**实验8 鸿蒙LiteOS-a根文件系统制作**

22920212204396 黄子安

# 实验目的

1. 为demochip单板移植根文件系统
2. 通过实验的方式进一步理解操作系统的文件系统

# 二、实验环境

1、物理机：Windows

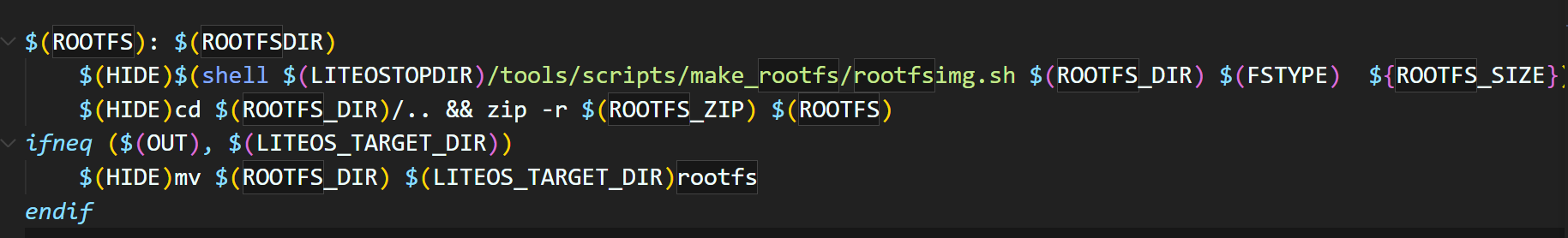
2、虚拟机：Ubuntu 18.04.6

3、开发板：imx6ull mini

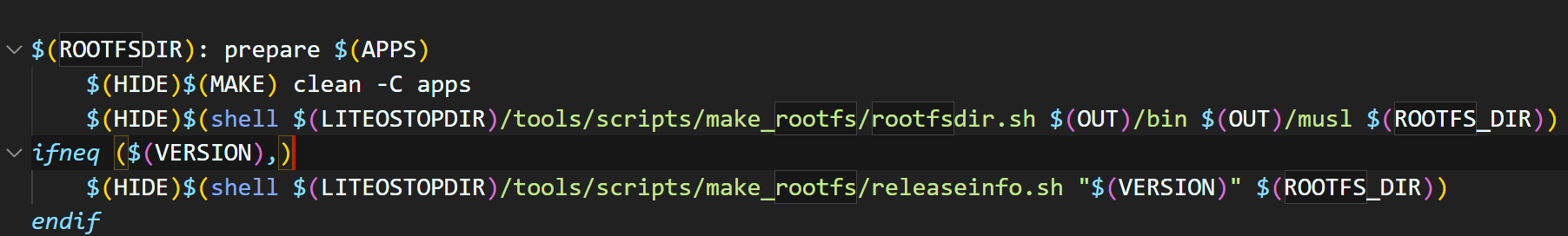
# 三、实验内容

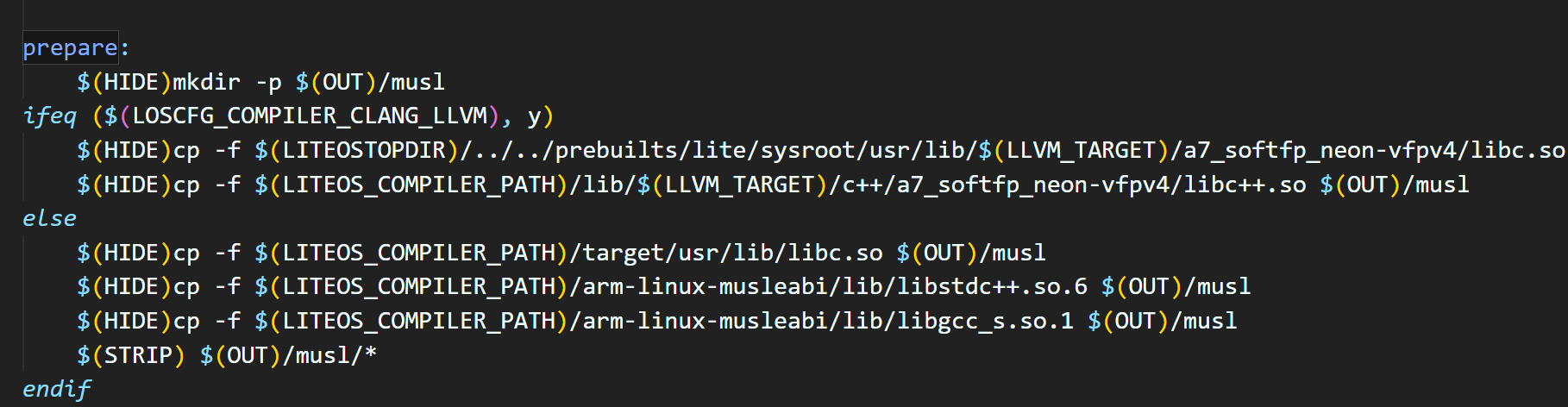
