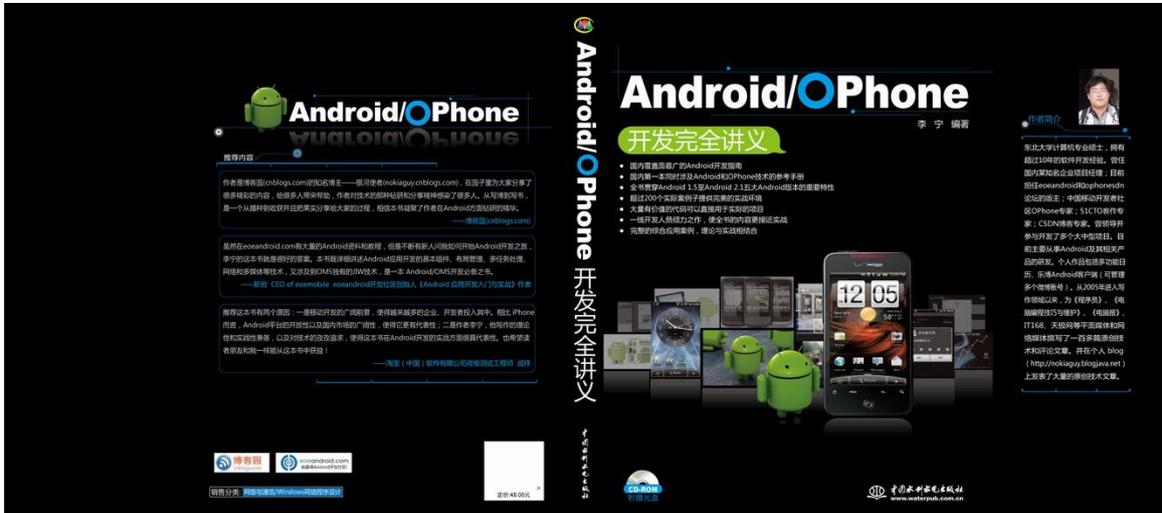


《Android/OPhone 开发完全讲义》样章

李宁 编著

<http://nokiaguy.blogjava.net>



内容简介

本书近 500 页，共 25 章，分为 5 篇，超过 200 个完整的例子、超过 2 万行代码。

第 1 篇：介绍了 Android 的相关内容，其中包括 Android 的基本概念、搭建 Android 开发环境、Android SDK 中常用命令行工具的使用方法、在 PC 上安装 Android，并测试程序、Android 的学习资源等，还会给出一个简单的 Android 程序，使读者可以了解开发 Android 程序的基本过程。

第 2 篇：主要讲解了 Android 的用户接口，包括 View、定制组件、对话框、Toast 和 Notification、各种菜单、布局；Android 中中的组件（android.widget 包中的组件）；移动存储解决方案，包括 SharedPreferences、PreferenceActivity、文件存储、XML 存储、Sqlite 数据库及其应用、ContentProvider 等；应用程序之间的通讯，包括跨进程调用 Activity、接收和发送广播；Android 服务详细解，包括 Service 的生命周期，系统服务、时间服务和 AIDL 服务；网络，包括从网络上获得数据装载 Gallery、ListView 等组件。以及从 google 上获得图像，WebView 组件、访问 Http 资源以及 Webservice 等；多媒体技术，包括图形的基本操作、旋转图像、透明度、扭曲和拉伸图像、路径，音频和视频等技术。

第 3 篇：2D 和 OpenGL ES 动画，国际化，Android 中的 14 种资源，访问 Android 手机的硬件，包括通过 USB 驱动在手机上直接调试程序，录音，控制手机的摄像头、传感器（电子罗盘，计步器），GPS 和地图定位，wifi 等。App Widget、在桌面上添加快捷方式，实时文件夹（LiveFolder），NDK，脚本语言（Python、perl 等），手势输入、TTS、蓝牙等。

第 4 篇：介绍了 OPhone 的基础知识，以及 OPhone 与 Android 的差异，OPhone 的 API 扩展，并详细介绍了 JIL Widget 在多媒体，获得系统信息，控制硬件等方面的内容。

第 5 篇：给出了两个综合的例子：万年历，知道当前位置的 GTalk 机器人。

1

Android 入门

Google 于 2005 年并购了成立仅 22 个月的高科技企业 Android，展开了短信、手机检索、定位等业务，同时基于 Linux 的 Android 平台也进入了开发阶段。Google 在 2007 年 11 月 5 日发布了 Android 的第一个版本。在刚发布之初，Android 并没有引起业界太多的关注。但随着 Google 组建的开放手机联盟不断有新生力量加入，Android 这个初出茅庐的小子已成为与 iPhone 分庭抗礼的生力军。

在作者写作本书时，至少有数十家不同规模的手机厂商宣布加入 Android 阵营。基于 Android 的手机也是琳琅满目。现在让我们进入时空隧道，回到 2008 年 9 月 23 日（北京时间 2008 年 9 月 23 日 22:30）的美国纽约，Google 和运营商 T-Mobile 共同发布了世界上第一款安装 Android 系统的手机 T-Mobile G1。由于这款手机的出色表现，使 Android 真正成为了万众瞩目的焦点。正是因为 Android 和其他几项创新，在 17 个月后的 2010 年 2 月 25 日，美国著名商业杂志《Fast Company》评选的 2010 年全球最具创新力公司 50 强中，Google 位列移动领域十大最具创新力公司榜首。



