

# Android 软件开发

## Android 系统简介

秦兴国

xgqin@guet.edu.cn

计算机与信息安全学院  
桂林电子科技大学

2017 年 9 月 27 日

- ① 智能手机发展简介
- ② Android 发展历史简介
- ③ Android 框架简介
- ④ Android 四大组件及应用开发特色
- ⑤ 如何学习 Android

# 变革性的产品 — iPhone

Steve Jobs of Apple introduces the iPhone which he refers to as a “revolutionary and magical product” in Jan 2007.<sup>1,2,3</sup>



---

<sup>1</sup> <https://en.wikipedia.org/wiki/Smartphone>

<sup>2</sup> <https://www.theguardian.com/technology/2012/jan/24/smartphones-timeline>

<sup>3</sup> <https://www.mobileindustryreview.com/2016/10/the-history-of-the-smartphone.html>

# Microsoft 的古董 — Windows Mobile

In the first three months of 2007 Microsoft's **Windows Mobile** had an 18% share of the smartphone market <sup>4, 5</sup>.



In Dec 2008, Microsoft decides to kill off Windows Mobile because it can't compete with the iPhone and Android, and develop Windows Phone – a completely new mobile operating

# 姗姗来迟的竞争者 — Android

First Android phone, the G1<sup>6</sup>, launches in Nov 2008. It has a slide-out keyboard and limited touchscreen capability<sup>7</sup>.



Google announces it will offer the Android mobile operating system for free in Nov 2007. Anyone can use it and change it.

<sup>6</sup> or HTC Dream outside of the United States)

<sup>7</sup> <https://www.androidcentral.com/androids-early-days>

# 帝国的余辉 — Symbian & Blackberry

RIM<sup>8</sup> has a 20% share of the smartphone market from July-September, says Gartner. Second only to Nokia's Symbian<sup>9</sup>, which has 44% in Aut 2009.



<sup>8</sup> <https://en.wikipedia.org/wiki/BlackBerry>

<sup>9</sup> <https://en.wikipedia.org/wiki/Symbian>

# 从 pocket 到 tablet — iPad

Apple launches the iPad, a 10in tablet, **Jan 2010**.



# 渐渐追赶上来的新霸主 — Android

Android phones with full touchscreen interaction like the iPhone's appear<sup>10</sup>, **Feb 2010**.



<sup>10</sup>

Apple sues Taiwan's HTC over its touchscreen Android phone, Mar 2010.



# 不甘失败的竞争者 — Windows Phone

Microsoft's first phones running Windows Phone. Sales are low, **Oct 2010**.



# 移动互联网来得比想象中的更快

Researchers Gartner and IDC announce that smartphones outsold PCs worldwide in the last three months of 2010, 100m as against 93m, **Jan 2011**<sup>11</sup>.

Worldwide smart phone and client PC shipments				
Shipments and growth rates by category, Q4 2011 and full year 2011				
Category	Q4 2011 shipments (millions)	Growth Q4'11/Q4'10	Full year 2011 shipments (millions)	Growth 2011/2010
Smart phones	158.5	56.6%	487.7	62.7%
Total client PCs	120.2	16.3%	414.6	14.8%
- Pads	26.5	186.2%	63.2	274.2%
- Netbooks	6.7	-32.4%	29.4	-25.3%
- Notebooks	57.9	7.3%	209.6	7.5%
- Desktops	29.1	-3.6%	112.4	2.3%
Source: Canalys estimates © Canalys 2012				

# 移动设备中利润的霸主 — Apple

Apple becomes the largest smartphone vendor by numbers and revenue, selling 37m iPhones, in the last quarter of 2011. Android becomes the best-selling smartphone platform, with a 48.8% share, ahead of iOS's 19.1%, **at the end of 2011.**

Worldwide smart phone market Shipments by platform, Q4 2011			
Platform	Q4 2011 shipments (millions)	Share (%)	Growth Q4'11/Q4'10
Total	158.5	100.0%	56.6%
Android	81.9	51.6%	148.7%
iOS	37.0	23.4%	128.1%
Symbian	18.3	11.6%	-40.9%
BlackBerry	13.2	8.3%	-9.7%
bada	3.8	2.4%	39.1%
Windows Phone	2.5	1.6%	-14.0%
Others	1.8	1.1%	117.9%
Source: Canalys estimates © Canalys 2012			

