[**[Android 编译(一)] Ubuntu 16.04 LTS 成功编译 Android 6.0 源码教程**](http://blog.csdn.net/fuchaosz/article/details/51487585)

标签： [android](http://www.csdn.net/tag/android)[源码](http://www.csdn.net/tag/%e6%ba%90%e7%a0%81)[ubuntu](http://www.csdn.net/tag/ubuntu)

2016-05-24 09:33 38251人阅读 [评论](http://blog.csdn.net/fuchaosz/article/details/51487585/#comments)(68) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/fuchaosz/article/details/51487585/#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

android（22） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg 系统源码（1） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

目录[(?)](http://blog.csdn.net/fuchaosz/article/details/51487585/" \o "系统根据文章中H1到H6标签自动生成文章目录)[[+]](http://blog.csdn.net/fuchaosz/article/details/51487585/)

1 前言

经过3天奋战，终于在Ubuntu 16.04上把Android 6.0的源码编译出来了，各种配置，各种error，各种爬坑，特写此博客记录爬坑经历。先上图，Ubuntu上编译完后成功运行模拟器，如图：



2 编译环境

* UbuntuKylin 16.04 LTS
* Android 6.0\_r1
* Open JDK 7

3 准备工作

(1) 下载Android 6.0源码.

Androdi 6.0源码下载地址: <http://pan.baidu.com/s/1o6N86a2>

感谢下面这位博主上传源码，我也是从他的博客下载的源码：

<http://blog.csdn.net/ilittleone/article/details/6823441>

下载源码请在Windows下面下载，迅雷不解释。   
**注意：源码的存放路径不能有中文，否则编译到系统内置的apk的时候会报错无法解析class.dex文件**

(2) 解压源码,合并文件。   
下载完毕后，使用下面命令合并：

cat Android6\_r1\_\*>Android6\_r1.tgz

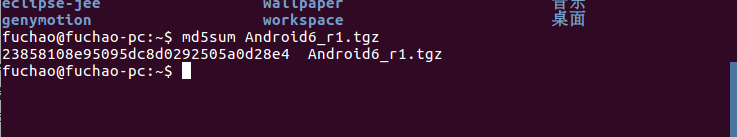
* 1

合并完后，使用下面命令校验hash:

md5sum Android6\_r1.tgz

* 1

此时会生成一个hash串，如图：



对比你下载的文件里面的hash.txt，如果一致那么继续，不一致则文件损坏，重新下载。

(3) 解压源码。

tar -zxvf Android6\_r1.tgz

解压源码是个漫长的过程，解压后会在当前目录下生成一个mydroid的文件夹。**如果你发现源码的路径有中文，需要移动源码，请复制压缩包，然后再使用命令解压，而不要直接复制解压后的文件夹**。

(4) 安装Ubuntu 16.04   
编译Android源码最好直接安装Linux，不要使用虚拟机，编译源码是个漫长的过程，虚拟机要十几个小时，博主直接在电脑上装Ubuntu，编译只需要3个小时。实体机可避免虚拟机的很多问题，例如：磁盘空间不足，内存不够等等。更重要的是，编译源码是个不断爬坑的过程，实体机可以很快的遇到错误，然后解决问题，继续编译，再次遇到错误，而虚拟机这个周期会很漫长，因而浪费大量时间。实体机解压速度也比虚拟机快得多。Ubuntu与windows兼容做的很好，可以直接把Ubuntu当作一个windows程序安装，也可以单独安装，如果单独安装请务必给/home分区分配至少120G空间，因为编译后的Android系统真的很大。   
**强烈建议下载Ubuntu镜像，然后使用软碟通刻录到U盘，通过U盘把Ubuntu安装到物理机器上。**

(5) Ubuntu安装教程参考下面链接,Ubuntu 16.04安装方法也是一样的 ：

<http://jingyan.baidu.com/article/59703552e0a6e18fc007409f.html>

(6) 再次声明：转载请注明来自”梧桐那时雨”的博客:<http://blog.csdn.net/fuchaosz/article/details/51487585>

