android（Google公司开发的操作系统）

Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用 “安卓”或“安致”。Android 操作系统最初由 Andy Rubin 开发，主要支持手机。 2005 年 8 月由 Google 收购注资。 2007 年 11 月， Google 与 84 家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良 Android 系统。随后 Google 以 Apache 开源许可证的授权方式，发布了 Android 的源代码。第一部 Android 智能手机发布于 2008 年 10 月。Android 逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机等。 2011 年第一季度， Android 在全球的市场份额首次超过塞班系统，跃居全球第一。 2013 年的第四季度， Android 平台手机的全球市场份额已经达到 78.1%。[1] 2013 年 09 月 24 日谷歌开发的操作系统 Android 在迎来了 5 岁生日，全世界采用这款系统的设备数量已经达到 10 亿台。2014 第一季度 Android 平台已占所有移动广告流量来源的 42.8%，首度超越 iOS。但运营收入不及 iOS。

编程语言

C/C++（底层） Java 等（应用层）

系统家族

类 Unix,Linux

源码模式

自由及开放源代码软件

内核类型

宏内核 (Linux 内核)

软件许可

Apache License、 GPL 等

1 系统简介编辑

Android 一词的本义指 “机器人”，同时也是 Google 于 2007 年 11 月 5 日

Android logo 相关图片 (36 张)

宣布的基于 Linux 平台的开源手机操作系统的名称，该平台由操作系统、中间件、用户界面和应用软件组成。

Android 一词最早出现于法国作家利尔亚当（ Auguste Villiers de l'Isle-Adam ）在 1886 年发表的科幻小说《未来夏娃》 （L' è ve future）中。他将外表像人的机器起名为 Android。

Android 的 Logo 是由 Ascender 公司设计的，诞生于 2010 年，其设计灵感源于男女厕所门上

的图形符号， [1] 于是布洛克绘制了一个简单的机器人，它的躯干就像锡罐的形状，头上还有两根天线， Android 小机器人便诞生了。 其中的文字使用了 Ascender 公司专门制作的称之为“ Droid 的字体。 Android 是一个全身绿色的机器人， 绿色也是 Android 的标志。 颜色采用了 PMS 376C 和 RGB 中十六进制的 #A4C639 来绘制，这是 Android 操作系统的品牌象徵。有时候，它们还会使用纯文字的 Logo。[1]

2012 年 7 月美国科技博客网站 BusinessInsider 评选出二十一世纪十款最重要电子产品，Android 操作系统和 iPhone 等榜上有名。

（Android logo 相关图片相册图片来源： [3] ）

2 发展历程编辑

2003 年 10 月， Andy Rubin 等人创建 Android 公司，并组建 Android 团队。 [4]

2005 年 8 月 17 日， Google 低调收购了成立仅 22 个月的高科技企业 Android 及其团队。安迪鲁宾成为 Google 公司工程部副总裁，继续负责 Android 项目。

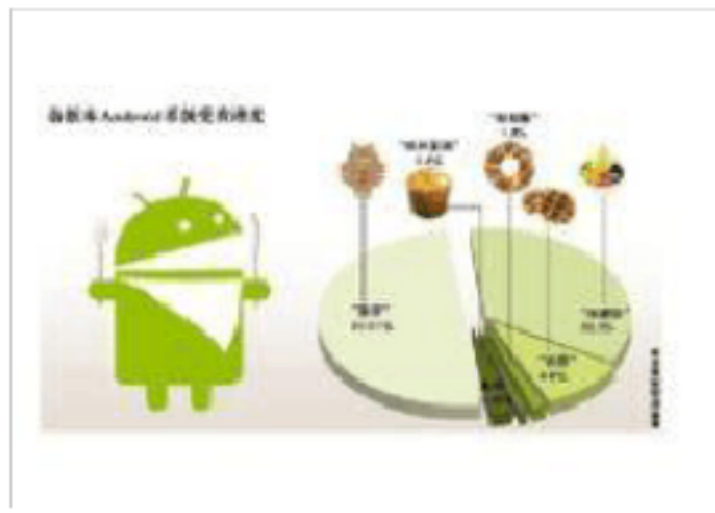
2007 年 11 月 5 日，谷歌公司正式向外界展示了这款名为 Android 的操作系统，并且在这天谷歌宣布建立一个全球性的联盟组织，该组织由 34 家手机制造商、软件开发商、电信运营商以及芯片制造商共同组成，并与 84 家硬件制造商、软件开发商及电信营运商组成开放手持设备联盟 （Open Handset Alliance）来共同研发改良 Android 系统， 这一联盟将支持谷歌发布的手机操作系统以及应用软件， Google 以 Apache 免费开源许可证的授权方式，发布了 Android 的源代码。 [5-6]