Worldwide smart phone market Shipments by platform, full year 2011			
Platform	Full year 2011 shipments	Share (%)	Growth Q4'11/Q4'10
Total	487.7	100.0%	62.7%
Android	237.8	48.8%	244.1%
iOS	93.1	19.1%	96.0%
Symbian	80.1	16.4%	-29.1%
BlackBerry	51.4	10.5%	5.0%
bada	13.2	2.7%	183.1%
Windows Phone	6.8	1.4%	-43.3%
Others	5.4	1.1%	14.4%
Source: Canalys estimates © Canalys 2012			

# 游戏的生存者与掌局者

Worldwide smartphone OS market share<sup>12</sup> and smartphone vendor market share<sup>13</sup> in the first quarter of 2017.

Period	Android	iOS	Windows Phone	Others
2016Q1	83.4%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%
2016Q3	86.8%	12.5%	0.3%	0.4%
2016Q4	81.4%	18.2%	0.2%	0.2%
2017Q1	85.0%	14.7%	0.1%	0.1%

Source: IDC, May 2017

Period	Samsung	Apple	Huawei	OPPO	vivo	Others
2016Q1	23.8%	15.4%	8.4%	5.9%	4.4%	42.1%
2016Q2	22.7%	11.7%	9.3%	6.6%	4.8%	45.0%
2016Q3	20.9%	12.5%	9.3%	7.1%	5.9%	44.3%
2016Q4	18.0%	18.2%	10.5%	7.3%	5.7%	40.2%
2017Q1	23.3%	14.7%	10.0%	7.5%	5.5%	39.0%

Source: IDC, May 2017

# Andy Rubin 与 Android

*I don' t believe that your phone should be an assistant. -  
Andy Rubin<sup>14, 15 , 16</sup>*



---

<sup>14</sup> <https://successstory.com/people/andy-rubin>

<sup>15</sup> <http://www.ifanr.com/265116>

<sup>16</sup> [https://en.wikipedia.org/wiki/Andy\\_Rubin](https://en.wikipedia.org/wiki/Andy_Rubin)

# Google 与 Android

Google 于 2005 年收购 Android 公司，Rubin 从那时起开始出任 Google 移动与数字内容部门高级副总裁，并主导 Android 项目开发。

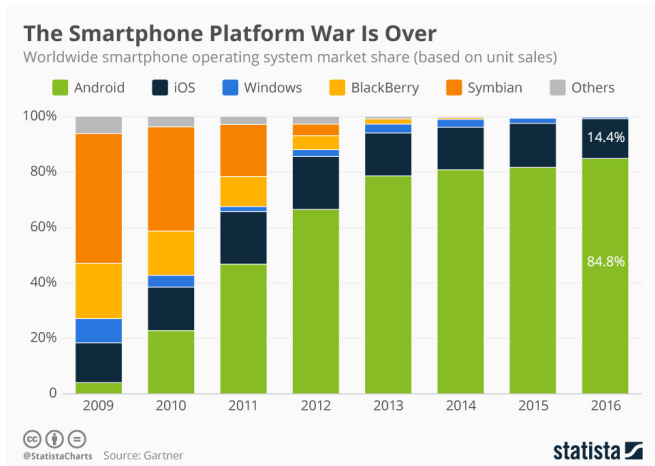


Google 于 2007 年 11 月发起成立 OHA(Open Handset Alliance)<sup>17</sup>。

<sup>17</sup> [https://en.wikipedia.org/wiki/Open\\_Handset\\_Alliance](https://en.wikipedia.org/wiki/Open_Handset_Alliance)

