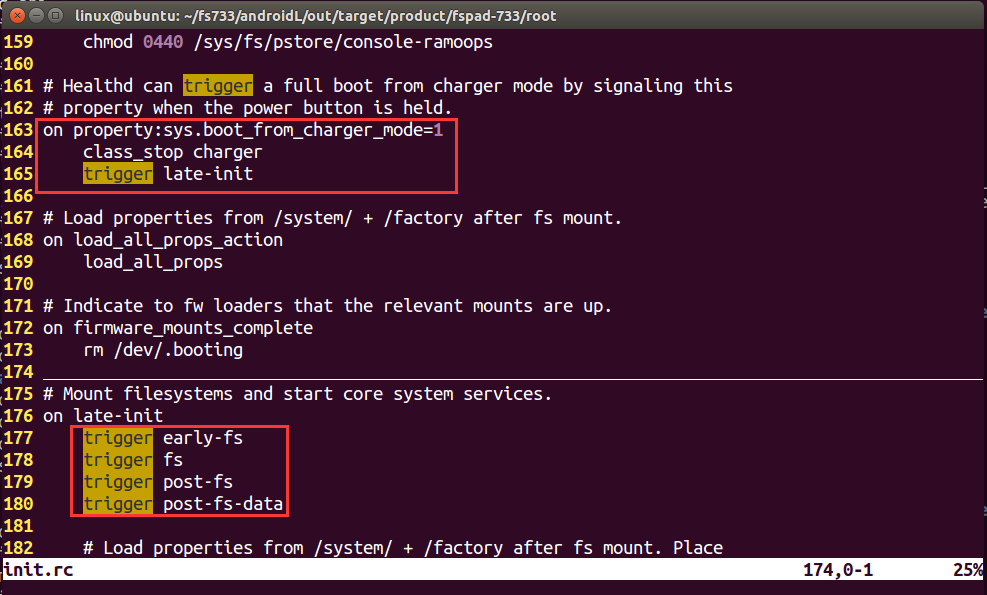
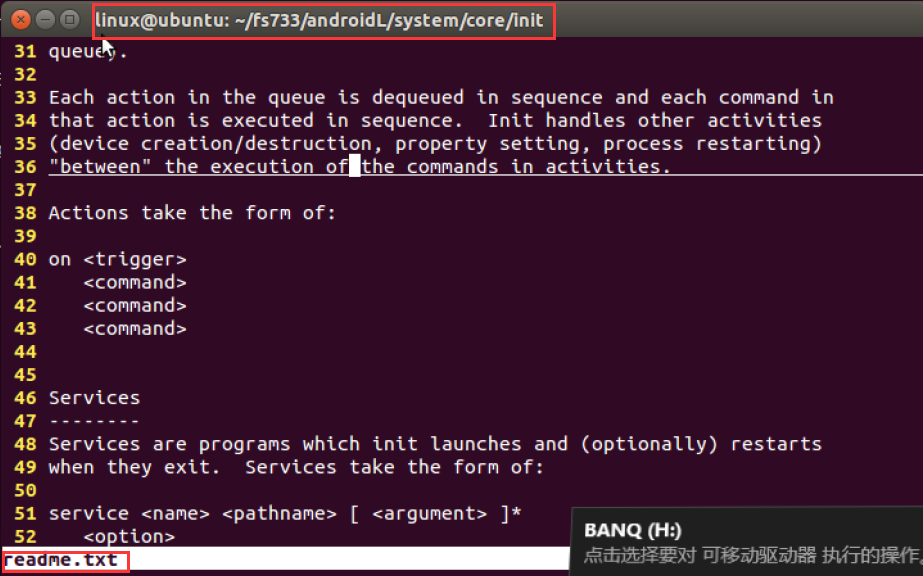
## Init.rc文件



