


Android中AM、PM、dumpsys命令使用总结



吾若成疯 (/u/24c2a4e74009) + 关注

2017.10.12 13:54* 字数 1719 阅读 42 评论 0 喜欢 3

(/u/24c2a4e74009)

在平时开发中，通过命令行有时候能够快速的帮我们实现一些功能，这里对常用的命令做一些总结。

1、AM命令用法

1.1、简单介绍

am指令是 activity manager的缩写，可以启动Service、Broadcast，杀进程，监控等功能，这些功能都非常便捷调试程序。

可以通过adb shell 进入Android 的Linux命令界面，输入am -help查看详细命令，先介绍几个简单用法，

```
//使用Action方式打开系统设置-输入法设置
am start -a android.settings.INPUT_METHOD_SETTINGS

//使用Action方式打开网站https://amberweather.com
am start -a android.intent.action.VIEW -d https://amberweather.com

//打开拨号界面，并传递一个DATA_URI数据给拨号界面
am start -a android.intent.action.CALL -d tel:10086
```

1.2、AM命令

命令格式如下

```
am [command] [options]
```

命令列表：

命令	功能	实现方法
----	----	------

命令	功能	实现方法
am start [options]	启动Activity	startActivityAsUser
am startservice	启动Service	startService
am stopservice	停止Service	stopService
am broadcast	发送广播	broadcastIntent
am restart	重启	restart
am dumpheap <pid> <file>	进程pid的堆信息输出到file	dumpheap
am send-trim-memory <pid> <level>	收紧进程的内存	setProcessMemoryTrimLevel
am kill <PACKAGE>	杀指定后台进程	killBackgroundProcesses
am kill-all	杀所有后台进程	killAllBackgroundProcesses
am force-stop <PACKAGE>	强杀进程	forceStopPackage
am hang	系统卡住	hang
am monitor	监控	MyActivityController.run

原理分析：**am**命令实的实现方式在**Am.java**，最终几乎都是调用**ActivityManagerService**相应的方法来完成的，**am monitor**除外。比如前面概述中介绍的命令**am start -a and roid.intent.action.VIEW -d https://amberweather.com (https://amberweather.com)**，启动**Acitvty**最终调用的是**ActivityManagerService**类的**startActivityAsUser()**方法来完成的。再比如**am kill-all**命令，最终的实现工作是由**ActivityManagerService**的**killBackgroundProcesses()**方法完成的。


下面说一下[options]和 <INTENT>参数的意义以及如何正确取值。

1.3、Options

1.3.1 启动Activity

主要是启动Activity命令**am start [options] <INTENT>**使用options参数，接下来列举Activi
ty命令的[options]参数：

```
-D: 开启debug模式
-W: 等待启动完成
--start-profiler<FILE>: 将profiler中的结果输出到指定文件中
-P: 和--start-profiler一样，区别在于，在app进入idle状态时profiler结束
-R <Count>: 重复启动Activity，但每次重复启动都会关闭掉最上面的Activity
-S: 关闭Activity所属的App进程后再启动Activity
--opengl-trace: 开启OpenGL tracing
--user <USER_ID> : 使用指定的用户来启动activity，如果不输入，则使用当前用户执行
```



启动Activity的实现原理：存在-W参数则调用startActivityAndWait()方法来运行，否则sta

rtActivityAsUser()。

1.3.2 收紧内存

命令

```
am send-trim-memory <pid> <level>
```

例如：向pid=12345的进程，发出level=RUNNING_LOW的收紧内存命令

```
am send-trim-memory 12345 RUNNING_LOW。
```

level取值范围为：HIDDEN、RUNNING_MODERATE、BACKGROUND、RUNNING_LOW、MODERATE、RUNNING_CRITICAL、COMPLETE

1.3.3 其他

am的子命令，startservice, stopservice, broadcast, kill, profile start, profile stop, dumph eap的可选参数都允许设置--user <USER_ID>。目前市面上的绝大多数手机还是单用户模式，因此可以忽略该参数，默认为当前用户。

