**Unisoc710平台在android10上编译指南**

文档版本：v1.0

发布日期：2021-3-17

作者：赵春春

**变更信息**

|  |  |  |  |
| --- | --- | --- | --- |
| **文档版本** | **发布日期** | **更新说明** | **作者** |
| V1.0 | 2021-3-17 | 初始完成《710平台上android10编译指南》文档的编写 | 赵春春 |
|  |  |  |  |
|  |  |  |  |

目录

**[1. 编译环境配置 3](#_Toc3835)**

[(1) 硬件要求 3](#_Toc1497)

[(2) 操作系统要求 3](#_Toc14031)

[(3) 系统依赖库的安装 3](#_Toc1703)

[(4) 标准软件安装（python，jdk，git，gcc） 3](#_Toc17572)

**[2. 编译命令 4](#_Toc16659)**

[(1) 把闭源的代码复制到开源代码中： 4](#_Toc11202)

[(2) 在代码的根目录下 4](#_Toc564)

[(3) 常见问题： 4](#_Toc22542)

**[3. 生成PAC命令 5](#_Toc13532)**

[(1) 修改build\_pac.sh： 5](#_Toc1424)

[(2) 修改.sign： 5](#_Toc6022)

[(3) pac生成的目录： 6](#_Toc26912)

**[4. 单编模块和下载 6](#_Toc25294)**

1. 编译环境配置
   1. 硬件要求

ROM 至少 250G：

对于android10.0的code，压缩包就有200G，解压后大概在260G，在编译完成后能够有380G左右；

RAM 至少 16G：

否则编译大概率会出现 Exception in thread "main" java.lang.OutOfMemoryError: Java heap space。

* 1. 操作系统要求

建议选择 14.04 版本的 64 位 ubuntu 系统，查看 ubuntu 的具体版本号命令为：

lsb\_release –a

但是现实中我们电脑上很少有14.04的系统了，都是以16.04和20.04多一些，后面会对需要特殊处理的系统有标注。

* 1. 系统依赖库的安装

sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev ccache libgl1-mesa-dev libxml2-utils xsltproc unzip libssl-dev

* 1. 标准软件安装（python，jdk，git，gcc）

其中git和gcc在上面命令里已经安装，只需要安装python和jdk

jdk安装：参考网址：<https://blog.csdn.net/baidu_38172402/article/details/80747916>

Python2.7的安装：

对于Ubuntu20.04以前版本的安装：直接安装Python2.7就可以

sudo apt-get install python2

对于Ubuntu20.04及以后的版本：因为系统本身就包含了Python3.0，而且这个Python3.0是不可以被卸载的，因为卸载后Ubuntu20.04的桌面系统就不可以使用了，会出现登陆界面上鼠标键盘没有反应的情况。所以需要在保留Pyhon3.0的情况下，安装Python2.7。

参考网址：https://blog.csdn.net/weixin\_39981624/article/details/110964940

sudo apt install python2 //安装python2

python2 -V或者(--version)//查看python2是否安装成功

python -V或者(--version)// 查看默认的python版本

ls /usr/bin/python\*//查看系统上安装的python版本

sudo update-alternatives --list python//列出当前python配置的alternative，如果没有配置，会有提示

sudo update-alternatives --install /usr/bin/python python /usr/bin/python2 1 //在alternative把python2的item设置为1

sudo update-alternatives --install /usr/bin/python python /usr/bin/python3 2 //在alternative把python3的item设置为2

sudo update-alternatives --list python //列出当前python配置的alternative，查看自己配置的是否正确

sudo update-alternatives --config python //更新alternative中的配置，选择python2所在的item，这样python就会被设置为默认的python命令

python -V或者(--version)//查看当前默认的python命令是否正确。

1. 编译命令
   1. 把闭源的代码复制到开源代码中：

编译之前需要把展锐的闭源code复制到开源code中，对于展锐的闭源code有两种获得方式：

展锐会提供闭源的压缩包：如proprietories-s9863a1h10\_Natv-userdebug-native.zip

等，这种情况，把压缩解压后，会获得bsp和out两个目录，这里把这两个目录覆盖开源code根目录下的bsp和out目录即可；

展锐通过git发布他们的code，这个时候我们拉下来的code中包含了闭源code，一般

他们的闭源code存放的位置在vendor/sprd/release/IDH/下，这里会有对应项目的文件夹，例如我们710项目目前开发板编译需要的是ud710\_2h10u\_native-userdebug-native，这个时候进到该项目文件夹下，也会发现有bsp和out两个目录，把他们覆盖到项目根目录下的bsp和out即可。