Init.rc文件语法说明文档



## Android.mk文件语法分析

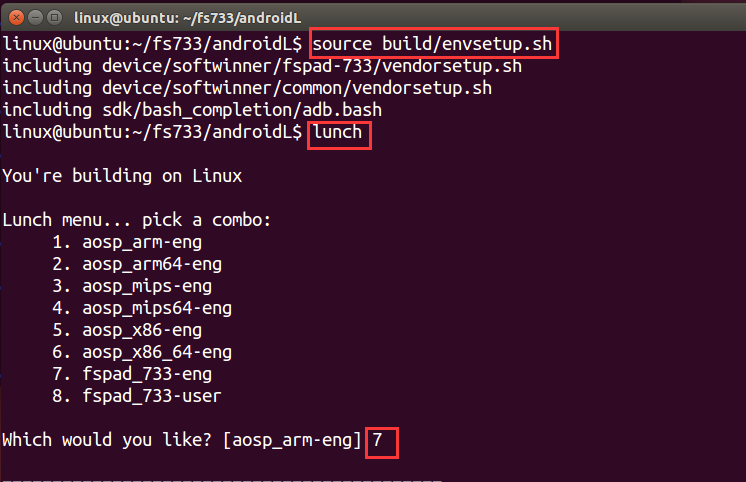
## Android源码编译与一键刷机

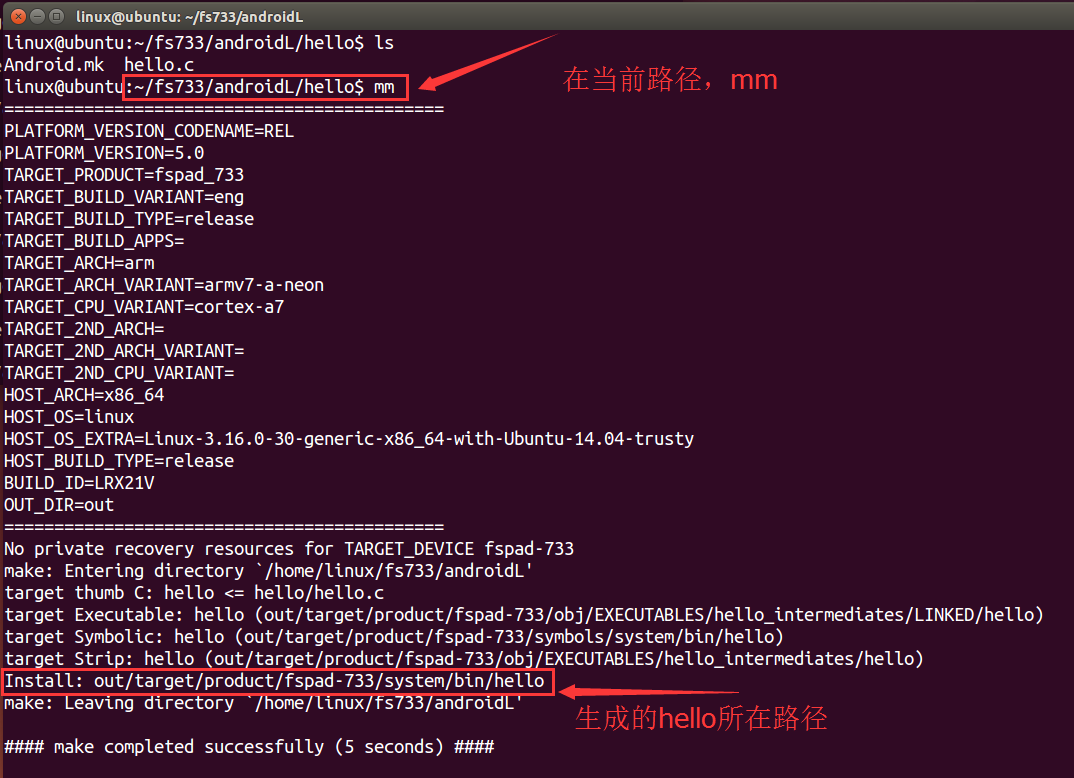