例如：启动id=10001的用户的指定service。

```
am startservice --user 10010 [Intent]
```

1.4 Intent

Intent的参数和flags较多，为了方便，这里分为3种类型参数，常用参数，Extra参数，Flags参数

1.4.1 常用参数

```
-a <ACTION>: 指定Intent action，实现原理Intent.setAction();  
-n <COMPONENT>: 指定组件名，格式为{包名}/{主Activity名}，实现原理Intent.setComponent();  
-d <DATA_URI>: 指定Intent data URI  
-t <MIME_TYPE>: 指定Intent MIME Type  
-c <CATEGORY> [-c <CATEGORY>] ...]:指定Intent category，实现原理Intent.addCategory()  
-p <PACKAGE>: 指定包名，实现原理Intent.setPackage();  
-f <FLAGS>: 添加flags，实现原理Intent.setFlags(int)，紧接着的参数必须是int型；
```

实例



```
am start -a android.intent.action.VIEW
am start -n mobi.infolife.ezweather.locker.locker_2/.LockerActivity
am start -d content://contacts/people/1
am start -t image/png
am start -c android.intent.category.APP_CONTACTS
```

1.4.2 Extra参数

(1). 基本类型

参数	-e/-es	-esn	-ez	-ei	-el	-ef	-eu
类型	String	(String)	boolean	int	long	float	uri

参数es是Extra String首字母简称，实例：

```
am start -n com.cnr.tlive/.MainActivity -es test
```

(2). 数组类型

参数	-esa	-eia	-ela	-efa
数组类型	<div>String[]</div>	int[]	long[]	float[]

参数eia，是Extra int array首字母简称，多个value值之间以逗号隔开，实例：

```
am start -n com.cnr.tlive/com.cnr.tlive.activity.MainActivity -ela day 1,2,3,4,5
```

(3). ArrayList类型

参数	-esal	-eial	-elal	-efal
List类型	String	int	long	float

参数efal，是Extra float Array List首字母简称，多个value值之间以逗号隔开，实例：

```
am start -n com.cnr.tlive.activity.MainActivity-efal nums 1.2,2.2
```

2、PM命令用法

2.1、简单介绍

pm工具为包管理（package manager）的简称，可以使用pm工具来执行应用的安装和查询应用宝的信息、系统权限、控制应用，pm工具是Android开发与测试过程中必不可



少的工具，shell命令格式如下：

pm <command>

命令	功能	实现方法
list packages	列举app包信息	PMS.getInstalledPackages
install [options] <PATH>	安装应用	PMS.installPackageAsUser
uninstall [options] <package>	卸载应用	IPackageInstaller.uninstall
enable <包名或组件名>	enable	PMS.setEnabledSetting
disable <包名或组件名>	disable	PMS.setEnabledSetting
hide <package>	隐藏应用	PMS.setApplicationHiddenSettingAsUser
unhide <package>	显示应用	PMS.setApplicationHiddenSettingAsUser
get-install-location	获取安装位置	PMS.getInstallLocation
set-install-location	设置安装位置	PMS.setInstallLocation
path <package>	查看App路径	PMS.getPackageInfo
clear <package>	清空App数据	AMS.clearApplicationUserData
get-max-users	最大用户数	UserManager.getMaxSupportedUsers
force-dex-opt <package>	dex优化	PMS.forceDexOpt
dump <package>	dump信息	AM.dumpPackageStateStatic
trim-caches <目标size>	紧缩cache目标大小	PMS.freeStorageAndNotify

原理分析：**pm**命令实的实现方式在**Pm.java**，最后大多数都是调用**PackageManagerService**相应的方法来完成的。**disbale**之后，在桌面和应用程序列表里边都看到不该app。

