MTK Android软件培训

内容提要

- Android编译、调试环境搭建
- MTK Android编译、下载、启动
- Android 软件系统软件架构
- Android软件系统内存分配
- 总结 & Q/A

Android编译、调试环境搭建

- Android编译环境搭建
  - 系统编译环境搭建
  - MMI应用层开发环境搭建
- Android调试环境搭建
  - 系统程序调试环境搭建
  - MMI应用程序调试环境搭建

Android系统编译环境搭建

- 操作系统
  - Linux ubuntu (64bits) 10.04
  - Windows不支持
- 内存和磁盘空间
  - 至少2g物理内存
  - 至少10G物理磁盘空间
- Make
  - Make v3.81
- Perl, python 解释器
  - Perl 5.10.x
  - Python 2.6.x

Android 2.3系统编译环境搭建

- 编译器
  - Arm-eabi-4.4.3
  - Gcc 4.4.3
  - Jdk 1.6
- 其他工具
  - Wine 1.1
  - Bison 2.4.x, flex 2.5.x,gperf 3.0.x
  - Mingw32
  - Unix2dos/tofrodos

MMI应用层开发环境搭建

- Eclipse + Android sdk
  - 下载Java JDK 1.5版本以上
  - 下载Eclipse 3.72版本
  - 下载Android SDK，可根据开发需要下载所需要版本的sdk组件
  - 安装完这些基础软件后，进入eclipse中下载ADT，ADT是eclipse的Android支持包
  - 创建AVD（Android Virtual Device），就可以启动Android模拟器了
  - 在Eclipse中编译完应用程序后，安装到AVD中进行调试

系统程序调试环境搭建

- 系统调试环境主要在Linux下建立比较方便
  - 使用gdb命令行和带有图形界面的insight
  - 把android系统编译成debug模式
  - 通过adb shell启动shell找到要调试的进程
  - 使用gdbserver :5050 –-attach pid
  - Adb forward tcp:5050 tcp:5050
  - 在linux下使用gdb prog
  - 载入符号库和路径名就可以开始进行命令行调试了
  - Insight是图形程序终端，也可以用类似的命令

MMI应用程序调试环境搭建

- MMI应用程序Java调试主要使用Eclipse+ddms，在eclipse里面可以设置断点，监视变量值，查看全局变量，在ddms中可以查看打印的变量值和trace

MTK Android编译

- makeMtk
  - 这个命令为mtk封装的编译命令，它主要用来确定项目名，平台，以及要进行的动作。当用户传递的参数被该脚本通过后，系统执行mtk/make/Makefile.yusu来完成动作。

MTK Android编译

- makeMtk用法

```
Usage: (makeMtk|mk) [options] project actions [modules]
Options:
  -t, -tee      : Print log information on the standard-out. -o, -
    opt=bypass_argument_to_make
                : Pass extra arguments to make.
  -h, -help     : Print this message and exit
Projects:
 one of available projects.
Actions:
listp, listproject  : List all available projects.
check-env     : Check if build environment is ready.
check-dep     : Check feature dependency.
n, new        : Clean and perform a full build.
c, clean      : Clean the immediate files(such as, objects, libraries etc.).
r, remake     : Rebuild(target will be updated if any dependency updats).
bm_new        : "new" + GNU make's "-k"(keep going when encounter error)
    feature.
bm_remake     : "remake" + GNU make's "-k"(keep going when encounter
    error) feature.
mm            : Build module through Android native command "mm"
```

makeMtk用法

emigen : Generate EMI setting source code.
nandgen : Generate supported NAND flash device list.
codegen : Generate trace DB(for META/Cather etc. tools used).
drvgen : Generate driver customization source.
custgen : Generate customization source.
javaoptgen : Generate the global java options.
ptgen : Generate partition setting header & scatter file.
sign-image : Sign all the image generated.
encrypt-image : Encrypt all the image generated.
update-api : Android default build action
(be executed if system setting or anything removed from API).
check-modem : Check modem image consistency.
upadte-modem : Update modem image located in system.img.
modem-info : Show modem version
gen-relkey : Generate releasekey for application signing.
check-appres : Check unused application resource.
sdk : Build sdk package.
win_sdk : Build sdk package with a few Windows tools.
banyan_addon : Build MTK sdk addon.
cts : Build cts package.