本章内容

- Android 的系统构架
- 搭建 Android 开发环境
- Android SDK 中的常用命令行工具（包括 adb、android 和 mkcdsard）
- 可以在 PC 上运行的 Android 系统（用于在没有真机的情况下测试程序）
- 应用程序商店

1.1 Android 的基本概念

Android 的中文意思是“机器人”。但在移动领域，大家一定会将 Android 与 Google 联系起来。Android 本身就是一个操作系统，只是这个操作系统是基于 Linux 内核的。也就是说，从理论上，基于 Linux 的软件移植到 Android 上是最容易的。Android 是一个由 30 多家科技公司和手机公司组成的“开放手机联盟”共同研发的，而且完全免费开源，这将大大降低新型手机设备的研发成本，甚至 Android 已成了“山寨”机的首选。

1.1.1 Android 简介

Android 作为 Google 最具创新的产品之一，正受到越来越多的手机厂商、软件厂商、运营商及个人开发者的追捧。目前 Android 阵营主要包括 HTC（宏达电）、T-Mobile、高通、三星、LG、摩托罗拉、ARM、软银移动、中国移动、华为等。虽然这些机构有着不同的性质，但它们都在 Android 平台的基础上不断创新，让用户体验到最优质的服务。尤其要提一下的是中国移动与播思公司联合研发的基于 Android 的 OMS 系统已取得了不俗的业绩(关于 OMS 和 OPhone 将在第 4 篇详细介绍)。下面欣赏几款具有代表性的 Android 手机。第一款毫无疑问，就是世界是第一部 Android 手机 T-Mobile G1，如图 1.1 所示。这款手机带有一个物理键盘（硬键盘），可以通过侧划拉出。第二款是创下了销售奇迹的 HTC Hero，也称为 G3，如图 1.2 所示。这款手机的显著特征是下方有一个突起的小“下巴”。除此之外，HTC Hero 绚丽的 Sense 界面也成为 Android 手机中一道亮丽的风景。最后一款创下了配置之最，这就是 2009 年底发布的几款拥有 1GHz CPU 的手机之一：SonyEricsson X10，如图 1.3 所示。

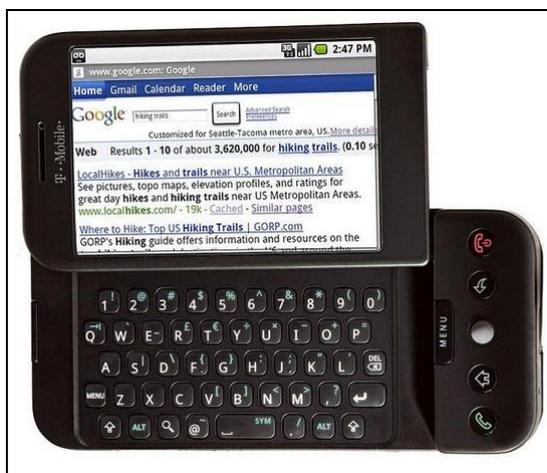


图 1.1 T-Mobile G1



图 1.2 HTC Hero



图 1.3 拥有 1GHz CPU 的 SonyEricsson X10

欣赏完这么多“超酷”的手机，现在来看一下 Android 到底有什么魔力，可以让众多的粉丝为之疯狂。据粗略统计，Android 至少有如下 8 件制胜法宝：

- 开放性。Android 平台是免费、开源的。而且 Google 通过与运营商、设备制造商、开发商等机构形成的战略联盟，希望通过共同制定标准使 Android 成为一个开放式的生态系统。
- 应用程序的权限由开发人员决定。编写过 Symbian、Java ME 程序的读者应该能体会到这些程序在发布时有多麻烦。如果访问到某些限制级的 API，不是出现各种各样的提示，就是根本无法运行。要想取消这些限制，就得向第三方的认证机构购买签名，而且价格不菲。而 Android 平台的

应用程序就幸福得多。要使用限制级的 API，只需要在自己的应用程序中配置一下即可，完全是 DIY。这也在某种程度上降低了 Android 程序的开发成本。

- 我的平台我作主。Android 上的所有应用程序都是可替换和扩展的，即使是拨号、Home 这样的核心组件也是一样。只要有足够的想象力，就可以缔造出一个独一无二、完全属于自己的 Android 世界。
- 应用程序之间的无障碍沟通。应用程序之间的通信一直令人头痛，而在 Android 平台上无疑是一种享受。在 Android 平台上应用程序之间至少有 4 种沟通方式。很难说哪一种方式更好，但它们的确托起了整个 Android 的应用程序框架。
- 拥抱 Web 的时代。如果想在 Android 应用程序中嵌入 HTML、JavaScript，那真是再容易不过了。基于 Webkit 内核的 WebView 组件会完成一切。更值得一提的是，JavaScript 还可以和 Java 无缝地整合在一起（见第 9 章的实例 57 中的介绍）。
- 物理键盘和虚拟键盘双管齐下。从 Android 1.5 开始，Android 同时支持物理键盘和虚拟键盘，从而可大大丰富用户的输入选择。尤其是虚拟键盘，已成为 Android 手机中主要的输入方式。
- 个性的充分体现。21 世纪是崇尚个性的时代。Android 也紧随时代潮流提供了众多体现个性的功能。例如，Widget、Shortcut、Live WallPapers，无一不尽显手机的华丽与时尚。
- 舒适的开发环境。Android 的主流开发环境是 Eclipse + ADT + Android SDK。它们可以非常容易地集成到一起，而且在开发环境中运行程序要比 Symbian 这样的传统手机操作系统更快，调试更方便。