2.2、详细参数

2.2.1 list packages

查看所有的package，

list packages [options] <FILTER>



[options]参数：

```
-f: 显示包名所关联的文件；  
-d: 只显示disabled包名；  
-e: 只显示enabled包名；  
-s: 只显示系统包名；  
-3: 只显示第3方应用的包名；  
-i: 包名所相应的installer；  
-u: 包含uninstalled包名。
```

disabled + enabled = 总应用个数；系统 + 第三方 = 总应用个数。

查看第3方应用：

```
pm list packages -3
```

查看已经被禁用的包名

```
pm list packages -d
```

<FILTER>参数

当FILTER不为空时，则只会输出包名带有FILTER字段的应用；当FILTER为空时，则默认显示所有满足条件的应用。

例如，查看包名带有weather字段的包名

```
pm list packages weather
```

2.2.3 pm Install

```
pm install [options] <PATH>
```

[options]参数：

```
-r: 覆盖安装已存在Apk，并保持原有数据；  
-d: 运行安装低版本Apk；  
-t: 运行安装测试Apk  
-i : 指定Apk的安装器；  
-s: 安装apk到共享快存储，比如sdcard；  
-f: 安装apk到内部系统内存；  
-l: 安装过程，持有转发锁  
-g: 准许Apk manifest中的所有权限；
```

<PATH>参数：指的是需要安装的apk所在的路径



2.2.4其他参数

```
pm list users //查看当前手机用户
pm list libraries //查看当前设备所支持的库
pm list features //查看系统所有的features
pm list instrumentation //所有测试包的信息
pm list permission-groups //查看所有的权限组
pm list permissions [options] <group> 查看权限
    -g: 以组形式组织；
    -f: 打印所有信息；
    -s: 简要信息；
    -d: 只列举危险权限；
    -u: 只列举用户可见的权限。
```

3、dumpsys

3.1 dumpsys原理简介

dumpsys是Android自带的强大debug工具,从名字就可以看出,主要是用于dump 当前android system的一些信息,是一项分析手机问题,运行状态,使用情况等十分有效的手段。

实现原理

dumpsys的源码结构其实很简单,只有一个dumpsys.cpp
/frameworks/native/cmds/dumpsys/dumpsys.cpp

```
int main(int argc, char* const argv[])
{
    ...
    //获取defaultServiceManager
    sp<IServiceManager> sm = defaultServiceManager();
    ...
    Vector<String16> services;
    ...
    services = sm->listServices();
    ...
    const size_t N = services.size();

    for (size_t i=0; i<N; i++) {
        sp<IBinder> service = sm->checkService(services[i]);
        ...
        int err = service->dump(STDOUT_FILENO, args);
        ...
    }

    return 0;
}
```

先通过defaultServiceManager()函数获得ServiceManager对象,然后根据dumpsys传进来的参数通过函数checkService来找到具体的service,并执行该service的dump方法,达到dump service的目的。

3.2 dumpsy命令



3.2.1 服务列表

不同的Android系统版本支持的命令有所不同，可通过下面命令查看当前手机所支持的dump服务，先进入adb shell，再执行如下命令：dumpsys -l。 这些服务名可能并看不出其调用的哪个服务，可以通过下面指令：service list。

服务列表有很多，这里简单介绍几种

服务名	类名	功能
activity	ActivityManagerService	AMS相关信息
package	PackageManagerService	PMS相关信息
window	WindowManagerService	WMS相关信息
input	InputManagerService	IMS相关信息
power	PowerManagerService	PMS相关信息
battery	BatteryService	电池信息
dropbox	DropboxManagerService	<div>调试相关</div>
cpuinfo	CpuBinder	CPU
meminfo	MemBinder	内存
dbinfo	DbBinder	数据库

3.2.2 查询服务

通过下面命令可打印具体某一项服务：dumpsys <service>，其中service便是前面表格中的服务名

```
dumpsys cpuinfo //打印一段时间进程的CPU使用百分比排行榜
dumpsys meminfo -h //查看dump内存的帮助信息
dumpsys package <packagename> //查看指定包的信息
```

