# 《嵌入式系统》

## （第十一讲）

厦门大学信息学院软件工程系 曾文华

2024年12月3日

- 第1章：嵌入式系统概述
- 第2章：ARM处理器和指令集
- 第3章：嵌入式Linux操作系统
- 第4章：嵌入式软件编程技术
- 第5章：开发环境和调试技术
- 第6章：Boot Loader技术
- 第7章：ARM Linux内核
- 第8章：文件系统
- 第9章：设备驱动程序设计基础
- 第10章：字符设备和驱动程序设计
- 第11章：Android操作系统（增加）
- 第12章：块设备和驱动程序设计
- 第13章：网络设备驱动程序开发
- 第14章：嵌入式GUI及应用程序设计

# 第11章　Android 操作系统

- 11.1　Android 操作系统介绍
- 11.2　Android 软件架构介绍
- 11.3　Android 内核
- 11.4　Android 子系统介绍
- 11.5　Android 应用程序开发过程
- 11.6　Android 源码目录结构

# 11.1 Android 操作系统介绍

- Android 是 Google 公司于 2007 年 11 月发布的一款非常优秀的智能移动平台操作系统。到 2011 年第一 季度 Android 在全球的市场份额首次超过 Nokia 的 Symbian 系统，跃居全球第一。

- Android 系统最初由 Andy Rubin 等人于 2003 年 10 月创建。Google 于 2005 年 8 月 17 日收购 Android 并组建 OHA（Open Handset Alliance，开放手机联盟）开发改良 Android，之后逐渐扩展到平板电脑及其他移动平台领域上。

- Android 系统是一个基于 Apache License（Apache 许可证）、GPL（GNU General Public License，GNU 通用公共许可证）软件许可的开源手机操作系统，底层由 Linux 操作系统作为内核，我们可以直接从 Android 的官方网站上下载最新的 Android 源码和相关开发工具包：
  - 面向开发者的Android网址：https://developer.android.google.cn/
  - Android开源项目网址：http://source.android.com/

https://developer.android.google.cn

developers 平台 Android Studio Google Play Jetpack Kotlin 文档 游戏 中文 – 简体

# 面向开发者的 Android

这些现代工具和资源可协助您在各种 Android 设备上打造深受用户喜爱的更快捷、轻松的体验。

搜索

下载 Android Studio

启动 Play 管理中心

查看 Android 课程

精选

## Android 13 开发者预览版

让您的应用做好迎接下一版本 Android 的准备！了解新变化、测试应用并向我

精选

## 2022 年 Google for Games 开发者峰会

https://source.android.google.cn

source 文档 转到代码↗ 中文 – 简体

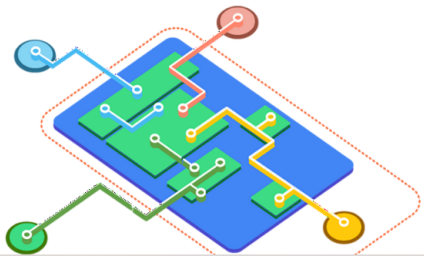
# 安卓
# 开源项目

Android 团结了世界。使用开源 Android 操作系统为您的设备供电。

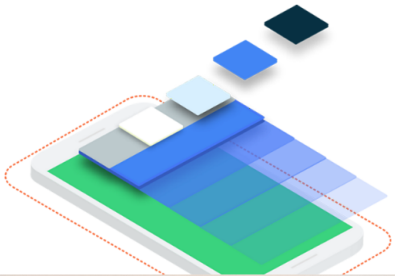
获取源

搜索

translated by Google 此页面由 Cloud Translation API 翻译。 Switch to English



VMware Workstation Pro



# 11.2 Android 软件架构介绍

- Android 的软件架构采用了分层结构，如下图所示，由上至下分别为：Applications 应用层、Application Framework 应用框架层、Android Runtime & Libraries 运行时库和本地库层、Linux Kernel 内核层。


应用层

应用框架层

运行时库和本地库层

内核层

① Applications 应用层：用户安装应用程序及系统自带应用程序层，主要用来与用户进行交互，如 Home 指 Android 手机的桌面，Phone 指电话应用，用来拨打电话等。

② Application Framework 应用框架层：系统框架层，封装了大量应用程序所使用的类，从而达到组件 重用的目的，它主要向上层应用层提供 API，如：ActivityManager 主要用于管理所有的 Activity、画面导航、 回退等生命周期相关的操作，PackageManager 主要用来管理程序安装包的安装、更新、删除等操作。