制作根文件系统的两个步骤：先创建文件夹rootfs，在其中填充内容，例如/lib，/bin，即程序的bin文件和相关的库文件；之后将这个文件制作成镜像文件烧写到开发板中

执行make rootfs的过程：执行make rootfs的时候会先去执行目标rootfsdir



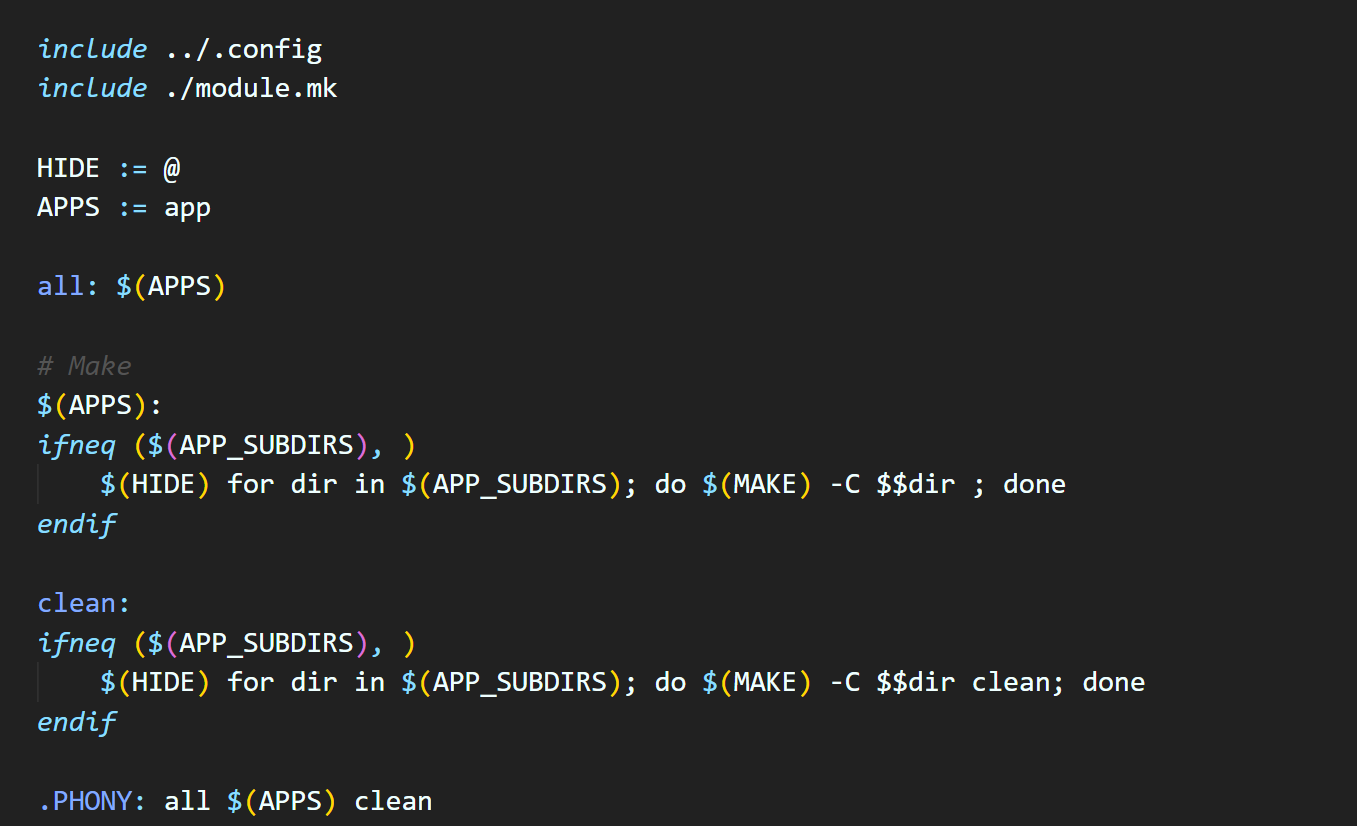
rootfsdir依赖于prepare和apps，其中prepare完成一些准备工作，先创建对应的目录，之后拷贝库文件，apps又依赖于liteos\_target，apps的作用是进入apps目录，之后执行all目标