2008 年， 在 GoogleI/O 大会上， 谷歌提出了 AndroidHAL 架构图， 在同年 8 月 18 号， Android 获得了美国联邦通信委员会（ FCO 的批准，在 2008 年 9 月，谷歌正式发布了 Android 1.0 系统，这也是 Android 系统最早的版本。

2009 年 4 月，谷歌正式推出了 Android 1.5 这款手机，从 Android 1.5 版本开始，谷歌开始将 Android 的版本以甜品的名字命名， Android 1.5 命名为 Cupcake（纸杯蛋糕）。该系统与 Android 1.0 相比有了很大的改进。

2009 年 9 月份，谷歌发布了 Android 1.6 的正式版，并且推出了搭载 Android 1.6 正式版的手机 HTC Hero（G3），凭借着出色的外观设计以及全新的 Android 1.6 操作系统， HTC Hero（G3）成为当时全球最受欢迎的手机。 Android 1.6 也有一个有趣的甜品名称，它被称为 Donut（甜甜圈）。

2010 年 2 月份， Linux 内核开发者 Greg Kroah-Hartman 将 Android 的驱



安卓各个版本在市场受欢迎度

动程序从 Linux 内核“状态树 ”（“ stagingtree ”）上除去，从此， Android 与 Linux 开发主流将分道扬镳。在同年 5 月份，谷歌正式发布了 Android 2.2 操作系统。谷歌将 Android 2.2 操作系统命名为 Froyo，翻译完名为冻酸奶。

2010 年 10 月份，谷歌宣布 Android 系统达到了第一个里程碑，即电子市场上获得官方数字认证的 Android 应用数量已经达到了 10 万个， Android 系统的应用增长非常迅速。在 2010 年 12 月，谷歌正式发布了 Android 2.3 操作系统 Gingerbread （姜饼）。

2011 年 1 月，谷歌称每日的 Android 设备新用户数量达到了 30 万部，到 2011 年 7 月，这个数字增长到 55 万部，而 Android 系统设备的用户总数达到了 1.35 亿，Android 系统已经成为智能手机领域占有量最高的系统。

2011 年 8 月 2 日，Android 手机已占据全球智能机市场 48%的份额，并在亚太地区市场占据统治地位，终结了 Symbian（塞班系统）的霸主地位，跃居全球第一。

2011 年 9 月份，Android 系统的应用数目已经达到了 48 万，而在智能手机市场，Android 系统的占有率已经达到了 43%。继续在排在移动操作系统首位。谷歌将会发布全新的 Android 4.0 操作系统，这款系统被谷歌命名为 Ice Cream Sandwich（冰激凌三明治）。

2012 年 1 月 6 日，谷歌 Android Market 已有 10 万开发者推出超过 40 万活跃的应用，大多数的应用程序为免费。 Android Market 应用程序商店目录在新年首周周末突破 40 万基准，距离突破 30 万应用仅 4 个月。在 2011 年早些时候，Android Market 从 20 万增加到 30 万应用也花了四个月。 [7]

3 发行版本编辑

测试版本

Android 各代版本 Logo

Android 在正式发行之前，最开始拥有两个内部测试版本，并且以著名的机器人名称来对其进行命名，它们分别是：阿童木（ AndroidBeta），发条机器人（ Android 1.0）。后来由于涉及到版权问题，谷歌将其命名规则变更为用甜点作为它们系统版本的代号的命名方法。甜点命名法开始于 Android 1.5 发布的时候。作为每个版本代表的甜点的尺寸越变越大，然后按照 26 个字母数序：纸杯蛋糕（ Android 1.5），甜甜圈（ Android 1.6），松饼（ Android 2.0/2.1），冻酸奶（Android 2.2），姜饼（Android 2.3），蜂巢（Android 3.0），冰激凌三明治 （Android 4.0），果冻豆（ Jelly Bean，Android4.1 和 Android 4.2）。

