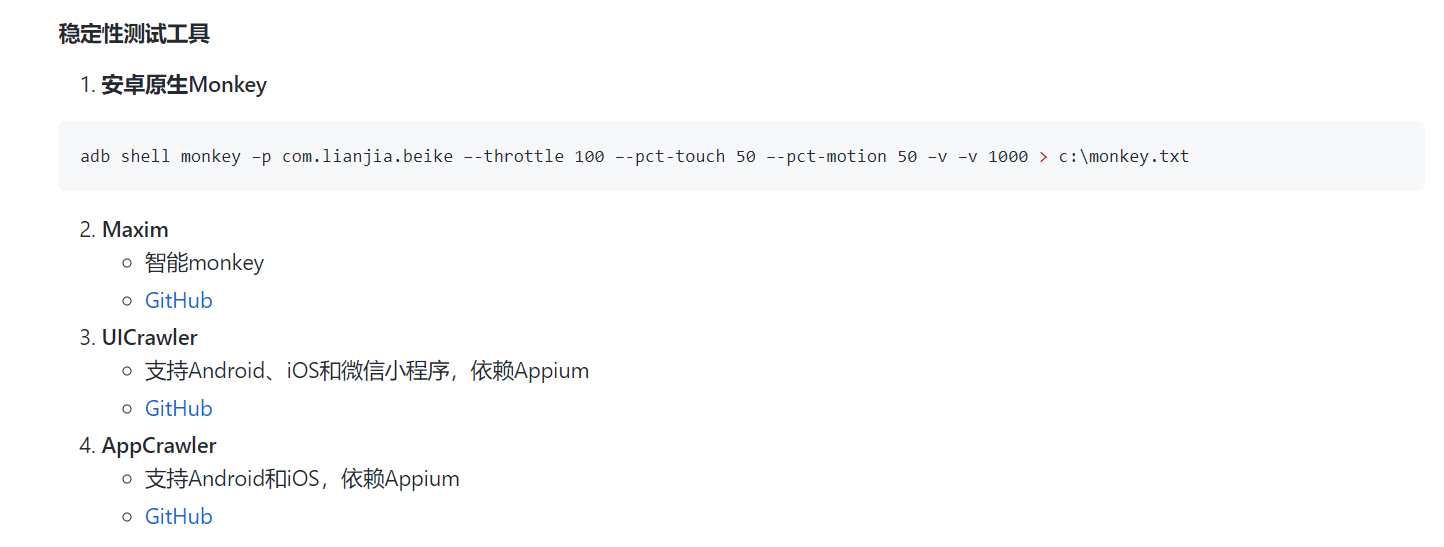
Monkey虽然可以根据一个指定的命令脚本发送按键消息，但它不支持条件判断，也不支持读取待测界面的信息来执行验证操作。

# 5 智能Monkey测试

## 5.1 Maxim & UICrawler

常规的Monkey测试存在着若干问题，比如遍历界面有限，路径回环等。

所以我对于更人性化，效率更高的智能Monkey测试进行了调研。最终在Github上发现了对于智能Monkey测试第三方库/软件的总结列表 <https://github.com/GitDzreal93/dev-tester/blob/fdc4c02569976b099e1a6546aa77878b38f579ad/page/app_testing/basics.md>



经过筛选对比，最终我选择Maxim & UICrawler来进行第二部分的实验。（其中Maxim是基于遍历规则的高性能Android Monkey，适用于真机/模拟器的APP UI压力测试，UICrawler则是基于Appium实现，可面向Android和IOS移动App，或H5或微信等应用的，基于元素识别的UI自动化遍历测试。）

