

守护升级进程设计方案

一、设计原则

守护进程是整个系统的核心进程，负责系统中其它任务的启动、停止、检测其它进程以及设备升级维护等功能。由上述的特点，要求程序必须简洁、稳定、易拓展、移植性高。

二、程序功能

功能划分 3 部分，分别为：确定要守护的进程；初始化守护进程；启动设备升级维护线程。

确定要守护的进程：确定要守护的进程，通过读取系统的配置文件，获取当前设备要启动的进程。如果没有配置文件，则启动系统默认的进程。

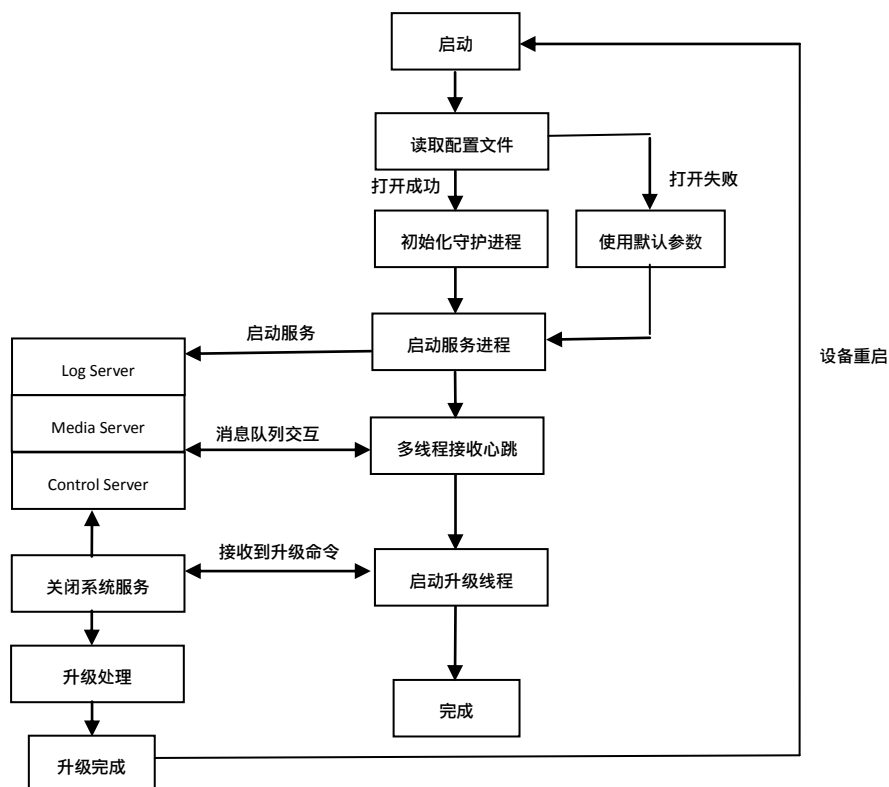
初始化守护进程：根据第一步获取的配置信息，创建多个线程，分别检测要守护的进程，启动要守护的进程。暂定守护进程和要守护进程之间交互，方式为消息队列。

启动设备升级维护线程：设备升级实现不同分区的单独升级，主要包括，boot、kernel、rootfs、application 四部分。目前暂定使用 socket 方式接收升级文件，然后分别对不同分区进行升级。

三、核心技术

- (1) 守护进程的 Debug 日志。
- (2) 心跳机制，使用 RUDP 机制。
- (3) 设备升级，使用 socket 机制。

四、流程图



五、守护进程主要接口

(1) 初始化心跳连接

```
/*=====
name           :    GMI_DaemonInit
function       :    Init Heardbeat .Get Application Socket Handle
algorithm implementation :    no
global variable :    no
parameter declaration :    pDaemonData : Daemon struct data;
                                Appld : System server Id;
                                ServerPort: daemon  server Port;
                                LocalPort: System  server Port;

return         :    FD_HANDLE: Application Socket handle
*****/

GMI_RESULT GMI_DaemonInit(DAEMON_DATA_CFG *pDaemonData, int32_t Appld, long_t ServerPort, long_t
LocalPort);
```

(2) 反初始化, 注销退出

```
/*=====
name           :    UnInitHeardbeat
function       :    Delete Application Handle
algorithm implementation :    no
global variable :    no
parameter declaration :    pDaemonData : Daemon struct data;

return         :    no
*****/

void GMI_DaemonUnInit(DAEMON_DATA_CFG *pDaemonData);
```

(3) client 向守护进程注册

```
/*=====
name           :    GMI_DaemonRegister
function       :    Register Application to Daemon Server
algorithm implementation :    no
global variable :    no
parameter declaration :    pDaemonData : Daemon struct data;
return         :    0:success
                                fail:  ERROR CODE
*****/

GMI_RESULT GMI_DaemonRegister(DAEMON_DATA_CFG *pDaemonData);
```

(4) 发送心跳连接给守护进程

```
/*=====
name          :    GMI_DaemonReport
function       :    Report Heardbeat to Daemon Server
algorithm implementation :    no
global variable      :    no
parameter declaration :    pDaemonData : Daemon struct data;
return          :    0:success
                                fail:  ERROR CODE
*****/
GMI_RESULT GMI_DaemonReport(DAEMON_DATA_CFG *pDaemonData);
```