

Compilation of the Android Source Codes in the Ubuntu System

Zhenglu Jiang

School of Mathematical and Computational Science, Sun Yat-Sen University, Guangzhou Guangdong
Email: mcsjzl@mail.sysu.edu.cn

Received: May 21st, 2015; accepted: Jun. 6th, 2015; published: Jun. 10th, 2015

Copyright © 2015 by author and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

It is very important for the beginners to compile the Android source codes. But various problems appear if any software is not installed in the Ubuntu system together with its environment variables. This paper introduces in turn the install of some pieces of software in the Ubuntu system and their settings of environment variables, shows how to download and compile the Android source codes, and finally gives the methods of solving some general problems which perhaps occur in the compilation process.

Keywords

Ubuntu System, Environment Variable, Android Compilation

Ubuntu系统下Android源码编译

姜正禄

中山大学数学与计算科学学院, 广东 广州
Email: mcsjzl@mail.sysu.edu.cn

收稿日期: 2015年5月21日; 录用日期: 2015年6月6日; 发布日期: 2015年6月10日

摘 要

Android源码编译对一个初学者来说非常重要。可是, 如果Ubuntu系统下软件和环境变量没装好, 就会

出现各种各样的问题。本文依次介绍Ubuntu系统下软件安装和环境设置，展示如何下载和编译Android源码，并对可能出现的常见问题提出相应的解决方案。

关键词

Ubuntu系统，环境变量，Android编译

1. 引言

Android，中文名称“安卓”，是一种自由开放源代码的操作系统，主要用于诸如智能手机、平板电脑、电视、数码相机和游戏机等移动设备。它最初由美国的软件工程师 Andy Rubin 开发，后来被 Google 公司收购。之后，Google 公司与硬件制造商、软件开发商及电信运营商组建一个开放手机联盟，以 Apache 开源许可证的授权方式发布 Android 的源代码，大家可以共同研发。目前，采用 Android 系统的设备数量不断上升，研发队伍也在不断壮大，这套系统也越来越好，也越来越被大众喜爱。作为众多爱好者之一，近来开始热衷于 Ubuntu 系统下的 Android 源码编译，遇到各种各样的问题。网上有关这方面的介绍很多，各有优点，可是，地点变了，时间变了，系统环境不一样了，这些介绍都有这样那样的问题。为了让其他喜好者少走弯路，本文介绍 Ubuntu 系统和 Java [1] [2] 软件下载和安装、以及 Android 的源码下载和编译，并对可能出现的常见问题提出相应的解决方案。不同版本 Ubuntu 系统中的 gcc 编译器、Java 编译器和 Android 之间的兼容性尤为重要，在下载安装时要特别注意。本文的工作之一就是解决了这三者间的兼容性。Android 编译仿真时的环境变量设置也很重要，给出这些变量的合理设置便是本文的工作之二。另外，还解决了部分 Android 源码中的 64 进制编译问题，用 sed 命令语句直接修改部分 Android 源码。

这种安装编译还是需要一定的硬件要求的。一般来说，奔腾或 AMD 的 CPU 都行，内存最低 512MB，硬盘最低 30 GB，当然，CPU 配制越高，编译运行就越快，存储容量越大就越好。这次在老旧的型号为 Thinkpad R51e 的笔记本电脑上进行实验，硬件配置相对现在的科技来说很低，参见列表 1。

另外，还准备了一个 8 GB 的 U 盘，用于制作 Ubuntu 系统安装 U 盘。

除了硬件要求之外，大家还需要会操作 Linux 系统的基本命令，比如，Linux 的 sed 用法往往在修改文件内容时很有用，能编写 shell 脚本就更好了。推荐大家参见美国 Richard Blum 和 Christine Bresnahan 编著的书[3]，这本书已于 2012 年由武海峰翻译成中文，中文书名为《Linux 命令行与 shell 脚本编程大全》[4]。如果具备有关 Android 系统的基本知识[5]-[14]，那就更好了。

通过实验，可以掌握 Ubuntu 系统、Java 软件和 Android 源码的下载安装以及 Android 源码编译的基本技能，领会 Android 源码编译与 Ubuntu 系统和 Java 软件之间的兼容性，缺少其中任何环节都不行。Android 源码编译可谓是个庞大的系统工程，在这基础上，可以进行更加广泛的软件开发。