* 1. 在代码的根目录下
     1. source build/envsetup.sh
     2. lunch 86 // ud710\_2h10u\_native-userdebug-native对应的是86的选项
     3. make -j16
  2. 常见问题：
     1. Fatal error:openssl/bio.h等

安装openssl： sudo apt-get install openssl

如果还是编译报错，则安装

sudo apt-get install libssl-dev build-essential zlibc zlib-bin libidn11-dev libidn11

* + 1. You are attempting to build with the incorrect version of java等：

这是android和java版本不匹配，需要确定是否安装了jdk，以及是否配置好了环境变量；

* + 1. ImportError: No module named Crypto.Signature

这个需要安装pycrypto和python-dev

sudo apt-get install python-dev

对于pycrypto的安装比较麻烦，因为需要使用pip安装，所要提前安装对应的pip；

对于低于ubuntu20.04的系统，可以直接sudo apt-get intall pip，然后sudo pip install pycrypto

对于ubuntu20.04以及以上的系统，则涉及到pip2和pip3的问题，这里需要安装pip2，而不是pip3。

安装pip2： wget https://bootstrap.pypa.io/pip/2.7/get-pip.py && sudo python2.7 get-pip.py

这里注意下载之前看下当前目录是否已经有了get-pip.py文件，如果有了，记得删除。

然后可以通过pip --version查看安装到哪里，pip会根据当前python的版本安装到对于版本的dist-packages下面，如果pip安装在python2.7下才是正确，如果在python3下面就需要重新安装。

安装pycrypto：

sudo pip install pycrypto

1. 生成PAC命令

编译成功后，需要生成PAC文件才能用Unisoc的ResearcehDownload工具下载到手机中：

具体生成PAC文件的命令如下：

//进入到unisoc-android10-UFS/vendor/sprd/release/IDH/Script目录下：

./build\_pac.sh -a ud710\_2h10u\_native-userdebug-native -b PAC

在执行上述命令前需要做两件事，一个是把modem签名，一个是把modem的对应文件copy到对应的打包目录，所以这里需要先修改两个文件：

* 1. 修改build\_pac.sh：

unisoc-android10-UFS/vendor/sprd/release/IDH/Script

在310行，去掉#，从而打开SIGN函数

- #SIGN

+ SIGN

* 1. 修改.sign：

给modem签名，并copy到指定目录

unisoc-android10-UFS/vendor/sprd/release/IDH/

ud710\_2h10u\_native-userdebug-native/ROC1\_ORCA\_Phase7\_Q\_SIGN\_U\_20C

把下面文件的路径

（/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS）

修改为自己本地code的路径(要修改为绝对路径)

Modem\_NR\_PMSYS=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/5G\_MODEM\_20C\_W20.51.5\_P9\_udx710\_PH\_CUSTOMER/Orca\_cm4\_builddir/Orca\_cm4.bin

Modem\_NR\_DSP1=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/5G\_MODEM\_20C\_W20.51.5\_P9\_udx710\_PH\_CUSTOMER/Orca\_NR\_phy\_builddir/Orca\_NR\_phy\_NR\_XC0\_DSP.bin

Modem\_NR\_AGDSP=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/5G\_MODEM\_20C\_W20.51.5\_P9\_udx710\_PH\_CUSTOMER/Orca\_PSCP\_modem\_builddir/Orca\_PSCP\_modem\_AGCP\_DSP.bin

MODEM\_L\_AGDSP=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/5G\_MODEM\_20C\_W20.51.5\_P9\_udx710\_PH\_CUSTOMER/Orca\_PSCP\_modem\_builddir/Orca\_PSCP\_modem\_AGCP\_DSP\_ROC1\_PLUS\_ORCA.bin