安装“豌豆荚”驱动后，重启电脑



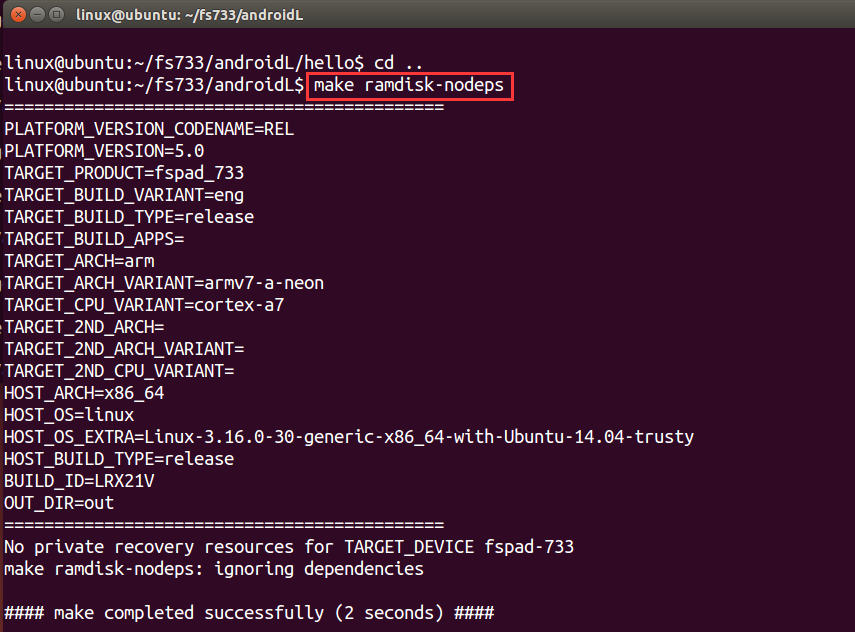


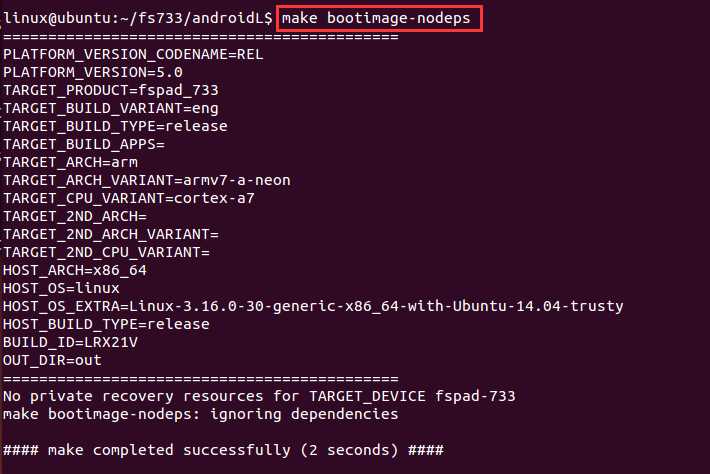
### 对指定路径的源码进行编译—当前终端环境配置