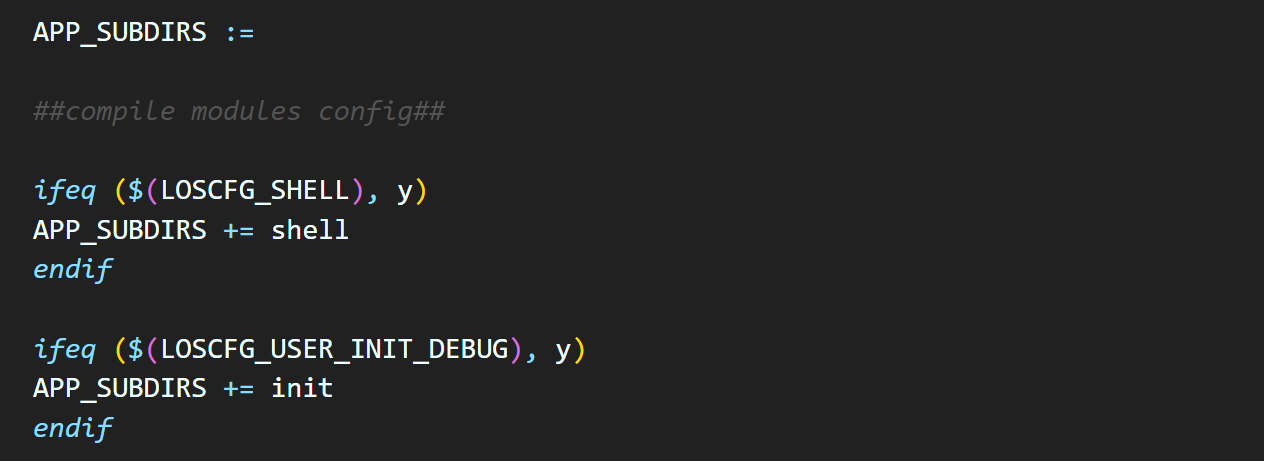




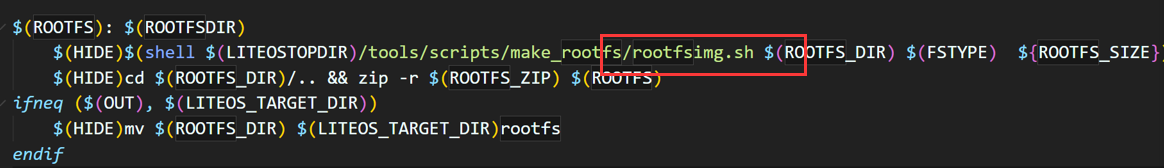


