

第 1 章 Android 程序分析环境搭建

在实际的 Android 软件开发过程中，可能很多开发人员有过这样的经历：

- 我有一个不错的 idea，正在开发一款类似想法的软件，可是涉及到的一些功能上的具体代码细节却难以下手，我看到别人的程序中有这个功能，它们是如何实现的呢？
- 我不小心安装了一个流氓软件，软件运行时会自动下载木马程序、恶意扣费、篡改手机系统，它是如何做到这些的呢？
- 我按照网上介绍的方法来分析 Android 程序，可是根本就无法正确地反编译程序，或是反编译出的代码语法混乱，根本无法阅读。

这些场景都提出了一个疑问，那就是如何分析一个 Android 应用程序？如何掌握这些软件的架构思想？分析别人的程序在很多人看来是不能够接受的行为，在他们眼中这种行为都应被视为盗窃。其实任何技术的起源本身就是从学习开始的，用正确的态度对待程序分析技术是可以的。

如果说，开发 Android 程序是一种学问，那么分析 Android 程序更像是一门艺术。在浩瀚如海的反汇编代码中分析出程序的执行流程与架构思想是一件很了不起的事情，这需要分析人员有着扎实的编程基础与深厚的思维分析能力。分析软件的过程犹如一次艰难的旅程，这条旅程会有多长？该怎么走？会有多少崎岖险路？没有人知道，但是先行者已经为我们铺下了台阶，我们只需沿着它慢慢前行。

1.1 Windows 分析环境搭建

搭建 Windows 分析平台的系统版本要求不高，Windows XP 或以上即可。本书的 Windows 平台的分析环境采用 Windows XP 32 位系统，如果读者使用 Windows 7 或其他版本，操作上是大同小异的。

1.1.1 安装 JDK

JDK 是 Android 开发必须的运行环境，在安装 JDK 之前，首先到 Oracle 公司官网上下载它。下载地址为：<http://www.oracle.com/technetwork/java/javase/downloads/index.html>，打开下载页面，目前最新版本为 Java SE 6 Update 33，如图 1-1 所示。

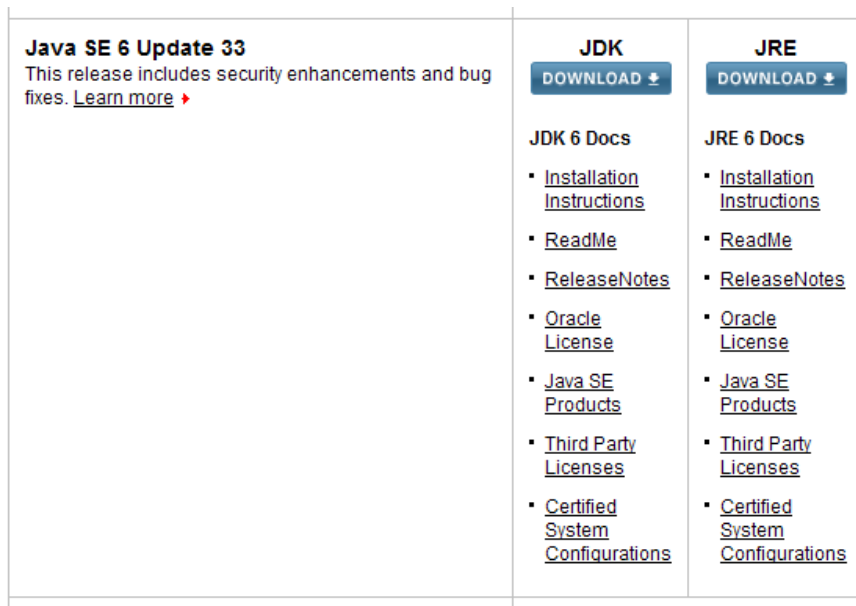


图1-1 下载JDK

点击 JDK 下面的 DOWNLOAD 按钮进入下载页面，勾选“Accept License Agreement”单选框，然后点击 jdk-6u33-windows-i586.exe 进行下载。下载完成后双击安装文件，启动 JDK 安装界面，如图 1-2 所示。



图1-2 JDK安装界面

与安装其他 Windows 软件一样，JDK 的安装过程也很简单，只需要不停点击下一步就可以顺利安装完成。安装完成后手动添加 JAVA_HOME 环境变量，值为“C:\Program Files\Java\jdk1.6.0_33”，并将“C:\Program Files\Java\jdk1.6.0_33\bin”添加到 PATH 变量中。如图 1-3 所示。



图1-3 设置Java环境变量

完成所有步骤后检查一下 Java 是否安装成功。单击“开始”按钮，选择“运行”，在出现的对话框中输入 CMD 命令打开 CMD 窗口，在 CMD 窗口中输入 `java -version`，如果屏幕上出现如图 1-4 所示的提示，说明安装成功。

