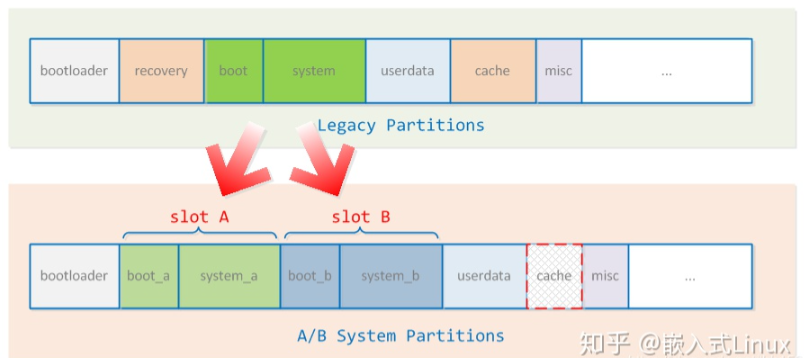
**Android A/B系统**

###### 什么是Android A/B系统

A/B 系统升级，在磁盘上开辟两个存储空间A/B存储空间，在升级过程中保证有一个可以正常运行的系统，采用这种方式可以大大提升更新的成功性。

###### A/B系统分区和普通系统分区

普通系统只需要一个boot存储空间和一个system存储空间，但是A/B系统需要开辟两个boot存储空间和两个system存储空间。



##### **Android A/B系统怎么用？**

##### 典型的应用场景有以下4个（假定当前从B分区启动）：



**普通场景（Normal cases）**

最常见的情形，例如设备出厂时，A分区和B分区都可以成功启动并正确运行，所以两个分区都设置为bootable，successful，但由于是从B分区启动，所以只有B分区设置为active。

**升级中（Update in progress）**

B分区检测到升级数据，在A分区进行升级，此时将A分区标识为unbootable，另外清除successful标识；B分区仍然为active，bootable和successful。

**更新完成，等待重启（Update applied, reboot pending）**

B分区将A分区成功更新后，将A分区标识为bootable。外，由于重启后需要从A分区启动，所以也需要将A分区设置为active，但是由于还没有验证过A分区是否能成功运行，所以不设置successful；B分区的状态变为bootable和successful，但没有active。

**从新系统成功启动（System rebooted into new update）**

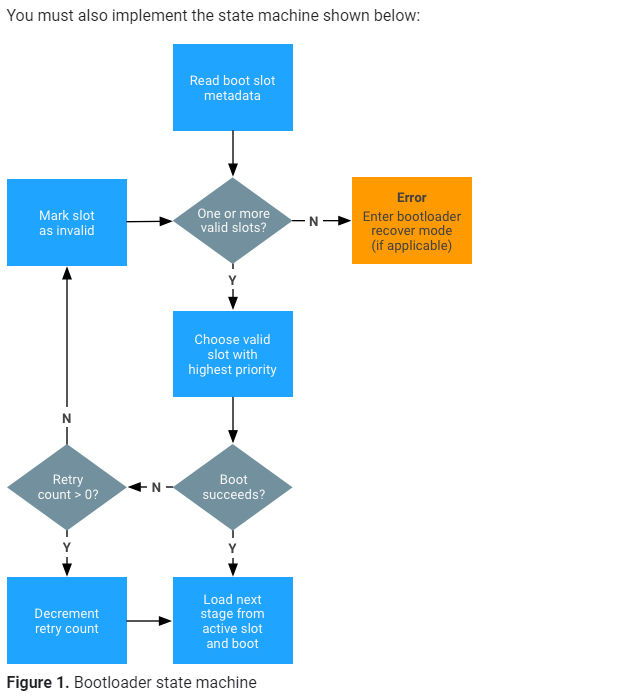
设备重启后，bootloader检测到A分区为active，所以加载A分区系统。进入A系统后如果能正确运行，需要将A分区标识为successful。对比第1个普通场景，A和B系统都设置为bootable和successful，但active从B分区切换到A分区。至此，B分区成功更新并切换到A分区，设备重新进入普通场景。