makeMtk用法

bootimage : Build bootimage.
cacheimage : Build cacheimge.
systemimage : Build systemimage.
recoveryimage : Build recoveryimage.
secroimage : Build secroimge.
factoryimage : Build factoryimage.
userdataimage : Build userdataimage.
Modules:
pl, preloader : Specify to build preloader.
ub, uboot : Specify to build uboot.
k, kernel : Specify to build kernel.
dr, android : Specify to build android.
NULL : Specify to build all components/modules in default.
k <module path>
: Specify to build kernel component/module with the source path.
dr <module name>
: Specify to build android component/module with module name.

makeMtk用法例子

- Make /Build 一个全新的ALPS项目，清除所有旧的目标文件，库和Log文件
  - ./makeMtk ginwave75_gb2 new
- 重新制作uboot,kernel...目标库和临时文件
  - ./makeMtk ginwave75_gb2 remake uboot kernel
- 生成系统镜像
  - ./makeMtk ginwave75_gb2 systemimage

makeMtk用法例子

- 使用android原始函数和子程序，编译AlarmClock程序包
  - $ source build/envsetup.sh
  - $ cd packages/apps/AlarmClock
  - $ TARGET_PRODUCT=ginwave75_gb2 mm

如何编译一个新的apk

- Android.mk

```
LOCAL_PATH := $(call my-dir)
Include $(CLEAR_VARS)
# build all java files in the java subdirectory
LOCAL_SRC_FILES :=$(call all-subdir-java-files)
LOCAL_PACKAGE_NAME := LocalPackage
Include $(BUILD_PACKAGE)
```

编译一个依赖于静态库的APK

- Android.mk

```
LOCAL_PATH := $(call my-dir)
Include $(CLEAR_VARS)
LOCAL_STATIC_JAVA_LIBRARIES := static-
  library
# build all java files in the java subdirectory
LOCAL_SRC_FILES :=$(call all-subdir-java-files)
LOCAL_PACKAGE_NAME := LocalPackage
Include $(BUILD_PACKAGE)
```

用平台key来签名应用

- Android.mk

```
LOCAL_PATH := $(call my-dir)
Include $(CLEAR_VARS)
# build all java files in the java subdirectory
LOCAL_SRC_FILES :=$(call all-subdir-java-files)
LOCAL_PACKAGE_NAME := LocalPackage
LOCAL_CERTIFICATE := platform
Include $(BUILD_PACKAGE)
```

