1    log文件分类简介

            实时打印 的主要有： logcat main ， logcat radio ， logcat events ， tcpdump，还有高通平台的还会有 QXDM 日志

          状态信息 的有： adb shell cat /proc/kmsg ， adb shell dmesg ， adb shell dumpstate ， adb shell dumpsys ， adb bugreport ，工程模式等

2    LOG抓取详解

l  实时打印

      adb logcat -b main -v time>app.log  打印应用程序的 log

    adb logcat -b radio -v time> radio.log 打印射频相关的 log ， SIM STK 也会在里面，modem 相关的 ATcommand 等，当然跟 QXDM 差的很远了。

    adb logcat -b events -v time  打印系统事件的日志，比如触屏事件。。。

    tcpdump 是很有用的，对于 TCP/IP 协议相关的都可以使用这个来抓， adb shell tcpdump -s 10000 -w /sdcard/capture.pcap ，比如抓 mms 下载的时候的 UA profile ，browser 上网的时候，使用 proxy 的 APN 下载， streaming 的相关内容包括 UA profile等。

    最后是高通平台的 QXDM ，不管是不是 Android ，只要使用高通芯片，都会对它很熟悉，当然了，不是高通的芯片就不用提它了。这个不多讲，内容丰富，射频，电话，上网， ... 凡是高通提供的解决方案，这个都可以抓。(QXDM 的 LOG抓取方法请参考 QPST、 QXDM的基本使用说明及作用 )

l  状态信息

o   bugreport （命令 adb bugreport>bugreport.log) 。里面包含有 dmesg ，dumpstate 和 dumpsys 。

o   dumpstate 是系统状态信息，里面比较全，包括手机当前的内存信息、 cpu 信息、 logcat 缓存， kernel 缓存等等。

o     adb shell dumpsys 这个是关于系统 service 的内容都在这个里面，这个命令还有更详尽的用法，比如 adb shell dumpsys meminfo system 是查看 system 这个process 的内存信息。

o   kmsg 抓取

adb shell cat /proc/kmsg > kmsg.txt ，打开后查 msm\_kgsl 字段

说明：用于检索用 printk 生成的内核消息。任何时刻只能有一个具有超级用户权限的进程可以读取这个文件。也可以用系统调用 syslog 检索这些消息。通常使用工具dmesg 或守护进程 klogd 检索这些消息。 proc 是一个内存文件系统 , 每次读文件kmsg 实际是内核内部的循环缓冲区 , 每读过后 , 循环缓冲区的东西就被认为已经处理过了 ( 也就是变成无效内容 ), 所以你再次读为空是很正常的 为什么会这样处理呢 , 循环缓冲区大小有限 , 内核又随时可能往里面写东西 , 所以这样处理很正常 . 你去查一下/proc/kmsg 的信息有没有跟系统日志关联 , 如果有的话 , 你就可以读日志文件

o   dmsg 抓取

adb shell dmesg > dmesg.txt

说明： dmesg 用来显示开机信息， kernel 会将开机信息存储在 ring buffer 中。您若是开机时来不及查看信息，可利用 dmesg 来查看。 dmesg 是 kernel 的 log ，凡是跟 kernel 相关的，比如 driver 出了问题（相机，蓝牙， usb ，启动，等等）开机信息亦保存在 /var/log 目录中，名称为 dmesg 的文件里。 more /var/log/dmesg

o   工程模式下 log 的抓取

            对于 Apollo 手机请拨打 \*#\*#8888#\*#\* , 然后勾选相应的 LOG 。待测试结束后，通过 SD 卡导出 LOG 到 PC.

3.Log分析：

Get Log from Android System

*adb bugreport > bugreport.txt*

*copy bugreport to the current directory.*

*bugreport 里面包含了各种 log 信息 , 大部分 log 也可以通过直接运行相关的程序来直接获得 .*

*步骤如下 :*

*1.adb shell 2. 进入相关工具程式的目录 3. 执行相关程式 4. 得到相关信息*

*下面以输出进程信息为例 1.adb shell 2. 输入 ps -P 3. 可以看到相关进程信息*

Log Archive Analysis

1.bugreport

*bugreport 记录 android 启动过程的 log, 以及启动后的系统状态 , 包括进程列表，内存信息， VM 信息等等到 .*

2.bugreport 结构分析

(1)dumpstate

*MEMORY INFO*

*获取该 log: 读取文件 /proc/meminfo*

*系统内存使用状态*

*CPU INFO*

*获取该 log: 执行 /system/bin/top -n 1 -d 1 -m 30 -t*

*系统 CPU 使用状态*

*PROCRANK*

*获取该 log: 执行 /system/bin/procrank*

*执行 /system/xbin/procrank 后输出的结果 , 查看一些内存使用状态*

*VIRTUAL MEMORY STATS*

*获取该 log: 读取文件 /proc/vmstat*

*虚拟内存分配情况*

*vmalloc 申请的内存则位于 vmalloc\_start ～ vmalloc\_end 之间，与物理地址没有简单的转换关系，虽然在逻辑上它们也是连续的，但是在物理上它们不要求连续。*

