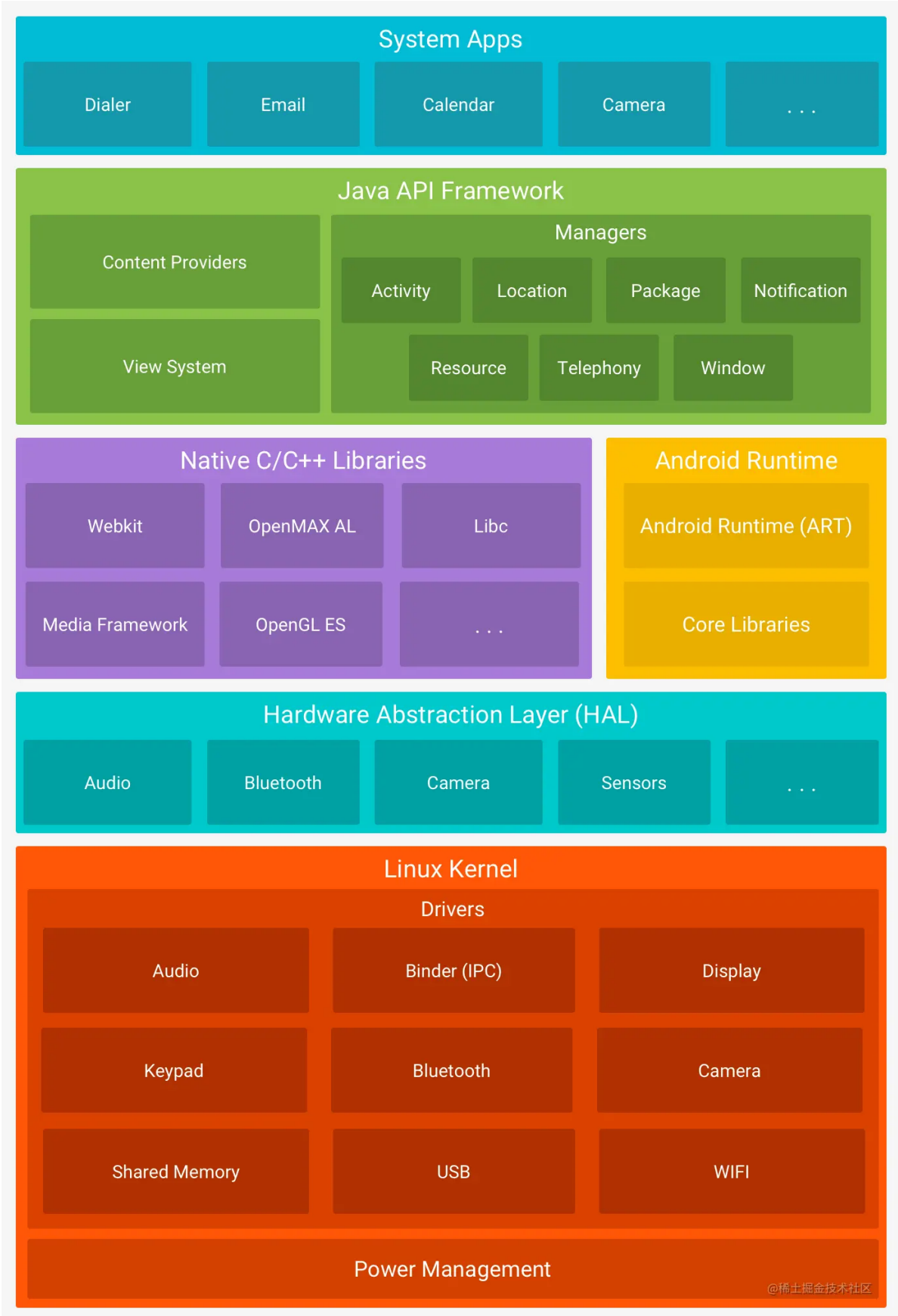
一、Android 架构简介

1.1 总体架构

Android系统架构分为五层，从上到下依次是应用层、应用框架层、系统运行库层、硬件抽象层和Linux内核层。


System Apps
Dialer
Email
Calendar
Camera
. . .
Java API Framework
Content Providers
View System
Managers
Activity
Location
Package
Notification
Resource
Telephony
Window
Native C/C++ Libraries
Webkit
OpenMAX AL
Libc
Media Framework
OpenGL ES
. . .
Android Runtime
Android Runtime (ART)
Core Libraries
Hardware Abstraction Layer (HAL)
Audio
Bluetooth
Camera
Sensors
. . .
Linux Kernel
Drivers
Audio
Binder (IPC)
Display
Keypad
Bluetooth
Camera
Shared Memory
USB
WIFI
Power Management
@稀土掘金技术社区


Android系统源码
（Android9.0.0）
Android.bp
Android7.0开始代替Android.mk文件，它是告诉ndk将jni代码编译生成动态库的一个编译脚本
art/
全新的ART运行环境
bionic/
Android的C Library，通俗地说，就是NDK的API
bootable/
启动引导相关代码
bootstrap.bash
build/
存放系统编译规则及generic等基础开发包配置，主要是一些脚本和工具
compatibility/
Android兼容性计划
cts/
Compatibility Test Suite的缩写，Android兼容性测试套件标准
dalvik/
dalvik虚拟机
development
应用程序开发相关
device/
设备相关的配置
external/
Android使用的一些开源的模组相关文件
frameworks/
应用程序核心框架，Android系统核心部分，由Java及C++编写
hardware/
部分厂家开源的硬件适配层HAL代码
kernel/
Android系统内核
libcore/
核心库相关文件
libnativehelper/
动态库，实现JNI库的基础
Makefile
全局Makefile文件，用来定义编译规则
out/