Android.mk变量解释

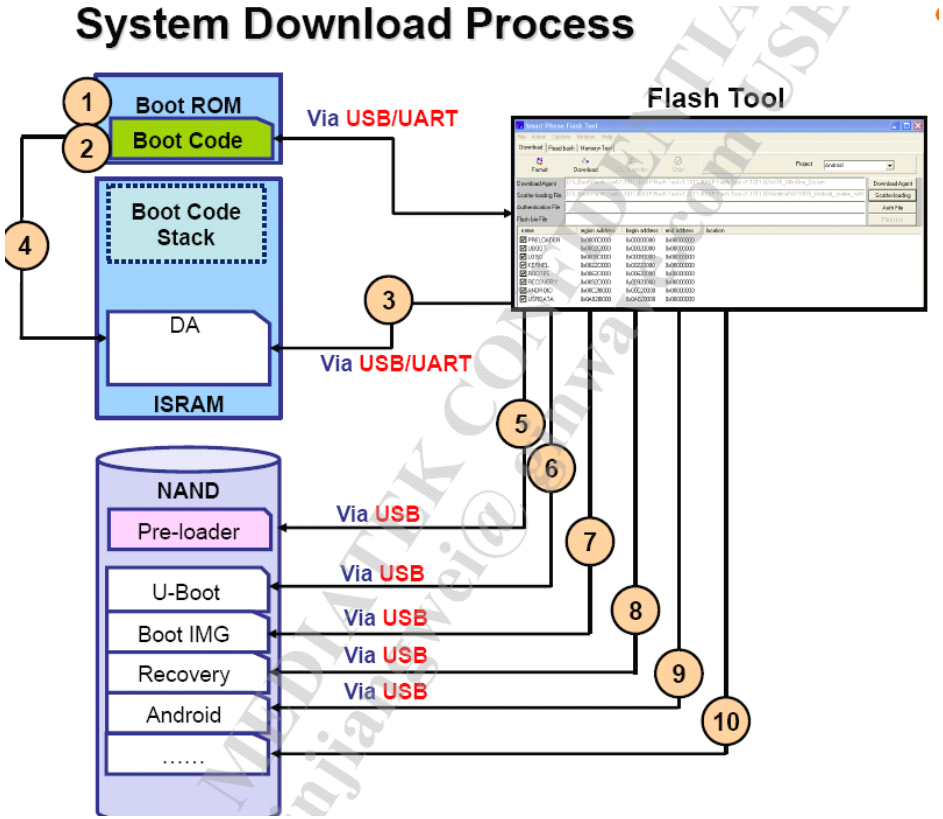
- LOCAL_
  - 以LOCAL开头的变量为局部变量，仅在模块内使用
- PRIVATE_
  - 以PRIVATE_开头的变量为目标项目特定的变量
- HOST_和TARGET_
  - 本地和目标板编译相关的变量
- BUILD_ 和CLEAR_VARS
  - 预先写好的编译模版

MTK软件下载

- Download Agent
  - DA为运行于目标板上的软件，用来执行下载请求
- Scatter-loading 文件
  - 用来描述下载到Nand Flash中的分区地址

