

华清远见

FARIGHT[®]
始于 2004

嵌入式培训专家

在线大讲堂 – Android系统启动及系统优化

唐老师，北京总部

华清远见全国免费咨询电话：400-706-1880

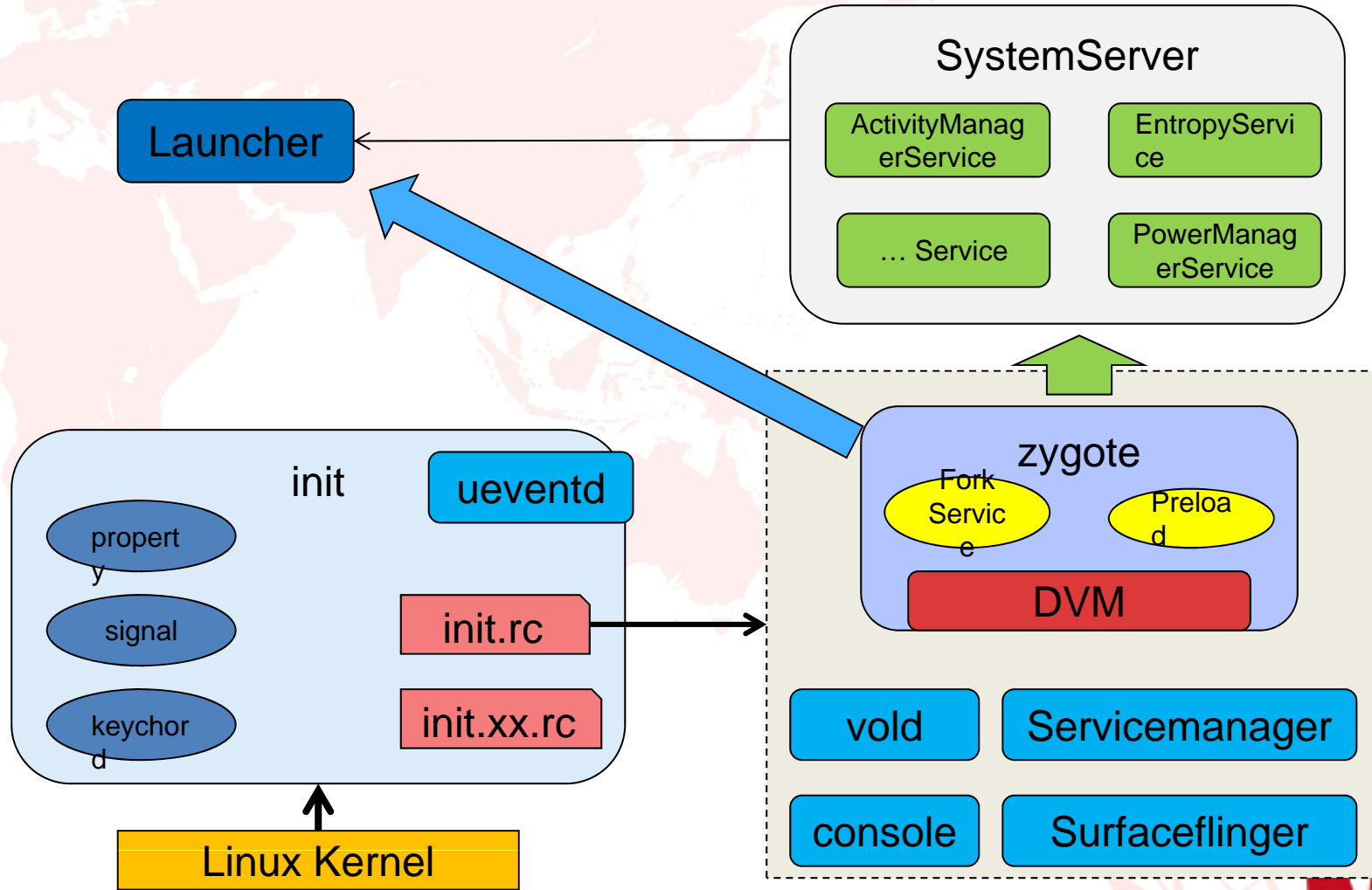
北京总部咨询电话：
010-82600385/6

专业始于专注 卓识源于远见

主题

- Android系统启动分析
 - Android系统的启动流程分析
 - init进程启动及Native服务启动
 - Zygote进程及SystemService服务进程启动
 - Home桌面启动
- 定制Android系统
 - 定制Boot Logo界面
 - 定制手机开机界面
 - 定制Android动画
 - 定制Android系统桌面

