DAY01

1. 手机测试背景 无限通讯设备、移动设备
2. 手机软硬件的提示
3. 硬件
   * 1. CPU：中央处理器

高通骁龙、联发科、华为海思麒麟、苹果A14、A15、三星、、、

1. 运存：运行内存，开机时临时加载的数据，运存越大，同时跑的进程越多

早期：1G 2G 4G

近期：4G 8G 12G 16G

1. 机身存储：长期、大量保存文件数据

空间越大，存储的文件越多

早期：16G 32G 64G

近期：128G 256G 512G 1T

1. GPU：图形处理芯片 为显示效果提速 类似电脑的显卡
2. 网络天线：蜂窝（运营商信号）

WIFI、蓝牙、红外、NFC（近距离无限技术）、、、

1. 摄像头：前置、后置（主摄像头）

分为：单摄、多摄（正常、广角、长焦、微距）

核心感光元件：CMOS 外键光信号，变为数码图像

1. 传感器：将外界信号转化为手机内部信号

触摸屏（输入、输出）、重力感应、陀螺仪、光学感应、距离感应、指纹识别、按压识别、、、

了解：目前主要关注软件测试，硬件有电子厂家生产和测试，比如华为、小米（工程机的硬件测试）

1. 软件：操作系统、应用软件app
2. 操作系统：

IOS：苹果公司

Android：Google公司 底层：Linux内核+Java开发语言（Oracle SUN）

原生系统、定制化：小米、华为、三星、OPPO、、、

WP：Windows Phone 微软 市场份额极少

鸿蒙：HarmonyOS：华为 兼容Android 物联网特性强

1. 应用软件：满足不同功能的App

功能繁多，界面简约而不简单！ Application应用程序

1. 手机的大量普及，移动互联网用户大量增加！

早期：网络购票、购物、使用PC

近期：绝大部分使用手机 80%以上

1. 用户对于手机的依赖程度高

娱乐：王者荣耀、吃鸡、穿越火线、、、、

生活：外卖—美团、饿了么、、、

影音-腾讯视频、爱奇艺、优酷、、

支付-微信支付、支付宝、美团支付、抖音支付、云闪付、、、

共同点：绑定银行卡

地图导航-百度地图、阿里的高德地图、腾讯地图、、、

对比：企业级应用 主流是PC端，通过电脑访问后台服务器

比如银行业务员，使用内部管理系统管理财务

比如OA办公自动化

有些业务适合移动办公，比如银行办理业务的App（平板）

不同终端分工完成业务

趋势：许多厂商研发的App逐步向移动端迁移。

早期：QQ、博客、、、、PC端为主

最有名：微信、抖音、、、 移动端为主

1. 关注软件测试思路
2. 简单理解：执行软件发现缺陷的过程：
3. 案例：出发到某地

方式：步行、打的、骑行、公交、飞机、高铁、、、

角度：线路最短、时间最短、躲避拥堵、、、 不同策略

目的：到达目的地

1. 总和应用

测试方式：手工、自动化 UI、接口、单元

测试方法：数据分析--等价类、边界值、、、

业务分析---场景法

测试目的：分析缺陷并解决，提高软件质量

1. 手机测试和PC端测试有何区别？
2. 操作方式不同

PC：键盘、鼠标、显示器

（输入） （输出）

手机：触摸屏，结合各种传感器

比如：扫描二维码

1. 操作系统的不同

PC：Windows、MacOS、Linux、Unix、、、

手机端：IOS、Android、鸿蒙等

特点：Android存在‘碎片化‘ 多而杂

考察App的兼容性问题

小米（MIUI）、华为（EMUI）、OPPO（ColorOS）、VIVO（FuntouchOS）、一加（氢OS 氧气OS）

1. 手机屏幕设计不同

之前：非全面屏

目前：全面屏（高屏占比）

异形屏：刘海屏、水滴屏、挖孔屏、、、

真全面屏：弹拉式摄像头（过度）、滑盖（过度）、屏下摄像头、、、

1. 手机屏幕尺寸不同：

PC：屏幕远大于手机 14寸 、27寸、60寸、、、

为UI界面布局提供便利

手机：5～6寸 方寸之间，界面如何布局？ 少而精，层层引导

比如购物车图标，打开新的界面（模块）

多少寸？ 算对角线

从左到右：宽度width 从上到下：height UI界面设计角度

1. 分辨率不同：一定范围的像素点的数量 越高越精细

手机端：早期720p 低端

中期1080p 中端

近期2k 2k+ 高端

规律：分辨率越高对系统资源消耗越大，比如CPU、内存、GPU、耗电、、、

1. 网络类型不同

PC端：有线网络--网线 水晶头

无限路由--WIFI

特点：固定、单一，稳定、高速（千兆以太网）

手机端：

移动网络（蜂窝cell 借助地面基站）

运营商：移动、电信、联通

网络类型：2G、3G、4G、5G、、、、

局域网：wifi 无线路由器

特点：信号多样性、网络可能会不稳定；

后期多考虑进行弱网测试！

1. 手机软件测试的定义：

在不同的网络、不同的操作系统、不同分辨率下，执行被测app，发现缺陷，提高软件质量的过程；

目前主要测试移动端（App客户端软件），也离不开后台服务器（Web服务器--安装接口程序、数据库服务器--长期存储海量数据）

Apache或Tomcat Mysql

目前环境：真机、模拟器、综合测试工具

1. 手机软件测试的流程：
2. 前期的准备工作

测试计划（宏观）->测试方案（具体）

->测试准备（测试数据、测试环境---工具、不同技术组合）

->测试分析、设计用例（不同操作流程描述）

项目、模块、子模块、功能点 根据业务数据分类管理

1. 测试的执行阶段

不同策略：

冒烟测试（monkey）、功能测试、接口测试、UI测试、兼容性测试、客户端性能测试（CPU、内存、耗电量、启动速度等）、网络类型测试、弱网测试、交叉事件测试、推送测试、稳定性测试（monkey）、服务器端性能测试（JMeter或LoadRunner模拟多用户访问服务器接口，压测后台服务器的性能）

-->迭代更新、回归测试（建议使用自动化技术）

-->上线测试（模拟客户环境下部署，展开系统测试）

-->上线发布 比如‘应用商城’

说明：手机测试中注重‘专项测试’

比如弱网测试、耗电量测试、交叉事件测试、、、

1. 冒烟测试
2. 目的：确认软件的主题功能是否正常，是否可以进行测试；

（可测性---能不能测）

1. 方式：

之前：手工测试 若干测试人员进行功能确认，耗费人力

目前：自动化测试

使用monkey工具代替自动发起不同的时间，随机执行；

能够实现用例的快速覆盖、确认（对运行日志进行分析）

1. 作用：

monkey可以模拟各种常见的用户操作（触摸touch、点击click、滑动、拖拽、系统按键home back等）随机发送到手机设备中，进行大量的业务操作，确定软件的不稳定因素，比如：