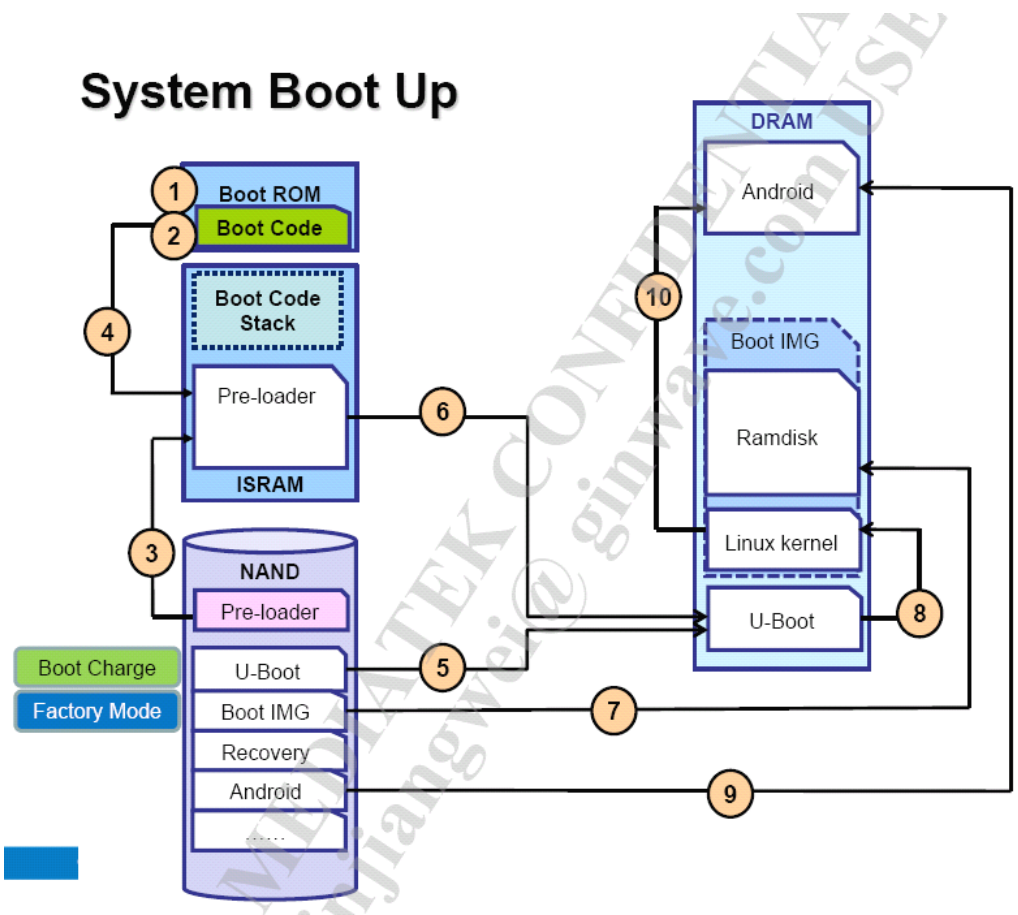
Mtk软件下载

- 系统下载过程



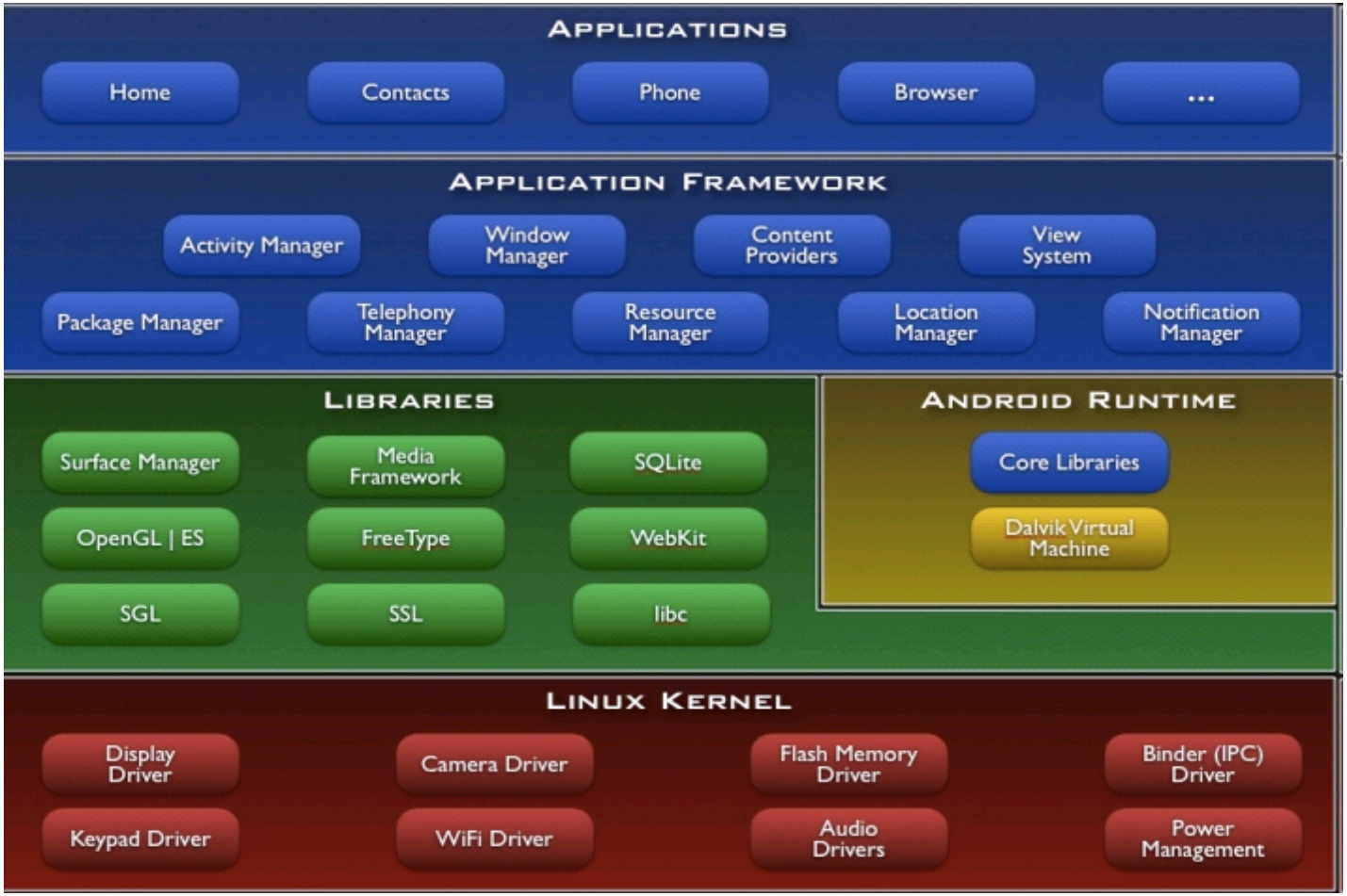
Mtk Android启动过程

- MTK 系统启动过程



Android软件系统架构

- 系统架构图



Android软件系统架构

- Application，应用层
  - 应用层用Java语言开发，包括手机应用的一系列核心程序，如电话本，电话，日历，邮件客户端等。

Android软件系统架构

- Application Framework
  - 视图支持，views
    - 各种视图类，如list, grid, text box, button等，用来构建应用的基础
  - 内容提供，content provider
    - 用来封装数据访问或者不同应用之间共享数据
  - 资源管理，resource manager
    - 用来访问字符串，布局等资源
  - 通知栏管理，notification manager
    - 用来在通知栏中显示提示
  - 活动管理，activity manager
    - 管理应用程序的生命周期和活动切换的堆栈管理

Android软件系统架构

- Libraries
  - 系统C库
    - 标准C库,基于BSD的一个标准实现
  - 媒体库
    - 基于opencore，可以用来支持主流图像显示，音频播放。
  - 层管理
    - 管理多个应用访问显示子系统，2d/3d之间无缝混合
  - Web浏览器引擎
    - 现代web浏览器引擎
  - SGL
    - 2D图形引擎
  - 3D 库
    - Opengl es 的一个3d实现
  - Freetype字体显示
    - 矢量和位图字体显示
  - SQLite数据库
    - 轻型关系型数据库管理系统

