Android 实战基础教程

湖南爱博思科技有限责任公司
刘群 刘文 刘军 赵蒙 喻旅游 编著

西安电子科技大学出版社
http://www.xduph.com

Android实战基础教程

湖南爱博思科技有限责任公司
刘群　刘文　刘军　赵蒙　喻旅游　编著

西安电子科技大学出版社

内 容 简 介

本书主要介绍了 Android 开发过程中常用的知识点，包括四大组件中的 Activity、Service、Broadcast Receiver 以及自定义控件，详细介绍了 Android 开发过程中的重点难点，并给出了三个实际案例。

本书适用于有 Java 编程基础的学习者，可作为高等学校、高职高专及相关培训机构的教材。

图书在版编目(CIP)数据

Android 实战基础教程/湖南爱博思科技有限责任公司等编著.
—西安：西安电子科技大学出版社，2016.11
ISBN 978-7-5606-4338-0

Ⅰ. ① A… Ⅱ. ① 湖… Ⅲ. ① 移动终端—应用程序—程序设计—教材 Ⅳ. ① TN929.53

中国版本图书馆 CIP 数据核字(2016)第 263774 号

策　　划　杨丕勇
责任编辑　杨丕勇
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西天意印务有限责任公司
版　　次　2016 年 11 月第 1 版　2016 年 11 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印 张 15.25
字　　数　362 千字
印　　数　1～2000 册
定　　价　32.00 元

ISBN 978 - 7 - 5606 - 4338 - 0/TN

XDUP 4630001-1

前　言

移动互联网已经成为当今世界发展最快、市场潜力最大、前景最诱人的领域，而 Android 则是移动互联网上市场占有率最高的平台。

Android 技术应用范围非常广泛，智能手机、智能终端等越来越多的智能设备都采用了 Android 技术。科技的发展，使得 Android 技术应用领域迅速扩张，市场对于 Android 开发人员的需求也呈爆炸式增长。本书是为具有一定基础的 Android 技术人员编写的，主要介绍在实际开发过程中常见的知识要点，并结合这些知识要点引入相关案例，以便增加学习者的学习兴趣和学习能力。

由于编者水平有限，书中难免有不足之处，敬请专家和读者批评指正。

编　者

2016 年 5 月

目　录

第 1 章　Android 应用开发环境

Android 系统已经成为全球应用最广泛的手机操作系统，三星、摩托罗拉、HTC 等手机厂商早已通过 Android 取得了巨大成功。目前国内对 Android 开发人才的需求也在迅速增长，而且搭载 Android 系统的手机越来越不像“手机”，更像是一台小型计算机，因此手机软件必将在未来 IT 行业中具有举足轻重的地位——你不可能带着一台电脑到处跑，而且时时开着机，但手机可以做到。从发展趋势上看，Android 软件人才的需求会越来越大。

本书所介绍的平台是 Android4.2 平台，该版本的 Android 平台经过几年的沉淀，不仅功能强大，而且高效、稳定。本章主要介绍 Android 应用开发环境，既包括 Android SDK、Android 开发工具的安装步骤，也包括 Android 提供的 ADB、DDMS、AAPT、DX 等工具的使用方法。这些工具是开发 Android 应用的基础技能。

1.1　Android 的发展和历史

Android 是由 Android 公司的创始人 Andy Rubin 创立的一个手机操作系统，后来该公司被 Google 收购，而 Andy Rubin 也成为 Google 公司的 Android 产品负责人。Google 希望与各方共同建立一个标准化、开放式的移动电话软件平台，从而在移动产业内形成一个开放式的操作平台。

1.1.1　Android 的发展和简介

Android 1.0 手机操作系统是 Google 于 2007 年 11 月 5 日发布的，这个版本的 Android 系统并没有赢得市场的广泛支持。

2009 年 5 月，Google 发布了 Android1.4，该版本提供了一个十分“豪华”的用户界面，而且提供了蓝牙连接支持。这个版本的 Android 吸引了大量开发者的目光。接下来，Android 版本更新得较快，目前最新的 Android 版本是 7.0。

