Android tools简介

1. Android DeBug Bridge工具常用命令
2. adb devices

可以通过使用devices 命令来得到一系列相关联的模拟器/设备实例

1. adb install +包路径

安装apk程序

1. adb uninstall + 包明

卸载apk

1. adb push

上传文件

adb push <本地文件> <远程路径> adb push helloworld.apk system/app

1. adb pull

下载文件

adb pull <远程路径> <本地路径> adb pull system/app/helloworld.apk F:\

1. adb shell

getprop 显示系统属性信息

setprop <key> <value> 设置系统属性信息

service list 显示系统服务列表

ps 列出系统进程

ls

df 检查文件系统的磁盘空间占用情况

cat

kill

dumpsys

显示系统数据信息 如：

dumpsys wifi 显示wifi信息

dumpsys cpuinfo 显示cpu信息

dumpsys meminfo com.android.launcher2 显示launcher2的内存使用情况

adb shell dumpsys activity 显示系统Activity信息

1. adb shell screencap

截屏adb shell screencap -p /sdcard/screen.png

截图直接保存到电脑

adb shell screencap -p | sed 's/\r$//' > screen.png

1. adb logcat（如果需要打印时间，加参数-v time）
2. adb bugreport > xxx.log

会有从开机之后详细的dumpsys,dumpstate和logcat信息，是一份完整的日志记录。对分析用户行为，异常信息，系统状态有很大的参考作用。一般我们会把bugreport导出到电脑上分析。

1. Android Log分析

通常情况下，通过查找：null，error"，E/，Fatal， crash，IndexOutOfBoundsException

，NullPointerException，Build fingerprint:，exception，ANR，DexOpt等的错误信息，来定位问题。

1. NullPointerException：空指针问题

例子：

java.lang.NullPointerException

at cn.com.jbbis.loan.night.desk.NightAppHost.getTranCode(NightAppHost.java:253)

at cn.com.jbbis.loan.night.desk.NightAppHost.invokeApplication(NightAppHost.java:166)

at cn.com.jbbis.loan.night.desk.NightAppHost.start(NightAppHost.java:56)

at cn.com.jbbis.loan.night.desk.NightAppHost.main(NightAppHost.java:315)

java.lang.NullPointerException

1. IndexOutOfBoundsException: 数组越界问题

例子：

ava.lang.IndexOutOfBoundsException: Index: 0, Size: 0  
at java.util.ArrayList.RangeCheck(ArrayList.java:507)  
at java.util.ArrayList.get(ArrayList.java:324)  
at com.crystaldecisions.sdk.occa.infostore.internal.al.get(Unknown Source)  
at com.businessobjects.BO.getDocumentInstance(BO.java:207)  
at com.neusoft.tax.hxzg.hblt.bo.bl.applogic.BOUtil.getWebiPrompts(BOUtil.java:740)  
at com.neusoft.tax.hxzg.hblt.bo.web.action.BOQueryAction.getWebiViewParameters(BOQueryAction.java:658)  
at com.neusoft.tax.hxzg.hblt.bo.web.action.BOQueryAction.WebiView(BOQueryAction.java:785)  
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)  
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)  
at java.lang.reflect.Method.invoke(Method.java:324)  
at org.apache.struts.actions.DispatchAction.dispatchMethod(DispatchAction.java:276)  
at org.apache.struts.actions.DispatchAction.execute(DispatchAction.java:196)  
at org.apache.struts.action.RequestProcessor.processActionPerform(RequestProcessor.java:421)  
at org.apache.struts.action.RequestProcessor.process(RequestProcessor.java:226)  
at com.neusoft.unieap.action.EAPActionServlet.process(EAPActionServlet.java:483)  
at org.apache.struts.action.ActionServlet.doPost(ActionServlet.java:415)  
at javax.servlet.http.HttpServlet.service(HttpServlet.java:760)  
at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)  
at weblogic.servlet.internal.ServletStubImpl$ServletInvocationAction.run(ServletStubImpl.java:1077)  
at weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletStubImpl.java:465)

1. Exception：异常捕获
2. ANR：好事问题

例子：

I/dalvikvm( 1014): Wrote stack traces to '/data/anr/traces.txt'