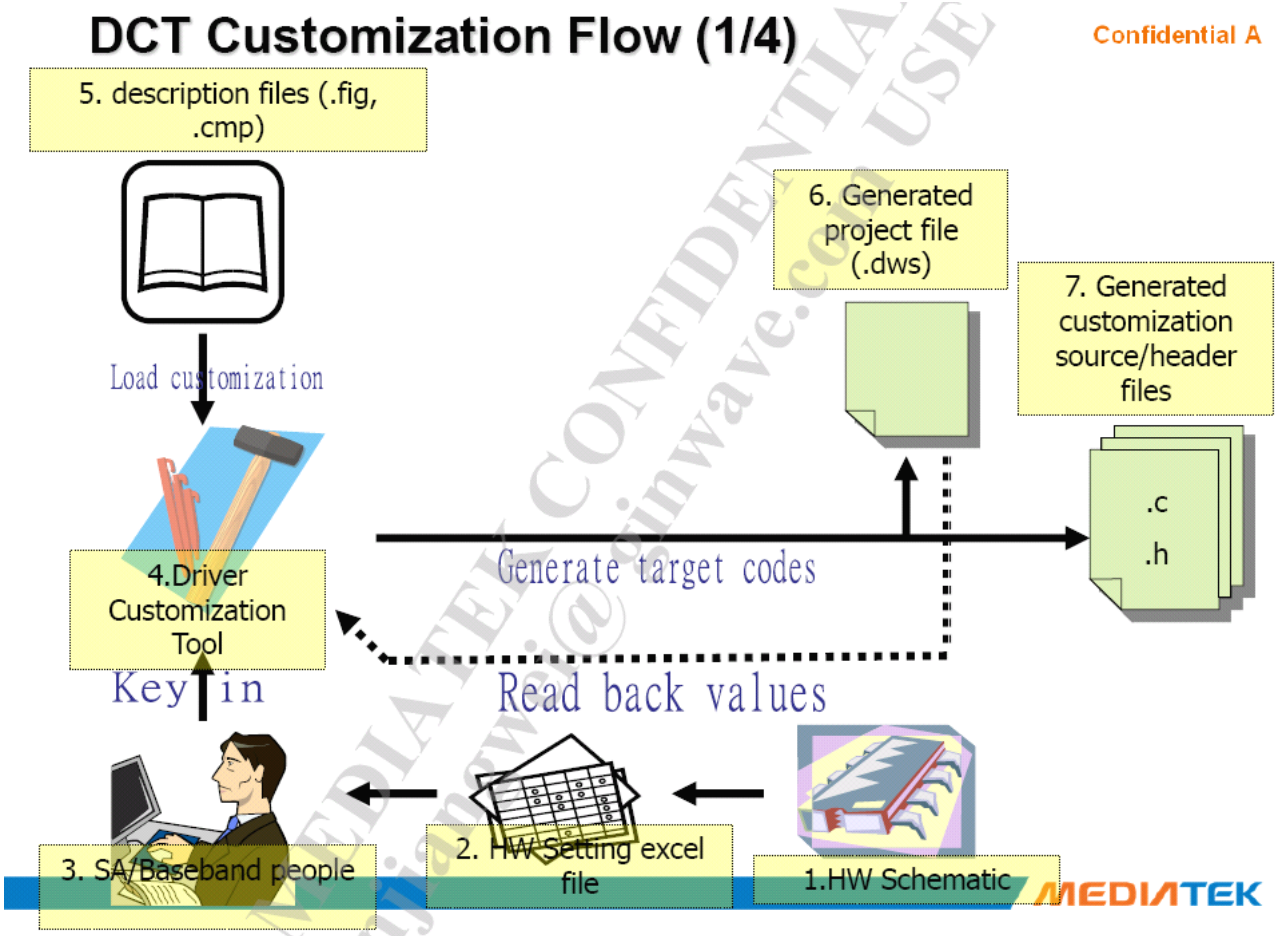
Android软件系统架构

- Android Runtime
  - Java 核心库
  - Dalvik Java虚拟机
- Linux kernel
  - 提供操作系统的核心功能，如进程管理，内存管理，安全管理，网络协议栈.