Crash：崩溃、闪退

ANR：应用无响应 anr

Application Not Responding

1. 环境资源
2. 手机设备：

真机：推荐 更真实、但成本高

模拟器：

推荐使用‘夜神模拟器’、‘手机模拟大师’（谷歌的第三方产品）

Android自带的AVD（android虚拟设置）（谷歌提供的）

Android Virutal Device

1. 测试项目：家具酷App（家具电商App）
2. 测试工具：

Adb

Adb shell monkey 从PC端测试、调试手机

易测App：monkey工具 手机端测试

DAY02

1. 环境部署
2. 手机设备：安装模拟器（夜神模拟器、AVD）

夜神模拟器

注意1:如果显卡驱动过低，需要更新版本！

注意2:如果PC未设置VT选项，需要开启！

VT（Virtualization Technology） 虚拟化技术

需要开机时进入BIOS进行设置

不同电脑打开方式不同

比如VMware也需要VT的支持

注意3:可以不用更新，新版本会影响adb的自动连接；

1. 安装测试项目：家具酷App

安装包：测试项目、jiajuk.apk

安装方式：直接将PC的apk文件，拖拽到手机模拟器中

注意：先不要更新App，旧版本中容易发现bug

1. 测试工具：adb shell monkey

易测App的monkey （由阿里提供的小工具 安装在手机中）

易测：安装包

测试工具、手机测试工具\easytest2.2.4.apk

拖拽到模拟器中，自动安装

打开时需要输入信息 淘宝账号、邮箱

提示：以root权限使用产品 永久允许

性能测试中，经常关注‘阈值’，设置报警边界，超过该值会有性能问题；

比如CPU使用率、内存使用率、网络传输率，如果过高，会产生性能瓶颈（资源不够用）

比如CPU使用率一般不要超过80%；

内存使用率不能超过80%。

Adb Android Debug Bridge

Android调试桥

PC----adb命令-----手机

adb位于SDK（软件开发工具包）中

包括了进行Android开发、测试的基本工具！

1. 安装Android工具（ADT，包括了SDK） 工具kit

Android开发工具Tool

<1> adt环境部署：

资源包：adt4\_2\_32bit.rar 600mb

Android4.2版本

Windows32位系统可用，64位系统向下兼容，也可用

目录结构：

1. eclipse目录：是一种IDE（集成开发环境）

集成了Java、Android开发、运行、测试、调试等工具；

目前目的：提供AVD的管理器（新建、启动模拟器）

Android虚拟设备

Android Virtual Device

Eclipse 日食 IBM公司 3000万$使用Java技术开发的IDE 免费

Sun 公司 Java产品 被Oracle 74亿$收购

1. sdk目录：软件开发工具包（Google-Android提供）

目的：提供开发和运行的环境，带有测试所需的常用工具

重要子目录：

platform-tools目录：平台工具

比如：adb.exe Android调试桥

进一步使用adb shell monkey

比如：aapt.exe 分析提取apk的基本信息

App的安装包

tools目录：工具

比如：emulator.exe 模拟器

monkeyrunner.bat Android功能自动化测试工具

使用Python脚本指挥手机自动操作

基于精确坐标点来控制！

Ui auto mator viewer.bat UI测试的辅助工具

能够获取手机界面的基本信息，比如坐标点位置

1. SDK Manager.exe : SDK管理器

目的：在线更新SDK版本、工具

前提：需要访问Google官网资源，目前受限

<2>JDK的环境部署：Java开发工具包

环境资源：jdk-6-windows-i586.exe

JDK1.6 Win32位系统可用

目的：Java开发工具包，为所有Java软件产品提供运行的支持；

比如：Eclipse，JMeter、Tomcat、、、

安装步骤：默认下一步，安装位置默认在C盘下（JDK的安装目录）

<3> 配置系统环境变量（重点、难点）

检查方式：每次修改完环境变量后要重启cmd才能生效。

Set 变量名=变量值 设置局部系统变量，临时有效，关闭cmd就失效

Set 系统变量名 查看系统变量的值

比如：

Set uname=tom 设置系统临时的、局部的环境变量，无法长期保存

Set uname 查看系统变量uname的值，目前是tom

Set uname=Mary 修改系统变量uname的值为Mary

比如：查看重要的环境变量的值

Set jajava\_home

查看JDK的安装目录

Set path 命令的搜索路径

Java\_version 查看java目录的版本号，要是1.6

配置步骤：

右击‘计算机’->属性->高级系统设置->高级->环境变量->系统变量 （注意：不要选‘用户变量’，作用范围有限）

如果没有就新建，如果存在可以修改

1. JDK的安装目录

变量名：JAVA\_HOME

变量值：（从实际目录拷贝路径名）

检查环境：重启cmd

1. 检查系统变量Path的值 命令的搜索路径

Set Path

1. 检查是否能找到java命令，并查看版本号

Java -version

比如：

D\>java -version

Java version ‘1.6.0’

1. 检查adb能否使用

Adb

1. 检查aapt能否使用

Aapt

5、检查monkeyrunner能否使用

monkeyrunner

Ctrl+d退出

1. AVD的创建和使用

SDK自带的Android虚拟设备

（模拟器）

<1>启动eclipse.exe (需要JDK环境）

<2> 打开AVD管理器：工具栏AVD Manager

<3>新建一个AVD：点击New

配置内容：

1. AVD name ：myAVD 虚拟设备名称
2. Device：设备信号、屏幕大小、分辨率、 和硬件配置有关

比如：第4项 3.7寸 480x800

3、Target：指定Android系统版本4.2

其他配置默认即可：

CPU、前后摄像头、运存、、、

配置完成，点击OK

启动AVD：选择已创建的AVD，点击‘start’启动、开始

在启动设置窗口中，点击‘launch’启动

打开cmd：adb devices 查看PC所管理的手机设备状态

List of devices attached

没有列表

说明：以上为Android自动化测试最基本的环境

JDK、SDK

命令：adb aapt monkeyrinner

使用的模拟器：AVD、夜神

1. adb工具（Android调试桥、联调测试工具）——重点
2. adb基本介绍

来源：sdk\platform-tools下的一个命令：adb。Exe

底层使用Java开发，由于对手机设备进行连接和调试；

Android Debug Bridge

PC---------adb---------手机设备

作用：PC端借助adb服务，连接访问手机（真机、模拟器-夜神、AVD），对手机进行自动控制和调试。

1. adb运行原理：是一种服务，默认端口？ 5037

查看端口号：netstat -ano ｜more 回车、空格、q

Netstat -ano｜findstr 5037

当前PC本机：端口号 PID



查看任务管理器：找到PId对应的进程名：adb.exe

原理：

PC端->启动adb->产生adb服务进程->手机端（1台或者多台）

默认端口5037 —>生成adb后台进程->控制手机设备

1. adb常用的子命令

<1>adb devices

作用：检查当前环境连接的手机设备、状态

语法：adb devices （没有参数）

Adb devices -l 详细列表list

说明：一台PC可以同时调试多个手机设备（真机、模拟设备）

DAY03

结果：

1. 手机设备显示设备名称，可以区分真机和模拟器

真机：随机十六进制数 作为唯一的标识 Android id

范围：0～9 a～f

比如：83e52f63a

模拟器：模拟器-端口号

1. 连接状态：

成功：devices 本意：设备 表示设备连接畅通

失败：offline 表示识别到，但离线

如果都不显示，表示未识别到手机设备

如果未找到设备，建议重新连接 或 重启

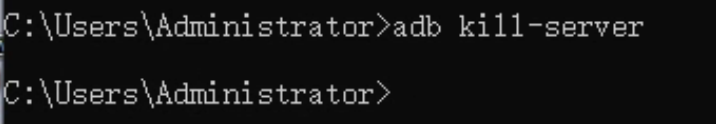
练习：连接其他模拟器，比如‘夜神’，属于第三方模拟器；

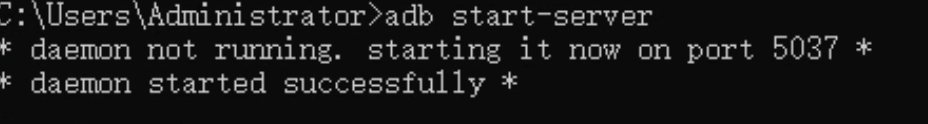
注意：要求先关闭杀毒软件服务，比如360安全卫士，以面连接手机失效！

如果启动adb时，遇到问题，可以管理adb服务

Adb kill-server 关闭adb服务

Adb start-server 开启adb服务





问题：未找到‘夜神’