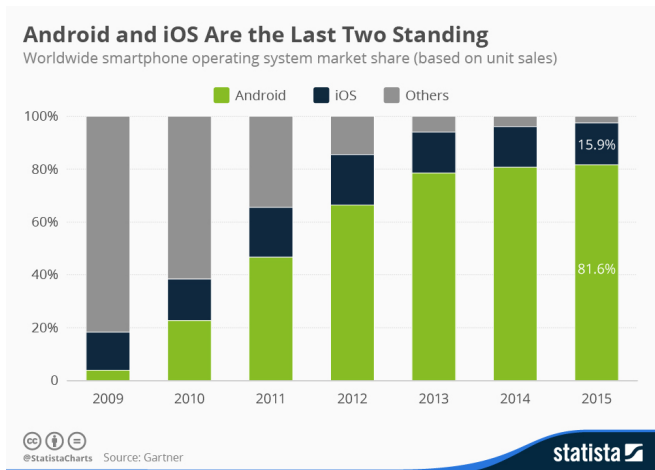
# Android 与智能手机系统 I

2009 至 2016 年，各智能手机系统占有率变化情况<sup>18</sup>。



# Android 与智能手机系统 II

2009 年至 2015 年，Android 与 iOS 系统占有率变化情况<sup>19</sup>。



18

<https://www.statista.com/chart/4112/smartphone-platform-market-share/>

19

<https://www.statista.com/chart/4431/smartphone-operating-system-market-share/>



# Android 各版本

Android 系统自 2008 年 9 月推出以来，平均每半年升级一次，并且在 1.5 版 (Cupcake) 之后，每个版本均以甜点命名。这些甜点的名称首字母依照 CDEFGHI 排序<sup>20</sup>。



# Android 各版本 续

Android 各版本代号及 API 版本号<sup>21</sup>。

Code name	Version	API level	Code name	Version	API level
no code name	1.0	API level 1	Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14
no code name	1.1	API level 2	Ice Cream Sandwich	4.0.4 - 4.0.4	API level 15
Cupcake	1.5	API level 3	Jelly Bean	4.1.x	API level 16
Donut	1.6	API level 4	Jelly Bean	4.2.x	API level 17
Eclair	2.0	API level 5	Jelly Bean	4.3.x	API level 18
Eclair	2.0.1	API level 6	Kitkat	4.4 - 4.4.4	API level 19
Eclair	2.1	API level 7	Lollipop	5.0	API level 21
Froyo	2.2.x	API level 8	Lollipop	5.1	API level 22
Gingerbread	2.3-2.3.2	API level 9	Marshmallow	6.0	API level 23
Gingerbread	2.3.3-2.3.7	API level 10	Nougat	7.0	API level 24
Honeycomb	3.0	API level 11	Nougat	7.1	API level 25
Honeycomb	3.1	API level 12	Oreo	8.0.0	API level 26
Honeycomb	3.2.x	API level 13			

Cupcake

纸杯蛋糕

Donut

甜甜圈

Eclair

闪电泡芙

Froyo

冻酸奶

Gingerbread

姜饼

Honeycomb

蜂窝

Ice Cream Sandwich

冰激凌三明治

Jelly Bean

果冻豆

Kitkat

奇巧

Lollipop

棒棒糖

Marshmallow

棉花糖

Nougat

牛轧糖

Oreo

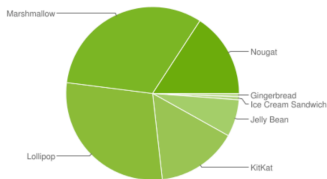
奥利奥

<sup>21</sup><https://source.android.com/source/build-numbers>

# Android 各版本市场份额

- 4.0 以上的版本占据超过 98% 的份额；
- 各版本份额分布过于分散，不利于 App 开发者；

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.6%
4.1.x	Jelly Bean	16	2.4%
4.2.x		17	3.5%
4.3		18	1.0%
4.4	KitKat	19	15.1%
5.0	Lollipop	21	7.1%
5.1		22	21.7%
6.0	Marshmallow	23	32.2%
7.0	Nougat	24	14.2%
7.1		25	1.6%

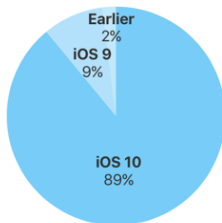
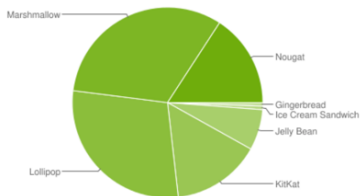


Data collected during a 7-day period ending on September 11, 2017.

Any versions with less than 0.1% distribution are not shown.