1.1

2008 年 9 月发布的 Android 第一版。

1.5

Cupcake（纸杯蛋糕）：2009 年 4 月 30 日发布。

主要的更新如下：

拍摄/播放影片，并支持上传到 Youtube；支持立体声蓝牙耳机，同时改善自动配对性能；最新的采用 WebKit 技术的浏览器，支持复制/贴上和页面中搜索； GPS性能大大提高；提供屏幕虚拟键盘；主屏幕增加音乐播放器和相框 widgets；应用程序自动随着手机旋转；短信、Gmail、日历，浏览器的用户接口大幅改进，如 Gmail 可以批量删除邮件；相机启动速度加快，拍摄图片可以直接上传到 Picasa；来电照片显示。

1.6

Donut（甜甜圈）：2009 年 9 月 15 日发布。
主要的更新如下：
重新设计的 Android Market 手势；支持 CDMA 网络；文字转语音系统（ Text-to-Speech）；快速搜索框；全新的拍照接口；查看应用程序耗电；支持虚拟私人网络（ VPN）；支持更多的屏幕分辨率；支持 OpenCore2 媒体引擎；新增面向视觉或听觉困难人群的易用性插件。

2.0

2009 年 10 月 26 日发布。
主要的更新如下：
优化硬件速度； "Car Home" 程序；支持更多的屏幕分辨率；改良的用户界面；新的浏览器的用户接口和支持 HTML5 新的联系人名单； 更好的白色 /黑色背景比率； 改进 Google Maps3.1.2 ；支持 Microsoft Exchange；支持内置相机闪光灯；支持数码变焦；改进的虚拟键盘；支持蓝牙 2.1；支持动态桌面的设计。

Android 2.2/2.2.1 Froyo （冻酸奶） ：2010 年 5 月 20 日发布。主要的更新如下：
整体性能大幅度的提升； 3G 网络共享功能； Flash 的支持； App2sd 功能；全新的软件商店；更多的 Web 应用 API 接口的开发。

2.3.x

Gingerbread（姜饼） ：2010 年 12 月 7 日发布。
主要的更新如下：
增加了新的垃圾回收和优化处理事件；原生代码可直接存取输入和感应器事件、EGL/OpenGLES, OpenSL ES;新的管理窗口和生命周期的框架； 支持 VP8 和 WebM 视频格式, 提供 AAC和 AMR 宽频编码, 提供了新的音频效果器； 支持前置摄像头、 SIP/VOIP和 NFC(近场通讯）；简化界面、速度提升；更快更直观的文字输入；一键文字选择和复制 /粘帖；改进的电源管理系统；新的应用管理方式。

3.0

Honeycomb（蜂巢）：2011 年 2 月 2 日发布。
主要更新如下：
优化针对平板 ；全新设计的 UI 增强网页浏览功能 ；n-app purchases 功能。

3.1

Honeycomb（蜂巢） ：2011 年 5 月 11 日布发布。
版本主要更新如下：
经过优化的 Gmail 电子邮箱 ；全面支持 Google Maps ；将 Android 手机系统跟平板系统再次合并从而方便开发者；任务管理器可滚动, 支持 USB输入设备（键盘、鼠标等） ；支持 Google TV.可以支持 XBOX 360无线手柄； widget 支持的变化, 能更加容易的定制屏幕 widget 插件。

3.2

Honeycomb（蜂巢）：2011 年 7 月 13 日发布。
版本更新如下：
支持 7 英寸设备；引入了应用显示缩放功能。

4.0

Ice Cream Sandwich（冰激凌三明治） ：2011 年 10 月 19 日在香港发布。

安卓 2.0 版本

版本主要更新如下：

全新的 UI；全新的 Chrome Lite 浏览器，有离线阅读， 16 标签页，隐身浏览模式等；截图功能；更强大的图片编辑功能；自带照片应用堪比 Instagram，可以加滤镜、加相框，进行 360 度全景拍摄，照片还能根据地点来排序； Gmail 加入手势、离线搜索功能， UI 更强大；新功能 People：以联系人照片为核心， 界面偏重滑动而非点击， 集成了 Twitter、Linkedin、Google+ 等通讯工具。 有望支持用户自定义添加第三方服务； 新增流量管理工具， 可具体查看每个应用产生的流量，限制使用流量，到达设置标准后自动断开网络。