编译完成后的内容输出于此目录
packages/
应用程序包
pdk/
Plug Development Kit的缩写，本地开发套件
platform_testing/
平台测试
prebuilts/
x86和arm架构下预编译的一些资源
sdk/
adk及模拟器
system/
底层文件系统库、应用及组件——C语言

Android源码根目录	描述
abi	abi相关代码，abi:application binary interface，应用程序二进制接口
art	全新的运行环境，需要和Dalvik VM区分开来
bionic	系统C库
bootable	启动引导相关代码
build	存放系统编译规则及generic等基础开发配置包
cts	Android兼容性测试套件标准
dalvik	dalvik虚拟机
developers	开发者目录
development	应用程序开发相关
device	设备相关配置
docs	参考文档目录
external	Android使用的一些开源的模组
frameworks	应用程序框架，Android系统核心部分，由Java和C++编写
hardware	主要是硬件适配层HAL代码
libcore	核心库相关文件
libnativehelper	是Support functions for Android's class libraries的缩写，表示动态库，是实现的JNI库的基础
ndk	Android NDK (Android Native Development Kit)是一系列的开发工具，允许程序开发人员在Android应用程序中嵌入C/C++语言编写的非托管代码
out	编译完成后代码输出在此目录
packages	应用程序包
pdk	Plug Development Kit 的缩写，本地开发套件
platform_testing	平台测试
prebuilts	x86和arm架构下预编译的一些资源
sdk	sdk和模拟器
system	文件系统和应用及组件，是用C语言实现的
toolchain	工具链文件
tools	工具文件
Makefile	全局Makefile文件，用来定义编译规则 (通常文件后缀为.mk)

1.2 应用层

应用层位于整个Android系统的最上层，开发者开发的应用程序以及系统内置的应用程序都是在应用层。源码根目录中的*packages*目录对应着系统应用层。

packages目录	描述
apps	核心应用程序
experimental	第三方应用程序
inputmethods	输入法目录
providers	内容提供者目录
screensavers	屏幕保护
services	通信服务
wallpapers	墙纸

1.3 应用框架层（Java Framework)