③ Android Runtime & Libraries 运行时库和本地库层：Runtime 是 Android 的运行环境，在该层有 DalvikVirtualMachine（Android 的虚拟机简称 DVM）的实现，在 DVM 中运行着 Java 的核心语言库代码和 Java 程序。同时，在 DVM 运行期间要调用系统库代码，如：负责显示的 SurfaceManager 本地代码，负责多媒体处理相关的 MediaFrameworks 代码及 C 库 libc 等。

④ Linux Kernel 内核层：Android 系统是基于 Linux 系统的，所以 Android 底层系统相关的框架和标准 的 Linux 内核没有什么很大的区别，只不过添加了几个 Android 系统运行必备的驱动，如：BinderIPC 进程 间通信驱动、PowerManager 电源管理驱动等。

- 总结：Android 的软件架构是我们学习 Android 开发必须要掌握的知识点，它对我们将来编写 Android 应用程序、理解 Android 框架代码、编写本地代码、修改底层驱动都有重要的指导意义，可谓是学习 Android 的灵魂。

# 11.3 Android 内核

- 内核（Kernel）是Android系统最核心的部分，其主要作用在于与计算机硬件进行交互，实现对硬件的编程控制和接口操作。内核的主要功能包括：内存管理、进程间通信、进程调度、虚拟文件系统、网络等功能。和标准的 Linux 内核一样，Android 内核主要实现内存管理、进程间通信及进程调度等功能。Android 对 Linux 内核的更改较少，但增加了一些没有加入标准内核的内容，例如：yaffs 文件系统。

- Android 内核结构和标准 Linux 4.1.15 内核基本上相同，为了适应嵌入式硬件环境和移动应用程序的开 发，Android 在其基础上，增加了私有内容。

- Android 在标准 Linux 内核中增加的内容是一些驱动程序，这些驱动主要分为两种类型：
  ① Android 专用驱动
  ② Android 使用的设备驱动

## ① Android 专用驱动：

- Android 专用驱动和组件并非 Linux 标准的内容，它们实际上是纯软件的内容和体系结构，与硬件平台无关。Android 专用驱动类似于 Linux 内存设备驱动程序。随着 Andorid 内核的升级，Android 专用驱动的目录几经变化，在较新的版本中，主要的驱动程序放在 drivers/staging/android/目录中，另外也有几个驱动程序分布在其他的目录中。

- 主要的 Android 专用驱动如下：
  - ① Binder：基于 OpenBinder 系统的驱动，为 Android 平台提供 IPC 支持；
  - ② Logger：轻量级的 log 驱动；
  - ③ LowMemoryKiller：缺少内存的情况下，杀死进程；
  - ④ Ashmem：匿名共享内存；
  - ⑤ PowerManagement（PM）：电源管理模块；
  - ⑥ PMEM：物理内存驱动。