虽然 Android 的特点还有很多，但这已经不重要。重要的是现在 Android 已经成为万众瞩目的国际巨星，她的未来将令人充满期望。

1.1.2 Android 的系统构架

通过上一节的介绍，我们对 Android 的特点已经有了一个初步的了解。本节将介绍 Android 的系统构架。先来看看 Android 的体系结构，如图 1.4 所示。

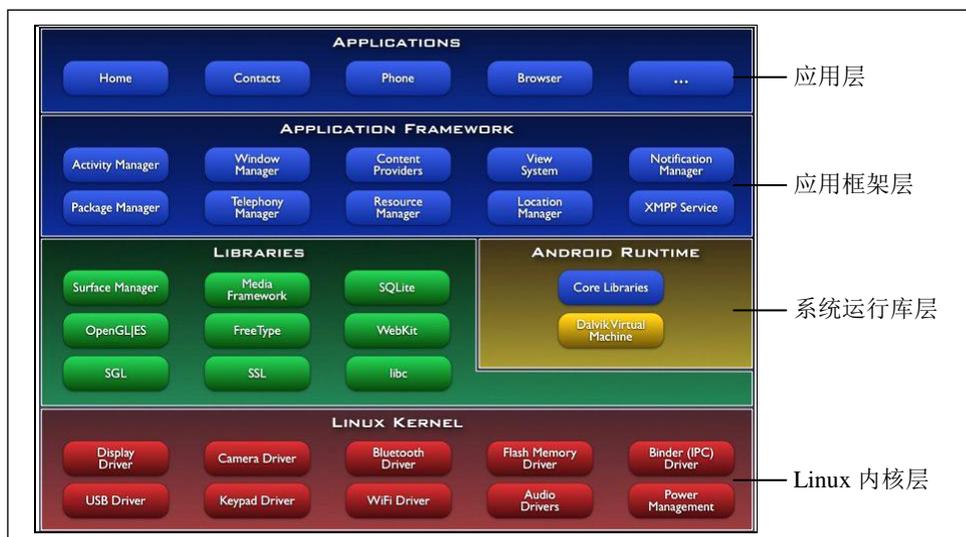


图 1.4 Android 的体系结构

从图 1.4 可以看出 Android 分为 4 层，从高到低分别是应用层、应用框架层、系统运行库层和 Linux 内核层。下面将对这 4 层进行简单介绍。

- 应用层。该层由运行在 Dalvik 虚拟机（为 Android 专门设计的基于寄存器的 Java 虚拟机，运行 Java 程序的速度更快）上的应用程序（主要由 Java 语言编写）组成。例如，日历、地图、浏览器、联系人管理，都属于应用层上的程序。

- 应用框架层。该层主要由 View、通知管理器(Notification Manager)、活动管理器(Activity Manager) 等由开发人员直接调用的组件组成。
- 系统运行库层。Java 本身是不能直接访问硬件的。要想让 Java 访问硬件, 必须使用 NDK 才可以。NDK 是一些由 C/C++ 语言编写的库。这些程序也是该层的主要组成部分。该层主要包括 C 语言标准库、多媒体库、OpenGL ES、SQLite、Webkit、Dalvik 虚拟机等。也就是说, 该层是对应用框架层提供支持的层。
- Linux 内核层。该层主要包括驱动、内存管理、进程管理、网络协议栈等组件。目前 Android 的版本基于 Linux 2.6 内核。

1.2 Android 开发环境的搭建

工欲善其事, 必先利其器。开发 Android 应用程序总不能直接用记事本开发吧(那些超级大牛除外)。找到合适的开发工具是学习 Android 开发的第一步。而更多地了解 Android 的开发环境将会对进一步学习 Android 保驾护航。

1.2.1 开发 Android 程序需要些什么

开发 Android 程序至少需要如下工具和开发包:

- JDK (建议安装 JDK1.6 或以上版本)
- Eclipse
- Android SDK
- ADT (Android Development Tools, 开发 Android 程序的 Eclipse 插件)

其中 JDK 的安装非常简单, 读者可以在官方网站下载 JDK 的最新版, 并按着提示进行安装。Eclipse 下载后直接解压即可运行。在 1.2.2 节和 1.2.3 节将介绍 Android SDK 和 ADT 的安装。

1.2.2 安装 Android SDK

读者可以从下面的两个地址下载 Android SDK 最新版本:

地址 1

<http://developer.android.com/intl/zh-CN/sdk/index.html>

地址 2

<http://androidappdocs.appspot.com/sdk/index.html>

该版本可以同时安装 6 个 Android SDK 版本(1.1 至 2.1)。要注意, Android SDK 是在线安装。在安装 Android SDK 之前, 要保证有稳定而快速的 Internet 连接。如果完全安装 Android SDK, 安装时间会比较长, 读者需要耐心等待。

如果安装 Android SDK 时下载文件失败, 可以将如图 1.5 所示的安装界面右下角的第 1 个复选框选中, 然后重新安装 Android SDK。如果安装过程顺利, 将会出现如图 1.6 所示的下载界面。