##### **如何实现Android A/B系统？**

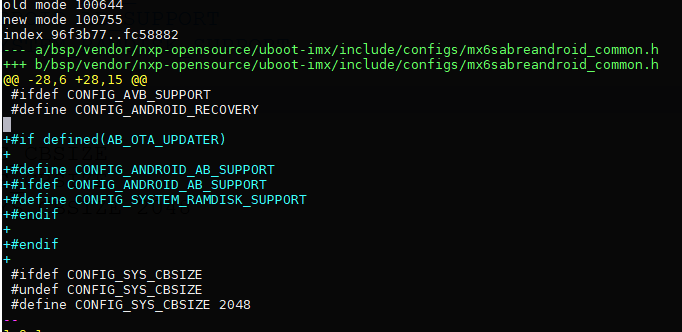
* 实现boot control HAL

Boot control HAL简单说就是下图的状态机，来选择启动slot，标记slot有效性。

在此次Android 9中，uboot已经实现了该状态机，只需要打开相关的编译开关。



修改

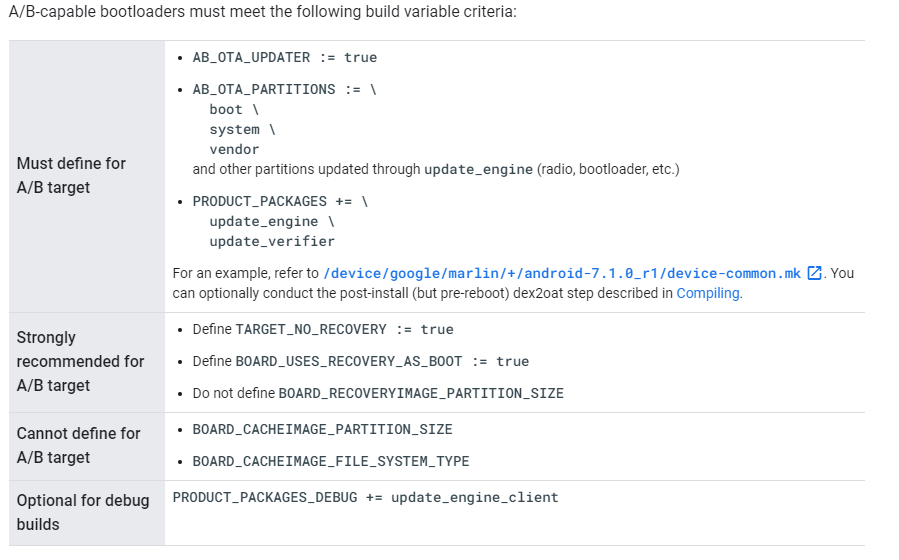


* 配置Kernel

Ensure kernel command line arguments contain the following extra arguments

skip\_initramfs rootwait ro init=/init root="/dev/dm-0 dm=system none ro,0 1 android-verity <public-key-id> <path-to-system-partition>"