原始Monkey有两大缺点亟需改善：

① 有可能点击到页面的顶部，下拉通知栏，或者点击到底部返回键菜单栏（尤其是在模拟器里执行的时候），会有各种误点操作；

② 纯随机，没有任何遍历算法加持。

而Maxim是基于原始Monkey实现的二次开发，相较于原始

Monkey，Maxim新增以下功能：

① 多种随机测试模式：DFS（深度遍历），MIX模式（Monkey随机测试+控件识别），troy模式（按照控件选择器进行遍历）；

② 可进行执行时长设定；

③ 可以自定义activity黑白名单；

④ 兼容多个Android版本；

⑤ 防跳出，防休眠，防假死。

同时UICrawler相较于原始Monkey，它也具有以下优势：

① 基于深度优先的原则，点击UI上的元素。当发现Crash时会提供操作步骤截图及相应的log；

② 元素遍历结束或按下Ctrl + C键会生成HTML测试报告。测试报告中汇集了测试统计、log及截图的相应信息；

③ 同一个元素只会点击一次(白名单中的元素可能会被多次点击)；

④ 支持对输入框的文本输入(需在文件中进行配置 INPUT\_TEXT\_LIST)；

⑤ 支持更高级的配置，可配置用户名、密码等。

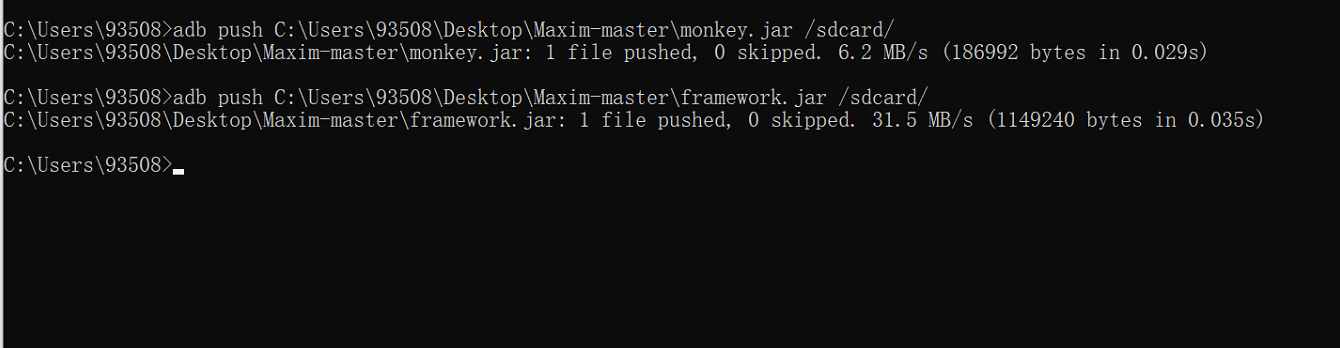
## 5.2 Maxim环境配置

① 下载<https://github.com/zhangzhao4444/Maxim>中的压缩包；

② USB连接安卓机，打开cmd，先输入adb devices检验设备是否连接成功。确认成功后执行以下2条命令将monkey.jar和framework.jar安装到安卓机的/sdcard 文件夹中：

adb push C:\Users\93508\Desktop\Maxim-master\monkey.jar /sdcard/

adb push C:\Users\93508\Desktop\Maxim-master\framwork.jar /sdcard/



至此环境配置成功，我们接下来可以使用命令行来运行Maxim。

## 5.3 命令行运行Maxim

打开cmd，执行如下命令：

adb shell CLASSPATH=/sdcard/monkey.jar:/sdcard/framework.jar exec app\_process /system/bin tv.panda.test.monkey.Monkey -p com.netease.cloudmusic --uiautomatormix --running-minutes 1 -v -v > C:\Users\93508\Desktop\test2.out

其中：

① tv.panda.test.monkey.Monkey 为主调入口 ，无需修改；

② -p com.netease.cloudmusic 待测APP；

③ --uiautomatormix 混合模式；（70%控件解析随机点击，其余30%按原Monkey事件概率分布）

④ --running-minutes 1代表运行时间为1分钟；

⑤ 运行日志test2.out会存储至C:\Users\93508\Desktop

具体日志结果和录制视频可见压缩包中文件夹Maxim。对比原始Monkey测试，可发现Maxim控制下的智能Monkey测试更加灵活，较少出现从指定APP返回桌面的情况，且各种操作触发的事件和遍历的界面也比原始Monkey更多，基本未出现路径回环。

## 5.4 分析日志

日志开头展示了Maxim版本号，安卓标志，Seed种子序号以及执行操作次数count = 1000。

version:1.0.21.0612.2259

            ..                    ..:.

           ..t..     Maxim      .h..

            ..i..  . ....... .  ..si

              sandroidmonkeystreste

              .sttoolMaxim.thistool.

           .helpyoutofoundappcrash.thi

         ..stooluserandomordfsalgorithm:.

        .smonkeyMaxim machine gunmonkey..

      ..monkey,..,monkeymonkeym,..,onkeym.

      .onkeymon..keymon.keymonke..ymonkeym.