4 更换软件源为阿里云

为了更快的安装软件，我们需要更换软件源为国内的软件源，这里推荐使用Ubuntu官方指定的国内软件源阿里云，博主之前使用的是网易的软件源，结果发现OpenJDK7装不上去，于是换成了阿里的，具体方法是：

sudo gedit /etc/apt/sources.list

* 1

在文件最前面加入下面代码：

deb http://mirrors.aliyun.com/ubuntu/ quantal main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ quantal-security main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ quantal-updates main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ quantal-proposed main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ quantal-backports main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ quantal main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ quantal-security main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ quantal-updates main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ quantal-proposed main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ quantal-backports main restricted universe multiverse

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

5 安装依赖

sudo apt-get install -y git flex bison gperf build-essential libncurses5-dev:i386

sudo apt-get install libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-dev g++-multilib

sudo apt-get install tofrodos python-markdown libxml2-utils xsltproc zlib1g-dev:i386

sudo apt-get install dpkg-dev libsdl1.2-dev libesd0-dev

sudo apt-get install git-core gnupg flex bison gperf build-essential

sudo apt-get install zip curl zlib1g-dev gcc-multilib g++-multilib

sudo apt-get install libc6-dev-i386

sudo apt-get install lib32ncurses5-dev x11proto-core-dev libx11-dev

sudo apt-get install lib32z-dev ccache

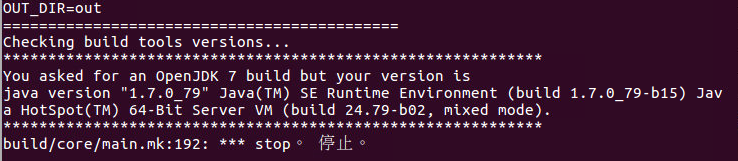
sudo apt-get install libgl1-mesa-dev libxml2-utils xsltproc unzip m4

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

**Ubuntu 16.04请务必使用上面的依赖**，Ubuntu 16.04需要的依赖和Ubuntu 14.04所需要的依赖是不同的，博主照着Ubuntu 14.04的依赖结果掉进大坑，详见后面的编译过程记录.

6 安装open JDK7

**从Android 6.0开始Android源码的编译需要安装OpenJDK,不能使用Oracle JDK,而且Android 6.0只能使用OpenJDK7**,如下图：



由于Ubuntu 16.04没有OpenJDK7的源，因此在16.04上安装OpenJDK7需要执行下面的命令：

sudo add-apt-repository ppa:openjdk-r/ppa

sudo apt-get update

sudo apt-get install openjdk-7-jdk

* 1
* 2
* 3

配置OpenJDK.打开/etc/profile文件:

sudo gedit /etc/profile

* 1

在末尾追加下面代码：

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH=${JAVA\_HOME}/bin:$PATH

* 1
* 2
* 3
* 4

修改了/etc/profile文件需要重启才能生效，但使用下面命令可以在不重启的情况下在当前bash环境生效：

source /etc/profile

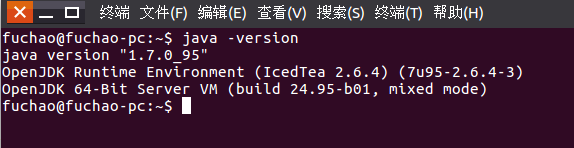
* 1

检查OpenJDK配置是否正确:

java -version

* 1

配置正确则如下图:



7 修改源码

修改 mydroid/art/build/Android.common\_build.mk 文件，定位到75行，将下面的代码：

ifneq ($(WITHOUT\_HOST\_CLANG),true)

* 1

改为：

ifeq ($(WITHOUT\_HOST\_CLANG),false)

* 1

如果不修改这里，会遇到一个比较棘手的编译错误，修改的目的是把CLANG这个编译选项关掉，详见后面的编译记录，百度搜狗都无解，这个错误只会在Ubuntu16.04上遇到，Ubuntu14.04则不存在这个问题，解决这个编译错误花费一天时间，还好咱英语好，在歪果人的论坛上看了几篇帖子，终于搞定了。这里顺便说一下小技巧：   
**遇到编译错误，请把error后面的错误信息全部复制，然后用搜狗搜英文网页，不要用百度那个渣渣**

8 开始编译

(1) 在 .bashrc文件末尾添加：export USE\_CCACHE = 1

echo export USE\_CCACHE=1 >> ~/.bashrc

* 1

(2) 为了提高编译效率，设置编译器高速缓存:

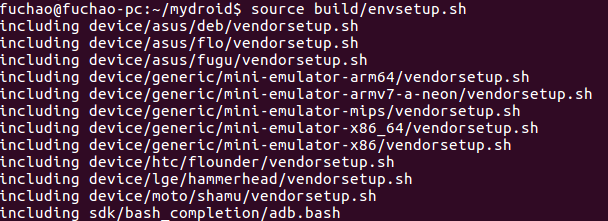
prebuilts/misc/linux-x86/ccache/ccache -M 50G

* 1

(3) 接着导入编译Android源码所需的环境变量和其它参数:

source build/envsetup.sh

* 1

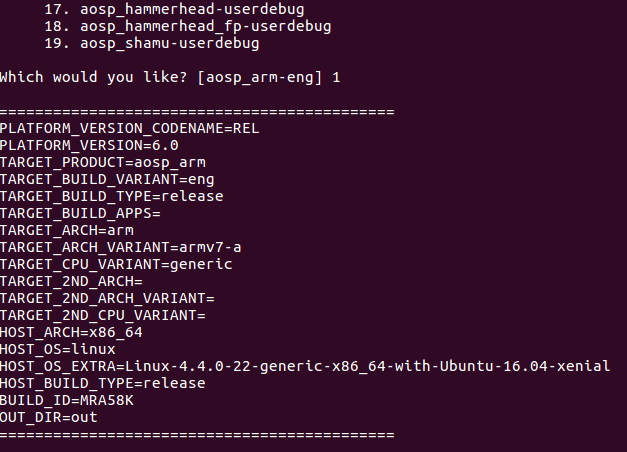


(4) 运行lunch命令选择编译目标:

lunch

* 1



选择1：   


(5) 开始编译，执行make -j8, 8为同时编译的线程数，一般google推荐这个数字为2倍的cpu个数再加上2，比如4核，就是10。博主使用的是8:

make -j8

* 1

关于cpu个数，可以用下面命令查看:

cat /proc/cpuinfo

* 1

好了，开始编译了，接下来就是不断error，不断修复，继续编译的过程,下面记录我的爬坑经历

9 编译记录

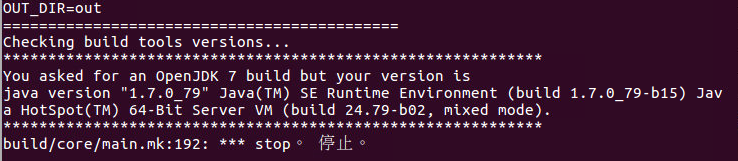
我是参照下面这篇文章开始编译的：

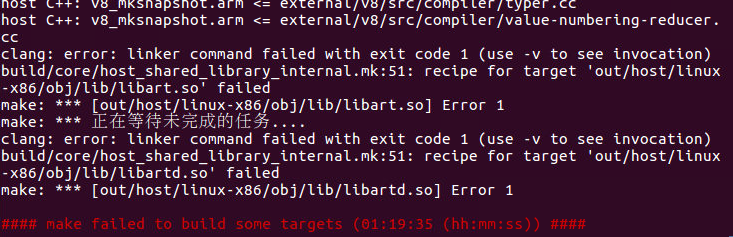
<http://www.linuxidc.com/Linux/2016-01/127292p2.htm>

如果一帆风顺，我也就用不着写这篇博客了，然而他用的是Ubuntu14.04，我用的最新的Ubuntu16.04，按照上面这篇博文，遇到很多坑，这里记录以下爬坑经历：   
(1) 依赖不同。Ubuntu 14和16的依赖选项是不同的：   


于是将libsdl1.2-dev改为libsdl-dev,将mingw32去掉

(2) JDK配置错误。   
先用Oracle JDK8报下面错：   


改为Oracle JDK7报下面错：   
   
后改为OpenJDK7解决问题。

(3) 大坑。接着遇到下面错误:   


仔细看错误：

clang: error: linker command failed with exit code

* 1

其实这个是系统使用clang编辑器报的错，百度了一下，解决方法一般是将art/build/Android.common\_build.mk文件ifneq ($(WITHOUT\_HOST\_CLANG),true)改为ifeq ($(WITHOUT\_HOST\_CLANG),false)   
然而纸上得来终觉浅，绝知此事要躬行，因为，这并没有什么卵用，至少现在没有卵用。不信，请看下面的帖子：