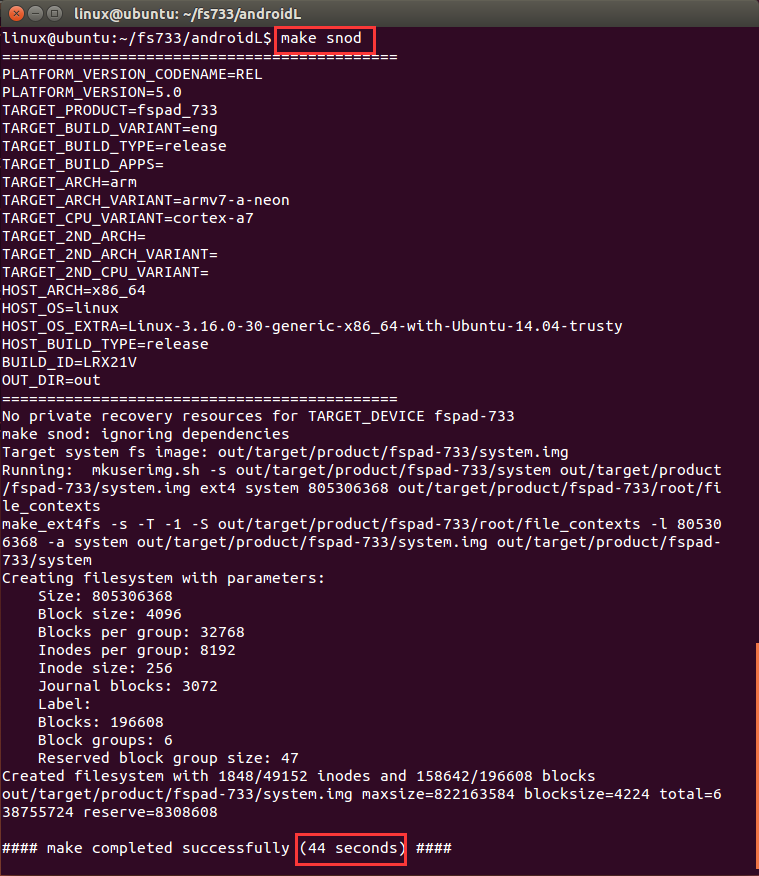




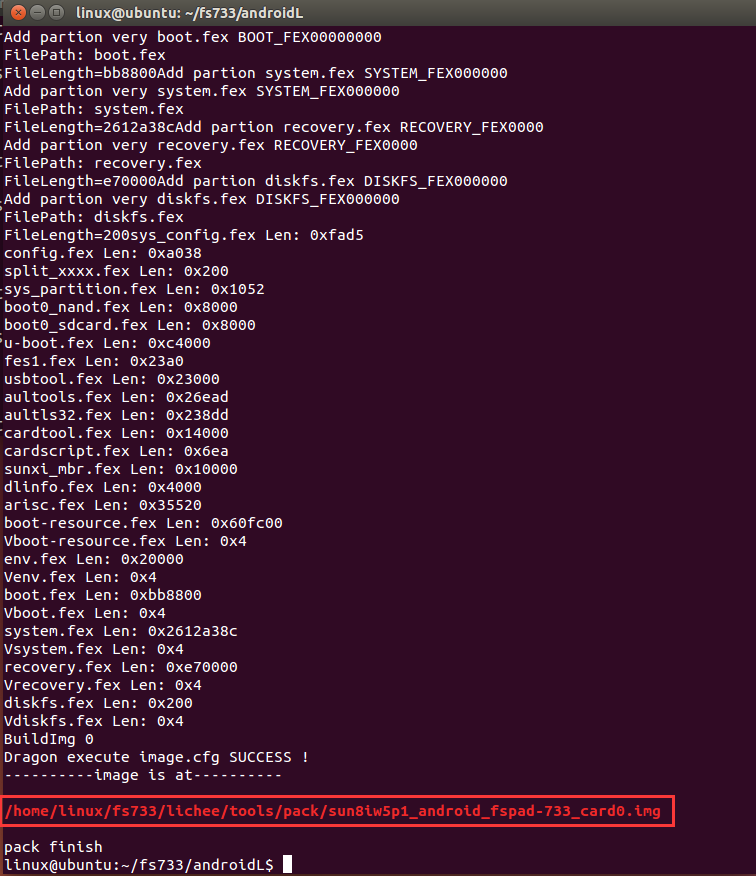
在linux@ubuntu:~/fs733/androidL$ androidL路径下