      ,onkeymonkeymonkeymonkeymonkeymon.key.

     .monkeymonkeymonkeymonkeymonkeymonkeym.

     .onkeymonkeymonkeythxforphdguxiaotian.:

     .anythingUcancontactpanda\_zzaqq77227005

      .                                   .

      .                                   .

[Maxim] // Monkey: seed=1639829207807 count=1000

这里与原始Monkey同理， 带有NOT USING的均代表该安装包未被测试。

[Maxim] // Selecting main activities from category

android.intent.category.LAUNCHER

[Maxim] // - Not Using main activity com.android.deskclock.AlarmsMainActivity (from package com.android.deskclock

[Maxim] // - Not Using main activity com.huawei.gallery.app.GalleryMain (from package com.android.gallery3d

[Maxim] // - Not Using main activity com.android.mediacenter.PageActivity (from package com.android.mediacenter

[Maxim] // - Not Using main activity com.android.mms.ui.ConversationList (from package com.android.mms

[Maxim] // - Not Using main activity com.android.settings.HWSettings (from package com.android.settings

[Maxim] // - Not Using main activity com.huawei.android.findmyphone.ui.findphone.FindMyPhoneActivity (from package com.huawei.android.findmyphone

[Maxim] // - Not Using main activity com.huawei.browser.Main2 (from package com.huawei.browser

带有+ USING的代表该安装包被测试。

[Maxim] // + Using main activity

com.netease.cloudmusic.activity.LoadingActivity (from package com.netease.cloudmusic)

这里显示了安装包的白名单和黑名单，安卓机基本信息以及随机测试的策略。（这里为默认的混合模式uiautomatormix，70%控件解析随机点击，其余30%按原Monkey事件概率分布）

[Maxim] // Add Package To PackageWhiteList : com.oppo.camera

[Maxim] // Add Package To PackageWhiteList : com.android.packageinstaller

[Maxim] // Add Package To PackageWhiteList : com.android.documentsui

[Maxim] // AllowPackage: com.netease.cloudmusic

[Maxim] // AllowPackage: com.oppo.camera

[Maxim] // AllowPackage: com.android.packageinstaller

[Maxim] // AllowPackage: com.android.documentsui

[Maxim] // IncludeCategory: android.intent.category.LAUNCHER

[Maxim] // IncludeCategory: android.intent.category.MONKEY

[Maxim] // Special Event:

[Maxim] // Black Widget List:

[Maxim] // InputMethod ID: InputMethodInfo{com.baidu.input\_huawei/.ImeService, settings: com.baidu.input.ImeMainConfigActivity}.id

[Maxim] // Device Info: HUAWEI-TAS-AN00, Android Version: 10

[Maxim] strategy : uiautomator-Mix.

这里显示了各类操作的占比。

// Event percentages:

//   TOUCH : 62.5%

//   MOTION : 10.0%

//   PINCHZOOM : 4.0%

//   TRACKBALL : 5.0%

//   ROTATION : 5.0%

//   PERMISSION : -0.0%

//   NAV : -0.0%

//   MAJORNAV : -0.0%

//   SYSOPS : -0.0%

//   APPSWITCH : 1.0%

//   FLIP : 1.0%

//   ANYTHING : -0.0%

//   BACK : 10.0%

//   MUTE : 1.0%

//   SCHEMA : -0.0%

//   NULLINTENT : 0.2%

//   RESET : 0.3%

这里显示了各操作事件的执行开始时间，并且从Switch正式开始各种随机操作。

[Maxim] // event0, 2021-12-16 18:12:04.539

[Maxim] // Switch: #Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.netease.cloudmusic/.activity.LoadingActivity;end

[Maxim] // the top activity is com.netease.cloudmusic.activity.LoadingActivity, phone launcher activity is com.huawei.android.launcher.unihome.UniHomeLauncher

[Maxim] // the top activity is com.netease.cloudmusic.activity.LoadingActivity, phone launcher activity is com.android.settings.FallbackHome

[Maxim] // Allowing start of Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.netease.cloudmusic/.activity.LoadingActivity } in package com.netease.cloudmusic

[Maxim] // : debug, currentActivity is com.netease.cloudmusic.activity.LoadingActivity