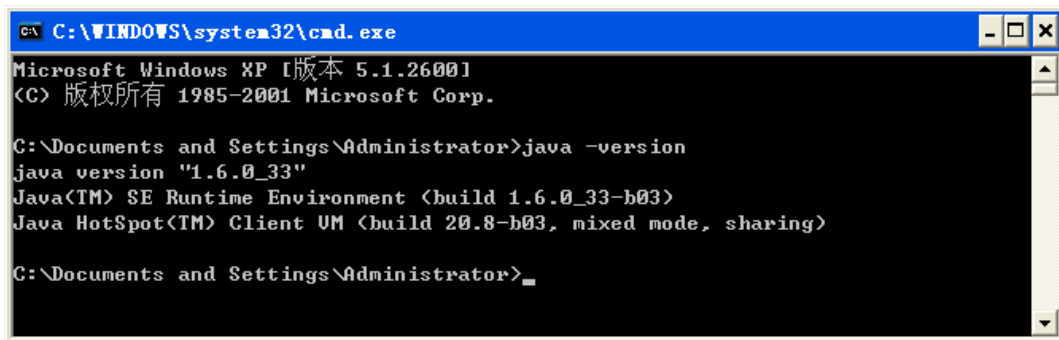
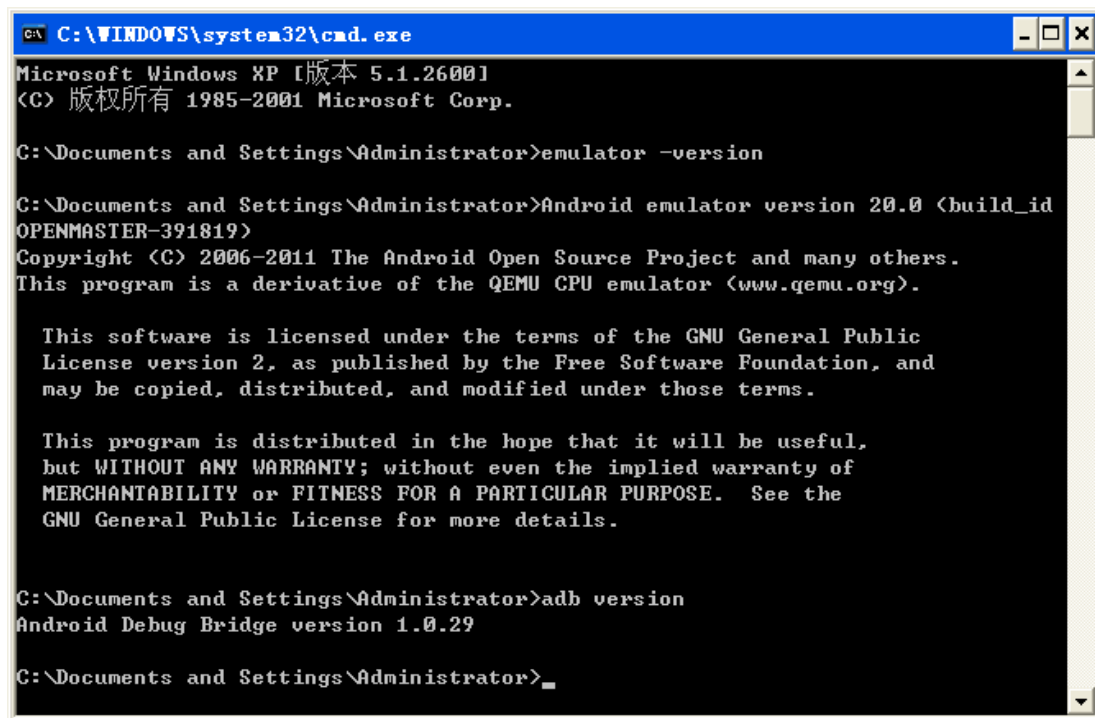


图1-4 查看Java是否正确安装

1.1.2 安装 Android SDK

Android SDK 是以 zip 压缩包的形式提供给开发人员的。首先到 Android 官网下载最新版本的 SDK，下载地址为：<http://developer.android.com/sdk/index.html>。SDK 提供了压缩包与安装文件两种方式供开发者下载，为了方便部署，本书采用下载安装文件的方式直接安装，目前 Android SDK 的最新版本为 r20，完整下载地址为：http://dl.google.com/androidinstaller_r20-windows.exe。

双击下载后的安装文件，将 Android SDK 安装到任意位置，本书安装环境为 `D:\android-sdk` 目录，然后将“`D:\android-sdk\tools`”与“`D:\android-sdk\platform-tools`”目录添加到系统的 PATH 环境变量中。添加完成后打开一个 CMD 窗口，输入“`emulator -version`”与“`adb version`”命令查看是否能成功运行。执行结果如图 1-5 所示。



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>emulator -version

C:\Documents and Settings\Administrator>Android emulator version 20.0 (build_id
OPENMASTER-391819)
Copyright (C) 2006-2011 The Android Open Source Project and many others.
This program is a derivative of the QEMU CPU emulator (www.qemu.org).

This software is licensed under the terms of the GNU General Public
License version 2, as published by the Free Software Foundation, and
may be copied, distributed, and modified under those terms.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

C:\Documents and Settings\Administrator>adb version
Android Debug Bridge version 1.0.29