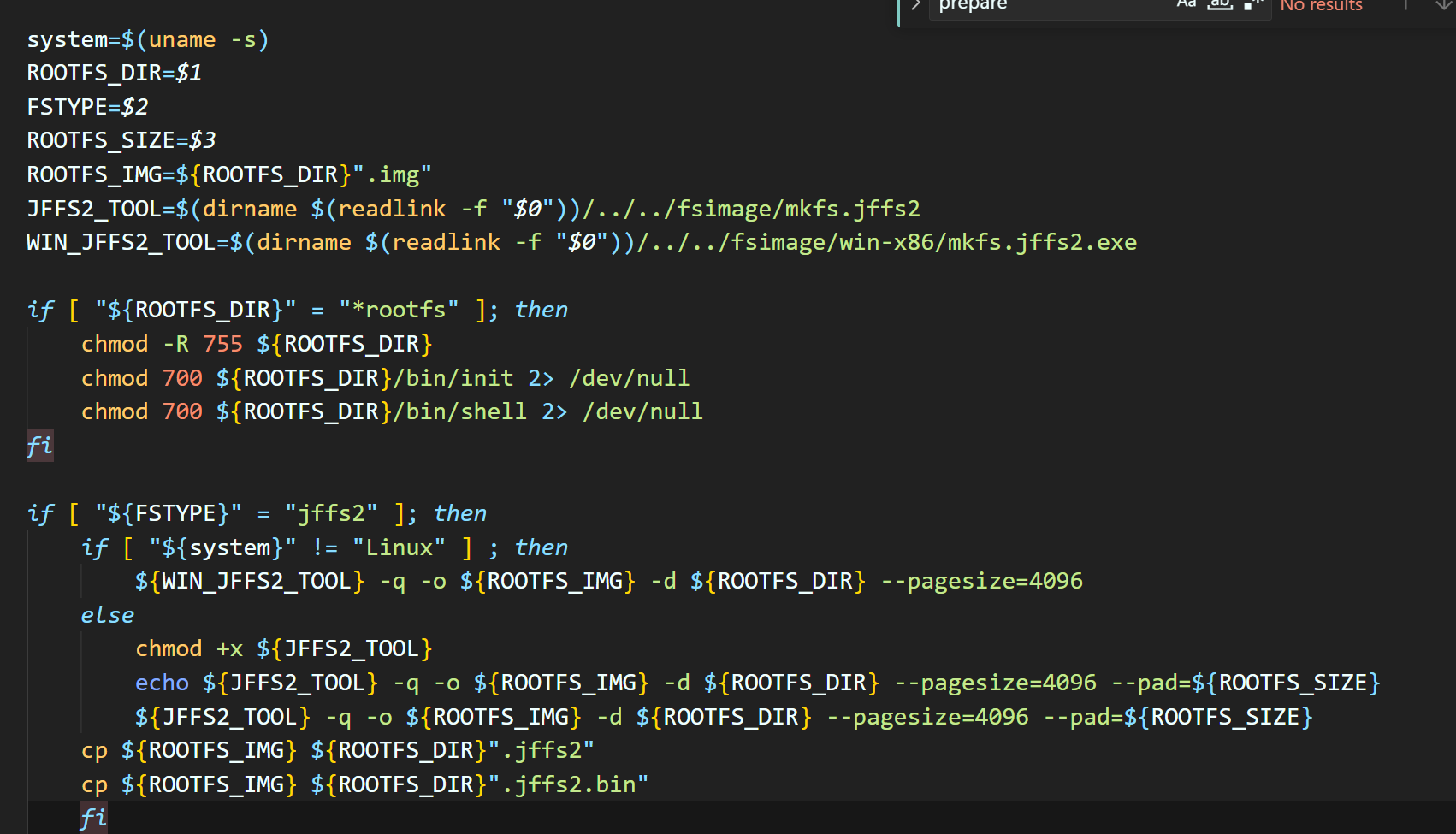
All目标里面会执行对应文件夹里的全部make，也就是对shell和init进行编译，生成对应的应用程序