2. 系统软件安装和环境变量设置

Android 的源码是基于 Linux 系统的。为此，首先要下载并安装一个 Linux 系统。由于电脑 CPU 不

Table 1. The hardware configuration of the Thinkpad R51e laptop

表 1. Thinkpad R51e 笔记本电脑的硬件配置

CPU	英特尔 1.70 GHz 主频，Dothan 核心，400 MHz 前端总线，1 MB 二级缓存
显卡	ATI Radeon Xpress 200 M
内存	756 MB DDR2 533 MHz
硬盘	80 GB 4500 转

支持 64 进制，所以本次实验选用的是 32 进制的 Ubuntu 系统。如 CPU 支持 64 进制，就选用 64 进制的系统。Ubuntu 系统有许多的镜像网址，最好选个国内的网址。从如下的 Sohu 镜像网址：

<http://mirrors.sohu.com/ubuntu-releases/10.04.4/release/>

就可下载 ubuntu-10.04.4-dvd-i386.iso，然后用 UltraISO 或电脑店软件将这个映像文件刻录到 U 盘，最后用这个装有 Ubuntu 系统安装软件的 U 盘将 32 进制的 Ubuntu 系统安装到电脑。实验发现这个版本的 Ubuntu 系统更适合编译 Android 源码。

启动电脑，进入 Ubuntu 系统，设置好网络配置，联网。接着，还需要安装下载和编译 Android 源码所需的相关软件。为此，先在终端执行

```
sudo apt-get update
```

去对 Ubuntu 系统的升级包更新，如果不执行此更新，就无法进行接下来的软件安装。再在终端执行如下命令行：

```
sudo apt-get install git-core curl
sudo apt-get install gnupg flex bison gperf build-essential zip
sudo apt-get install tofrodos pngcrush schedtool squashfs-tools
sudo apt-get install g++-multilib python-markdown mingw32
sudo apt-get install libxml2-utils xsltproc zlib1g-dev libc6-dev
sudo apt-get install libncurses5-dev libreadline5-dev libz-dev
sudo apt-get install libsdl1.2-dev libesd0-dev libwxgtk2.6-dev
sudo apt-get install zlib1g-dev libgl1-mesa-dev libx11-dev
sudo apt-get install lib64z1-dev x11proto-core-dev
```

其中，第一行是下载 Android 源码所需的工具，其它行是编译 Android 源码所需的软件。另外，还要下载 Java。大家可以从网上获取 jdk-6u45-linux-i586.bin 并保存在目录~/下，在终端执行：

```
chmod a+x ~/jdk-6u45-linux-i586.bin & sudo ~/jdk-6u45-linux-i586.bin
```

就把 Java 安装在 Ubuntu 系统上了，Java 就在目录~/jdk1.6.0_45 下。Java 官方网址如下：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

从此可以获取最新版本，不过，最新版本不支持较早的 Android 源码编译。获取旧版的 Java 的网址很多，比如，<http://ghaffarian.net/downloads/Java/JDK/>，这个网址上还有其他版本，可自由下载。为了能正常运行 Java，还需设置 Java 运行的环境变量，并把这些变量存放在文件 jdk.bashrc 中。为此，可在终端执行如下命令：

```
echo "export JAVA_HOME=$HOME/jdk1.6.0_45" >> jdk.bashrc
echo "export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib/dt.jar" >> jdk.bashrc
echo "export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib/tools.jar" >> jdk.bashrc
echo "export PATH=$HOME/bin:$JAVA_HOME/bin:$PATH" >> jdk.bashrc
source jdk.bashrc
```

其中前 4 行将环境变量写入文件 jdk.bashrc 中；JAVA_HOME 是指向 jdk 的安装目录的环境变量，Java 编译器通过搜索 JAVA_HOME 变量来找到并使用安装好的 jdk；CLASSPATH 环境变量指定类搜索路径，Java 编译器通过 CLASSPATH 来寻找已经编写好的类；PATH 环境变量指定命令搜索路径；最后一行使得这些变量生效。在终端运行 java -version，如显示 Java 版本号，就说明安装设置成功。到这里，相关软件安装准备就绪。如果不设置这些环境，Android 源码环境变量安装时就会出错停止，显示要求安装 Java 1.6。

3. Android 源码下载、编译和仿真

有了前面的准备工作，下面就可以下载和编译 Android 的源码了。

国内 Android 的源码镜像有，下载下来的都无法编译，不是缺这就是缺那。上 Google 公司网址 google.com 嘛，连接不上，据说被封了，至于被谁封，那就搞不清了。就连 gmail.com 同样连接不上。上网查询一下，发现可以通过第三方的镜像网址下载。下面就讲述具体的下载步骤。