C:\Documents and Settings\Administrator>
```

图1-5 检查Android SDK是否正确安装

Android SDK 安装成功后，需要通过 SDK 管理器下载具体版本的 SDK，双击“D:\android-sdk\SDK Manager.exe”文件，打开 Android SDK Manager，运行后如图 1-6 所示。

读者可以根据自己的需要选择相应的一个或多个版本进行下载，本书选择了 2.2、2.3.3、4.0、4.0.3、4.1 等几个版本，点击 Install package 按钮打开“Choose Package to Install”对话框，选择“Accept All”单选框，最后点击“Install”按钮开始下载，下载所需的时间根据网络环境差异可能会有所不同。

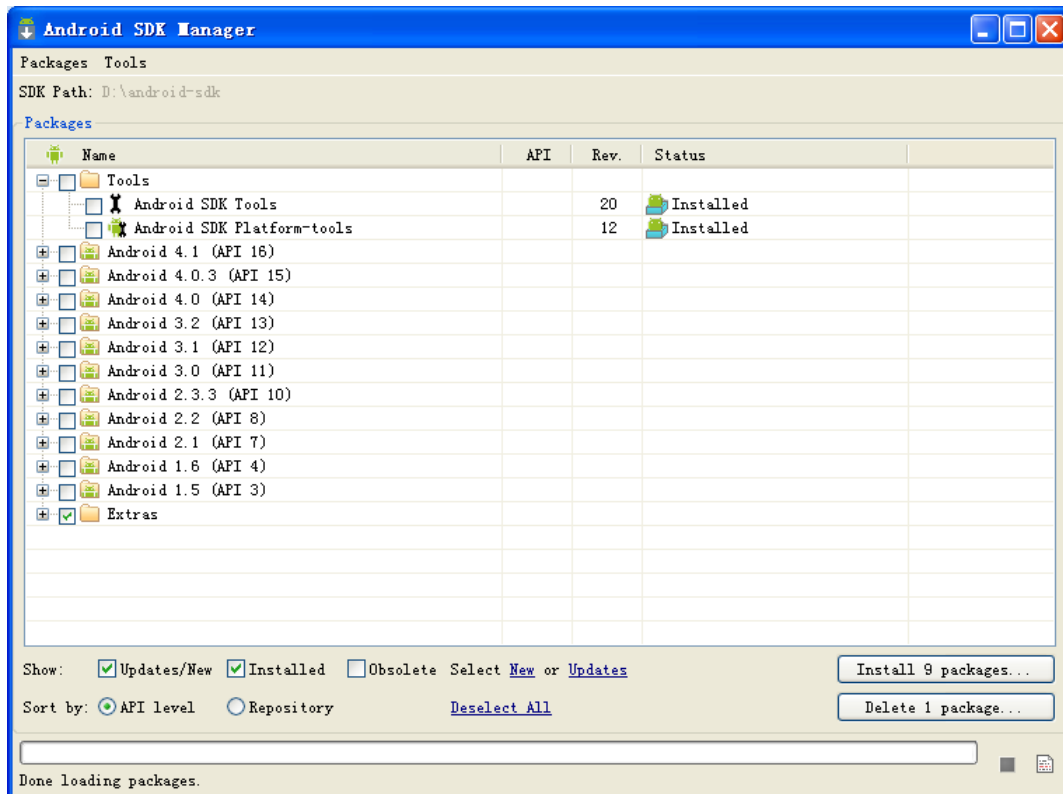


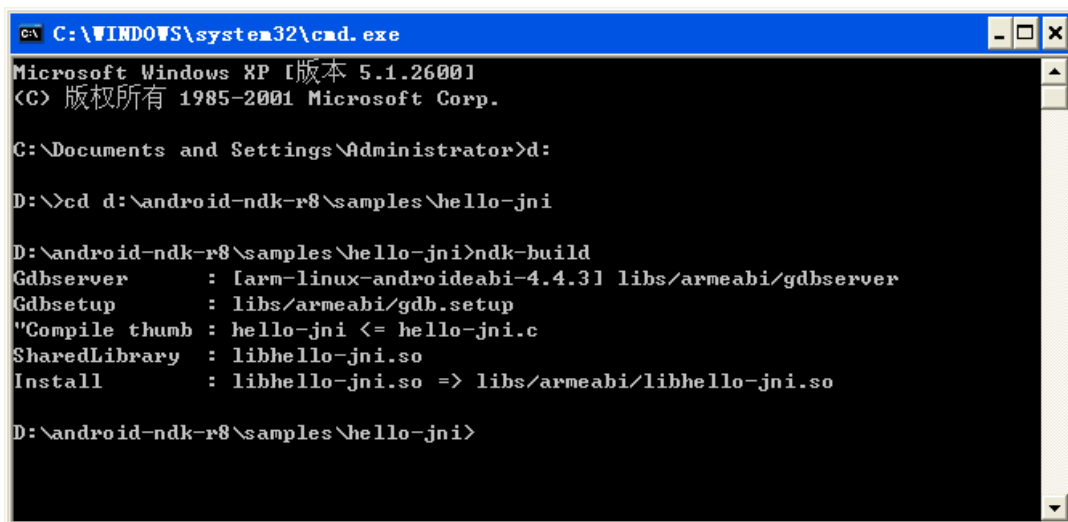
图1-6 Android SDK Manager运行界面

1.1.3 安装 Android NDK

Android NDK 是 Google 提供的开发 Android 原生程序的工具包。如今越来越多的软件与病毒采用了基于 Android NDK 动态库的调用技术，隐藏了程序在实现上的很多细节，掌握 Android NDK 程序的分析技术也成为了分析人员必备的技能，本书将会在第 7 章对 Android NDK 程序的特点以及分析技术进行详细的讲解。

Android NDK 的下载地址为：<http://developer.android.com/sdk/ndk/index.html>，目前最新版本为 R8，Windows 平台下的完整下载链接为：<http://dl.google.com/android/ndk/android-ndk-r8-windows.zip>。将下载后的压缩包解压到硬盘任意位置，本书为 D 盘根目录。新建环境变量 ANDROID_NDK，值为 D:\android-ndk-r8，然后将 ANDROID_NDK 添加到 PATH 环境变量中，做好这一步，Android NDK 就算安装完成了。

接下来测试配置是否正确，打开一个 CMD 窗口，进入目录“D:\android-ndk-r8\samples\hello-jni”，输入“ndk-build”命令编译 Android NDK 中自带的 hello-jni 工程，如果输出如图 1-7 所示的结果，就说明 Android NDK 安装成功了。



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>cd d:\android-ndk-r8\samples\hello-jni

D:\>cd d:\android-ndk-r8\samples\hello-jni

D:\android-ndk-r8\samples\hello-jni>ndk-build
Gdbserver      : [arm-linux-androideabi-4.4.3] libs/armeabi/gdbserver
Gdbsetup       : libs/armeabi/gdb.setup
"Compile thumb : hello-jni <= hello-jni.c
SharedLibrary  : libhello-jni.so
Install        : libhello-jni.so => libs/armeabi/libhello-jni.so

D:\android-ndk-r8\samples\hello-jni>
```

图1-7 使用Android NDK编译hello-jni工程

1.1.4 Eclipse 集成开发环境

Eclipse 是 Android 开发推荐使用的 IDE。它的下载地址为：<http://www.eclipse.org/downloads>，选择下载“Eclipse IDE for Java Developers”或“Eclipse for Mobile Developers”版本即可。强烈建议下载使用后者，后者自带了 CDT（C/C++ Development Tools）插件，并针对手机开发做了优化。

Eclipse 是一款绿色软件，下载完成后解压到硬盘任意目录，本书为 D 盘根目录。进入“D:\eclipse”目录，运行 eclipse.exe，Eclipse 会根据前面设置的环境变量自动进行初始化，如果启动时没有提示错误说明安装成功。

1.1.5 安装 CDT、ADT 插件

如果读者使用的 Eclipse 是 For Mobile Developers 版本或自带 CDT 插件，可以跳过 CDT 插件的安装；否则需要手动安装 CDT 插件。安装 Eclipse 的插件比较简单，有在线安装与离线安装两种方式，步骤分别为：