解决方法：对于第三方模拟器，需要主动连接！

Adb connect 127.0.0.1:62001 （或52001）

注意：要开启‘夜神’模拟器

关闭杀毒服务

执行命令的参数要准确

补充：查看adb版本

Adb version

建议：关闭杀毒进程，避免和adb服务冲突；

关闭其他模拟器，只留下1台AVD 5554

小结：目前子命令

Adb devices 查看设备名，状态

Adb connect 设备名 主动连接某设备

Adb kill-server 关闭adb服务

Adb start-sercer 启动adb服务

Adb version 查看adb版本号

<2> adb push 推送上传

作用：从PC端将文件推送给手机端的指定位置

语法：adb push 参数1 参数2

总命令 子命令

参数1:PC端文件的路径名

参数2:手机系统中的路径名，比如：/data/

Android 系统内核就是Linux（微型版、定制版）

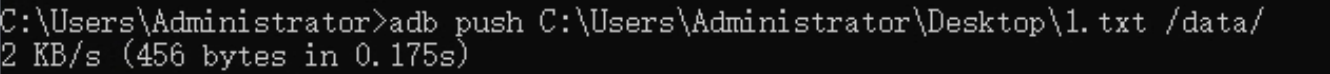
案例：将PC中的D：\的1.txt文件，推送到手机的/data/下

Adb push D:\1.txt /data/

补充说明：adb shell 子命令

进入Android Linux shell 环境（系统命令行）

可以执行常用的Linux命令（管理手机内部系统资源）



<3> adb pull 拉回，下载

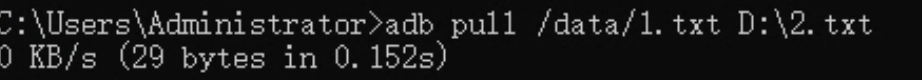
作用：将手机端文件拉回PC端；

比如手机内部日志文件，需要拉回PC进一步分析；

语法：adb pull 参数1 参数2

参数1:手机端文件路径名 比如：/data/1.txt

参数2:PC端文件的路径名 比如：D：\2.txt



练习：针对夜神模拟器进行文件的收发

<1>启动夜神模拟器

<2>主动连接模拟器：adb connect 127.0.0.1:62001

<3>查看设备连接状态

Adb devices

问题1:如果不只是一个设备，如何处理？

方法1:关闭多余的设备，只保留一个；

会针对目前唯一的设备进行收发；

方法2:adb使用选项，指定某设备访问（面试题）

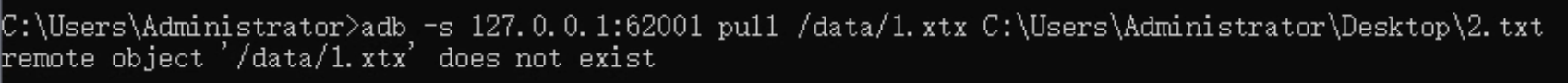
Adb -d 唯一真机（usb数据线连接的设备）

Adb -e 唯一模拟器

Adb -s 设备名 指定具体的设备



问题2:文件路径名不存在，会报错的！



远程 目标、对象 不存在

问题3:‘拉回’时无法在PC端创建文件，比如3.txt

方法1:提升权限

搜索cmd，右键‘以管理员身份运行’