* 配置编译变量

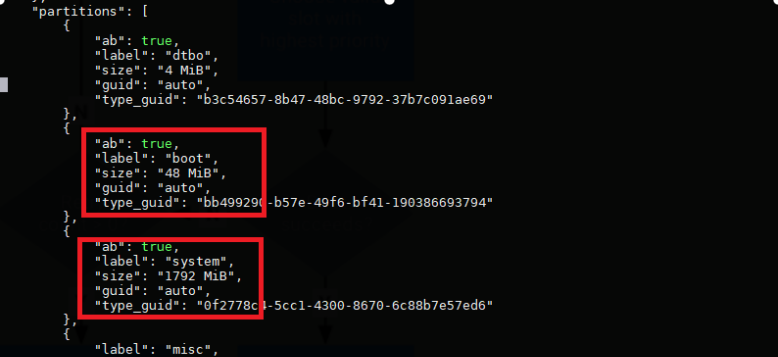


* 设置分区

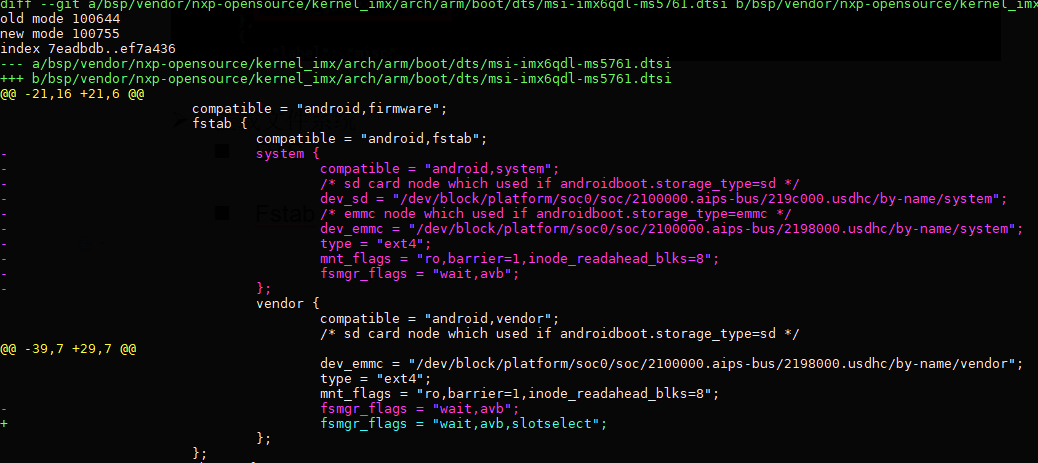
A/B devices do not need a recovery partition or cache partition because Android no longer uses these partitions. The data partition is now used for the downloaded OTA package, and the recovery image code is on the boot partition. All partitions that are A/B-ed should be named as follows (slots are always named a, b, etc.): boot\_a, boot\_b, system\_a, system\_b, vendor\_a, vendor\_b.

