**Android系统启动之init.rc文件解析过程**

image.png

**目录**

第一篇：[Android系统启动之bootloader](https://www.jianshu.com/p/75039ea24937)  
第二篇：[Android系统启动之Init流程(上)](https://www.jianshu.com/p/297135f6497d)  
第三篇：[Android系统启动之Init流程(下)](https://www.jianshu.com/p/8eeea0e6edbd)  
第四篇：[Android系统启动之init.rc文件解析过程](https://www.jianshu.com/p/a4c17f0110d0)  
第五篇：[Android系统启动之zyogte进程](https://www.jianshu.com/p/a718a49f3533)  
第六篇：[Android系统启动之zyogte进程java(上)](https://www.jianshu.com/p/de4f4abe7a05)  
第七篇：[Android系统启动之zyogte进程java(下)](https://www.jianshu.com/p/6bfd37fe9203)  
第八篇：[Android系统启动之SystemServer](https://www.jianshu.com/p/fe4f6523cf16)

**什么是init.rc**

Android init.rc文件由系统第一个启动的init程序解析。是启动系统服务使用的文件。

**rc规则**

主要包含了四种类型的语句:

1. Action
2. Commands
3. Services
4. Options.

Action和services显式声明了一个语句块，而commands和options属于最近声明的语句块。

在第一个语句块之前 的commands和options会被忽略.

基本规则如下：

在init.rc文件中一条语句通常是占据一行.  
单词之间是通过空格符来相隔的.  
如果需要在单词内使用空格，那么得使用转义字符"",如果在一行的末尾有一个反斜杠，那么是换行折叠符号，应该和下一行合并成一起来处理，与C语言中的含义是一致的。  
注释是以#号开头。

**关键字**

| **关键字** | **含义** |
| --- | --- |
| token | 计算机语言中的一个单词，就跟英文中的单词几乎相同一人概念. |
| Section | 语句块，相当于C语言中大括号内的一个块。一个Section以Service或On开头的语句块.以Service开头的Section叫做服务,而以On开头的叫做动作(Action). |
| services | 服务. |
| Action | 动作 |
| commands | 命令. |
| options | 选项. |
| trigger | 触发器。或者叫做触发条件. |
| class | 类属，即能够为多个service指定一个同样的类属，方便操作同一时候启动或停止. |

**动作(Action)**

动作表示了一组命令(commands)组成.动作包括一个触发器，决定了何时运行这个动作。

注意：**当触发器的条件满足时，这个动作会被增加到已被运行的队列尾。假设此动作在队列中已经存在，那么它将不会运行.**

一个动作所包括的命令将被依次运行。

**on <trigger> ## 触发条件**

**<command> ##执行命令**

**<command1> ##可以执行多个命令**

**触发器(trigger)**

在"动作"(action)里面的，on后面跟着的字符串是触发器(trigger)，trigger是一个用于匹配某种事件类型的字符串，它将对应的Action的执行。

触发器(trigger)有几种格式：

1、最简单的一种是一个单纯的字符串。比如“on boot”。这种简单的格式可以使用命令"trigger"来触发。  
2、还有一种常见的格式是"on property<属性>=<值>"。如果属性值在运行时设成了指定的值，则"块"(action)中的命令列表就会执行。

常见的格式：

| **格式** | **含义** |
| --- | --- |
| on early-init | 在初始化早期阶段触发 |
| on init | 在初始化阶段触发 |
| on late-init | 在初始化晚期阶段触发 |
| on boot/charger | 当系统启动/充电时触发 |
| on property | 当属性值满足条件时触发 |

**commands(命令)**

command是action的命令列表中的命令，或者是service中的选项 onrestart 的参数命令.

命令将在所属事件发生时被一个个地执行.

常见命令：

| **命令** | **描写叙述** |
| --- | --- |
| exec <path> [ <argument> ]\* | 运行指定路径下的程序，并传递參数. |
| export <name> <value> | 设置全局环境參数。此參数被设置后对全部进程都有效. |
| ifup <interface> | 使指定的网络接口"上线",相当激活指定的网络接口 |
| import <filename> | 导入一个额外的init配置文件. |
| hostname <name> | 设置主机名 |
| chdir <directory> | 改变工作文件夹. |
| chmod <octal-mode> <path> | 改变指定文件的读取权限. |
| chown <owner> <group> <path> | 改变指定文件的拥有都和组名的属性. |
| chroot <directory> | 改变进行的根文件夹. |
| class\_start <serviceclass> | 启动指定类属的全部服务，假设服务已经启动，则不再反复启动. |
| class\_stop <serviceclass> | 停止指定类属的所胡服务. |
| domainname <name> | 设置域名 |
| insmod <path> | 安装模块到指定路径. |
| mkdir <path> [mode] [owner] [group] | 用指定參数创建一个文件夹，在默认情况下，创建的文件夹读取权限为755.username为root,组名为root. |
| mount <type> <device> <dir> [ <mountoption> ]\* | 类似于linux的mount指令 |
| setkey TBD(To Be Determined), | 待定. |
| setprop <name> <value> | 设置属性及相应的值. |
| setrlimit <resource> <cur> <max> | 设置资源的rlimit(资源限制），不懂就百度一下rlimit |
| start <service> | 假设指定的服务未启动，则启动它. |
| stop <service> | 假设指定的服务当前正在执行。则停止它. |
| symlink <target> <path> | 创建一个符号链接. |
| sysclktz <mins\_west\_of\_gmt> | 设置系统基准时间. |
| trigger <event> Trigger an event. Used to queue an action from another action. | 这名话没有理解，望高手指点. |
| write <path> <string> [ <string> ]\* | 往指定的文件写字符串. |

**服务(services)**

服务是指那些须要在系统初始化时就启动或退出时自己主动重新启动的程序.