Modem\_NR\_DELTANV=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/5G\_MODEM\_20C\_W20.51.5\_P9\_udx710\_PH\_CUSTOMER/Orca\_PSCP\_modem\_Phone\_T7510\_builddir/Orca\_PSCP\_modem\_Phone\_T7510\_deltanv.bin

NV\_NR=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/5G\_MODEM\_20C\_W20.51.5\_P9\_udx710\_PH\_CUSTOMER/Orca\_PSCP\_modem\_Phone\_T7510\_builddir/Orca\_PSCP\_modem\_Phone\_T7510\_nvitem.bin

Modem\_L\_PMSYS=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/FM\_BASE\_V2\_CM4\_W20.48.2\_CUSTOMER/roc1\_cm4\_builddir/roc1\_cm4.bin

Modem\_NR\_PHY=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/5G\_MODEM\_20C\_W20.51.5\_P9\_udx710\_PH\_CUSTOMER/Orca\_NR\_phy\_builddir/SC9600\_Orca\_NR\_phy\_modem.bin

Modem\_NR=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/5G\_MODEM\_20C\_W20.51.5\_P9\_udx710\_PH\_CUSTOMER/Orca\_PSCP\_modem\_builddir/SC9600\_Orca\_PSCP\_modem\_modem.dat

Modem\_NR\_V3PHY=/home/chunchunzhao/local/code/new\_unisoc-10.0/unisoc-android10-UFS/vendor/sprd/release/IDH/Modem/5G\_MODEM\_20C\_W20.51.5\_P9\_udx710\_PH\_CUSTOMER/Orca\_V3\_phy\_builddir/SC9600\_Orca\_V3\_phy\_modem.bin

* 1. pac生成的目录：

unisoc-android10-UFS/vendor/sprd/release/IDH/ImageFiles目录下会生成一个.pac文件。同时也可以看到很多img和bin文件，这些都是可以用来替换ResearchDownload工具中的单个文件，达到单独更新文件的目的。

1. 单编模块和下载

在完成一次全编之后，在不改变当前编译项目的前提下，修改代码后可以使用单项的编译来编译对应的部分，加快开发的速度。

Android10.0上新增了 Super逻辑分区，引入了super镜像，Unisoc默认开启了Super逻辑分区，以方便OTA升级等，全编译的时候会在out目录下生成super镜像 。

* 1. 单独编译u-boot:

make bootloader

生成：fdl2-sign.bin u-boot-sign.bin u-boot\_autopoweron-sign.bin

* 1. 单独编译fdl1和uboot-16k：

make chipram

生成：fdl1-sign.bin u-boot-spl-16k-sign.bin

* 1. 单独编译bootimage

make bootimage

生成boot.img dt.img kernel.img ramdisk.img

* 1. 单独编译boot-debug image

make bootimage\_debug

生成 boot-debug.img，用于 XTS 测试。使用 user 版本搭配 boot-debug，可进行 root 操作。

* 1. 单独编译 Super Image：

make superimage

生成：super.img、super\_gsi32.img、super\_gsi64.img，GSI 相关 image 用于制作 GSI 版本。

* 1. 单独编译 system image：

make systemimage

生成：system.img，需要使用 vbmeta\_system.img 进行校验。

* 1. 单独编译 vendor image：

make vendorimage

生成：vendor.img，需要使用 vbmeta\_vendor.img 进行校验。

* 1. 单独编译 product image：

make productimage

生成：product.img

* 1. 单独编译 userdata image：

make userdataimage

生成：userdata.img

* 1. 如果单独编译 Settings 应用，可使用如下集中方式编译：

在目录下：

make Settings –j8

在根目录下：

mmm packages/apps/Settings/

工程整编过或通过 make 命令编译后，可在 Settings 的 Android.mk 所在目录执 行 mm 或 mma 命令来编译。

* 1. Android 10.0 上 BSP 支持独立编译，如果单独编译 Kernel，可使用 BSP 独立编译，在源码目录 bsp 下面进行 source/lunch/make。