要下载 Android 的源码，首先需要下载 repo，它只是 google 用 Python 脚本写的调用 git 的一个脚本，主要是用来下载和管理 Android 项目的软件仓库 [15]。由于无法连接 android.git.kernel.org 和 git-repo.googlecode.com 等 google 网址，所以只能通过第三方获取。先在用户根目录下新建一个子目录 bin，并设置好路径，然后用 curl 从一个第三方连接网址中获取 repo，为此，在终端运行如下命令就可下载：

```
mkdir -p ~/bin & export PATH=~/.bin:$PATH
curl "http://php.webtutor.pl/en/wp-content/uploads/2011/09/repo" >~/bin/repo
chmod a+x ~/bin/repo
```

其中最后一行命令改变 repo 的属性，使得 repo 成为一个可以执行的脚本。如果原先有这个脚本，那就不需要这些步骤了。接着，同样在用户根目录下新建一个子目录 work，并在这个子目录 work 下开始下载 Android 的源码。为此，可在终端运行如下命令：

```
mkdir -p ~/work & cd ~/work
repo init -u git://git.omapzoom.org/platform/manifest.git -b android-2.3.3_r1
cd ~/work/.repo/manifests & cp default.xml default.xml.bk
sed -i -e 's/fetch=".." /fetch="git://git.omapzoom.org/" /g' default.xml
cd ~/work & repo sync -j8
```

其中，执行第 3 行和第 4 行命令的目的就是要修改 google 默认网址为第三方网址，最后一行命令指的是回到这个子目录 work 来执行同步下载 Android 源码。特别要说明的是，在执行第 2 行命令的过程中，还需要 gmail 的账户和密码。如果没有，就得借助第三方软件登入 gmail.com 网址注册一个用户。再到 google 下载 Android 源码的网址去用自己的 gmail.com 信箱注册获得下载许可，并把下载许可的账户名和密码按要求的格式存放在一个文本文件中，其存放格式如下：

```
machine android.googleusercontent.com login git-***.gmail.com password *$*
machine android-review.googleusercontent.com login git-***.gmail.com password *$*
```

其中，***表示下载许可的账户名，*\$*表示密码。而且这个文件必须以.netrc 为名存放在用户根目录下。这步是必须的，需要在开始下载 Android 源码之前完成。如果一切顺利，那就等待下载完吧。

以上命令是下载版本号为 2.3.3_r1 的 Android 源码。如果不知道那种版本可以下载，那就执行如下命令：

```
cd ~/work & repo init -u git://git.omapzoom.org/platform/manifest.git
cd ~/work/.repo/manifests & git branch -a > ~/android-branch.txt
```

这时，在用户目录下就有一个文本文件 android-branch.txt，打开查看即可。

由于我们采用 32 进制 Ubuntu 系统，所以还得将版本号为 2.3.3_r1 的 Android 源码中设定的 64 进制编译选项改为 32 进制的编译选项，这些选项就在 5 个文件中。在子目录 work 下，只需在终端作如下命令：

```
sed -i -e '74a ifneq (0,0)' -e '81a endif' ./build/core/main.mk
```

```
sed -i -e 's/FLAGS += -m64/FLAGS += -m32/' ./external/clearsilver/cgi/Android.mk
sed -i -e 's/FLAGS += -m64/FLAGS += -m32/' ./external/clearsilver/cs/Android.mk
sed -i -e 's/FLAGS += -m64/FLAGS += -m32/' ./external/clearsilver/java-jni/Android.mk
sed -i -e 's/FLAGS += -m64/FLAGS += -m32/' ./external/clearsilver/util/Android.mk
```