打包生成最终的镜像文件 pack -d



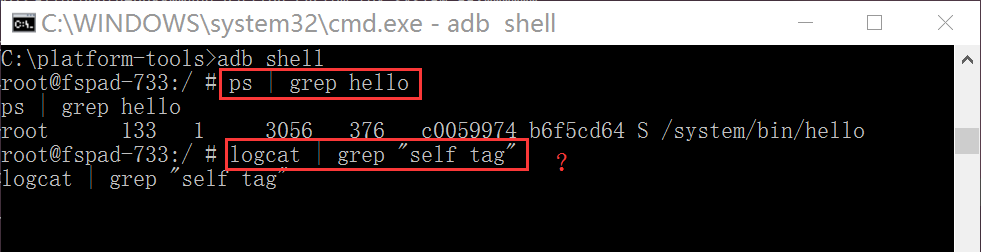


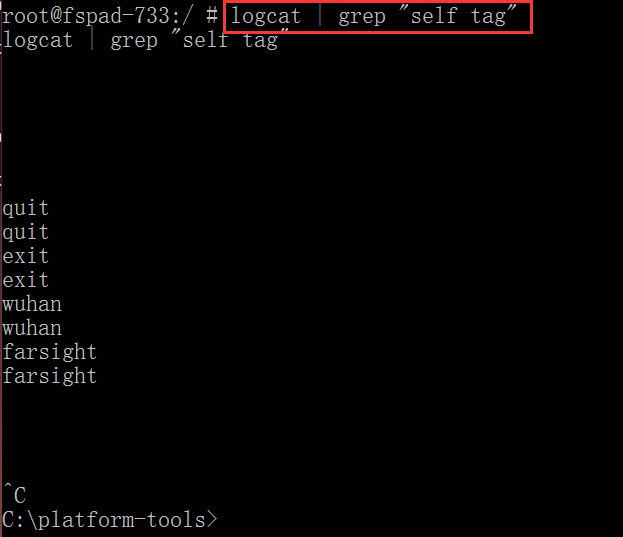


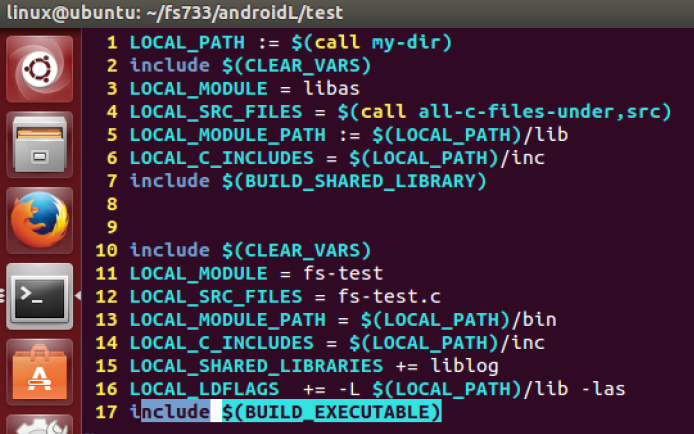


刷机后Android平板启动可能需要7~8min时间。



