步骤：

1. 初始化编译环境：source build/envsetup.sh
2. 设置编译目标 ：lunch aosp\_arm64-eng

所谓的编译目标就是生成的镜像要运行在什么样的设备上.这里我们设置的编译目标是aosp\_arm64-eng。编译目标的格式:BUILD-BUILDTYPE,比如上面的aosp\_arm-eng的BUILD是aosp\_arm,BUILDTYPE是eng。BUILD指的是特定功能的组合的特定名称,即表示编译出的镜像可以运行在什么环境.其中,aosp(Android Open Source Project)代表Android开源项目;arm表示系统是运行在arm架构的处理器上,arm64则是指64位arm架构;处理器,x86则表示x86架构的处理器;此外,还有一些单词代表了特定的Nexus设备。

1. 编译：make -j8

-j参数来设置参与编译的线程数量,以提高编译速度.比如这里我们设置8个线程同时编译。

# 注：make -j8 SHOWCOMMAND=TRUE 2>&1 | tee build.log

（1） make 是编译命令，

（2）-j8 这里的 8 指的是线程数量，就是要用几个线程去编译这个工程，一般会是 CPU核心数的2 倍。

（3）通过参数showcommands 将编译的命令显示出来，比如原来编译命令是make -j8 但是编译的细节没看到，如果要将编译的命令显示出来可以

（4）2是标准错误，&1是标准输出，2>&1意思就是将标准错误输出到标准输出中。 如果没有2>&1,只会有标准输出，没有错误；

（5）tee 的作用同时输出到控制台和文件

tee file //覆盖

tee -a file //追加 -a(append)

（6）make > build.log 是将所有标准输出到这个文件中，并没有定义标准错误应该是定义到了标准输出，也就是说如果make执行出现错误，那么就不会写到 build.log中，而是输出到屏幕上，

2>&1是错误和结果都重定向到build.log中！可以到当前目录看到 build.log