NOTE: 下载干净代码并做好本地分支

1, 拷贝 prebuilt toolchain 至 prebuilt 目录，并解压；

2, 拷贝需要的库至 /usr 目录;

NOTE: 继续按如下步骤做 patch: cd ..; patch -p1 < XXX\_name.patch

3, 对 build project 应用 patch： build\_support\_sb51.patch

设置环境变量 export WITH\_SB51=1 后，才可以使用 sb51 toolchain 编译；

如果欲退回原有的4.8 toolchain，则可以设置 WITH\_SB51=0；

4, 对 system/core 目录应用 patch： system-core\_support\_sb51.patch

5, 对 U-boot64 目录应用 patch：u-boot\_support\_sb51.patch

以下烧写 pac 时，可以不用替换这个新编译的 uboot，老版本应该依旧可用；

这里仅为支持 sb51 的编译，做了些修改；

6, [暂时忽略]对 kernel 目录应用 patch 以支持 sb51 的编译：

ArchArmIncAsmFtraceH\_support\_sb51

7, 对 kernel 目录应用 patch 以支持 kasan 功能：

kernel\_kasan.patch

注意两点：

A, 需要在 .../arch/arm/configs 下面适当的 defconfig 中加入 kasan 的配置：

3174 CONFIG\_HAVE\_ARCH\_KGDB=y

3175 # CONFIG\_KGDB is not set // 定位到这一项，在其后面加下面两行

3176 CONFIG\_KASAN=y

3177 CONFIG\_KASAN\_OUTLINE=y

3178 CONFIG\_TEST\_KASAN=y

3179 # CONFIG\_TEST\_STRING\_HELPERS is not set

7731gea 手机对应的 sp7731gea\_hd-dt\_defconfig，已经由该 patch 添加。

其他手机可根据型号不同添加。

B，应用patch 的时候，因为其中包含Ruby 先前这个文件中打的 patch , 所以会可能

会出现 reject 的错误，根据实际情况修改即可：

arch/arm/vfp/vfpmodule.c

8, 对kernel 目录 VBC Codec Driver patch:

kernel\_SPRD\_ASoC\_VBC\_Codec\_Driver.patch

这是原来代码驱动中的问题，应fix掉。

9, 重新编译 mali module. 并替换进 system.img 中去。

cd vendor/sprd/open-source/libs/gpu/utgard/driver/mali; mm

7731 因为 mali driver 放在 system.img 中，所以我们要重新编译，否则在新产

生内核环境下这个 driver insert 不进去，导致 surfaceflinger 等进程不断的重启又

不断的被 kille。9830 因为这个driver被放在bootimage中,所以无需再替换。

替换的时候，可用 loop 方式挂载 system.img 到主机，然后find/cp mali.ko即可。

注意，现在还不能直接用 sb51 去编译 system.img。若要支持，可能会需要较多的

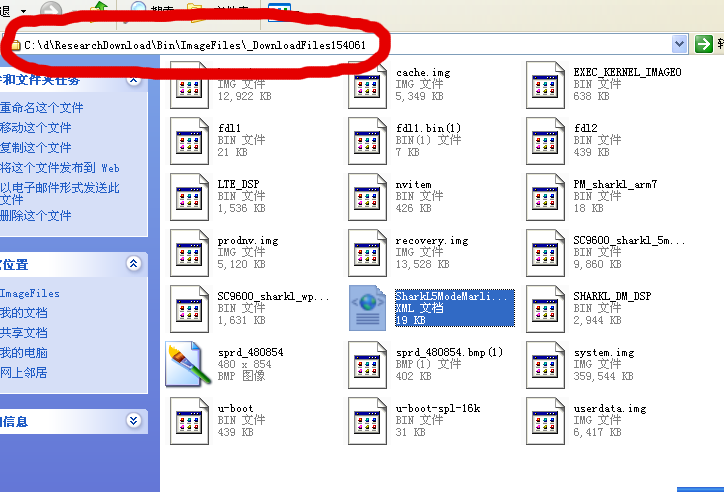
工作量来修改代码。

10, 烧录注意，先扩大 boot 分区烧一次后，然后再代入自己的 boot.img /u-boot.bin/system.img 进行烧录，否则可能在烧 boot.img 时候会出现 Failed 错误(具体原因不明确，可能是烧录工具的问题，没有同步支持分区大小的修改和较大boot.img的设定)

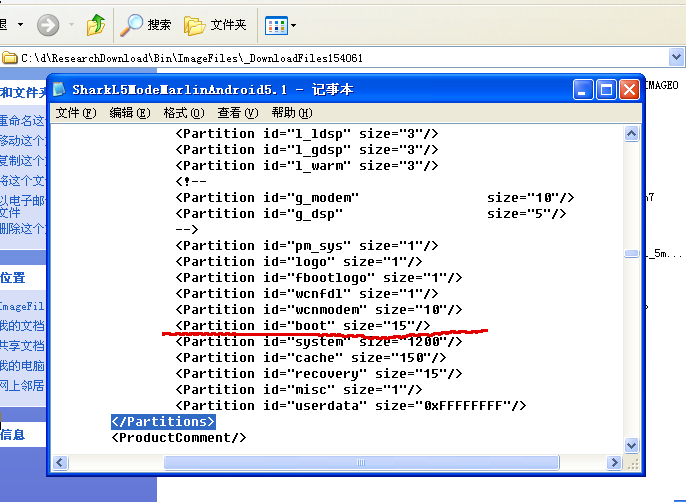
扩大boot 分区的步骤：

A，先将 pac 包load到RD工具，然后在RD工具所在 Bin 目录下的

ImageFiles\\_DownloadFilesXXXXX 目录中找到对应的 xml 文件，进行编辑：



B，修改boot 分区的大小，由原来的 15MB 扩大至35MB。



C，先直接打一个新的 pac 包，烧一次，待能重启后，再代入几个自己的 image

进行烧录。

