# TODO

adb shell dumpsys activity b –help查询广播 按照包名

# cmds架构

先举一个示例，怎么通过android系统的命令，找到对应的代码，以input为例。有两个方法：

方法1：猜测，一般cmd都在framework/base/cmds，然后一看就知道是input这个包

方法2：mk索引，一定有个module的名字是input

方法3：源码分析，这也是重点介绍的

找到input 命令的位置system//bin/input，内容如下:

# Script to start "input" on the device, which has a very rudimentary  
# shell.  
#  
base=/system  
export CLASSPATH=$base/framework/input.jar  
exec app\_process $base/bin com.android.commands.input.Input "$@"

对应的入口函数

find ./ -iname  input

/frameworks/base/cmds/input/src/com/android/commands/input/Input.java

实际开发中，靠猜就好了。直接去framework/base/cmds找，实在不行再通过源码分析

# Input

input也是一个进程，在framework/base/cmds目录下

## 模拟事件分类

D

private static final Map<String, Integer> SOURCES = new HashMap<String, Integer>() {{

put("keyboard", InputDevice.SOURCE\_KEYBOARD);

put("dpad", InputDevice.SOURCE\_DPAD);

put("gamepad", InputDevice.SOURCE\_GAMEPAD);

put("touchscreen", InputDevice.SOURCE\_TOUCHSCREEN);

put("mouse", InputDevice.SOURCE\_MOUSE);

put("stylus", InputDevice.SOURCE\_STYLUS);

put("trackball", InputDevice.SOURCE\_TRACKBALL);

put("touchpad", InputDevice.SOURCE\_TOUCHPAD);

put("touchnavigation", InputDevice.SOURCE\_TOUCH\_NAVIGATION);

put("joystick", InputDevice.SOURCE\_JOYSTICK);

}};

## 源码

Input

1. private void run(String[] args) {
2. if (args.length < 1) {
3. showUsage();
4. return;
5. }
7. int index = 0;
8. String command = args[index];
9. int inputSource = InputDevice.SOURCE\_UNKNOWN;
10. if (SOURCES.containsKey(command)) {
11. inputSource = SOURCES.get(command);
12. index++;
13. command = args[index];
14. }
15. final int length = args.length - index;
17. try {
18. if (command.equals("text")) {
19. if (length == 2) {
20. inputSource = getSource(inputSource, InputDevice.SOURCE\_KEYBOARD);
21. sendText(inputSource, args[index+1]);
22. return;
23. }
24. } else if (command.equals("keyevent")) {
25. if (length >= 2) {
26. final boolean longpress = "--longpress".equals(args[index + 1]);
27. final int start = longpress ? index + 2 : index + 1;
28. inputSource = getSource(inputSource, InputDevice.SOURCE\_KEYBOARD);
29. if (length > start) {
30. for (int i = start; i < length; i++) {
31. int keyCode = KeyEvent.keyCodeFromString(args[i]);
32. if (keyCode == KeyEvent.KEYCODE\_UNKNOWN) {
33. keyCode = KeyEvent.keyCodeFromString("KEYCODE\_" + args[i]);
34. }
35. sendKeyEvent(inputSource, keyCode, longpress);
36. }
37. return;
38. }
39. }
40. }.........

最终调用InputManager 完成所有事件的转发

private void injectKeyEvent(KeyEvent event) {

Log.i(TAG, "injectKeyEvent: " + event);

InputManager.getInstance().injectInputEvent(event,

InputManager.INJECT\_INPUT\_EVENT\_MODE\_WAIT\_FOR\_FINISH);

}

/\*\*

\* Builds a MotionEvent and injects it into the event stream.

\*

\* @param inputSource the InputDevice.SOURCE\_\* sending the input event

\* @param action the MotionEvent.ACTION\_\* for the event

\* @param when the value of SystemClock.uptimeMillis() at which the event happened

\* @param x x coordinate of event

\* @param y y coordinate of event

\* @param pressure pressure of event

\*/

private void injectMotionEvent(int inputSource, int action, long when, float x, float y, float pressure) {

final float DEFAULT\_SIZE = 1.0f;

final int DEFAULT\_META\_STATE = 0;

final float DEFAULT\_PRECISION\_X = 1.0f;

final float DEFAULT\_PRECISION\_Y = 1.0f;

final int DEFAULT\_EDGE\_FLAGS = 0;

MotionEvent event = MotionEvent.obtain(when, when, action, x, y, pressure, DEFAULT\_SIZE,

DEFAULT\_META\_STATE, DEFAULT\_PRECISION\_X, DEFAULT\_PRECISION\_Y,

getInputDeviceId(inputSource), DEFAULT\_EDGE\_FLAGS);

event.setSource(inputSource);

Log.i(TAG, "injectMotionEvent: " + event);

InputManager.getInstance().injectInputEvent(event,

InputManager.INJECT\_INPUT\_EVENT\_MODE\_WAIT\_FOR\_FINISH);

}

# Screencap截屏

源码路径：frameworks/base/cmds/screencap/

可执行文件：system/bin/screencap

## 使用介绍

usage: screencap [-hp] [-d display-id] [FILENAME]

   -h: this message

   -p: save the file as a png.

   -d: specify the display id to capture, default 0.

If FILENAME ends with .png it will be saved as a png.

If FILENAME is not given, the results will be printed to stdout.

使用相当简单：

adb shell screencap -p /sdcard/c1.png

adb pull /sdcard/c1.png

adb shell screencap -p /sdcard/c2.png

adb pull /sdcard/c2.png

adb shell screencap -p /sdcard/findui/c2.png

adb pull /sdcard/findui

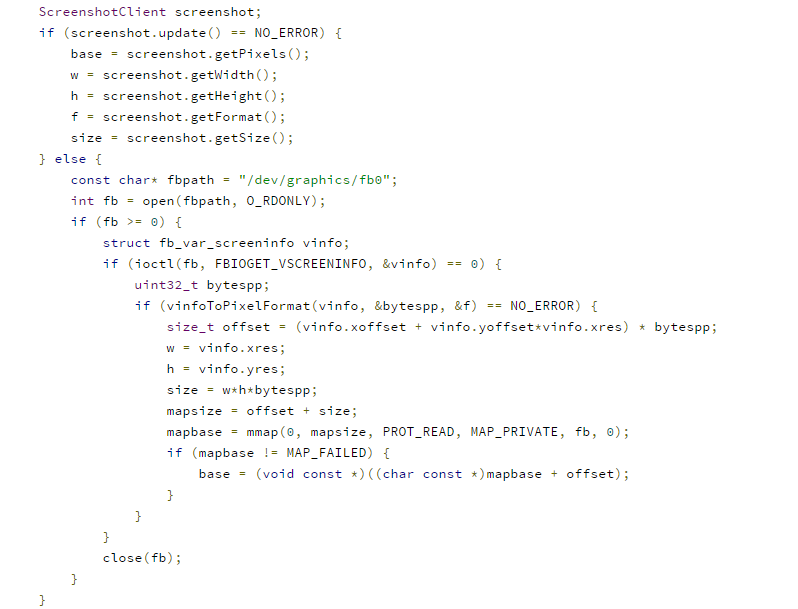
备注：

-d 参数为支持扩展的display设备

如果文件名以.png结尾时，它将保存为png文件，如果文件名没有给出,则结果被会被输出到stdout（导出会相当的卡

## 源码分析

核心代码



Fgx

整个程序最核心的代码就是上图。

获取截图信息的流程是通过ScreenshotClient对象，如果发现ScreenshotClient获取失败则读取/dev/graphics/fb0文件。

不知道/dev/graphics/fb0是什么文件，可以google下。

1、 fb0读取比较容易看懂：（如果有兴趣的话可以继续去看linux内核文件中的显示文件）

直接open()打开文件

通过fb\_var\_screeninfo结构体来定位fb0的文件头格式

根据偏移地址把一贞的数据读到一个mmap中，

base = (void const \*)((char const \*)mapbase + offset);

2、ScreenshotClient:

创建ScreenshotClient screenshot

直接调用screenshot.update()

base = screenshot.getPixels();

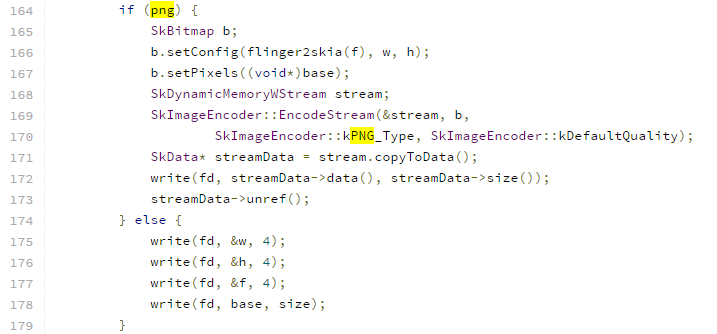
后续我们再继续阅读ScreenshotClient类，了解它是如何显示屏幕显示获取的：

ScreenshotClient类位于frameworks/native/libs/gui/SurfaceComposerClient.cpp中。

Screencap的实现就是通过访问“SurfaceFlinger”服务或者读取fb0文件进行截屏的。

最后我们看看Screencap保存图片的代码：

4

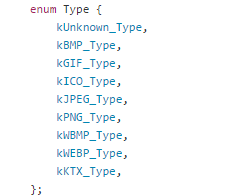


程序只提供了png后缀的图片格式，如果想要返回其他的图片格式，需要要进行类型type的修改。

图片格式：

1、SkImageEncoder 中的Type 枚举（注意有些android版本是没有完全实现的）

5



2、kDefaultQuality默认值为80 图片的清晰度

如果想提高这个工具的截屏效率，可以通过选择图片格式或者选择图片的质量。

总结：

经实际测试，1s 不到10帧，screencap效率是相当差的（包含保存为jpeg等格式）。

如果需要修改保存图片的格式，需要修改源码重新编译。

https://blog.csdn.net/omnispace/article/details/72802450

# Android cmd

## Start/stop

## am

Am extends BaseCommand

adb shell am broadcast -a android.intent.action.test.XX --ei ei3 1：发送传整数ei3 = 1d的XX广播

adb shell am start -n packageName/ActivityFullName：发送启动Activity的广播

am force-stop packageName

adb shell am start -a android.intent.action.ACTION\_REQUEST\_SHUTDOWN -d <http://gityuan.com>

adb shell am broadcast -a android.intent.action.ACTION\_REQUEST\_SHUTDOWN --ez android.intent.extra.KEY\_CONFIRM false

adb shell am start -n com.android.settings/com.android.settings.FallbackHome

查询版本

rm500:/ $ ps | grep dji

root 144 2 0 0 0 0000000000 S dji\_bat\_charge\_

root 325 1 45520 10472 0 0000000000 S /system/bin/djilink

system 1291 326 1574668 77052 0 0000000000 S com.dji

u0\_a38 1498 327 1819772 156924 0 0000000000 R dji.go.v4

u0\_a38 1523 1498 1799968 57196 0 0000000000 S dji.go.v4

rm500:/ $ dumpsys package dji.go.v4 | grep version

versionCode=31245 minSdk=19 targetSdk=21

versionName=4.2.12

## input

模拟上下左右 确定键值  
adb shell input keyevent  "value"  
value:是键值对应的数值  
 KEYCODE\_DPAD\_UP     19  
 KEYCODE\_DPAD\_DOWN  20  
 KEYCODE\_DPAD\_LEFT 21  
 KEYCODE\_DPAD\_RIGHT 22  
 KEYCODE\_DPAD\_CENTER 23

其他模拟手段：

<https://itimetraveler.github.io/2016/06/02/%E3%80%90Android%E3%80%91%E6%8A%80%E6%9C%AF%E8%B0%83%E7%A0%94%EF%BC%9A%E7%94%A8%E4%BB%A3%E7%A0%81%E6%A8%A1%E6%8B%9F%E5%B1%8F%E5%B9%95%E7%82%B9%E5%87%BB%E3%80%81%E8%A7%A6%E6%91%B8%E4%BA%8B%E4%BB%B6/>

## 实用功能

* 1、 adb shell input keyevent <keycode>  
  adb 命令代替键盘操作，不同的 keycode 能实现不同的功能  
  keycode 移步：<https://developer.android.com/reference/android/view/KeyEvent.html>
* 2、adb shell input text 123  
  输入文本在焦点处于某文本框时，可以通过 input 命令来输入文本。现在 123 出现在文本框中。
* 3、adb shell screencap -p /sdcard/sc.png  
  屏幕截图
* 4、adb shell screenrecord /sdcard/filename.mp4  
  录制 mp4 格式的视频保存到 /sdcard

# Pm

frameworks/base/cmds/pm/src/com/android/commands/pm/Pm.java

pm clear packagename 在root设备才生效！

pm list packages –f

查看所有已安装的应用的包名

pm disable <package name>

禁用启用app

* pm list packages [-f] [-d] [-e] [-s] [-3] [-i] [-u] [--user USER\_ID] [FILTER]  
  查看应用列表,**pm即PackageManager**

| **参数** | **解释** |
| --- | --- |
| -f | 显示应用关联的 apk 文件 |
| -d | 只显示 disabled 的应用 |
| -e | 只显示 enabled 的应用 |
| -s | 只显示系统应用 |
| -3 | 只显示第三方应用 |
| -i | 显示应用的 installer |
| -u | 包含已卸载应用 |
| <FILTER> | 包名包含 <FILTER> 字符串 |

# Settings

SettingsCmd.java

https://blog.csdn.net/u012439416/article/details/61930082

# dumpsys

http://gityuan.com/2016/05/14/dumpsys-command/

在Android开发过程中，经常会遇到很多随机的问题，比如CPU过高、内存泄露等。Dumpsys是一个可以查看系统信息的命令，便于我们通过系统日志快速定位问题。

## dumpsys实现原理

dumpsys工具比较简单，通过调用ServiceManager服务的listServices来查询系统注册的所有服务，并通过checkService接口来获取服务的Binder远程代理对象，使用每个服务的dump函数打印该服务的相关信息。

相关源码路径：

./frameworks/native/cmds/dumpsys/dumpsys.cpp

./frameworks/native/libs/binder/IServiceManager.cpp



图 1 dumpsys实现原理图

//dumpsys.cpp的函数入口

int main(int argc, char\* const argv[])

{

signal(SIGPIPE, SIG\_IGN);

//取得ServiceManager服务的远程Binder代理对象

sp<IServiceManager> sm = defaultServiceManager();

fflush(stdout);

if (sm == NULL) {

ALOGE("Unable to get default service manager!");

aerr << "dumpsys: Unable to get default service manager!" << endl;

return 20;

}

Vector<String16> services;

Vector<String16> args;

if (argc == 1) {

//使用ServiceManager服务的Binder远程代理对象查询所有的服务

services = sm->listServices();

services.sort(sort\_func);

args.add(String16("-a"));

} else {

//添加命令行指定的服务及参数

services.add(String16(argv[1]));

for (int i=2; i<argc; i++) {

args.add(String16(argv[i]));

}

}

//根据服务名称查找服务对应的Binder代理对象，如果存在，打印该服务

const size\_t N = services.size();

if (N > 1) {

// first print a list of the current services

aout << "Currently running services:" << endl;

//调用服务的dump函数打印服务相关信息

for (size\_t i=0; i<N; i++) {

sp<IBinder> service = sm->checkService(services[i]);

if (service != NULL) {

aout << " " << services[i] << endl;

}

}

}

for (size\_t i=0; i<N; i++) {

sp<IBinder> service = sm->checkService(services[i]);

if (service != NULL) {

if (N > 1) {

aout << "------------------------------------------------------------"

"-------------------" << endl;

aout << "DUMP OF SERVICE " << services[i] << ":" << endl;

}

//调用每个service中的dump方法，输出log

int err = service->dump(STDOUT\_FILENO, args);

if (err != 0) {

aerr << "Error dumping service info: (" << strerror(err)

<< ") " << services[i] << endl;

}

} else {

aerr << "Can't find service: " << services[i] << endl;

}

}

return 0;

}

## dumpsys使用方法

通过service list命令查看可以用于dumpsys的所有服务

130|root@android:/ # service list

Found 77 services:

0 sip: [android.net.sip.ISipService]

…

图 3 dumpsys Service示意图

大部分Android services都是在SystemServer中的main函数里注册的，它调用了一个叫init2的函数,然后ServerThead的run方法中进行服务注册。

.\framework\base\services\java\com\android\server\SystemServer.java)



其他 native services是在SystemServer中的main函数里注册的,它调用本地函数init1()，



然后调用的是JNI的里面的android\_server\_SystemServer\_init1函数。

.\framework\base\services\jni\com\_android\_server\_SystemServer.cpp



然后调用的system\_init函数。.\frameworks\base\cmds\system\_server

\library\System\_init.cpp

### CM系统上面可支持的dumpsys的服务如下表：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HiDisplay | **SurfaceFlinger** | accessibility | account | **activity** | alarm | appwidget | hardware |
| audio | backup | battery | batteryinfo | bluetooth\_manager | clipboard | commontime\_management | connectivity |
| content | country\_detector | cpuinfo | dbinfo | device\_policy | diskstats | devicestoragemonitor | display |
| dreams | drm.drmManager | dropbox | entropy | ethernet | fm\_receiver | fm\_transmitter | gfxinfo |
| input | input\_method | isms | **meminfo** | iphonesubinfo | **mount** | assetredirection | phone |
| netpolicy | lock\_settings | notification | **package** | media.audio\_flinger | uimode | servicediscovery | power |
| netstats | pieservice | statusbar | updatelock | media.audio\_policy | usagestats | network\_management | profile |
| sip | textservices | throttle | vibrator | media.camera | search | permission |  |
| usb | telephony.registry | user | wallpaper | media.player | serial | samplingprofiler |  |
| wifip2p | simphonebook | wifi | window | sensorservice | location | scheduling\_policy |  |

黄色表示：无dump打印信息的服务，大家可以到相应服务（.\framework\native\services\）的源码下，查看dump接口，即可明白为什么黄色背景的服务没有dump打印信息。

HiDisplay ，Hisi添加的服务，用于查看支持的分辨率。参数字段为1表示支持。

红色表示：在以后调试中常用到的相关服务。

bullhead:/ # service list | grep Activity

103 activity: [android.app.IActivityManager]

## dumpsys |grep BroadcastRecord

查看Android系统广播发送记录

dumpsys |grep BroadcastRecord

dumpsys |grep android.intent.action.SCREEN\_ON

## dumpsys dropbox

Usage: dumpsys dropbox [--print|--file] [YYYY-mm-dd] [HH:MM:SS] [tag]

DropBoxManager是Android引入的用来持续化存储系统数据的机制，主要用于记录Android运行过程中，内核、系统进程、用户进程、等出现严重问题时的log.