目前 Android 已成为一个重要的手机操作系统，除此之外，市场上常见的其他手机操作系统还有：

- iOS：Apple 公司的手机、平板操作系统，市场占有率较高。
- WindowsPhone：Microsoft 公司的手机操作系统，2012 年发布的最新版本为 WindowsPhone 8，但应用前景依然不够明朗。
- Symbian：已经放弃，基本被淘汰。

- BlackBerry：即将被淘汰。

从 2008 年 9 月 22 日，T-Mobile 在纽约正式发布第一款 Android 手机——T-Mobile G1 开始，Android 系统受到各个手机厂商的青睐。

2010 年 1 月 7 日，Google 在其美国总部正式向外界发布了旗下首款合作品牌手机 Nexus One(HTC G5)，同时开始对外发售。

目前 Android 系统的市场占有率已经远超 iOS。WindowsPhone 作为最后的“赌注”，Microsoft 自然是全力以赴，希望至少能够与 iOS、Android 三足鼎立，但目前局势似乎并不乐观。因而无论从哪个角度来讲，Android 都已成为最主流的手机操作系统。

目前国内手机厂商主要生产 Android 操作系统的手机，因为 Android 手机平台是一个真正开放式的平台，无需支付任何费用即可以使用。出于自身研发费用的考虑，不管是对于知名手机生产厂商，还是大量的“山寨”手机厂商，Android 操作平台都是一个不错的选择。

目前，已发布搭载 Android 系统的主要手机厂商包括摩托罗拉、三星、HTC、索尼爱立信、LG、华为、联想、中兴等。

1.1.2 Android 平台架构及其特性

Android 系统的底层建立在 Linux 系统之上，该平台由操作系统、中间件、用户界面和应用软件 4 层组成，采用一种被称为软件叠层(SoftwareStack)的方式进行构建。软件叠层结构使得层与层之间相互分离，各层有明确的分工，这种分工保证了层与层之间的低耦合，当下层的层内或层下发生改变时，上层应用程序无需任何改变。

图 1.1 显示了 Android 系统的体系结构。

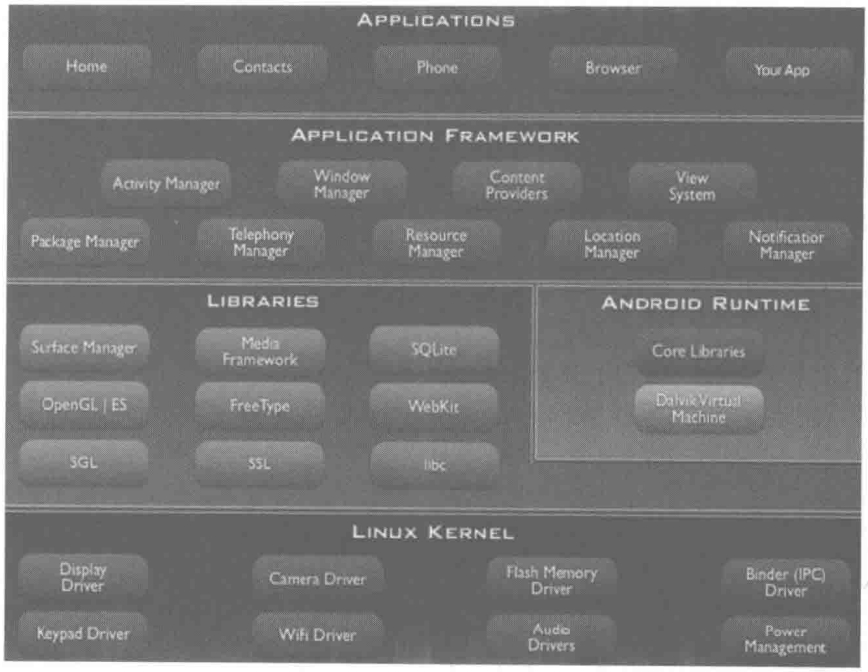

图 1.1

从图 1.1 中可以看出，Android 系统主要由 5 部分组成，下面分别对这 5 部分进行简单介绍。

1．应用程序层(APPLICATIONS)

Android 系统的核心应用程序，包括电子邮件客户端、SMS 程序、日历、地图、浏览器、联系人等。这些应用程序都是用 Java 编写的。本书所要介绍的主要内容就是如何编写 Android 系统上的应用程序。