② Android 使用的设备驱动：

– 作为主要为智能手机定制的操作系统，Android 通常使用 Linux 中一些标准的设备驱动程序。

– Android 使用的设备驱动有：

① Framebuffer 显示驱动；

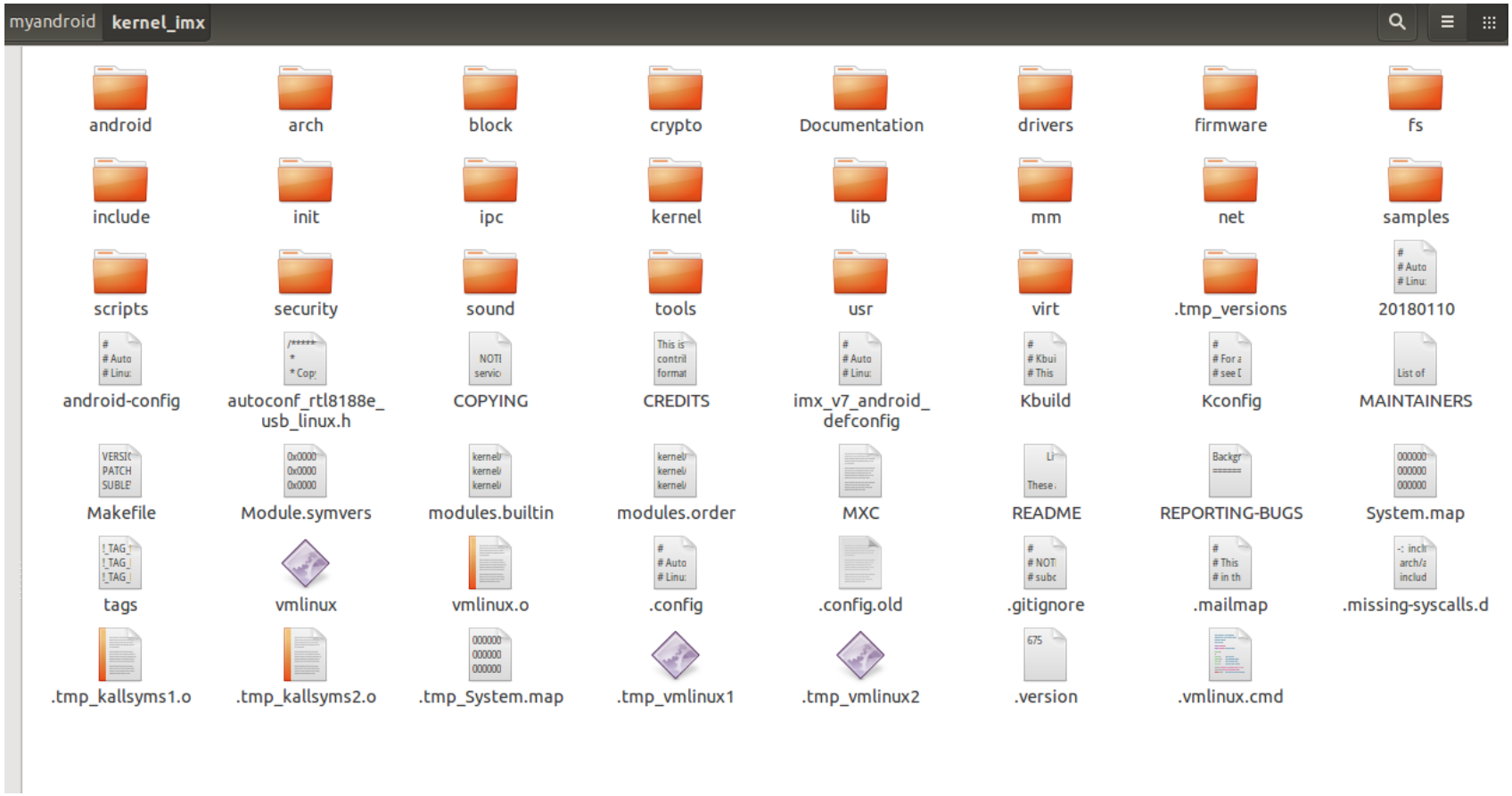
② Event 输入设备驱动；

③ ASLA 音频驱动；

④ WLAN 驱动；

⑤ 蓝牙驱动等。

- **Android 内核目录结构**
  - **Android 内核结构和标准 Linux 4.1.15 内核基本上相同，IMX6 综合嵌入式教学科 研平台（实验箱）运行的 Android 内核版本位于 /Android/myandroid/kernel_imx，其源码目录结构如图：**



- **arch**：与体系结构相关的代码全部放在这里，我们的实验设备中使用的是其中的 arm 目录。

- **Documentation**：这里存放着内核的所有开发文档，其中的文件会随版本的演变发生变化，通过阅读这里的文件是获得内核最新的开发资料的最好的地方。

- **drivers**：此目录包括所有的驱动程序，下面又建立了多个目录，分别存放各个分类的驱动程序源代码。drivers 目录是内核中最大的源代码存放处，大约占整个内核的一多半。其中我们经常会用到的目录有：
  - drivers/**char**：字符设备是 drivers 目录中最为常用，也许是最为重要的目录，因为其中包含了大量与驱动程序无关的 代码。通用的 tty 层在这里实现，console.c 定义了 linux 终端类型，vt.c 中定义了虚拟控制台；lp.c 中实现了 一个通用的并口打印机的驱动，并保持设备无关性；kerboard.c 实现高级键盘处理，它导出 handle_scancode 函数，以便于其他与平台相关的键盘驱动使用。
  - driver/**block**：其中存放所有的块设备驱动程序，也保存了一些设备无关的代码。如 nbd.c 实现了网络块设备，loop.c 实现了回环块设备。
  - drives/**ide**：专门存放针对 IDE 设备的驱动。
  - drivers/**scsi**：存放 SCSI 设备的驱动程序，当前的 cd 刻录机、扫描仪、U 盘等设备都依赖这个 SCSI 的通用设备。
  - drivers/**staging**：Android 专用驱动程序大部分存放在这个目录中。
  - drivers/**net**：存放网络接口适配器的驱动程序，还包括一些线路规程的实现，但不实现实际的通信协议，这部分在 顶层目录的 net 目录中实现。
  - drivers/**video**：这里保存了所有的帧缓冲区视频设备的驱动程序，整个目录实现了一个单独的字符设备驱动。/dev/graphics/fb/fb 设备的入口点在 fbmem.c 文件中，该文件注册主设备号并维护一个此设备的清单，其中记 录了哪一个帧缓冲区设备负责哪个次设备号。
  - drivers/**media**：这里存放的代码主要是针对无线电和视频输入设备，比如目前流行的 usb 摄像头。