就对这些文件完成了 32 进制的编译设置。如果 Ubuntu 系统是 64 进制的，这些文件就不需要修改了。

有了前面的准备，就可以进行 Android 源码编译，终端执行下列命令即可：

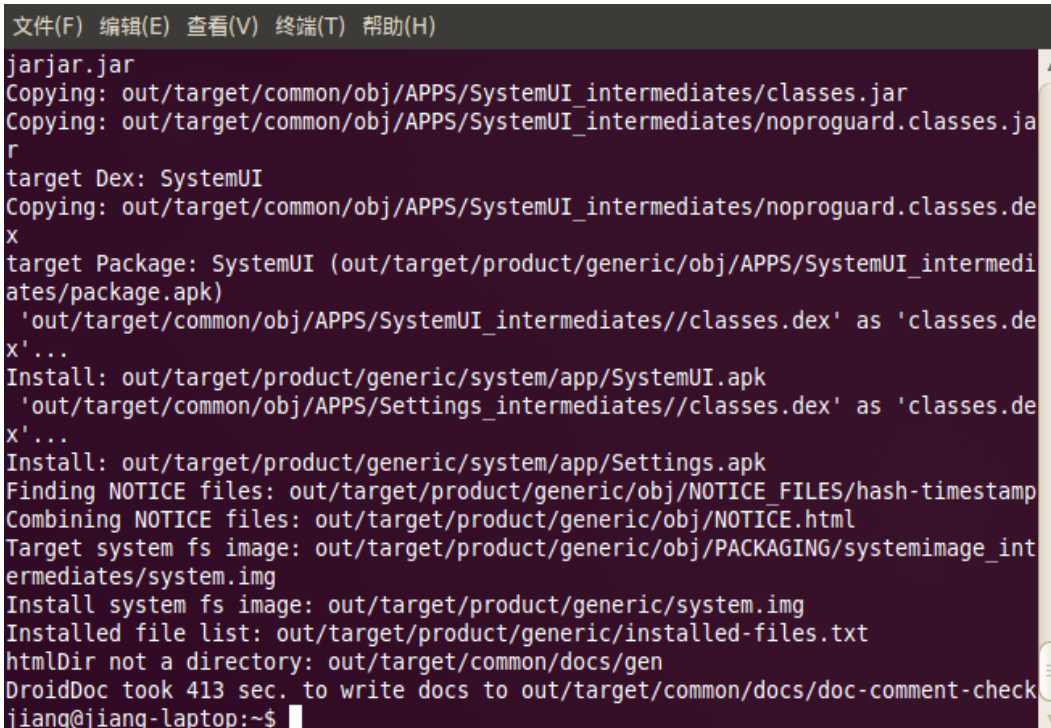
```
cd ~/work & source ~/jdk.bashrc & source ./build/envsetup.sh
lunch generic-eng & make -j8
```

这里就不多解释了。耐心等待编译完成之后，终端出现如图 1 所示的结果，这表明已成功编译好了。

接着，就可以仿真显示了。终端运行如下命令：

```
export ANDROID_PRODUCT_OUT=~/.work/out/target/product/generic
ANDROID_PRODUCT_OUT_BIN=~/.work/out/host/linux-x86/bin
export PATH=${PATH}:${ANDROID_PRODUCT_OUT_BIN}
export PATH=${PATH}:${ANDROID_PRODUCT_OUT}
cd ${ANDROID_PRODUCT_OUT}
emulator -system system.img -data userdata.img -ramdisk ramdisk.img
```

这里，前 4 行是设置仿真的环境变量，ANDROID_PRODUCT_OUT 指向 Android 系统映像文件所在的目录，ANDROID_PRODUCT_OUT_BIN 指向 Android 系统的仿真执行文件所在的目录；第 5 行转入 Android 系统映像文件所在的目录；第 6 行仿真运行 Android 系统。如果不把 ANDROID_PRODUCT_OUT 和 ANDROID_PRODUCT_OUT_BIN 添加到命令搜索路径 PATH 中，执行第 6 行时就会出错，显示没有



```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jarjar.jar
Copying: out/target/common/obj/APPS/SystemUI_intermediates/classes.jar
Copying: out/target/common/obj/APPS/SystemUI_intermediates/noproguard.classes.jar
target Dex: SystemUI
Copying: out/target/common/obj/APPS/SystemUI_intermediates/noproguard.classes.dex
target Package: SystemUI (out/target/product/generic/obj/APPS/SystemUI_intermediates/package.apk)
'out/target/common/obj/APPS/SystemUI_intermediates//classes.dex' as 'classes.dex'...
Install: out/target/product/generic/system/app/SystemUI.apk
'out/target/common/obj/APPS/Settings_intermediates//classes.dex' as 'classes.dex'...
Install: out/target/product/generic/system/app/Settings.apk
Finding NOTICE files: out/target/product/generic/obj/NOTICE_FILES/hash-timestamp
Combining NOTICE files: out/target/product/generic/obj/NOTICE.html
Target system fs image: out/target/product/generic/obj/PACKAGING/systemimage_intermediates/system.img
Install system fs image: out/target/product/generic/system.img
Installed file list: out/target/product/generic/installed-files.txt
htmlDir not a directory: out/target/common/docs/gen
DroidDoc took 413 sec. to write docs to out/target/common/docs/doc-comment-check
jiang@jiang-laptop:~$
```

Figure 1. The screenshot of a terminal where the Android source codes are successfully compiled

图 1. Android 源码编译成功的终端截图

emulator 命令。值得提一下的是，如果不知道如何设置仿真的环境变量，那么前 4 行设置仿真环境变量的命令也可用如下命令代替：

```
cd ~/work & source ./build/envsetup.sh & lunch generic-eng
```

