## 深入浅出 - Android系统移植与平台开发（一）

一、Android4.0系统的下载与编译

Android系统的下载与编译，Google的官方网站上已经给出了详细的说明，请参照Android的官方网址：

<http://source.android.com/source/index.html>

内容主要分为：

1.        准备Android下载与编译环境

2.        下载源码

3.        编译源码及内核源码

下面我们依次来说明其步骤及注意事项。

1.      准备Android下载与编译环境

Android源码的编译Google只给出了在类Unix系统，确切的说是Ubuntu及MacOS上的详细过程，我们下面只描述Ubuntu上的编译过程。

1.1硬件环境：

l  足够快的PC机，如果有条件，最好是使用实体PC

l  内存最少1G，硬盘最少要30GB

注：也可以使用Vmware或VirtualBox等虚拟机软件，但是编译速度太慢，至于内存，如果小于1G，在编译系统时可能会出错。

1.2软件环境

我们使用Ubuntu12.04作为编译主机系统，（官方不建议使用Ubuntu12.04版本），由于Android系统里代码大部分是由Java语言写的，所以必然要安装JDK，不过，不同的版本，对JDK的版本有不同的要求：

l  在Gingerbread(2.3.x) 版本及其之后版本，使用JDK6.0，并且使用64位系统编译

l  在Gingerbread(2.3.x) 以前版本，使用JDK5.0，并且可以在32位系统上编译

注：当然我们通过修改一些配置文件，在32位系统上编译，但是不能保证编译出来的系统将来能否不出问题。

JDK6.0建议直接从SUN公司的网站上下载：

<http://java.sun.com/javase/downloads/>

安装编译时依赖的包：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ sudo apt-get install git-core gnupg flex bison gperf build-essential \
2. zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev \
3. libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \
4. libgl1-mesa-dev g++-multilib mingw32 tofrodos \
5. python-markdown libxml2-utils xsltproc zlib1g-dev:i386

执行下面的命令：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ sudo  ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1/usr/lib/i386-linux-gnu/libGL.so

2.      下载Android4.0及Goldfish源码

2.1 下载并初始化repo工具

依次执行下面的命令：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ mkdir ~/bin
2. $ PATH=~/bin:$PATH

通过curl命令下载repo工具：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ curl https://dl-ssl.google.com/dl/googlesource/git-repo/repo > ~/bin/repo

给repo工具加下执行权限：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ chmod  a+x ~/bin/repo

创建我们的源码目录：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ mkdir WORKING\_DIRECTORY
2. $ cd WORKING\_DIRECTORY

初始化repo工具，准备访问google源码仓库：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ repo  init -u  https://android.googlesource.com/platform/manifest -b android-4.0.1\_r1

有的时候使用上面的命令不能连接android服务器，我们可以使用下面的命令连接：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ repo init -u   https://android.googlesource.com/a/platform/manifest -b android-4.0.1\_r1

这个时候，要保证您的ubuntu能够上网，并且能正常连接到Android服务器，在初始化完repo工具后，会让你输入gmail邮箱及用户名验证。

2.2下载Android源码

通过执行下面的命令，开始下载Android源码：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $repo sync

如果你的PC机足够给力，可以在命令后面加上–jn，n表示多个线程同时下载，可以提高下载速度。

然后是漫长的等待，Android4.0.1的源码大概有6GB左右，一般1~2个晚上的时间。

有的时候，网络可能会断，我们可以使用下面的脚本来进行下载，当连接出错时，会自动下载。

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. #!/bin/bash
2. PATH=~/bin:$PATH
3. repo init-u https://android.googlesource.com/a/platform/manifest-b android-4.0.1\_r1
4. repo sync
5. while [ $? != 0]; do
6. echo“  \*\*Error: sync failed, re-sync again”
7. sleep 5
8. repo sync
9. done