[Maxim] // event1, 2021-12-16 18:12:04.562

[Maxim] // Sleeping for 0 milliseconds

[Maxim] // event1, 2021-12-16 18:12:04.563

[Maxim] // Sleeping for 6000 milliseconds

[Maxim] // event1, 2021-12-16 18:12:10.566

[Maxim] // Sleeping for 0 milliseconds

[Maxim] Random Number = 0.44295365

[Maxim] random hit touch

[Maxim]  // Click: null, android.widget.FrameLayout, (540.0,1071.0)

[Maxim] // event1, 2021-12-16 18:12:10.626

[Maxim] :Sending Touch (ACTION\_DOWN): 0:(540.0,1071.0)

[Maxim] // event2, 2021-12-16 18:12:10.636

[Maxim] :Sending Touch (ACTION\_UP): 0:(545.8,1077.7322)

日志的最后部分对本次智能Monkey随机测试进行了数据总结：操作总数，Dropped操作数，Activity of Coverage（Activity生命周期）等等。

[Maxim] // Try to stop package com.netease.cloudmusic

[Maxim] // event1312, 2021-12-16 18:13:00.369

[Maxim] // Sleeping for 6000 milliseconds

[Maxim] // realtime=143321757, mEndTime=143319909

[Maxim] // Events injected: 1312

[Maxim] // Monkey is over!

[Maxim] // Sending rotation degree= 0, false

[Maxim] // Unaccounted for: 61884

[Maxim] // Tested activities 17

[Maxim] // 1 - com.netease.cloudmusic.activity.LoadingActivity

[Maxim] // 2 - com.netease.cloudmusic.activity.MainActivity

[Maxim] // 3 - com.netease.cloudmusic.activity.PlayerActivity

[Maxim] // 4 - com.netease.cloudmusic.activity.ArtistActivityV3

[Maxim] // 5 - com.netease.cloudmusic.music.biz.rn.activity.CloudMusicRNActivity

[Maxim] // 6 - com.netease.cloudmusic.activity.PlayListActivity

[Maxim] // 7 - com.netease.cloudmusic.music.biz.mlogdetail.MLogVideoActivity

[Maxim] // 8 - com.netease.cloudmusic.music.biz.rn.activity.MainProcessRNActivity

[Maxim] // 9 - com.netease.cloudmusic.activity.H5CustomViewActivity

[Maxim] // 10 - com.netease.cloudmusic.activity.RedirectActivity

[Maxim] // 11 - com.netease.cloudmusic.activity.ImageBrowseActivity

[Maxim] // 12 - com.netease.cloudmusic.activity.AlbumActivity

[Maxim] // 13 - com.netease.cloudmusic.music.biz.comment.activity.CommentActivity

[Maxim] // 14 - com.netease.cloudmusic.activity.AddToPlayListActivity

[Maxim] // 15 - com.netease.cloudmusic.activity.MLogAggregationMusicActivity

[Maxim] // 16 - com.netease.cloudmusic.activity.SharePanelActivity

[Maxim] // 17 - com.netease.cloudmusic.module.musicshare.PlayerMusicShareActivity

[Maxim] // Activity of Coverage: 3.1858406

[Maxim] // How many Events Dropped: keys=0, pointers=100, trackballs=0, flips=0, rotations=0

[Maxim] // Monkey finished

本想继续结合Appetizer来进行图形化界面的Monkey测试。但插桩后的安装包在手机上安装完成后，一直打开就闪退。虽然网站指引说将安装包重新签名即可。但我用了各种方式对其进行重新签名（Jarsigner，apksigner，Android Studio），均无法打开。只好作罢。