方法2:提前在指定目录下新建一个同名的空文件，比如3.txt

后续通过覆盖，记录文件内容

adb小结：

选项：-d 唯一真机

-e 唯一模拟器

-s 设备id 指定具体某设备

小结：子命令

Adb devices （重点） 查看PC管理的设备名称、状态

Adb connect （重点） 让PC主动连接设备 指定 主机名：端口号

Adb shell （重点） 进入到Android 的shell环境 命令行环境

Adb push 在PC电脑和手机设备之间推送、拉回文件

Adb pull

Adb kill-server 关闭和开启adb 服务

Adb start-sercer

Adb version 查看adb版本号

<4> adb logcat 查看手机系统日志 （重点）

作用：针对手机端运行的日志进行抓取（系统日志），对执行步骤、问题进行跟踪，

作为手机测试的依据。

形式1:adb logcat

目的：实时持续跟踪日志并返回到PC端显示

看到最近的日志，并追加打印最新的日志消息；--没完没了

如何结束：ctrl+c 结束命令行当前正在执行的进程

比如：

级别/标签（应用的PID） ：问题的描述 近距离无线通信技术



技巧：如何根据Android的pid找到是哪个应用？（重要）

查看Android系统进程的列表

Adb shell ps 尽快访问shell，执行ps，并将结果返回Windows系统显示

（速去速回）

Adb shell ps｜findstr 进程号

Adb shell ps｜findstr 1667

进程号 应用的包名





形式2:adb logcat -d

目的：一次读取当前日志并返回

不会追加打印，而是立即结束监控进程； ---适可而止

形式3:adb logcat >D：\logcat1.txt

目的：将当前系统日志保存在文件中，并且可以自动追加内容，知道结束为止；

如何结束：ctrl+c 结束控制台当前进程

形式4:adb logcat -d > D:\logcat2.txt

目的：将当前系统日志一次写入文件中，结合输出重定向，对结果进文件保存；

软件=程序+文档（规范）

程序=数据结构+算法 都是开发方考虑的

如何存放数据 计算方法：如何获取和使用数据、操作步骤和方法

Adb logcat 日志管理的原理：环形队列

队列Queue 一种数据结构：先进先出 FIFO First In First Out

日志临时的内存空间是有限的，比如60kb，只能记录最近产生的日志；

新的日志会循环覆盖旧的日志：队尾 不断 覆盖 原先的队头

启发：查看日志要及时！避免被新日志覆盖

技巧：使用重定向方式，将内存中日志写入文件中保存

比如形式3:adb logcat >D:\logcat1.txt

形式5: adb logcat -d \*:E >D:\logcat3.txt

目的：根据日志级别、标签过滤和筛选日志

语法：标签:级别

\* 表示任意标签

\*:I 级别不低于I 等于或高于I

\*:W 级别不低于W

\*:E 级别不低于E

常用的日志级别：从严重到一般

Error（错误）->Warning（警告）->Info（信息）

技巧：查看命令的帮助手册

Adb logcat -help

所有级别：从低到高

V Verbose 最低级别 比较详细的细节，比较普遍

D Debug 调试级别 适合代码调试时监控执行过程，程序员编程时查看

I Info 一般消息

W Warn 警告，需要关注 （问题不大）

E Error 错误，重点关注！

F Fatal 严重，比如闪退等，要关注！

S Silent （supress all output） 屏蔽 把其他的结果屏蔽掉

比如： adb logcat -d hehe:E \*:S

表示：只过滤级别为E，标签为hehe的日志

其他信息都屏蔽掉

日志结果：E/hehe（进程id）：错误消息描述

练习：针对AVD自带的计算器进行日志跟踪和查看

技巧：清空日志队列，使用-c选项，对之前的环形日志队列进行清理

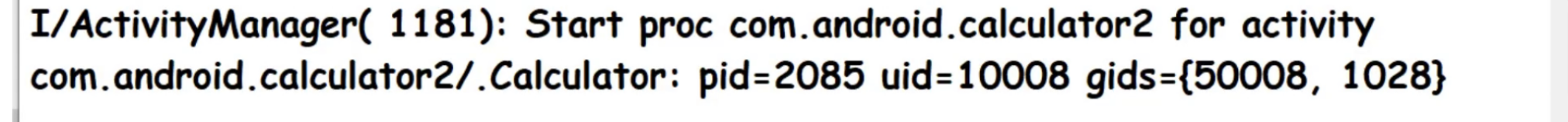
Adb logcat -c

需求1:实时读取系统日志，写入D:\logcat4.txt

通过ctrl+c结束

级别/标签 活动Acticity：就是手机App的某个界面、窗体，目前是计算器界面

信息/活动管理器 开始 进程 包名：就是计算器 活动



练习：确定App和进程id的关系

Adb shell ps 查看Android系统进程列表

Adb shell ps ｜findstr 进程id 根据pid找到对应的‘包名’对应某App

比如：adb shell ps ｜ findstr 1182

进程号Pid



需求2:一次获取级别不低于E的日志，保存在D：\logcat5.txt

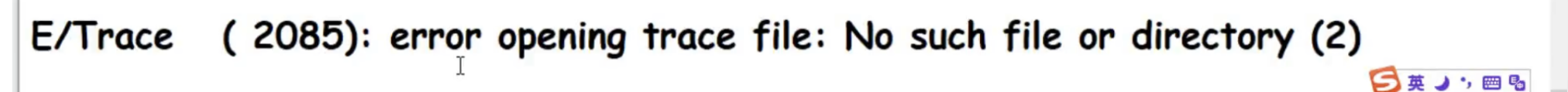
Adb logcat \*:E >

需求3:根据日志进行分析，寻找问题

思路1:确定进程号pid，找到与之有关的执行轨迹；

思路2:根据crach（崩溃、闪退）、anr（应用无响应）、exeption（异常）、error（错误）等作为线索搜索日志；

测试目的：根据日志查看和分析测试中的隐含的、容易遗漏的缺陷。



常识：不同手机系统App的安装包？

Android：apk 全称：Android application package

Android应用程序 包

IOS：ipa

<5> adb install 安装App

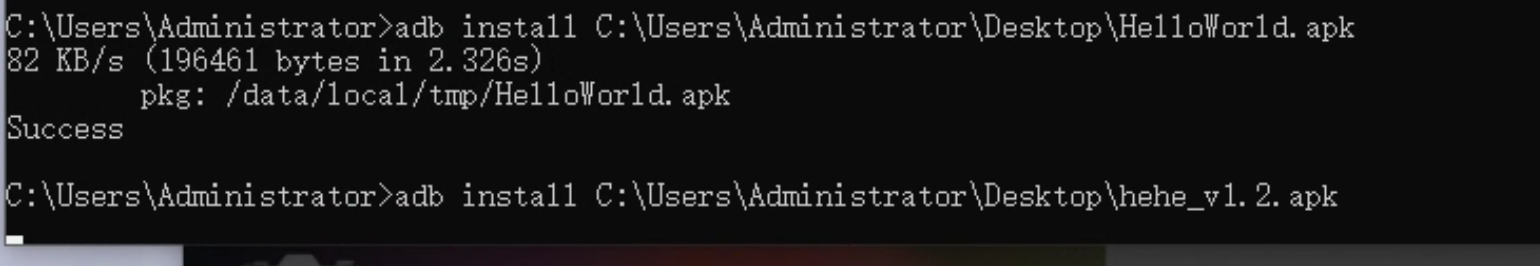
作用：通过命令行将apk安装包 安装到手机中，成为App

语法：adb install 参数1

参数1:PC端apk文件的路径名，推荐使用绝对路径

技巧：使用拖拽法，快速生成路径名

案例：安装HelloWorld.apk



安装过程原理：

上传apk文件（传输速度、文件大小、总耗时）

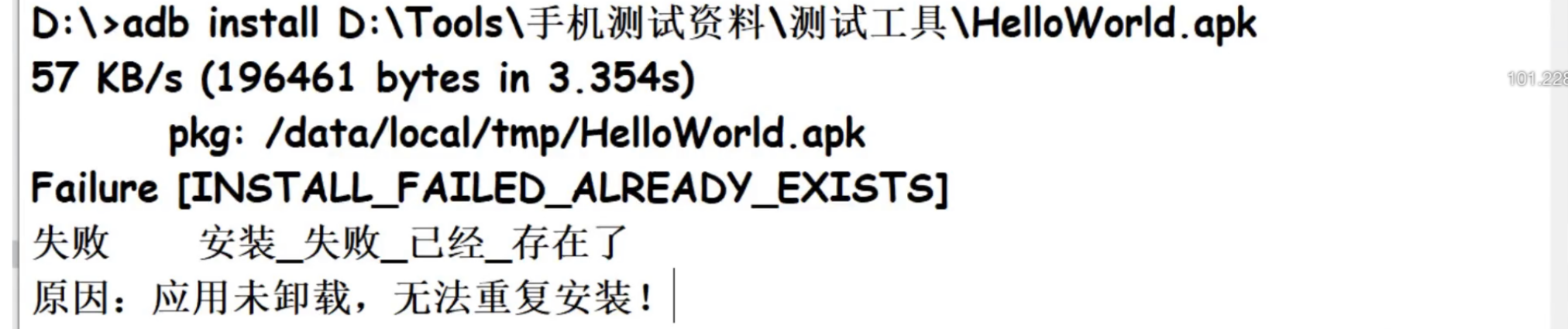
->手机端临时保存目录：/data/local/tmp/目录下

->执行安装，给出反馈：

Success 成功

其他 表示有问题

->删除临时安装文件



说明开发细节：每次点击按钮button时，都会执行记录日志的程序

Log.i(“mes”,”按钮button1被点击了～～”+系统时间):

日志 I级别 标签mes 消息：按钮button1被点击了～系统时间

测试时：根据条件捕捉日志，只捕捉级别为I，标签为mes的日志

标签：级别 其他任意：屏蔽

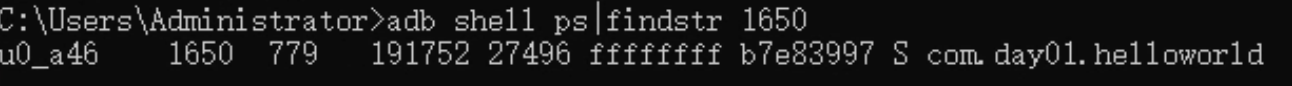
D:\>adb logcat mes:I \*:S > D:\logcat6.txt



根据pid查找哪个进程？

Adb shell ps ｜findstr 1650

Pid 应用的包名



建议：开发方在开发App时，在遇到关键问题（错误）时，适当打印输出日志信息，给定日志的级别、标签、提示信息（作为预期结果），供猴戏 测试时跟踪查看使用！

日志：特定的信号，反馈问题的方式。

测试目的：执行手机专项测试——安装与卸载测试

日常App版本不断更新，需要经常安装与卸载；

检查能否正常安装，安装后能否正常启动；

检查能否正常卸载，卸载是否彻底，是否有残留；

<6>adb uninstall 卸载App

作用：根据App的‘包名’自动卸载应用

语法：adb uninstall 参数1

参数1:App的“包名” 即是一种目录的层次结构，更是App的代号！

是“安装包”？ 不是

是开发时指定的包名？ 是

比如：HelloWorld 应用的包名：com.day01.helloworld

包名规范：全小写字母，多层之间点分隔

使用公司域名倒着写，后加应用名，达到：全球唯一

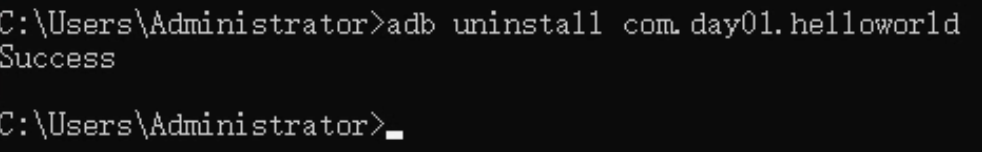
比如：腾讯的微信App的包名：com.tencent.mm

域名：[www.tencent.com](http://www.tencent.com) 应用名：mm

比如：达内版微信App的包名：

域名：[www.tedu.cn](http://www.tedu.cn) 应用名：mm

卸载HelloWorld应用



面试题：如何获取应用的包名？

方法1:进入到手机的目录中，查看子目录名获取包名

命令：adb shell 进入手机系统的shell环境 命令行环境

cd /data/data/

Ls

Ls -l 查看子目录名，就是包名

Exit

com.android.browser 浏览器

com.android.calculator2 计算器

com.android.calendar 日历

com.android.camera 相机

com.android.certinstaller

com.day01.helloworld

启发：如何判断卸载是否彻底？

查看系统的/data/data/目录下，应用的包（目录）是否删除；

方法2:拆包解决 前提需要有apk文件 安装包

命令：aapt dump badging apk文件路径名

定位：解析结果信息的首行 package:name=“包名”

aapt dump badging C:\Users\Administrator\Desktop\HelloWorld.apk

package: name='com.day01.helloworld' versionCode='1'

练习1:获取其他App的包名

D:\>aapt dump badging C:\Users\Administrator\Desktop\hehe\_v1.2.apk

package: name='com.zmhj.hehe' versionCode='120'