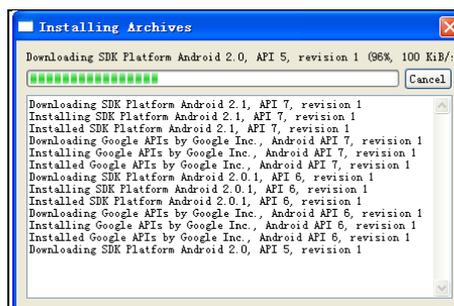
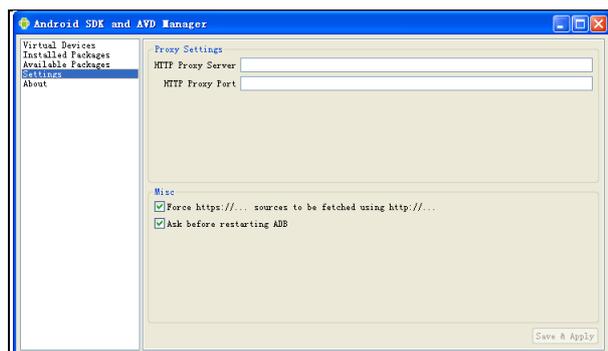


图 1.5 使用 http 下载文件

图 1.6 安装过程的下载界面

Android SDK 安装成功后，会看到如图 1.7 所示的 Android SDK 根目录结构。platforms 目录包含当前 Android SDK 支持的所有版本，如图 1.8 所示。

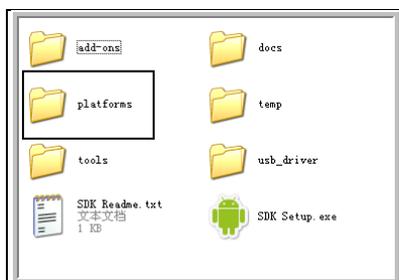


图 1.7 Android SDK 根目录结构



图 1.8 已经安装的所有 Android SDK 版本

1.2.3 安装 Eclipse 插件 ADT

在作者写作本书时，ADT 的最新版本是 0.9.5。该版本必须下载最新的 Android SDK 才能使用。读者可以在 Eclipse 中直接安装 ADT。

如果读者使用的是 Eclipse 3.4 (Ganymede)，单击【Help】>【Software Updates...】菜单项。在显示的对话框中单击【Available Software】标签页，然后单击【Add Site...】按钮。在显示的对话框的文本框中输入如下地址：

<https://dl-ssl.google.com/android/eclipse/>

单击【OK】按钮关闭该对话框。回到【Available Software】标签页，选中刚才增加的地址，然后单击右侧的【Install】按钮开始安装 ADT 插件。在弹出的安装对话框中选中 Android DDMS 和 Android Development Tools 两项，单击【Next】按钮进入下一个安装界面，选中接受协议复选框，最后单击【Finish】按钮开始安装。当成功安装 ADT 后，重启 Eclipse 即可使用 ADT 来开发 Android 程序。

如果使用的是 Eclipse 3.5 (Galileo)，单击【Help】>【Install New Software...】菜单项，显示安装对话框，然后单击右侧的【Add...】按钮，在弹出的对话框的第 1 个文本框输入一个名字，在第 2 个文本框输入上面的地址。剩下的安装过程与 Eclipse 3.4 类似。读者可以参考 Eclipse 3.4 的安装过程或通过如下两个地址查看官方的安装文档。

地址 1

<http://developer.android.com/intl/zh-CN/sdk/eclipse-adt.html>

地址 2

<http://androidappdocs.appspot.com/sdk/eclipse-adt.html>

安装完 ADT 后，还需要设置一下 Android SDK 的安装目录。单击【Window】>【Preferences】菜单项。在弹出的对话框中选中左侧的【Android】节点。在右侧的【SDK Location】文本框中输入 Android SDK 的安装目录，如图 1.9 所示。

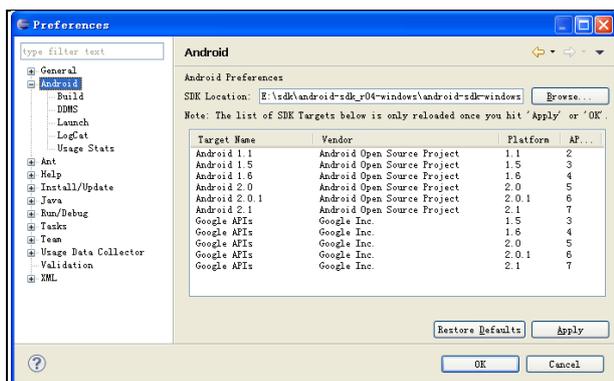


图 1.9 设置 Android SDK 的安装目录

1.2.4 测试 ADT 是否安装成功

本节将新建一个 Android 工程来测试一下 ADT 是否安装成功。单击【New】>【Android Project】菜单项（如果没有该菜单项，可以单击【New】>【Other】菜单项，在弹出的对话框的树中寻找【Android】节点），显示建立 Android 工程的对话框。在弹出的对话框中按照图 1.10 所示输入相应的内容（黑框内的内容必须输入或选择）。单击【Finish】按钮创建 Android 工程。

在运行 firstandroid 工程之前还需要建立一个 AVD 设备。一个 AVD 设备对应一个 Android 版本的模拟器实例。由于 firstandroid 使用的是 Android 2.1，因此，需要建立一个支持 Android 2.1 的 AVD 设备。单击 Eclipse 左侧的  按钮。在显示的对话框中单击【New...】按钮新建一个 AVD 设备。并按照图 1.11 所示黑框中的内容输入相应的值，然后单击【Create AVD】按钮建立 AVD 设备。