2．应用程序框架(APPLICATION FRAMEWORK)

当我们开发 Android 应用程序时，就是面向对象的应用程序框架进行的。从这个意义上看，Android 系统上的应用程序是完全平等的，不论是 Android 系统提供的程序，还是普通开发者提供的程序，都可以访问 Android 提供的 API 框架。

Android 应用程序框架提供了大量 API 供开发者使用，关于这些 API 的具体功能和用法将在本书后面详细介绍，此处不再展开阐述。

应用程序框架除了可以作为应用程序开发的基础之外，也是软件复用的重要手段，任何一个应用程序都可以发布它的功能模块——只要发布时遵守了框架的约定，其他应用程序都可以使用这个功能模块。

3．函数库(LIBRARIES)

Android 包含了一套被不同组件所用的 C/C++库的集合。一般来说，Android 应用开发者不能直接调用这套 C/C++库集，但可以通过它上面的应用程序框架来调用这些库。

下面列出一些核心库。

- 系统 C 库：一个从 BSD 系统派生出来的标准 C 系统库(libc)，并且专门为嵌入式 Linux 设备调整过。
- 媒体库：基于 PacketVideo 的 OpenCore，这套媒体库支持播放和录制许多流行的音频和视频格式，以及查看静态图片，主要包括 MPEG4、H.264、MP3、AAC、AMR、JPG、PNG 等多媒体格式。
- SurfaceManager：管理对于现实子系统的访问，并可以对多个应用系统的 2D 和 3D 图层机提供无缝整合。
- LibWebCore：一个全新的 Web 浏览器引擎，该引擎为 Android 浏览器提供支持，也为 WebView 提供支持，WebView 可以完全嵌入开发者自己的应用程序中。本书后面会有关于 WebView 的介绍。
- SGL：底层的 2D 图形引擎。
- 3Dlibraries：基于 OpenGLES 1.0 API 实现的 3D 系统，该套 3D 库既可以使用硬件 3D 加速(如果硬件系统支持)，也可以使用高度优化的软件 3D 加速。
- FreeTye：位图和向量字体显示。
- SQLite：供所有应用程序使用的、功能强大的轻量级关系数据库。

4．Android 运行时(ANDROID RUNTIME)

Android 运行时由两部分组成：Android 核心库集和 Dalvik 虚拟机。其中核心库集提供了 Java 语言核心库所能使用的绝大部分功能，而虚拟机则负责运行 Android 应用程序。

提示：Android 运行时和 JRE 有点类似。就像疯狂 Java 体系的《疯狂 Java 讲义》一书中解释的，JRE 包括 JVM 和其他功能函数库；而此处的 Android 运行时则包括 Dalvik 虚拟机和核心库集。

每个 Android 应用程序都运行在单独的 Dalvik 虚拟机内(即每个 Android 应用程序对应一条 Davlik 进程)，Dalvik 专门针对同时高效地运行多个虚拟机进行了优化，因此 Android 系统已方便地实现对应用程序进行隔离。

由于 Android 应用程序的编写语言是 Java，因此有些人会把 Dalvik 虚拟机和 JVM 搞混，但实际上二者是有区别的：Dalvik 并未完全遵守 JVM 规范，两者也不兼容。实际上，JVM 虚拟机运行的是 Java 字节码(通常就是.class 文件)，但 Dalvik 运行的是其专有的 dex(Dalvik Executable)文件。JVM 直接从.class 文件或 JRE 包中加载字节码然后运行，而 Dalvik 则无法直接从.class 文件或 JRE 包中加载字节码，它需要通过 DX 工具将应用程序的所有.class 文件编译成.dex 文件再运行。

Dalvik 虚拟机非常适合在移动终端上使用，相对于 PC 或服务器上运行的虚拟机而言，Dalvik 虚拟机不需要很快的 CPU 计算速度和大量的内存空间，它主要有如下几个特点。

- 运行专有的.dex 文件。专有的.dex 文件减少了.class 文件中的冗余信息，而且会把所有.class 文件整合到一个文件中，从而提高运行性能；而且 DX 工具会对.dex 文件进行一些性能优化。
- 基于寄存器实现。大多数虚拟机(包括 JVM)都是基于栈的，而 Dalvik 虚拟机则是基于寄存器的。一般来说，基于寄存器的虚拟机具有更好的性能表现，但在硬件通用性上略差。
- Dalvik 虚拟机依赖于 Linux 内核提供的核心功能，如线程和底层内存管理。