Dropbox存储的log，位于/data/system/dropbox目录下，它可以记录的系统错误有：

* crash(应用程序强制关闭， Force Cloce)
* anr (应用程序没响应, Application Not Responding, ANR)
* wtf (What a Terrible Failure)
* strict\_mode (StrictMode Violation): 通常用来监测不应当在主线程执行的网络, 文件等操作
* lowmem (低内存): 在内存不足的时候, Android 会终止后台应用程序来释放内存, 但如果没有后台应用程序可被释放时, ActivityManagerService 就会在 DropBoxManager 中记录一次。
* watchdog：如果 WatchDog 监测到系统进程(system\_server)出现问题, 会增加一条 watchdog 记录到 DropBoxManager 中, 并终止系统进程的执行。
* netstats\_error：NetworkStatsService 负责收集并持久化存储网络状态的统计数据, 当遇到明显的网络状态错误时, 它会增加一条 netstats\_error 记录到 DropBoxManager。

## dumpsys HiDisplay

Hisi添加的服务，用于查看支持的分辨率。参数字段为1表示支持。

--- Format ---Brightness--- Contrast ---Saturation--- Hue --- DispMode ---EyePriority---

--- 6--- 50--- 50--- 50--- 50--- 0--- 0---

Screen Out Range : [0,0,0,0]

| MacroVersion : -1 | HDCP 0 |

| Is support 3d = 0 |

| HI\_UNF\_ENC\_FMT\_1080P\_60 = 1 |

| HI\_UNF\_ENC\_FMT\_1080P\_50 = 1 |

| HI\_UNF\_ENC\_FMT\_1080P\_30 = 0 |

## dumpsys activity

### 语法

dumpsys activity [options] [WHAT]

其中options为可选项，以-开头， 主要有以下几类：

| **options** | **含义** |
| --- | --- |
| -a | 包括所有可用Server状态 |
| -c | 包括Client状态，即App端情况 |
| -p PACKAGE | 限定输出指定包名 |

1.2 WHAT参数

列举常见的WHAT参数：

| **序号** | **WHAT** | **解释** | **对应源码** |
| --- | --- | --- | --- |
| 1 | a[ctivities] | activity状态 | dumpActivitiesLocked() |
| 2 | b[roadcasts] [PACKAGE\_NAME] | broadcast状态 | dumpBroadcastsLocked() |
| 3 | s[ervices] [COMP\_SPEC …] | service状态 | newServiceDumperLocked().dumpLocked |
| 4 | prov[iders] [COMP\_SPEC …] | content provider状态 | dumpProvidersLocked() |
| 5 | p[rocesses] [PACKAGE\_NAME] | 进程状态 | dumpProcessesLocked() |
| 6 | o[om] | 内存管理 | dumpOomLocked() |
| 7 | i[ntents] [PACKAGE\_NAME] | pending intent状态 | dumpPendingIntentsLocked() |
| 8 | r[ecents] | 最近activity | dumpRecentsLocked() |
| 9 | perm[issions] | URI授权情况 | dumpPermissionsLocked() |
| 10 | all | 所有activities信息 | dumpActivity() |
| 11 | top | 顶部activity信息 | dumpActivity() |
| 12 | package | package相关信息 | dump() |

其中PACKAGE\_NAME是指可跟包名，COMP\_SPEC是指可跟具体组件信息，中括号是指缩写字母；

### dumpsys activity

前面介绍dumpsys activity根据后面跟着的不同参数则输出相应的内容，当不跟任何参数， dumpsys activity等价于依次输出下面8条命令：

dumpsys activity intents

dumpsys activity broadcasts //广播

dumpsys activity providers //provider

dumpsys activity permissions

dumpsys activity services //服务

dumpsys activity recents

dumpsys activity activities //activity

dumpsys activity processes

依次简要说明这8条命令：

#### 2.1 intents

//标志性开头，dumpPendingIntentsLocked

ACTIVITY MANAGER PENDING INTENTS (dumpsys activity intents)

输出对象：

* PendingIntentRecord

#### 2.2 broadcasts

//标志性开头，dumpBroadcastsLocked

ACTIVITY MANAGER BROADCAST STATE (dumpsys activity broadcasts)

Registered Receivers：

Receiver Resolver Table：

Historical broadcasts [foreground]:

Historical broadcasts summary [foreground]:

Historical broadcasts [background]:

Historical broadcasts summary [background]:

Sticky broadcasts

mHandler

主要输出的对象：

* ReceiverList, BroadcastFilter,
* IntentResolver,
* BroadcastQueue, BroadcastRecord
* Handler, Looper

#### 2.3 provider

//标志性开头，dumpProvidersLocked

ACTIVITY MANAGER CONTENT PROVIDERS (dumpsys activity providers)

Published single-user content providers (by class):

Published user [n] content providers (by class):

Single-user authority to provider mappings:

User [n] authority to provider mappings:

主要输出的对象：

* ProviderMap
* ContentProviderRecord， ContentProviderConnection

#### 2.4 permissions

//标志性开头，dumpPermissionsLocked

ACTIVITY MANAGER URI PERMISSIONS (dumpsys activity permissions)

主要输出的对象：

* UriPermission

#### 2.5 Service

//标志性开头，newServiceDumperLocked().dumpLocked

ACTIVITY MANAGER SERVICES (dumpsys activity services)

主要输出的对象：

* ActiveServices,
* ServiceRecord, ConnectionRecord,ProcessRecord

#### 2.6 recents

//标志性开头，dumpRecentsLocked

ACTIVITY MANAGER RECENT TASKS (dumpsys activity recents)

主要输出的对象：

* TaskRecord

#### 2.7 activities

//标志性开头，dumpActivitiesLocked

ACTIVITY MANAGER ACTIVITIES (dumpsys activity activities)

Display #0 (activities from top to bottom):

Stack #0:

Task id #[n]

\* Hist #[m]:

Stack #1:

主要输出的对象：

* ActivityStackSupervisor, ActivityStack,
* TaskRecord, ActivityRecord
* ActivityThread, Activity
* ViewRootImpl
* Looper, MessageQueue

输出格式样例：

//{Hashcode #TaskId Affinity UserId 该task的Activity个数}；

TaskRecord{e6d7a8e #156 A=com.gityuan.demo U=0 sz=1}

userId=0 effectiveUid=1000 mCallingUid=1000 mCallingPackage=android

realActivity=com.gityuan.demo/.Blog

//ProcessRecord{Hashcode pid:进程名/uid}

ProcessRecord{7c8a2af 12265:com.gityuan.demo/1000}

#### 2.8 processes

//标志性开头，dumpProcessesLocked

ACTIVITY MANAGER RUNNING PROCESSES (dumpsys activity processes)

All known processes:

Isolated process list (sorted by uid):

UID states:

UID validation:

Process LRU list (sorted by oom\_adj, 60 total, non-act at 2, non-svc at 2):

PID mappings:

Foreground Processes:

主要输出的对象：

* AMS各种进程对象
* ProcessRecord, UidRecord



cmd：上表加粗项是指直接跟包名，另外services和providers还可以跟组件名；

缩写：基本都是cmd首字母或者前几个字母，用cmd和缩写是等效： dumpsys activity broadcasts dumpsys activity b //等效

列出目标设备上的activity栈(back stack)和任务(task)的信息, 还有其他组件的一些信息和一些关于进程的信息。

获取所用应用: dumpsys activity

获取自己的应用: dumpsys activity | grep com.xxx.xxx.xx

获取处于栈顶的activity: dumpsys activity | grep mFocusedActivity

adb shell dumpsys activity b --help

### adb shell dumpsys activity b

子命令格式：

adb shell dumpsys activity [-a] [-c]…

|  |  |
| --- | --- |
| 参数 | 说明 |
| -a | 包括所有可用的服务器状态 |
| -c | 包括客户端状态 |
| -p | 限制输出为给定的包，例如：  adb shell dumpsys activity -p com.android.browser |
| -h | 打印帮助信息 |
| cmd | 子命令 |

 adb shell dumpsys activity b

|  |  |
| --- | --- |
| cmd子命令 | 说明 |
| a[ctivities] | activity堆栈状态 |
| r[recents] | 最近activity的状态 |
| b[rodacasts] [package\_name] [histpry [-s]] | 广播状态 |
| i[ntents] [package\_name] | 挂起的intent状态 |
| p[rocesses] [package\_name] | 进程状态 |
| o[om] | oom管理 |
| perm[issions] | url权限授权状态 |
| prov[iders] [comp\_spec…] | content provider状态 |
| provider [comp\_spec] | provider客户端状态 |
| s[ervices] [comp\_spec…] | 服务状态 |
| as[sociations] | 跟踪应用程序的关联 |
| service [comp\_spec] | 服务客户端状态 |
| package [package\_name] | 给的包的所有状态 |
| all | 转储所有的activityes |
| top | 转储栈顶的activity |
| write | 写入所有挂起状态存储 |
| track-associations | 允许会话跟踪 |
| untrack-associations | 禁用和明确会话跟踪，命令参数可能也是一个comp\_spec 转储的activity |

### dumpsys activity –h

通过帮助可以了解到dumpsys activity的全部用法。

### dumpsys activity a

Main stack:表示当前运行activity的应用。

Running activities：表示正在运行的activity。

Recent tasks:表示最近运行的activiy顺序。

Main stack:

\* TaskRecord{40f26e18 #3 A com.pplive.androidpad U 0}

numActivities=1 rootWasReset=true userId=0

affinity=com.pplive.androidpad

…………………………………………….

…………………………………………….

Running activities (most recent first):

TaskRecord{40f26e18 #3 A com.pplive.androidpad U 0}

Run #1: ActivityRecord{4123fb58 u0 com.pplive.androidpad/.ui.RecommendActivity}

TaskRecord{40f49860 #2 A com.cyanogenmod.trebuchet U 0}

Run #0: ActivityRecord{40f47930 u0 com.cyanogenmod.trebuchet/.Launcher}

Recent tasks:

\* Recent #0: TaskRecord{40f26e18 #3 A com.pplive.androidpad U 0}

numActivities=1 rootWasReset=true userId=0

\* Recent #1: TaskRecord{40f648f0 #4 I com.android.deskclock/.AlarmAlertFullScreen U 0}

intent={flg=0x800000 cmp=com.android.deskclock/.AlarmAlertFullScreen}

--------------------------------------------------------------------------------------------------------------------------------

### **场景1：查询某个App所有的Service状态**

dumpsys activity s com.sina.weibo



解读：

* Service类名为com.morgoo.droidplugin.PluginManagerService；
* 运行在进程pid=7220，进程名为com.sina.weibo，uid=10094；
* 通过bindService连接该服务的进程pid=7306，进程名为com.sina.weibo:PluginP03。

### **场景2：查询某个App所有的广播状态**

dumpsys activity s com.sina.weibo



* android.intent.action.SCREEN\_ON代表手机亮屏广播；
* 接收该广播的receiver有很多个，其中一个所在进程为pid=7220，进程名为com.sina.weibo

### **场景3：查询某个App所有的Activity状态**

* 输出结果较多，尤其是View Hierarchy，下面截取部分：

dumpsys activity a com.sina.weibo



解读：

* 格式：TaskRecord{Hashcode #TaskId Affinity UserId=0 Activity个数=1}；所以上图信息解析后就是TaskId=1802，Affinity=com.sina.weibo，当前Task中Activity个数为1。
* effectiveUid为当前task所属Uid，mCallingUid为调用者Uid=u0a94，mCallingPackage为调用者包名，这里是com.sina.weibo；
* realActivity:task中的已启动的Activity组件名com.sina.weibo/.SplashActivity。

### **场景4：查询某个App的进程状态**

dumpsys activity p com.sina.weibo



* 格式：ProcessRecord{Hashcode pid:进程名/uid}，进程pid=7306，进程名为com.sina.weibo:PluginP03，uid=10094.
* 该进程中还有Services，Connections, Providers, Receivers，可以看出该进程是没有Activity的进程。

### **其他**

* 还有很多场景，会用到不同的参数，这里就不再一一列举，建议大家多去尝试，慢慢地就更加熟练，再比如：
* dumpsys activity top //当前界面app状态
* dumpsys activity oom //查看进程状态

### 栈顶详细消息

dumpsys activity top查看栈顶Activity,可以用来获取包名

可以看到Activity的名字和状态以及这个Activity的视图结构，很厉害的样子

### 参考

http://gityuan.com/2017/07/04/ams\_dumpsys/



## dumpsys appwidget

User: 0

Providers:

[0] provider com.andrew.apollo/.appwidgets.AppWidgetSmall:

min=(64001x10241) minResize=(64001x10241) updatePeriodMillis=0 resizeMode=03 autoAdvanceViewId=-1 initialLayout=#7f030009 uid=10000 zombie=false

[1] provider com.andrew.apollo/.appwidgets.AppWidgetLarge:

min=(64001x32769) minResize=(64001x32769) updatePeriodMillis=0 resizeMode=03 autoAdvanceViewId=-1 initialLayout=#7f030005 uid=10000 zombie=false

[2] provider com.andrew.apollo/.appwidgets.AppWidgetLargeAlternate:

min=(64001x32769) minResize=(64001x32769) updatePeriodMillis=0 resizeMode=03 autoAdvanceViewId=-1 initialLayout=#7f030006 uid=10000 zombie=false

[3] provider com.andrew.apollo/.appwidgets.RecentWidgetProvider:

min=(64001x46081) minResize=(64001x28161) updatePeriodMillis=0 resizeMode=33 autoAdvanceViewId=-1 initialLayout=#7f030007 uid=10000 zombie=false

[4] provider com.android.browser/.widget.BookmarkThumbnailWidgetProvider:

min=(46081x46081) minResize=(28161x10241) updatePeriodMillis=0 resizeMode=31 autoAdvanceViewId=-1 initialLayout=#7f04000e uid=10004 zombie=false

[5] provider com.android.calculator2/.CalculatorWidget:

min=(64001x64001) minResize=(64001x64001) updatePeriodMillis=0 resizeMode=23 autoAdvanceViewId=-1 initialLayout=#7f040010 uid=10009 zombie=false

[6] provider com.android.calendar/.widget.CalendarAppWidgetProvider:

min=(28161x46081) minResize=(28161x28161) updatePeriodMillis=0 resizeMode=33 autoAdvanceViewId=-1 initialLayout=#7f04000d uid=10010 zombie=false

[7] provider com.android.contacts/.socialwidget.SocialWidgetProvider:

min=(66561x10241) minResize=(66561x10241) updatePeriodMillis=86400000 resizeMode=01 autoAdvanceViewId=-1 initialLayout=#7f04008e uid=10001 zombie=false

[8] provider com.android.deskclock/com.android.alarmclock.AnalogAppWidgetProvider:

min=(28161x28161) minResize=(28161x28161) updatePeriodMillis=0 resizeMode=01 autoAdvanceViewId=-1 initialLayout=#7f040006 uid=10016 zombie=false

[9] provider com.android.deskclock/com.android.alarmclock.DigitalAppWidgetProvider:

min=(115201x51201) minResize=(94721x30721) updatePeriodMillis=0 resizeMode=33 autoAdvanceViewId=-1 initialLayout=#7f040015 uid=10016 zombie=false

[10] provider com.android.email/.provider.WidgetProvider:

min=(46081x28161) minResize=(46081x28161) updatePeriodMillis=0 resizeMode=33 autoAdvanceViewId=-1 initialLayout=#7f040045 uid=10020 zombie=false

[11] provider com.android.gallery3d/.gadget.PhotoAppWidgetProvider:

min=(61441x61441) minResize=(61441x61441) updatePeriodMillis=86400000 resizeMode=01 autoAdvanceViewId=-1 initialLayout=#7f040004 uid=10024 zombie=false

[12] provider com.cyanogenmod.lockclock/.ClockWidgetProvider:

min=(76801x30721) minResize=(51201x30721) updatePeriodMillis=0 resizeMode=33 autoAdvanceViewId=-1 initialLayout=#7f030000 uid=10034 zombie=false

[13] provider com.android.quicksearchbox/.SearchWidgetProvider:

min=(66561x10241) minResize=(66561x10241) updatePeriodMillis=0 resizeMode=01 autoAdvanceViewId=-1 initialLayout=#7f040006 uid=10042 zombie=false

[14] provider com.android.settings/.widget.SettingsAppWidgetProvider:

min=(83201x12801) minResize=(83201x12801) updatePeriodMillis=0 resizeMode=01 autoAdvanceViewId=-1 initialLayout=#7f0400b7 uid=1000 zombie=false