所以为了更好的体现智能Monkey测试的优越性，我决定继续调研UICrawler。

## 5.5 UICrawler环境配置

（1）安装Appium

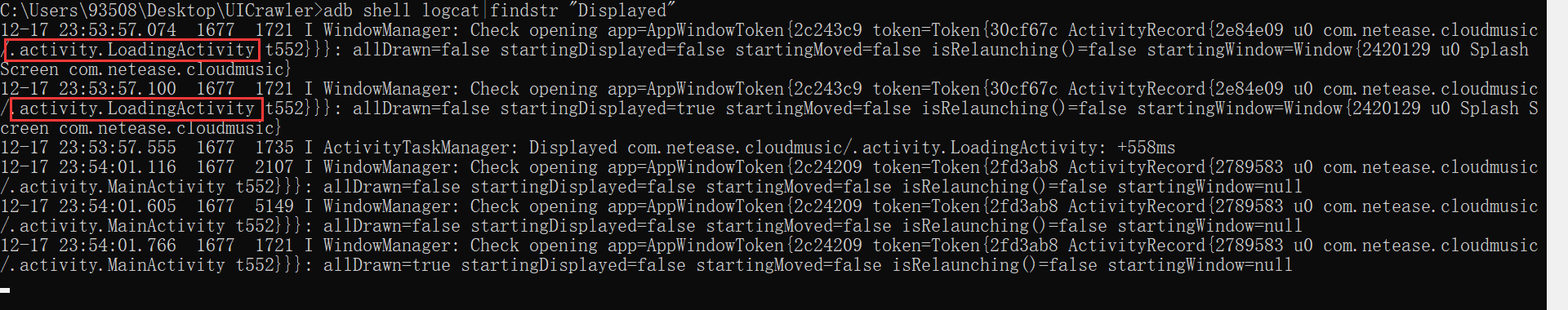
① 进入官网<http://appium.io/>下载Windows客户端的GUI，按照默认配置安装即可；

② 打开Appium并设置IP为本机IP：127.0.0.1。

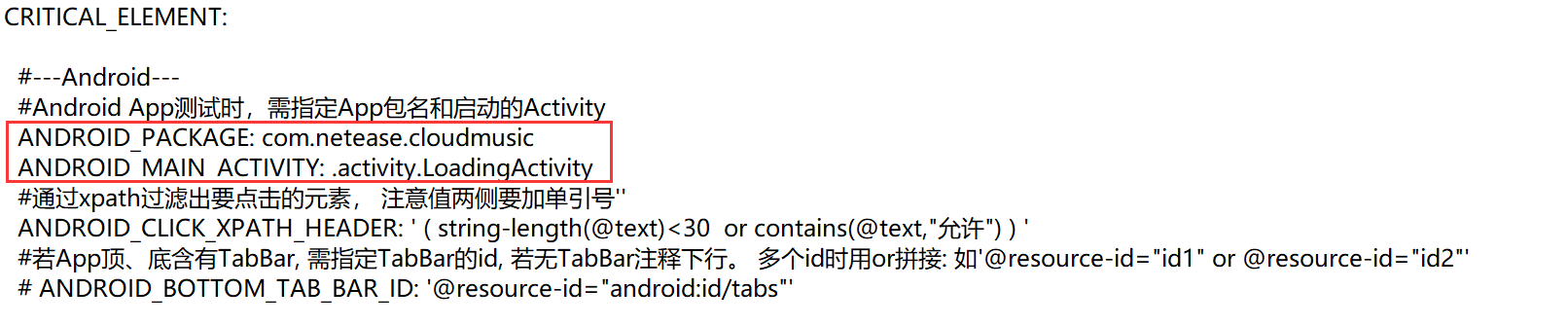
（2）打开[https://pan.baidu.com/s/1mNci6SWNHPuLj\_mvrfgIbg?errmsg=Auth+Login+Params+Not+Corret&errno=2&ssnerror=0#list/path=%2F，下载UICrawler.jar](https://github.com/lgxqf/UICrawler下载UICrawler.jar)（内含配置文件config.yml）