Android系统的启动流程分析



init进程启动及Native服务启动

- init进程，它是一个由内核启动的用户级进程。内核自行启动（已经被载入内存，开始运行，并已初始化所有的设备驱动程序和数据结构等）之后，就通过启动一个用户级程序init的方式，完成引导进程。init始终是第一个进程。
- Init进程一起来就根据init.rc和init.xxx.rc脚本文件建立了几个基本的服务：
 - servicemanamger
 - zygote
 - ...
- 最后Init并不退出，而是担当起service的功能,用来监听sigal, keychod和property。

Android Init Language

- 官方说明: `system/core/init/readme.txt`
- 包含四个类型的声明:
 - Actions
 - Commands
 - Services
 - Options.
- 特点:
 - 所有这些都是以行为单位的, 各种记号由空格来隔开。
 - 行末的反斜杠用于折行, 注释行以井号 (#) 开头 (允许以空格开头)。
 - Actions和Services声明一个新的分组Section。所有的命令或选项都属于最近声明的分组。位于第一个分组之前的命令或选项将会被忽略
 - Actions和Services有唯一的名字。如果有重名的情况, 第二个申明的将会被作为错误忽略。

Android Init Language (cont.)

- **Actions**

- Actions其实就是一序列的Commands（命令）。Actions都有一个trigger（触发器），它被用于决定action的执行时间。当一个符合action触发条件的事件发生时，action会被加入到执行队列的末尾，除非它已经在队列里了。
- 队列中的每一个action都被依次提取出，而这个action中的每个command（命令）都将被依次执行。

Actions的形式如下：

```
on <trigger/name>  
  <command1>  
  <command2>  
  <command3>
```

- 当trigger被触发时，command1，command2，command3，会依次执行，直到下一个Action或下一个Service。
简单来说，Actions就是Android在启动时定义的一个启动脚本，当条件满足时，会执行该脚本，脚本里都是一些命令commands，不同的脚本用on来区分。

Android Init Language (cont.)

- 常用的**Commands**

- exec <path> [<argument>]*
- export <name> <value>
- ifup <interface>
- hostname <name>
- chmod <octal-mode> <path>
- chown <owner> <group> <path>
- class_start <serviceclass>
- class_stop <serviceclass>
- insmod <path>
- mkdir <path> [mode] [owner] [group]
- mount <type> <device> <dir> [<mountoption>]*
- setprop <name> <value>
- start/stop <service>
- write <path> <string> [<string>]*

Android Init Language (cont.)

- **Services**

Services（服务）是一个程序，它在初始化时启动，并在退出时可选择让其重启。

Services（服务）的形式如下：

```
service <name> <pathname> [ <argument> ]*  
    <option>  
    <option>
```