- 启动 Eclipse，点击菜单“Help→Install New Software”打开 Install 对话框，在“Work With”旁边的编辑框中输入 <http://download.eclipse.org/tools/cdt/releases/juno> 并回车，稍等片刻后下面列表框就会解析出 CDT 插件。
- 到 Eclipse 官网上手动下载最新版的 CDT 插件，目前最新 8.1.0。下载地址为：<http://www.eclipse.org/cdt/downloads.php>。启动 Eclipse，点击菜单“Help→Install New Software”打开 Install 对话框，点击界面上的 Add 按钮，打开 Add Repository 对话框，接着点击 Archive 按钮，选择下载的 CDT 压缩包，点击 OK 按钮返回。

无论采用上面哪一种方式进行安装，最终都会在 Name 下面的列表中列出可供安装的 CDT 插件，如图 1-8 所示，全部勾选后点击 Next 按钮即可安装，在线安装耗费的时间根据网络环境差异可能有所不同。

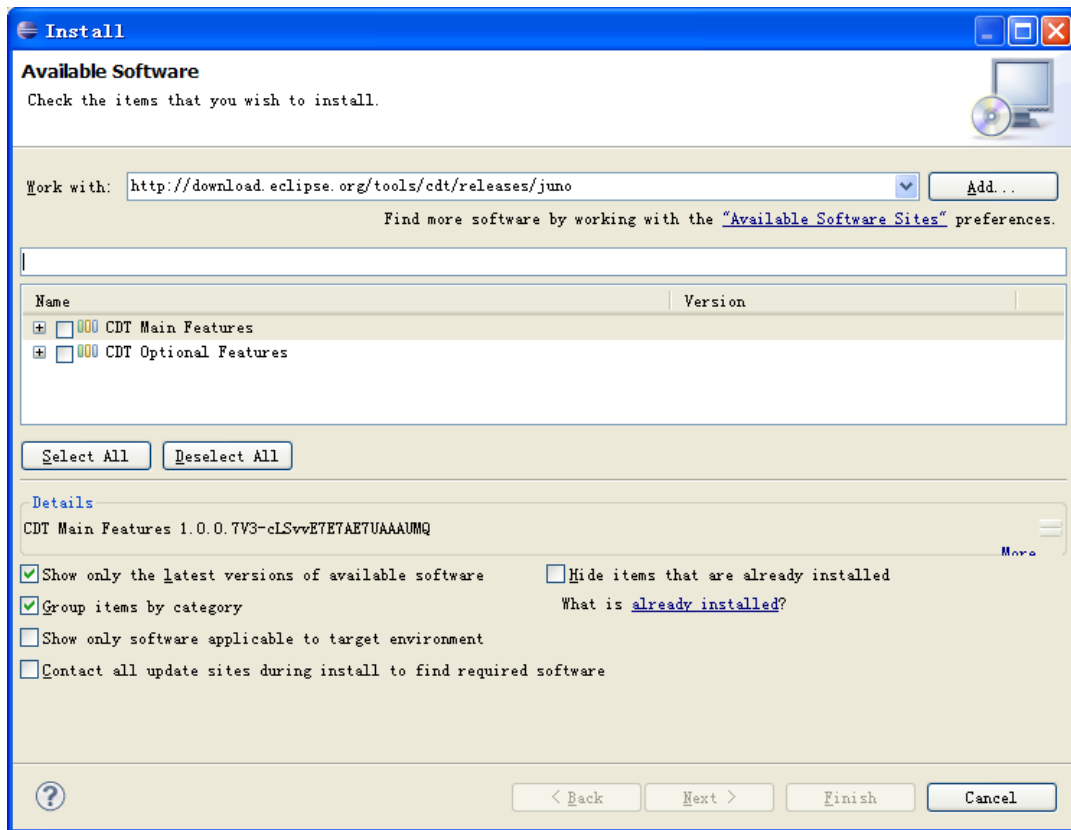


图1-8 安装CDT插件

ADT 插件是 Google 为 Android 开发提供的 Eclipse 插件，方便在 Eclipse 开发环境中创建、编辑、调试 Android 程序。安装过程与 CDT 插件类似。目前最新版本为 20.0.0，官方下载地址为：<http://dl.google.com/android/ADT-20.0.0.zip>，在线安装的 repository 地址为：<https://dl-ssl.google.com/android/eclipse/>，读者可以按照上面的步骤自行完成安装。

ADT 插件安装完成后需要进行相应的配置。点击 Eclipse 菜单项“Window-Preferences”，选择 Android 列表项，在右侧 SDK Location 处选择 Android SDK 的安装位置，如 D:\android-sdk，展开 Android 列表项，选择 NDK，在右侧 NDK Location 处选择 Android NDK 的安装位置，如 D:\android-ndk-r8。设置完后点击 OK 按钮关闭对话框。到此，CDT 与 ADT 插件就安装完成了。

1.1.6 创建 Android Virtual Device

Android SDK 中提供了“Android Virtual Device Manager”工具，方便在没有真实 Android 设备环境的情况下调试运行 Android 程序。

双击运行“D:\android-sdk-windows\AVD Manager.exe”，点击“New”按钮，打开 AVD 创建对话框，在“Name”栏输入 AVD 的名称，如输入“Android2.3.3”，在 Target 一栏选择要模拟的 Android 版本，这里选择“Android 2.3.3 – API Level 10”，SD Card 大小指定为 256MB，其它选项保持不变，结果如图 1-9 所示。