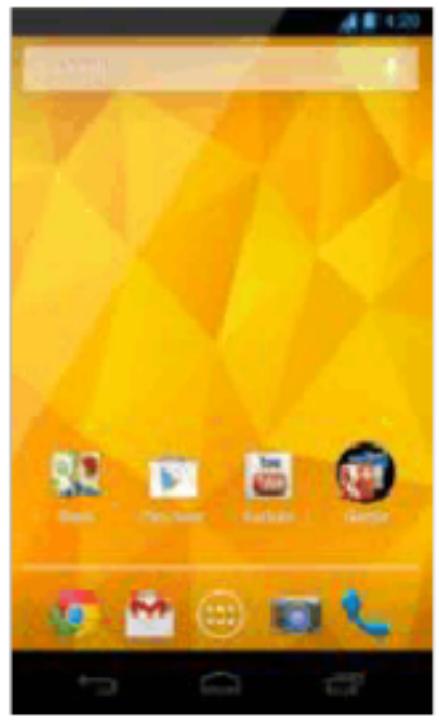
4.1

Android 4.1Jelly Bean（果冻豆） :2012 年 6 月 28 日

新特性：

更快、更流畅、更灵敏；特效动画的帧速提高至 60fps，增加了三倍缓冲；增强通知栏；全新搜索；搜索将会带来全新的 UI、智能语音搜索和 Google Now 三项新功能；桌面插件自动调整大小；加强无障碍操作；语言和输入法扩展；新的输入类型和功能；新的连接类型。

4.2



Android 4.2 Jelly Bean 原生系统用户界面

Android 4.2Jelly Bean（果冻豆） :2012 年 10 月 30 日

Android 4.2 沿用‘果冻豆’这一名称，以反映这种最新操作系统与 Android 4.1 的相似性，但 Android 4.2 推出了一些重大的新特性，具体如下：

Photo Sphere 全景拍照功能；键盘手势输入功能；改进锁屏功能，包括锁屏状态下支持桌面挂件和直接打开照相功能等； 可扩展通知， 允许用户直接打开应用； Gmail 邮件可缩放显示；Daydream 屏幕保护程序；用户连点三次可放大整个显示频，还可用两根手指进行旋转和缩放显示， 以及专为盲人用户设计的语音输出和手势模式导航功能等； 支持 Miracast 无线显示共享功能； Google Now 现可允许用户使用 Gamail 作为新的数据来源，如改进后的航班追踪功能、酒店和餐厅预订功能以及音乐和电影推荐功能等。

4.4

Android 4.4KitKat （奇巧巧克力） ：待定

2013 年 9 月 4 日凌晨， 谷歌对外公布了 Android 新版本 Android 4.4KitKat（奇巧巧克力），并且于 2013 年 11 月 01 日正式发布，新的 4.4 系统更加整合了自家服务，力求防止安卓系统继续碎片化、分散化。 [8]

4 系统架构编辑

Android 的系统架构和其操作系统一样，采用了分层的架构。

Android 结构

从架构图看， Android 分为四个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和 Linux 内核层。

应用程序

Android 会同一系列核心应用程序包一起发布， 该应用程序包包括客户端， SMS 短消息程序，日历，地图，浏览器，联系人管理程序等。所有的应用程序都是使用 JAVA语言编写的。

应用程序框架

开发人员也可以完全访问核心应用程序所使用的 API 框架。该应用程序的架构设计简化了组件的重用 ;任何一个应用程序都可以发布它的功能块并且任何其它的应用程序都可以使用其所发布的功能块（不过得遵循框架的安全性） 。同样，该应用程序重用机制也使用户可以方便的替换程序组件。

隐藏在每个应用后面的是一系列的服务和系统 ，其中包括 ；

丰富而又可扩展的视图 （Views)，可以用来构建应用程序， 它包括列表 （ Lists)，网格（Grids)，文本框（ Text boxes)，按钮（ Buttons)， 甚至可嵌入的 web 浏览器。