name:服务名

pathname:当前服务对应的程序位置

option: 当前服务设置的选项

- **Options**

Options（选项）是一个Services（服务）的修正者。他们影响Services（服务）在何时，并以何种方式运行。

critical

disabled

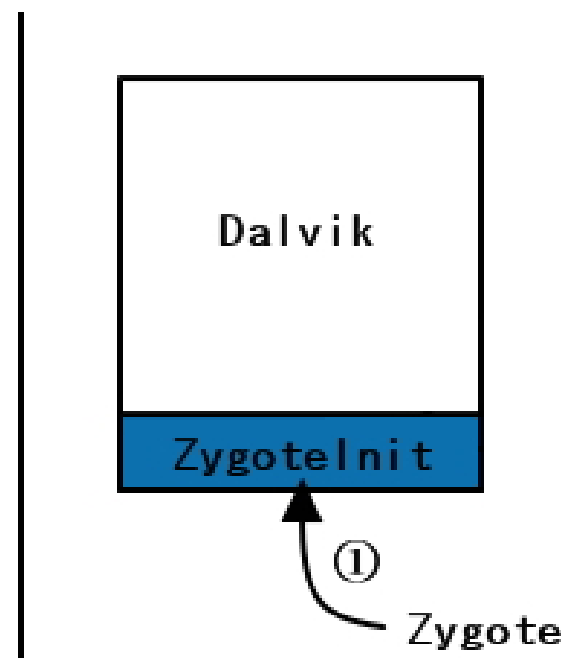
setenv <name> <value>

socket <name> <type> <perm> [<user> [<group>]]

class <name>

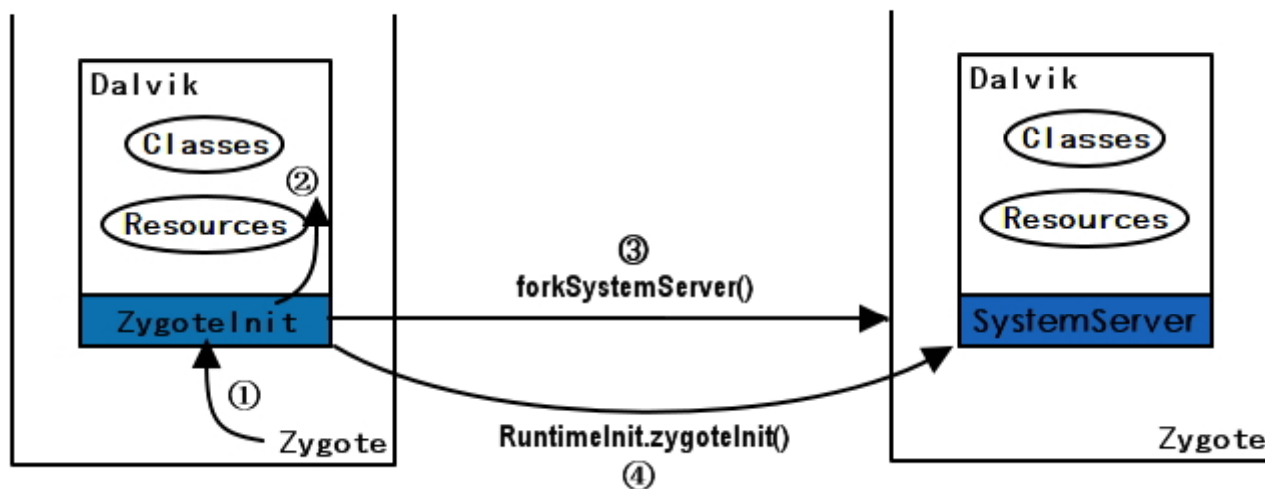
Zygote进程的启动

- Zygote进程是整个Android世界的创建者，在Zygote进程的启动过程中，开启了Dalvik虚拟机，对Android的启动做初始化。
- Zygote进程的创建
 - `service zygote /system/bin/app_process -Xzygote /system/bin --zygote --start-system-server`
- DVM的开启
 - `startVm()`
- 本地DVM初始化
 - `startReg()`
- 第一个Java代码的执行
 - `Zygotelnit.main()`



ZygoteInit

- ZygoteInit作为Android系统中第一段运行的Java代码，对Android系统运行有着以下重要的意义：
 - 开启ZygoteForkService Socket
 - 预加载类和资源用于加速应用程序启动
 - 监听ZygoteForkService Socket创建并启动新的Android应用程序
 - 启动了SystemServer



SystemServer进程的启动

- SystemServer 是Android服务守护进程，里面运行了大量的Android系统服务，是整个Android系统的服务提供者。
- SystemServer进程的创建
 - Zygote.forkSystemServer
- 本地系统服务的启动
 - 通过JNI在本地代码中启动必要本地系统服务
- ServerThread
 - 用于创建并注册Android系统服务
- 启动桌面应用程序Launcher