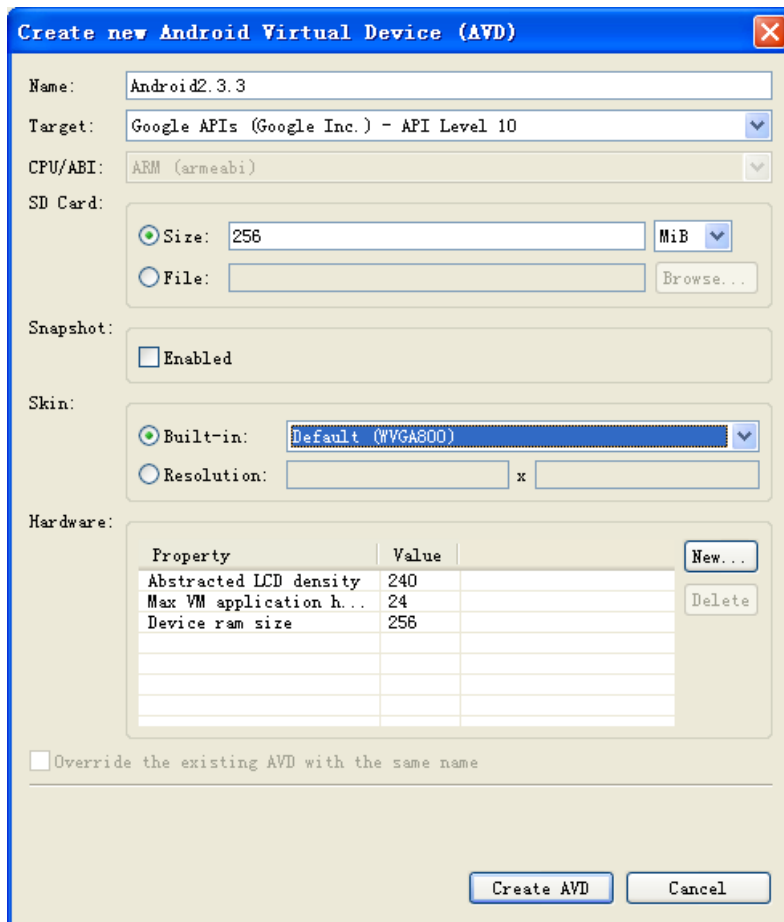


图1-9 创建AVD

点击“Create AVD”按钮完成 AVD 的创建。选中创建的 AVD，点击右侧的“Start”按钮，如果没有错误会成功启动这个 Android 虚拟设备。

如果使用真实 Android 设备来调试程序，需要先在设备的“设置→程序→开发”选项中开启“USB 调试”，然后安装相应设备厂商提供的 USB 驱动程序。一份常见的 USB 驱动程序下载地址列表可以在 <http://developer.android.com/tools/extras/oem-usb.html> 中找到。安装完驱动后在命令提示符下输入“adb devices”就可以列出连接的 Android 设备了。

1.1.7 使用到的工具

分析 Android 程序需要用到很多工具，包括：反编译工具、静态分析工具、动态调试工具、集成分析环境等。所有的工具中，大多数是开源或免费的软件，笔者在此不详细列出使用到的工具，在每一章对知识点进行介绍时，会同时介绍相关工具的使用方法与下载地址。

1.2 Linux 分析环境搭建

本书中介绍的 Windows 环境下的大多数操作，在 Linux 环境下同样适用。本节将介绍如何在 Linux 环境下搭建 Android 程序的分析环境。

1.2.1 本书的 Linux 环境

本书在写作时使用的 Linux 环境为 Ubuntu 10.04 32 位系统，在最后一章进行 Android

病毒分析时使用的是 Ubuntu 12.04 32 位系统。

1.2.2 安装 JDK

首先在当前用户主目录下新建 tools 目录来存放 Android 分析常用的工具。

到 Oracle 官方网站下载 JDK 安装包。本书 Ubuntu 平台使用的版本为 jdk-6u33-linux-i586，下载地址为 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>，将下载的 jdk-6u33-linux-i586.bin 文件放到 /home/feicong/tools 目录（feicong 为本机用户名），打开一个终端环境输入以下命令：

```
cd tools
chmod +x ./jdk-6u33-linux-i586.bin
./jdk-6u33-linux-i586.bin
```

此时安装程序会自动将 JDK 安装到当前目录的 jdk1.6.0_33 目录下。接着设置环境变量，执行：

```
sudo gedit /etc/profile
```

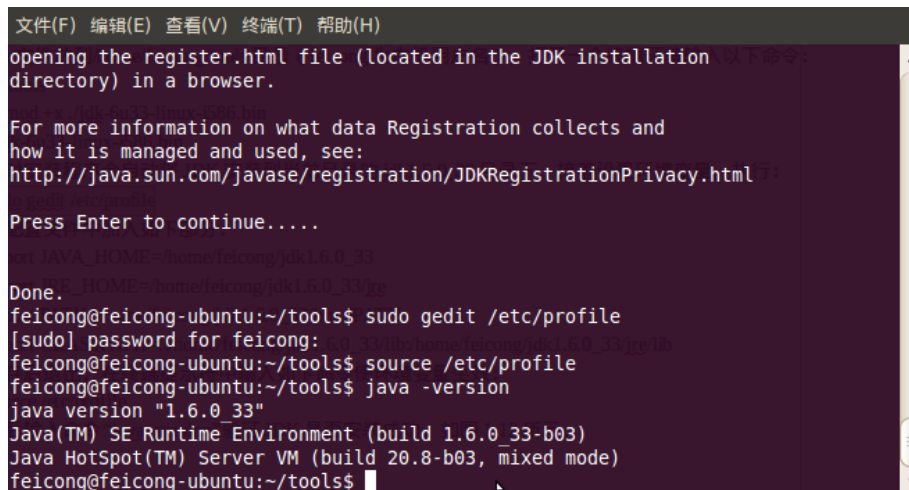
在配置文件中加入如下部分：

```
export JAVA_HOME=/home/feicong/jdk1.6.0_33
export JRE_HOME=/home/feicong/jdk1.6.0_33/jre
export PATH=/home/feicong/jdk1.6.0_33/bin:$PATH
export CLASSPATH=.:/home/feicong/jdk1.6.0_33/lib:/home/feicong/jdk1.6.0_33/
jre/lib
```

保存后退出，在终端提示符中输入如下命令使环境变量生效：

```
source /etc/profile
```

最后输入命令“java -version”验证 JDK 是否安装成功，如图 1-10 所示。



```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
opening the register.html file (located in the JDK installation
directory) in a browser.

For more information on what data Registration collects and
how it is managed and used, see:
http://java.sun.com/javase/registration/JDKRegistrationPrivacy.html

Press Enter to continue....

Done.
feicong@feicong-ubuntu:~/tools$ sudo gedit /etc/profile
[sudo] password for feicong:
feicong@feicong-ubuntu:~/tools$ source /etc/profile
feicong@feicong-ubuntu:~/tools$ java -version
java version "1.6.0_33"
Java(TM) SE Runtime Environment (build 1.6.0_33-b03)
Java HotSpot(TM) Server VM (build 20.8-b03, mixed mode)
feicong@feicong-ubuntu:~/tools$
```