开发公司：[www.zmhj.hehe’](http://www.zmhj.hehe’) versionCode=’120’

产品名称：禾禾小镇

业务领域：有机农产品购票平台

方法3:通过包管理工具获取所有应用的包名

命令：adb shell pm list packages

Pm 就是包的管理 package manager

List 列表

技巧1:根据条件过滤筛选包名

命令1:adb shell pm list packages -s (system 系统）

过滤出系统自带的应用的包名

命令2: adb shell pm list package -3 (3 第三方产品)

过滤出第三方应用的包名

命令3: adb shell pm list package hello

过滤出含有hello的结果行

方法4:通过查看进程列表，找到对应的包名。 前提：应用已经启动

说明：正在执行的App为进程！

Adb shell ps

Adb shell ps |findstr hello

比如：

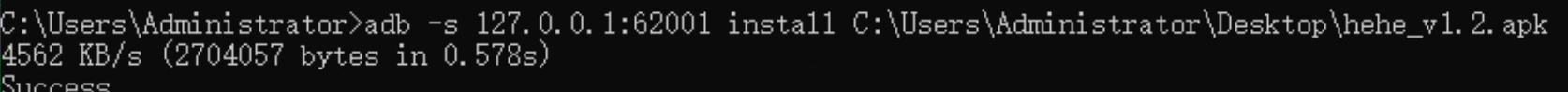
C:\Users\Administrator>adb shell ps |findstr hello

u0\_a46 1911 779 196492 25372 ffffffff b7e83997 S com.day01.helloworld

练习：针对禾禾小镇App进行安装、卸载测试

技巧：获取包名、系统目录查看、安装、卸载、日志抓取

1、安装



技巧：启动App之前，使用日志抓取，获取系统日志

Adb logcat -c 清空日志

Adb logcat > D:>hehe\_logcat1.txt 知道ctrl+c 为止

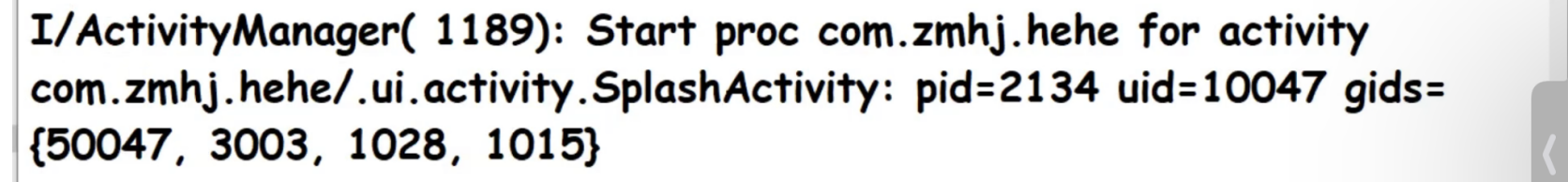
经过手工操作、记录系统日志，根据关键词提前信息：

Crash 崩溃、anr应用无响应、exception异常、error错误

1. 启动App，手工操作，关注系统日志的异常信息

原因：目前没有后台服务器的支持，导致前端App无法和后台联系，出现远程访问异常、连接超时异常。

活动管理器 启动 进程 hehe的包名 pid是2134



的进程 引起问题原因：网络通讯超时异常

1. 获取包名：

方法1：进入系统目录查看

Adb shell

Cd /data/data/

Pwd

Ls

Ls -l |grep hehe

Exit

方法2:通过apk文件提取

Aapt dump badging apk路径名

方法3:使用包管理命令提取，比如只提取第3方应用的包名

Adb shell pm list package -3

结果：

Package：com.day01.helloworld

Package:com.zmhj.hehe

1. 卸载
2. 检查/data/data/目录下是否移除了com.zmhj.hehe目录？

Adb shell

Cd /data/data

Ls -l |grep hehe

Exit

或：D:\>adb shell ls -ls | findstr

<7>adb shell

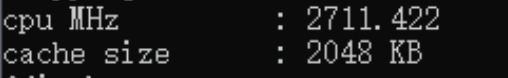
常用命令（Linux部分命令，比如ps，Android扩展的命令，比如pm、monkey）

1. id 查看当前用户详细身份 root超级用户
2. Pwd 查看当前工作目录
3. Ls -l 查看目录下的内容（文件、子目录）详细属性

文件类型 -文件 d目录、权限、属主、属组、修改日期

1. cd 改变工作目录 ：cd /data/data/
2. Cat 查看文件内容

Cat /proc/cpuinfo 查看CPU的状态信息



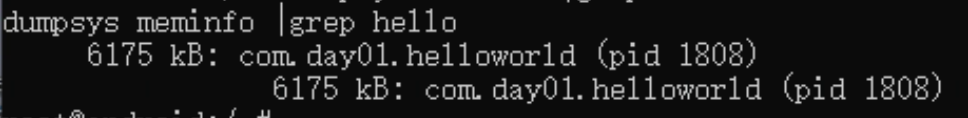
Cat /proc/meminfo 查看运行内存的信息



1. 获取系统信息，关注目前内存、CPU针对不同进程的占用情况：

Dumpsys meminfo ｜grep 指定App包名或pid？

比如：查看helloworld占用多少内存？



比如：查看helloworld的CPU使用率

1. ps 查看系统进程列表

尤其：pid和包名

1. pm 包管理工具

比如： pm list package 查看应用包名

1. date 查看系统日期、时间
2. Top 实时刷新查看Android系统的性能情况

CPU、Mem的使用率等

如何结束：ctrl+c

1. ｜管道符

将前一个命令的输出，作为后一个命令的输入（就是参数、条件）

比如：之前的命令｜grep 过滤条件

1. 文件管理

Touch 新建文件

Mv 重命名

Rm 删除

Rm -r rm -rf 删除目录

1. 模拟手机按键输入：

Adb shell input keyevent 事件编号

事件编号：

3 home键

4 back键

24 增加音量

25 减小音量

比如：

Adb shell input keyevent 3

总结：adb是Android自动化测试的基础，是较强大的手机测试工具（更高级的工具都是对adb进行封装、包装），可以进行各种常用的控制，包括：连接设备、应用管理、应用交互、网络查看和配置、文件管理、日志打印、查看属性、修改设置、发送广播等。

面试题：简述对adb的理解

1. Android调试桥，服务默认端口：5037

是Android自动化测试的基础，是SDK平台工具下的一个命令：adb.exe

选项：

-d 唯一真机

-e 唯一模拟器

-s 设备id 指定具体设备

1. adb常用子命令 分类管理
   1. 资源管理

Adb devices 设备名、设备状态 devices offline

Adb kill-server 关闭服务

Adb start-server 启动服务

Adb connect 设备名 主动连接设备

Adb version 查看adb版本

b）日志管理

Adb logcat -d \*:级别 对系统日志提取和过滤

I W E 不低于

\*:S 屏蔽其他

c） 文件收发

Adb push PC文件路径 手机端路径 推送

Adb pull 手机端文件路径 PC路径 拉回

D） shell 环境

Adb shell 大量的Linux命令、Android系统目录 比如pm、monkey

E） 安装与卸载

Adb install apk文件路径名

Adb uninstall app的包名

1. 获取包名的方法？
2. 进入系统目录查看：/data/data/ 前提：已经安装好App

Adb shell ls -l /data/data/ 查看目录下内容的详细属性

如需过滤：｜findstr xxx

B）通过aapt命令，结合apk文件提取 前提：具有apk安装包

Aapt dump badging apk文件路径名

C） 通过adb shell pm获取包信息，可以过滤细节 前提：安装好App

Adb shell pm list package -3

D）查看进程列表找到包名 前提：安装好App，并且启动该进程

Adb shell ps ｜findstr pid

补充说明：如何使用PC连接和管理真机（有线连接）

PC端：安装（SDK），含有adb工具

安装对应手机的驱动程序（不同手机不一样）

使用USB数据线和手机连接

手机端：开机，解锁屏幕

开启USB调试模式（文件传输、MIDI更全面）

开发者选项：USB Debugging

关闭锁屏设置、其他有关手机设置

注意：电脑端下载手机驱动程序，可以通过手机助手（豌豆荚、91、360、电脑管家）自动批匹配下载安装，之后关闭所有手机助手进程！

（手机助手底层也会使用adb，有线接管手机，占据5037端口，和我们自己的adb服务端口冲突！）

1. adb shell monkey测试
2. 冒烟测试之前和现在的异同点：
   * 1. 不同点：测试方式和测试目标 侧重点

之前：手工测试，主要关注业务

现在：自动化测试，更关心软件的不稳定因素

Crash（崩溃）、anr（应用无响应）、exception（异常）等

Ii：形同点：测试方向

都是确认软件的可测性，是否能够继续进入具体测试阶段；

B）monkey工具介绍 adb shell monkey ...