应用框架层是系统的核心部分，一方面向上提供接口给应用层调用，另一方面向下与C/C++程序库以及硬件抽象层等进行衔接。 应用框架层的主要实现代码在/frameworks/base和/frameworks/av目录下。

其中系统服务部分在frameworks/base/services中，在frameworks/av/services里存放着音频和照相机的服务。

/frameworks/base目录结构如下：

/frameworks/base目录	描述
api	定义API
core	核心库
docs	文档
include	头文件
libs	库
media	多媒体相关库
nfc-extras	NFC相关
opengl	2D/3D 图形API
sax	XML解析器
telephony	电话通讯管理
tests	测试相关
wifi	wifi无线网络
cmds	重要命令：am、app_proce等
data	字体和声音等数据文件
graphics	图形图像相关
keystore	和数据签名证书相关
location	地理位置相关库
native	本地库
obex	蓝牙传输
packages	设置、TTS、VPN程序
services	系统服务
test-runner	测试工具相关
tools	工具

1.4 C/C++程序库

系统运行库层（Native)中的 C/C++程序库的类型繁多，功能强大，C/C++程序库并不完全在一个目录中，这里给出几个常用且比较重要的C/C++程序库所在的目录位置。

目录位置	描述
bionic/	Google开发的系统C库，以BSD许可形式开源。
/frameworks/av/media	系统媒体库
/frameworks/native/opengl	3D图形库/第三方图形渲染库
/frameworks/native/services/surfaceflinger	图形显示库，主要负责图形的渲染、叠加和绘制等功能
external/sqlite	轻量型关系数据库SQLite的C++实现
external/webp,external/webrtc	网络引擎库

1.5 系统运行库

Dalvik虚拟机的实现: /dalvik，从Android 5.0开始，Android应用程序的默认运行环境为ART (Android Runtime)，ART模式拥有更快更高的运行效率。ART: /art

1.6 硬件抽象层部分

hardware/ libhardware: audio、nfc、power实现 ril: 无线硬件设备与电话的实现

二、配置要求

2.1 硬件配置

- 使用 Ubuntu 16.04 系统
- 电脑内存16G以上（官方要求），实测 8G 以上可以成功
- 硬盘200G以上

2.2 软件配置

- openjdk-8
- Android Studio
- Git
-
- 安装所需的软件包

```
sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev libgl1-mesa-dev libxml2-utils xsltproc unzip
```

2.3 技能要求

- 拥有 Android 开发经验，熟悉 Java 语言
- 对 C / C++ 语言可以达到基本的看懂即可
- 会使用简单的 ubuntu 命令

三、AOSP 环境搭建

3.1 源码下载

3.1.1 网络下载方法

国内不可以访问 Google 网络，故无法下载 aosp 源码，

所以使用国内清华大学源有相关镜像：

参考链接：https://mirror.tuna.tsinghua.edu.cn/help/AOSP/

3.1.2 本地解压方式

下载链接地址：

https://pan.baidu.com/s/1Jwsrb-zwrQO-HEHo5eo9Jg

提取码：uu1j

百度云下载相关的源码包，进行本地解压，

然后执行一下命令：

1.

```
sudo apt-get install p7zip
```

2.

```
7zr x android-8.1.0_r1.7z
```

3.2 编译 aosp 代码

1.

```
. build/envsetup.sh
```

2.

```
lunch
```

选择：6 --- > aosp_x86_64

3.

```
make
```

经历大概几个小时等待，出现如下打印代表成功：

```
#### build completed successfully (05:44:08 (hh:mm:ss)) ####
```

4. 执行命令：emulator，出现模拟器运行成功的页面

3.3 加载源代码到 AndroidStudio

执行以下四条命令：

1.

```
. build/envsetup.sh
```

source 可以用 . 代替，即". build/envsetup.sh"

2.

```
lunch
```

选择要编译的项目

3.

```
make idegen -j4
```

这里的 -j4 表示用 4 线程来编译，可以不加

4.

```
sudo development/tools/idegen/idegen.sh
```