5. Linux 内核(LINUX KERNEL)

Android 系统建立在 Linux2.6 之上。Linux 内核提供了安全性、内存管理、进程管理、网络协议栈和驱动模型等核心系统服务。除此之外，Linux 内核也是系统硬件和软件叠层之间的抽象层。

1.2 搭建 Android 开发环境

在开始搭建 Android 开发环境之前，笔者假定读者已经具有一定的 Java 编程基础，像 JDK 安装、环境设置之类的入门知识不在本书介绍范围内。如果读者暂时还不会这些知识，建议先学习 Java 入门知识。

下面将从 Android SDK 的安装开始，详细说明 Android 开发、调试环境的安装和使用，这些内容是 Android 开发的基础。

1.2.1 下载和安装 Android SDK

Android 的官方网站是 http://www.android.com，登录该站点即可下载 Android SDK。下载和安装 Android SDK 的步骤如下：

(1) 登录 http://developer.android.com/sdk/index.html 页面，点击最下方的 DOWNLOAD FOROTHER PLATFORMS 链接。

(2) 找到页面上的“android-sdk_r21-windows.zip”链接，通过该链接即可下载 Android 4.2 SDK 压缩包。

(3) 下载完成后得到一个 android-sdk_r21-windows.zip 文件，将该文件解压缩到任意路径下。解压缩后得到一个 android-sdk-windows 文件夹，该文件夹下包含如下文件结构：

- Add-ons：该目录下存放第三方公司为 Android 平台开发的附加功能系统。刚解压缩时该目录为空。
- Platforms：该目录下存放不同版本的 Android 系统。刚解压缩时该目录为空。
- Tools：该目录存放了大量 Android 开发、调试的工具。
- AVDManager.exe：该程序是 AVD(Android 虚拟设备)管理器。通过该工具可以管理 AVD。
- SDK Manager.exe：该程序就是 Android SDK 管理器。通过该工具可以管理 Android SDK。

(4) 启动 SDK Manager.exe，即可看到如图 1.2 所示窗口。

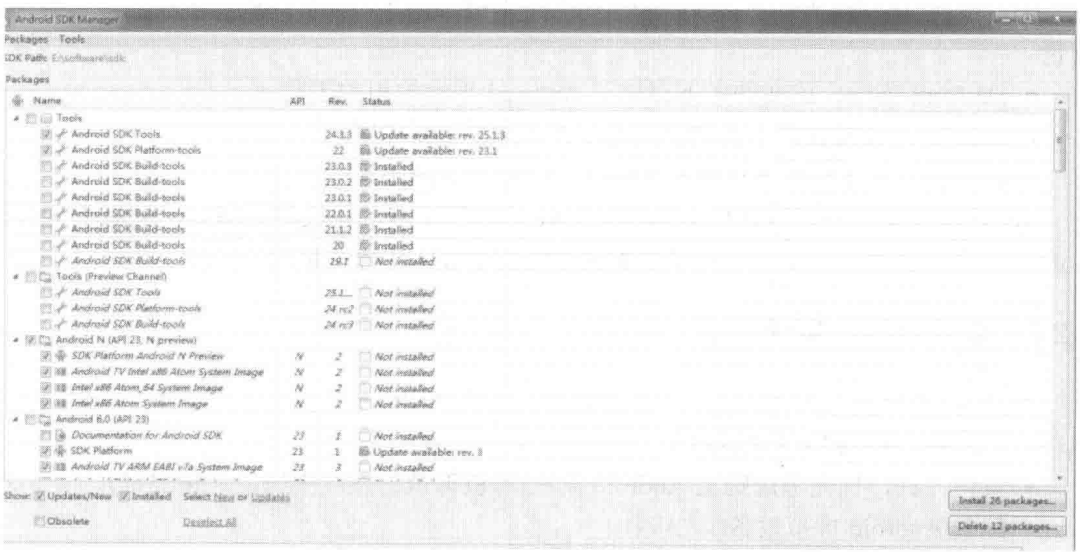

图 1.2

(5) 在图 1.2 所示窗口左侧的列表中勾选需要安装的平台和工具，比如 Android 4.2 的工具和平台，其中 Android 文档、SDK Platform 是必选的。如果想查看 Android 官方提供的示例程序、使用 Android SDK 的源代码，则可以勾选“Samples for SDK”和“Sources for Android SDK”两个列表项。至于是否需要安装 Android 早期版本的 SDK，则取决于读者喜好。选中所需要安装的工具之后，点击“Install Selected”按钮，将出现如图 1.3 所示窗口。