- fs：此目录下包括了大量的文件系统的源代码，其中在嵌入式开发中要使用的包括：devfs、cramfs、 ext2、jffs2、romfs、yaffs、vfat、nfs、proc 等。 文件系统是 Linux 中非常重要的子系统，这里实现了许多重要的系统调用；用于文件访问的系统调用在 open.c、read_write.c 等文件中定义，select.c 实现了 select 和 poll 系统调用，pipe.c 和 fifo.c 实现了管道和命 名管道， mkdir、rmdir、rename、link、symlink、mknod 等系统调用在 namei.c 中实现。 文件系统的挂装和卸载和用于临时根文件系统的 initrd 在 super.c 中实现。Devices.c 中实现了字符设备 和块设备驱动程序的注册函数；file.c、inode.c 实现了管理文件和索引节点内部数据结构的组织。Ioctl.c 实 现 ioctl 系统调用。

- include：内核所有头文件存放的地方，其中 Linux 目录是头文件最多的地方，也是驱动程序经常要包含的目录。

- init：Linux 的 main.c 程序，通过这个比较简单的程序，我们可以理解 Linux 的启动流程。

- ipc：systemV 的进程间通信的原理实现，包括信号量、共享内存。

- kernel：这个目录下存放的是除网络、文件系统、内存管理之外的所有其他基础设施，其中至少包括进程调度 sched.c，进程建立 fork.c，定时器的管理 timer.c，中断处理，信号处理等。

- lib：包括一些通用支持函数，类似于标准 C 的库函数。其中包括了最重要的 vsprintf 函数的实现，它是 printk 和 sprintf 函数的核心。还有将字符串转换为长整形数的 simple_atol 函数。

- mm：包含实现内存管理的代码，包括所有与内存管理相关的数据结构，我们在驱动中需要使用的 kmalloc 和 kfree 函数在 slab.c 中实现，mmap 定义在 mmap.c 中的 do_mmap_pgoff 函数。将文件映射到内存的实现在 filemap.c 中，mprotect 在 mprotect.c，remap 在 remap.c 中实现；vmscan.c 中实现了 kswapd 内核线程。

- net：包含了套接字抽象和网络协议的实现，每一种协议都建立了一个目录，但是其中的 core、 bridge、ethernet、 sunrpc、khttpd 不是网络协议。我们使用最多的是 ipv4、ipv6、802、ipx 等。

- scripts：这个目录存放许多脚本，主要用于配置内核。

# 11.4 Android 子系统介绍

- Android 是一个庞大的手机的系统，它不仅仅实现了手机的基本的打电话，发信息的功能，还实现了更复杂的多媒体处理、2D 和 3D 游戏处理、信息感知处理等。

- Android 的子系统主要包含：
  ① Android RIL 子系统： Radio Interface Layer（简称：RIL子系统，即：无线电接口层子系统）用于管理用户的电话、短信、数据通信等相关功能，它是每个移动通信设备必备的系统。

  ② Android Input 子系统： Input 输入子系统用来处理所有来自己用户的输入数据，如：触摸屏，声音控制物理按键等。

  ③ Android GUI 子系统： GUI 即：图形用户接口子系统，也就是所谓的图形界面，它用来负责显示系统图形化界面，形象让用户和系统操作及信息进行交互。Android 的 GUI 系统和其它各子系统关系密切相关，是 Android 中最重要的子系 统之一，如：绘制一个 2D 图形、通过 OpenGL 库处理 3D 游戏、通过 SurfaceFlinger（Android 中图形混合器，用于将屏幕上显示的多个图形进行混合显示）来重叠几个图形界面。

④ Android Audio 子系统： Android 的音频处理子系统，主要用于音频方面的数据流传输和控制功能，也负责音频设备的管理。 Android 的 Audio 系统和多媒体处理紧密相连，如：视频的音频处理和播放、电话通信及录音等。

⑤ Android Media 子系统： Android 的多媒体子系统，它是 Android 系统中最庞大的子系统，与硬件编解码、OpenCore 多媒体框架、 Android 多媒体框架等相关，如：音频播放器，视频播放器，Camera 摄像预览等。