*VMALLOC INFO*

*获取该 log: 读取文件 /proc/vmallocinfo*

*虚拟内存分配情况*

*SLAB INFO*

*获取该 log: 读取文件 /proc/slabinfo*

*SLAB 是一种内存分配器 . 这里输出该分配器的一些信息*

*ZONEINFO*

*获取该 log: 读取文件 /proc/zoneinfo*

*zone info*

*SYSTEM LOG( 需要着重分析 )*

*获取该 log: 执行 /system/bin/logcat -v time -d \*:v*

*会输出在程序中输出的 Log, 用于分析系统的当前状态*

*VM TRACES*

*获取该 log: 读取文件 /data/anr/traces.txt*

*因为每个程序都是在各自的 VM 中运行的 , 这个 Log 是现实各自 VM 的一些 traces*

*EVENT LOG TAGS*

*获取该 log: 读取文件 /etc/event-log-tags*

*EVENT LOG*

*获取该 log: 执行 /system/bin/logcat -b events -v time -d \*:v*

*输出一些 Event 的 log*

*RADIO LOG*

*获取该 log: 执行 /system/bin/logcat -b radio -v time -d \*:v*

*显示一些无线设备的链接状态 , 如 GSM ， PHONE,STK(Satellite Tool Kit)…*

*NETWORK STATE*

*获取该 log: 执行 /system/bin/netcfg ( 得到网络链接状态 )*

*获取该 log: 读取文件 /proc/net/route ( 得到路由状态 )*

*显示网络链接和路由*

*SYSTEM PROPERTIES*

*获取该 log: 参考代码实现*

*显示一些系统属性 , 如 Version,Services,network…*

*KERNEL LOG*

*获取该 log: 执行 /system/bin/dmesg*

*显示 Android 内核输出的 Log*

*KERNEL WAKELOCKS*

*获取该 log: 读取文件 /proc/wakelocks*

*内核对一些程式和服务唤醒和休眠的一些记录*

*KERNEL CPUFREQ*

*(Linux kernel CPUfreq subsystem) Clock scaling allows you to change the clock speed of the CPUs on the fly.*

*This is a nice method to save battery power, because the lower the clock speed is, the less power the CPU consumes.*

*PROCESSES*

*获取该 log: 执行 ps -P*

*显示当前进程*

*PROCESSES AND THREADS*

*获取该 log: 执行 ps -t -p -P*

*显示当前进程和线程*

*LIBRANK*

*获取该 log: 执行 /system/xbin/librank*

*剔除不必要的 library*

*BINDER FAILED TRANSACTION LOG*

*获取该 log: 读取文件 /proc/binder/failed\_transaction\_log*

*BINDER TRANSACTION LOG*

*获取该 log: 读取文件 /proc/binder/transaction\_log*

*BINDER TRANSACTIONS*

*获取该 log: 读取文件 /proc/binder/transactions*

*BINDER STATS*

*获取该 log: 读取文件 /proc/binder/stats*

*BINDER PROCESS STATE*

*获取该 log: 读取文件 /proc/binder/proc/\**

*bind 相关的一些状态*

*FILESYSTEMS*

*获取该 log: 执行 /system/bin/df*

*主要文件的一些容量使用状态 (cache,sqlite,dev…)*

*PACKAGE SETTINGS*

*获取该 log: 读取文件 /data/system/packages.xml*

*系统中 package 的一些状态 ( 访问权限 , 路径 …) ，类似 Windows 里面的一些 lnk 文件吧 .*

*PACKAGE UID ERRORS*

*获取该 log: 读取文件 /data/system/uiderrors.txt*

*错误信息*

*KERNEL LAST KMSG LOG*

*最新 kernel message log*

*LAST RADIO LOG*

*最新 radio log*

*KERNEL PANIC CONSOLE LOG*

*KERNEL PANIC THREADS LOG*

*控制台 / 线程的一些错误信息 log*

*BACKLIGHTS*

*获取该 log: 获取 LCD brightness 读 /sys/class/leds/lcd-backlight/brightness*

*获取该 log: 获取 Button brightness 读 /sys/class/leds/button-backlight/brightness*

*获取该 log: 获取 Keyboard brightness 读 /sys/class/leds/keyboard-backlight/brightness*

*获取该 log: 获取 ALS mode 读 /sys/class/leds/lcd-backlight/als*

*获取该 log: 获取 LCD driver registers 读 /sys/class/leds/lcd-backlight/registers*

*获取相关亮度的一些信息*

(2)build.prop

*VERSION INFO 输出下列信息   
当前时间   
当前内核版本 : 可以读取文件 (/proc/version) 获得   
显示当前命令 : 可以读取文件夹 (/proc/cmdline) 获得   
显示系统 build 的一些属性 : 可以读取文件 (/system/build.prop) 获得   
输出系统一些属性   
gsm.version.ril-impl  
gsm.version.baseband  
gsm.imei  
gsm.sim.operator.numeric  
gsm.operator.alpha*

(3)dumpsys

*执行 /system/bin/dumpsys 后可以获得这个 log.  
经常会发现该 log 输出不完整 , 因为代码里面要求该工具最多只执行 60ms, 可能会导致 log 无法完全输出来 .  
可以通过修改时间参数来保证 log 完全输出 .  
信息 :  
Currently running services  
DUMP OF SERVICE services-name(running)*

Log Code Analysis

*Site: .\frameworks\base\cmds\dumpstate\*

*相关 Log 程序的代码可以从上面目录获取*

Log Analysis Experience

分析步骤

*1. 查看一些版本信息   
确认问题的系统环境   
2. 查看 CPU/MEMORY 的使用状况   
看是否有内存耗尽 ,CPU 繁忙这样的背景情况出现 .  
3. 分析 traces  
因为 traces 是系统出错以后输出的一些线程堆栈信息 , 可以很快定位到问题出在哪里.   
4. 分析 SYSTEM LOG  
系统 Log 详细输出各种 log, 可以找出相关 log 进行逐一分析*