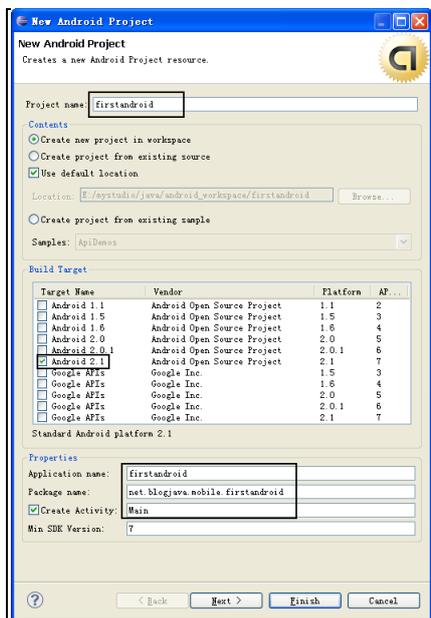


图 1.10 创建 Android 工程

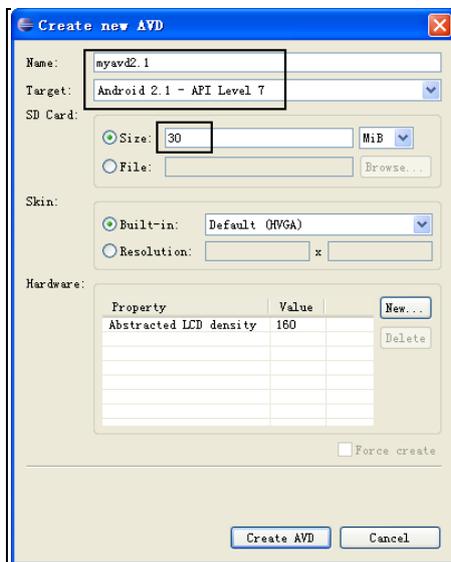


图 1.11 建立 AVD 设备

建立完 AVD 设备后，找到 firstandroid 工程，单击右键菜单的【Run As】>【Android Application】菜单项，运行 firstandroid。这时 ADT 会自动启动模拟器，并在模拟器上运行 firstandroid。在模拟器成功启动后，会出现如图 1.12 所示的模拟器锁定状态的界面。用鼠标按住屏幕左下方的小锁，将其拖动到屏幕右下方的喇叭处，就会解除这种状态。这时会显示 firstandroid 的运行结果，如图 1.13 所示。



图 1.12 模拟器的锁定状态

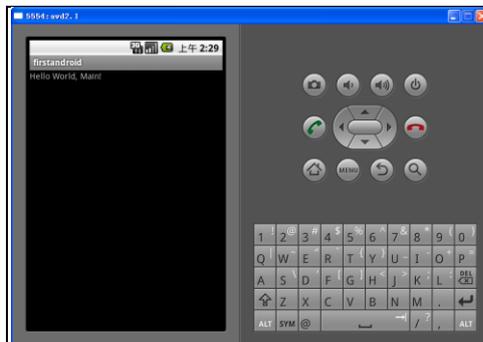


图 1.13 firstandroid 的运行结果



虽然 Android 模拟器可以测试大多数 Android 应用程序，但有一些和硬件相关的 API（例如，蓝牙、传感器）无法测试。在这种情况下就需要通过 USB 数据线连接真机进行测试，详细的测试方法请见 14.1 节的内容。

1.3 Android SDK 中的常用命令行工具

在<Android SDK 安装目录>\tools 目录中带了很多命令行工具。虽然一般的开发人员并不需要完全掌握这些工具的使用方法，但了解这些工具的一些基本使用方法还是会对以后的开发工作起到一定的辅助作用。本节将介绍几种常用的命令行工具的使用方法，这些工具主要包括 adb、android 和 mksdcard。在使用这些命令行工具之前，建议读者将<Android SDK 安装目录>\tools 目录加到 PATH 环境变量中，这样在任何目录中都可以使用这些工具了。

1.3.1 启动和关闭 ADB 服务 (adb start-server 和 adb kill-server)

经作者测试，模拟器在运行一段时间后，adb 服务有可能（在 Windows 进程中可以找到这个服务，该服务用来为模拟器或通过 USB 数据线连接的真机服务）会出现异常。这时需要重新对 adb 服务关闭和重启。当然，重启 Eclipse 可能会解决问题，但那比较麻烦。如果想手工关闭 adb 服务，可以使用如下命令：

```
adb kill-server
```

在关闭 adb 服务后，要使用如下命令启动 adb 服务：

```
adb start-server
```

1.3.2 查询当前模拟器/设备的实例 (adb devices)

有时需要启动多个模拟器实例，或启动模拟器的同时通过 USB 数据线连接了真机。在这种情况下就需要使用如下命令查询当前有多少模拟器或真机在线：

```
adb devices
```

执行上面的命令后，会输出如图 1.14 所示的信息。

其中第 1 列的信息 (emulator-5554、HT9BYL904399) 表示模拟器或真机的标识。emulator-5554 表示模拟器，其中 5554 表示 adb 服务为该模拟器实例服务的端口号。每启动一个新的模拟器实例，该端口号都不同。HT9BYL904399 表示通过 USB 数据线连接的真机。如果在运行 Android 程序时有多个模拟器或真机在线，会出现一个选择对话框。如果选择在真机运行，ADT 会直接将程序安装在手机上。详细介绍读者可以查看 14.1.2 节的内容。