[15] provider net.cactii.flash2/.TorchWidgetProvider:

min=(8961x10241) minResize=(8961x10241) updatePeriodMillis=3600000 resizeMode=03 autoAdvanceViewId=-1 initialLayout=#7f030004 uid=10048 zombie=false

AppWidgetIds:

[0] id=1

hostId=1024 com.cyanogenmod.trebuchet/10049

provider=com.andrew.apollo/.appwidgets.AppWidgetLarge

host.callbacks=com.android.internal.appwidget.IAppWidgetHost$Stub$Proxy@425a4070

views=android.widget.RemoteViews@425c53e8

[1] id=2

hostId=1024 com.cyanogenmod.trebuchet/10049

provider=com.cyanogenmod.lockclock/.ClockWidgetProvider

host.callbacks=com.android.internal.appwidget.IAppWidgetHost$Stub$Proxy@425a4070

views=android.widget.RemoteViews@42959f58

Hosts:

[0] hostId=1024 com.cyanogenmod.trebuchet/10049:

callbacks=com.android.internal.appwidget.IAppWidgetHost$Stub$Proxy@425a4070

instances.size=2 zombie=false

Deleted Providers:

Deleted Hosts:

## dumpsys connectivity

Connectivity Service提供有关网络当前状态信息。

dumpsys connectivity用于查看wifi和以太网网络连接状态、IP地址等信息。

NetworkStateTracker for WIFI:

NetworkInfo: type: wifi[], state: UNKNOWN/IDLE, reason: (unspecified), extra: (none), roaming: false, failover: false, isAvailable: false

{LinkAddresses: [] Routes: [] DnsAddresses: [] Domains: null}

android.net.wifi.WifiStateTracker@4276f068

NetworkStateTracker for BLUETOOTH:

NetworkInfo: type: BLUETOOTH\_TETHER[], state: UNKNOWN/IDLE, reason: (unspecified), extra: (none), roaming: false, failover: false, isAvailable: false

{LinkAddresses: [] Routes: [] DnsAddresses: [] Domains: null}

android.bluetooth.BluetoothTetheringDataTracker@4253c1b8

NetworkStateTracker for ETHERNET:

Active network: ethernet

NetworkInfo: type: ethernet[], state: CONNECTED/CONNECTED, reason: (unspecified), extra: (none), roaming: false, failover: false, isAvailable: true

{InterfaceName: eth0 LinkAddresses: [10.9.8.121/24,] Routes: [10.9.8.0/24 -> 0.0.0.0,0.0.0.0/0 -> 10.9.8.254,] DnsAddresses: [10.9.8.254,] Domains: }

android.net.ethernet.EthernetStateTracker@4243e6b0

NetworkStateTracker for WIFI\_P2P:

NetworkInfo: type: wifi\_p2p[], state: UNKNOWN/IDLE, reason: (unspecified), extra: (none), roaming: false, failover: false, isAvailable: false

{LinkAddresses: [] Routes: [] DnsAddresses: [] Domains: null}

android.net.wifi.WifiStateTracker@4253b7e0

Network Requester Pids:

9:

1:

13:

7:

FeatureUsers:

NetworkTranstionWakeLock is currently not held.

It was last requested for

mUpstreamIfaceTypes:

Tether state:

Inet condition reports:

## dumpsys content

data connected: true

auto sync: u0=true

memory low: false

accounts: 0

now: 229020101 (2014-03-02 14:54:40)

offset: 23:44:56 (HH:MM:SS)

uptime: 63:37:00 (HH:MM:SS)

time spent syncing: 0:00 (HH:MM:SS), sync not in progress

next alarm time: 230420022 (23:19 (HH:MM:SS) from now)

notification info: isActive false, startTime null

Active Syncs: 0

SyncQueue: 0 operation(s)

Sync Status

Sync adapters for UserInfo{0:Owner:13}:

ServiceInfo: SyncAdapterType {name=com.android.email.provider, type=com.android.exchange, userVisible=true, supportsUploading=false, isAlwaysSyncable=false, allowParallelSyncs=false, settingsActivity=null}, ComponentInfo{com.android.exchange/com.android.exchange.EmailSyncAdapterService}, uid 10021

ServiceInfo: SyncAdapterType {name=com.android.email.provider, type=com.android.email, userVisible=true, supportsUploading=false, isAlwaysSyncable=false, allowParallelSyncs=false, settingsActivity=null}, ComponentInfo{com.android.email/com.android.email.service.PopImapSyncAdapterService}, uid 10020

ServiceInfo: SyncAdapterType {name=com.android.calendar, type=com.android.exchange, userVisible=true, supportsUploading=true, isAlwaysSyncable=false, allowParallelSyncs=false, settingsActivity=null}, ComponentInfo{com.android.exchange/com.android.exchange.CalendarSyncAdapterService}, uid 10021

ServiceInfo: SyncAdapterType {name=com.android.contacts, type=com.android.exchange, userVisible=true, supportsUploading=true, isAlwaysSyncable=false, allowParallelSyncs=false, settingsActivity=null}, ComponentInfo{com.android.exchange/com.android.exchange.ContactsSyncAdapterService}, uid 10021

Observer tree:

settings/secure/long\_press\_timeout: pid=1974 uid=1000 user=0 target=42555fd8

settings/secure/incall\_power\_button\_behavior: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/secure/ring\_home\_button\_behavior: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/secure/default\_input\_method: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/secure/default\_input\_method: pid=1974 uid=1000 user=0 target=42711a88

settings/secure/enabled\_input\_methods: pid=1974 uid=1000 user=0 target=42711a88

settings/secure/selected\_input\_method\_subtype: pid=1974 uid=1000 user=0 target=42711a88

settings/secure/accessibility\_enabled: pid=1974 uid=1000 user=-1 target=42578ca0

settings/secure/touch\_exploration\_enabled: pid=1974 uid=1000 user=-1 target=42578ca0

settings/secure/accessibility\_display\_magnification\_enabled: pid=1974 uid=1000 user=-1 target=42578ca0

settings/secure/enabled\_accessibility\_services: pid=1974 uid=1000 user=-1 target=42578ca0

settings/secure/touch\_exploration\_granted\_accessibility\_services: pid=1974 uid=1000 user=-1 target=42578ca0

settings/secure/pubkey\_blacklist: pid=1974 uid=1000 user=0 target=426adff0

settings/secure/serial\_blacklist: pid=1974 uid=1000 user=0 target=426a5c30

settings/secure/adb\_port: pid=1974 uid=1000 user=0 target=425fb080

settings/secure/screensaver\_enabled: pid=1974 uid=1000 user=-1 target=426c20d8

settings/secure/screensaver\_activate\_on\_sleep: pid=1974 uid=1000 user=-1 target=426c20d8

settings/secure/screensaver\_activate\_on\_dock: pid=1974 uid=1000 user=-1 target=426c20d8

settings/secure/locationPackagePrefixBlacklist: pid=1974 uid=1000 user=-1 target=425e9f88

settings/secure/locationCoarseAccuracy: pid=1974 uid=1000 user=0 target=424c1e80

settings/secure/location\_providers\_allowed: pid=1974 uid=1000 user=-1 target=424b9f40

settings/secure/location\_providers\_allowed: pid=2092 uid=10046 user=0 target=42667c30

settings/secure/user\_setup\_complete: pid=2092 uid=10046 user=0 target=426d3800

settings/secure/kill\_app\_longpress\_back: pid=2092 uid=10046 user=-1 target=42592100

settings/system: pid=2300 uid=10049 user=0 target=42676dd0

settings/system: pid=2300 uid=10049 user=0 target=429f7770

settings/system: pid=2300 uid=10049 user=0 target=42a288c8

settings/system: pid=2300 uid=10049 user=0 target=425d61b0

settings/system: pid=2300 uid=10049 user=0 target=42a18cc0

settings/system: pid=2300 uid=10049 user=0 target=42461128

settings/system/battery\_light\_enabled: pid=1974 uid=1000 user=0 target=42371158

settings/system/battery\_light\_pulse: pid=1974 uid=1000 user=0 target=42371158

settings/system/battery\_light\_low\_color: pid=1974 uid=1000 user=0 target=42371158

settings/system/battery\_light\_medium\_color: pid=1974 uid=1000 user=0 target=42371158

settings/system/battery\_light\_full\_color: pid=1974 uid=1000 user=0 target=42371158

settings/system/quiet\_hours\_enabled: pid=1974 uid=1000 user=0 target=42371158

settings/system/quiet\_hours\_enabled: pid=1974 uid=1000 user=0 target=4270f348

settings/system/quiet\_hours\_start: pid=1974 uid=1000 user=0 target=42371158

settings/system/quiet\_hours\_start: pid=1974 uid=1000 user=0 target=4270f348

settings/system/quiet\_hours\_end: pid=1974 uid=1000 user=0 target=42371158

settings/system/quiet\_hours\_end: pid=1974 uid=1000 user=0 target=4270f348

settings/system/quiet\_hours\_dim: pid=1974 uid=1000 user=0 target=42371158

settings/system/quiet\_hours\_dim: pid=1974 uid=1000 user=0 target=4270f348

settings/system/end\_button\_behavior: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/home\_wake\_screen: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/volume\_wake\_screen: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/volbtn\_music\_controls: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/accelerometer\_rotation: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/user\_rotation: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/screen\_off\_timeout: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/screen\_off\_timeout: pid=1974 uid=1000 user=-1 target=426c20d8

settings/system/pointer\_location: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/fancy\_rotation\_anim: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/expanded\_desktop\_state: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/expanded\_desktop\_state: pid=2092 uid=10046 user=-1 target=42592100

settings/system/expanded\_desktop\_style: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/expanded\_desktop\_style: pid=2092 uid=10046 user=-1 target=42592100

settings/system/accelerometer\_rotation\_angles: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/key\_home\_long\_press\_action: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/key\_menu\_action: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/key\_menu\_long\_press\_action: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/key\_assist\_action: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/key\_assist\_long\_press\_action: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/key\_app\_switch\_action: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/key\_app\_switch\_long\_press\_action: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/hardware\_key\_rebinding: pid=1974 uid=1000 user=-1 target=423e4cc0

settings/system/pointer\_speed: pid=1974 uid=1000 user=-1 target=423fed08

settings/system/show\_touches: pid=1974 uid=1000 user=-1 target=4273b4a0

settings/system/stylus\_icon\_enabled: pid=1974 uid=1000 user=0 target=424d7780

settings/system/swap\_volume\_keys\_on\_rotation: pid=1974 uid=1000 user=0 target=4263dc28

settings/system/status\_bar\_ime\_switcher: pid=1974 uid=1000 user=0 target=42642658

settings/system/notification\_light\_pulse: pid=1974 uid=1000 user=0 target=42728698

settings/system/notification\_light\_pulse\_default\_color: pid=1974 uid=1000 user=0 target=42728698

settings/system/notification\_light\_pulse\_default\_led\_on: pid=1974 uid=1000 user=0 target=42728698

settings/system/notification\_light\_pulse\_default\_led\_off: pid=1974 uid=1000 user=0 target=42728698

settings/system/notification\_light\_pulse\_custom\_enable: pid=1974 uid=1000 user=0 target=42728698

settings/system/notification\_light\_pulse\_custom\_values: pid=1974 uid=1000 user=0 target=42728698

settings/system/quiet\_hours\_mute: pid=1974 uid=1000 user=0 target=4270f348

settings/system/quiet\_hours\_still: pid=1974 uid=1000 user=0 target=4270f348

settings/system/mode\_ringer\_streams\_affected: pid=1974 uid=1000 user=0 target=42496408

settings/system/volume\_link\_notification: pid=1974 uid=1000 user=0 target=42496408

settings/system/safe\_headset\_volume: pid=1974 uid=1000 user=0 target=42496408

settings/system/volume\_keys\_control\_ring\_stream: pid=1974 uid=1000 user=0 target=42496408

settings/system/vibrate\_input\_devices: pid=1974 uid=1000 user=-1 target=425bde08

settings/system/auto\_brightness\_lux: pid=1974 uid=1000 user=-1 target=42593288

settings/system/auto\_brightness\_backlight: pid=1974 uid=1000 user=-1 target=42593288

settings/system/screen\_brightness: pid=1974 uid=1000 user=-1 target=426c20d8

settings/system/screen\_brightness: pid=2092 uid=10046 user=0 target=424942e0

settings/system/screen\_brightness\_mode: pid=1974 uid=1000 user=-1 target=426c20d8

settings/system/screen\_brightness\_mode: pid=2092 uid=10046 user=0 target=424942e0

settings/system/screen\_brightness\_mode: pid=2092 uid=10046 user=0 target=4269a730

settings/system/auto\_brightness\_responsiveness: pid=1974 uid=1000 user=-1 target=426c20d8

settings/system/status\_bar\_collapse\_on\_dismiss: pid=2092 uid=10046 user=0 target=4274a998

settings/system/status\_bar\_battery: pid=2092 uid=10046 user=0 target=42453af8

settings/system/status\_bar\_battery: pid=2092 uid=10046 user=0 target=4276e268

settings/system/status\_bar\_battery: pid=2092 uid=10046 user=0 target=42494958

settings/system/bugreport\_in\_power\_menu: pid=2092 uid=10046 user=0 target=424942e0

settings/system/next\_alarm\_formatted: pid=2092 uid=10046 user=0 target=424942e0

settings/system/quick\_settings\_tiles: pid=2092 uid=10046 user=-1 target=42454728

settings/system/qs\_dyanmic\_alarm: pid=2092 uid=10046 user=-1 target=42454728

settings/system/qs\_dyanmic\_bugreport: pid=2092 uid=10046 user=-1 target=42454728

settings/system/qs\_dyanmic\_dock\_battery: pid=2092 uid=10046 user=-1 target=42454728

settings/system/qs\_dyanmic\_ime: pid=2092 uid=10046 user=-1 target=42454728

settings/system/qs\_dyanmic\_usbtether: pid=2092 uid=10046 user=-1 target=42454728

settings/system/qs\_dyanmic\_wifi: pid=2092 uid=10046 user=-1 target=42454728

settings/system/expanded\_view\_widget: pid=2092 uid=10046 user=0 target=42667c30

settings/system/expanded\_hide\_scrollbar: pid=2092 uid=10046 user=0 target=42667c30

settings/system/expanded\_haptic\_feedback: pid=2092 uid=10046 user=0 target=42667c30

settings/system/haptic\_feedback\_enabled: pid=2092 uid=10046 user=0 target=42667c30

settings/system/expanded\_widget\_buttons: pid=2092 uid=10046 user=0 target=42667c30

settings/system/expanded\_ring\_mode: pid=2092 uid=10046 user=0 target=42667c30

settings/system/vibrate\_when\_ringing: pid=2092 uid=10046 user=0 target=42667c30

settings/system/nav\_buttons: pid=2092 uid=10046 user=-1 target=42592100

settings/system/pie\_controls: pid=2092 uid=10046 user=-1 target=42592100

settings/system/pie\_positions: pid=2092 uid=10046 user=-1 target=42592100

settings/system/pie\_sensitivity: pid=2092 uid=10046 user=-1 target=42592100

settings/system/status\_bar\_brightness\_control: pid=2092 uid=10046 user=0 target=4269a730

settings/system/status\_bar\_signal: pid=2092 uid=10046 user=0 target=42663730

settings/system/status\_bar\_signal: pid=2092 uid=10046 user=-1 target=42632938

settings/system/status\_bar\_am\_pm: pid=2092 uid=10046 user=0 target=4255f290

settings/system/status\_bar\_am\_pm: pid=2092 uid=10046 user=0 target=423da5b8

settings/system/status\_bar\_clock: pid=2092 uid=10046 user=0 target=4255f290

settings/system/status\_bar\_clock: pid=2092 uid=10046 user=0 target=423da5b8

settings/system/status\_bar\_notif\_count: pid=2092 uid=10046 user=0 target=423720a0

settings/system/enable\_stylus\_gestures: pid=2300 uid=10049 user=0 target=42727690

settings/bookmarks: pid=1974 uid=1000 user=0 target=426313c0

settings/global: pid=1974 uid=1000 user=0 target=425f5d78

settings/global/wifi\_suspend\_optimizations\_enabled: pid=1974 uid=1000 user=0 target=42717528

settings/global/wifi\_networks\_available\_notification\_on: pid=1974 uid=1000 user=0 target=426362d0

settings/global/http\_proxy: pid=1974 uid=1000 user=0 target=424739f8

settings/global/wifi\_watchdog\_poor\_network\_test\_enabled: pid=1974 uid=1000 user=0 target=4248f960

settings/global/wifi\_watchdog\_on: pid=1974 uid=1000 user=0 target=4248fd00

settings/global/nsd\_on: pid=1974 uid=1000 user=0 target=42496cb0

settings/global/dock\_audio\_media\_enabled: pid=1974 uid=1000 user=0 target=42496408

settings/global/device\_provisioned: pid=1974 uid=1000 user=0 target=426694b0

settings/global/device\_provisioned: pid=2092 uid=10046 user=0 target=42749818

settings/global/sampling\_profiler\_ms: pid=1974 uid=1000 user=0 target=4254ad48

settings/global/stay\_on\_while\_plugged\_in: pid=1974 uid=1000 user=-1 target=426c20d8

settings/global/overlay\_display\_devices: pid=1974 uid=1000 user=0 target=425b3d48

settings/global/throttle\_polling\_sec: pid=1974 uid=1000 user=0 target=4245b490

settings/global/throttle\_threshold\_bytes: pid=1974 uid=1000 user=0 target=4245b490

settings/global/throttle\_value\_kbitsps: pid=1974 uid=1000 user=0 target=4245b490

settings/global/throttle\_reset\_day: pid=1974 uid=1000 user=0 target=4245b490

settings/global/throttle\_notification\_type: pid=1974 uid=1000 user=0 target=4245b490

settings/global/throttle\_help\_uri: pid=1974 uid=1000 user=0 target=4245b490

settings/global/throttle\_max\_ntp\_cache\_age\_sec: pid=1974 uid=1000 user=0 target=4245b490

settings/global/auto\_time: pid=1974 uid=1000 user=0 target=425678c0

settings/global/auto\_time: pid=2240 uid=1001 user=0 target=425259d0

settings/global/sms\_short\_code\_confirmation: pid=2240 uid=1001 user=0 target=4277d788

settings/global/sms\_outgoing\_check\_max\_count: pid=2240 uid=1001 user=0 target=4277d788

settings/global/auto\_time\_zone: pid=2240 uid=1001 user=0 target=42518120

settings/global/sms\_short\_code\_rule: pid=2240 uid=1001 user=0 target=4276f348

settings/global/data\_roaming: pid=2240 uid=1001 user=0 target=4264a4d0

com.cyanogenmod.trebuchet.settings/favorites: pid=2300 uid=10049 user=0 target=4277d240

com.cyanogenmod.trebuchet.settings/appWidgetReset: pid=2300 uid=10049 user=0 target=4266beb0

user\_dictionary/words: pid=2207 uid=10031 user=0 target=42780b70

telephony/carriers: pid=2240 uid=1001 user=0 target=426d7b40

com.android.contacts/contacts: pid=2207 uid=10031 user=0 target=42735ab8

pid 1974: 99 observers

pid 2092: 41 observers

pid 2300: 9 observers

pid 2240: 7 observers

pid 2207: 2 observers

Total number of nodes: 140

Total number of observers: 158

## dumpsys cpuinfo

查看每个进程占用用户空间和内核空间的情况。

Load: 5.08 / 5.07 / 5.07

CPU usage from 105554ms to 45554ms ago:

12% 2893/com.pplive.androidpad: 11% user + 1.3% kernel / faults: 21594 minor

2.2% 3097/com.android.deskclock: 1.4% user + 0.8% kernel / faults: 12 minor

1.8% 1590/surfaceflinger: 1.1% user + 0.6% kernel

0.4% 1596/displaysetting: 0% user + 0.4% kernel

0.3% 18/HI\_VPSS\_Process: 0% user + 0.3% kernel

0.2% 16/log\_udisk\_task: 0% user + 0.2% kernel

0% 2233/system\_server: 0% user + 0% kernel

0% 20/HI\_HDMI\_kThread: 0% user + 0% kernel

0% 1598/android\_ir\_user: 0% user + 0% kernel

0% 2347/com.android.systemui: 0% user + 0% kernel

0% 4/kworker/0:0: 0% user + 0% kernel

0% 15/kworker/1:1: 0% user + 0% kernel

0% 17/x5hdqam\_sop: 0% user + 0% kernel

0% 21/HI\_HDMI\_kCEC: 0% user + 0% kernel

0% 448/cfinteractive: 0% user + 0% kernel

0% 2538/com.cyanogenmod.trebuchet: 0% user + 0% kernel / faults: 2 minor

0% 3133/com.pplive.androidpad:remote: 0% user + 0% kernel / faults: 1 minor

9.1% TOTAL: 7.1% user + 1.9% kernel + 0% softirq

## dumpsys dbinfo

Applications Database Info:

\*\* Database info for pid 2318 [com.android.location.fused] \*\*

\*\* Database info for pid 2092 [com.android.systemui] \*\*

\*\* Database info for pid 2207 [com.android.inputmethod.latin] \*\*

\*\* Database info for pid 1974 [system] \*\*

Connection pool for /data/system/users/0/accounts.db:

Open: true

Max connections: 1

Available primary connection:

Connection #0:

isPrimaryConnection: true

onlyAllowReadOnlyOperations: true

Most recently executed operations:

0: [2014-02-28 10:18:25.208] executeForCursorWindow took 2ms - succeeded, sql="SELECT uid FROM grants GROUP BY uid"

1: [2014-02-28 10:18:25.207] prepare took 0ms - succeeded, sql="SELECT uid FROM grants GROUP BY uid"

2: [1970-01-02 00:00:06.060] executeForCursorWindow took 0ms - succeeded, sql="SELECT \_id, type, name FROM accounts"

3: [1970-01-02 00:00:06.055] prepare took 1ms - succeeded, sql="SELECT \_id, type, name FROM accounts"

4: [1970-01-02 00:00:06.054] executeForCursorWindow took 0ms - succeeded, sql="SELECT \_id, type, name FROM accounts"

5: [1970-01-02 00:00:06.053] prepare took 0ms - succeeded, sql="SELECT \_id, type, name FROM accounts"

6: [1970-01-02 00:00:06.030] executeForCursorWindow took 16ms - succeeded, sql="SELECT uid FROM grants GROUP BY uid"

7: [1970-01-02 00:00:06.029] prepare took 0ms - succeeded, sql="SELECT uid FROM grants GROUP BY uid"

8: [1970-01-02 00:00:06.029] executeForLong took 0ms - succeeded, sql="PRAGMA user\_version;"

9: [1970-01-02 00:00:06.029] prepare took 0ms - succeeded, sql="PRAGMA user\_version;"

10: [1970-01-02 00:00:06.007] executeForString took 21ms - succeeded, sql="SELECT locale FROM android\_metadata UNION SELECT NULL ORDER BY locale DESC LIMIT 1"

11: [1970-01-02 00:00:06.002] execute took 5ms - succeeded, sql="CREATE TABLE IF NOT EXISTS android\_metadata (locale TEXT)"

12: [1970-01-02 00:00:05.996] executeForLong took 5ms - succeeded, sql="PRAGMA wal\_autocheckpoint=100"

13: [1970-01-02 00:00:05.994] executeForLong took 2ms - succeeded, sql="PRAGMA wal\_autocheckpoint"

14: [1970-01-02 00:00:05.992] executeForLong took 2ms - succeeded, sql="PRAGMA journal\_size\_limit=524288"

15: [1970-01-02 00:00:05.989] executeForLong took 2ms - succeeded, sql="PRAGMA journal\_size\_limit"

16: [1970-01-02 00:00:05.986] executeForString took 3ms - succeeded, sql="PRAGMA synchronous"

17: [1970-01-02 00:00:05.979] executeForString took 7ms - succeeded, sql="PRAGMA journal\_mode=PERSIST"

18: [1970-01-02 00:00:05.958] executeForString took 21ms - succeeded, sql="PRAGMA journal\_mode"

19: [1970-01-02 00:00:05.956] executeForLong took 2ms - succeeded, sql="PRAGMA foreign\_keys"

Available non-primary connections:

<none>

Acquired connections:

<none>

Connection waiters:

<none>

Connection pool for /data/system/locksettings.db:

Open: true

Max connections: 4

Available primary connection:

Connection #0:

isPrimaryConnection: true

onlyAllowReadOnlyOperations: false

Most recently executed operations:

0: [1970-01-02 00:00:04.209] prepare took 0ms - succeeded, sql="PRAGMA user\_version;"

1: [1970-01-02 00:00:04.208] executeForString took 0ms - succeeded, sql="SELECT locale FROM android\_metadata UNION SELECT NULL ORDER BY locale DESC LIMIT 1"

2: [1970-01-02 00:00:04.208] execute took 0ms - succeeded, sql="CREATE TABLE IF NOT EXISTS android\_metadata (locale TEXT)"

3: [1970-01-02 00:00:04.208] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint=100"

4: [1970-01-02 00:00:04.207] executeForLong took 1ms - succeeded, sql="PRAGMA wal\_autocheckpoint"

5: [1970-01-02 00:00:04.207] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit=524288"

6: [1970-01-02 00:00:04.207] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit"

7: [1970-01-02 00:00:04.207] executeForString took 0ms - succeeded, sql="PRAGMA synchronous"

8: [1970-01-02 00:00:04.201] executeForString took 6ms - succeeded, sql="PRAGMA journal\_mode"

9: [1970-01-02 00:00:04.201] executeForLong took 0ms - succeeded, sql="PRAGMA foreign\_keys"

10: [1970-01-02 00:00:04.201] executeForLong took 0ms - succeeded, sql="PRAGMA page\_size"

Available non-primary connections:

Connection #1:

isPrimaryConnection: false

onlyAllowReadOnlyOperations: true

Most recently executed operations:

0: [1970-01-02 00:00:05.843] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lock\_pattern\_autolock"]

1: [1970-01-02 00:00:05.838] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

2: [1970-01-02 00:00:05.838] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lockscreen.password\_type\_alternate"]

3: [1970-01-02 00:00:05.837] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

4: [1970-01-02 00:00:05.837] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lockscreen.password\_type"]

5: [1970-01-02 00:00:05.836] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

6: [1970-01-02 00:00:04.230] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lockscreen.password\_type"]

7: [1970-01-02 00:00:04.229] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

8: [1970-01-02 00:00:04.228] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lock\_pattern\_autolock"]

9: [1970-01-02 00:00:04.227] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

10: [1970-01-02 00:00:04.227] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lockscreen.password\_type\_alternate"]

11: [1970-01-02 00:00:04.227] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

12: [1970-01-02 00:00:04.226] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lockscreen.password\_type"]

13: [1970-01-02 00:00:04.226] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

14: [1970-01-02 00:00:04.223] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lock\_pattern\_autolock"]

15: [1970-01-02 00:00:04.222] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

16: [1970-01-02 00:00:04.222] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lockscreen.password\_type\_alternate"]

17: [1970-01-02 00:00:04.221] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

18: [1970-01-02 00:00:04.221] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?", bindArgs=["0", "lockscreen.password\_type"]

19: [1970-01-02 00:00:04.220] prepare took 0ms - succeeded, sql="SELECT value FROM locksettings WHERE user=? AND name=?"

Acquired connections:

<none>

Connection waiters:

<none>

Connection pool for /data/data/com.android.providers.settings/databases/settings.db:

Open: true

Max connections: 4

Available primary connection:

Connection #0:

isPrimaryConnection: true

onlyAllowReadOnlyOperations: false

Most recently executed operations:

0: [2014-03-03 01:03:03.255] executeForLastInsertedRowId took 161ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)", bindArgs=["com.andrew.apollo/com.andrew.apollo.MediaButtonIntentReceiver", "media\_button\_receiver"]

1: [2014-03-03 01:03:03.255] prepare took 0ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)"

2: [2014-03-01 07:21:20.182] executeForLastInsertedRowId took 11ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)", bindArgs=["1", "com.baidu.pushservice.PushSettings.connect\_state"]

3: [2014-03-01 07:21:20.182] prepare took 0ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)"

4: [2014-03-01 07:21:20.019] executeForLastInsertedRowId took 100ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)", bindArgs=["0", "com.baidu.pushservice.PushSettings.connect\_state"]

5: [2014-03-01 07:21:20.019] prepare took 0ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)"

6: [2014-02-28 10:18:01.156] executeForLastInsertedRowId took 12ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)", bindArgs=["1", "com.baidu.pushservice.PushSettings.connect\_state"]

7: [2014-02-28 10:18:01.156] prepare took 0ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)"

8: [1970-01-02 00:00:11.493] executeForLastInsertedRowId took 40ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)", bindArgs=["0", "com.baidu.pushservice.PushSettings.connect\_state"]

9: [1970-01-02 00:00:11.493] prepare took 0ms - succeeded, sql="INSERT INTO system(value,name) VALUES (?,?)"

10: [1970-01-02 00:00:03.007] prepare took 0ms - succeeded, sql="PRAGMA user\_version;"

11: [1970-01-02 00:00:03.006] executeForString took 0ms - succeeded, sql="SELECT locale FROM android\_metadata UNION SELECT NULL ORDER BY locale DESC LIMIT 1"

12: [1970-01-02 00:00:03.006] execute took 0ms - succeeded, sql="CREATE TABLE IF NOT EXISTS android\_metadata (locale TEXT)"

13: [1970-01-02 00:00:03.000] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint=100"

14: [1970-01-02 00:00:03.000] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint"

15: [1970-01-02 00:00:03.000] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit=524288"

16: [1970-01-02 00:00:03.000] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit"

17: [1970-01-02 00:00:03.000] executeForString took 0ms - succeeded, sql="PRAGMA synchronous"

18: [1970-01-02 00:00:02.980] executeForString took 19ms - succeeded, sql="PRAGMA journal\_mode"

19: [1970-01-02 00:00:02.979] executeForLong took 1ms - succeeded, sql="PRAGMA foreign\_keys"

Available non-primary connections:

Connection #1:

isPrimaryConnection: false

onlyAllowReadOnlyOperations: true

Most recently executed operations:

0: [1970-01-02 00:00:03.104] executeForCursorWindow took 1ms - succeeded, sql="SELECT shortcut, intent FROM bookmarks"

1: [1970-01-02 00:00:03.072] executeForCursorWindow took 32ms - succeeded, sql="SELECT shortcut, intent FROM bookmarks"

2: [1970-01-02 00:00:03.070] prepare took 0ms - succeeded, sql="SELECT shortcut, intent FROM bookmarks"

3: [1970-01-02 00:00:03.018] executeForCursorWindow took 1ms - succeeded, sql="SELECT name, value FROM system LIMIT 201"

4: [1970-01-02 00:00:03.018] prepare took 0ms - succeeded, sql="SELECT name, value FROM system LIMIT 201"

5: [1970-01-02 00:00:03.016] executeForCursorWindow took 1ms - succeeded, sql="SELECT name, value FROM secure LIMIT 201"

6: [1970-01-02 00:00:03.016] prepare took 0ms - succeeded, sql="SELECT name, value FROM secure LIMIT 201"

7: [1970-01-02 00:00:03.015] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM secure WHERE name=?", bindArgs=["long\_press\_timeout"]

8: [1970-01-02 00:00:03.014] prepare took 0ms - succeeded, sql="SELECT value FROM secure WHERE name=?"

9: [1970-01-02 00:00:03.014] executeForCursorWindow took 0ms - succeeded, sql="SELECT name, value FROM global LIMIT 201"

10: [1970-01-02 00:00:03.013] prepare took 0ms - succeeded, sql="SELECT name, value FROM global LIMIT 201"

11: [1970-01-02 00:00:03.012] executeForCursorWindow took 0ms - succeeded, sql="SELECT value FROM secure WHERE (name=?)", bindArgs=["android\_id"]

12: [1970-01-02 00:00:03.011] prepare took 0ms - succeeded, sql="SELECT value FROM secure WHERE (name=?)"

13: [1970-01-02 00:00:03.010] executeForLong took 0ms - succeeded, sql="PRAGMA user\_version;"

14: [1970-01-02 00:00:03.010] executeForString took 0ms - succeeded, sql="SELECT locale FROM android\_metadata UNION SELECT NULL ORDER BY locale DESC LIMIT 1"

15: [1970-01-02 00:00:03.009] execute took 1ms - succeeded, sql="CREATE TABLE IF NOT EXISTS android\_metadata (locale TEXT)"

16: [1970-01-02 00:00:03.009] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint=100"

17: [1970-01-02 00:00:03.009] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint"

18: [1970-01-02 00:00:03.009] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit=524288"

19: [1970-01-02 00:00:03.009] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit"

Acquired connections:

<none>

Connection waiters:

<none>

\*\* Database info for pid 2584 [com.cyanogenmod.lockclock] \*\*

\*\* Database info for pid 2300 [com.cyanogenmod.trebuchet] \*\*

Connection pool for /data/user/0/com.cyanogenmod.trebuchet/databases/launcher.db:

Open: true

Max connections: 1

Available primary connection:

Connection #0:

isPrimaryConnection: true

onlyAllowReadOnlyOperations: true

Most recently executed operations:

0: [2014-03-03 01:41:33.859] executeForCursorWindow took 1ms - succeeded, sql="SELECT \* FROM favorites"

1: [2014-03-03 01:41:33.856] prepare took 0ms - succeeded, sql="SELECT \* FROM favorites"

2: [2014-02-28 10:22:38.190] executeForLastInsertedRowId took 313ms - succeeded, sql="INSERT INTO favorites(cellX,icon,cellY,screen,iconPackage,itemType,intent,title,iconResource,\_id,spanX,container,spanY,iconType) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?)", bindArgs=[5, <byte[]>, 0, 2, "com.youku.tv", 1, "#Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.youku.tv/.WelcomeActivity;end", "优酷", "com.youku.tv:drawable/icon", 17, 1, -100, 1, 0]

3: [2014-02-28 10:22:38.189] prepare took 1ms - succeeded, sql="INSERT INTO favorites(cellX,icon,cellY,screen,iconPackage,itemType,intent,title,iconResource,\_id,spanX,container,spanY,iconType) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?)"

4: [2014-02-28 10:22:38.124] executeForCursorWindow took 3ms - succeeded, sql="SELECT intent FROM favorites WHERE (intent=?)", bindArgs=["#Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;component=com.youku.tv/.WelcomeActivity;end"]

5: [2014-02-28 10:22:38.122] prepare took 0ms - succeeded, sql="SELECT intent FROM favorites WHERE (intent=?)"

