TPVisonDebug log分析

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Comment | Date |
| 1.0 | Wujun | First version | 2019-07-22 |

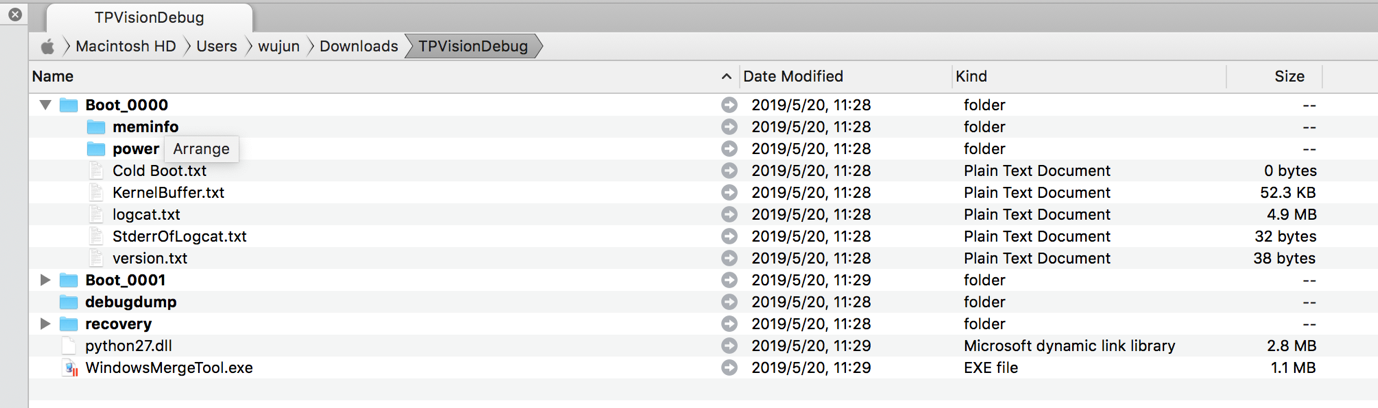
# 常用工具介绍

1. glogd， 这一个是免费的工具，在公司中可以免费的安装使用， 不会被公司稽核。（公司对于非法软件管控非常严格，盗版-破解版绿色版，社区版，个人版等不可用于商业应用的必须严格禁止。而且IT有后台稽核，如果被稽核到，会受到相应的处罚，主管也会有连带责任）。
   1. 可以打开大文件，如上G的文件，
   2. 搜索可以使用’|’ （或操作）进行多关键字查询
   3. 确定是分类高亮不太方面，不过也可以通过其菜单设置关键字的高亮色
2. Notepad++ 开源软件，可以在公司办公电脑使用，方面的用于不是很大的log，也可以用于简单编辑与查看一些文件。

# TPVsionDebug目录介绍

TPVisionDebug，是我们公司为了简化log采集，从而客制化的一套log抓去流程，只需要使用U盘即可（在使用之前，需要在根目录创建。 待补充）

所有的log都会采集到logcat目录下面，所以会看到下面一些目录



根据目录结果来介绍

1. WindowMergeTool.exe是在window 64bit的文件合成工具，当点击后，会将Boot\_00X中的logcat全部合并。
2. Debugdump ：这个目录中保存的系统在运行过程中生成的， ANR, Crash等文件，最后面的是使用PID结尾
   1. ANR zip里面包含的内容
      1. 系统会保留ANR时的部分logcat，
      2. 当时的kernel的部分信息，
      3. 以及与对应的trace文件，在trace文件里，已经客制化，一并打印出发生anr时，与之相关的父进程与子进程想着的backtrace，用于查看deadlock与当时发生的操作。

【所以在trace发生时，进程信息会比较多，如果要查看当前进程的PID来查找，而且ANR主要发生在主线程，所以优先查看这一部分log】

* 1. Crash文件，里面有相应的tombstone等信息可以很方面的查看

# UART Log

UART log中包含的主要是MTK中的很多Middleware的信息，同时也会有我们redirect到Kernel中的log（这一部分会比较少，与power想关的会有这样子的处理。），根据经验，当测试时，最好请DQE team提供。

对于我们来说，主要查看的Boot Reason，主要分为WD0（MTK soc WD0， 重启，Power过程中的pause Resume等，都需要uart log来获取）

# 目录生成原理及使用方法

这些目录可能与Android原来的不太一致，但是最后的内容都是一样的。

1. Logcat与串口输出的以及android studio中打印的完全一致。（当然会有log丢失的情况，所以在DOD中会有规定log的数量，log太多会有Stability测试中造成系统的loading很重，导致系统CPU过高，从而导致ANR.）
2. Boot\_0001：
   1. U盘插入时，如果没有相应的目录，则会创建一个，当再次插入时会生成Boot\_0002，一次递增，
   2. 同时当系统重启或进行Power on/off时，Boot\_000x也会是顺序递增。（这个在Stability测试时经常用到。）
3. 大家在开发的时候也可以使用这种方式抓去log，然后再离线分析。特别是与DQE部门进行沟通交流，或者跨site进行分析时，都比较有用。

# 常见问题分析

1. ANR（**A**pplication is **N**ot **R**esponding）
   1. Input dispatch Timeout
      1. UI stuck
      2. Time consuming operations in mainThread.
   2. Service timeout
      1. Start time out
      2. Handling timeout
2. Crash
   1. Fatal exception，对于java都会有打印
   2. Native， 可以通过sig：9来搜索查询。
3. System Reboot，

# 常用关键字

1. ANR， 当有ANR 发生时，都会有ANR in使用
2. FATAL EXCEPTION
3. CPU usage， 用于查看系统运行时cpu信息，在我司系统中，当cpu超过95%时，会打印前20的占用cpu的fjincheng列表。
4. Memory usage可从logcat目录中找到。 与之前相关的是oom与memleak。
5. 各模块自己的关键字。
6. SIG:9 来查询我们有哪些进程被干掉。
7. Start u0，来查看程序被启动的顺序
8. Activity与Service的lifecycle的部分，onCreate, onStart, onStop, onResume, onDestory, etc.
9. AVC ，用于查找违反selinux规则的事项

# 常见问题分析

Attach the document from BLR etc.