内容提供器（ Content Providers) 使得应用程序可以访问另一个应用程序的数据（如联系人数据库)， 或者共享它们自己的数据

资源管理器（ Resource Manager) 提供 非代码资源的访问，如本地字符串，图形，和布局文件（ Layout files)。

通知管理器 （Notification Manager) 使得应用程序可以在状态栏中显示自定义的提示信息。

活动管理器（ Activity Manager) 用来管理应用程序生命周期并提供常用的导航回退功能。

系统运行库

Android 包含一些 C/C++库，这些库能被 Android 系统中不同的组件使用。 它们通过 Android 应用程序框架为开发者提供服务。以下是一些核心库：

* 系统 C 库 - 一个从 BSD继承来的标准 C 系统函数库 Libc)， 它是专门为基于 Embedded linux 的设备定制的。

* 媒体库 - 基于 PacketVideo OpenCORE;该库支持多种常用的音频、视频格式回放和录制，同时支持静态图像文件。编码格式包括 MPEG4, H.264, MP3, AAC, AMR, JPG, PNG。

* Surface Manager - 对显示子系统的管理，并且为多个应用程序提 供了 2D 和 3D 图层的无缝融合。

* LibWebCore - 一个最新的 web 浏览器引擎用， 支持 Android 浏览器和一个可嵌入的 web 视

图。

5 系统结构编辑

系统内核

Android 是运行于 Linux kernel 之上，但并不是 GNU/Linux。因为在一般 GNU/Linux 里支持的功能，Android 大都没有支持，包括 Cairo、X11、Alsa、FFmpeg、GTK、Pango 及 Glibc 等都被移除掉了。Android 又以 Bionic 取代 Glibc、以 Skia 取代 Cairo、再以 opencore 取代 FFmpeg 等等。Android 为了达到商业应用，必须移除被 GNU GPL授权证所约束的部份，例如 Android 将驱动程序移到 Userspace，使得 Linux driver 与 Linux kernel 彻底分开。Bionic/Libc/Kernel/ 并非标准的 Kernel header files。Android 的 Kernel header 是利用工具由 Linux Kernel header 所产生的，这样做是为了保留常数、数据结构与宏。

Android 的 Linux kernel 控制包括安全（Security），存储器管理（Memory Management），程序管理（Process Management），网络堆栈（Network Stack），驱动程序模型（Driver Model）等。下载 Android 源码之前，先要安装其构建工具 Repo 来初始化源码。Repo 是 Android 用来辅助 Git 工作的一个工具。

后缀简介

[9] APK 是安卓应用的后缀，是 AndroidPackage 的缩写，即 Android 安装包 (apk)。APK是类似 Symbian Sis或 Sisx 的文件格式。通过将 APK文件直接传到 Android 模拟器或 Android 手机中执行即可安装。apk 文件和 sis 一样，把 android sdk 编译的工程打包成一个安装程序文件，格式为 apk。APK 文件其实是 zip 格式，但后缀名被修改为 apk，通过 UnZip 解压后，可以看到 Dex 文件，Dex 是 Dalvik VM executes 的全称，即 Android Dalvik 执行程序，并非 Java ME 的字节码而是 Dalvik 字节码。

[9] APK 文件结构

一个 APK 文件结构为：

1. META-INF\ （注： Jar 文件中常可以看到）；
2. res\ (注：存放资源文件的目录)；
3. AndroidManifest.xml (注：程序全局配置文件)；
4. classes.dex （注： Dalvik 字节码）；
5. resources.arsc (注：编译后的二进制资源文件)。

总结下我们发现 Android 在运行一个程序时首先需要 UnZip，然后类似 Symbian 那样直接执行安装，和 Windows Mobile 中的 PE文件有区别，这样做对于程序的保密性和可靠性不是很高，通过 dexdump 命令可以反编译，但这样做符合发展规律，微软的 Windows Gadgets 或者说 WPF 也采用了这种构架方式。

在 Android 平台中 dalvik vm 的执行文件被打包为 apk 格式，最终运行时加载器会解压然后获取编译后 androidmanifest.xml 文件中的 permission 分支相关的安全访问，但仍然存在很多安全限制，如果你将 apk 文件传到 /system/app 文件夹下会发现执行是不受限制的。

