大象杂货

adb shell bugreport分析

2016-11-03 18:40 838人阅读 评论(0) 收

分类：

Android（11）

目录(?) [+]

BugReport格式分析

BugReport设置

bugreport本身并没有什么选项，主要是通过dumpsys等命令配合完成：

1. 重置电池统计信息

```
adb shell dumpsys batterystats --reset
```

2. Wakelock analysis全部wakelock信息

```
adb shell dumpsys batterystats --enable full-wake-history
```

3. Kernel trace analysis分析内核，主要分析wakeup source和wakeup activities，首先使能k

```
$ adb shell

# Set the events to trace.
$ echo "power:wakeup_source_activate" >> /d/tracing/set_event
$ echo "power:wakeup_source_deactivate" >> /d/tracing/set_event

# The default trace size for most devices is 1MB, which is relatively low and might cause  the logs to overflow
# 8MB to 10MB should be a decent size for 5-6 hours of logging.

$ echo 8192 > /d/tracing/buffer_size_kb

$ echo 1 > /d/tracing/tracing_on
```

然后获得log

```
$ echo 0 > /d/tracing/tracing_on
$ adb pull /d/tracing/trace <some path="">
```

评论排行


```
# Take a bug report at this time.
$ adb bugreport > bugreport.txt
```

BugReport内容

Battery History

1. bugreport记录以下信息

Brightness
CPU running
Charging on
Charging status
Health
JobScheduler
Kernel only uptime
Level
Package active
Partial wakelock
Phone scanning
Phone state
Plug
Plugged
Screen
Temperature
Top app
Voltage
Wifi on
Wifi running
Wifi supplicant

2. 总括信息

```
Battery History (15% used, 40KB used of 256KB, 240 strings using 15KB):
        0 (9) RESET:TIME: 2016-08-23-10-41-35
        0 (3) 031 status=discharging health=good plug=none temp=310 volt=7392   +running
```

40KB used of 256KB 256KB缓存用了40KB

3. 时间轴

```
    +14m02s086ms (2) 030 +wake_lock=1000:"WifiSuspend"
    +14m02s095ms (1) 030 -wake_lock
```

+14m02s086ms 自RESET TIME经历时间

030 当前电量百分比

+wake_lock wake_lock开始

-wake_lock wake_lock结束

```
temp=330 volt=7414
```

temp 温度

volt 电压

```
current=-861
```

current 为当前电流，+为充电，-为放电

Per-PID Stats

PID 3270 wake time: +1ms

PID 0 wake time: +847ms

某个PID获取电源锁的时间

Discharge step durat

#0: +41m55s146ms to 31 (power-save-off)

#1: +2m40s252ms to 32 (screen-on, power-save-off, device-idle-off)

#2: +3m20s224ms to 34 (power-save-off, device-idle-off)

#3: +2m40s495ms to 35 (screen-on, power-save-off, device-idle-of

#1: +2m40s252ms to 32 从33放电到32经历时间为2m40s252ms，

Daily Stats

Daily stats:

Current start time: 2016-08-23-11-08-25

Next min deadline: 2016-08-24-01-00-00

Next max deadline: 2016-08-24-03-00-00

Current daily discharge step durations:

#0: +41m55s146ms to 31 (power-save-off)

#1: +2m40s252ms to 32 (screen-on, power-save-off, device-idle-off)

#2: +3m20s224ms to 34 (power-save-off, device-idle-off)

按天统计放电过程

Package changes:

Update com.tencent.mobileqq vers=398

Update com.jingdong.app.mall vers=34669

Update com.tencent.mm vers=840

Update com.sdu.didi.psnger vers=150

Update com.qzone vers=94

发生改变的package

Statistics since last charge

System starts: 5, currently on battery: false

Time on battery: 4h 50m 51s 103ms (62.2%) realtime, 2h 10m 39s

Time on battery screen off: 3h 9m 1s 524ms (40.4%) realtime, 28m 50s 181ms (6.2%) uptime

Total run time: 7h 47m 49s 583ms realtime, 5h 7m 38s 242ms uptime

Start clock time: 2016-08-23-10-41-27

Screen on: 1h 41m 49s 579ms (35.0%) 38x, Interactive: 1h 41m 41s 490ms (35.0%)