<1>本质：使用adb shell环境下的命令，进入Android系统中执行；

底层功能使用Java开发，结合Linux系统内核访问控制手机

<2>作用：monkey可以模拟用户操作（点击、触摸、滑动、拖拽、系统按键等）随机发送到手机设备（随机事件），对手机触发常见的事件，确认手机的稳定性；----Android系统的稳定性测试工具

<3>目的：确认应用的不稳定因素：crash、anr等

<4>阶段：确认应用的可测性、稳定性。

C）monkey配置选项类型：

<1>基本配置选项，比如：运行次数1000 必填参数

<2>运行约束选项，比如：指定包名 针对具体App进行测试

<3>运行的事件及频率，比如：触摸操作50 百分之50

<4>调试选项，比如：忽略崩溃

1. -p

含义：p就是package包

作用：指定App的包名，指定启动哪个App（被测应用）

语法：-p 参数1

案例：启动helloworld应用 ，随机触发10动作

Adb shell monkey -p com.day01.helloworld 10

如果App未安装，需要先安装

Adb shell apk 路径名

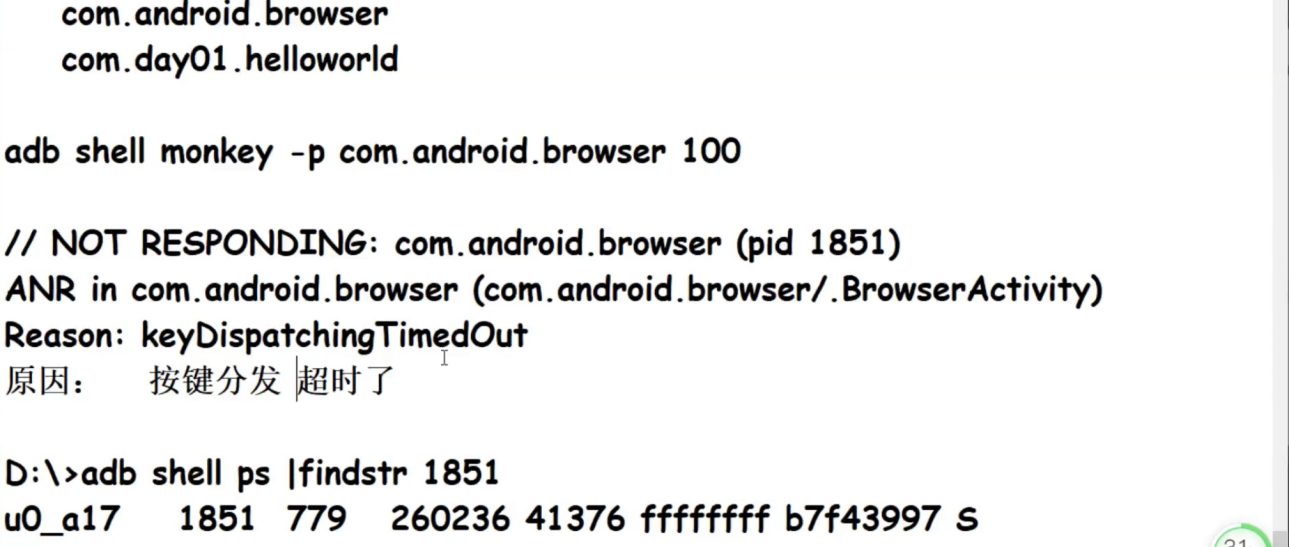
形式1:指定单个应用测试

命令：adb shell monkey -p 包名 10

形式1:指定多个应用测试

命令：adb shell monkey -p 包名1 -p 包名2 -p 包名3.....100

比如：adb shell pm list packages



形式3:对整个系统进行测试（不指定包名）

命令：adb shell monkey 10

练习：针对AVD自带的计算器执行monkey测试

1. 先找到包名：

Adb shell pm list package ｜findstr ca

1. 执行monkey：

adb shell monkey -p com.android.calculator2 100

<2>=

作用：控制monkey测试按照不同的等级进行日志输出，等级越高，日志越详细；

（日志的详略程度）

语法： -v 或多个-v

比如： -v -v -v

形式：

-v 第1等级 level1

-v -v 第2等级 level2

-v -v -v 第3等级 level3

特点：等级越高，日志内容越详细，推荐使用2、3

Level1:

Adb shell monkey -p com.android.calculator2 -v 10

跟踪日志内容：

日志内容较少，主要显示运行启动、结束、事件比例等；

Level2:

Adb shell monkey -p com.android.calculator2 -v -v 10

跟踪日志内容：

日志内容较为详细，跟踪到了被测应用的包名、活动窗口Activity，显示目前触发的事件的细节、步骤

Level3:

Adb shell monkey -p com.android.calculator2 -v -v -v 10

跟踪日志内容：

日志内容较为详细，跟踪到了选中和未选中的应用和包名、活动窗口Activity，显示目前触发的事件的细节、步骤；

技巧：可以将执行结果重定向到文件中保存

Adb shell monkey -p com.android.calculator2 -v -v 10 > D:\monkey1.txt

补充SDK工具：UI界面辅助测试工具 获取UI界面基本信息

SDK\tools\unautomatorviewer.bat 批处理工具

->双击启动->Devices Screenshot

设备 屏幕 快照 截屏、截图

鼠标经过之处，获取UI界面信息，包括空间的坐标点、类别属性

monkey运行结果小结：

1. 日志内容
   1. 通用的信息统计：包括随机种子数（seed）、运行动作次数（count）、运行应用（包名）、活动窗口、操作事件和运行比例（频率）；
   2. 运行信息统计，在第1个切换：Switch：向后查看

包括启动的应用、操作的事件和细节、运行异常信息、事件统计、运行结束提升；

1. 运行次数：不是指操作的事件测试（点击、触摸、滑动、跟踪球），而是指操作的动作次数：

比如：进行 触摸1次，包含2个动作（按下down和抬起up）

5 10

滑动1次，包含8个动作

按下1次，经过6个坐标点，抬起1次

跟踪球1次，包含15个动作

规律：事件与事件之间 一般有时间间隔Sleeping

总共的动作次数，包含了启动应用的1次；

1. 日志分析：根据关键词搜索

比如：ANR、crash、exception等，整理错误信息并反馈。

<3>--throttle 事件间隔时间

作用：针对操作的时间进行时间间隔，好比等待时间

既可以模拟时间间隔，让步骤有秩序进行

又可以调节压力大小，时间间隔越小，被测系统的压力越大

与语法：——throttle 时间毫秒数 单位：ms 1s=1000ms

举例：adb shell monkey -p 包名 --throttle 300 -v -v -v

练习：针对计算器进行monkey测试，要求事件间隔时间为0.5秒，执行30次动作，消息级别为3级，将结果写入D:\monkey3.txt

Adb shell monkey -p com.android.calculator2 --throttle 500 -v -v -v 30 >D:\monkey3.txt