最终我们平时安装的文件可能不是这个文件夹，而在 android rom 中系统的 apk 文件默认会放入这个文件夹，它们拥有着 root 权限。

硬件抽像层

Android 的 HAL（硬件抽像层）是能以封闭源码形式提供硬件驱动模块。HAL 的目的是为了把 Android framework 与 Linux kernel 隔开，让 Android 不至过度依赖 Linux kernel，以达成 Kernel independent 的概念，也让 Android framework 的开发能在不考量驱动程序实现的前提下进行发展。

HAL stub 是一种代理人（Proxy）的概念，Stub 是以 *.so 档的形式存在。Stub 向 HAL 'i提供 '操作函数（Operations），并由 Android runtime 向 HAL 取得 Stub 的 Operations，再

Callback 这些操作函数。 HAL 里包含了许多的 Stub（代理人）。Runtime 只要说明 “类型 ”，即 Module ID，就可以取得操作函数。

中介软件

操作系统与应用程序的沟通桥梁， 应用分为两层： 函数层（Library)和虚拟机（Virtual Machine ）。Bionic 是 Android 改良 libc 的版本。Android 同时包含了 Webkit，所谓的 Webkit 就是 Apple Safari 浏览器背后的引擎。 Surface flinger 是就 2D 或 3D 的内容显示到屏幕上。 Android 使用工具链（ Toolchain)为 Google 自制的 Bionic Libc。

Android 采用 OpenCORE作为基础多媒体框架。 Open CORE可分 7 大块：PVPlayer、PVAuthor、Codec、PacketVideo Multimedia Framework （PVMF)、Operating System Compatibility Library （OSCL)、Common、OpenMAX。

Android 使用 skia 为核心图形引擎，搭配 OpenGL/ES。skia 与 Linux Cairo 功能相当，但相较于 Linux Cairo, skia 功能还只是雏形的。 2005 年 Skia 公司被 Google 收购， 2007 年初， Skia GL 源码被公开， Skia 也是 Google Chrome 的图形引擎。

Android 的多媒体数据库采用 SQLite 数据库系统。数据库又分为共用数据库及私用数据库。用户可通过 ContentResolver 类（ Column）取得共用数据库。

Android 的中间层多以 Java 实现，并且采用特殊的 Dalvik 虚拟机（ Dalvik Virtual Machine ）。Dalvik 虚拟机是一种 “暂存器型态 ”（Register Based) 的 Java虚拟机，变量皆存放于暂存器中，虚拟机的指令相对减少。

Dalvik 虚拟机可以有多个实例 （Instance)， 每个 Android 应用程序都用一个自属的 Dalvik 虚拟机来运行，让系统在运行程序时可达到优化。 Dalvik 虚拟机并非运行 Java 字节码（Bytecode），而是运行一种称为 .dex 格式的文件。

安全权限机制

Android 本身是一个权限分立的操作系统。在这类操作系统中，每个应用都以唯一的一个系统识别身份运行 （Linux 用户 ID 与群组 ID)。系统的各部分也分别使用各自独立的识别方式。Linux 就是这样将应用与应用，应用与系统隔离开。

系统更多的安全功能通过权限机制提供。 权限可以限制某个特定进程的特定操作， 也可以限制每个 URI 权限对特定数据段的访问。

Android 安全架构的核心设计思想是，在默认设置下，所有应用都没有权限对其他应用、系统或用户进行较大影响的操作。这其中包括读写用户隐私数据（联系人或电子邮件） ，读写其他应用文件，访问网络或阻止设备待机等。

安装应用时， 在检查程序签名提及的权限， 且经过用户确认后， 软件包安装器会给予应用权限。从用户角度看，一款 Android 应用通常会要求如下的权限：

拨打电话、发送短信或彩信、修改 / 删除 SD 卡上的内容、读取联系人的信息、读取日程信的息，写入日程数据、读取电话状态或识别码、精确的（基于 GPS) 地理位置、模糊的（基于网络获取）地理位置、创建蓝牙连接、对互联网的完全访问、查看网络状态，查看 WiFi 状态、避免手机待机、修改系统全局设置、读取同步设定、开机自启动、重启其他应用、终止运行中的应用、设定偏好应用、震动控制、拍摄图片等。