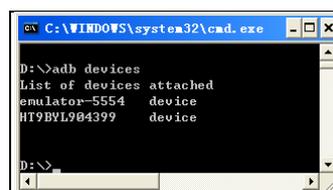


图 1.14 查询模拟器/设备的实例

输出信息的第 2 列都是 device，表示当前设备都在线。如果该列的值是 offline，表示该实例没有连接到 adb 上或实例没有响应。

1.3.3 安装、卸载和运行程序 (adb install、adb uninstall 和 am)

在 Eclipse 中运行 Android 程序必须得有 Android 源码工程。如果只有 apk 文件（Android 应用程序的发行包，相当于 Windows 中的 exe 文件），该如何安装和运行呢？答案就是 adb 命令。假设要安装一个 ebook.apk 文件，可以使用如下命令：

```
adb install ebook.apk
```

假设 ebook.apk 中的 package 是 net.blogjava.mobile.ebook，可以使用如下命令卸载这个应用程序：

```
adb uninstall net.blogjava.mobile.ebook
```

关于 package 的概念在以后的学习中会逐渐体会到，现在只要知道 package 是 Android 应用程序的唯一标识即可。如果在安装程序之前，该程序已经在模拟器或真机上存在了，需要先使用上面的命令卸载这个

应用程序，然后再安装。或使用下面的命令重新安装。

```
adb install -r ebook.apk
```

在卸载应用程序时可以加上 -k 命令行参数保留数据和缓冲目录，只卸载应用程序。命令如下所示：

```
adb uninstall -k net.blogjava.mobile.ebook
```

如果机器上有多个模拟器或真机实例，需要使用 -s 命令行参数指定具体的模拟器或真机。例如，下面的命令分别在模拟器和真机上安装、重新安装和卸载应用程序。

在 emulator-5554 模拟器上安装 ebook.apk:

```
adb -s emulator-5554 install ebook.apk
```

在真机上安装 ebook.apk:

```
adb -s HT9BYL904399 install ebook.apk
```

在 emulator-5554 模拟器上重新安装 ebook.apk:

```
adb -s emulator-5554 install -r ebook.apk
```

在真机上重新安装 ebook.apk:

```
adb -s HT9BYL904399 install -r ebook.apk
```

在 emulator-5554 模拟器上卸载 ebook.apk（不保留数据和缓冲目录）:

```
adb -s emulator-5554 uninstall net.blogjava.mobile.ebook
```

在真机上卸载 ebook.apk（保留数据和缓冲目录）:

```
adb -s HT9BYL904399 uninstall -k net.blogjava.mobile.ebook
```

如果在模拟器和真机上成功安装 ebook.apk，将会分别输出如图 1.15 和图 1.16 所示的信息。

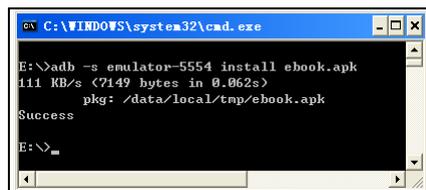


图 1.15 在模拟器上成功安装 ebook.apk

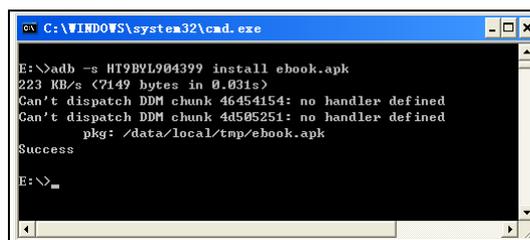


图 1.16 在真机上成功安装 ebook.apk

如果想在模拟器或真机上运行已安装的应用程序，除了直接在模拟器或真机上操作外，还可以使用如下命令直接运行程序。

在 emulator-5554 模拟器上运行 ebook.apk:

```
adb -s emulator-5554 shell am start -n net.blogjava.mobile.ebook/net.blogjava.mobile.ebook.Main
```

在真机上运行 ebook.apk:

```
adb -s HT9BYL904399 shell am start -n net.blogjava.mobile.ebook/net.blogjava.mobile.ebook.Main
```

其中 Main 是 ebook.apk 的主 Activity，相当于 Windows 应用程序的主窗体或 Web 应用程序的主页面。am 是 shell 命令。关于 shell 命令将在 1.3.5 节详细介绍。

1.3.4 PC 与模拟器或真机交换文件 (adb pull 和 adb push)

在开发阶段或其他原因，经常需要将 PC 上的文件复制到模拟器或真机上，或将模拟机和真机上的文件复制到 PC 上。使用 adb pull 和 adb push 命令可以很容易地完成这个工作。例如，下面的命令将真机的 SD 卡根目录下的 camera.jpg 文件复制到 PC 的当前目录，取名为 picture.jpg。又把 picture.jpg 文件复制到真机的 SD 卡的根目录，取名为 abc.jpg。

从真机上复制文件到 PC:

```
adb -s HT9BYL904399 pull /sdcard/camera.jpg picture.jpg
```

从 PC 复制文件到真机:

```
adb -s HT9BYL904399 push picture.jpg /sdcard/abc.jpg
```

如果读者安装了 ADT，可以通过 DDMS 透视图的【File Explorer】视图右上方的几个按钮方便地从模拟器或真机上导入、导出和删除文件。

1.3.5 Shell 命令

Android 是基于 Linux 内核的操作系统，因此，在 Android 上可以执行 Shell 命令。虽然在手机上提供了可以输入命令的 Shell 程序，但在手机上输入程序实在不方便。为了方便地在模拟器或手机上执行 Shell 命令，可以使用如下命令在 PC 上进入 Shell 控制台：

```
adb -s HT9BYL904399 shell
```