6: [2014-02-28 10:22:38.109] executeForCursorWindow took 10ms - succeeded, sql="SELECT itemType, container, screen, cellX, cellY, spanX, spanY FROM favorites"

7: [2014-02-28 10:22:38.106] prepare took 1ms - succeeded, sql="SELECT itemType, container, screen, cellX, cellY, spanX, spanY FROM favorites"

8: [1970-01-02 00:00:06.600] executeForCursorWindow took 19ms - succeeded, sql="SELECT \* FROM favorites"

9: [1970-01-02 00:00:06.598] prepare took 0ms - succeeded, sql="SELECT \* FROM favorites"

10: [1970-01-02 00:00:05.846] executeForCursorWindow took 10ms - succeeded, sql="SELECT MAX(\_id) FROM favorites"

11: [1970-01-02 00:00:05.845] prepare took 1ms - succeeded, sql="SELECT MAX(\_id) FROM favorites"

12: [1970-01-02 00:00:05.817] executeForLong took 28ms - succeeded, sql="PRAGMA user\_version;"

13: [1970-01-02 00:00:05.816] prepare took 1ms - succeeded, sql="PRAGMA user\_version;"

14: [1970-01-02 00:00:05.816] executeForString took 0ms - succeeded, sql="SELECT locale FROM android\_metadata UNION SELECT NULL ORDER BY locale DESC LIMIT 1"

15: [1970-01-02 00:00:05.815] execute took 0ms - succeeded, sql="CREATE TABLE IF NOT EXISTS android\_metadata (locale TEXT)"

16: [1970-01-02 00:00:05.809] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint=100"

17: [1970-01-02 00:00:05.809] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint"

18: [1970-01-02 00:00:05.809] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit=524288"

19: [1970-01-02 00:00:05.808] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit"

Available non-primary connections:

<none>

Acquired connections:

<none>

Connection waiters:

<none>

\*\* Database info for pid 12945 [com.youku.tv:multiscreen] \*\*

\*\* Database info for pid 12912 [com.hiapk.markettv] \*\*

Connection pool for /data/data/com.hiapk.markettv/databases/hiapk\_apk.db:

Open: true

Max connections: 1

Available primary connection:

Connection #0:

isPrimaryConnection: true

onlyAllowReadOnlyOperations: true

Most recently executed operations:

0: [2014-03-03 01:03:18.340] executeForCursorWindow took 0ms - succeeded, sql="SELECT a.\*, b.source, b.recd\_id FROM app\_download a LEFT JOIN app\_download\_extend b ON (a.app\_id=b.app\_id)"

1: [2014-03-03 01:03:18.339] prepare took 0ms - succeeded, sql="SELECT a.\*, b.source, b.recd\_id FROM app\_download a LEFT JOIN app\_download\_extend b ON (a.app\_id=b.app\_id)"

2: [2014-03-03 01:03:18.327] executeForLong took 0ms - succeeded, sql="PRAGMA user\_version;"

3: [2014-03-03 01:03:18.327] prepare took 0ms - succeeded, sql="PRAGMA user\_version;"

4: [2014-03-03 01:03:18.326] executeForString took 1ms - succeeded, sql="SELECT locale FROM android\_metadata UNION SELECT NULL ORDER BY locale DESC LIMIT 1"

5: [2014-03-03 01:03:18.326] execute took 0ms - succeeded, sql="CREATE TABLE IF NOT EXISTS android\_metadata (locale TEXT)"

6: [2014-03-03 01:03:18.325] executeForLong took 1ms - succeeded, sql="PRAGMA wal\_autocheckpoint=100"

7: [2014-03-03 01:03:18.325] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint"

8: [2014-03-03 01:03:18.325] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit=524288"

9: [2014-03-03 01:03:18.325] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit"

10: [2014-03-03 01:03:18.325] executeForString took 0ms - succeeded, sql="PRAGMA synchronous"

11: [2014-03-03 01:03:18.325] executeForString took 0ms - succeeded, sql="PRAGMA journal\_mode=PERSIST"

12: [2014-03-03 01:03:18.324] executeForString took 1ms - succeeded, sql="PRAGMA journal\_mode"

13: [2014-03-03 01:03:18.324] executeForLong took 0ms - succeeded, sql="PRAGMA foreign\_keys"

14: [2014-03-03 01:03:18.324] executeForLong took 0ms - succeeded, sql="PRAGMA page\_size"

Available non-primary connections:

<none>

Acquired connections:

<none>

Connection waiters:

<none>

Connection pool for /data/data/com.hiapk.markettv/databases/hiapk\_market.db:

Open: true

Max connections: 1

Available primary connection:

Connection #0:

isPrimaryConnection: true

onlyAllowReadOnlyOperations: true

Most recently executed operations:

0: [2014-03-03 01:03:18.466] executeForCursorWindow took 1ms - succeeded, sql="SELECT \* FROM software\_installed"

1: [2014-03-03 01:03:18.466] prepare took 0ms - succeeded, sql="SELECT \* FROM software\_installed"

2: [2014-03-03 01:03:18.312] executeForLong took 1ms - succeeded, sql="PRAGMA user\_version;"

3: [2014-03-03 01:03:18.312] prepare took 0ms - succeeded, sql="PRAGMA user\_version;"

4: [2014-03-03 01:03:18.311] executeForString took 1ms - succeeded, sql="SELECT locale FROM android\_metadata UNION SELECT NULL ORDER BY locale DESC LIMIT 1"

5: [2014-03-03 01:03:18.310] execute took 1ms - succeeded, sql="CREATE TABLE IF NOT EXISTS android\_metadata (locale TEXT)"

6: [2014-03-03 01:03:18.309] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint=100"

7: [2014-03-03 01:03:18.309] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint"

8: [2014-03-03 01:03:18.309] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit=524288"

9: [2014-03-03 01:03:18.309] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit"

10: [2014-03-03 01:03:18.309] executeForString took 0ms - succeeded, sql="PRAGMA synchronous"

11: [2014-03-03 01:03:18.308] executeForString took 1ms - succeeded, sql="PRAGMA journal\_mode=PERSIST"

12: [2014-03-03 01:03:18.307] executeForString took 1ms - succeeded, sql="PRAGMA journal\_mode"

13: [2014-03-03 01:03:18.307] executeForLong took 0ms - succeeded, sql="PRAGMA foreign\_keys"

14: [2014-03-03 01:03:18.307] executeForLong took 0ms - succeeded, sql="PRAGMA page\_size"

Available non-primary connections:

<none>

Acquired connections:

<none>

Connection waiters:

<none>

\*\* Database info for pid 12853 [com.qiyi.video] \*\*

\*\* Database info for pid 2683 [com.qiyi.video:bdservice\_v1] \*\*

Connection pool for /data/data/com.qiyi.video/databases/moplus\_server\_config.db:

Open: true

Max connections: 1

Available primary connection:

Connection #0:

isPrimaryConnection: true

onlyAllowReadOnlyOperations: true

Most recently executed operations:

0: [2014-02-28 10:18:29.824] executeForCursorWindow took 0ms - succeeded, sql="select \* from server\_config\_table where type=0"

1: [2014-02-28 10:18:29.823] prepare took 1ms - succeeded, sql="select \* from server\_config\_table where type=0"

2: [1970-01-02 00:00:10.971] executeForCursorWindow took 1ms - succeeded, sql="select \* from server\_config\_table where type=1"

3: [1970-01-02 00:00:10.970] prepare took 1ms - succeeded, sql="select \* from server\_config\_table where type=1"

4: [1970-01-02 00:00:10.912] execute took 58ms - succeeded, sql="COMMIT;"

5: [1970-01-02 00:00:10.912] executeForChangedRowCount took 0ms - succeeded, sql="PRAGMA user\_version = 1"

6: [1970-01-02 00:00:10.912] prepare took 0ms - succeeded, sql="PRAGMA user\_version = 1"

7: [1970-01-02 00:00:10.911] executeForChangedRowCount took 1ms - succeeded, sql="CREATE TABLE server\_config\_table (\_id INTEGER PRIMARY KEY,name TEXT,type INTEGER,value TEXT);"

8: [1970-01-02 00:00:10.911] prepare took 0ms - succeeded, sql="CREATE TABLE server\_config\_table (\_id INTEGER PRIMARY KEY,name TEXT,type INTEGER,value TEXT);"

9: [1970-01-02 00:00:10.910] execute took 0ms - succeeded, sql="BEGIN EXCLUSIVE;"

10: [1970-01-02 00:00:10.910] executeForLong took 0ms - succeeded, sql="PRAGMA user\_version;"

11: [1970-01-02 00:00:10.909] prepare took 1ms - succeeded, sql="PRAGMA user\_version;"

12: [1970-01-02 00:00:10.753] execute took 156ms - succeeded, sql="COMMIT"

13: [1970-01-02 00:00:10.753] execute took 0ms - succeeded, sql="REINDEX LOCALIZED"

14: [1970-01-02 00:00:10.752] execute took 1ms - succeeded, sql="INSERT INTO android\_metadata (locale) VALUES(?)", bindArgs=["zh\_CN"]

15: [1970-01-02 00:00:10.752] execute took 0ms - succeeded, sql="DELETE FROM android\_metadata"

16: [1970-01-02 00:00:10.751] execute took 1ms - succeeded, sql="BEGIN"

17: [1970-01-02 00:00:10.751] executeForString took 0ms - succeeded, sql="SELECT locale FROM android\_metadata UNION SELECT NULL ORDER BY locale DESC LIMIT 1"

18: [1970-01-02 00:00:10.599] execute took 152ms - succeeded, sql="CREATE TABLE IF NOT EXISTS android\_metadata (locale TEXT)"

19: [1970-01-02 00:00:10.598] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint=100"

Available non-primary connections:

<none>

Acquired connections:

<none>

Connection waiters:

<none>

\*\* Database info for pid 2357 [com.android.smspush] \*\*

\*\* Database info for pid 2240 [com.android.phone] \*\*

Connection pool for /data/data/com.android.providers.telephony/databases/telephony.db:

Open: true

Max connections: 1

Available primary connection:

Connection #0:

isPrimaryConnection: true

onlyAllowReadOnlyOperations: true

Most recently executed operations:

0: [1970-01-02 00:00:11.709] executeForCursorWindow took 0ms - succeeded, sql="SELECT apn FROM carriers WHERE (\_id = -1) ORDER BY name ASC"

1: [1970-01-02 00:00:11.707] prepare took 0ms - succeeded, sql="SELECT apn FROM carriers WHERE (\_id = -1) ORDER BY name ASC"

2: [1970-01-02 00:00:11.706] executeForLong took 1ms - succeeded, sql="PRAGMA user\_version;"

3: [1970-01-02 00:00:11.706] prepare took 0ms - succeeded, sql="PRAGMA user\_version;"

4: [1970-01-02 00:00:11.701] executeForString took 1ms - succeeded, sql="SELECT locale FROM android\_metadata UNION SELECT NULL ORDER BY locale DESC LIMIT 1"

5: [1970-01-02 00:00:11.701] execute took 0ms - succeeded, sql="CREATE TABLE IF NOT EXISTS android\_metadata (locale TEXT)"

6: [1970-01-02 00:00:11.700] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint=100"

7: [1970-01-02 00:00:11.700] executeForLong took 0ms - succeeded, sql="PRAGMA wal\_autocheckpoint"

8: [1970-01-02 00:00:11.688] executeForLong took 12ms - succeeded, sql="PRAGMA journal\_size\_limit=524288"

9: [1970-01-02 00:00:11.688] executeForLong took 0ms - succeeded, sql="PRAGMA journal\_size\_limit"

10: [1970-01-02 00:00:11.688] executeForString took 0ms - succeeded, sql="PRAGMA synchronous"

11: [1970-01-02 00:00:11.687] executeForString took 0ms - succeeded, sql="PRAGMA journal\_mode=PERSIST"

12: [1970-01-02 00:00:11.682] executeForString took 5ms - succeeded, sql="PRAGMA journal\_mode"

13: [1970-01-02 00:00:11.681] executeForLong took 0ms - succeeded, sql="PRAGMA foreign\_keys"

14: [1970-01-02 00:00:11.681] executeForLong took 0ms - succeeded, sql="PRAGMA page\_size"

Available non-primary connections:

<none>

Acquired connections:

<none>

Connection waiters:

<none>

\*\* Database info for pid 2280 [com.coship.krdservice] \*\*

\*\* Database info for pid 2222 [com.hisilicon.dlna.dms] \*\*

\*\* Database info for pid 2272 [com.bel.android.dspmanager] \*\*

## dumpsys diskstats

系统内部存储设备的使用情况。

diskstats服务注册位置：

(源码路径：.\framework\base\services\java\com\android\server\SystemServer.java)



dump实现位置:

(源码路径：.\framework\base\services\java\com\android\server\DiskStatsService.java)



dumpsys diskstats用法：

Latency: 0ms [512B Data Write]

Data-Free: 894012K / 1032088K total = 86% free

Cache-Free: 31512K / 100784K total = 31% free

System-Free: 234744K / 503944K total = 46% free

图 8 dumpsys diskstats示意图

## dumpsys devicestoragemonitor

Current DeviceStorageMonitor state:

mFreeMem=894 MB mTotalMemory=0.98 GB

mFreeMemAfterLastCacheClear=894 MB

mLastReportedFreeMem=895 MB mLastReportedFreeMemTime=-3h41m36s839ms

mLowMemFlag=false mMemFullFlag=false

mClearSucceeded=false mClearingCache=false

mMemLowThreshold=101 MB mMemFullThreshold=1.00 MB

mMemCacheStartTrimThreshold=75.84 MB mMemCacheTrimToThreshold=151 MB

## dumpsys package

查看一些系统信息和所有应用的信息。

不建议使用dumpsys package去打印所有信息，可以通过添加参数去获得需要的信息。

如果一定要dumpsys package,则可到盒子的可写分区/mnt，重定向到文本里。

dumpsys package > package.txt

后续的命令都可以重定向到文本，这样便于查看分析问题。

### dumpsys package –h(help)

Package manager dump options:

[-h] [-f] [cmd] ...

-f: print details of intent filters

-h: print this help

cmd may be one of:

l[ibraries]: list known shared libraries

f[ibraries]: list device features

r[esolvers]: dump intent resolvers

perm[issions]: dump permissions

pref[erred]: print preferred package settings

preferred-xml: print preferred package settings as xml

prov[iders]: dump content providers

p[ackages]: dump installed packages

s[hared-users]: dump shared user IDs

m[essages]: print collected runtime messages

v[erifiers]: print package verifier info

<package.name>: info about given package

### dumpsys package l

作用：列出所有共享库。

Libraries:

android.test.runner -> /system/framework/android.test.runner.jar

com.android.location.provider -> /system/framework/com.android.location.provider.jar

javax.obex -> /system/framework/javax.obex.jar

com.android.future.usb.accessory -> /system/framework/com.android.future.usb.accessory.jar

### dumpsys package f

作用：列出设备特征，这个主要判断设备是否支持某个应用。

Android Market会根据uses-feature过滤所有你设备不支持的应用。通过使用<uses-feature>元素，一个应用可以指定它所支持的硬件型号。

例如：

当你的游戏需要一个支持多点触控的屏幕的时候，我们可以使用 <uses-feature>元素来剔除所有不支持多点触控的设备，就像下面这样：  
<uses-feature android:name="android.hardware.touchscreen.multitouch" android:required="true"/>

通过dumpsys package f命令查看设备特征，即可确定设备是否支持某些应用。

Features:

android.hardware.wifi

android.hardware.location.network

android.hardware.location

android.hardware.sensor.gyroscope

android.hardware.screen.landscape

android.hardware.screen.portrait

com.cyanogenmod.android

android.hardware.wifi.direct

android.hardware.usb.accessory

android.hardware.camera.any

android.hardware.bluetooth

android.hardware.touchscreen.multitouch.distinct

### dumpsys package p

可以查看某个应用程序的版本号（versionName）、安装时间（firstInstallTime）、升级时间（lastUpdateTime）、安装路径（codePath）、共享用户（sharedUser）、权限（grantedPermissions）

Package [com.android.email] (41106fe0):

userId=10019 gids=[3003, 1015, 1028]

sharedUser=null

pkg=Package{411df440 com.android.email}

codePath=/system/app/Email2.apk

resourcePath=/system/app/Email2.apk

nativeLibraryPath=/data/app-lib/Email2

versionCode=410000

applicationInfo=ApplicationInfo{411df4e0 com.android.email}

flags=[ SYSTEM HAS\_CODE ALLOW\_CLEAR\_USER\_DATA ALLOW\_BACKUP ]

versionName=4.1

dataDir=/data/data/com.android.email

targetSdk=17

supportsScreens=[small, medium, large, xlarge, resizeable, anyDensity]

timeStamp=2014-02-25 13:55:03

firstInstallTime=2014-02-25 13:55:03

lastUpdateTime=2014-02-25 13:55:03

signatures=PackageSignatures{411070a8 [410e7c80]}

permissionsFixed=false haveGids=true installStatus=1

pkgFlags=[ SYSTEM HAS\_CODE ALLOW\_CLEAR\_USER\_DATA ALLOW\_BACKUP ]

User 0: installed=true stopped=false notLaunched=false enabled=0

disabledComponents:

com.android.email.activity.MessageCompose

com.android.email.activity.ShortcutPicker

com.android.email.service.AttachmentDownloadService

com.android.email.service.MailService

enabledComponents:

com.android.email.widget.WidgetConfiguration

grantedPermissions:

android.permission.READ\_EXTERNAL\_STORAGE

android.permission.ACCESS\_NETWORK\_STATE

### dumpsys package s

查看共享用户的应用程序。每个应用程序以SharedUser[包名]分开，可以互相访问数据。

参考：<http://blog.csdn.net/a345017062/article/details/6236263>

SharedUser [android.uid.bluetooth] (41011880):

userId=1002 gids=[3003, 3002, 3001, 1015, 3005, 1016, 3008, 1028]

grantedPermissions:

com.android.email.permission.ACCESS\_PROVIDER

android.permission.READ\_EXTERNAL\_STORAGE

android.permission.WRITE\_EXTERNAL\_STORAGE

SharedUser [android.uid.calendar] (40fc6210):

userId=10011 gids=[3003, 1015, 1028]

grantedPermissions:

android.permission.READ\_EXTERNAL\_STORAGE

android.permission.USE\_CREDENTIALS

android.permission.WRITE\_EXTERNAL\_STORAGE

android.permission.GET\_ACCOUNTS

### dumpsys package $package-name

查看某个应用程序的安装路径、权限、开机启动（Non-Data Actions的android.intent.action.BOOT\_COMPLETED表示开机启动）等信息。

Receiver Resolver Table:

Schemes:

file:

411d6f38 com.android.providers.downloads/.DownloadReceiver filter 411d73f8

Non-Data Actions:

android.intent.action.BOOT\_COMPLETED:

411d6f38 com.android.providers.downloads/.DownloadReceiver filter 411d71c8

android.net.conn.CONNECTIVITY\_CHANGE:

411d6f38 com.android.providers.downloads/.DownloadReceiver filter 411d71c8

## dumpsys meminfo

### dumpsys meminfo –h(help)

meminfo dump options: [-a] [--oom] [process]

-a: include all available information for each process.