之后执行对应的脚本制作镜像文件，脚本文件如下所示，会生成shell和init对应的jffs2文件



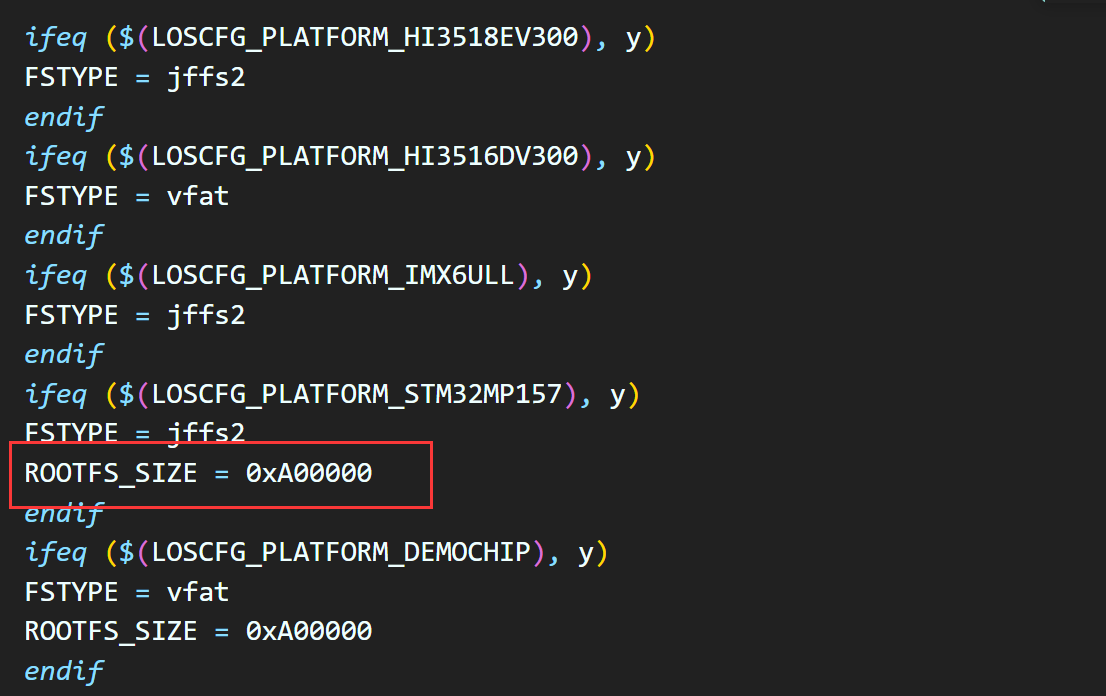


之后存在一个问题，导出的rootfs的jffs2文件中有2M，但是模拟的外存有10M，后面的8M会被变为随机值，运行的时候会警告，解决这个问题有两种方法：1、在uboot烧写前将对应的10M都设置为ff，2、使用uboot将其扩充成10M（后面8M全写ff）；imx6ull使用了第一种，157单板使用了第二种

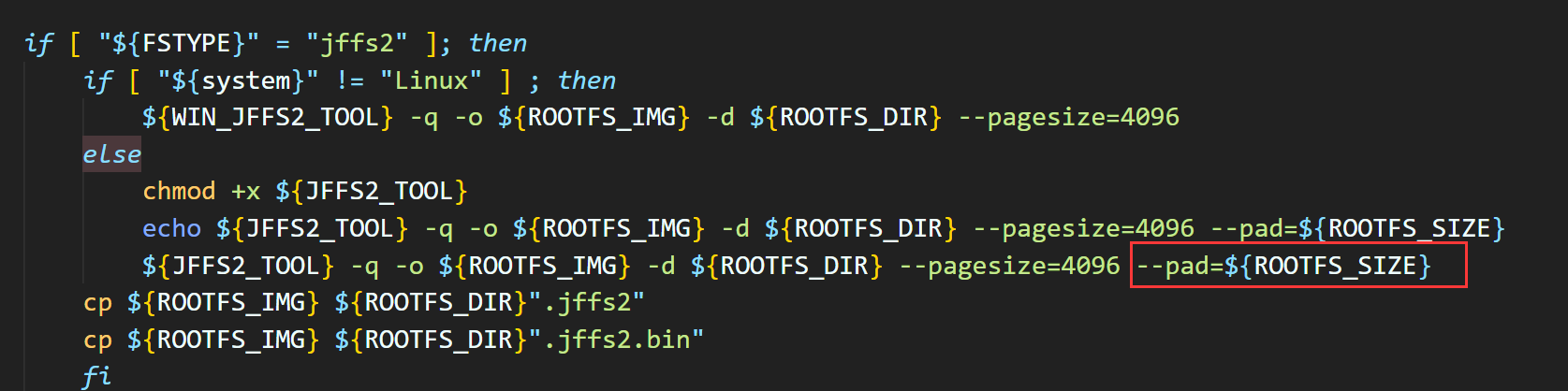
直接make rootfs，可以发现生成的rootfs大小是10M，但是理论上应该是2M



