




Android 4.4 成功编译运行，为下一步修改作准备

原创

helloDesword  于 2014-11-13 23:19:04 发布  1560  收藏

分类专栏: [Android Related](#) 文章标签: [android](#) [操作系统](#) [linux](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/ls1160/article/details/41089243>

版权



[Android Related](#) 专栏收录该内容

10 篇文章 0 订阅

订阅专栏

经过多天断断续续的努力, 今天下午终于成功编译的Android 的系统。

由于限制的关系, repo之类的方法已经不适用了, 完全无法下载源代码。

因此借鉴了这位朋友的, 4.4 编译ok, 2.3.6的不行。再次非常感谢 @橘小兔-一米阳光 的贡献。

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

一会我也将自己压缩的zip上传, 方便在windows下面直接查看源代码。

md5 (7b8f155d04089c66a9aa6f4d39b079f9)

编译过程遇到不少问题, 开始的时候, 不明确项目需求。想着来编译android 修改kernel, 于是参考了看雪论坛的@JoenChen。再次感谢

能够成功编译的, 但是后面没有继续操作下去了, 需要利用eclipse进行调试的设置等等。

后来发现仅仅是在framework层对网络相关的代码进行修改的。于是开始编译android 的代码。

开始的时候在虚拟中运行, 编译。也是曲折不少。

首先在实验室运行, 结果实验室的硬盘出现问题了, 或者有人关掉了、或者线程数设置太多, 直接出现线程错误。

--错误--

```
collect2: ld terminated with signal 9 [Killed]
```

```
make: *** [out/host/linux-x86/obj/EXECUTABLES/clang_intermediates/clang] Error 1
```

```
make: *** Waiting for unfinished jobs....
```

--说是swap区的问题。其实应是线程数太多, 内存不够了。

接着在自己的电脑的虚拟机上运行, 运行了五个小时, 突然告诉我, 空间不足, 我分配了总共40G的, 也不够。真是纠结。

最终一咬牙, 直接装双系统。感觉爽爽的。然后编译5个小时, ok了。运行emulator, ok了。指定kernel运行, 崩掉了。 然后目前错误还没有解决。

恩。留着之后再解决吧。目前应该不设计kernel的修改, 只是部分模块的修改。届时修改部分模块即可。

心路历程之后，开始正题。

1、安装java环境：

jdk-6u33-linux-x64.bin , to be installed.

[设置环境变量]

```
set /etc/profile
```

```
export JAVA_HOME=/home/desword/jdk1.6.0_33
```

```
export PATH=$JAVA_HOME/bin:$PATH
```

```
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

[环境变量生效]

```
source /etc/profile
```

2、操作系统ubuntu 12.04 LTS 64位。 Android4.4

[合并] @橘小兔-一米阳光 同学的android代码。

```
cat droidSplit*>droid.tar.gz
```

[对应的md5]

```
thinkinbunny@thinkinbunny-Vostro-270:~$ md5sum droid.tar.gz
```

```
2fcd78deb4582187bb7c9665d67484cc droid.tar.gz
```

```
thinkinbunny@thinkinbunny-Vostro-270:~$ md5sum droidSplit*
```

```
ecce8ccdc0d99958b8864d6f86275e8b droidSplitaa
```

```
7d761fa4e53ca04de256767a270c3ffb droidSplitab
```

```
9bb0b8ce34a77f082f6700211e08dd45 droidSplitac
```

```
8b285586c105d689a00fd8c77081fad7 droidSplitad
```

```
3db2b09e307d8e4fa497be68e9745c7a droidSplitae
```

```
3796622283244fe41447289bdba4172a droidSplitaf
```

3、配置对应的编译环境：[参考[百度知道](#)]

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

```
curl zlib1g-dev libc6-dev lib32ncurses5-dev ia32-libs x11proto-core-dev
```

```
libx11-dev libreadline-gplv2-dev lib32z-dev
```

```
libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown
```

```
libxml2-utils xsltproc
```

如果有一些包存在问题，对应按照推荐的修改好了。参考百度知道的，基本这些包就可以解决问题了。

4、编译初始化：

在终端中执行：

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

或：

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

执行后输出：

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