--oom: only show processes organized by oom adj.

If [process] is specified it can be the name or

pid of a specific process to dump.

### dumpsys meminfo

显示各个进程所占内存信息。

* Total PSS by process ：各个进程占用的物理内存。
* Total PSS by OOM adjustment ：OOM 杀进程后，可以要回来的内存。
* Total PSS by category ：按memory使用类型统计内存使用情况。

Applications Memory Usage (kB):

Uptime: 34986361 Realtime: 34986361

Total PSS by process:

32849 kB: system (pid 2214)

26765 kB: com.android.settings (pid 3372)

23280 kB: com.android.browser (pid 3643)

Total PSS by OOM adjustment:

28935 kB: Persistent

14770 kB: com.android.systemui (pid 2326)

7231 kB: com.android.phone (pid 2469)

3611 kB: com.bel.android.dspmanager (pid 2483)

Total PSS by category:

91823 kB: Dalvik

8184 kB: .apk mmap

Total PSS: 199528 kB

KSM: 0 kB saved from shared 0 kB

### dumpsys meminfo $package\_name or $pi//使用程序的包名或者进程id

dumpsys meminfo pkname

一般来说内存占用大小有如下规律：VSS >= RSS >= PSS >= USS  
VSS - Virtual Set Size 虚拟耗用内存（包含共享库占用的内存）是单个进程全部可访问的地址空间  
RSS - Resident Set Size 实际使用物理内存（包含共享库占用的内存）是单个进程实际占用的内存大小，对于单个共享库， 尽管无论多少个进程使用，实际该共享库只会被装入内存一次。  
PSS - Proportional Set Size 实际使用的物理内存（比例分配共享库占用的内存）  
USS - Unique Set Size 进程独自占用的物理内存（不包含共享库占用的内存）USS 是一个非常非常有用的数字， 因为它揭示了运行一个特定进程的真实的内存增量大小。如果进程被终止， USS 就是实际被返还给系统的内存大小。  
**USS** 是针对某个进程开始有可疑内存泄露的情况，进行检测的最佳数字。怀疑某个程序有内存泄露可以查看这个值是否一直有增加

android程序内存被分为2部分：native和dalvik，dalvik就是我们平常说的java堆，我们创建的对象是在这里面分配的，而bitmap是直接在native上分配的

对于内存的限制是 native+dalvik 不能超过最大限制。

可以通过查看cat /system/ build.prop里面的dalvik.vm.heapsize=512m字段确定应用程序默认内存大小。

Applications Memory Usage (kB):

Uptime: 35562088 Realtime: 35562088

\*\* MEMINFO in pid 5749 [com.tencent.mobileqq] \*\*

Shared Private Heap Heap Heap

Pss Dirty Dirty Size Alloc Free

------ ------ ------ ------ ------ ------

Native 20 8 20 4676 3064 15

Dalvik 11955 5200 11628 19056 10892 8164

Cursor 0 0 0

Ashmem 0 0 0

Other dev 4 20 0

.so mmap 1739 2168 1276

.jar mmap 4 0 0

.apk mmap 539 0 0

.ttf mmap 121 0 0

.dex mmap 6031 0 96

Other mmap 194 16 144

Unknown 1368 376 1348

TOTAL 21975 7788 14512 23732 13956 8179

Objects

Views: 41 ViewRootImpl: 2

AppContexts: 6 Activities: 2

Assets: 2 AssetManagers: 2

Local Binders: 21 Proxy Binders: 26

Death Recipients: 1

OpenSSL Sockets: 0

SQL

MEMORY\_USED: 0

PAGECACHE\_OVERFLOW: 0 MALLOC\_SIZE: 62

当Heap Size:Native+Dalvik超过512m,则会出现OOM。

meminfo服务注册位置：

代码路径./frameworks/base/services/java/com/android/server/am/ActivityManagerService.java）



meminfo对应的dump函数实现封装在MemBinder(m)里面

## dumpsys mount

用于查看盒子开机启动后执行移动硬盘或U盘挂载和卸载流程和次数。

……………….

…………………

mConnection:

13:00:04 - SND -> {1 volume list}

13:00:04 - RCV <- {110 1 block /mnt/sdcard 1 UUID= DevType=SDCARD 66319 (null)}

13:00:04 - RCV <- {200 1 Volumes listed.}

13:00:04 - RMV <- {110 1 block /mnt/sdcard 1 UUID= DevType=SDCARD 66319 (null)}

13:00:04 - RMV <- {200 1 Volumes listed.}

13:00:05 - RCV <- {605 Volume block /mnt/sdcard state changed from 1 (Idle-Unmounted) to 3 (Checking)}

13:00:05 - Error handling '605 Volume block /mnt/sdcard state changed from 1 (Idle-Unmounted) to 3 (Checking)': java.lang.NullPointerException

13:00:05 - RCV <- {605 Volume block /mnt/sdcard state changed from 3 (Checking) to 4 (Mounted) UUID= DevType=SDCARD 66319 (null)}

图 4第一次开机启动dumpsys mount状态示意图

……………..

……………

mConnection:

13:00:04 - SND -> {1 volume list}

13:00:04 - RCV <- {110 1 block /mnt/sdcard 1 UUID= DevType=SDCARD 66319 (null)}

13:00:04 - RCV <- {200 1 Volumes listed.}

13:00:04 - RMV <- {110 1 block /mnt/sdcard 1 UUID= DevType=SDCARD 66319 (null)}

13:00:04 - RMV <- {200 1 Volumes listed.}

13:00:05 - RCV <- {605 Volume block /mnt/sdcard state changed from 1 (Idle-Unmounted) to 3 (Checking)}

13:00:05 - Error handling '605 Volume block /mnt/sdcard state changed from 1 (Idle-Unmounted) to 3 (Checking)': java.lang.NullPointerException

13:00:05 - RCV <- {605 Volume block /mnt/sdcard state changed from 3 (Checking) to 4 (Mounted) UUID= DevType=SDCARD 66319 (null)}

13:04:19 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from -1 (Initializing) to 0 (No-Media)}

13:04:19 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from 0 (No-Media) to 1 (Idle-Unmounted)}

13:04:21 - RCV <- {630 Volume sda1 /mnt/sda/sda1 disk inserted (8:1)}

13:04:21 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from 1 (Idle-Unmounted) to 3 (Checking)}

13:04:21 - Error handling '605 Volume sda1 /mnt/sda/sda1 state changed from 1 (Idle-Unmounted) to 3 (Checking)': java.lang.NullPointerException

13:04:21 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from 3 (Checking) to 4 (Mounted) UUID= DevType=USB 2049 (null)}

图 5插上U盘后dumpsys mount示意图

|  |  |
| --- | --- |
| SDcard常用状态 |  |
| Environment.MEDIA\_MOUNTED | sd卡在手机上正常使用状态 |
| Environment.MEDIA\_UNMOUNTED | 用户手工到手机设置中卸载sd卡之后的状态 |
| Environment.MEDIA\_REMOVED | 用户手动卸载，然后将sd卡从手机取出之后的状态 |
| Environment.MEDIA\_BAD\_REMOVAL | 用户未到手机设置中手动卸载sd卡，直接拨出之后的状态 |
| Environment.MEDIA\_SHARED | 手机直接连接到电脑作为u盘使用之后的状态 |
| Environment.MEDIA\_CHECKINGS | 手机正在扫描sd卡过程中的状态 |
| Environment.MEDIA\_UNMOUNTABLE | SDCard 不可被安装 如果 SDCard 是存在但不可以被安装 |
| Environment.MEDIA\_NOFS | 表明对象为空白或正在使用不受支持的文件系统 |

表格 1 SDcard常用状态

在图5中红色部分表示U盘挂载上去的状态变化过程。

………………

………………

mConnection:

13:00:04 - SND -> {1 volume list}

13:00:04 - RCV <- {110 1 block /mnt/sdcard 1 UUID= DevType=SDCARD 66319 (null)}

13:00:04 - RCV <- {200 1 Volumes listed.}

13:00:04 - RMV <- {110 1 block /mnt/sdcard 1 UUID= DevType=SDCARD 66319 (null)}

13:00:04 - RMV <- {200 1 Volumes listed.}

13:00:05 - RCV <- {605 Volume block /mnt/sdcard state changed from 1 (Idle-Unmounted) to 3 (Checking)}

13:00:05 - Error handling '605 Volume block /mnt/sdcard state changed from 1 (Idle-Unmounted) to 3 (Checking)': java.lang.NullPointerException

13:00:05 - RCV <- {605 Volume block /mnt/sdcard state changed from 3 (Checking) to 4 (Mounted) UUID= DevType=SDCARD 66319 (null)}

13:04:19 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from -1 (Initializing) to 0 (No-Media)}

13:04:19 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from 0 (No-Media) to 1 (Idle-Unmounted)}

13:04:21 - RCV <- {630 Volume sda1 /mnt/sda/sda1 disk inserted (8:1)}

13:04:21 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from 1 (Idle-Unmounted) to 3 (Checking)}

13:04:21 - Error handling '605 Volume sda1 /mnt/sda/sda1 state changed from 1 (Idle-Unmounted) to 3 (Checking)': java.lang.NullPointerException

13:04:21 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from 3 (Checking) to 4 (Mounted) UUID= DevType=USB 2049 (null)}

13:10:35 - RCV <- {632 Volume sda1 /mnt/sda/sda1 bad removal (8:1)}

13:10:35 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from 4 (Mounted) to 5 (Unmounting)}

13:10:36 - RCV <- {605 Volume sda1 /mnt/sda/sda1 state changed from 5 (Unmounting) to 1 (Idle-Unmounted)}

图 7拔掉U盘后dumpsys mount示意图

褐色部分表示U盘卸载过程。

通过dumpsys mount 可以查看盒子启动过后，U盘挂载和卸载的状态和次数，直到盒子重启后，记录才会被清除。

## dumpsys display

DISPLAY MANAGER (dumpsys display)

mHeadless=false

mOnlyCode=false

mSafeMode=false

mPendingTraversal=false

mAllDisplayBlankStateFromPowerManager=2

mNextNonDefaultDisplayId=1

mDefaultViewport=DisplayViewport{valid=true, displayId=0, orientation=0, logicalFrame=Rect(0, 0 - 1280, 720), physicalFrame=Rect(0, 0 - 1280, 720), deviceWidth=1280, deviceHeight=720}

mExternalTouchViewport=DisplayViewport{valid=false, displayId=0, orientation=0, logicalFrame=Rect(0, 0 - 0, 0), physicalFrame=Rect(0, 0 - 0, 0), deviceWidth=0, deviceHeight=0}

mSingleDisplayDemoMode=false

Display Adapters: size=2

LocalDisplayAdapter

OverlayDisplayAdapter

mCurrentOverlaySetting=

mOverlays: size=0

Display Devices: size=1

DisplayDeviceInfo{"内置屏幕": 1280 x 720, 60.000004 fps, density 160, 160.15764 x 160.42105 dpi, touch INTERNAL, FLAG\_DEFAULT\_DISPLAY, FLAG\_ROTATES\_WITH\_CONTENT, FLAG\_SECURE, FLAG\_SUPPORTS\_PROTECTED\_BUFFERS, rotation 0, type BUILT\_IN, address null}

mAdapter=LocalDisplayAdapter

mDisplayToken=android.os.BinderProxy@423fd038

mCurrentLayerStack=0

mCurrentOrientation=0

mCurrentLayerStackRect=Rect(0, 0 - 1280, 720)

mCurrentDisplayRect=Rect(0, 0 - 1280, 720)

mCurrentSurface=null

mBuiltInDisplayId=0

mPhys=PhysicalDisplayInfo{1280 x 720, 60.000004 fps, density 1.0, 160.15764 x 160.42105 dpi, secure true}

mBlanked=false

Logical Displays: size=1

Display 0:

mDisplayId=0

mLayerStack=0

mHasContent=true

mPrimaryDisplayDevice=内置屏幕

mBaseDisplayInfo=DisplayInfo{"内置屏幕", app 1280 x 720, real 1280 x 720, largest app 1280 x 720, smallest app 1280 x 720, 60.000004 fps, rotation 0, density 160, 160.15764 x 160.42105 dpi, layerStack 0, type BUILT\_IN, address null, FLAG\_SECURE, FLAG\_SUPPORTS\_PROTECTED\_BUFFERS}

mOverrideDisplayInfo=DisplayInfo{"内置屏幕", app 1280 x 720, real 1280 x 720, largest app 1280 x 1255, smallest app 720 x 695, 60.000004 fps, rotation 0, density 160, 160.15764 x 160.42105 dpi, layerStack 0, type BUILT\_IN, address null, FLAG\_SECURE, FLAG\_SUPPORTS\_PROTECTED\_BUFFERS}

## dumpsys window

Window Manager Service 位于Surface Flinger 之上，将要绘制到机器画面上的内容传递给Surface Flinger

### dumpsys window lastanr

用于查看从开机到现场是否发生ANR事件。

WINDOW MANAGER LAST ANR (dumpsys window lastanr)

<no ANR has occurred since boot>

### dumpsys window policy

WINDOW MANAGER POLICY STATE (dumpsys window policy)

mSafeMode=false mSystemReady=true mSystemBooted=true

mLidState=-1 mLidOpenRotation=-1 mHdmiPlugged=true

mUiMode=1 mDockMode=0 mCarDockRotation=-1 mDeskDockRotation=-1

mUserRotationMode=0 mUserRotation=0 mAllowAllRotations=-1

mCurrentAppOrientation=0

mCarDockEnablesAccelerometer=true mDeskDockEnablesAccelerometer=true

mLidKeyboardAccessibility=0 mLidNavigationAccessibility=0 mLidControlsSleep=true

mLongPressOnPowerBehavior=-1 mHasSoftInput=true

mScreenOnEarly=true mScreenOnFully=true mOrientationSensorEnabled=false

mUnrestrictedScreen=(0,0) 1280x720

mRestrictedScreen=(0,0) 1280x720

mStableFullscreen=(0,0)-(1280,720)

mStable=(0,25)-(1280,720)

mSystem=(0,0)-(1280,720)

mCur=(0,0)-(1280,720)

mContent=(0,0)-(1280,720)

mDock=(0,0)-(1280,720)

mDockLayer=268435456 mStatusBarLayer=151000

mShowingLockscreen=false mShowingDream=false mDreamingLockscreen=false

mStatusBar=Window{42600d28 u0 StatusBar}

mFocusedWindow=Window{42688fc8 u0 com.youku.tv/com.youku.tv.ui.activity.AllViedosActivity}

mFocusedApp=Token{427490e8 ActivityRecord{4277a408 u0 com.youku.tv/.ui.activity.AllViedosActivity}}

mWinDismissingKeyguard=Window{424a1310 u0 com.youku.tv/com.youku.player.activity.YoukuTVPlayerActivity}

mTopFullscreenOpaqueWindowState=Window{42688fc8 u0 com.youku.tv/com.youku.tv.ui.activity.AllViedosActivity}

mTopIsFullscreen=true mHideLockScreen=false

mForceStatusBar=false mForceStatusBarFromKeyguard=false

mDismissKeyguard=0 mWinDismissingKeyguard=Window{424a1310 u0 com.youku.tv/com.youku.player.activity.YoukuTVPlayerActivity} mHomePressed=false

mAllowLockscreenWhenOn=false mLockScreenTimeout=-1 mLockScreenTimerActive=false

mEndcallBehavior=2 mIncallPowerBehavior=1 mRingHomeBehavior=1 mLongPressOnHomeBehavior=2

mLandscapeRotation=0 mSeascapeRotation=2

mPortraitRotation=3 mUpsideDownRotation=1

mHdmiRotation=0 mHdmiRotationLock=true

### dumpsys window animator

WINDOW MANAGER ANIMATOR STATE (dumpsys window animator)

App Animators:

App Animator #3 com.android.server.wm.AppWindowAnimator@427cda60

App Animator #2 com.android.server.wm.AppWindowAnimator@4277fdc0

App Animator #1 com.android.server.wm.AppWindowAnimator@425f78c8

App Animator #0 com.android.server.wm.AppWindowAnimator@42482f28

Wallpaper tokens:

Wallpaper #0 WindowToken{426e4120 android.os.Binder@426f7178}

DisplayContentsAnimator #0:

Window #0: WindowStateAnimator{42745db0 com.android.systemui.ImageWallpaper}

Window #1: WindowStateAnimator{425ba800 com.youku.tv/com.youku.tv.ui.activity.AllViedosActivity}