**service <name><pathname> [ <argument> ]\***

**<option>**

**<option>**

解释一下各个参数：

| **参数** | **含义** |
| --- | --- |
| <name> | 表示此服务的名称 |
| <pathname> | 此服务所在路径因为是可执行文件，所以一定有存储路径。 |
| <argument> | 启动服务所带的参数 |
| <option> | 对此服务的约束选项 |

**选项（option）**

options是Service的修订项。它们决定一个服务何时以及如何运行.

| **选项** | **描述** |
| --- | --- |
| critical | 据设备相关的关键服务，如果在4分钟内，此服务重复启动了4次，那么设备将会重启进入还原模式。 |
| disabled | 服务不会自动运行，必须显式地通过服务器来启动。 |
| setenv | 设置环境变量 |
| socket [ [ ] ] | 在/dev/socket/下创建一个unix domain的socket，并传递创建的文件描述符fd给服务进程.其中type必须为dgram或stream,seqpacket. | 用户名和组名默认为0 |
| user | 在执行此服务之前先切换用户名。当前默认为root. |  |
| group [ ]\* | 类似于user,切换组名 |  |
| oneshot | 当此服务退出时不会自动重启. |  |
| class | 给服务指定一个类属,这样方便操作多个服务同时启动或停止.默认情况下为default. |  |
| onrestart | 当服务重启时执行一条指令， |  |

使用例子：

**service bootanim /system/bin/bootanimation**

**class core //给服务指定一个类属,这样方便操作多个服务同时启动或停止**

**user graphics //在执行此服务之前先切换用户名**

**group graphics audio**

**disabled //服务不会自动运行**

**oneshot //当此服务退出时不会自动重启**

**rc文件实例**

**# not complete -- just providing some examples of usage**

**#**

**on boot**

**export PATH /sbin:/system/sbin:/system/bin**

**export LD\_LIBRARY\_PATH /system/lib**

**mkdir /dev**

**mkdir /proc**

**mkdir /sys**

**mount tmpfs tmpfs /dev**

**mkdir /dev/pts**

**mkdir /dev/socket**

**mount devpts devpts /dev/pts**

**mount proc proc /proc**

**mount sysfs sysfs /sys**

**write /proc/cpu/alignment 4**

**ifup lo**

**hostname localhost**

**domainname localhost**

**mount yaffs2 mtd@system /system**

**mount yaffs2 mtd@userdata /data**

**import /system/etc/init.conf**

**class\_start default**

**service adbd /sbin/adbd**

**user adb**

**group adb**

**service usbd /system/bin/usbd -r**

**user usbd**

**group usbd**

**socket usbd 666**

**service zygote /system/bin/app\_process -Xzygote /system/bin --zygote**

**socket zygote 666**

**service runtime /system/bin/runtime**

**user system**

**group system**

**on device-added-/dev/compass**

**start akmd**

**on device-removed-/dev/compass**

**stop akmd**

**service akmd /sbin/akmd**

**disabled**

**user akmd**

**group akmd**

**rc文件解析**

源码路径system/core/init/init.cpp中：

**parser.ParseConfig("/init.rc");**

开始解析rc文件．

ParseConfig函数在文件core/init/init\_parser.cpp140行：

**bool Parser::ParseConfig(const std::string& path) {**

**if (is\_dir(path.c\_str())) {**

**return ParseConfigDir(path);**

**}**

**return ParseConfigFile(path);**

**}**

ParseConfigFile函数：

**bool Parser::ParseConfigFile(const std::string& path) {**

**LOG(INFO) << "Parsing file " << path << "...";**

**Timer t;**

**std::string data;**

**if (!read\_file(path, &data)) {**

**return false;**

**}**

**data.push\_back('\n'); // TODO: fix parse\_config.**

**ParseData(path, data);**

**for (const auto& sp : section\_parsers\_) {**

**sp.second->EndFile(path);**

**}**

**LOG(VERBOSE) << "(Parsing " << path << " took " << t << ".)";**

**return true;**

**}**

**参考**

[Android init.rc文件浅析](https://links.jianshu.com/go?to=https%3A%2F%2Fwww.cnblogs.com%2Flxjshuju%2Fp%2F6915468.html)  
[安卓系统启动--3init.rc解析](https://www.jianshu.com/p/cb73a88b0eed)  
[init.rc深入学习](https://links.jianshu.com/go?to=https%3A%2F%2Fblog.csdn.net%2Fch853199769%2Farticle%2Fdetails%2F78427362)