```
including device/asus/grouper/vendorsetup.sh
```

```
including device/asus/tilapia/vendorsetup.sh
```

```
including device/generic/armv7-a-neon/vendorsetup.sh
```

```
including device/generic/armv7-a/vendorsetup.sh
```

```
including device/generic/mips/vendorsetup.sh
```

```
including device/generic/x86/vendorsetup.sh
```

```
including device/samsung/maguro/vendorsetup.sh
```

```
including device/samsung/manta/vendorsetup.sh
```

```
including device/samsung/toroplus/vendorsetup.sh
```

```
including device/samsung/toro/vendorsetup.sh
```

```
including device/ti/panda/vendorsetup.sh
```

```
including sdk/bash_completion/adb.bash
```

5、编译选中目标：

在终端中执行下面的命令：

```
lunch
```

之后选择你要编译的设备，选择后，会初始化编译的项目的相关文件。

（一般选择第一个）

在终端中执行：`make -j4`

`-j4`：表示用四个线程来编译

编译根据不同机器的配置不同所需时间也有很大差异。

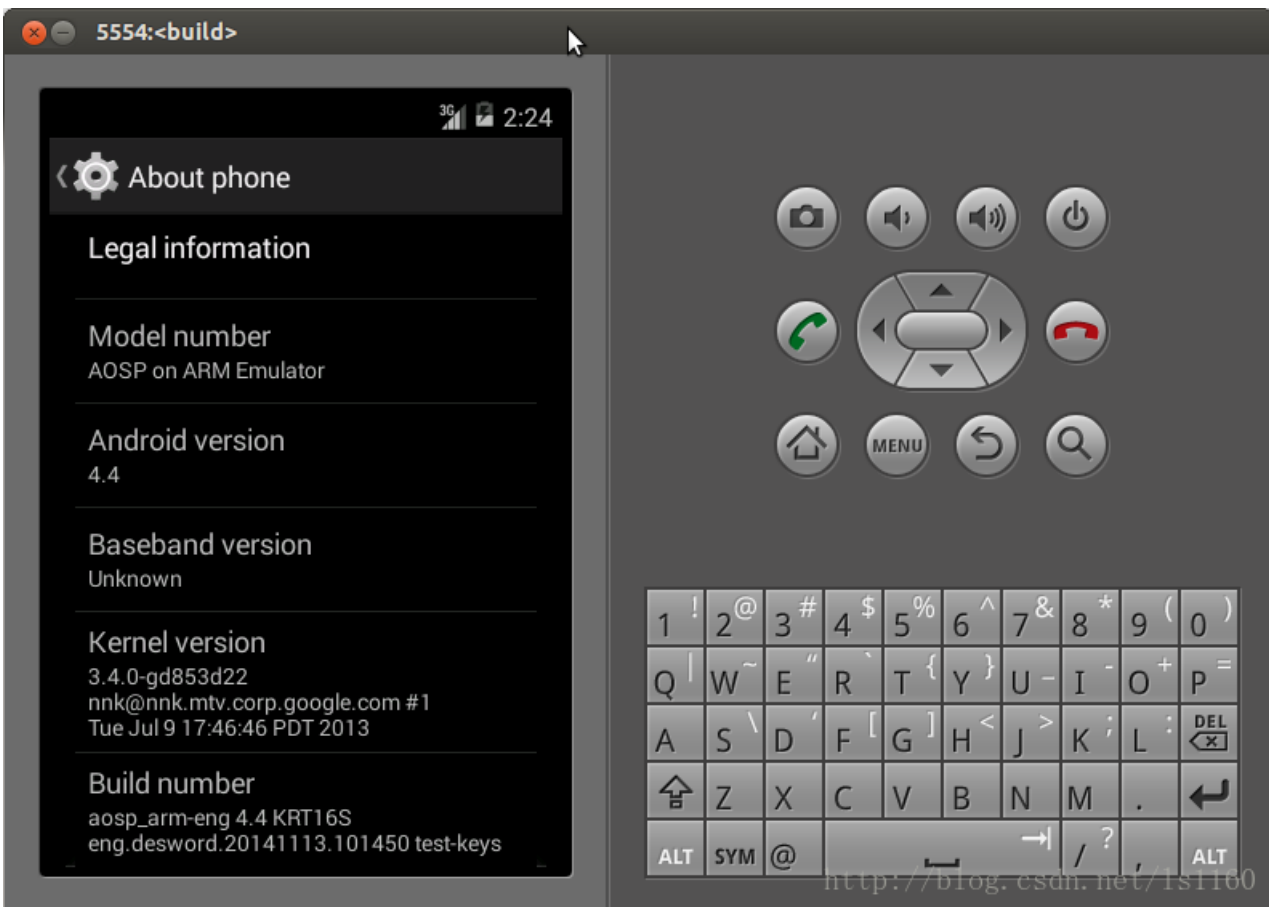
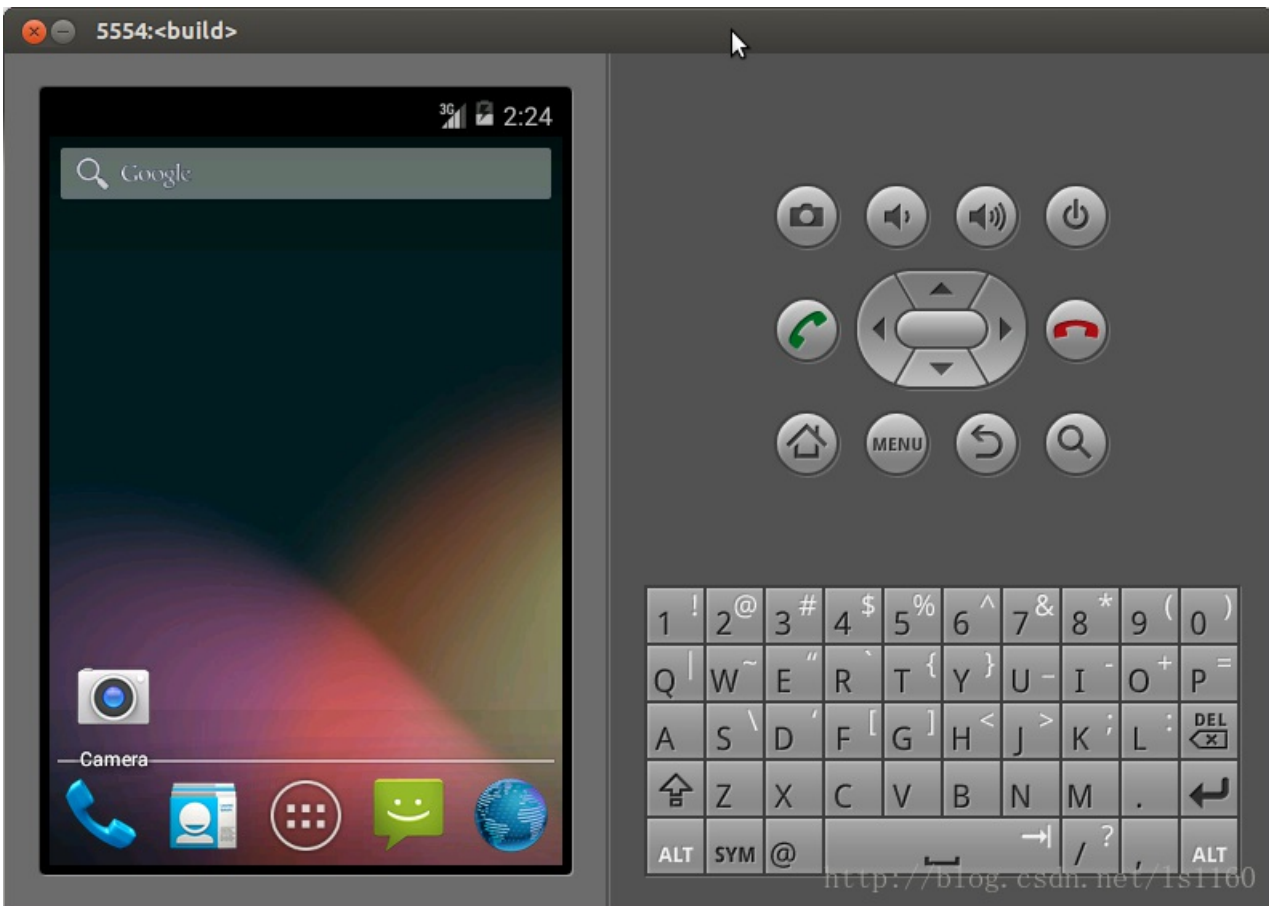
我编译了5个多小时以后，ok了。【intel i5, 2.5Ghz, 4G内存, `make -j2`】

6、运行虚拟机测试：