Shell 控制台的提示符是一个井号 (#)。进入 Shell 后，输入 `cd system/bin` 命令，再输入 `ls` 命令，可以看到当前 Android 系统支持的命令文件，如图 1.17 所示。读者可以根据实际情况使用相应的命令。

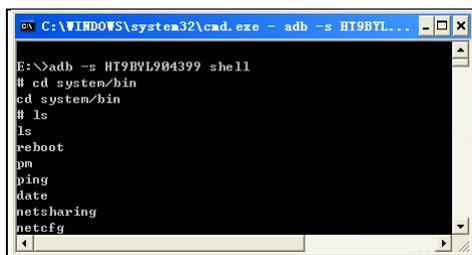


图 1.17 Shell 控制台

1.3.6 创建、删除和浏览 AVD 设备 (android)

在 1.2.4 节介绍了如何在 Eclipse 中建立一个 AVD 设备。本节将介绍直接使用 android 命令建立和删除 AVD 设备。建立 AVD 设备的命令如下：

```
android create avd -n myandroid1.5 -t 2
```

其中 `myandroid1.5` 表示 AVD 设备的名称，该名称可以任意设置，但不能和其他 AVD 设备冲突。`-t 2` 中的 2 指建立 Android 1.5 的 AVD 设备，1 表示 Android 1.1 的 AVD 设备，以此类推。目前最新的 Android 2.1 应使用 `-t 6` 来建立 AVD 设备。在执行完上面的命令后，会输出如下信息来询问是否继续定制 AVD 设备：

```
Android 1.5 is a basic Android platform.
```

```
Do you wish to create a custom hardware profile [no]
```

如果读者不想继续定制 AVD 设备，直接按回车键即可。如果想定制 AVD 设备，输入 `y`，然后按回车键。系统会按步提示该如何设置。中括号内是默认值，如果某个设置项需要保留默认值，直接按回车键即可。如果读者使用的是 Windows XP，默认情况下 AVD 设备文件放在如下目录中：

```
C:\Documents and Settings\Administrator\.android\avd
```

如果想改变 AVD 设备文件的默认存储路径，可以使用 `-p` 命令行参数，命令如下：

```
android create avd -n myandroid1.5 -t 2 -p d:\my\avd
```

删除 AVD 设备可以使用如下命令：

```
android delete avd -n myandroid1.5
```

通过下面的命令可以列出所有的 AVD 设备：

```
android list avds
```

1.3.7 创建 SD 卡

在模拟器上测试程序经常需要使用 SD 卡。在 PC 上需要使用 `mksdcard` 命令创建一个虚拟的 SD 卡文件，创建一个 10MB 大小的 SD 卡文件的命令如下：

```
mksdcard -l sdcard 10MB sd.img
```

其中 `sdcard` 表示 SD 卡的卷标，10M 表示 SD 卡的大小，单位还可以是 KB。但要注意，SD 卡的大小不能小于 8MB，否则无法创建 SD 卡文件。`sd.img` 是 SD 卡的文件名。如果要在 Eclipse 中启动模拟器，或直接启动模拟器（使用 `emulator` 命令），需要使用 `-sdcard` 命令行参数指定 SD 卡文件的绝对路径。

1.4 PC 上的 Android

在程序发布之前，最好在真机上测试一下，毕竟模拟器无法 100%地模拟真机的环境。如果没有真机该怎么办呢？模拟器虽然可以正常运行大多数 Android 程序，但模拟器的环境毕竟是模拟出来的，并不是真正的 Android 操作系统。当然，还可以想其他的办法。Android 从理论上也可以运行在 PC 上。因此，可以采用像 Android LiveCD 一样的 PC 版 Android 操作系统来测试程序。这可是真正的 Android 操作系统，只是运行在 PC 上，而不是手机上。下面来感受一下 PC 上的 Android。

1.4.1 Android LiveCD

Android LiveCD 是 code.google.com 上的一个开源项目，以 ISO 形式发布，可直接从光盘启动。在作者写作本书时，Android LiveCD 的最新版是 0.3。读者可以从下面的地址下载 Android LiveCD 的最新版：

<http://code.google.com/p/live-android/downloads/list>

下载后，使用 VMWare 或其他虚拟机软件装载 ISO 文件，然后从光盘（ISO 文件）启动即可。启动后的界面如图 1.18 所示。运行程序后的效果如图 1.19 所示。



图 1.18 Android LiveCD 的运行界面



图 1.19 在 Android LiveCD 中运行程序的效果

向 Android LiveCD 上传 apk 程序可以采用多种方法。如果 PC 可以上网，可以采用 `wget` 命令从 Internet 下载 apk 程序。

```
wget http://ip 地址/blogger.apk
```

1.4.2 AndroidX86

AndroidX86 是另外一个可以在 PC 上安装的 Android 系统，不同的是该系统不仅可以从光盘启动，还可以直接安装在硬盘上，并从硬盘启动。因此，AndroidX86 是一个真正的操作系统。读者可以从如下地址下载 AndroidX86 的最新版：

<http://www.androidx86.org>

AndroidX86 的使用方法与 Android LiveCD 类似。图 1.20 是 AndroidX86 的安装界面。图 1.21 是 AndroidX86 的运行界面。