图1-10 验证JDK是否安装成功

1.2.3 在 Ubuntu 上安装 Android SDK

Ubuntu 上安装 Android SDK 与 Windows 安装步骤类似，首先到官方网站 <http://developer.android.com/sdk/index.html> 下载 Android SDK，目前最新版本为 r20，下载后将 android-sdk_r20-linux.tgz 压缩包文件放到 tools 目录，执行以下命令进行解包：

```
tar zvxf android-sdk_r20-linux.tgz
```

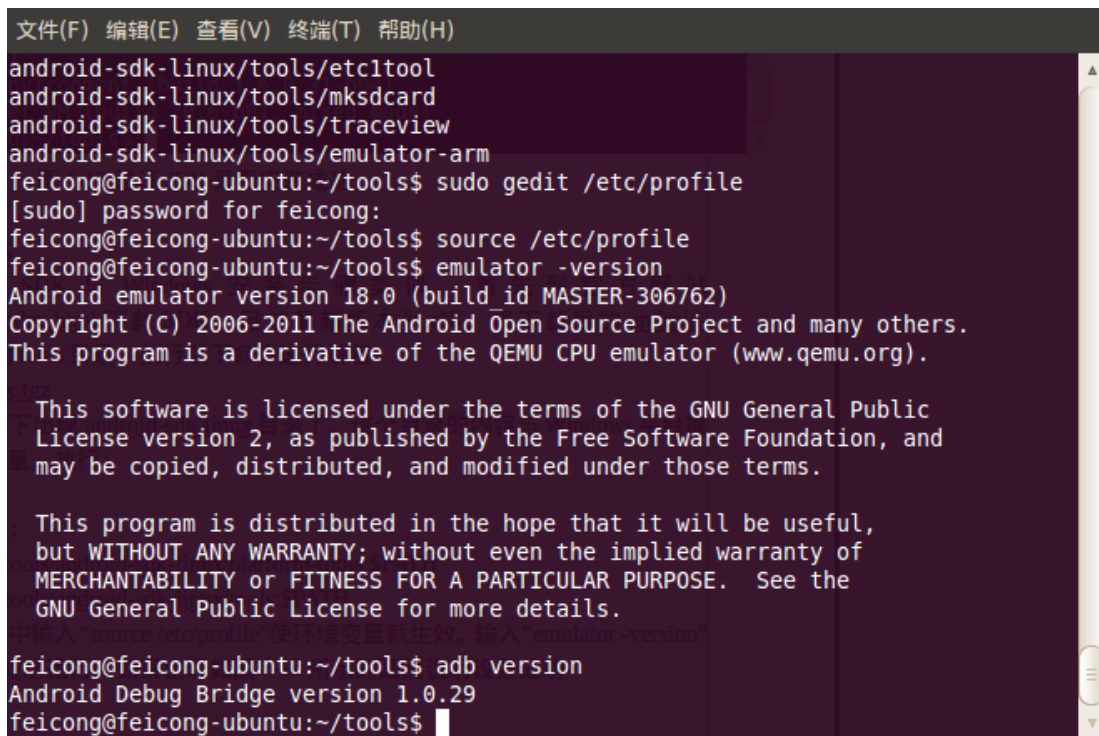
解包完毕后就会在当前目录下出现 `android-sdk-linux` 目录了。这个目录的内容与 Windows 平台提供的工具类似。接着设置环境变量，执行：

```
sudo gedit /etc/profile
```

在配置文件中加入如下部分：

```
export PATH=/home/feicong/tools/android-sdk-linux/platform-tools:$PATH  
export PATH=/home/feicong/tools/android-sdk-linux/tools:$PATH
```

保存后退出，在终端提示符中输入“`source /etc/profile`”使环境变量生效。输入“`emulator -version`”与“`adb version`”命令查看是否能成功运行。如果出现如图 1-11 所示的画面说明设置成功。



```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)  
android-sdk-linux/tools/etcltool  
android-sdk-linux/tools/mksdcard  
android-sdk-linux/tools/traceview  
android-sdk-linux/tools/emulator-arm  
feicong@feicong-ubuntu:~/tools$ sudo gedit /etc/profile  
[sudo] password for feicong:  
feicong@feicong-ubuntu:~/tools$ source /etc/profile  
feicong@feicong-ubuntu:~/tools$ emulator -version  
Android emulator version 18.0 (build_id MASTER-306762)  
Copyright (C) 2006-2011 The Android Open Source Project and many others.  
This program is a derivative of the QEMU CPU emulator (www.qemu.org).  
  
This software is licensed under the terms of the GNU General Public  
License version 2, as published by the Free Software Foundation, and  
may be copied, distributed, and modified under those terms.  
  
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.  
  
feicong@feicong-ubuntu:~/tools$ adb version  
Android Debug Bridge version 1.0.29  
feicong@feicong-ubuntu:~/tools$
```

图1-11 测试Android SDK

配置好环境后就需要下载具体版本的 SDK 了，在终端提示符中输入 `android` 命令启动 Android SDK Manager，接下来的下载步骤与 Windows 平台是一样的，具体操作这里就不再赘述了。

1.2.4 在 Ubuntu 上安装 Android NDK

首先到 Android 官方网站 <http://developer.android.com/sdk/ndk/index.html> 下载 Android NDK，目前 Linux 平台的最新版本为 r18，将下载的 `android-ndk-r8-linux-x86.tar.bz2` 文件放到 `tools` 目录，在终端提示符下输入以下命令解包：

```
tar jxvf ./android-ndk-r8-linux-x86.tar.bz2
```

