## 1.<stdio.h>可以按照之前的mk文件直接导入。可以参考读取字典过程。

## 2.throw exception编译报错

## 

异常处理被禁用，在mk文件中加 LOCAL\_CPPFLAGS += -fexceptions 详细说明可以参考：

<http://excing.iteye.com/blog/1756612>

## 3.链接报错



链接报错一般都是undefined refrence的错误。

而一般引起这种错误的原因是 1。没有实现方法 2。导入库出错，或者是没导入库

上面一个报错信息是指的exception & throw这些变量未定义。在网上查找，或直接在源码中找lgnustl\_static库。找到后加上 LOCAL\_STATIC\_LIBRARIES := libgnustl\_static

## 4.命名冲突



在libgnustl\_static 和libutils.so中命名冲突，去掉LOCAL\_LDLIBS += 里的libutils.so 则编译通过。部分库的作用可以参考<http://blog.csdn.net/lizhiguo0532/article/details/7219346>。

## 5.static定义函数、变量作用域为本文件可用

## 6.JNI里面C++写env-> C里面写(\*env)->

## 7.使用namespace时记得在函数头前面写如：pinyinime::function()这种格式

## 8.JNI crash info会存储在/data/tombstones里面，错误会定位到汇编文件的某一行

## 9. LOCAL\_AAPT\_FLAGS标签有很多功能，-0的时候可以不将其中的某些文件打包，如：LOCAL\_AAPT\_FLAGS := -0 .dat 会将.dat文件不打包

## 10.有两个dict ：

sys\_dict user\_dict

* sys\_dict（系统词库，没有绝对路径，由以下语句获得资源，实际情况下也是每次开机的时候就在这初始化）：

AssetFileDescriptor afd = getResources().openRawResourceFd(R.raw.dict\_pinyin);

* user\_dict位置是（用户词库，是被生成出来的，代码中有判断，删除也不会影响运行，可以被动态改变）：

mUsr\_dict\_file = getFileStreamPath("usr\_dict.dat").getPath();

## 11.JNI注册的时候对象的写法是：

Ljava/io/FileDescriptor; 注意后面有分号，一定有分号

ERROR：使用fdopen函数读取文件不为空，但开头就是eof。想办法解决。source insight标记处。

## 12.AssetFileDescriptor方式将资源文件传入JNI层注意：

1. 将AssetFileDescriptor对象传入JNI然后获得其jint标签值，然后dup得到int型的文件描述符，用fdopen打开文件描述符，输出之后发现，实际上会将整个res资源文件压缩输出。

代码调用：

java层：

AssetFileDescriptor engAfd = getResources().openRawResourceFd(

R.raw.gendiction);

nativeImOpenEnglishDecoder(engAfd.getFileDescriptor());

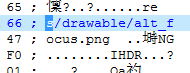
jni层：

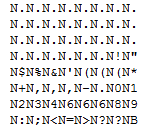
jint fileAdress = env->GetIntField(fd\_sys\_dict, gFileDescriptorOffsets.mDescriptor);

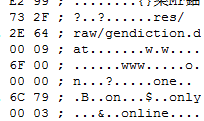
int newFileAdress = dup(fileAdress);

FILE \*filePath = fdopen(newFileAdress, "rb");

文件结构：



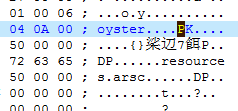




实际上，我们需要的数据只在最后一张图片所示的res/raw/gendiction.dat，但是这个文件描述符实际上把整个资源文件都打开了。

1. Android.mk文件aapt标签标记 -0 不打包文件结果就是文件不会被压缩。如上最后一幅图所示，不然的话aapt会像第二幅图一样压缩这些文件。

如果要获得的话，最好就进行判断从哪开始，而且文件读取长度最好限定好，文件结尾处不易于区分，如下图所示：



## 13.logcat eof的错误可能是由于输出文件的问题

## 14.后期如果要加速运算可以把AssetFileDescriptor里的文件第一次打开后就存一个少量数据的副本在/data/data/app下。

问题：

JNI：

1. 解决输入一个字符第二个字符就出错的问题 （solved）
2. 把当前词添加到第一个候选词里 （solved）
3. 删除一个拼音的时候没有更新（solved）
4. 解决每一页显示个数问题 （solved）

JAVA：

1. 解决点击一个字符之后联想词问题（禁用）（solved）
2. 解决点击一个字符输入到输入框的词不一致的问题 （solved）
3. 解决每一页显示个数问题（solved）
4. 显示与点击事件不匹配问题 （solved but using hard code define offset）
5. 把默认的设为拼音输入法（delay , it should exist）
6. 数据结束还是可以往后滑（delay）

XML：

1. 修改界面到UI需求（solved）
2. log输出PinyinIME.java中的mCandidatesList
3. 查找绘制时字符串从PinyinIme传 与mCandidatesList相同
4. 查找数据源从哪赋值(important)