<http://ask.csdn.net/questions/253318>

我试了一下，也没用，还是没解决问题，好像这个是新系统的兼容问题，百度搜狗一天，没有进展，后来发现了下面这篇帖子,也是14.04编译Android6.0的：

<http://blog.csdn.net/lewif/article/details/49762993>

抱着反正无聊试一试的想法，把上面这篇帖子中的依赖一通复制(所以最终用的依赖是两篇帖子加起来的依赖)：

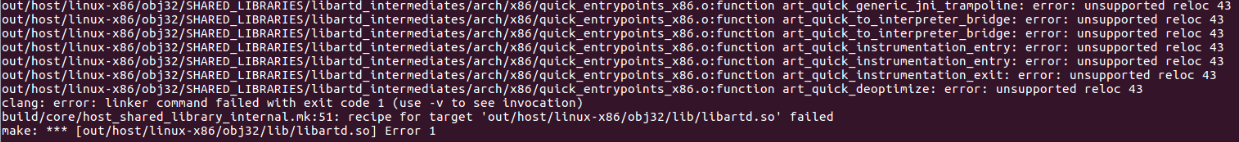
sudo apt-get install git-core gnupg flex bison gperf build-essential \

zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \

lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev ccache \

libgl1-mesa-dev libxml2-utils xsltproc unzip m4

* 1
* 2
* 3
* 4

然后make clean，重新再来，接着就睡觉去了。   
一觉醒来，然后……..你以为我就编粗来了？呵呵，这个错误是终于编过去了，可是又遇到下面的错误：   


解决这个错误，又花费一番力气，在歪果人的论坛上看到下面这篇帖子：

<http://forum.xda-developers.com/chef-central/android/guide-how-to-setup-ubuntu-16-04-lts-t3363669/page2>

注意看这个帖子的15楼，哈哈，这个时候找到解决方法了,再修改art / build/Android.common\_build.mk文件

ifneq ($(WITHOUT\_HOST\_CLANG),true)

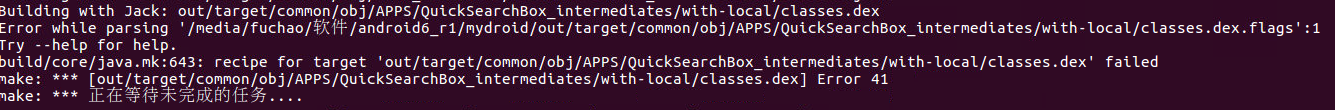
* 1

改为：

ifeq ($(WITHOUT\_HOST\_CLANG),false)

* 1

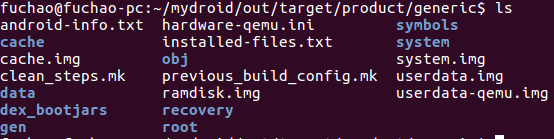
重新编译，解决问题。

(4) 中文路径，坑。   


遇到这个问题也是很难百度到，哎，最后把源码换到了全英文的路径下面，解决这个问题。注意移动源码的时候不要直接复制mydroid文件夹，一个小时都复制不完，所以正确的方法是复制压缩包，然后再解压一遍。   
接下来就编译成功了。

10 编译成功

编译成功后如下图：   


进入mydroid/out/target/product/generic目录，可以看到编译好的文件：   


启动模拟器：

emulator

* 1

稍等一会，就可以出现开头的模拟器画面了，哈哈，打完收工。

11 附录：Ubuntu 16.04编译ROM英文教程

最后，附上在看帖子的时候发现的，在Ubuntu16.04上编译AOSP、CyanogenMod的教程，当然是英文的啦：

<http://forum.xda-developers.com/chef-central/android/guide-how-to-setup-ubuntu-16-04-lts-t3363669>

12 总结

Android系统源码的编译就是不断和各种error做斗争的过程，考验程序员获取信息的能力，遇到error的时候只能各种百度搜狗，但这也是非常锻炼能力的过程，博主在编译过程中最大的体会就是：英语一定要好，哈哈。如果你成功编译出来了，或者编译遇到问题，欢迎给我留言。

13 转载请注明来自”梧桐那时雨”的博客:<http://blog.csdn.net/fuchaosz/article/details/51487585>

Tips   
如果觉得这篇博客对你有帮助或者喜欢博主的写作风格，就给博主留个言或者顶一下呗，鼓励博主创作出更多优质博客，Thank you.