（3）USB线连接安卓机和电脑，打开cmd，输入：

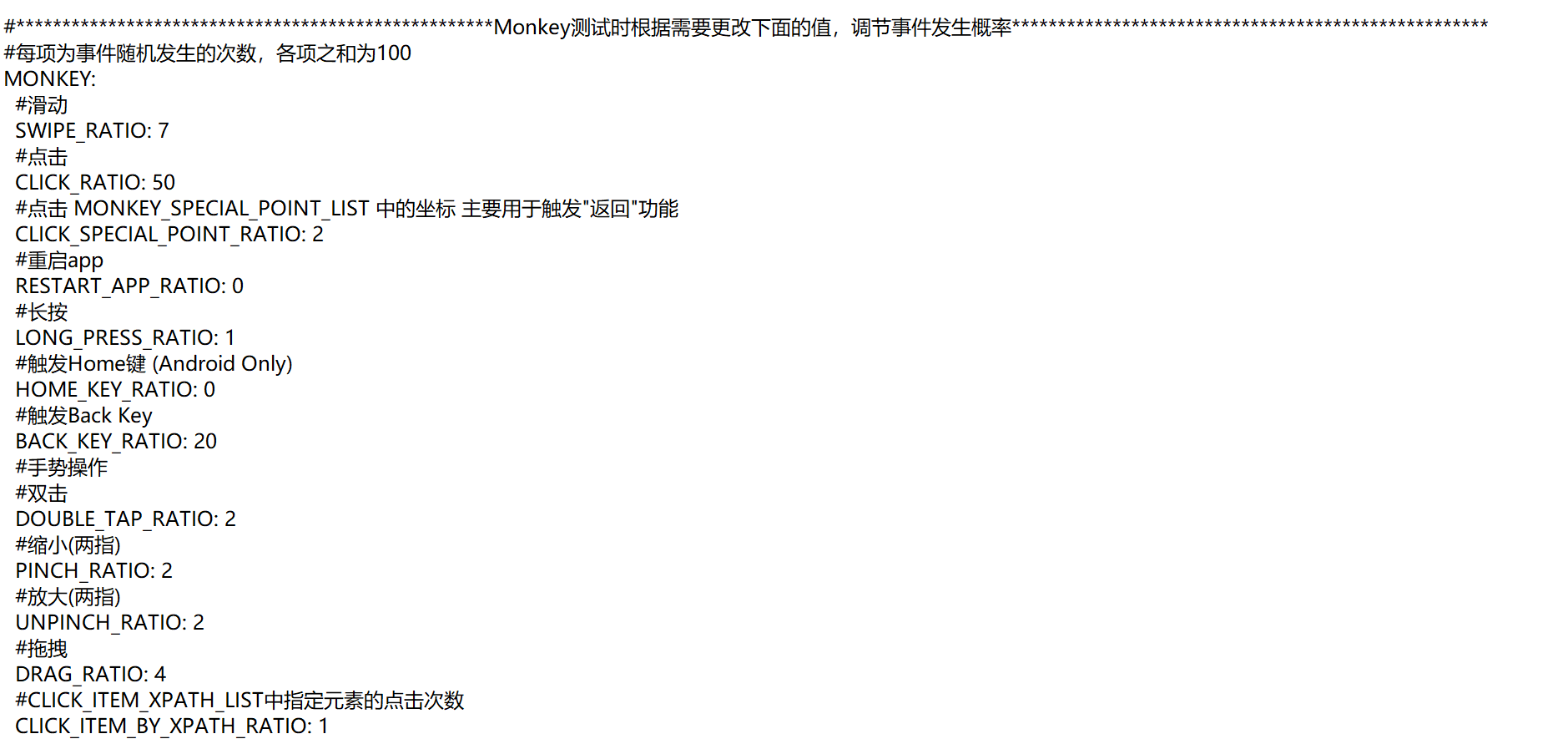
adb shell logcat|findstr "Displayed"查看打开网易云音乐所对应的activity。