一款应用应该根据自身提供的功能，要求合理的权限。用户也可以分析一款应用所需权限，从而简单判定这款应用是否安全。 如一款应用是不带广告的单机版， 也没有任何附加的内容需要下载，那么它要求访问网络的权限就比较可疑。

6 应用组件编辑

Android 开发四大组件分别是：活动（ Activity) ： 用于表现功能。服务（ Service)： 后台运行服务，不提供界面呈现。 广播接收器 （BroadcastReceiver) :用于接收广播。 内容提供商 （Content Provider）： 支持在多个应用中存储和读取数据，相当于数据库。

活动

Android 中， Activity 是所有程序的根本，所有程序的流程都运行在 Activity 之中， Activity 可以算是开发者遇到的最频繁，也是 Android 当中最基本的模块之一。在 Android 的程序当中， Activity 一般代表手机屏幕的一屏。如果把手机比作一个浏览器，那么 Activity 就相当于一个网页。 在 Activity 当中可以添加一些 Button 、Check box 等控件。 可以看到 Activity 概念和网页的概念相当类似。

一般一个 Android 应用是由多个 Activity 组成的。这多个 Activity 之间

Android 的应用 (20 张)

可以进行相互跳转，例如，按下一个 Button 按钮后，可能会跳转到其他的 Activity 。和网页跳转稍微有些不一样的是， Activity 之间的跳转有可能返回值，例如，从 Activity A 跳转到 Activity B ，那么当 Activity B 运行结束的时候， 有可能会给 Activity A 一个返回值。 这样做在很多时候是相当方便的。

当打开一个新的屏幕时， 之前一个屏幕会被置为暂停状态， 并且压入历史堆栈中。 用户可以通过回退操作返回到以前打开过的屏幕。 可以选择性的移除一些没有必要保留的屏幕， 因为 Android 会把每个应用的开始到当前的每个屏幕保存在堆栈中。

（Android 的应用图册图片来源： [10] ）

服务

Service 是 android 系统中的一种组件，

安卓不同版本图片

它跟 Activity 的级别差不多，但是他不能自己运行，只能后台运行，并且可以和其他组件进行交互。 Service 是没有界面的长生命周期的代码。 Service 是一种程序，它可以运行很长时间，但是它却没有用户界面。这么说有点枯燥，来看个例子。打开一个音乐播放器的程序，这个时候若想上网了，那么，打开 Android 浏览器，这个时候虽然已经进入了浏览器这个程序，但是， 歌曲播放并没有停止，而是在后台继续一首接着一首的播放。其实这个播放就是由播放音乐的 Service 进行控制。当然这个播放音乐的 Service 也可以停止，例如，当播放列表里边的歌曲都结束，或者用户按下了停止音乐播放的快捷键等。 Service 可以在和多场合的应用中使用，比如播放多媒体的时候用户启动了其他 Activity 这个时候程序要在后台继续播放，比如检测 SD 卡上文件的变化，再或者在后台记录地理信息位置的改变等等，总之服务嘛，总是藏在后头的。

开启 Service 有两种方式 ：

（1） Context.startService（）：Service 会经历 onCreate -> onStart（如果 Service 还没有运行，则 android 先调用 onCreate（）然后调用 onStart（）；如果 Service 已经运行， 则只调用 onStart（），所以一个 Service 的 onStart 方法可能会重复调用多次 ）；StopService 的时候直接 onDestroy，如果是调用者自己直接退出而没有调用 StopService 的话， Service 会一直在后台运行。该 Service 的调用者再启动起来后可以通过 stopService 关闭 Service。 注意，多次调用 Context.startservice （）不会嵌套（即使会有相应的 onStart（）方法被调用），所以无论同一个服务被启动了多少次，一旦调用 Context.stopService（）或者 StopSelf（），他都会被停止。补充说明：传递给 StartService（0 的 Intent 对象会传递给 onStart （）方法。调用顺序为： onCreate --> onStart （可多次调用) --> onDestroy。

（2） Context.bindService （）：Service 会经历 onCreate（） -->onBind （）， onBind 将返回给客户端一个 IBind 接口实例， IBind 允许客户端回调服务的方法，比如得到 Service 运行的状态或其他操作。 这个时候把调用者 （Context，例如 Activity ）会和 Service 绑定在一起， Context 退出了，Srevice 就会调用 onUnbind --> onDestroyed 相应退出， 所谓绑定在一起就共存亡了。

