## 深入浅出 - Android系统移植与平台开发（四）－ Android启动流程

一、Android init进程启动

还是从Linux的启动开始吧。Linux被bootloader加载到了内存之后，开始运行，在初始化完Linux运行环境之后，挂载ramdisk.img根文件系统映像，运行里面的init程序，这也是Linux的第一个用户程序，其pid为1。下面的文章是作者关于init进程启动的描述。

<http://blog.csdn.net/mr_raptor/article/details/7666906>

二、Android本地服务的启动

 init进程启动完之后，开始初始化并启动Dalvik虚拟机，在Dalvik虚拟机启动之前做了一些工作，请看下面一篇文章。

<http://blog.csdn.net/mr_raptor/article/details/7804984>

Android启动总结：

init进程在执行过程中可以分为以下几个阶段：

Ø  启动准备：创建文件系统的基本目录、打开标准输入、标准输出、标准错误，初始化log日志功能等

Ø  解析init.rc和init.hardware.rc文件：将rc文件逐行解析成Action或Service。解析出来的Action和Service分别存放在action\_list和service\_list链表里，每个Action都对应一个或多个Commands，每个依附于Action的Commands也由一个链表维护。

Ø  将early-initAction添加到action\_queue队列里，等待执行

Ø  将init Action添加到action\_queue队列里，等待执行

Ø  添加其它条件的Action到action\_queue队列里

Ø  进入死循环

o    从action\_queue队列里依次取出每个Action，执行其维护的Commands链表里的命令

o    重新启动service\_list中标记为SVC\_RESTARTING服务

o    监听系统属性状态变化事件、子进程信号、Keychord组合按键事件

注：在代码里没有明显运行service\_list里服务的代码，每个服务都有一个class属性，该属性决定了服务的分类，在init.rc文件的on boot Action最后有两个命令：

1. on boot
2. …
3. ...
4. class\_start core
5. class\_start main

class\_start命令是指运行某一类的服务，先启动了class为core的服务，然后再启动了class为main的服务。

init.rc中class为core的服务有：

|  |  |
| --- | --- |
| Service名 | 对应程序及参数 |
| ueventd | /sbin/ueventd |
| console | /system/bin/sh |
| adbd | /sbin/adbd |
| servicemanager | /system/bin/servicemanager |
| vold | /system/bin/vold |

class为main的服务有：

|  |  |
| --- | --- |
| Service名 | 对应程序及参数 |
| netd | /system/bin/netd |
| debuggerd | /system/bin/debuggerd |
| ril-daemon | /system/bin/rild |
| surfaceflinger | /system/bin/surfaceflinger |
| zygote | /system/bin/app\_process -Xzygote /system/bin --zygote--start-system-server |
| drm | /system/bin/drmserver |
| media | /system/bin/mediaserver |
| bootanim | /system/bin/bootanimation |
| dbus | /system/bin/dbus-daemon --system --nofork |
| bluetoothd | /system/bin/bluetoothd -n |
| installd | /system/bin/installd |
| flash\_recovery | /system/etc/install-recovery.sh |
| racoon | /system/bin/racoon |
| mtpd | /system/bin/mtpd |
| keystore | /system/bin/keystore /data/misc/keystore |
| dumpstate | /system/bin/dumpstate -s |