

# Android 源码下载与编译过程

## 目录

- Android 源码下载与编译过程..... 1
- 一、目标..... 2
- 二、环境..... 2
- 三、流程..... 2
  - 1、安装 openjdk ..... 2
  - 2、安装编译时各种依赖的库..... 2
  - 3、安装 ccache..... 3
  - 4、安装 repo ..... 3
  - 5、下载源码..... 4
  - 6、编译源码..... 5
- 四、编译 SDK..... 6

# 一、目标

编译 android-5.1.1\_r38 源码，并且编译 linux 下的 sdk 和 windows 下的 sdk。

# 二、环境

Ubuntu 操作系统：ubuntu-14.04.5-desktop-amd64.iso。

磁盘空间：最好是找一台机器新装系统，磁盘空间越大越好。

Java 环境：用 openjdk7，不要用 oracle 的，会报错。

# 三、流程

## 1、安装 openjdk

执行如下命令：

```
sudo apt-get install openjdk-7-jdk
sudo update-alternatives --config java
sudo update-alternatives --config javac
```

## 2、安装编译时各种依赖的库

```
sudo apt-get install flex
sudo apt-get install bison
sudo apt-get install gperf
sudo apt-get install libsdl-dev
sudo apt-get install libesd0-dev
sudo apt-get install libwxgtk2.8-dev
sudo apt-get install build-essential
sudo apt-get install zip
sudo apt-get install curl
sudo apt-get install valgrind
sudo apt-get install git-core
sudo apt-get install gnupg
sudo apt-get install git
sudo apt-get install libc6-dev
sudo apt-get install libncurses5-dev:i386
sudo apt-get install x11proto-core-dev
```

```
sudo apt-get install libx11-dev:i386
sudo apt-get install libreadline6-dev:i386
sudo apt-get install libgl1-mesa-dev
sudo apt-get install g++-multilib
sudo apt-get install mingw32
sudo apt-get install tofrodos
sudo apt-get install python-markdown
sudo apt-get install libxml2-utils
sudo apt-get install xsltproc
sudo apt-get install zlib1g-dev:i386
sudo apt-get install dpkg-dev
```

### 3、安装 ccache

执行如下命令：

```
sudo apt-get install ccache
```

编辑`~/.bashrc` 文件，在最后加上如下文字：

```
export USE_CCACHE=1
```

保存后执行如下命令：

```
source ~/.bashrc
```

### 4、安装 repo

此时可能需要翻墙，先在`~/`目录下执行如下命令：

```
mkdir bin
```

建立一个文件夹来保存 repo 文件。

再执行如下命令来下载（第一条不行用第二条，都不行就自己去网上搜吧）：

```
curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
```

或

```
curl http://commodatastorage.googleapis.com/git-repo-downloads/repo >
~/bin/repo
```

下好之后修改 repo 文件里面的 REPO\_URL 的值改为:

```
'https://android.googlesource.com/tools/repo'
```

保存后执行如下命令给 repo 文件增加可执行权限:

```
chmod a+x ~/bin/repo
```

## 5、下载源码

先在~/目录下创建一个文件夹来保存源码, 执行:

```
mkdir Android
```

以后可能下不同版本的源码, 所以在~/Android 目录下再创建一个要编译的版本的文件夹, 本文档的目标是编译 android-5.1.1\_r38, 所以再执行:

```
cd Android  
mkdir android-5.1.1_r38
```

创建好文件夹之后, 先到此文件夹下:

```
cd android-5.1.1_r38
```

再执行如下命令来初始化:

```
repo init -u https://android.googlesource.com/platform/manifest  
-b android-5.1.1_r38
```

初始化好之后, 就可以下源码了, 由于要翻墙, 可能网络不稳定, 所以写一个 shell 脚本 (sync.sh) 来执行, 避免意外中断, 先创建一个 shell 文件:

```
gedit ~/sync.sh  
输入如下代码:  
#!/bin/bash  
echo "=====start repo sync====="  
repo sync  
while [ $? == 1 ];do  
    echo "=====sync failed,re-sync again====="  
    sleep 3  
    repo sync  
done  
echo "=====sync succeed !====="
```

保存后退出，赋予此文件可执行权限：

```
chmod a+x ~/sync.sh
```

之后执行此文件：

```
~/sync.sh
```

执行之后，就可以等着了，因为代码非常多，而且翻墙，所以网络很不稳定。

## 6、编译源码

下载好之后，源代码就会在~/Android/android-5.1.1\_r38 目录下，此时进入这个目录：

```
cd ~/Android/android-5.1.1_r38
```

执行如下命令来对 cache 做设置，“-M” 的含义是设置 cache 大小：

```
prebuilts/misc/linux-x86/ccache/ccache -M 50G
```

执行如下命令来初始化编译环境：

```
source build/envsetup.sh
```

之后输入如下命令来选择一个产品：

```
Lunch
```

输入编号即可，具体的这些产品的含义可以参考 Android Build 系统，这些产品有不同架构平台，也包括 sdk。

选择好之后，输入如下命令执行编译：

```
make -jN
```

其中 N 是 CPU 核心数×2+2，例如 “make -j4”。

## 四、编译 SDK

编译 sdk 就是在执行 “lunch” 命令的时候选择 sdk 即可，或者直接输入如下命令：

**lunch sdk-eng**

因为有不同平台和 32 位、64 位之分，所以也有不同的 sdk，基本是如下几个：

**sdk-eng**

**sdk\_x86-eng**

**sdk\_mips-eng**

**sdk\_arm64-eng**

**sdk\_x86\_64-eng**

选好之后执行如下命令即可：

**make PRODUCT-sdk-sdk -j4**

或者执行如下命令来编译 windows 下的 sdk：

**make PRODUCT-sdk-win\_sdk -j4**