```
desword@desword-Inspiron-5520:~/mydroid$ emulator
```

7、另外`emulator`在目录`./out/host/...`(不记得了，往下找找)，在环境变量中设置，可以避免每次编译了再`emulator`。

ok, it's work!



00-我的其他测试：（指定kernel）

```
desword@desword-Inspiron-5520:~/mydroid$ emulator -kernel ./prebuilts/qemu-kernel/arm/kernel-qemu-armv7
-sysdir ./out/target/product/generic -system system.img -data userdata.img -ramdisk ramdisk.img
```

[qemu: could not load ramdisk]--[all is didn't work]

```
emulator -kernel ./prebuilts/qemu-kernel/arm/kernel-qemu-armv7 -sysdir ./out/target/product/generic -system
system.img -data userdata.img
```

(

1.delete the [-ramdisk ramdisk.img] order; --[black screen]

2.check the dir [./prebuilts/qemu-kernel/arm/kernel-qemu-armv7] is valibale--[stay on 'android']

3. in [out/target/product/generic]

```
desword@desword-Inspiron-5520:~/mydroid/out/target/product/generic$ chmod -R * 777[not work]
```

后续执行时使用全路径名称)

[11-17]

网上更新的一种解决方法，不适用prebuilts里面的emulator，使用adt中的emulator。

参考网址：<http://ieroot.com/2014/07/09/1674.html>

01-获取kernel的记录。

按照看雪论坛获取kernel之后，获取kernel的命令：

```
cd goldfish
```

```
ls -la
```

```
git branch -a
```

```
git checkout remoutes/origin/android-goldfish-2.6.29
```

android4.4 源代码下载：

链接：<http://pan.baidu.com/s/1sj8t1et> 密码：4qcf