# Android 各版本市场份额 续

与 iOS 系统相比，Android 系统存在明显的碎片化问题。

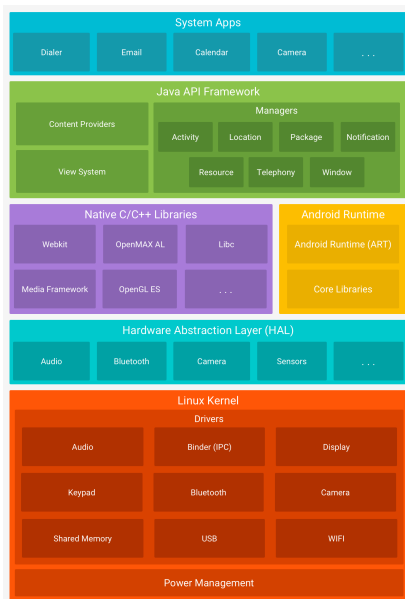


As measured by the App Store on September 6, 2017.

这导致:

- 开发者需针对不同 Android 版本进行适配;
- 手机用户无法及时体验新版本系统;
- Google 也不敢轻易停止对旧版本系统的维护;

# Android 系统整体架构



# Android 系统架构介绍 I

Android 系统大致可以分为 5 层<sup>22</sup>：Linux 内核层、硬件抽象层、系统运行库层、应用框架层、系统应用层；

- **The Linux Kernel**: Android 系统基于 Linux 内核，Android 运行时 (Android Runtime, ART) 基于内核层的关键功能，例如线程管理、底层内存管理等；Linux 内核层为 Android 设备提供底层驱动；
- **Hardware Abstraction Layer(HAL)**: 硬件抽象层为上层提供标准接口用于对提升硬件管理能力；该层包含众多库模块，每一个库针对一类硬件；
- **系统运行库层**：包括 Android Runtime 及本地 C/C++ 库两部分；

## Android 系统架构介绍 II

- **Android Runtime:** 从 Android 5.0 开始, Android App 运行的虚拟机从 Dalvik 转到 ART; 以期为 App 运行提供更快的运行环境; 其包含一系列特性: Ahead-of-time(AOT), just-in-time(JIT) 编译等<sup>23</sup>;
- **Native C/C++ libraries:** Android 系统的众多核心组件及服务 (ART, HAL) 均基于这类库; 特别是如果需要在 App 中使用 C 或 C++ 代码时, 可通过 Android NDK<sup>24</sup>对这些本地库进行访问;
- **Java API Framework:** 应用框架层使用 Java 语言编写, 为 App 提供所有核心特性、可重用模块代码及组件服务:
  - **丰富和可扩展的视图系统 (View System),** 用户可通过视图系统构建 App UI;

# Android 系统架构介绍 III

- **资源管理器 (Resource Manager)**, 提供对非代码资源 (localized strings, graphics, layout files) 的访问;
- **通知消息管理器 (Notification Manager)**, 为 App 提供定制通知消息功能;
- **活动管理器 (Activity Manager)**, 管理 App 的生命周期以及提供通用的活动导航栈 (navigation back stack);
- **内容提供者 (Content Providers)**, 为 App 提供访问第三方 App 数据的功能;
- **System Apps**: 系统应用层主要包含 Android 系统的一些核心 App, 例如邮件、短信、日历、浏览器、联系人等; 系统应用除了作为应用对用户提供服务外, 还为其他 App 提供功能入口;

---

<sup>22</sup> <https://developer.android.com/guide/platform/index.html>

<sup>23</sup> <http://www.jianshu.com/p/58f817d176b7>

<sup>24</sup> <https://developer.android.com/ndk/index.html>



# Android 四大组件 I

Android 应用组件是进行 App 开发所必须的模块组件，每一个组件可视为系统或用户进入 App 的入口，部分组件之间存在依赖关系，在 Android 系统中，组件被分为四类：

- **活动 (Activities)**: An activity is the entry point for interacting with the user. It represents a single screen with a user interface.
- **服务 (Services)**: A service is a general-purpose entry point for keeping an app running in the background for all kinds of reasons.

# Android 四大组件 II

- **广播接收者 (Broadcast receivers):** A broadcast receiver is a component that enables the system to deliver events to the app outside of a regular user flow, allowing the app to respond to system-wide broadcast announcements.
- **内容提供者 (Content Providers):** A content provider manages a shared set of app data that you can store in the file system, in a SQLite database, on the web, or on any other persistent storage location that your app can access. Through the content provider, other apps can query or modify the data if the content provider allows it.