⑥ Android Connectivity 子系统： Android 连接子系统是智能设备的重要组成部分，它除了一般所谓的网络连接，如：以太网、WIFI 外， 还包含：蓝牙连接、GPS 定位连接、NFC 等。

⑦ Android Sensor 子系统： Android 的传感器子系统为当前智能设备大大提高了交互性，它在创新应用程序和应用体验里发挥了重 要作用，传感器子系统和手机的硬件设备紧密相关，如： gyroscope 陀螺仪、accelerometer 加速度计、 proximity 距离感应器、magnetic 磁力传感器等。

# 11.5 Android 应用程序开发过程

- Android 应用程序开发是基于 Android 架构提供的 API 和类库编写程序，这些应用程序是完全的 Java 代码程序，它们构建在 Android 系统提供的 API 之上。

- Android 开发方式：开发 Android 应用程序可以基于 Google 提供的 Android SDK 开发工具包，也可以直接在 Android 源码中进行编写。

  ① Android SDK 开发：它提供给程序员一种最快捷的开发方式，基于 IDE 开发环境（Android Studio）和 SDK 套件（Android SDK），快速开发出标准的 Android 应用程序，但是，对于一些要修改框架代码或基于自定义 API 的高级开发，这种方式难以胜任。

  ② Android 源码开发：基于 Android 提供的源码进行开发，可以最大体现出开源的优势，让用户自定义个性的 Android 系统，开发出更高效、更与众不同的应用程序，这种方式更适合于系统级开发，对程序员要求比较高。

API：Application Programming Interface，应用程序接口
SDK：Software Development Kit，软件开发工具包
IDE：Integrated Development Environment，集成开发环境

- Android 源码开发过程：

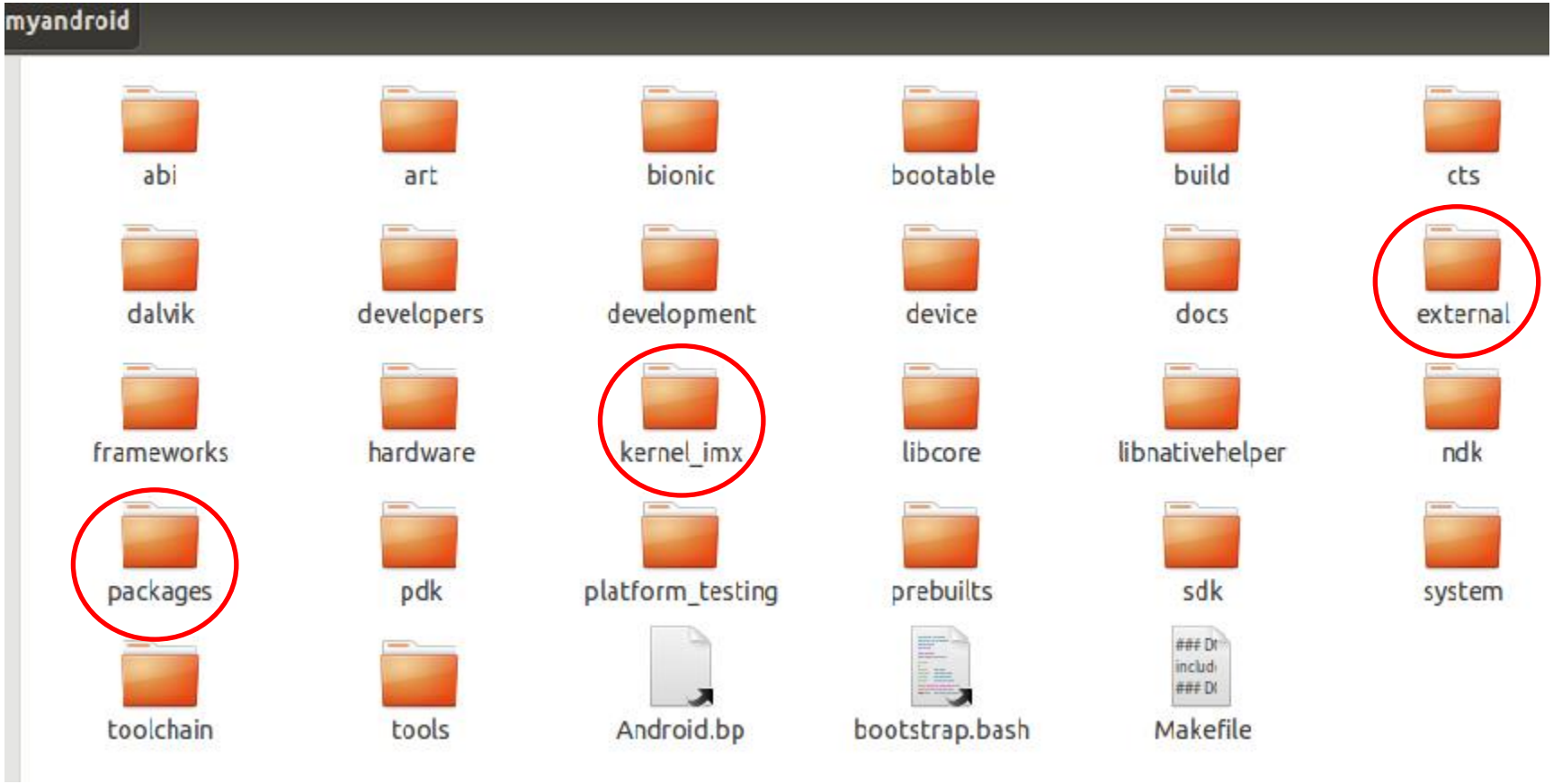
① 搭建开发环境：根据两种开发方式的不同，搭建开发环境略有不同。如果侧重于系统底层源码开发，则采用第二种开发方式（Android 源码开发）；如果侧重于应用软件开发，则采用第一种开发方式（Android SDK 开发）。