Window #2: WindowStateAnimator{42704d88 StatusBar}

mBulkUpdateParams=0x8 ORIENTATION\_CHANGE\_COMPLETE

mWallpaperTarget=null

mWpAppAnimator=null

### dumpsys window sessions

WINDOW MANAGER SESSIONS (dumpsys window sessions)

Session Session{4272d218 1974:1000}:

mNumWindow=0 mClientDead=false mSurfaceSession=android.view.SurfaceSession@425eb868

Session Session{425e8270 2092:u0a10046}:

mNumWindow=2 mClientDead=false mSurfaceSession=android.view.SurfaceSession@424bd1a8

Session Session{4246e9f0 3563:u0a10059}:

mNumWindow=2 mClientDead=false mSurfaceSession=android.view.SurfaceSession@42740888

Session Session{423e6e50 2300:u0a10049}:

mNumWindow=1 mClientDead=false [mSurfaceSession=android.view.SurfaceSession@426be340](mailto:mSurfaceSession=android.view.SurfaceSession@426be340)

### dumpsys window tokens

WINDOW MANAGER TOKENS (dumpsys window tokens)

All tokens:

WindowToken{42771590 null}

AppWindowToken{425170f8 token=Token{42515000 ActivityRecord{42514bc8 u0 com.cyanogenmod.trebuchet/.Launcher}}}

WindowToken{426e4120 android.os.Binder@426f7178}

AppWindowToken{42595520 token=Token{423ca098 ActivityRecord{42549870 u0 com.youku.tv/.ui.activity.HomeActivityWithViewPager}}}

AppWindowToken{4277fca8 token=Token{427490e8 ActivityRecord{4277a408 u0 com.youku.tv/.ui.activity.AllViedosActivity}}}

WindowToken{4258ffd8 android.os.Binder@4258ff98}

Wallpaper tokens:

Wallpaper #0 WindowToken{426e4120 android.os.Binder@426f7178}

Application tokens in Z order:

App #2 AppWindowToken{4277fca8 token=Token{427490e8 ActivityRecord{4277a408 u0 com.youku.tv/.ui.activity.AllViedosActivity}}}:

windows=[Window{42688fc8 u0 com.youku.tv/com.youku.tv.ui.activity.AllViedosActivity}]

windowType=2 hidden=false hasVisible=true

app=true userId=0

allAppWindows=[Window{42688fc8 u0 com.youku.tv/com.youku.tv.ui.activity.AllViedosActivity}]

groupId=9 appFullscreen=true requestedOrientation=0

hiddenRequested=false clientHidden=false willBeHidden=false reportedDrawn=true reportedVisible=true

numInterestingWindows=1 numDrawnWindows=1 inPendingTransaction=false allDrawn=true (animator=true)

startingData=null removed=false firstWindowDrawn=true

App #1 AppWindowToken{42595520 token=Token{423ca098 ActivityRecord{42549870 u0 com.youku.tv/.ui.activity.HomeActivityWithViewPager}}}:

windows=[Window{4251d7f8 u0 com.youku.tv/com.youku.tv.ui.activity.HomeActivityWithViewPager}]

windowType=2 hidden=true hasVisible=true

app=true userId=0

allAppWindows=[Window{4251d7f8 u0 com.youku.tv/com.youku.tv.ui.activity.HomeActivityWithViewPager}]

groupId=9 appFullscreen=true requestedOrientation=0

hiddenRequested=true clientHidden=true willBeHidden=false reportedDrawn=false reportedVisible=false

numInterestingWindows=1 numDrawnWindows=1 inPendingTransaction=false allDrawn=true (animator=true)

startingData=null removed=false firstWindowDrawn=true

App #0 AppWindowToken{425170f8 token=Token{42515000 ActivityRecord{42514bc8 u0 com.cyanogenmod.trebuchet/.Launcher}}}:

windows=[Window{423f1628 u0 com.cyanogenmod.trebuchet/com.cyanogenmod.trebuchet.Launcher}]

windowType=2 hidden=true hasVisible=true

app=true userId=0

allAppWindows=[Window{423f1628 u0 com.cyanogenmod.trebuchet/com.cyanogenmod.trebuchet.Launcher}]

groupId=2 appFullscreen=true requestedOrientation=-1

hiddenRequested=true clientHidden=true willBeHidden=false reportedDrawn=false reportedVisible=false

numInterestingWindows=1 numDrawnWindows=1 inPendingTransaction=false allDrawn=true (animator=true)

startingData=null removed=false firstWindowDrawn=true

### dumpsys window windows

WINDOW MANAGER WINDOWS (dumpsys window windows)

Window #0 Window{42600d28 u0 StatusBar}:

mDisplayId=0 mSession=Session{425e8270 2092:u0a10046} mClient=android.os.BinderProxy@427311b0

mOwnerUid=10046 mShowToOwnerOnly=false

mAttrs=WM.LayoutParams{(0,0)(fillx25) gr=#37 sim=#20 ty=2000 fl=#1800048 fmt=-3}

Requested w=1280 h=25 mLayoutSeq=8017

mPolicyVisibility=false mPolicyVisibilityAfterAnim=false mAttachedHidden=false

mHasSurface=true mShownFrame=[0.0,0.0][1280.0,25.0] isReadyForDisplay()=false

WindowStateAnimator{42704d88 StatusBar}:

Surface: shown=false layer=151000 alpha=1.0 rect=(0.0,0.0) 1280.0 x 25.0

Window #1 Window{42688fc8 u0 com.youku.tv/com.youku.tv.ui.activity.AllViedosActivity}:

mDisplayId=0 mSession=Session{4246e9f0 3563:u0a10059} mClient=android.os.BinderProxy@4263d3c0

mOwnerUid=10059 mShowToOwnerOnly=true

mAttrs=WM.LayoutParams{(0,0)(fillxfill) sim=#120 ty=1 fl=#1810580 pfl=0x8 wanim=0x1030001}

Requested w=1280 h=720 mLayoutSeq=8017

mHasSurface=true mShownFrame=[0.0,0.0][1280.0,720.0] isReadyForDisplay()=true

WindowStateAnimator{425ba800 com.youku.tv/com.youku.tv.ui.activity.AllViedosActivity}:

Surface: shown=true layer=21015 alpha=1.0 rect=(0.0,0.0) 1280.0 x 720.0

Window #2 Window{4251d7f8 u0 com.youku.tv/com.youku.tv.ui.activity.HomeActivityWithViewPager}:

mDisplayId=0 mSession=Session{4246e9f0 3563:u0a10059} mClient=android.os.BinderProxy@42495cb8

mOwnerUid=10059 mShowToOwnerOnly=true

mAttrs=WM.LayoutParams{(0,0)(fillxfill) sim=#120 ty=1 fl=#1810580 pfl=0x8 wanim=0x1030001}

Requested w=1280 h=720 mLayoutSeq=207

mHasSurface=false mShownFrame=[0.0,0.0][1280.0,720.0] isReadyForDisplay()=false

WindowStateAnimator{426ba9d8 com.youku.tv/com.youku.tv.ui.activity.HomeActivityWithViewPager}:

Window #3 Window{423f1628 u0 com.cyanogenmod.trebuchet/com.cyanogenmod.trebuchet.Launcher}:

mDisplayId=0 mSession=Session{423e6e50 2300:u0a10049} mClient=android.os.BinderProxy@42781e98

mOwnerUid=10049 mShowToOwnerOnly=true

mAttrs=WM.LayoutParams{(0,0)(fillxfill) sim=#20 ty=1 fl=#1910100 pfl=0x8 fmt=-2 wanim=0x1030295 vsysui=0x400}

Requested w=1280 h=720 mLayoutSeq=187

mHasSurface=false mShownFrame=[0.0,0.0][1280.0,720.0] isReadyForDisplay()=false

WindowStateAnimator{4275a2a8 com.cyanogenmod.trebuchet/com.cyanogenmod.trebuchet.Launcher}:

mWallpaperX=0.5 mWallpaperY=0.5

mWallpaperXStep=0.25 mWallpaperYStep=1.0

Window #4 Window{4245bfb8 u0 com.android.systemui.ImageWallpaper}:

mDisplayId=0 mSession=Session{425e8270 2092:u0a10046} mClient=android.os.BinderProxy@42647aa8

mOwnerUid=10046 mShowToOwnerOnly=true

mAttrs=WM.LayoutParams{(0,0)(2015x1280) gr=#800033 ty=2013 fl=#318 fmt=2 wanim=0x10301ec}

Requested w=2015 h=1280 mLayoutSeq=191

mIsImWindow=false mIsWallpaper=true mIsFloatingLayer=true mWallpaperVisible=false

Offsets x=-368 y=-280

mHasSurface=true mShownFrame=[-368.0,-280.0][1647.0,1000.0] isReadyForDisplay()=false

WindowStateAnimator{42745db0 com.android.systemui.ImageWallpaper}:

Surface: shown=false layer=21000 alpha=1.0 rect=(-368.0,-280.0) 2015.0 x 1280.0

mWallpaperX=0.5 mWallpaperY=0.5

mWallpaperXStep=0.25 mWallpaperYStep=1.0

DisplayContents:

Display: mDisplayId=0

init=1280x720 160dpi cur=1280x720 app=1280x720 rng=720x695-1280x1255

layoutNeeded=false

mCurConfiguration={1.0 ?mcc?mnc zh\_CN ldltr sw720dp w1280dp h695dp 160dpi xlrg long land finger qwerty/v/v dpad/v s.4 themeResource=null}

mCurrentFocus=Window{42688fc8 u0 com.youku.tv/com.youku.tv.ui.activity.AllViedosActivity}

mFocusedApp=AppWindowToken{4277fca8 token=Token{427490e8 ActivityRecord{4277a408 u0 com.youku.tv/.ui.activity.AllViedosActivity}}}

mInTouchMode=true mLayoutSeq=8017

## 其他

4、查看后台services信息：

adb shell service list

5、 adb shell dumpsys notification

获取通知信息

## REF

[dumpsys命令用法](http://gityuan.com/2016/05/14/dumpsys-command/)

# 3.性能分析

## Top

root@zs600b:/ # top -m 10

-t表示查看线程

CTRL+C结束统计，

# Linux cmd

## 网络

netstat | grep 8088：查看网络端口，通常用来调试。

### tcpdump

普通情况下，直接启动tcpdump将监视第一个网络接口上所有流过的数据包。

监视指定网络接口的数据包

tcpdump -i eth1

如果不指定网卡，默认tcpdump只会监视第一个网络接口，一般是eth0，下面的例子都没有指定网络接口。

#### 监视指定主机和端口的数据包

如果想要获取主机210.27.48.1接收或发出的telnet包，使用如下命令

tcpdump tcp port 23 host 210.27.48.1

对本机的udp 123 端口进行监视 123 为ntp的服务端口

tcpdump udp port 123

#### tcpdump 与wireshark联合使用

Wireshark(以前是ethereal)是Windows下非常简单易用的抓包工具。但在Linux下很难找到一个好用的图形化抓包工具。  
还好有Tcpdump。我们可以用Tcpdump + Wireshark 的完美组合实现：在 Linux 里抓包，然后在Windows 里分析包。

tcpdump tcp -i eth1 -t -s 0 -c 100 and dst port ! 22 and src net 192.168.1.0/24 -w ./target.cap

(1)tcp: ip icmp arp rarp 和 tcp、udp、icmp这些选项等都要放到第一个参数的位置，用来过滤数据报的类型  
(2)-i eth1 : 只抓经过接口eth1的包  
(3)-t : 不显示时间戳  
(4)-s 0 : 抓取数据包时默认抓取长度为68字节。加上-S 0 后可以抓到完整的数据包  
(5)-c 100 : 只抓取100个数据包  
(6)dst port ! 22 : 不抓取目标端口是22的数据包  
(7)src net 192.168.1.0/24 : 数据包的源网络地址为192.168.1.0/24  
(8)-w ./target.cap : 保存成cap文件，方便用ethereal(即wireshark)分析

### WIFI

#### Softap wpa\_cli的使用

在平台开发初期阶段，对于WIFI相关问题，上层应用如果还无法使用，或者需要定位问题的层次，这时可以借助wpa\_cli，下面介绍一些wpa\_cli 调试WIFI的基础用法。

借助下面命令可以排除：

Wifi扫描、Wifi连接、Wifi配置等问题。

在串口中执行下面命令：

root@k200:/ # wpa\_cli -i wlan0 -p wlan0

wpa\_cli v2.1-devel-4.3

Copyright (c) 2004-2013, Jouni Malinen <j@w1.fi> and contributors

This software may be distributed under the terms of the BSD license.

See README for more details.

Interactive mode

>

此时进入与wpa\_supplicant 的交换模式。输入h能查看所有wpa\_cli的支持的命令，由于命令太多，此处不贴出来。下面以表格形式介绍wpa\_cli基本命令以及用法：