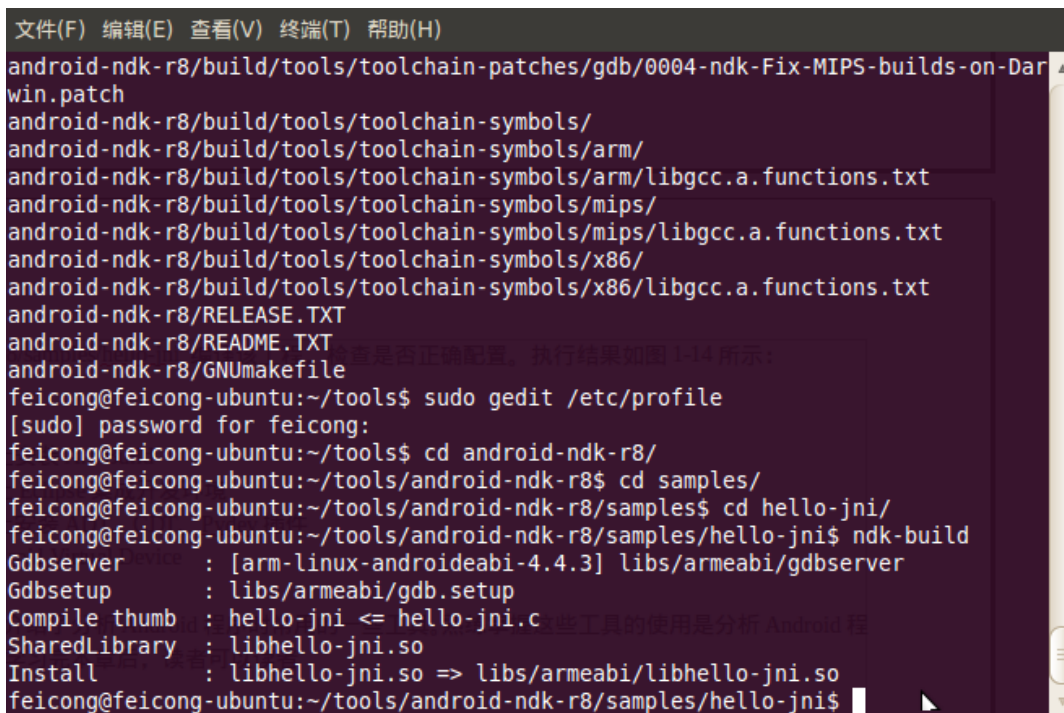
解包完毕后就会在当前目录下出现 `android-ndk-r8` 目录了。接着设置环境变量，执行：

```
sudo gedit /etc/profile
```

在配置文件中加入如下部分：

```
export ANDROID_NDK=/home/feicong/tools/android-ndk-r8  
export PATH=/home/feicong/tools/android-ndk-r8:$PATH
```

保存文件后退出，在终端提示符中输入“source /etc/profile”使环境变量生效。接下来在终端提示符下进入 android-ndk-r8/samples/hello-jni 目录，然后输入 ndk-build 命令编译 hello-jni 工程，如果配置正确，执行结果如图 1-12 所示。



```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
android-ndk-r8/build/tools/toolchain-patches/gdb/0004-ndk-Fix-MIPS-builds-on-Dar
win.patch
android-ndk-r8/build/tools/toolchain-symbols/
android-ndk-r8/build/tools/toolchain-symbols/arm/
android-ndk-r8/build/tools/toolchain-symbols/arm/libgcc.a.functions.txt
android-ndk-r8/build/tools/toolchain-symbols/mips/
android-ndk-r8/build/tools/toolchain-symbols/mips/libgcc.a.functions.txt
android-ndk-r8/build/tools/toolchain-symbols/x86/
android-ndk-r8/build/tools/toolchain-symbols/x86/libgcc.a.functions.txt
android-ndk-r8/RELEASE.TXT
android-ndk-r8/README.TXT
android-ndk-r8/GNUMakefile
feicong@feicong-ubuntu:~/tools$ sudo gedit /etc/profile
[sudo] password for feicong:
feicong@feicong-ubuntu:~/tools$ cd android-ndk-r8/
feicong@feicong-ubuntu:~/tools/android-ndk-r8$ cd samples/
feicong@feicong-ubuntu:~/tools/android-ndk-r8/samples$ cd hello-jni/
feicong@feicong-ubuntu:~/tools/android-ndk-r8/samples/hello-jni$ ndk-build
Gdbserver      : [arm-linux-androideabi-4.4.3] libs/armeabi/gdbserver
Gdbsetup       : libs/armeabi/gdb.setup
Compile thumb  : hello-jni <= hello-jni.c
SharedLibrary  : libhello-jni.so
Install        : libhello-jni.so => libs/armeabi/libhello-jni.so
feicong@feicong-ubuntu:~/tools/android-ndk-r8/samples/hello-jni$
```

图1-12 使用Android NDK编译工程

1.2.5 在 Ubuntu 上安装 Eclipse 集成开发环境

首先到 Eclipse 官方网站 <http://www.eclipse.org/downloads/> 下载 Eclipse IDE for Java Developers 版本，将下载到的 eclipse-jee-indigo-SR2-linux-gtk.tar.gz 文件放到 tools 目录，输入以下命令解包：

```
tar zxvf ./eclipse-jee-indigo-SR2-linux-gtk.tar.gz
```

解包完毕后就会在当前目录下出现 eclipse 目录。目录中的 eclipse 文件就是主程序，为方便以后使用可以在桌面上建立快捷方式。

1.2.6 在 Ubuntu 上安装 CDT、ADT 插件

Ubuntu 上安装 CDT、ADT 插件与 Windows 平台上的安装步骤是一样的，读者可以参考 1.1.5 小节内容进行安装，这里不再赘述。

1.2.7 创建 Android Virtual Device

Linux 版的 Android SDK 没有提供可视化的 AVD Manager 管理工具，创建 AVD 可以使用 android 命令。在终端提示符下输入“android list targets”列出本机已经下载好的 SDK，本机输出结果如下：

```
feicong@feicong-ubuntu:~/tools$ android list targets
Available Android targets:
.....
-----
id: 2 or "android-8"
```

```
Name: Android 2.2
Type: Platform
API level: 8
Revision: 3
Skins: WVGA800 (default), HVGA, QVGA, WVGA854, WQVGA432, WQVGA400
ABIs : armeabi
-----
id: 3 or "android-10"
Name: Android 2.3.3
Type: Platform
API level: 10
Revision: 2
Skins: WVGA800 (default), HVGA, QVGA, WVGA854, WQVGA432, WQVGA400
ABIs : armeabi

.....
-----
id: 6 or "android-15"
Name: Android 4.0.3
Type: Platform
API level: 15
Revision: 3
Skins: WSVGA, WVGA800 (default), HVGA, QVGA, WVGA854, WXGA720, WXGA800,
WQVGA432, WQVGA400
ABIs : armeabi-v7a
```