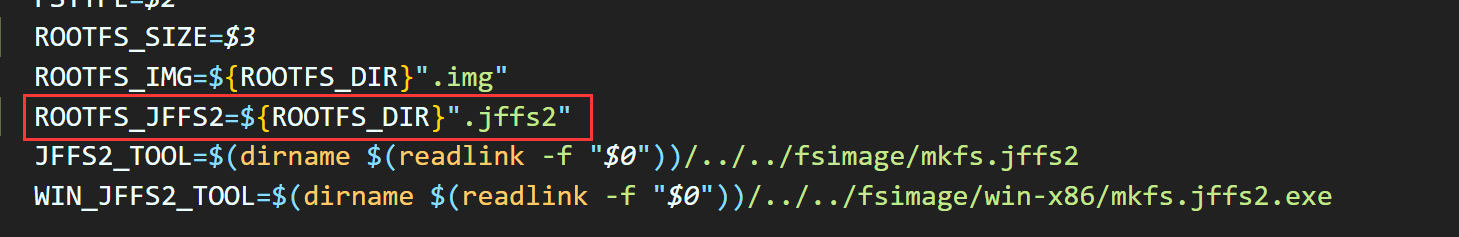
原因在于在一开始的makefile中指定了rootfs的大小是10M



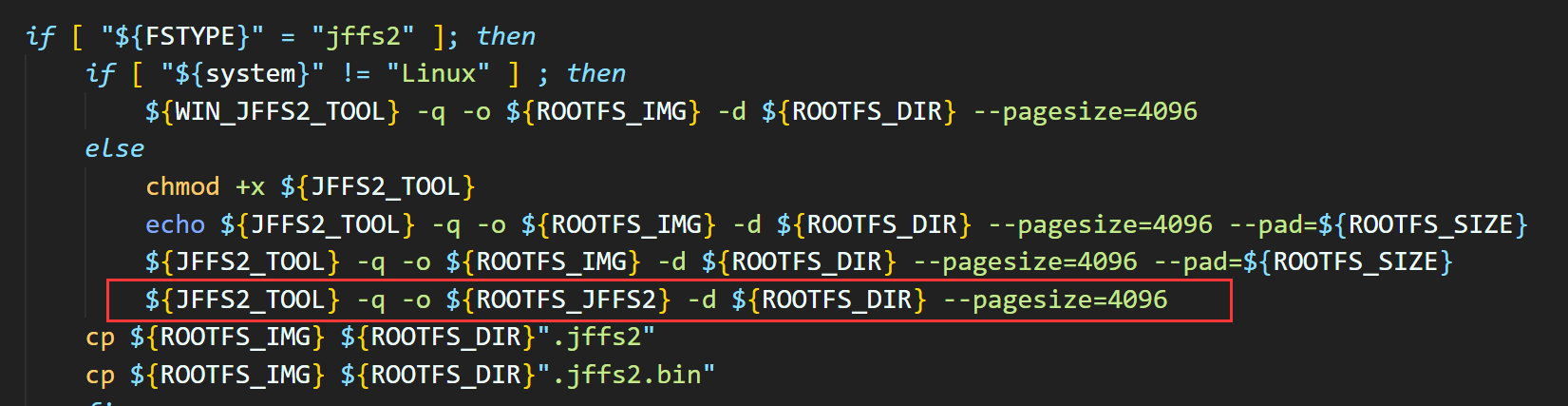
在脚本打成镜像的时候rootfs会被扩充成10M，因此重新定义个语句生成2M的rootfs