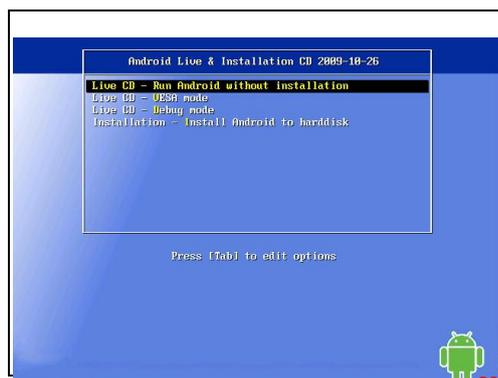


图 1.20 AndroidX86 的安装界面

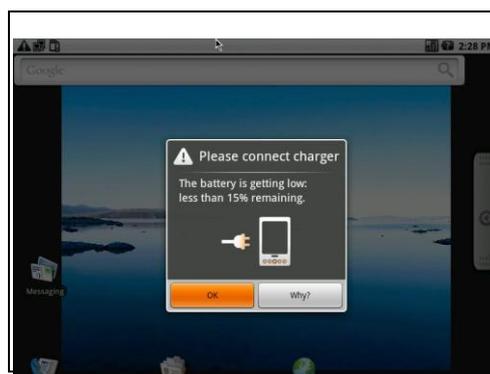


图 1.21 AndroidX86 的运行界面

1.5 Android 的学习资源

获得第一手的资源是学习 Android 的关键。通过如下两上地址可以访问 Android 的官方页面，在该页面有最新的开发指南、API、SDK 和其他资源。

地址 1

<http://developer.android.com>

地址 2

<http://android.appdocs.appspot.com>

除此之外，通过 Google 搜索也可以找到大量关于 Android 的学习资源，下面推荐几个国内比较受关注的 Android 学习网站。

- EOE Android 开发论坛，<http://www.eoandroid.com>。
- 安卓网，<http://www.hiapk.com>。
- 机锋网，<http://www.androidin.net>。
- 中国移动的开发者社区，<http://dev.chinamobile.com>。
- Google 的源代码托管网站，<http://code.google.com>。在该网站上有大量基于 Android 的应用程序源代码。直接通过源代码学习将会获得更佳的效果。

1.6 应用程序商店

写程序不是目的，写完程序我们能从中得到什么才是最终目的。当然，最直接得到的就是经验。可除此之外呢？相信大多数开发人员都希望从自己的程序中获利。当然，最好是名利双收。如果正在阅读本书的读者是这么想的，本节介绍的应用程序商店也许正好适合这些读者的口味。

1.6.1 Android Market

这是世界上第一个卖 Android 应用程序的在线商店。该在线商店由 Google 创办，地址如下：

<http://www.android.com/market>

在 Android 手机上可以通过 Android Market 客户端浏览和下载商店中的应用程序（在 Android Market 中有免费和收费两类程序）。客户端的主界面如图 1.22 所示。浏览和下载游戏程序的界面如图 1.23 所示。



图 1.22 Android Market 客户端的主界面 (HTC Hero)



图 1.23 浏览和下载游戏程序的界面 (HTC Hero)

1.6.2 Mobile Market (MM)

MM 是中国移动发起并创办的应用程序商店。该商店主要面向国内用户，更适合中国人的口味。主页的地址如下：

<http://www.mmarket.com>

MM 与 Android Market 的区别是 MM 不仅卖基于 Android/OVS 的应用程序 (OVS 是中国移动在 Android 的基础上定制的手机操作系统)，还可以卖基于 Symbian、Windows Mobile、Java ME 等手机平台的程序。也就是说，MM 是一个通用的应用程序商店。

MM 被集成在 OVS 系统中，所有的 OPhone 手机 (安装 OVS 的手机) 都包含一个 MM 客户端，主界面如图 1.24 所示。虽然目前在 MM 上还没有出现像在苹果的 App Store 上一样的创富神话，但随着 Android/OPhone 的手机用户不断增加。在 MM 上的应用程序会有一个很可观的收益 (当然，条件是自己的应用程序做得足够吸引人)。



图 1.24 MM 客户端主界面

1.6.3 其他应用程序商店

除了 Android Market 和 MM 外，国内还涌现出了很多 Android 应用程序商店，下面举几个比较著名的应用程序商店。

- EOE Market, <http://www.eoemarket.com>。
- 安卓 Market, <http://www.hiapk.com/bbs/devcenter.php>。

- Moto Market, <http://developer.motorola.com>。

虽然上面的 Market 目前都是免费上传和下载应用程序,也未实现什么赢利,但从长远来看,也会为推动 Android 起到一定作用。

1.7 本章小结

本章主要介绍 Android 开发环境的搭建。开发 Android 程序至少需要安装 JDK、Eclipse、Android SDK 和 ADT。在<Android SDK 安装目录>\tools 目录中有一些命令行工具,可以通过这些工具完全脱离 ADT 和 Eclipse 来完成开发工作。虽然作者并不建议这样做,但学习一些常用命令的使用方法会对开发工作起到一定的辅助作用。Android 不仅可以运行在手机上,从理论上说,Android 可以运行在所有的移动平台和基于 X86 框架的 PC 上。本章介绍了两种运行在 PC 上的 Android 系统: Android LiveCD 和 AndroidX86,通过这两种系统可以在没有真机的情况下在实际的 Android 系统中测试程序。做完程序后,需要将其发布到访问量较大的网站供用户免费或付费下载。而 Android Market 和 Mobile Market 是目前国内开发者的两个最好选择。当然,其他的 Market 也会为我们提供更多的选择。