桌面的启动

- 在ServerThread@SystemService.java后半段，我们可以看到系统在启动完所有的Android服务后，做了这样一些动作：
 - 1. 使用xxx.systemReady()通知各个服务，系统已经就绪。
 - 2. 特别对于ActivityManagerService.systemReady(回调)
Widget.wallpaper,imm(输入法)等ready通知。Home就是在ActivityManagerService.systemReady()通知的过程中建立的。
下面是ActivityManagerService.systemReady()的伪代码：
 - systemReady()@ActivityManagerService.java
 - resumeTopActivityLocked()
 - startHomeActivityLocked();//如果是第一个则启动HomeActivity。
 - startActivityLocked (。 。 。) CATEGORY_HOME

定制Android系统

- 在Android设备开机过程中，屏幕上可能会显示4种形式内容：
 - 在硬件Boot时出现Boot Logo画面
 - Linux启动时画面，可能会显示Linux Logo
 - Android系统本地启动阶段画面，可能显示“ANDROID”文字字样或定制画面
 - Android系统显示系统启动阶段，可以显示滚动字体或逐帧动画



定制开机Logo(fspad示例)

- 定制Boot Logo画面：
 - 替换Lichee/tools/pack/chips/sun5i/wboot/bootfs/linux/linux.bmp文件
 - linux.bmp文件：120x120，位度为32位
 - 重新打包镜像:pack
 - 烧写bootloader分区：
 - lichee/tools/pack/out/bootloader.fex



定制Android启动图片

- Linux系统里的第一个用户进程init由system/core/init/目录下的源码编译而成，其入口文件为：init.c，console_init_action函数就是用来在屏幕上打印一张静态图片或打印ANDROID字样的，如果想定制启动图片或字样，则直接将其内容修改了，重新编译init程序，然后重新生成ramdisk.img即可。
 - 通过分析console_init_action可知，修改/initlogo.rle图片文件即可定制启动图片
 - 修改console_init_action中代码直接修改字样

定制Android启动图片

- 1. 修改启动图片
 - 通过convert工具将和屏幕大小一致的png格式转化成raw文件，然后通过Android Host Tools中的rgb2565将raw文件转化成initlogo.rle文件
 - 将新生成的initlogo.rle文件拷贝到ramdisk.img中，重新刷机即可。
- 2. 修改启动字样
 - 将console_init_action中字符串字样直接修改，重新编译init，生成ramdisk.img，重新刷机即可。

定制Android动画

- Android的开机动画是由Linux本地程序bootanimation控制实现的，其代码在：`frameworks/base/cmds/bootanimation/`，通过分析源码可知，修改Android开机动画有两种方式：
 - 替换`frameworks/base/core/res/assets/images/`目录下的两个图片文件：`android-logo-mask.png`和`android-logo-shine.png`
 - 在`/data/local/`或`/system/media/`目录创建`bootanimation.zip`文件，里面存放的是一帧帧的动画，通过一个脚本来控制播放



定制Android动画

- bootanimation.zip文件内容：
 - desc.txt 动画属性描述文件
 - part0/ 第一阶段动画图片的目录
 - part1/ 第二阶段动画图片的目录
- desc.txt文件的格式为：
 - 480 250 15
 - p 1 0 part0
 - p 0 10 part1



Android系统企业级定制

- 经常会有客户说：“我们不想让我们的产品开机后看到Android系统的桌面，我们希望设备开机会直接进入我们自己的程序”。
- 解决方案：
 - 接收系统启动广播：BOOT_COMPLETED，在广播接收器代码里实现应用程序的启动
 - 编写一个桌面程序，替换掉系统默认的Launcher应用

Android系统企业级定制

- 桌面启动原理

- 在所有的Android服务启动完毕后，调用 `ActivityMangagerService.SystemReady()` 函数，该方法会发送一个category为 `android.intent.category.HOME` 的Intent对象。默认系统的Launcher桌面会响应这个Intent。，那么，我们可以编写一个应用程序，让它去响应这个Intent，然后将这个应用程序替换掉系统默认的Launcher桌面。

华清远见

FARIGHT 嵌入式培训专家

始于 2004

谢谢!