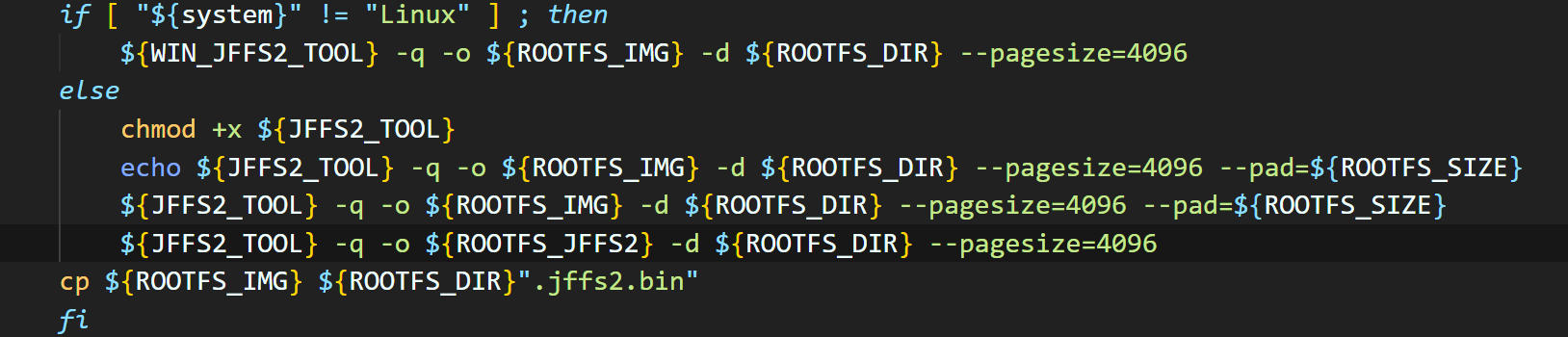
定义个新的变量ROOTFS\_JFFS2，后缀为.jffs2



之后打出一个新的镜像文件，不进行大小的扩充



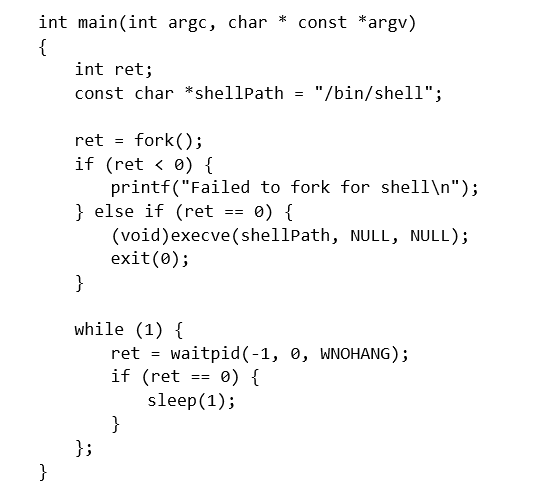
因为已经直接生成了jffs2，因此这里无需要在使用原来的cp命令来复制生成jffs2文件