\*\* Error: Unknow option: --throtle

错误 未知 选项

<4> -s

含义： -s 代表seed，指定随机种子数

作用： seed值如果🤷‍不指定，是由monkey随机指定；

由于每次产生的seed值不一样，导致产生的事件也不一样；

后续可以认为指定seed值，常用正整数，比如：10、150、16000、

1650887782785，就会产生统一的结果；

monkey会根据该值，生成一批随机动作；

后续再使用该值，产生的随机动作序列是相同的；（伪随机）

——适合对原有对原理进行回归测试！

重复原有的执行逻辑，便于返测，让问题重现。

语法： -s 随机整数 加在-p之后

举例：adb shell monkey -p 包名 -s 10 --throttle 500 -v -v -v 50 > D:\xxx.txt

练习：观察随机种子数对执行轨迹的影响

第一次：

Adb shell monkey -p com.android.calculator2 -s 10 --throttle 500 -v -v -v 50 >D:\monkey4.txt

执行逻辑：

打开计算器->点击导航 向下键->跟踪球事件 导航 确认->向右下方滑动->跟踪球事件->点击导航 向下键

->点击导航 向下键->点击菜单键 ->触摸某坐标点->点击菜单键->点击 菜单键->点击导航确认键->按下 导航向下键

第二次：

Adb shell monkey -p com.android.calculator2 -s 10 --throttle 500 -v -v -v 50 >D:\monkey5.txt

执行逻辑：

和第一次结果相同

第三次：

Adb shell monkey -p com.android.calculator2 -s 11 --throttle 500 -v -v -v 50 >D:\monkey6.txt

执行逻辑：

和前两次不同

打开计算器->点击 菜单键->向左下方滑动->向左下方滑动->触摸某坐标点->跟踪球（轨迹球）滑动->跟踪球

结论：随机种子数相同，则运行的逻辑、轨迹就相同；

如果种子数不同，则规则不同。

练习：使用之前monkey3的种子数，让测试轨迹重现；

之前执行结果的seed值：1650887782785

Adb shell monkey -p com.android.calculator2 -s 1650887782785 --throttle 500 -v -v -v 50 >D:\monkey1.txt

<5> --ignore-xxx 用于调试选项，表示忽略问题

作用：针对monkey测试遇到异常情况（anr、crash）进行规避、忽略，保证先执行完整个测试过程；

选项：

--ignore-crashes 忽略崩溃

--ignore-timeouts 忽略超时（由于未相应anr）

--ignore-security-exceptions 忽略安全异常

--monitor-native-crashes 监控本地崩溃

--ignore-native-crashes 忽略本地崩溃

举例：

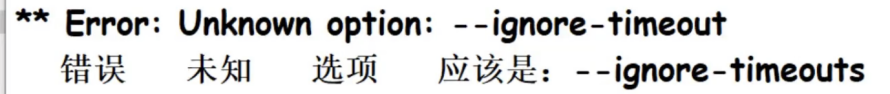
Adb shell monkey -p com.android.calculator2 -s 15 --throttle 600 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --monitor-native-crashes --ignore-native-crashes -v 50 >D:\monkey8.txt

常见问题：

\*\* No activities found to run, monkey aborted.

没有活动 找到 monkey 被放弃了

原因：App的包名写错了



<6> --pct-xxx 不同事件的百分比 percent

作用：指定操作事件以及百分比（运行比例）

语法：--pct-事件类型 百分比的整数值

举例：

--pct-touch 50 触摸事件占50%

--pct-motion 30 滑动事件占30%

剩余20 会随机分配

注意：如果事件百分比总和超过100%，会报错

技巧：查看命令帮助手册

Adb -help

Adb logcat -help

Adb shell monkey -help

面试题：monkey可选择的常用事件？ 只需记忆2～4个即可

--pct-touch 触摸、点击屏幕

--pct-motion 滑动（一般是直线滑动，途径若干个坐标点）

--pct-trackball 无规则的轨迹球滑动（曲线滑动） 跟踪球

--pct-syskeys 系统级事件，系统按键

比如：Home键、音量键、Back键、Menu键、下拉通知拦

--pct-nav 方向导航事件，比如光标上移

--pct-majornav 主要导航事件

比如：应用中的菜单选项，返回按钮（虚拟键）

--pct-appswitch App切换窗口

比如：应用内部窗口切换、应用之间的窗口切换

--pct-flip 键盘事件，比如键盘弹起、收回

--pct-anyevent 其他事件

--pct-pinchzoom 缩放事件，比如多指触控

穷举法：将以上比例都罗列一遍

Adb shell monkey -p com.android.calculator2 --pct-touch 10 --pct-motion 11 --pct-trackball 12 --pct-syskeys 13 --pct-nav 14 --pct-majornav 15 --pct-appswitch 16 --pct-flip 1 --pct-anyevent 3 --pct-pinchzoom 5 -s 15 --throttle 100 -v -v -v 300 >

// 0: 10.0% --pct-touch （重点）

// 1: 11.0% --pct-motion （重点）

// 2: 5.0% --pct-pinchzoom

// 3: 12.0% --pct-trackball （重点）

// 4: -0.0% 无效

// 5: 14.0% --pct-nav

// 6: 15.0% --pct-majornav

// 7: 13.0% --pct-syskeys （重点）

// 8: 16.0% --pct-appswitch

// 9: 1.0% --pct-flip

// 10: 3.0% --pct-anyevent

综合练习：针对家具酷App进行monkey冒烟测试

1. 测试机 连接 夜神模拟器

打开夜神模拟器

使用命令，主动连接设备；

Adb connect 127.0.0.1:62001

1. 针对被测App，获取包名（至少3种方法）

方法1:到系统安装目录下查看 前提：已经安装好 Com.ulang.jjk

Adb -s 127.0.0.1:62001 shell

Cd /data/data/

Ls-l

exit

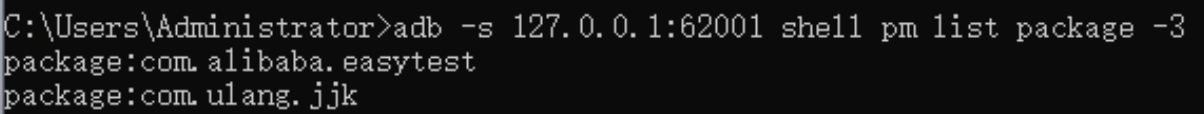
或：Adb -s 127.0.0.1:62001 shell ls -l /data/data/ ｜findstr jjk

方法2:使用apk文件提取 前提：具有安装包

Aapt dump badging apk安装包

方法3:使用pm命令列出 前提：先安装好App

针对夜神模拟器进入shell环境，执行pm命令列出第三方的应用的包名



1. 对家具酷进行安装和卸载测试

卸载：指定设备，根据应用的包名卸载

Adb -s 127.0.0.1:62001 uninstall com.ulang.jjk

检查：查看安装目录是否存在？

Adb -s 127.0.0.1:62001 shell

Cd /data/data/

Ls -l |grep jjk

Exit

安装：通过apk文件安装

Adb -s 127.0.0.1:62001 install apl路径名

先将apk发送到设备中

内部通过pm命令安装

成功 后续会自动删除临时安装文件

1. 设计monkey命令进行冒烟测试（用例的设计）
   1. 启动adb -s 127.0.0.1:62001 shell monkey

b）指定包名：-p com.ulang.jjk

C）指定事件比例：

触摸60% 滑动20% 主要导航10% 其他事件10%

--pct-touch --pct-motion --pct-majornav --pct-anyevent

D）随机种子数：-s 12

E） 事件间隔时间：500毫秒 --throttle 500

F）调试选项：忽略崩溃、忽略超时、忽略安全异常、监控本地崩溃、忽略本地崩溃

G）日志消息级别：3级 -v -v -v

H）动作次数：1000

I）：日志结果保存：D:\jjk\_monkey1.txt

DAY06

Adb -s 127.0.0.1:62001 shell monkey -p com.ulang.jjk --pct-touch 60 --pct-motion 20 --pct-majornav 10 --pct-anyevent 10 -s 12 --throttle 500 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --monitor-native-crashes --ignore-native-crashes -v -v -v 100 >