接下来主要说下dumpsys activity 用法

3.3 Activity

命令

```
dumpsys activity [options] [cmd]
```

options可选值




```
-a : dump所有 ;
-c : dump客户端 ;
-p [package] : dump指定的包名 ;
-h : 输出帮助信息 ;
```

dumpsys activity等价于依次输出下面7条指令：

```
dumpsys activity intents
dumpsys activity broadcasts
dumpsys activity providers
dumpsys activity services
dumpsys activity recents
dumpsys activity activities
dumpsys activity processes
```

cmd可选值

cmd	说明	缩写
activities	activity状态	a
broadcasts	广播	b
intents	pending intent状态	i
processes	进程	p
oom	内存溢出	o
services	Service状态	s
providers	ContentProvider状态	prov
provider	ContentProvider状态(Client端)	
package	package相关信息	
all	所有的activities信息	
recents	recent activity状态	
top	top activity信息	

注：

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 act
ivity broadcasts 和 dumpsys activity b 等效



3.4 dumpsys meminfo

命令

```
dumpsys meminfo
```

返回结果



```
Total PSS by process://以process来划分
122677 kB: com.android.browser (pid 5807 / activities)
96473 kB: system (pid 661)
93999 kB: com.android.systemui (pid 745)
90564 kB: mobi.infolife.ezweather.locker.locker_2 (pid 1115 / activities)
56021 kB: mobi.infolife.ezweather.widget.soccer_season (pid 1745)
50044 kB: mobi.infolife.ezweather.widget.gcolour (pid 5147)
40171 kB: com.android.launcher3 (pid 887 / activities)
...

Total PSS by OOM adjustment: //以oom来划分.
112247 kB: Native
    34819 kB: local_opengl (pid 231)
    24076 kB: zygote (pid 342)
    17226 kB: mediaserver (pid 340)
    9041 kB: genybaseband (pid 138)
    3881 kB: logd (pid 115)
    2071 kB: wpa_supplicant (pid 872)
    1976 kB: genyd (pid 132)
    1678 kB: drmserver (pid 238)
    1648 kB: surfaceflinger (pid 339)
    1444 kB: vold (pid 122)
    1330 kB: sdcard (pid 736)
    1118 kB: netd (pid 338)
    864 kB: adbd (pid 136)
    768 kB: debuggerd (pid 236)
    763 kB: keystore (pid 241)
    724 kB: /init (pid 1)

Total PSS by category:// 以category划分
216031 kB: Native
169950 kB: Dalvik
121137 kB: .dex mmap
73616 kB: Ashmem
68426 kB: Unknown
64952 kB: .apk mmap
63304 kB: .so mmap
53932 kB: .oat mmap
27704 kB: .art mmap
15919 kB: Other mmap
14777 kB: Dalvik Other
9222 kB: .ttf mmap
8548 kB: Stack
148 kB: Other dev
0 kB: Cursor
0 kB: Gfx dev
0 kB: .jar mmap
0 kB: EGL mtrack
0 kB: GL mtrack
0 kB: Other mtrack

Total RAM: 4049068 kB (status normal)//整体情况
Free RAM: 3105215 kB (65839 cached pss + 503700 cached kernel + 2535676 free)
Used RAM: 1020031 kB (841827 used pss + 178204 kernel)
Lost RAM: -76178 kB
Tuning: 96 (large 256), oom 184320 kB, restore limit 61440 kB (high-end-gfx)
```

上面的输出结果可以分为以下四个部分



PSS- Proportional Size 实际使用的物理内存（比例分配共享库占用的内存）

按比例包含共享库所占用的内存，比如有9k的共享库被3个进程使用，那个当前进程所占用的大小被计算为9/3k，也就是3k

-序列	划分类型	排序	解释
1	process	PSS	以进程的PSS从大到小依次排序显示，每行显示一个进程；
2	OOM adj	PSS	Native/System/Persistent/Foreground/Visible/Perceptible/A Services/Home/B Services/Cached，分别显示每类的进程情况
3	category	PSS	以Dalvik/Native/.art mmap/.dex map等划分的各类进程的总PSS情况
4	total	-	总内存、剩余内存、可用内存、其他内存

也可以只输出某个pid或package的进程信息：

```
dumpsys meminfo <pid> // 输出指定pid的某一进程
dumpsys meminfo --package <packagename> // 输出指定包名的进程，可能包含多个进程
```