　I/Process ( 957): Sending signal. PID: 1124 SIG: 9

　E/ActivityManager( 957): ANR in com.ipanel.join.appstore

　E/ActivityManager( 957): Reason: Broadcast of Intent { act=android.appwidget.action.APPWIDGET\_UPDATE cmp=com.ipanel.join.appstore/.widget.SmallWidget1 (has extras) }

　E/ActivityManager( 957): Load: 1.57 / 0.38 / 0.13

　E/ActivityManager( 957): CPU usage from 6290ms to 0ms ago:

　E/ActivityManager( 957): 70% 957/system\_server: 1.1% user + 69% kernel / faults: 204 minor

　E/ActivityManager( 957): 8.7% 1014/com.ipanel.join.home.jelly: 7.7% user + 0.9% kernel / faults: 899 minor

　E/ActivityManager( 957): 0.3% 687/yaffs-bg-1: 0% user + 0.3% kernel

　E/ActivityManager( 957): 0.3% 688/yaffs-bg-1: 0% user + 0.3% kernel

　E/ActivityManager( 957): 0.3% 935/irkeyservice: 0% user + 0.3% kernel

　E/ActivityManager( 957): 0% 890/galcore daemon : 0% user + 0% kernel

　E/ActivityManager( 957): 0.1% 1076/com.ipanel.join.browser: 0.1% user + 0% kernel

　E/ActivityManager( 957): 0.1% 1124/com.ipanel.join.appstore: 0.1% user + 0% kernel

　E/ActivityManager( 957): 75% TOTAL: 4.7% user + 70% kernel

　E/ActivityManager( 957): CPU usage from 622ms to 1169ms later:

　E/ActivityManager( 957): 71% 957/system\_server: 0% user + 71% kernel / faults: 2 minor

　E/ActivityManager( 957): 67% 965/SurfaceFlinger: 0% user + 67% kernel

　E/ActivityManager( 957): 1.9% 973/ActivityManager: 1.9% user + 0% kernel

　E/ActivityManager( 957): 1.9% 991/CursorMove: 0% user + 1.9% kernel

　E/ActivityManager( 957): 7.6% 1014/com.ipanel.join.home.jelly: 7.6% user + 0% kernel / faults: 31 minor

　E/ActivityManager( 957): 5.7% 1119/Thread-13: 5.7% user + 0% kernel

　E/ActivityManager( 957): 1.9% 1123/Thread-13: 1.9% user + 0% kernel

　 E/ActivityManager( 957): 76% TOTAL: 1.8% user + 72% kernel + 1.8% softirq

1. Fatal问题：

很多都很动态库和空指针有关，一般会接下来打印Build fingerprint:或着NullPointerException

例子：

15:07:40.609: I/DEBUG(85): Build fingerprint: 'unknown'

15:07:40.609: I/DEBUG(85): pid: 623, tid: 1000, name: Binder\_5 >>> system\_server <<<

15:07:40.609: I/DEBUG(85): signal 11 (SIGSEGV), code 1 (SEGV\_MAPERR), fault addr 00000014

15:07:40.679: I/DEBUG(85): r0 00000000 r1 649661d6 r2 6d12fc30 r3 00000001

15:07:40.679: I/DEBUG(85): r4 6740e3d0 r5 00000000 r6 6d12fc30 r7 6740e3d0

15:07:40.679: I/DEBUG(85): r8 00000001 r9 400aff10 sl 6740e3c8 fp 00000001

15:07:40.679: I/DEBUG(85): ip 00000000 sp 6d12fbe8 lr 40054dff pc 40054e0a cpsr 60000030