广播接收器

在 Android 中， Broadcast 是一种广泛运用的在应用程序之间传输信息的机制。而 BroadcastReceiver 是对发送出来的 Broadcast 进行过滤接受并响应的一类组件。可以使用 BroadcastReceiver 来让应用对一个外部的事件做出响应。 这是非常有意思的， 例如， 当电话呼入这个外部事件到来的时候，可以利用 BroadcastReceiver 进行处理。例如，当下载一个程序成功完成的时候， 仍然可以利用 BroadcastReceiver 进行处理。 BroadcastReceiver 不能生成 UI，也就是说对于用户来说不是透明的，用户是看不到的。 BroadcastReceiver 通过 NotificationManager 来通知用户这些事情发生了。 BroadcastReceiver 既可以在 AndroidManifest.xml 中注册，也可以在运行时的代码中使用 Context.registerReceiver （)进行注册。 只要是注册了，当事件来临的时候，即使程序没有启动，系统也在需要的时候启动程序。各种应用还可以通过使用 Context.sendBroadcast （） 将它们自己的 Intent Broadcasts 广播给其他应用程序。

内容提供

Content Provider 是 Android 提供的第三方应用数据的访问方案。

在 Android[11] 中，对数据的保护是很严密的， 除了放在 SD 卡中的数据， 一个应用所持有的数据库、文件等内容，都是不允许其他直接访问的。 Andorid 当然不会真的把每个应用都做成一座孤岛，它为所有应用都准备了一扇窗，这就是 Content Provider。应用想对外提供的数据，可以通过派生 Content Provider 类， 封装成一枚 Content Provider , 每个 Content Provider 都用一个 uri 作为独立的标识，形如： content://com.xxxxx 。所有东西看着像 REST的样子，但实际上，它比 REST更为灵活。和 REST类似， uri 也可以有两种类型，一种是带 id 的，另一种是列表的， 但实现者不需要按照这个模式来做， 给 id 的 uri 也可以返回列表类型的数据，只要调用者明白，就无妨，不用苛求所谓的 REST。

7 平台优势编辑

开放性

在优势方面， Android 平台首先就是其开发性，开发的平台允许任何移动终端厂商加入到 Android 联盟中来。显著的开放性可以使其拥有更多的开发者， 随着用户和应用的日益丰富，一个崭新的平台也将很快走向成熟。

开发性对于 Android 的发展而言，有利于积累人气，这里的人气包括消费者和厂商，而对于消费者来讲，最大的受益正是丰富的软件资源。开放的平台也会带来更大竞争，如此一来，消费者将可以用更低的价位购得心仪的手机。

不受束缚

在过去很长的一段时间，特别是在欧美地区，手机应用往往受到运营商制约，使用什么功能接入什么网络，几乎都受到运营商的控制。自从 2007 年 iPhone 上市后，用户可以更加方便地连接网络，运营商的制约减少。随着 EDGE、HSDPA 这些 2G 至 3G 移动网络的逐步过渡和提升，手机随意接入网络已不是运营商口中的笑谈。

丰富的硬件

这一点还是与 Android 平台的开放性相关，由于 Android 的开放性，众多的厂商会推出千奇百怪，功能特色各具的多种产品。功能上的差异和特色，却不会影响到数据同步、甚至软件的兼容，如同从诺基亚 Symbian 风格手机一下改用苹果 iPhone，同时还可将 Symbian 中优秀的软件带到 iPhone 上使用、联系人等资料更是可以方便地转移。

方便开发

Android 平台提供给第三方开发商一个十分宽泛、自由的环境，不会受到各种条条框框的阻扰，可想而知，会有多少新颖别致的软件会诞生。但也有其两面性，血腥、暴力、情色方面的程序和游戏如何控制正是留给 Android 难题之一。

Google 应用

在互联网的 Google 已经走过 10 年度历史，从搜索巨人到全面的互联网渗透，Google 服务如地图、邮件、搜索等已经成为连接用户和互联网的重要纽带，而 Android 平台手机将无缝结合这些优秀的 Google 服务。