Process 2 - test

RSDLite

Process 1 - RSDLite.jar

SDLFrame

UIEventHandler

SDLServer

SDLApp

Main()

Socket (localhost:3501)

CSDLClient

Thread1: ProcMsgHandlerThrd

MsgQueue: g\_nPRCQueueID

Thread2: FWMsgHandlerThrd

FlashThread

Process 3 - Detection

PST

MsgQueue: g\_nFWQueueID

Thread4: UdevMessageRcv

Thread3: CPSTSession::DeviceChangeNotice

queue<DeviceData\*>

m\_DeviceQue

Kernel hotplug will trigger process 3 when device arrive or removed.

MsgQueue: QUEUE\_ID

程序启动时，系统里在用户态最多同时有三个相关**进程**在运行。

进程1：RSDLite.jar ，java程序，负责GUI显示部分。

进程2：test，RSDLite的主要执行体。内部从模块上分，可以分成三大部分。

* test模块，主要由CSDLClient类构成，作为C的主进程启动，并负责和RSDLite.jar的Socket通讯部分。
* RSDLite模块，主要由libRSDLite.so(包含Flash库),libFlex.so。内部固定起两个线程，一个是Thread1 - ProcMsgHandlerThrd，其作用是和刷机时对应每个手机启动的flash线程通讯，接收flash线程发送的消息。还有一个固定的线程是Thread2 - FWMsgHandlerThrd，负责和PST模块通讯，接收PST模块的设备检测消息。
* PST模块，即libPST.so，内部会固定启动两个线程，其中线程Thread4 - UdevMessageRcv负责接收Detection进程的设备到达和移除消息，UdevMessageRcv得到消息后会将消息结构放在内部定义的一个队列m\_DeviceQue里，另外一个固定线程Thread3 - CPSTSession::DeviceChangeNotice负责从m\_DeviceQue里读取后进行实际的处理。

进程3：Detection，该进程不是一直运行的，而是由kernel的热插拔系统在设备到达或者移除时才会被调用一下，向系统队列QUEUE\_ID投递消息。

程序启动时 ，系统里共创建了三个系统级的消息队列，用于线程间和进程间的通讯。

* 队列1：g\_nPRCQueueID，用于Thread1 – ProcMsgHandlerThrd和对应每个设备的FlashThread之间的通讯，FlashThread可以有多个。该队列由RSDLite模块创建。
* 队列2：g\_nFWQueueID，用于Thread2 – FWMsgHandlerThrd和Thread3 - CPSTSession::DeviceChangeNotice之间的通讯，该队列由PST模块创建。
* 队列3：QUEUE\_ID，用于进程3和进程2的Thread4 – UdevMessageRcv之间的通讯，该队列由PST模块创建。

进程1和进程2之间通过系统Socket(localhost:3501)进行通讯。