所以可将config.yml中内容修改如下：



同时对于其他操作方面的内容进行以下修改（见下页）：



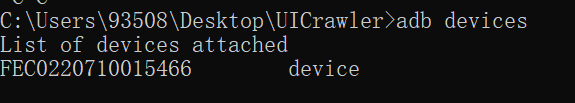


至此，环境配置已结束。

## 5.6 UICrawler运行结果

USB线连接安卓机和电脑后，在UICrawler-2.3.7.jar所在的文件夹打开cmd，执行如下命令：

adb devices查看设备ID：



再输入以下指令进行智能Monkey测试：

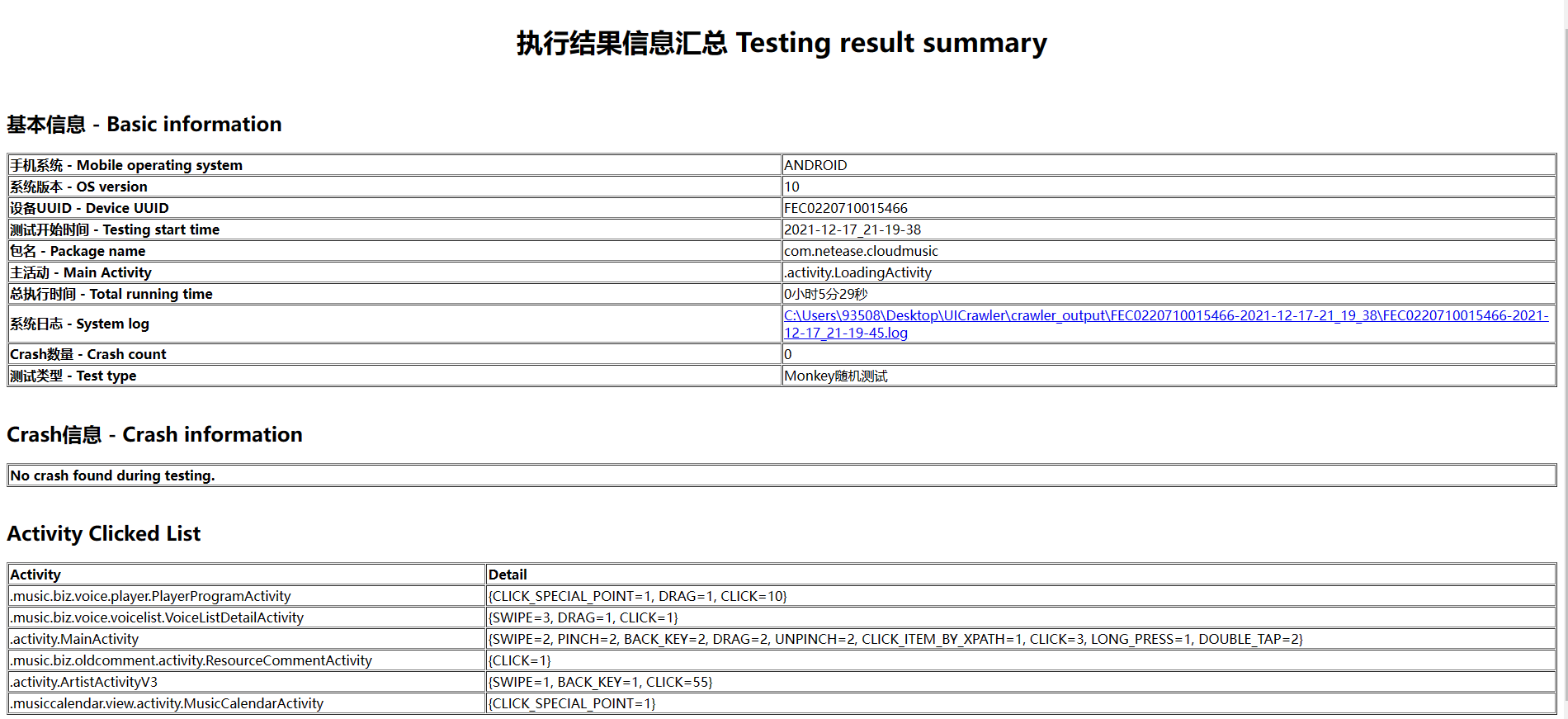
java -jar UICrawler-2.3.7.jar -u FEC0220710015466 -m

第一次执行时，手机端会自动安装两个Appium的相关APK：Appium Settings和io.appium.uiautomator2.server，同意即可。

之后程序会进行各种随机操作，UICrawler相较于原始Monkey测试的优势之处在于：它会自动截图每次随机操作（以红点表示点击），并对全过程进行录屏，同时生成log日志以及HTML版本的测试报告。使我们能够更加详细的了解到此次随机测试的各种信息。

截图，视频，log日志，HTML测试报告可见压缩包中的文件夹UICrawler\crawler\_output。

HTML测试报告如下（见下页）：



可见该测试报告比原始Monkey测试的log日志更为简洁易懂。

同时从视频可看出，UICrawler控制的Monkey测试遍历界面更多，无效操作更少，效率高于原始Monkey测试。

# 6 实验感想

本次实验让我对于移动端的稳定性测试有了初步且全面的了解。

明白了Monkey测试的本质以及当前互联网上更为优秀的一些第三方开发的智能Monkey测试并能灵活运用。

本次实验主要困难集中在前期环境的配置。由于我之前对于安卓环境的开发配置一无所知，所以踩了很多坑。好在通过大量的时间花费和资料查询逐一解决。唯一遗憾的是本次实验没有实现Appetizer图形化界面与Maxim的结合。一直卡在插桩安装包闪退这一步，即使重新生成签名也无济于事，最终被迫放弃。但我仍然实现了命令行调用Maxim来执行智能Monkey测试，并继续调研了另一种方法UICrawler来补充自己对于本次实验的理解。

通过此次实验，我对于安卓环境开发，移动端稳定性测试均有了更深一步的了解，这为我以后的研究打下了基础。