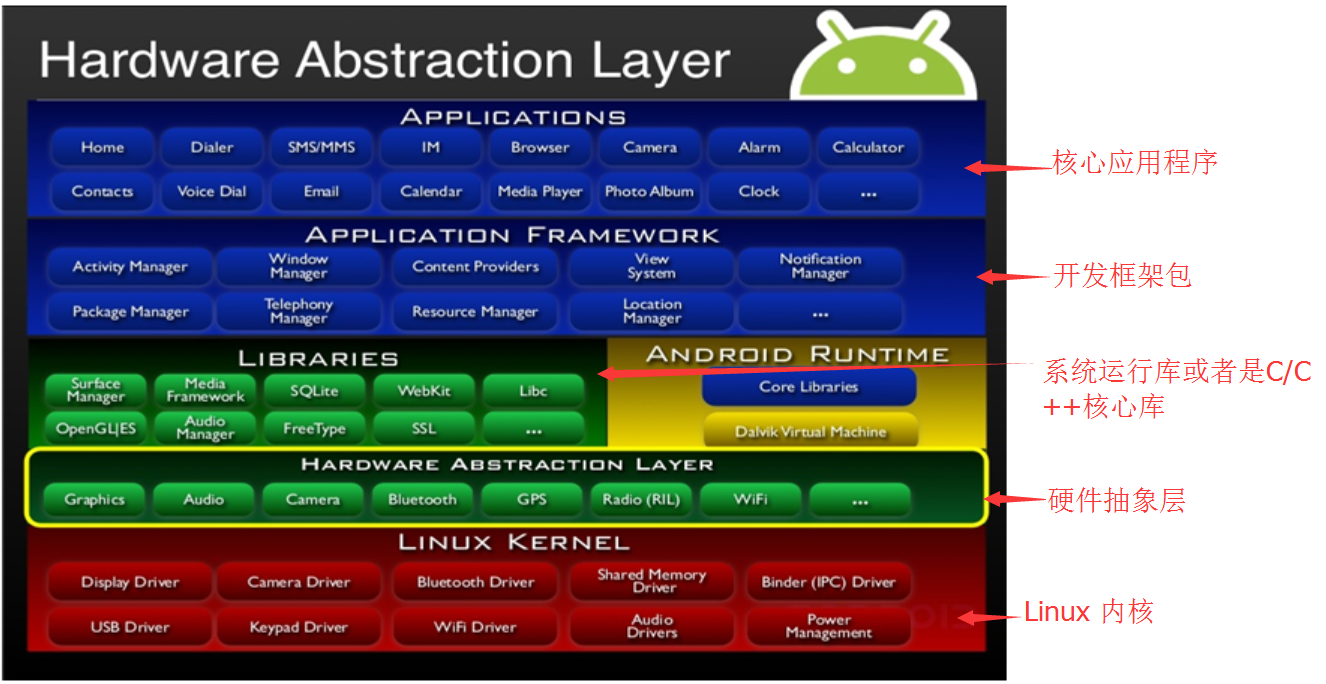


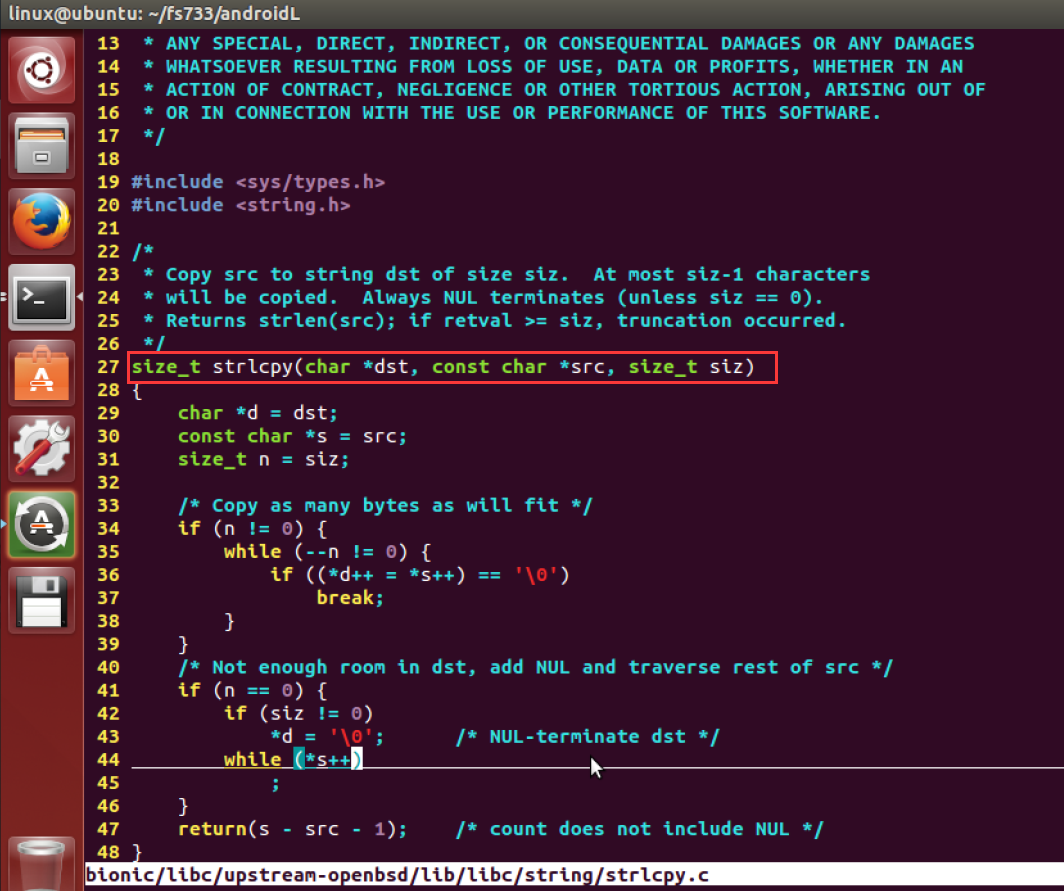




## Android架构分层之HAL层——硬件抽象层

### Android基本框架结构





strcpy有什么bug？这个strlcpy解决了什么问题？