② 下载 Android 源码： 得益于 Android 的开源特点，Android 源码中包含大量宝贵的技术知识，我们可以在阅读源码过程中更深入的了解 Android 系统的奥秘，为我们编写更高效、更有特点的应用程序打下基础，同时能展现给读者一 个更庞大系统的设计蓝图，为系统设计师及项目经理提供参考价值。同时，Android 的源码中提供的应用程序示例、设计模式、软件架构，为我们编写大型应用程序提供经验。

③ 编译 Android 源码： 通过编译 Android 源码，生成我们开发环境及目标系统，为我们做系统底层开发、系统定制与优化做准备，通过分析编译过程，让我们学习到大型工程的代码管理与编译原理。

④ 配置开发环境：为了更有效的开发，我们通常会对开发环境做配置，不同的程序员可能会有不同的编程习惯。

# 11.6 Android 源码目录结构




myandroid.tar.xz

/Android/myandroid/

1. **abi**：应用程序二进制接口

2. **art**：全新的**ART**运行环境

3. **bionic**：针对 **Android** 系统定制的仿生标准 **C** 库、链接器等所在目录，**Android** 系统并没有使用 **Linux** 的 **glibc** 库， **bioinc C** 库针对嵌入式系统做了优化，添加了一些 **Android** 特定的函数 **API** 同时大大减少库的体积，也避 免了 **LGPL** 版权的问题。

4. **bootable**： **Android** 系统引导启动代码，用来引导系统、更新系统、恢复系统。

5. **build**： **Android** 的编译系统目录，里面包含大量的 **Makefile**，用来编译目标系统、**Host** 主机开发环境等。

6. **cts**： 兼容性测试工具目录。

7. **dalvik**： **Dalvik** 虚拟机，**Android** 系统得以运行的虚拟执行环境。

8. **developers**：开发者目录

9. **development**： 程序开发所需要的模板和工具。

10. **device**：厂商设备配置目录，针对不同设备，由不同的子目录来分别管理，用来裁剪实现不同设备上 **Android** 目标 系统。

11. docs：参考文档目录

12. external： Android 系统使用的其它开源代码目录（外部资源），如 jpeg 图片解码开源库、opencore 开源代码等。

13. frameworks：框架层代码，frameworks/base 目录下存放目标系统的框架库，frameworks/policies/base 下存放应用程序框 架代码。

14. hardware： HAL（Hardware Abstraction Layer）硬件抽象层代码。

15. kernel_imx： Linux 内核目录，默认下载的 Android 源码里没有，需单独下载。

16. libcore：核心库相关文件

17. libnativehelper：动态库，实现JNI库的基础

18. ndk：NDK相关代码，帮助开发人员在应用程序中嵌入C/C++代码

19. packages： Android 系统级应用程序源码目录，如摄像应用、电话应用等。
    - packages/app：应用程序
    - packages/providers：内容提供者

