Android编译流程

前言

makefile不能满足安卓系统，性能不够，所有在makefile机制上面，增加了android mk和bp的方式，优化兼容makefile，soong就出现了。

一、编译流程

1、初始化编译环境

进入安卓根目录，执行命令source build/envsetup.sh即可初始化编译环境。

从source命令得出，envsetup.sh脚本是加载环境变量到当前终端。加载完后，即可在终端上执行脚本命令，只在当前终端生效。

2、选择编译平台

在安卓根目录，选择执行lunch命令来选择当前编译平台，可直接上参数，也可以通过菜单进行选择。

3、开始编译

执行make或者mm等命令进行编译，流程有些许不一样，make执行当前makefile脚本，但linux没有mm或者mma的，soong机制

二、soong工具

随着安卓工程越来越大，包含的module越来越多，以makefile组织编译时间越来越久，谷歌在之后引入ninja取代makefile，但现有项目还是由makefile组织，谷歌因此引入kati（soong）将makefile，android.mk翻译成ninja。

7.0 → Kati soong

8.0→ android.bp 9.0 all android.bp

1、工作原理

在/andrioid/build/soong/路径，使用soong\_ui.bash进行编译

。android.mk转换成ninja文件：bash将指定目录的mk和makefile文件转换成out/buile-<product\_name>.ninja

。android.bp转换成ninja文件：bash将指定的bp转换成out/soong/build.ninja文件

。组合ninja文件：bash会生成一个小的out/combined-<project\_name>.ninja文件，负责二者组合起来作为执行入口。

soong会根据ninja控制源码编译

kati是由golang和c++开发的，主要作用为转换

bp文件，不能分支、循环等控制，不能作核心运算。如需进行控制只能由go语言编写。

soong负责提供bp语义解析，可用于mk转bp文件但是仅限没有分支、循环流控制的mk文件

blueprint: 用于生成，解析bp的工具，解析文件格式。

三、make流程

根目录输入make命令，将树结构编译整个工程及子目录

make→ lunch → mk

1、编译开端main.mk

mm是执行当前目录的mk或者bp指定内容，在目录下执行make本质还是执行makefile，bp内容空，makefile直接指向了build/make/core/main.mk文件。

。1导入config.mk进行环境变量或重要参数配置CLEAR\_VARS\BUILD\_PACKAGE

。2导入build/make/core/definitions.mk定义了一些其他变量，my-dir、all-subdir-makefiles、all-subdir-java-files

。3定义规则，目标生成 make vendorimage make xximage

2、编译配置config.mk

config.mk定义许多变量，这些变量定义对应的mk以及逻辑功能，例：CLEAR\_VARS对应clear-ars.mk脚本

还可以导入其他的mk文件来帮他完成一些类似人物，例如配置全局变量的时候就导入当前目录下的envsetup.mk（加载产品的product\_config.mk)

3、配置产品product.mk

关于product的设定，由build/core/product\_config.mk处理，用product.mk提供宏载入。根据androidproducts.mk内容，确定product目标

4、加载产品product\_config.mk

根据lunch选择的product在COMMON\_LUNCH\_CHOICE寻找对应的product，加载androidproducts mk文件中的PRODUCT\_MAKRFILES指定mk文件