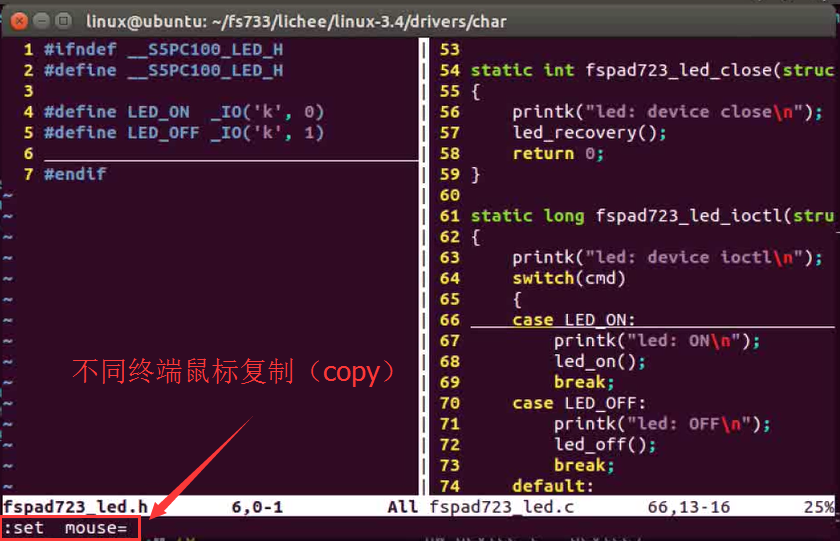
地址越界的问题；

加‘\0’的问题。



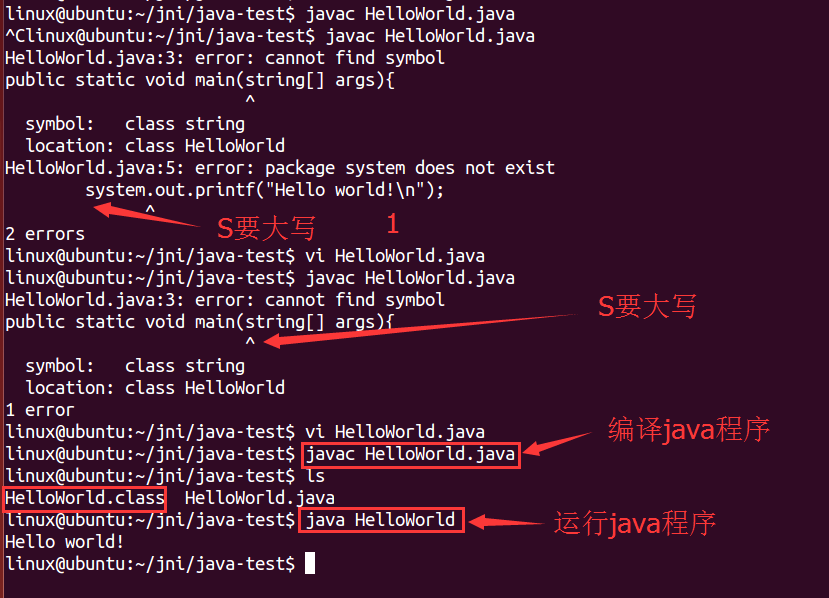
字符串替换 （ g ---全局替换）

在内核驱动中找到设备名，这里也可以看到操作硬件的接口函数ioctl( );