|  |  |  |  |
| --- | --- | --- | --- |
| **命令** | **命令缩写** | **描述** | **测试** |
| status | stat | 显示当前连接的状态信息 | > stat  bssid=c8:3a:35:16:89:88  ssid=Tenda\_168988  id=1  mode=station  pairwise\_cipher=CCMP  group\_cipher=CCMP  key\_mgmt=WPA-PSK  wpa\_state=COMPLETED  ip\_address=192.168.0.142  address=00:50:43:02:ff:01  > |
| ping |  | 测试指令，看service端是否有响应 | > ping  PONG  > |
| interface |  | 显示当前可用的interface，wpa\_supplicant 支持多interface | > interface  Available interfaces:  wlan0  > |
| set |  | 设置环境变量，如果不带参数，显示当前环境变量 | > set  set variables:  EAPOL::heldPeriod (EAPOL state machine held period, in seconds)  EAPOL::authPeriod (EAPOL state machine authentication period, in seconds)  EAPOL::startPeriod (EAPOL state machine start period, in seconds)  EAPOL::maxStart (EAPOL state machine maximum start attempts)  dot11RSNAConfigPMKLifetime (WPA/WPA2 PMK lifetime in seconds)  dot11RSNAConfigPMKReauthThreshold (WPA/WPA2 reauthentication threshold  percentage)  dot11RSNAConfigSATimeout (WPA/WPA2 timeout for completing security  association in seconds)  > |
| reassociate | reas | 设置连接标签，开启扫描，查找可用的AP进行连接，如果当前是连接状态，只会扫描一次。 | <3>CTRL-EVENT-SCAN-RESULTS  <3>WPS-AP-AVAILABLE  <3>Trying to associate with 94:ba:56:00:32:14 (SSID='yl\_B19W4' freq=2462 MHz)  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=5 BSSID=94:ba:56:00:32:14  <3>CTRL-EVENT-STATE-CHANGE id=0 state=6 BSSID=94:ba:56:00:32:14  <3>Associated with 94:ba:56:00:32:14  <3>CTRL-EVENT-CONNECTED - Connection to 94:ba:56:00:32:14 completed (reauth) [id=0 id\_str=]  <3>CTRL-EVENT-STATE-CHANGE id=0 state=9 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-STATE-CHANGE id=0 state=0 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-DISCONNECTED bssid=00:00:00:00:00:00 reason=0  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=3 BSSID=00:00:00:00:00:00 |
| preauthenticate | pr | 针对一个AP进行预认证 | > pr c8:3a:35:16:89:88  OK  <3>RSN: failed to get master session key from pre-auth EAPOL state machines  <3>RSN: pre-authentication with c8:3a:35:16:89:88 failed |
| identity | id | 设置identity | > id 0 xxxx  OK  > save\_config  OK  >  cat wpa\_supplicant.conf  network={  ssid="yl\_B19W4"  key\_mgmt=NONE  identity="xxxx"  password="123455678"  disabled=1  } |
| password |  | 设置密码 | password 0<network id> 11111111  network={  ssid="yl\_B19W4"  key\_mgmt=NONE  identity="xxxx"  password="11111111111"  disabled=1  } |
| disconnect | disc | 断开连接，并且会告知wpa\_supplicant 不在连接其他AP | > disc  OK  <3>CTRL-EVENT-STATE-CHANGE id=1 state=0 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-DISCONNECTED bssid=00:00:00:00:00:00 reason=0 |
| reconnect | reconn | 重新连接，在运行上面的过程后，运行此命令 | > reconn  OK  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=3 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-SCAN-RESULTS  <3>WPS-AP-AVAILABLE  <3>Trying to associate with a4:a8:0f:fc:18:3a (SSID='XXXXXXX' freq=2462 MHz)  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=5 BSSID=a4:a8:0f:fc:18:3a  <3>CTRL-EVENT-STATE-CHANGE id=1 state=6 BSSID=a4:a8:0f:fc:18:3a  <3>Associated with a4:a8:0f:fc:18:3a  <3>CTRL-EVENT-CONNECTED - Connection to a4:a8:0f:fc:18:3a completed (reauth) [id=1 id\_str=]  <3>CTRL-EVENT-STATE-CHANGE id=1 state=9 BSSID=00:00:00:00:00:00 |
| quit | q | 退出wpa\_cli |  |
| terminate | term | 杀掉后台wpa\_supplicant进程 |  |
| reconfigure | recon | 重新导入配置文件，可以在调试时，修改配置文件后运行此命令，使配置文件生效 |  |
| scan | scan | 扫描 | scan  > OK  <3>CTRL-EVENT-SCAN-RESULTS  <3>WPS-AP-AVAILABLE |
| scan\_result | scan\_r | 显示扫描结果 | scan\_r  > bssid / frequency / signal level / flags / ssid  a4:a8:0f:fc:18:3a 2462 -59 [ESS] XXXXXXX  d4:ee:07:03:ee:0c 2427 -68 [WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS] HiWiFi\_03EE0C  d8:5d:4c:25:d0:f2 2412 -72 [WPA-PSK-CCMP][WPA2-PSK-CCMP-preauth][ESS] OTG  c8:3a:35:16:89:88 2462 -83 [WPA-PSK-CCMP][ESS] Tenda\_168988 |
| bss |  | 获取扫描结果 某一个热点的具体信息 | bss <<idx> | <bssid>>  bss 1  > id=371  bssid=d4:ee:07:03:ee:0c  freq=2427  beacon\_int=100  capabilities=0x0431  qual=0  noise=0  level=-72  tsf=0000016060803119  ie=000d4869576946695f303345453043010882848b960c1218240301042a010032043048606c2d1a6e101bff000000000000000000000000000000000000000000003d16040d000000000000000000000000000000000000000030140100000fac040100000fac040100000fac020000dd160050f20101000050f20401000050f20401000050f202dd180050f2020101000003a4000027a4000042435e0062322f00dd06d4ee07010100  flags=[WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]  ssid=HiWiFi\_03EE0C  > |
| list\_networks | list\_n | 已经保存配置文件中的AP | > list  network id / ssid / bssid / flags  0 yl\_B19W4 c8:3a:35:16:89:88  1 XXXXXXX any  > |
| bssid |  | 配置ssid的bssid | bssid 0 c8:3a:35:16:89:88  结果下面可以看到 |
| add\_network | add\_n | 添加一个AP配置，在framework还没调通之前，可以用此命令进行配置 | > add\_n  7  > save\_config  OK  > list  network id / ssid / bssid / flags  0 yl\_B19W4 c8:3a:35:16:89:88 [DISABLED]  2 客厅里的飞看\_B186M any [DISABLED]  3 客厅里的飞看\_B1PKO any [DISABLED]  4 XXXXXXX any [CURRENT]  5 Tenda\_168988 any [DISABLED]  6 any [DISABLED]  7 any [DISABLED] --🡪添加的  >save\_config --🡪如果想保存到配置文件中 |
| set\_network | set\_n | 设置ap的属性，用set\_network 不接参数，可以看到所有可用信息。 | > set\_n  set\_network variables:  ssid (network name, SSID)  psk (WPA passphrase or pre-shared key)  key\_mgmt (key management protocol)  identity (EAP identity)  password (EAP password)  ...  ------------------------------------------------------  > set\_n 7 ssid "B2C2.5"  OK  > list  network id / ssid / bssid / flags  0 yl\_B19W4 c8:3a:35:16:89:88 [DISABLED]  2 客厅里的飞看\_B186M any [DISABLED]  3 客厅里的飞看\_B1PKO any [DISABLED]  4 XXXXXXX any [CURRENT]  5 Tenda\_168988 any [DISABLED]  6 any [DISABLED]  7 B2C2.5 any [DISABLED]  > set\_n 7 psk 22225555  FAIL  > set\_n 7 psk "22225555"  OK  > list  network id / ssid / bssid / flags  0 yl\_B19W4 c8:3a:35:16:89:88 [DISABLED]  2 客厅里的飞看\_B186M any [DISABLED]  3 客厅里的飞看\_B1PKO any [DISABLED]  4 XXXXXXX any [CURRENT]  5 Tenda\_168988 any [DISABLED]  6 any [DISABLED]  7 B2C2.5 any [DISABLED]  > save  OK  配置成功 |
| select\_network | select\_n | 选择一个已经配置的AP进行连接，这个命令用的比较多。上面set\_n 命令进行配置后，可以用这命令进行连接 | > sel 7  OK 这样supplicant会自动连接B2C2.5 这个热点  --------------  > stat  bssid=c8:3a:35:16:89:88  ssid=Tenda\_168988  id=5  mode=station  pairwise\_cipher=CCMP  group\_cipher=CCMP  key\_mgmt=WPA-PSK  wpa\_state=COMPLETED  ip\_address=192.168.0.142  address=00:50:43:02:ff:01  ---------------------------------------  查看当前状态  ---------------------------------------  > list  network id / ssid / bssid / flags  0 yl\_B19W4 c8:3a:35:16:89:88 [DISABLED]  2 客厅里的飞看\_B186M any [DISABLED]  3 客厅里的飞看\_B1PKO any [DISABLED]  4 XXXXXXX any [DISABLED]  5 Tenda\_168988 any [CURRENT]  ---------- 查看当前可选的AP  ----------  > sel 4  OK  <3>CTRL-EVENT-STATE-CHANGE id=5 state=0 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=3 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-DISCONNECTED bssid=00:00:00:00:00:00 reason=0  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=0 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=3 BSSID=00:00:00:00:00:00  <4>Failed to initiate AP scan  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=0 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=3 BSSID=00:00:00:00:00:00  <4>Failed to initiate AP scan  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=0 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=3 BSSID=00:00:00:00:00:00  <4>Failed to initiate AP scan  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=0 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-SCAN-RESULTS  <3>WPS-AP-AVAILABLE  <3>Trying to associate with a4:a8:0f:fc:18:3a (SSID='XXXXXXX' freq=2462 MHz)  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=5 BSSID=a4:a8:0f:fc:18:3a  <3>CTRL-EVENT-STATE-CHANGE id=4 state=6 BSSID=a4:a8:0f:fc:18:3a  <3>Associated with a4:a8:0f:fc:18:3a  <3>CTRL-EVENT-CONNECTED - Connection to a4:a8:0f:fc:18:3a completed (reauth) [id=4 id\_str=]  <3>CTRL-EVENT-STATE-CHANGE id=4 state=9 BSSID=00:00:00:00:00:00  ------- 选择进行连接 -------  stat  > bssid=a4:a8:0f:fc:18:3a  ssid=XXXXXXX  id=4  mode=station  pairwise\_cipher=NONE  group\_cipher=NONE  key\_mgmt=NONE  wpa\_state=COMPLETED  ip\_address=192.168.88.101  address=00:50:43:02:ff:01  >  查看当前状态，连接ok |
| enable\_network | enable\_n | 将network 的标志位  disable 置0，该位为1时，wifi在断开连接状态，不会去主动连接该热点。如果wifi不主动连接任何热点，可以查看配置文件。 | network={  ssid="客厅里的飞看\_B186M"  key\_mgmt=NONE  disabled=1  }  network={  ssid="客厅里的飞看\_B1PKO"  key\_mgmt=NONE  disabled=1  }  network={  ssid="XXXXXXX"  key\_mgmt=NONE  }  network={  ssid="Tenda\_168988"  psk="12345678"  key\_mgmt=WPA-PSK  disabled=1  } |
| disable\_network | disable\_n | 对应上面  disable\_network network\_id |  |
| remove\_network | remove\_n | 忘记网络  remove\_network id  会删除对应network配置 |  |
| get\_network | get\_n | 获取network的属性  对应之前的set\_network | set\_network variables:  ssid (network name, SSID)  psk (WPA passphrase or pre-shared key)  key\_mgmt (key management protocol)  identity (EAP identity)  password (EAP password)  ...  > get\_network 0 ssid  "yl\_B19W4"  > list  network id / ssid / bssid / flags  0 yl\_B19W4 c8:3a:35:16:89:88 [DISABLED]  2 客厅里的飞看\_B186M any [DISABLED]  3 客厅里的飞看\_B1PKO any [DISABLED]  4 XXXXXXX any [DISABLED]  5 Tenda\_168988 any [DISABLED]  6 any [DISABLED]  7 B2C2.5 any [DISABLED]  > |
| wps\_pbc |  | wps push button功能 | 前提:路由开启wps push button功能  下面我已经打开Tenda\_168988 wps push button功能，然后在盒子上运行wps\_pbc  wps\_pbc  > OK  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=3 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-SCAN-RESULTS  <3>WPS-AP-AVAILABLE  <3>CTRL-EVENT-SCAN-RESULTS  <3>WPS-AP-AVAILABLE  <3>CTRL-EVENT-SCAN-RESULTS  <3>WPS-AP-AVAILABLE  <3>CTRL-EVENT-SCAN-RESULTS  <3>WPS-AP-AVAILABLE-PBC  <3>Trying to associate with c8:3a:35:16:89:88 (SSID='Tenda\_168988' freq=2412 MHz)  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=5 BSSID=c8:3a:35:16:89:88  <3>CTRL-EVENT-STATE-CHANGE id=8 state=6 BSSID=c8:3a:35:16:89:88  <3>Associated with c8:3a:35:16:89:88  <3>CTRL-EVENT-EAP-STARTED EAP authentication started |
| wps\_pin |  | Wps pin 功能 | 前提:打开路由热点wps功能，并设置pin值 如12345678，用scan\_r查看你想要连接的ssid的bssid，然后用下面命令进行连接  > wps\_pin c8:3a:35:16:89:88 12345678  12345678  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=3 BSSID=00:00:00:00:00:00  <3>CTRL-EVENT-SCAN-RESULTS  <3>WPS-AP-AVAILABLE-AUTH  <3>Trying to associate with c8:3a:35:16:89:88 (SSID='Tenda\_168988' freq=2412 MHz)  <3>CTRL-EVENT-STATE-CHANGE id=-1 state=5 BSSID=c8:3a:35:16:89:88  <3>CTRL-EVENT-STATE-CHANGE id=8 state=6 BSSID=c8:3a:35:16:89:88  <3>Associated with c8:3a:35:16:89:88  <3>CTRL-EVENT-EAP-STARTED EAP authentication started |
| save\_config | save\_c | 保存信息到配置文件中，每次修改配置后记得用 |  |
|  |  |  |  |
|  |  |  |  |

#### Softap hostapd\_cli的使用

hostapd\_cli相对wpa\_cli来说比较简单，能使用的命令也非常少。

借助下面命令可以解决：

wifi热点配置、查看设备连接时长等问题。

hostapd\_cli –h

Options:

-h help (show this usage text)

-v shown version information

-p<path> path to find control sockets (default: /var/run/hostapd)

-a<file> run in daemon mode executing the action file based on events

from hostapd

-B run a daemon in the background

-i<ifname> Interface to listen on (default: first interface found in the

socket path)

hostapd\_cli 运行:

hostapd\_cli -i uap0 -p /data/misc/wifi/hostapd

下面列表列出了hostapd\_cli能使用的指令，以及使用方法：

|  |  |  |  |
| --- | --- | --- | --- |
| 命令 | 命令缩写 | 描述 | 测试 |
| all\_sta | al | 显示当前连入的STA,并记录连上的时间长度 | > all\_sta  40:f3:08:39:b6:74  connected\_time=579  00:90:4c:78:5d:5c  connected\_time=3359  > all  00:90:4c:78:5d:5c  connected\_time=3458 |
| deauthenticate <addr> | dea | 解除验证 | > all  40:f3:08:39:b6:74 ---🡪我的设备  connected\_time=59  00:90:4c:78:5d:5c  connected\_time=4008  > dea 40:f3:08:39:b6:74 -🡪可以看到设备断开，又启动连接  OK |
| disassociate <addr> |  | 断开连接 | 过程与上面差不多 |
| wps\_pin |  | AP端的wps功能 |  |
| wps\_pbc |  | Ap端的wps功能 |  |
| wps\_config <SSID> <auth> <encr> <key> |  | 配置Ap的属性 |  |

-s snaplen ] [ -T type ] [ -w file ]  
 [ -W filecount ]  
 [ -E spi@ipaddr algo:secret,... ]  
 [ -y datalinktype ] [ -Z user ]  
 [ expression ]

#### 打开wifi-log

|  |
| --- |
| 1. <4>[ 3303.925164] wifi\_power\_write: poweren = 1 2. <6>[ 3303.925191] [WLAN\_RFKILL]: rockchip\_wifi\_power: 1 3. <6>[ 3303.925201] [BT\_RFKILL]: rfkill\_get\_bt\_power\_state: rfkill-bt driver has not Successful initialized 4. <6>[ 3303.925210] [WLAN\_RFKILL]: rockchip\_wifi\_ref\_voltage: 1 5. <6>[ 3303.925218] [WLAN\_RFKILL]: rockchip\_wifi\_ref\_voltage: wifi io reference voltage control is disabled. 6. <6>[ 3304.027411] extfa98xx\_control(): status: ON. 7. <6>[ 3304.027443] extfa98xx\_control(): profile: 0, vstep: 0. 8. <6>[ 3304.027604] rk\_rt5650:Enter::rk\_rt5650\_hw\_params----85 9. <6>[ 3304.027623] rk\_rt5650:Enter:rk\_rt5650\_hw\_params, 129, rate=48000 10. <6>[ 3304.027637] rk\_rt5650:Enter:rk\_rt5650\_hw\_params, 143, pll\_out/4/params\_rate(params) = 64 11. <6>[ 3305.805955] EHCI: rk\_ehci\_hcd\_enable, enable host controller 12. <6>[ 3306.027223] usb 1-1: new high-speed USB device number 10 using rockchip\_ehci\_host 13. <6>[ 3306.177152] usb 1-1: New USB device found, idVendor=0cf3, idProduct=1022 14. <6>[ 3306.177339] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3 15. <6>[ 3306.177435] usb 1-1: Product: USBWLAN 16. <6>[ 3306.177511] usb 1-1: Manufacturer: Qualcomm Atheros. 17. <6>[ 3306.177590] usb 1-1: SerialNumber: 12345678 18. <6>[ 3306.883487] [WLAN\_RFKILL]: wifi turn on power. 171 19. <6>[ 3306.908010] Calling CRDA to update world regulatory domain 20. <6>[ 3306.908644] World regulatory domain updated: 21. <6>[ 3306.908722] (start\_freq - end\_freq @ bandwidth), (max\_antenna\_gain, max\_eirp) 22. <6>[ 3306.908812] (2302000 KHz - 2382000 KHz @ 20000 KHz), (300 mBi, 500 mBm) 23. <6>[ 3306.908893] (2402000 KHz - 2482000 KHz @ 20000 KHz), (300 mBi, 500 mBm) 24. <6>[ 3306.908972] (2474000 KHz - 2494000 KHz @ 20000 KHz), (300 mBi, 500 mBm) 25. <6>[ 3306.909048] (5110000 KHz - 5250000 KHz @ 20000 KHz), (300 mBi, 500 mBm) 26. <6>[ 3306.909125] (5250000 KHz - 5710000 KHz @ 20000 KHz), (300 mBi, 500 mBm) 27. <6>[ 3306.909201] (5735000 KHz - 5835000 KHz @ 20000 KHz), (300 mBi, 500 mBm) 28. <6>[ 3306.917316] 6ath6kl: 3VAP/1, P2P enable, concurrent on , with dedicate p2p-device, multi-channel-concurrent enable, p2p-compat disable, sta-p2p-ie r 29. d, p2p\_wise\_scan enable 30. <6>[ 3306.918561] 6ath6kl: target bootstrap: 0x00000800 31. <6>[ 3306.919563] 6ath6kl: target's subtype is 0x7, HT20/40 2SS Dual-band 32. <6>[ 3306.924038] 6ath6kl: Create HTC cookie, type 1 num 440 33. <6>[ 3306.924091] 6ath6kl: Create HTC cookie, type 2 num 66 34. <6>[ 3306.924112] 6ath6kl: P2P flowctrl enabled 35. <6>[ 3306.924126] 6ath6kl: RX aggregation drop disabled 36. <6>[ 3306.924142] 6ath6kl: Using driver's regdb. 37. <6>[ 3307.139933] 6ath6kl: ar6004 hw 3.0 usb fw 3.5.0.607 38. <6>[ 3307.140166] Calling CRDA for country: CN 39. <6>[ 3307.140567] 6ath6kl: Ignore reg notifier call back, since initiator is driver 40. <6>[ 3307.140620] Regulatory domain changed to country: CN 41. <6>[ 3307.140660] (start\_freq - end\_freq @ bandwidth), (max\_antenna\_gain, max\_eirp) 42. <6>[ 3307.140710] (2402000 KHz - 2482000 KHz @ 20000 KHz), (400 mBi, 2000 mBm) 43. <6>[ 3307.140758] (5170000 KHz - 5250000 KHz @ 20000 KHz), (500 mBi, 2000 mBm) 44. <6>[ 3307.140804] (5735000 KHz - 5835000 KHz @ 20000 KHz), (500 mBi, 3200 mBm) 45. <6>[ 3307.146820] 6ath6kl: Wait defer tasks done... 46. <6>[ 3307.147262] 6ath6kl: Create dedicated p2p interface 47. <6>[ 3307.168883] 6ath6kl: Enable Firmware crash notiry. 48. <6>[ 3307.169621] usbcore: registered new interface driver ath6kl\_usb 49. <4>[ 3307.172931] Current WiFi chip is AR1021X. 50. <6>[ 3307.209728] extfa98xx\_control(): status: OFF. 51. <6>[ 3307.278231] IPv6: ADDRCONF(NETDEV\_UP): wlan0: link is not ready 52. <4>[ 3307.634653] Current WiFi chip is AR1021X. 53. <6>[ 3307.642981] IPv6: ADDRCONF(NETDEV\_UP): p2p0: link is not ready |

adb shell cat /data/misc/wifi/\*.conf  
查看wifi密码：

## lsusb

lsusb命令用于显示本机的USB设备列表，以及USB设备的详细信息。 lsusb命令是一个学习USB驱动开发，认识USB设备的助手，推荐大家使用，如果您的开发板中或者产品中没有lsusb命令可以自己移植一个，放到文件系统里面。来自: <http://man.linuxde.net/lsusb>

## grep

grep -E 'aaaaa|bbbbbb'

grep -E ‘net|dns’