

一、在 SystemServer.java中添加自己的服务出现错误，解决方案

frameworks/base/services/java/com/android/server/SystemServer.java

```
try {
    Slog.i(TAG, "HelloWord Service");
    ServiceManager.addService("helloworld", new HelloWordService());
} catch (Throwable e) {
    reportWtf("starting HelloWord Service", e);
}
```

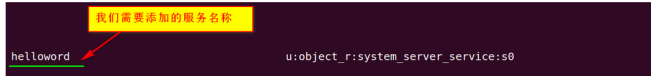
注意:添加完之后，重新生成镜像烧写到平板,在Android 5.x的系统中启动的过程中，会出现如下错误

```
E/SELinux ( 136): avc: denied { add } for service=helloworld scontext=u:r:system_server:s0 tcontext=u:object_r:default_android_service:s0
tclass=service_manager
E/ServiceManager( 136): add_service('helloworld',4d) uid=1000 - PERMISSION DENIED
```

出现这个错误的原因是因为在Android 5.x的系统中增加了SELinux的机制。SELinux的机制，大家可以参考:
<http://blog.csdn.net/luoshengyang/article/details/35392905> 这篇博客中关于Android SELinux的实现机制
http://wenku.baidu.com/link?url=magnHlGyOhzzRLKSs_ykiWMztP3mjD9S-T34Tcwf0QJ2PskR61J9vs1AzsMVqTElhpD9qELviyD4Qxthq4wRHSN-ElDpDR1cIfAMmAuVEa7这是百度文库中的一篇文章，是杂志提供的Android 5.x 权限问题解决方法

看我之后，我们就会知道Android系统为了安全限制了我们自己添加服务，如果想让Android系统允许我们添加的服务，我们只需要在相关的文件中做一下声明就可以了:

在external/sepolicy/service_contexts文件中，按照格式将我们的服务添加进去



添加完成之后执行如下命令:

```
mmm external/sepolicy/service_contexts
make ramdisk-nodexps
make bootimage-nodexps
pack -d 打包成镜像文件，然后烧写到平板中去
```

二、在Android系统中打开硬件设备失败，提示没有权限解决方案

在Android系统中是用ueventd进行创建设备文件的，默认它创建的设备文件只对root用户具有读写权限，其他用户是没有的。而我们Android应用层的APK对应的应用程序很多时候是无法获取root权限的，所以我们需要更改设备文件的权限，让其它用户也能读写它。

我们可以在out/target/product/fspad-733(Andoird系统对应的产品名字)/root/ueventd.rc添加如下内容



添加完之后执行如下命令:

```
make ramdisk-nodexps
make bootimage-nodexps
pack -d 打包成镜像文件，然后烧写到平板中去
```

注意:pack -d 是我们自己定义的命令，不是android 编译系统自带的命令，请根据自己的平台打包生成镜像的方法打包Android镜像

用adb shell登陆之后，查看设备文件的权限如下:

```
crwxrwxrwx root root 500, 0 2016-03-30 10:27 led
```

本以为万事大吉了，但是我们现在是在Android 5.x系统中，还有变态的SELinux机制，就算有权限了此时通过JNI层调用到HAL层打开设备文件仍然失败.提示“Permission denied ”。解决的问题方法是，我们需要在和SELinux相关的配置文件中声明访问设备文件的进程具有对设备文件操作的权限。

我们是在SystemServer中添加了服务,添加服务的时候，我们创建了一个服务对象(new HelloWordService()).创建对象的时候会调用构造函数(HelloWordService),接着我们在构造函数中通过JNI的方式调用到HAL层的open函数打开我们的LED设备的。说这一段的的话的目的是想告诉大家我们实际上是在SystemServer对应的进程中访问设备文件的。

SystemServer对应的进程名字是system_server,它对应的安全配置文件是:external/sepolicy/system_server.te,修改如下:



```
make ramdisk-nodexps
make bootimage-nodexps
pack -d 打包成镜像文件，然后烧写到平板中去
```