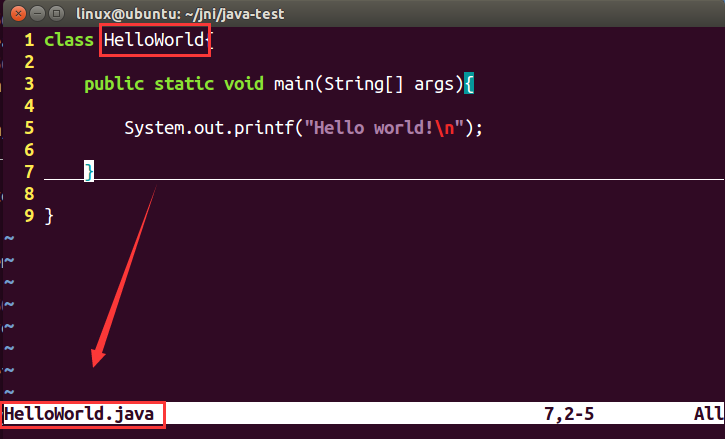
### 编写java程序



1）、虚拟机下编译java程序 javac 文件名（加后缀）

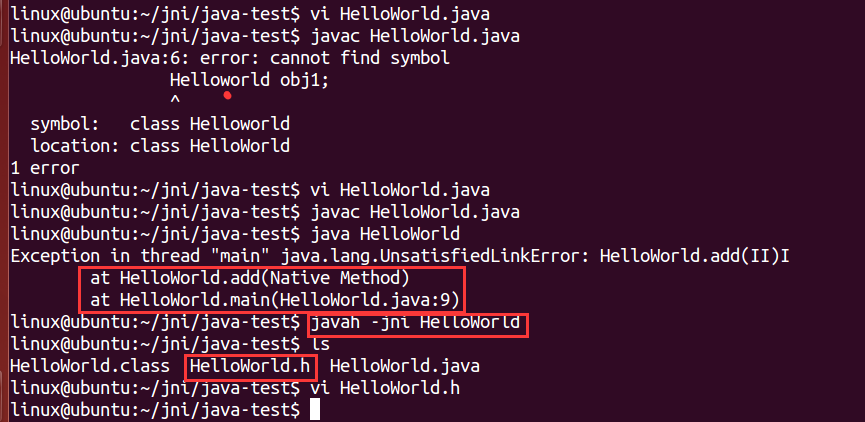
2）、虚拟机下运行java程序 java 文件名（不要后缀）

Java 中包含main函数的类名要与文件名一致，否则编译会报错。

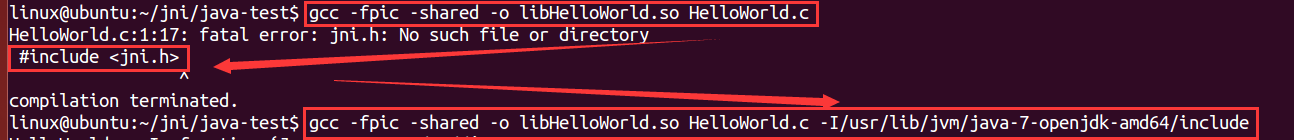


## JNI机制（Java Native Interface）

### JNI技术之静态注册



javah –jni 类名

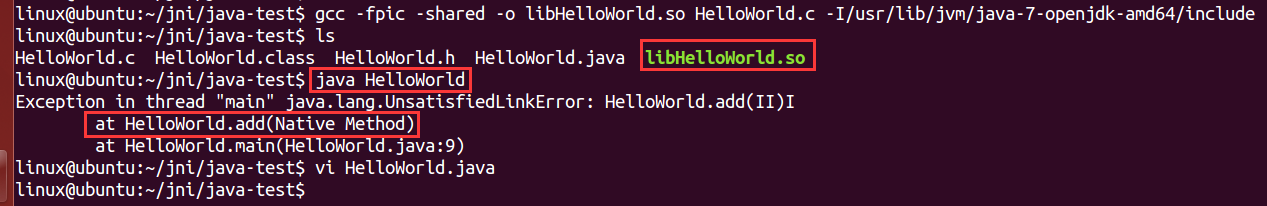


linux@ubuntu:~/jni/java-test$ gcc -fpic -shared -o libHelloWorld.so HelloWorld.c -I/usr/lib/jvm/java-7-openjdk-amd64/include

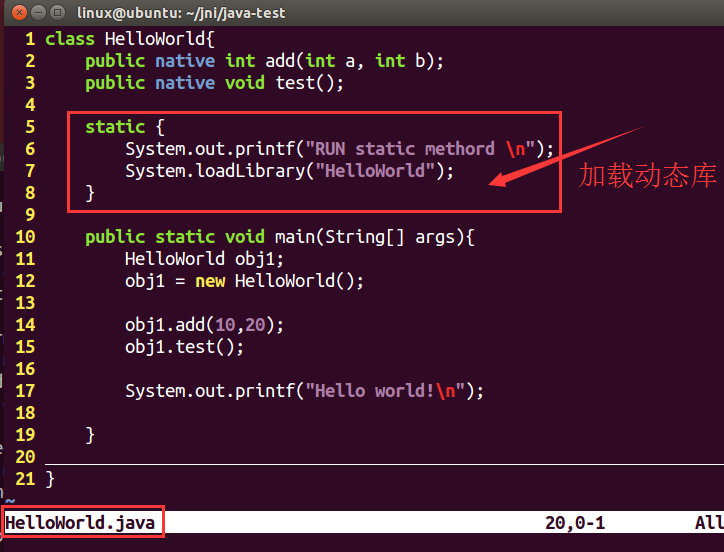


jni.h 头文件所在路径

/usr/lib/jvm/java-7-openjdk-amd64/include



Java程序运行的时候找不到动态库；



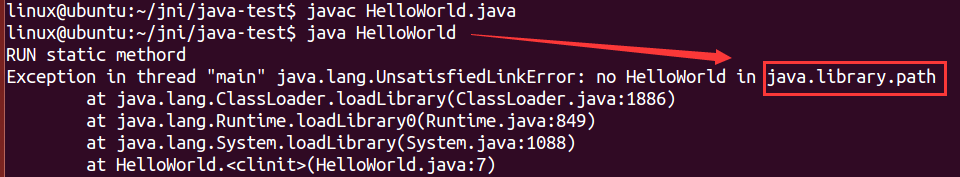
注：java中类定义的对象都是指针，都需要分配空间

static {

System.out.printf("RUN static methord \n");

System.loadLibrary("HelloWorld");

}



提示在java.library.path路径下找不到动态库文件

linux@ubuntu:~/jni/java-test$ java -Djava.library.path=. HelloWorld