(6) 单击图 1.3 所示窗口的“Accept”单选按钮——确认需要安装所有的工具包，然后单击“Install”按钮，系统开始在线安装 Android SDK 及其相关工具。安装时间取决于读者

的网络状态及选中的工具包的数量，在线安装时间不会太短，甚至可能花费一两个小时，耐心等待即可。

图 1.3

(7) 安装完成后将可以看到在 Android SDK 目录下增加了如下几个文件夹：

- docs：该文件夹下存放了 Android SDK 开发文件和 API 文档等。
- extras：该文件夹存放了 Google 提供的 USB 驱动、Intel 提供的硬件加速等附加工具包。
- platform-tools：该文件夹下存放了 Android 平台的相关工具。
- samples：该文件夹下存放了 Android 平台的示例程序。
- sources：该文件夹下存放了 Android SDK4.2 的源代码。

(8) 在命令窗口中可以使用 Android SDK 的各种工具，建议将 Android SDK 目录下的 tools 子目录、platform-tools 子目录添加到系统的 PATH 环境变量中。

1.2.2 安装运行、调试环境

Android 程序必须在 Android 手机上运行，因此 Android 开发时不需准备相关运行、调试环境。准备 Android 程序的运行、调试环境有以下两种方式。

- 条件允许，优先考虑购买 Android 真机(真机调试的速度更快，效果更好)。
- 配置 Android 虚拟设备(即 AVD)。

1. 使用真机作为运行、调试环境

使用真机作为运行、调试环境时，只需要完成以下 3 步。

(1) 用 USB 连接线将 Android 手机连接到电脑上。

(2) 在电脑上为手机安装驱动，不同手机厂商的 Android 手机的驱动略有差异，请登录该手机厂商官网下载手机驱动。

需要注意的是，电脑仅能识别 Android 手机的存储卡是不够的，安装驱动才能把 Android 手机整合成运行、调试环境。

(3) 打开手机的调试模式。打开手机，依次点击“所有应用—设置—开发者选项”，进入如图 1.4 所示的设置界面。勾选“不锁定屏幕”、“USB 调试”、“允许模拟位置”3 个选项即可。如果开发者还有其他需要，也可以勾选其他的开发者选项。

图 1.4

2. 使用 AVD 作为运行、调试环境

AndroidSDK 为开发者提供了可以在电脑上运行的“虚拟手机”，Android 把它称为 AndroidVirtualDevice(AVD)。如果开发者没有 Android 手机，则可以在 AVD 上运行编写的 Android 应用。

创建、删除和浏览 AVD 之前，通常应该先为 AndroidSDK 设置一个环境变量：ANDROID_SDK_HOME，该环境变量的值为磁盘上一个已有的路径。如果不设置该环境变量，开发者创建的虚拟设备默认保存在 C:\Documentsand Settings\<user_name>\.androidd 目录下(以 WindowsXP 为例)；如果设置了 ANDROID_SDK_HOME 环境变量，那么虚拟设备就会保存在%ANDROID_SDK_HOME%/.Android 路径下。

在图形界面下管理 AVD 比较简单，可以借助 AndroidSDK 和 AVD 管理器，在图形用户界面下完成操作，比较适合新上手的用户。

(1) 通过 AndroidSDK 安装目录下的 AVDManager.exe 启动 AVD 管理器，系统启动如图 1.5 所示的 AVD 管理器。单击该管理器左边的“Android Virtual Devices”项，管理器列出当前已有的 AVD 设备，如图 1.5 所示。