20. pdk：Plug Development Kit 的缩写，本地开发套件

21. **platform_testing：平台测试**

22. **prebuilts： 主机编译工具目录，如 arm-linux-gcc 交叉系统工具链等。**

23. **sdk： SDK 及模拟器。**

24. **system： init 进程、蓝牙、无线 WIFI 工具、uevent 进程目录。**

25. **toolchain：工具链文件**

26. **tools：工具文件**

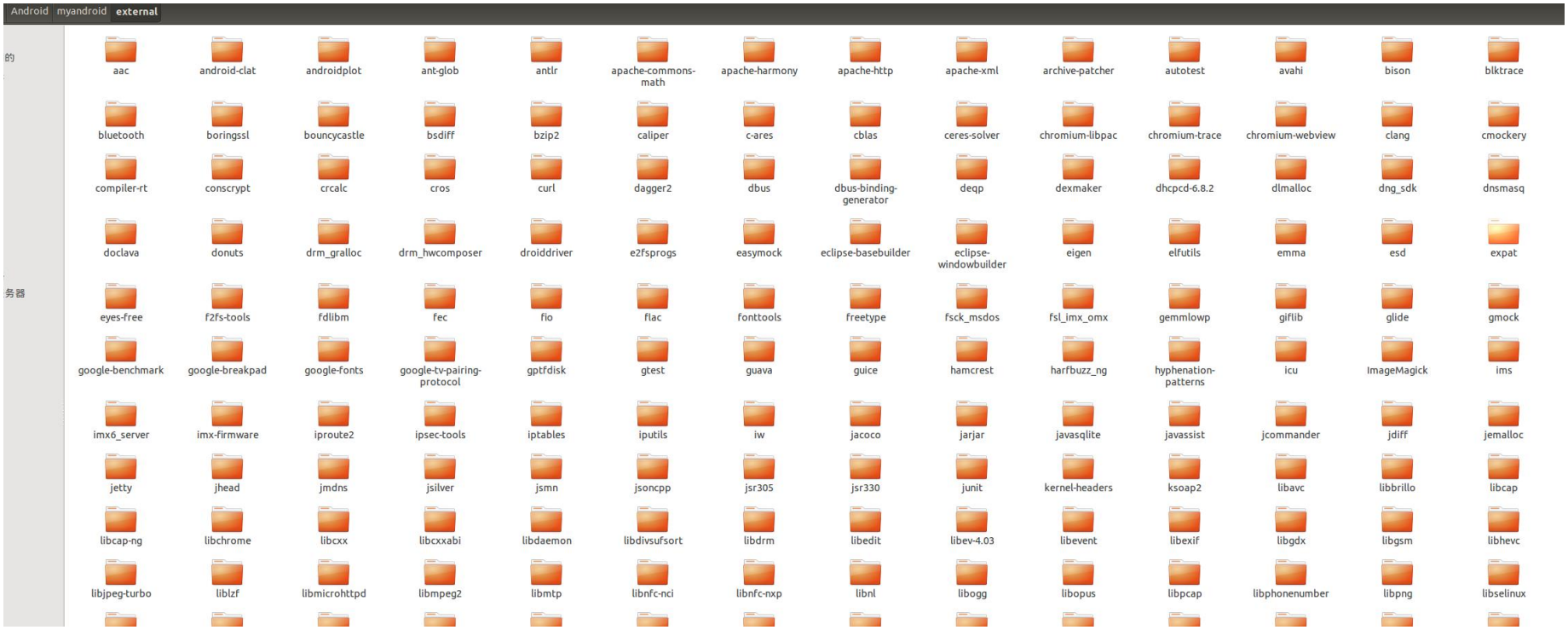
27. **Android.bp： Android.bp是用来替换Android.mk的配置文件，它使用Blueprint框架来解析**