2.3其它源码下载源

前面所述的下载步骤是Google放出一官方版本的Android代码，它里面通常只有模拟器和部分和Google有合作关系设备的源码，如果要想编译一个目标设备的Android4.0.1系统，我们肯定要对其进行定制，不过有一些知名的设备生产厂商或第三方厂商，也会开源针对自己设备的Android系统，如：三星，高通等，比较专业的，网址如下：

[www.linaro.org](http://www.linaro.org/)

linaro网站主要是ARM的芯片的最新Android解决方案，如：基于三星猎户座芯片的origen开发板，基于TI的OMAP系统芯片的PandaBoard板，基于Nova A9500 ARM 双核Cortex A9的snowball开发板等。

如果从上述网站上下载源码，要根据上述网站的说明进行相关操作。

2.4 下载模拟器Goldfish内核源码

Google在开发Android系统的同时，使用qemu开发了针对每个版本的一个模拟器，这大大降低了开发人员的开发成本，便于Android技术的推广。qemu是一个开源的模拟处理器软件，而Google使用qemu模拟的是ARM926ej-S的Goldfish处理器，如果开发人员在没有目标开发板的情况下，要对硬件进行操作的话，必然要去操作Android底层的驱动，我们可以使用模拟器的内核源码。

Ø  下载Goldfish内核源码

首先进入到Android的源码目录，创建源码目录kernel：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ cd WORKING\_DIRECTORY
2. $ mkdir kernel

从Android的远程源码仓库下载源码：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

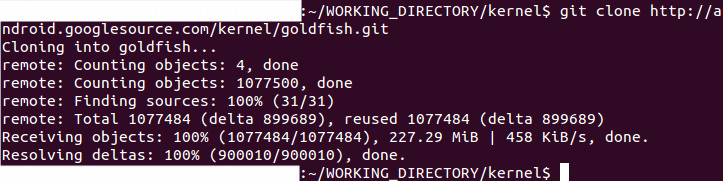
1. $git clone http://android.googlesource.com/kernel/goldfish.git

当然，如果你要编译指定目标平台的kernel的话，也可以从指定的仓库里下载，官方给出了下列处理器的kernel：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ git clone https://android.googlesource.com/kernel/common.git
2. $ git clone https://android.googlesource.com/kernel/exynos.git
3. $ git clone https://android.googlesource.com/kernel/goldfish.git
4. $ git clone https://android.googlesource.com/kernel/msm.git
5. $ git clone https://android.googlesource.com/kernel/omap.git
6. $ git clone https://android.googlesource.com/kernel/samsung.git
7. $ git clone https://android.googlesource.com/kernel/tegra.git

大概几十分钟后，下载完了，如下图所示：

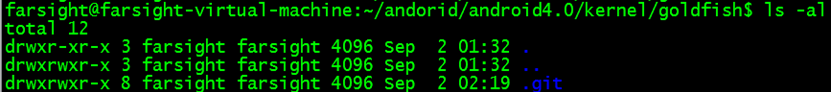


在kernel目录下产生一个goldfish目录，进入到该目录下：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ cd goldfish
2. $ ls -al

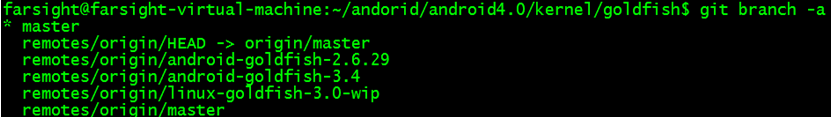
通过ls–al命令查看当前目录，可以看到隐藏的.git。



通过git branch –a来查看所以分支：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ git branch -a



通过git checkout来得到源码：

**[plain]** [view plain](http://blog.csdn.net/mr_raptor/article/details/7946254) [copy](http://blog.csdn.net/mr_raptor/article/details/7946254)

1. $ git checkout remotes/origin/android-goldfish-2.6.29

如下图所示，当前目录下出现了Linux内核源码的目录结构：