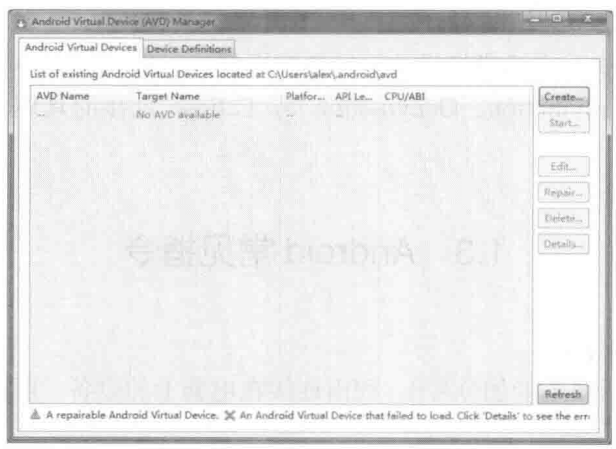

图 1.5

(2) 单击图 1.5 所示窗口右边的“Create...”按钮，AVD 管理器弹出如图 1.6 所示对话框。

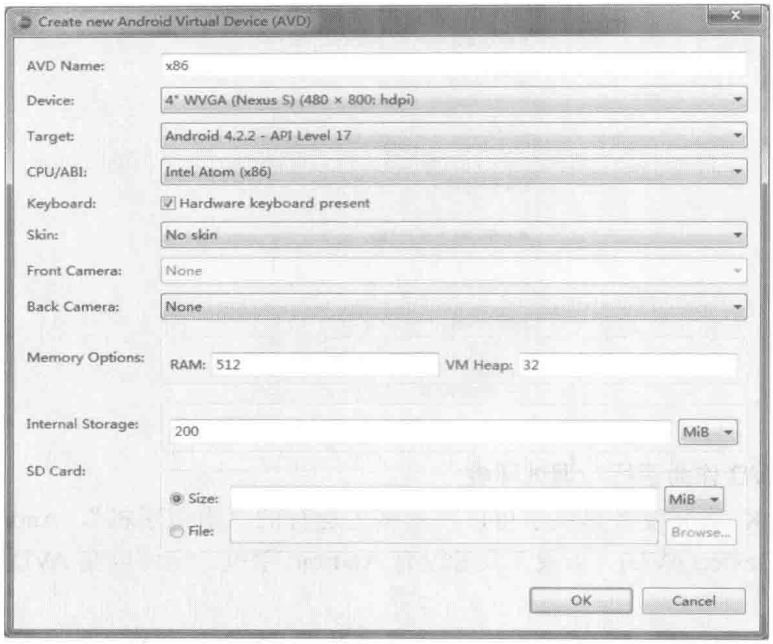

图 1.6

(3) 在图 1.6 所示的对话框中填写 AVD 设备的名称、Android 平台的版本和虚拟 SD 卡的大小，然后单击该对话框下面的“OK”按钮，管理器即将开始创建 AVD 设备，开发者只要稍作等待即可。由于运行速度较慢，因此笔者选择 HVGA(即 480×320 的分辨率)

创建完成后，返回图 1.5 所示的窗口，该管理器将会列出当前所有可用的 AVD 设备。如果开发者想删除某个 AVD 设备，只要在图 1.5 所示窗口中指定 AVD 设备，然后单击右边的“Delete...”按钮即可。

AVD 设备创建成功后，就可以使用模拟器来运行 AVD 了。在 Android SDK 和 AVD 管理器中运行 AVD 非常简单：在图 1.5 所示窗口中选择需要运行的 AVD 设备，单击图 1.5 所示窗口中的“Start...”按钮即可。

在实际开发过程中，上述模拟机的运行速度很慢，一般情况下不会使用，通常是通过真机调试或者使用 Genymotion。Genymotion 与 Eclipse 连接的具体步骤，读者可以自行学习。

1.3 Android 常见指令

Android 常用指令有：

(1) adb devices(后面不能加分号;)：列出连接在电脑上的设备，可以是模拟器或真实手机。例如：

```
C:\Users\hacket>adb devices
List of devices attached
emulator-5556   device
emulator-5554   device
```

(2) adb install helloworld.apk(一个设备)：安装一个 apk，例如：

```
C:\Users\hacket\Desktop>adb install helloworld.apk
error: more than one device and emulator
- waiting for device -
```