每一类组件都提供特定的服务，并且拥有该类组件特定的创建及销毁的生命周期。

# 使用 Intent 激活组件 I

Android 应用四大组件中的活动、服务、广播通过称为是 intent 的异步消息进行激活；Intent 可以在运行时将独立的组件进行绑定，从而为用户打造复杂、连续的用户交互界面及业务流；

可以将 Intent 视为某一组件动作 (action) 的请求者，而这些组件可以属于应用本身或其他应用；

There are separate methods for activating each type of component:

- You can start an activity or give it something new to do by passing an Intent to `startActivity ()` or `startActivityForResult ()` (when you want the activity to return a result).

## 使用 Intent 激活组件 II

- With Android 5.0 (API level 21) and later, you can use the `JobScheduler` class to schedule actions. For earlier Android versions, you can start a service (or give new instructions to an ongoing service) by passing an `Intent` to `startService()`. You can bind to the service by passing an `Intent` to `bindService()`.
- You can initiate a broadcast by passing an `Intent` to methods such as `sendBroadcast()`, `sendOrderedBroadcast()`, or `sendStickyBroadcast()`.
- You can perform a query to a content provider by calling `query()` on a `ContentResolver`.

# Android 系统特点 I

Android 应用程序使用 Java 语言进行开发 (Kotlin<sup>25</sup>正式成为 Android 应用开发语言)。Android SDK 工具将 Java 源码与数据及资源文件编译成 **APK** 文件, 该文件称为 Android 应用程序包 (package), 以`.apk` 为后缀。

每一个 Android 应用程序运行于一个独立的安全沙盒内 (security sandbox), 并由以下 Android 安全机制提供保护:

- Android 操作系统是一个多用户的 Linux 操作系统, 在 Android 系统中, 每一个应用程序对应一个用户;
- 系统默认为每一个应用程序分配一个独立的 Linux 用户 ID(该用户 ID 对应用程序而言是不可见的)。系统为该应用程序内部所需文件及数据分配权限, 从而使得只有该应用程序所对应的用户才具备访问权限;

## Android 系统特点 II

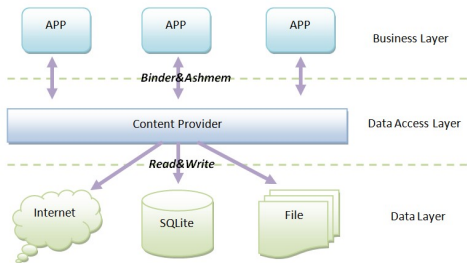
- 每一个进程运行于一个独立的虚拟机中 (VM)，因此应用程序之间的代码是相互隔绝的 (in isolation from other apps)；
- 每一个应用程序运行于独立的 Linux 进程中，当某个 App 的组件需要被执行是，Android 系统将启动该进程；当组件不在需要或系统处于低内存模式时，应用程序所在进程将被关闭；

Android 应用程序在执行时遵循最低访问权限原则 (principle of least privilege)，即应用程序在运行时，仅可访问运行其组件所需的授权权限。

# Android 应用开发特点 I

- 采用 XML 文件对界面进行布局描述，界面设计与程序逻辑分离，有利于界面的修改和维护；
- 采用轻量级进程间通讯机制 Intent，使得跨进程组件通信和发送系统广播称为可能；
- 提供 Service 作为无用户界面、长时间后台运行组件；
- 支持高效、快速的数据存储：
  - SharedPreferences;
  - 文件存储;
  - 轻量级关系型数据库 SQLite;
- 为跨进程共享数据，提供 ContentProvider 接口；

# Android 应用开发特点 II



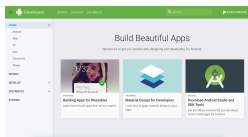
可以无需了解数据源、路径的情况下，对共享数据进行 CRUD 操作；

- 支持位置服务 (LBS) 和地图应用



# 如何学习 Android

- Android 开发者中心 <https://developer.android.com>



- Android 开发中文站 <http://www.androidchina.net>
- 《第一行代码 —Android(第二版)》 & 《疯狂的 Android 讲义》



- github <https://github.com>