# 需求分析和总体设计

## 软件部分的关键技术分析

## 项目的实现目标

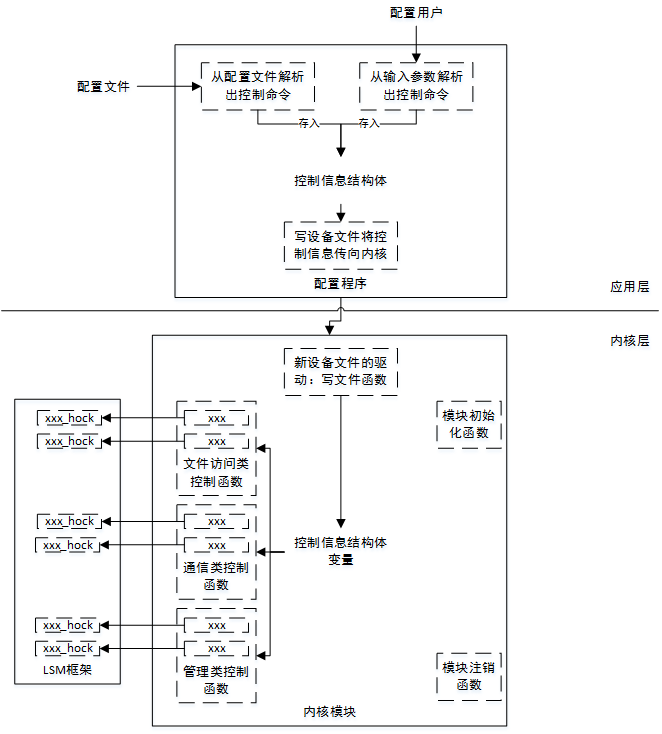
### 权限配置程序

1. 权限的配置操作（包括权限的查看、添加、删除、修改）
2. 权限的配置文件操作（包括权限的配置文件的导入、导出）

### 程序运行权限管理

1. 文件访问类操作
2. 各类文件（或目录）的创建
3. 各类文件（或目录）的打开
4. 各类文件（或目录）的读
5. 各类文件（或目录）的写
6. 各类文件（或目录）的执行
7. 各类文件（或目录）的删除
8. 通信类操作：以实现文件访问类为主，该类尽可能实现网络类的一些操作
9. 管理类操作：以实现文件访问类为主，该类尽可能实现关机、重启操作

## 项目总体设计



### 项目的模块划分

1. 配置程序：负责从用户输入或配置文件中得到控制信息，并转换为一个结构体，并将该结构体通过写新的设备文件将控制信息从用户态传到核心态。
2. 新设备文件的驱动：完成应用程序向内核模块发送配置信息，接收配置程序写入的控制信息，然后将该内容保存在控制信息结构体中。
3. 辅助模块：完成一些辅助功能，如根据PID得到实际的程序完整路径。
4. 一组访问控制函数：根据钩子函数传入的参数信息与辅助模块传入的进程信息，解析出哪个程序要执行什么操作，然后与控制信息结构体中的内容进行比对，如果要禁止则返回1否则返回0.
5. 内核模块的初始化：完成两方面的初始化工作，一是注册新设备文件的驱动程序，二是将预先设计好的相关控制函数注册到LSM框架中。
6. 内核模块的注销：完成设备文件驱动的卸载，以及从LSM框架中注销所注册的钩子函数。

### 模块之间的接口

1. 配置程序模块与新设备文件的驱动模块之间使用write\_controlledinfo，该函数是新注册设备的一个驱动函数，用于处理设备文件的写操作，其主要任务是调用函数copy\_from\_user()，将来自应用层空间的缓冲区中的内容复制到内核层空间的缓冲区。
2. 一组访问控制函数模块与辅助模块之间使用get\_fullpath，该函数从参数pid寻找该进程对应的程序，并返回该程序所在的全路径名。
3. 一组访问控制函数模块与辅助模块之间使用myown\_check，该函数检查进程所对应的程序所执行的操作是否被允许。