15:07:40.679: I/DEBUG(85): d0 3f8000003f800000 d1 11111111c1c00000

15:07:40.679: I/DEBUG(85): d2 1a1a1a1a1a1a1a1a d3 ffffffffffffffff

15:07:40.679: I/DEBUG(85): d4 0808080808080808 d5 1111111111111111

15:07:40.679: I/DEBUG(85): d6 000000001a1a1a1a d7 0000000000000000

15:07:40.679: I/DEBUG(85): d8 0000000000000000 d9 0000000000000000

15:07:40.679: I/DEBUG(85): d10 0000000000000000 d11 0000000000000000

15:07:40.679: I/DEBUG(85): d12 0000000000000000 d13 0000000000000000

15:07:40.679: I/DEBUG(85): d14 0000000000000000 d15 0000000000000000

15:07:40.679: I/DEBUG(85): d16 000000000bebc200 d17 0000000000000000

15:07:40.679: I/DEBUG(85): d18 0000000000000000 d19 0000000000000000

15:07:40.679: I/DEBUG(85): d20 0808080808080808 d21 1111111111111111

15:07:40.679: I/DEBUG(85): d22 1a1a1a1a1a1a1a1a d23 ffffffffffffffff

15:07:40.679: I/DEBUG(85): d24 0808080808080808 d25 1111111111111111

15:07:40.679: I/DEBUG(85): d26 1a1a1a1a1a1a1a1a d27 ffffffffffffffff

15:07:40.679: I/DEBUG(85): d28 0100010001000100 d29 0100010001000100

15:07:40.679: I/DEBUG(85): d30 0000000000000000 d31 0000000000000000

15:07:40.679: I/DEBUG(85): scr 60000010

例子2：

单例模式，初始化顺序不对造的log。

D/AndroidRuntime( 1257): Shutting down VM

W/dalvikvm( 1257): threadid=1: thread exiting with uncaught exception (group=0x41988ac8)

E/AndroidRuntime( 1257): FATAL EXCEPTION: main

E/AndroidRuntime( 1257): java.lang.ExceptionInInitializerError

E/AndroidRuntime( 1257): at com.android.systemui.statusbar.phone.PhoneStatusBar.start(PhoneStatusBar.java:499)

E/AndroidRuntime( 1257): at com.android.systemui.SystemUIService.onCreate(SystemUIService.java:94)

E/AndroidRuntime( 1257): at android.app.ActivityThread.handleCreateService(ActivityThread.java:2554)

E/AndroidRuntime( 1257): at android.app.ActivityThread.access$1600(ActivityThread.java:148)

E/AndroidRuntime( 1257): at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1323)

E/AndroidRuntime( 1257): at android.os.Handler.dispatchMessage(Handler.java:99)

E/AndroidRuntime( 1257): at android.os.Looper.loop(Looper.java:137)

E/AndroidRuntime( 1257): at android.app.ActivityThread.main(ActivityThread.java:5102)

E/AndroidRuntime( 1257): at java.lang.reflect.Method.invokeNative(Native Method)

E/AndroidRuntime( 1257): at java.lang.reflect.Method.invoke(Method.java:511)

E/AndroidRuntime(1257): at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:793)

E/AndroidRuntime( 1257): at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:560)

E/AndroidRuntime( 1257): at dalvik.system.NativeStart.main(Native Method)

E/AndroidRuntime( 1257): Caused by: android.content.res.Resources$NotFoundException: Resource ID #0x6080006

E/AndroidRuntime( 1257): at android.content.res.Resources.getValue(Resources.java:1027)

E/AndroidRuntime( 1257): at android.content.res.NubiaResources.getValue(NubiaResources.java:175)

E/AndroidRuntime( 1257): at android.content.res.Resources.getInteger(Resources.java:834)

E/AndroidRuntime( 1257): at miui.app.ToggleManager.<clinit>(ToggleManager.java:845)

1. DexOpt错误：

例子：

W/dalvikvm( 1803): DexOpt: --- END 'SettingsProvider.apk' --- status=0x000a, process failed

E/dalvikvm( 1803): Unable to extract+optimize DEX from '/system/app/SettingsProvider.apk'

E/SystemServer( 1803): Failure starting Input Manager Service

E/SystemServer( 1803): java.lang.RuntimeException: Unable to get provider com.android.providers.settings.SettingsProvider: java.lang.ClassNotFoundException: com.android.providers.settings.SettingsProvider in loader dalvik.system.PathClassLoader[/system/app/SettingsProvider.apk]

E/SystemServer( 1803): at android.app.ActivityThread.installProvider(ActivityThread.java:3557)

E/SystemServer( 1803): at android.app.ActivityThread.getProvider(ActivityThread.java:3356)

1. Android性能分析工具——DDMS