再次执行make rootfs可以看到生成的rootfs小了很多

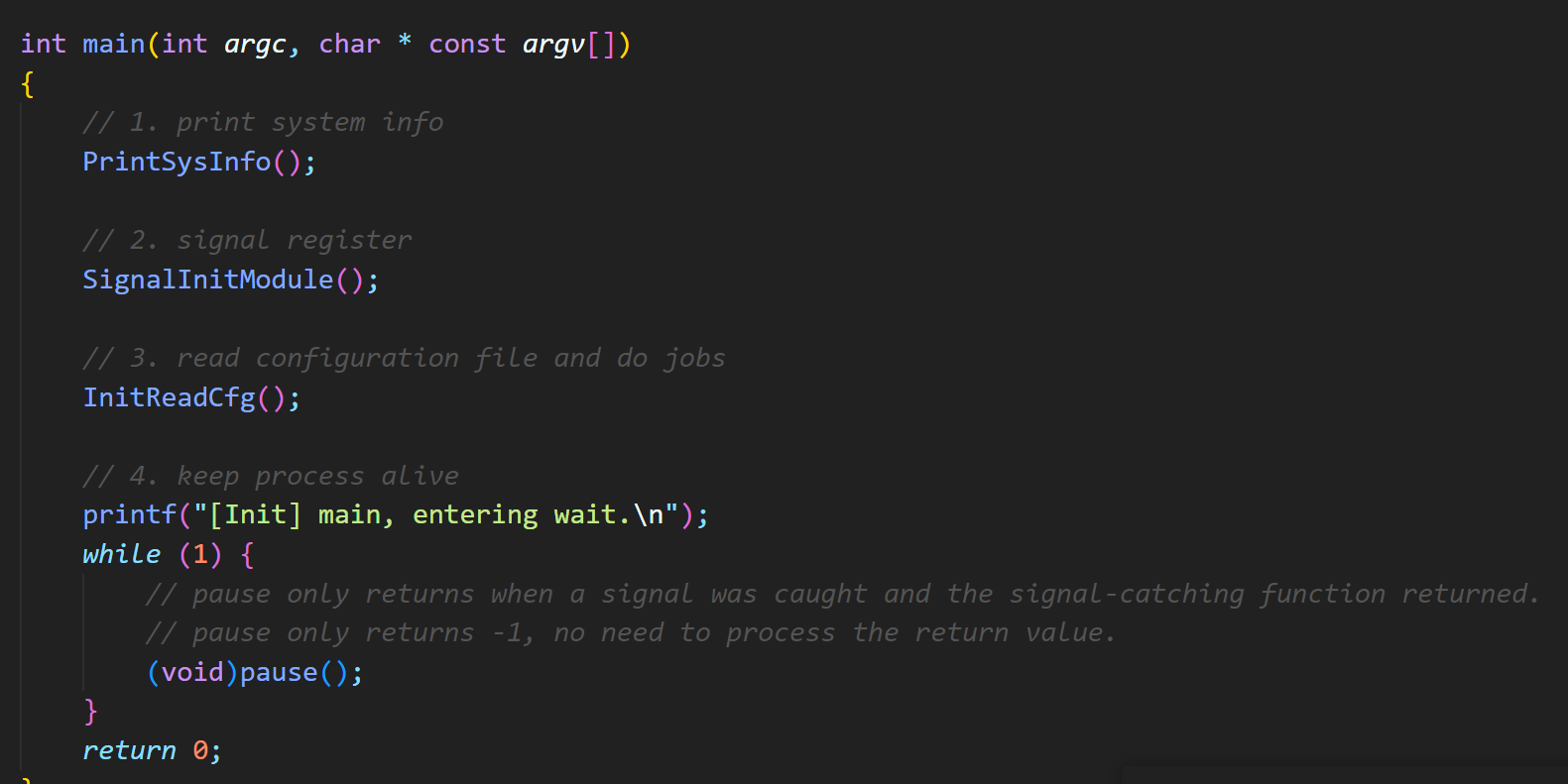


init进程的作用：读取/etc目录下的配置文件，之后用于启动对应的应用程序，测试版本的init进程只会启动shell程序，不会去根据配置文件启动别的程序，因此正式版本的需要在这个基础上扩充



正式版本的init程序如下所示，会先去读取配置文件，之后执行对应的任务；配置文件中内容可以分为两个部分：services中定义了多个服务，对应某些APP，jobs可以去定义一些APP，也可以去启动服务，包括pre-init预先执行的初始化，init初始化操作和post-init即最后的初始化操作

init进程最后启动了配置文件中被定义好的别的程序，从而整个操作系统完成启动工作。



# 四、实验心得

本次实验对Litsos-a的根文件进行了介绍，阅读了将对应文件夹制作成rootfs.jffs2的makefile执行过程，从而对Liteos一开机便进入shell程序的过程有初步了解，此外也介绍了init进程启动其他进程的步骤，通过配置文件来对程序启动进行控制，从操作系统到进程运行跨进了一步，加深了对于Liteos的理解。