3.4 使用场景

下面以AmberLocker作为实例进行分析

场景1：查询某个App所有的Service状态

```
dumpsys activity s mobi.infolife.ezweather.locker.locker_2
```

解读：Service类名为com.amber.lockscreen.LockerHeartService，包名为mobi.infolife.ezweather.locker.locker_2，baseDir(apk路径)为/data/app/mobi.infolife.ezweather.locker.locker_2-2/base.apk，dataDir((apk数据路径)运行在进程pid=1115，进程名为进程名为mobi.infolife.ezweather.locker.locker_2，，uid=10060,还有创建时间等信息

场景2：查询某个App所有的广播状态

```
dumpsys activity b mobi.infolife.ezweather.locker.locker_2
```

场景3：查询某个App所有的Activity状态



```
dumpsys activity a mobi.infolife.ezweather.locker.locker_2
```

场景4：查询某个App的进程状态

```
dumpsys activity p mobi.infolife.ezweather.locker.locker_2
```

格式：ProcessRecord{HashCode pid:进程名/uid}，进程pid=941，进程名为mobi.infolife.ezweather.locker.locker_2:live，uid=10060。

该进程中还有Services，Connections，Providers，Receivers，

场景5：查询栈顶Activity

```
dumpsys activity | grep mFocusedActivity
```

dumpsys 的命令还有很多，这里就不一一列举了。

Android学习笔记 (nb/5644448)

举报文章 © 著作权归作者所有



吾若成疯 (/u/24c2a4e74009) ♂

写了 15230 字，被 31 人关注，获得了 134 个喜欢
(/u/24c2a4e74009)

+ 关注

你可以暂时不优秀，但不可以一天不进步

如果觉得我的文章对您有用，请随意赞赏。您的支持将鼓励我继续创作！

赞赏支持

♡ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) | 3



更多分享

(<http://cwb.assets.jianshu.io/notes/images/18224731/1weihn>)





登录 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-com)

评论

智慧如你，不想发表一点想法 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-nocomments-text)咩~

被以下专题收入，发现更多相似内容

-  Android开发 (/c/d1591c322c89?utm_source=desktop&utm_medium=notes-included-collection)
-  Android... (/c/5139d555c94d?utm_source=desktop&utm_medium=notes-included-collection)
-  Android开发 (/c/0dc880a2c73c?utm_source=desktop&utm_medium=notes-included-collection)
-  程序员首页投稿 (/c/89995286335f?utm_source=desktop&utm_medium=notes-included-collection)
-  首页投稿 (/c/bDHhpK?utm_source=desktop&utm_medium=notes-included-collection)
-  半栈工程师 (/c/383594e9265f?utm_source=desktop&utm_medium=notes-included-collection)
-  Android 知识 (/c/ea06bc795515?utm_source=desktop&utm_medium=notes-included-collection)

展开更多

推荐阅读 更多精彩内容 >

Java虚拟机内存分区和类加载过程 (/p/991a2d9a1fc4?utm_campaign=ma...)



为什么Java能够跨平台 是因为Java程序编译之后的代码不是能够被硬件系统直接运行的代码，而是一种“中间码”-字节码，然后不同的硬件平台上安装有不同的Java虚拟机（JVM），有JVM来把字节码再翻译成对应的硬件平台能够执行的代码，所以Java是跨平台的。Java虚拟机及内存分区 1、jvm简介 Java虚拟机（Java Virtual Machine 简称JVM）是运行所有Java程序的抽象计算机，是Java语言的运行环境。Java虚拟机有自己完善的硬体架构，如处理器、堆栈、寄存器等，还具有相应的指令系统。JVM屏蔽了与具体操作系统平台相关的信息，使得Java程序只需生成在J...

(/p/991a2d9a1fc4?utm_campaign=male:utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

吾若成疯 (/u/24c2a4e74009?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

Android自定义View实现八卦图效果 (/p/42c87137fdbb?utm_campaign=...

