
@author : guohow & bpok

@date : 02.06.13

10年年底接触 android，到现在时间也不短了，从小白开始研究，完全是出于好奇，马上就要奔考研而去，写个几年来的技术总结，给大家做个参考。

【入门】

刚开始接触 android，并不了解任何编程知识，一些东西理解起来相当费劲，仅仅是记住怎么做，也没什么意思，建议大家不要仅仅满足于如何去做，要明白为什么要那么做。

* 作为一名合格的 android 开发者需要掌握的：

C++

JAVA

LINUX

* 深入研究还需要：

C

ANDROID（侧重于一些机制的理解，如果你有 JAVA 基础，一看就会）

Experience（当然就是经验了）

* 移植 SMALI 还需要（例如 MIUI 移植，sense 移植）：

Smali 代码阅读

Android 虚拟机指令

当然了，如果你掌握了上述全部知识，你就可以逆袭了。

【win & linux】

工欲善其事必先利其器，即使啥都不会，也要明白需要做什么

1，Windows 下需要安装：

必备：

* JDK

去百度找教程

* 文本编辑器

推荐 editplus，百度如何安装，极其简单

* apktool

百度如何下载，并添加到环境变量中

* android kitchen

大名鼎鼎的厨房，集成了很多工具

可选：

Eclipse (android apk 的编写，也可以导入 android 源代码编辑)

百度下载 eclipse 或者直接下载 android 官网提供的最新版 ADT

Github

用于同步、管理以及下载源代码，github.com 告诉你如何去下载

2, 安装 linux

Linux 下推荐 ubuntu，推荐虚拟机下安装：

安装：

下载 Vmware 最新版本是 9，安装好，填好注册码

下载最新的 ubuntu 13.04 推荐 X64

虚拟机下安装好 ubuntu，以及 VM tools (用于 WIN 和 ubuntu 之间的文件交换等)

使用：

第一次使用难免陌生，时间长了自然就会了，你大概需要掌握的一些技巧：

1, apt-get 的使用

2, terminal 终端 & shell 的使用

3, vim 编辑器的使用 (可以用更加简单的 gedit 代替)

4, 一些其他的常用程序以及指令，例如 sudo su 等

Ubuntu 初始化：

环 境 配 置 我 详 细 的 写 出 来 ， 不 用 大 家 去 百 度 了 ：

打开 terminal, # 开头的字行表示指令：

(以下来自格诺的安卓集装箱：<http://www.iamgeno.com/archives/543.html>)

配置之前最好设置一下软件源，选择较快的软件源，比如 163

参考 Android 官方文档：<http://source.android.com/source/initializing.html> 配置依赖

安装 SUN JDK 1.6

1.<sudo gedit /etc/apt/sources.list> 打 开 资 源 链 表

2.<deb http://us.archive.ubuntu.com/ubuntu/ hardy multiverse> 添 加 源

3.<sudo apt-get update>更新

4.<sudo apt-get install sun-java6-jdk>安装

安装其他依赖

```
$ sudo apt-get install git gnupg flex bison gperf build-essential \
zip curl libncurses5-dev:i386 x11proto-core-dev \
libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \
libgl1-mesa-dev g++-multilib mingw32 tofrodos \
python-markdown libxml2-utils xsltproc zlib1g-dev:i386
$ sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so
```

安装 Repo 和 GIT

```
mkdir ~/bin
curl https://dl-ssl.google.com/dl/googlesource/git-repo/repo > ~/bin/repo
chmod a+x ~/bin/repo
sudo apt-get install -y git
```

再安装 CM 需要的依赖和一些有用的小工具

```
$ sudo apt-get install -y wget schedtool pngcrush
$ sudo apt-get install -y nautilus-open-terminal unrar \
meld ghex gtk-recordmydesktop gitk
```

* 同步需要的 android 代码:

主要有

aosp:google 发布的原生 android 代码

Aokp: kang project 类似于 CM 一样

Cyanogenmod: 你懂得

Miui: 来自中国的小米社区

如何同步网上教程满天飞

【反编译】

以 ubuntu 下操作为例:

拆解：

```
# apktool d helloworld.apk
```

生成的各个文件夹，你需要慢慢了解其作用，建议阅读以下 android 入门的书籍，你会明白许多。

一般如果涉及到简单的 framework 等的反编译需要修改的：

Res 资源 例如 apk 汉化，修改背景图片

Xml 设置，互联等：例如设置菜单的添加、编辑

Layout 布局 例如修改排版，颜色使用，字体

android/Manifest.xml 注册权限、整个 app 主题、添加新布局等

合并：

```
apktool b helloworld
```

Dist 下生成了新的 apk，把你修改过的类目提取出来覆盖原来的 apk 中，并签名。

【关于 ROM】

* ROM 下各个文件夹作用相信你都了解了，如果不了解：

App 程序

Lib 共享库，以及厂商的底层库

Etc 配置文件

Bin 可执行文件

Fonts 字体

Media 媒体，例如声音、开机动画等

Usr 用户、单机配置

Preload 一般是厂商预装的 apk 等

Persist 请忽略该分区的修改

Build.prop 系统参数配置

* 精简：

所谓精简其实就是删除 app 下不需要的 apk 以及对应 lib 下的.so

为什么还要删除.so？

如果某些 apk 用到了一些库支持（如 QQ），需要 NDK 自己编译，因为系统并没有提供相应的 so，如果这些 apk 被厂商内置了，那么 lib 下一定有它所需要的 so，删除这些 apk 之前记得删除他们。

* 美化

基础美化极其简单，了解以下 apk 构造，修改、替换资源就可以

* 优化：

没什么好优化的，广泛使用的 zipa 优化也只是一点皮毛，因此没必要做什么所谓的优化。

* ROOT：

推荐内核 ROOT

用厨房操作：

ROOT 自己的 ROM 即可，如果要开启 ADB 支持，建议检查 boot.img，
解压 boot 打开 default 修改 allow_debug=1

* updater-script

里面各种操作简单的概括就是格式化各个分区，解压复制文件到各个分区，设置权限等，指令只是给 recovery 系统。

可以看看网上的指导，极其简单

【adb】

Adb 是最常用的利器，常用指令

adb devices

看看那些设备连接了

adb reboot

无参：重启

recovery：重启到 rec

-bootloader：重启到 bootloader

【fastboot】

Fastboot 仅限 fastboot 模式下操作：

Fastboot flash 【分区】 【镜像文件】

刷入编译好的 img 文件

-----以上涉及的指令，务必熟练使用-----

【recovery】

侧重于 c++知识

移植、汉化、修改 recovery 可以看我之前的文章：

https://github.com/HyperToxic/CWM_REC_6.0.3.2_CHN

【cyanogenmod 移植】

这个不是一般人做的来的，即使你掌握了以上要求全部编程知识，也需要慢慢学习。

当然了，用别人写好的 device，原生 cm 代码去编译还是极其简单的：

网上教程也是一大片，不过不建议你去做这些，没意义。

常用指令：

source build/envsetup.sh 添加环境变量

make

make otatools 编译 mkboot 等工具，建议完成后=复制出来，添加到环境变量

brunch desire 生成 zip

make clean 清除所有（建议定期操作）

【cyanogenmod 修改】

可以添加你自己喜欢的 app，设置，融合 aokp 的 rom control 等

首先你得学习 android 入门提高

修改的文件：

Framework/base 下

Packages/apps 下

各个 android.mk

Vendor/cm/config

【miui 的移植】

很不幸的是之前移植过程中许多错误并没有及时的记录下来，仅仅写了一些前期的初始化：

<https://github.com/HyperToxic/myDocuments>

印象深刻的是关于 framework 的拆解，如果 framework 类数目太多，编译器是不允许编译通过的，因此你需要将 framework 拆解出来，放到 framework2 下面，还需要修改 makefile，通知编译器如何去拆解。

【Smali 的注入】

我的做法是解析一个自己写的 apk，这个 apk 的各个部分我都了如指掌，因此解析出 smali 之后对照虚拟机指令指导看起来效率就很高了。

建议去百度以下大神的经验，关键是要理解各个虚拟机的指令，移植起来就得心应手了，如果你没有这些基础，就不要浪费精力了。

【建议】

起点低，飞得高，不要羡慕那些所谓大神，他们会的东西也就那么点，我们完全可以学会。

好吧，差不多就是这些，三年下来就剩这么几个不值钱的字，但是其中收获很多，凡是只要你愿意去做总会有收获。

我的邮箱：

HyperToxicG@gmail.com

微博：

@guohow

github:

<https://github.com/HyperToxic>