MTK驱动层介绍

- Device custom tool(DCT)
  - MTK使用DCT来自动生成GPIO, EINT,按键信息的C源码。

DCT 定制流程

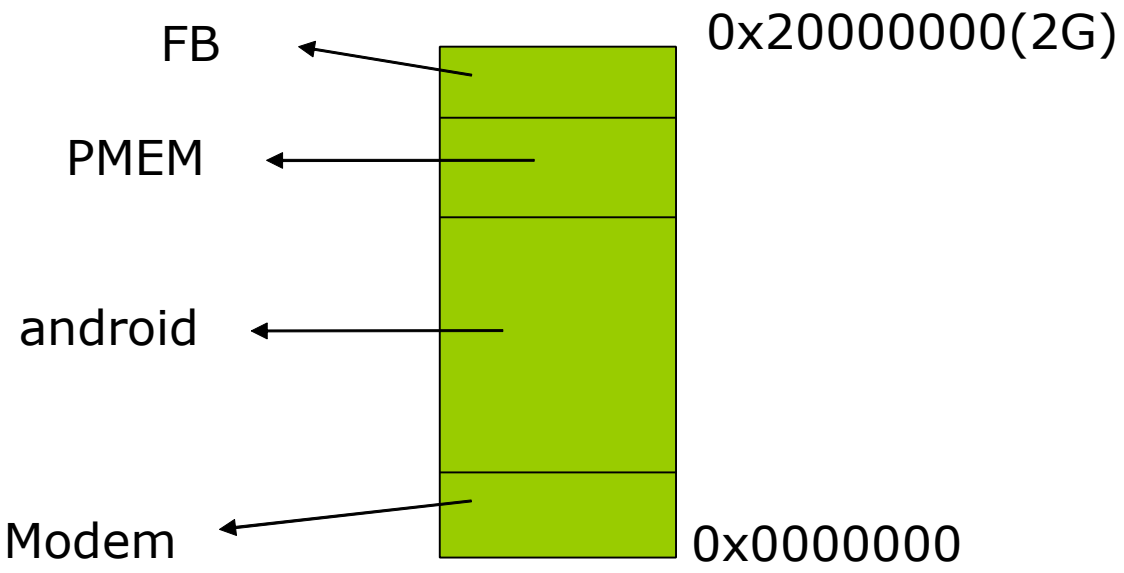

DCT Customization Flow (1/4)
Confidential A
5. description files (.fig, .cmp)
Load customization
6. Generated project file (.dws)
7. Generated customization source/header files
.c
.h
4.Driver Customization Tool
Generate target codes
Key in
Read back values
3. SA/Baseband people
2. HW Setting excel file
1.HW Schematic
MEDIATEK

MTK驱动层介绍

- Lights system
- Touch Panel
- LCM
- Sensor System
  - G-sensor
  - M-sensor
  - ALS/PS
- Connectivity
- Power Manager
- Audio

Android软件系统内存分配

- 内存分配示意图



Android软件系统内存分配

- FB, Framebuffer显示缓冲区
  - 这个尺寸和显示分辨率有关系，WVGA为4MB，HVGA为2MB
- PMEM，多媒体缓冲区，2g的物理内存配置中分配17MB
- Modem，22MB，Modem程序运行的内存
- Android，剩余的为Android使用的内存，2G物理内存使用202MB

LOWMEMKILLER

- 在系统内存空闲的情况下，尽量把多的程序缓冲在内存中，以提高程序的启动速度；如果系统内存不够，则Android系统会把内存中的进程杀掉，释放内存。
  - 杀内存的策略，android中程序按照优先级不同，可以分为若干类，优先级从0~15，0最高，15最低，0定义为前台进程，15定义为空进程
  - 对于系统定义的类型优先级，给定一个空闲内存阀值，当空闲内存少于阀值时，则比该类型优先级低的进程被杀掉，内存被释放

Android某个版本进程分类

进程	优先级	解释
foreground	0	前台进程，正在使用的进程
Visible	1	用户可见程序
Second serve	2	后台服务，如QQ后台程序
Home app	4	主界面
Hidden app	7	隐藏的程序
Content prov	14	内容提供者
empty	15	空程序，既不提供服务，也不提供内容

优先级内存警戒值

优先级	内存警戒值(4K)	动作
0	2048	
1	3072	
2	4096	
4	6144	
7	7168	当内存小于28M，杀掉优先级小于7的进程
15	8192	当内存小于32M，杀掉优先级小于15的进程

总结 & Q/A

- Android系统是一个比较复杂的系统，从底层到应用层所使用的编程语言语言分别为C/C++/Java，使用的程序设计方法分别为结构化程序设计/面向对象程序设计，在构建整个系统的过程中使用的makefile语法晦涩难懂，因此要想搞清楚的整个系统的完整运行，比较困难。但是复杂的系统都是分层的，我们从各层逐层分析，并理解各个层次之间的接口，就能很好的理解整个系统构成。
- Q/A

谢 谢!