这里，它还包含了编译环境变量的设置。仿真运行成功后，进入系统设置，选择中文语言，然后去查看系统信息，就可以找到如图 2 所示的信息。图 2 展示了一个仿真运行 Android 系统的截图。

4. 问题处理

一般来说，按以上介绍，Android 的下载编译仿真就没有问题。在整个实验过程中，如果遇到问题，我们如何处理呢？首先查看硬件是否符合要求，然后查看软件的兼容性是否有问题。硬件方面的常见问题就是硬盘空间不足或者内存不够，更换大硬盘和大内存条就行。编译出现的问题往往就是因为编译器 gcc 和 java 与 Android 源码不兼容，或者因为 make 的版本较低，或者因为选用 Linux 系统的进制与 Android 源码设置的进制不匹配。为此，在终端通过如下命令：

```
gcc -v
java -version
make -version
```

就可以分别查看 Linux 系统中安装的编译器 gcc 和 java 的版本以及 make 的版本。版本为 2.3 和 4 系列的 Android 源码要求的 java 版本为 1.6，就版本为 5 系列的 Android 源码而言，要求的 java 版本为 1.7。版本为 2.3 系列的 Android 源码要求的 gcc 版本为 4.4.3，而版本为 4 和 5 系列的 Android 源码所要求的 gcc 版本为 4.6 或 4.7。这三种版本的 Android 源码要求的 make 版本都为 3.81 或更高。依次排查，如果不符合要求，就得更换成相符的版本。



Figure 2. The screenshot of emulating the Android system

图 2. 仿真运行 Android 系统的截图

5. 小结

Android 系统用户逐年增加，这方面的研究和开发很有意义。通过 Android 源码编译，我们不仅加深了对这个系统的基本原理的认识，而且还认识了它与其他系统的兼容性。整个下载编译仿真的过程还可以从 Ubuntu Linux 系统推广到 Cynwin Linux 系统或 Mac OSX 系统，其变化无非就是设置环境变量的方式不同而已，这个工作将另文公布。我们还可以把这个过程写成一个 shell 脚本，通过这个脚本来完成整个任务。

致谢

作者感谢评审人对本文给予许多很有价值的修改建议！也感谢众多的无名网友提供有关 Android 源代码的各种各样的编译介绍！这个工作还得到国家自然科学基金(NSFC 11171356)和国家基础科学人才培养基金资助。

参考文献 (References)

- [1] Bloch, J., 著, 杨春花, 俞黎敏, 译 (2009) Effective Java 中文版. 第 2 版, 机械工业出版社, 北京.
- [2] Bollella, E.J.B.G., 著, 田思源, 译 (2010) Java 实时编程. 机械工业出版社, 北京.
- [3] Blum, R. and Bresnahan, C. (2011) Linux command line and shell scripting bible. Second Edition, Wiley Publishing, Inc., New York.
- [4] Blum, R., Bresnahan, C., 著, 武海峰, 译 (2012) Linux 命令行与 shell 脚本编程大全. 第 1 版, 人民邮电出版社, 北京.
- [5] Burnette, E. (2010) Hello, android: Introducing Google's mobile development platform. Pragmatic Bookshelf, Raleigh, North Carolina.
- [6] Burnette, E., 著, 张波, 高朝勤, 杨越, 译 (2009) Android 基础教程. 人民邮电出版社, 北京.
- [7] 杨丰盛 (2010) Android 应用开发揭秘. 机械工业出版社, 北京.
- [8] 余志龙, 陈昱勋, 郑名杰, 陈小凤, 郭秩均 (2009) Google Android SDK 开发范例大全. 人民邮电出版社, 北京.
- [9] 李宁 (2010) Android/OPhone 开发完全讲义. 中国水利水电出版社, 北京.
- [10] 韩超, 梁泉 (2010) Android 系统原理及开发要点详解. 电子工业出版社, 北京.
- [11] 靳岩, 姚尚朗 (2009) Google Android 开发入门与实战. 人民邮电出版社, 北京.
- [12] Silva, V. (2012) Pro Android Games. 2nd Edition, Apress, New York.
- [13] Silva, V., 著, 王恒, 苏金国, 译 (2011) 精通 Android 游戏开发. 人民邮电出版社, 北京.
- [14] 李宁 (2010) 人人都玩开心网: Ext JS + Android + SSH 整合开发 Web 与移动 SNS. 电子工业出版社, 北京.
- [15] Chacon, S. and Straub, B. (2014) Pro Git. Apress, New York.