28. **bootstrap.bash：一个用于前端开发的开源工具包**

29. **Makefile：编译文件**

# external

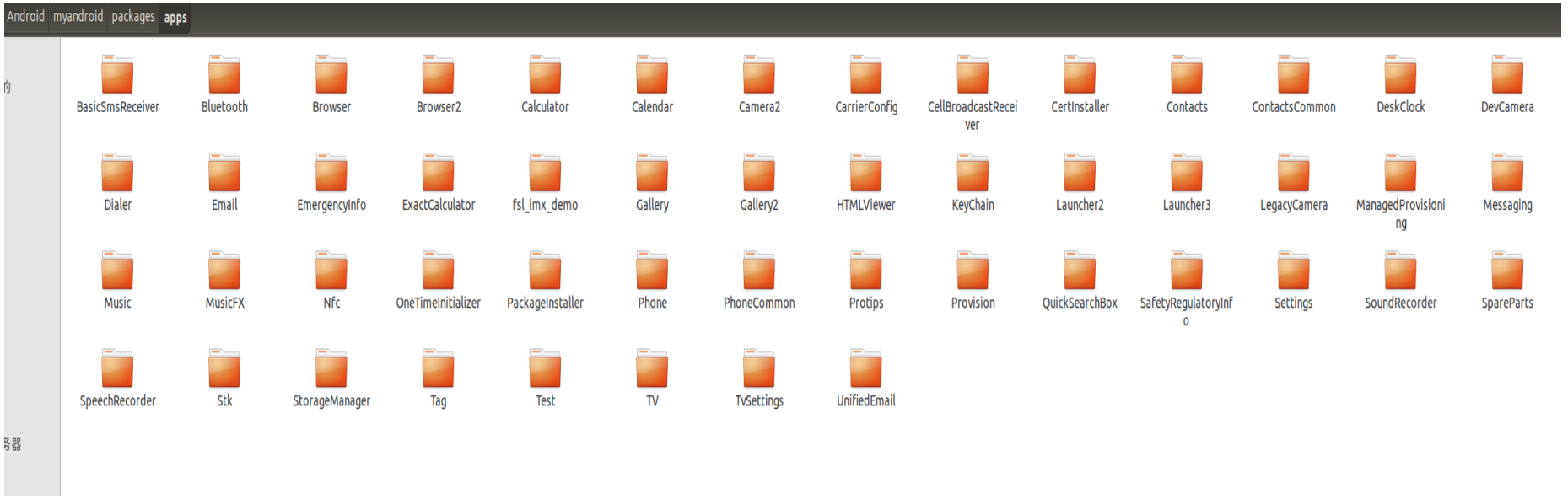
- 在做 Android 开发的时候，有时候需要调试工具、协议库等，Android 引用外部资源，这些都放在 external 目录下，下表列出 Android 源码自带的部分开源项目。



1. acc： 高级音频编码库
2. android-clat：AndroidCLAT 实现了android平台的CLAT，当应用不支持64位DNS， 通过CLAT 处理从IPV4到IPV6 的转换
3. bluetooth： Android 蓝牙（以前是 blueZ，现在是 bluedroid）
4. dbus： 低延时、低开销、高可用性的 IPC 机制
5. dhcpcd-6.8.2： DHCP 服务
6. elfutils： ELF 工具包
7. emma： Java 代码覆盖率统计工具
8. esd： 将多个音频混合在一个音频设备上
9. libnfc-nxp： NXP 的 nfc 库wireless_tools.30.rtl：wifi 调试工具
10. libpng： Png 图片库
11. libselinux： 类似 Linux 中的 selinux
12. libxml2： Xml 的解析库
13. wpa_supplicant_8：Wpa 加密，wifi 驱动和 Android 交互等都由它完成

# packages/apps

- 一套 Android 系统编译完成后，一般有一部分系统应用，而系统应用就是存放在源码的 packages/apps 目录下，这些系统应用基本是每个 Android 必备的应用软件。当然开发商也可以将自己的平台配套应用放在该目录下或者目标平台目录的 app 下，作为系统应用提供给客户。 主要的系统应用及说明如下表所示。这些系统应用基本是每个手机必须的，因此默认安装这些系统应用。



1. **AlarmClock** 闹钟
2. **Browser** 浏览器
3. **Calculator** 计算器
4. **Calendar** 日历
5. **Camera** 摄像头
6. **Contacts** 联系人
7. **Email** 邮件
8. **GoogleSearch Google** 搜索
9. **HTML Viewer** 浏览器附属界面，被浏览器应用调用，同时提供存储记录功能
10. **IM** 即时通讯，为手机提供信号发送、接收、通信的服务
11. **Launcher Android** 的桌面
12. **Mms** 彩信业务
13. **Music** 音乐播放器
14. **PackageInstaller** 应用程序安装、卸载器
15. **Phone** 电话应用

# packages/providers

- 如果做应用开发，肯定会有内容提供者，系统默认有些内容提供者，如联系人通讯录，他们存放在 **packages/providers** 目录下，具体如下表所示。

1. **CalendarProvider 日历提供器**
2. **ContactsProvider 联系人提供器**
3. **DownloadProvider 下载管理提供器**
4. **DrmProvider DRM 受保护数据存储服务，创建和更新数据库时调用**
5. **GoogleContactsProvider 谷歌联系人提供器**
6. **GoogleSubscribedFeedsProvider Google 同步功能**
7. **ImProvider 即时通讯提供器**
8. **MediaProvider 媒体提供器、提供存储数据**
9. **SettingsProvider 系统设置提供器**
10. **TelephonyProvider 彩信提供器**

# Thanks