建议：将平时的命令编辑到一个Windows的批处理文件中：

Test\_jjk\_monkey1.bat

便于维护、重复使用（双击即可运行）

建议：提取开始抓取Android系统日志，为整个测试过程提供记录；

先清空系统日志

Adb -s 127.0.0.1:62001 locat -c

再开始监控系统日志：

Adb -s 127.0.0.1:62001 logcat > C:\Users\Administrator\Desktop\test\_jjk\_logcat1.txt

最后将App界面清理一下，比如关闭广告弹窗等

执行monkey用例，双击test\_jjk\_monkey1.bat 文件

执行完毕，结束监控：ctrl+c

1. 跟踪日志，分析结果，分析问题（crash、anr）

可以根据系统日志、monkey执行结果，结合关键词搜索：

Anr（应用无响应）、crash（崩溃）、exception（异常）、error（错误）、GC（内存异常：内存溢出GC Error 内存空间满了，空间回收不及时）、force\_closed（强制退出）

D:\>adb -s 127.0.0.1:62001 shell ps |findstr jjk

进程id





1. 用例的反馈结果格式：

用Excel表格反馈结果，包括 ：测试阶段、测试项目（中英文、包名）、测试工具、测试命令、测试事件和比例、测试次数、测试结果（Passed、Failed）、缺陷提交、 备注说明（告知错误信息和建议）。

1. 功能测试
2. 测试流程：

1）熟悉需求：结合需求文档、结合实际App操作体验

采用XMind梳理业务

2）测试分析：用例设计，比如整理checklist（检查点——测试点）

3）执行用例

4）提交和跟踪报告

5）编写测试用例

2、熟悉需求：结合对家具酷App的手工操作

1）首页模块

包括：购物车、商品搜索、消息通知、功能菜单、顶部轮播广告、酷头条、中间轮播广告、限时秒杀、商品列表

2）自营模块

3）酷友圈模块

4）我的模块

5）用户管理模块

1. 用例设计思路

1）通过业务梳理 和 手工操作，结合场景法分析；

2）将测试点转化为逐条用例，依次执行；

比如：搜索商品信息时

价格、销量、种类、关键词等作为条件

3）整理测试点：以checklist整理

1对1:一条测试点，转化为一条用例

1对多：一条测试点，转化为多条用例

1. monkey稳定性测试

1、冒烟测试和稳定性测试有何异同点？

1. 区别：

冒烟测试：前期阶段（功能测试之前）

确定软件的可能性（能不能测）

稳定性测试：后期阶段（功能测试之后）

确定软件的可用性（长期使用是否稳定？）

1. 相同

使用工具：monkey

发现问题点：crash崩溃，anr应用无响应等

1. 测试执行过程：
2. 环境部署：

手机：真机 或 夜神模拟器

测试项目：移动App 和 后台服务器 （Web服务器、DB服务器）

家具酷App 家具酷后台服务器

测试工具：adb shell monkey

基础工具：SDK的adb

1. 设计命令：根据项目业务、测试强度，调整monkey选项，制定策略；

包括：操作事件 --pct-xxx

等待时间 --throttle 时间越短，对被测应用的压力越大

运行测试等

比如：

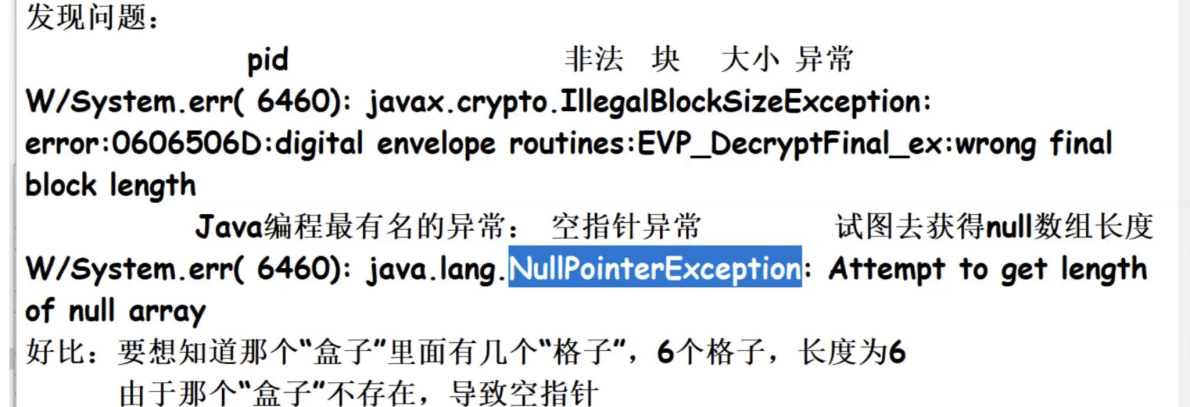
Adb -s 127.0.0.1:62001 shell monkey -p com.ulang.jjk --pct-touch 60 --pct-motion 20 --pct-majornav 10 --pct-anyevent 10 -s 12 --throttle 200 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --monitor-native-crashes --ignore-native-crashes -v -v -v 100 > C:\Users\Administrator\Desktop\test\_jjk\_monkey2.txt

运行期间获取系统日志：

Adb -s 127.0.0.1:62001 locat -c

Adb -s 127.0.0.1:62001 locat > C:\Users\Administrator\Desktop\test\_jjk\_logcat2.txt

NullPointerException 空指针异常



由于那个“盒子”不存在，导致空指针（数组长度 百事数组能够容纳元素的个数）

1. 面试题：Android自动化测试有哪些常用工具？举例说明

基础工具：adb SDK\platform-tools\adb.exe

1)monkey：是基于随机事件，适合Android冒烟测试、稳定性测试

是adb shell下的命令

2)monkeyrunner：是基于精确坐标点，适合Android功能自动化测试

是SDK的tools目录下的批处理命令：monkeyrunner.bat

使用Python语法 （jython Java实现的Python）写脚本

模拟界面的各种操作：比如：触摸、滑动、拖拽、长按、实体键、输入文本、

3)Appium：是基于控件，适合Android功能自动化测试，

由第三方提供的一套测试框架，通过编写Java、Python等脚本

对界面的元素、控件，比如文本框、按钮、图片、单选钮进行自动控制，完成功能操作。

（工作原理和Seleniun相同 Web界面功能自动化）

练习：针对AVD自带的计算器进行功能自动化测试（了解）

使用工具：monkeyrunner

先启动AVD模拟器，关闭夜神；

开始执行：有线进入到monkeyrunner环境 Jython环境 提示符:>>>

# 导入有关的工具类

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MonkeyImage

#等待连接设备，获得设备对象

device = MonkeyRunner.waitForConnection()

#针对该设备，进行一次触摸操作，坐标点（200，300）按下并抬起

device.touch(200,300,’DOWN\_AND\_UP’)

#针对该设备，当前文本框中输入文本：123456

device.type(‘123456’)

#针对该设备，点击Menu键 按下并抬起

device.press(‘KEYCODE\_MENU’,’DOWN\_AND\_UP’)

#针对该设备，进行截图，获得截图对象result

result = device.takeSnapshot()

#将截图导出为图片，png格式

result.writeToFile(‘C:\Users\Administrator\Desktop\1.png’,’png’)

#针对该设备，点击Home键 按下并抬起

device.press(‘KEYCODE\_HOME’,’DOWN\_AND\_UP’)

通过Ctrl+d结束输入，退出monkeyrunner命令行环境（为了单步调试脚本）

建议：将以上步骤写成Pyehon脚本，使用monkeyrunner命令自动批量执行！

比如：D:\ 新建文本 test1.py (扩展名必须为.py)

运行脚本：Windows cmd下

Monkeyrunner D:\test1.py

技巧：输入提示文本

print(‘start...’)

多个步骤之前要求添加等待时间：

MonkeyRunner.sleep(5) 等待5秒

print(‘END’)

归纳：

1. 手工测试

安装好App，结合需求设计用例，执行

1. 自动化工具基础

adb、adb shell monkey、monkeyrunner

基础 冒烟、稳定性测试 功能自动化

命令行 python脚本

3、