最近研究了下自定义View，就自己写了个Demo练练手。网上有很多讲自定义View实现的文章，我这里就不说了，直接上Demo。先看下效果 实现思路 1、左右两边分别画一个背景为黑色和白色的半圆。2、上下画两个小圆3、最后画鱼眼 实现步骤 1、设置自定义属性 2、定义画笔和其他属性 3、初始化画笔 4、画两个半圆效果如下 5、绘制两个鱼头 效果如下 6、绘制鱼眼 效果如下 7、让八卦转动起来 这样我们就通过自定义View实现了八卦图的功能。是不是很酷炫呀。完整代码 项目地址https://github.com/xingchenfengn/CustomBaGuaView.git

吾若成疯 (/u/24c2a4e74009?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

孤独几种 (/p/96469792e4f0?utm_campaign=maleski...

去看周杰伦的演唱会，坐在一个八百杆子都打不着的看台座位，依然激动得想把衣服扒了折成飞机。《晴天》钢琴版的前奏清澈地响起：从前从前，有个人爱你很久。我不知所措地把荧光棒放下，非常想跟谁打个电话，手机掏出来摇摇晃晃，没信号。那时我很疑惑，如果有信号我能打给谁呢。下扬州的时候差不多烟花三月，去了大名鼎鼎的食为天，心想吃碗扬州炒饭多应景。我端着菜单兴致勃勃地点了一份红烧鲢鱼头，服务员看了我一眼，笑意盈盈地说：先生不好意思，我们这道菜是两条鱼起做，您一个人恐怕吃不完。我呆了一会儿，说，哦。那是一间很大的餐厅，我所在的大厅在办婚宴，人们嬉笑着分食狮子头，点58元一份的扬州炒饭，大概可以盛八碗...

名贵的考拉熊 (/u/78f970537a5e?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

“别撩我了，就知道你想啪啪啪” (/p/3be4bd07ac1b?ut...

文/图/zhangxuyang 01 前几天，有一位读者朋友加了我的微信，向我恋爱方面的问题。她在微信上给我发了好长一段话：“有一个男生喜欢她很久了，自己要不要接受他？”这让她很烦恼。我问她：“如果你觉得对方人品没有问题的话，不妨可以考虑一下，再按照心里的想法看要不要接受他？”她说自己从小生活在单亲家庭，自己从小就渴望有一个人好好爱自己，母亲是一个很传统的妇女。虽然母亲将她带大，也非常不容易。但是经常和母亲在许多事情上意见不一致，存在隔阂。父亲在她儿时记忆中到现在长年酗酒，还喜欢赌钱。家里的经济情况不太好，收入来源于靠母亲在外面靠给人家缝补衣服和鞋维持到现在。现在自己也经常利用...

(/p/96469792e4f0?utm_campaign=male:utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

(/p/3be4bd07ac1b?utm_campaign=male:utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)



zhangxuyang (/u/2590c118fcb0?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

听了近六百首古风歌曲，推荐给你这八首。 (/p/46d0d562939a?utm_campaign=maleutm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

写在前面：喜欢古风的人，往往被周围人贴上中二症的标签。他人认为“夜晚城市的霓虹灯，星巴克的绿。”才是真实的，而那些整日里吟哦古调之人，不过是在附庸风雅，装腔作势，虚伪至极。“文艺青年”在不知不觉中也成为了讥讽的代名词。诚然，的确有伪文艺青年存在，往往读了几句酸诗便可以自诩孤傲，但这并不能成为你否认所有人的理由。 知乎上有一个问题：你为什么喜欢古风?我特别喜欢其中一个答主王二点的回复。 这群人所喜欢的，或许只是那段荡气回肠的情爱纠葛，是那段只能在梦回追忆，遥不可及的海蜃，是藏在心中爱之敬之的那片灿烂江湖。 所谓“大江东去，浪淘尽，千古风流人物”，所谓“杨柳岸，晓风残月，此去经年，应是良...

顾沉浮 (/u/eec714679ec1?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

 登录/注册

为你个性化推荐内容

(/sign_in?utm_source=desktop&utm_medium=note&utm_source=desktop&utm_medium=click-note-bottom-b

 下载简书App

随时随地发现和创作内容

(/apps/download?utm_source=desktop&utm_medium=click-note-bottom-b