* SD卡分区制作

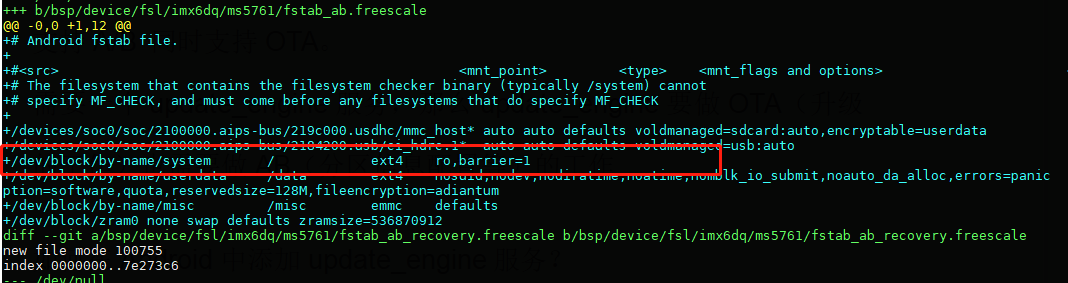
使用分区表制作SD卡分区。



* 挂载文件系统
  + DTS （优先级高）



* + Fstab



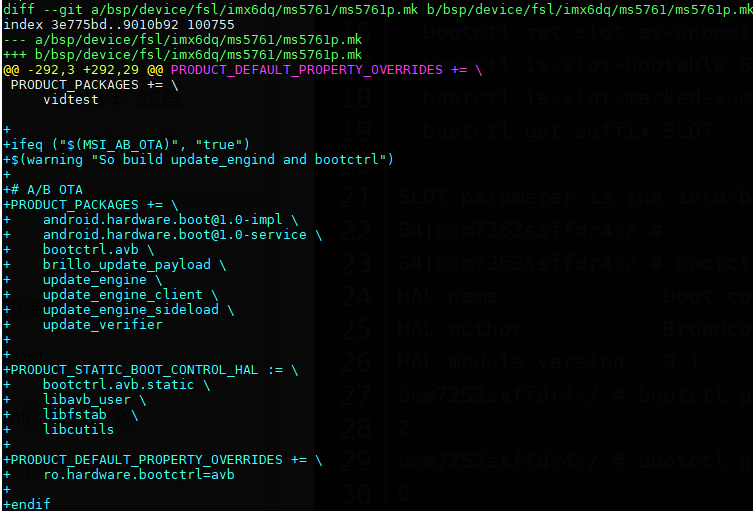
###### 什么是Android A/B OTA系统？

支持A/B 同时支持OTA。

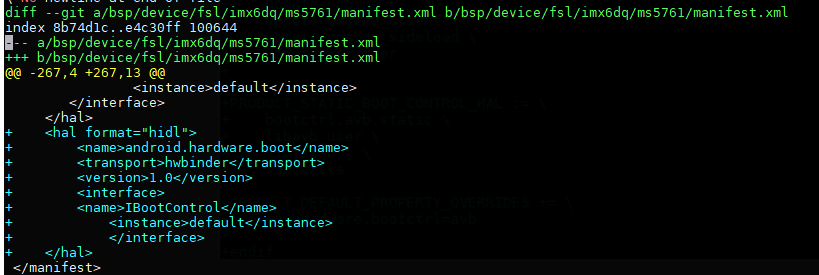
需要一个update\_engine服务，那么update\_engine要做OTA（升级的工作）也需要做AB（分区信息配置等）的工作

如何在Android中添加update\_engine服务？

依赖的包



AndroidManifest应用清单



生成OTA全量包和增量包



遇到的问题：

1. Uboot无法引导启动kernel？

原因：镜像做的有问题，修改下面编译变量。

/device/fsl/imx6dq/ms5761/BoardConfig.mk

#Enable AVB

BOARD\_AVB\_ENABLE := true

TARGET\_USES\_MKE2FS := true

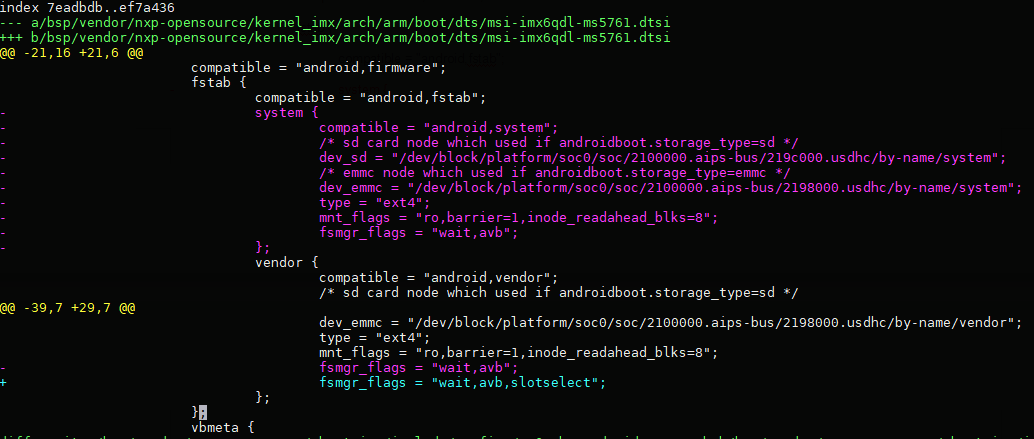
-BOARD\_INCLUDE\_RECOVERY\_DTBO := true

-BOARD\_USES\_FULL\_RECOVERY\_IMAGE := true

1. 启动后Init进程执行初始化脚本init.rc失败，找不到脚本文件？

原因：根目录已经mount，已经包含了system，不需要再重新mount。

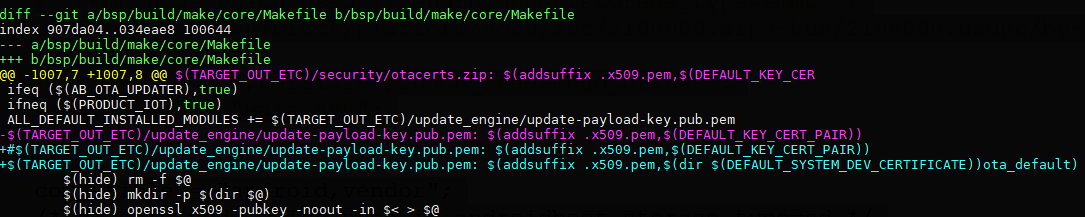
kernel command line中有 root="/dev/dm-0"



1. OTA升级校验失败？

原因：公钥私钥不匹配。

修改下面脚本。



参考链接：

<https://source.android.google.cn/devices/tech/ota/ab/ab_implement>

<https://blog.csdn.net/guyongqiangx/article/details/71334889>