如果有多个设备，会报错误，此时用-s 设备名指定设备，例如：adb -s emulator-5554 install 1.apk (多个设备)

```
C:\Users\hacket\Desktop>adb -s emulator-5554 install 1.apk
81 KB/s (225886 bytes in 2.707s)
        pkg: /data/local/tmp/1.apk
Success
```

(3) adb uninstall (包名) (一个设备)：卸载 apk。如果有多个设备，用-s 设备名指定设备，例如：adb -s emulator-5554 uninstall cn.zengfansheng.helloworld(多个设备)

```
C:\Users\hacket\Desktop>adb -s emulator-5554 uninstall cn.zengfansheng.helloworl
d
Success
```

(4) ddms 中 Reset adb 就是用下面两句命令实现的。

重启 adb 的服务：

adb kill-server——把 adb 调试桥的服务杀死 (注意：kill 和-server 没有空格)。

adb start-server——把 adb 调试桥的服务重新开启(注意：kill 和-server 没有空格)。

netstat –ano——查看网络连接状态

(5) adb pull：从手机里面提取一个文件。也可提取多个文件，例如：

adb -s emulator-5554 pull /mnt/sdcard/1.apk (多个模拟器和真机)

(6) adb push ：把电脑上的文件放在手机里面，例如：

adb -s emulator-5554 push Helloworld.apk /sdcard/1.apk (多个模拟器和真机)

```
C:\Users\hacket\Desktop>adb -s emulator-5554 push Helloworld.apk /sdcard/1.apk
82 KB/s (225886 bytes in 2.659s)
```

1.4　Android 的日志工具 Log

Android 中的日志工具类是 Log(android.util.Log)，该类中提供了以下几个方法供打印日志。

1. Log.v()

Log.v()方法用于打印那些最为琐碎的、意义最小的日志信息。对应级别 verbose，是 Android 日志里面级别最低的一种。

2. Log.d()

Log.d()方法用于打印一些调试信息，这些信息对调试程序和分析问题是有帮助的，对应级别 debug，比 verbose 高一级。

Log.d 方法中传入了两个参数，第一个参数是 tag，一般传入当前的类名，主要用于过滤打印信息。第二个参数是 msg，即要打印的具体内容。

3. Log.i()

Log.i()方法用于打印一些比较重要的数据，这些数据可以分析用户行为。对应级别 info，比 debug 高一级。

4. Log.w()

Log.w()方法用于打印一些警告信息，提示程序在这个地方可能会有潜在的风险，应尽快修复。对应级别 warn，比 info 高一级。

5. Log.e()

Log.e()方法用于打印程序中的错误信息，比如程序进入到了 catch 语句当中。当有错误信息打印出来的时候，一般代表程序出现了严重问题，必须尽快修复。对应级别 error，比 warn 高一级。

第2章 布 局

2.1 线性布局

线性布局分为 vertical 垂直线性布局和 horizontal 水平线性布局，开发者可以根据自己的需要选择。

LinearLayout 是线性布局控件，它包含的子控件将以横向或竖向的方式按照相对位置来排列所有的 widgets 或者其他的 containers，超过边界时，某些控件将缺失或消失。因此一个垂直列表的每一行只会有一个 widget 或者是 container，而不管他们有多宽；而一个水平列表将会只有一个行高(高度为最高子控件的高度加上边框高度)。LinearLayout 保持其所包含的 widgets 或者是 containers 之间的间隔以及对齐方式(相对一个控件的右对齐、中间对齐或者左对齐)。

LinearLayout 属性如下：

android:orientation：定义布局的方向——水平或垂直(horizontal/vertical)。

android:layout_weight：子元素对未占用空间【水平或垂直】分配权重值，其值越小，权重越大。

android:layout_width：宽度(fill_parent: 父元素决定，wrap_content: 本身的内容决定)。

android:layout_height：高度(直接指定一个 px 值)。

android:gravity：内容的排列形式(常用 top, bottom, left, right, center)。

下面根据一个实例来了解线性布局。

新建项目 MyLayout，在 activity_main 中添加 4 个按钮，代码如下：

activity_main.xml 文件：

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <!-- 按钮 1 -->
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button1"
        android:text="button1" />
```