Screen brightnesses:

dark 42m 58s 386ms (42.2%)

dim 58m 43s 967ms (57.7%)

light 6s 398ms (0.1%)

bright 819ms (0.0%)

Device idling: 1h 50m 43s 273ms (38.1%) 7x

自从上次充电到现在为止的各类统计，看名字可知意义。

Estimated power use (mAh)

Capacity: 3900, Computed drain: 1565, actual drain: 819-1053

Uid 1000: 338 (cpu=334 wake=2.32 wifi=0.764 sensor=0.522)

Screen: 304

Uid 0: 212 (cpu=212 wifi=0.00877)

Uid u0a112: 35.5 (cpu=35.4 wifi=0.00295 sensor=0.00000125)

Uid u0a101: 32.7 (cpu=26.8 wifi=4.60 sensor=1.28)

Uid u0a100: 28.1 (cpu=27.1 wake=0.913 wifi=0.00639)

Capacity: 3900, Computed drain: 1565... 头部为电池的整体信息

Uid u0a112: 根据uid统计电量消耗

All kernel wake locks

Kernel Wake lock wlan_rx_wake: 11m 4s 92ms (319 times) realtime

Kernel Wake lock wlan_wd_wake: 8m 51s 176ms (1432 times) realtime

Kernel Wake lock mmc1_detect : 2m 34s 710ms (482 times) realtime

kernel wake lock获取者信息、时长及次数

All partial wake locks

Wake lock 1000 eventloop: 1m 53s 858ms (92 times) realtime

Wake lock 1000 dispatcher: 29s 367ms (39 times) realtime

Wake lock u0a100 WakerLock:25822454: 21s 879ms (29 times) re

partial wake lock信息

All wakeup reasons

Wakeup reason unknown: 4m 36s 753ms (150 times) realtime

Wakeup reason Abort:Last active Wakeup Source: wlan_wd_wake: 1m 6s 628ms (32 times) realtime

Wakeup reason Abort:Last active Wakeup Source: ARD: 53s 17ms (25 times) realtime

系统唤醒原因

更加UID统计相关信息

1001:

Wake lock ServiceStateTracker realtime

Wake lock CallerInfoCache realtime

Wake lock RADIOPOWERDOWN_IFNOCARD_WAKELOCK: 14s 321ms partial (3 times) realtim

Wake lock SERVICESTATE_WAIT_DISCONNECT_WAKELOCK realtime

...
TOTAL wake: 17s 288ms partial realtime

Foreground activities: 2s 36ms realtime (2 times)

Foreground for: 4h 50m 34s 89ms

Active for: 4h 50m 36s 257ms

Running for: 4h 50m 36s 264ms
...
Total cpu time: u=1m 12s 192ms s=48s 219ms p=0mAh

Proc com.huawei.skytone:

CPU: 80ms usr + 30ms krn ; 0ms fg

28 starts

Proc com.android.phone:

CPU: 39s 770ms usr + 21s 950ms krn ; 0ms fg

8 starts

...

Apk com.huawei.ims:

Service com.huawei.ims.ImsService:

Created for: 2h 10m 4s 956ms uptime

Starts: 5, launches: 5

Apk com.android.phone:

Service com.android.phone.TelephonyDebugService:

Created for: 2h 10m 4s 945ms uptime

Starts: 5, launches: 5

TOTAL wake wake lock信息

Total cpu cpu信息

Proc ... 进入Proc信息

Apk ... ***.....待分析.....***

顶 0 踩 0

- 上一篇 Android 自定义控件方法
- 下一篇 Rosalloc简要分析

相关文章推荐

- android bugreport. .
- Python即将成为第一语言
- Android adb bugreport工具分析和使用
- 构建企业级高性能OLAP引擎--董西成
- Android bugreport工具分析和使用
- JDK9新特性解读
- android 使用开源工具ChkBugReport分析Bugreport
- 华为工程师，带你实战C++
- bugreport 源码篇
- Android自定义控件全知道
- [整理]Android测试日志文件抓取与分析
- TensorFlow入门基础知识详解
- Android adb bugreport工具分析和使用
- android bu
- chkbugrepo
- android bu

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场