每一个 id 对应一个版本的 SDK。这个 id 在创建 AVD 时会使用到。创建 AVD 的命令格式为“`android create avd --name <your_avd_name> --target <targetID>`”，比如想要创建 Android 系统版本为 2.3.3 且名称为 `android2.3.3` 的 AVD 只需在终端提示符下输入如下命令：

```
android create avd --name android2.3.3 --target android-10
```

创建 AVD 完成后可以使用 `emulator` 来启动它，在终端提示符下输入命令：

```
emulator -avd android2.3.3
```

最终运行效果如图 1-13 所示。

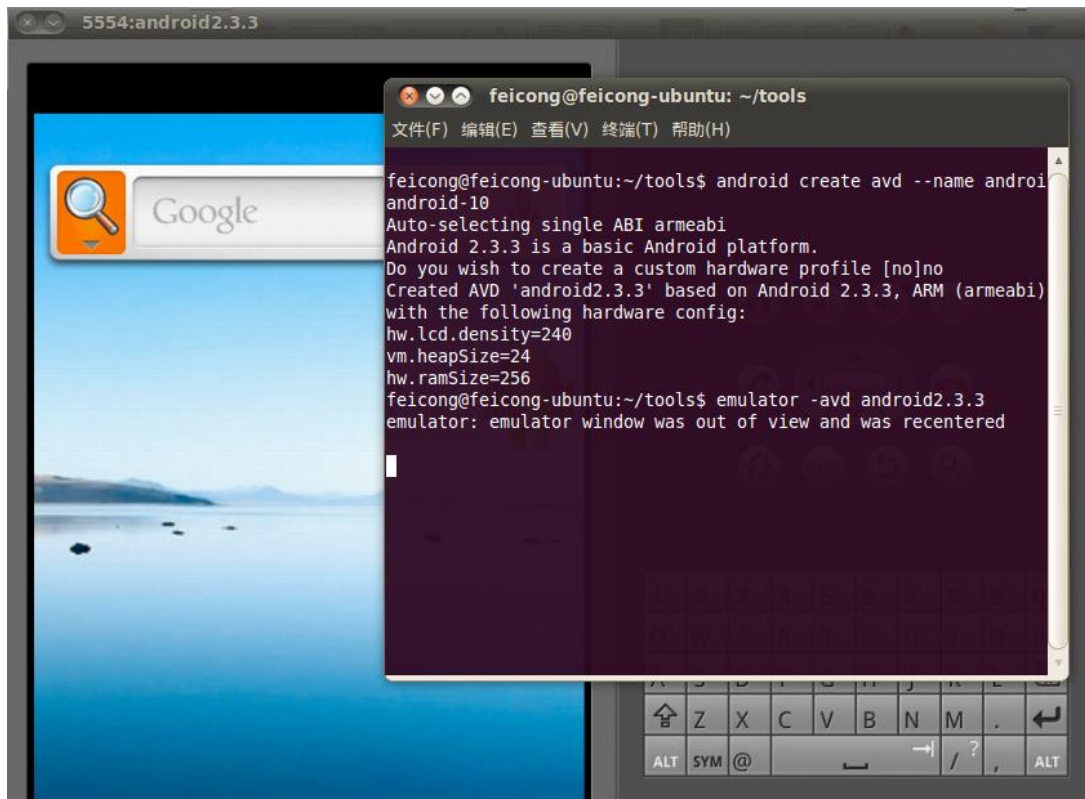


图1-13 Android模拟器运行界面

如果使用真实 Android 设备来调试程序，还需要做一些工作。首先需要在设备的“设置→程序→开发”选项中开启“USB 调试”，接着将设备连接电脑，在终端提示符中输入 `lsusb` 命令查看连接的 USB 设备。我的测试机型为 Moto XT615，命令执行后会得到如下输入。

```
feicong@feicong-ubuntu:~$ lsusb
Bus 003 Device 002: ID 15d9:0a4c Dexon
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 005: ID 0bda:0158 Realtek Semiconductor Corp. Mass Storage Device
Bus 001 Device 004: ID 22b8:2de6 Motorola PCS
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

其中 22b8 为 Vendor id 值，2de6 为 Product id。不同的设备厂商 Vendor id 值不同。可以在 <http://developer.android.com/tools/device.html#VendorIds> 找到一份常见设备厂商的 Vendor id 列表。Product id 则是具体产品的 id 值。同一厂商的不同设备 Vendor id 相同而 Product id 不同。记录下 Vendor id 与 Product id 值，然后编辑 udev 规则文件 `/etc/udev/rules.d/70-android.rules`，没有则创建，内容如下。

```
SUBSYSTEM=="usb", ATTR{idVendor}=="22b8", MODE="2de6", GROUP="plugdev"
```

其中的 22b8 与 2de6 根据自己的 Vendor id 与 Product id 值进行相应的更改，修改保存后退出，在终端提示符中输入命令“`adb devices`”就能列出配置好的 Android 设备了。

1.2.8 使用到的工具

本书讲解 Android 程序分析时使用到的工具大多数是跨平台的，这些工具同时拥有

Windows 版本与 Linux 版本，笔者在介绍它们时会给出相应工具的下载地址，并且给出其安装方法。另外，本书中也有个别的工具是与平台相关的，笔者在书中都有详细的说明。

1.3 本章小结

本章主要介绍了分析 Android 程序时常用到的一些工具，熟练掌握这些工具的使用是分析 Android 程序的基础，接着介绍了 Windows 平台与 Ubuntu 平台下 Android 开发环境的搭建，为后面的分析环境做准备。在学习完本章后，读者可以自行下载本章所提及到的工具，动手练习如何使用它们，这些工具的具体使用会在本书后面章节逐一进行介绍。