该框架为CRSCD新一代计算机联锁软件提供核心调度机制。该机制实现联锁逻辑的动态配置，使得在需求变化时，使用动态注册机制避免代码的更改。

一、具体功能：

1. 功能配置模板设计与加载：该模板完成联锁逻辑配置。功能配置模板特点如下：
   1. 层次排列，即并列标识几种进路类型。
   2. 顺序排列，即在每个层次内，各功能为顺序执行。
   3. 可重入，即代码将记录当前运行状态，选择顶层标识入口。
   4. 功能配置模板目前采用关键字标识，或XML实现。
2. 注册机制实现：实现对于模板功能配置关键字注册与可执行代码识别。
   1. 关键字-入口成对出现排列。
3. 调度模块功能要求
   1. 在程序初始阶段，功能配置模板，在内存中形成功能模板数据。
   2. 加载注册表，在内存中形成注册表。
   3. 从固定内存中获取当前系统记录的运行状态。
   4. 查找功能配置模板，查找到待执行函数。
   5. 查找注册机制，查找待执行函数入口并调用。
   6. 继续查找功能配置模板直到结束。

二、详细设计：

1. FunctionTemplate模块：负责函数模板的生成、解析与校验函数模板以及对功能模板的查询、访问。
   1. Sys\_LoadTemplate 函数：

输入char[]，文件路径字符串filepath,输出：BOOL\_STATUS，是否加载正确。

功能：完成配置文件路径加载，并解析到全局变量g\_functionTemplate中。

逻辑流程：打开文件，按字符读取，对于每个字符，判断几个终止符是否出现，当出现时，将字符拼成字符串，进行词法分析。

词法分析思路：词法分析关键字为PATH,ITEM,ENDPATH,内部设置状态机，path分析器状态机为PATH，PATH\_CLEAR，ITEM分析状态机为ITEM,ITEM\_CLEAR。状态机要求，

PATH\_CLEAR,ITEM\_CLEAR下，当且仅当解析出PATH，进入PATH，ITEM\_CLEAR状态；

PATH,ITEM\_CLEAR下，若出现ITEM，进入ITEM状态；若出现ENDPATH，进入PATH\_CLEAR, ITEM\_CLEAR状态，其他报错。

PATH，ITEM下，若出现关键字，进入PATH,ITEM\_CLEAR状态；其他报错。

禁止出现：PATH\_CLEAR,ITEM状态。

解析结束后，功能模板存储于全局变量g\_functionTemplate当中，并返回BOOL\_TRUE，否则为BOOL\_FALSE。

* 1. Sys\_getNextPath函数，获得下一条路径，

输入：status，当前路径名称，char\* pNextpath，下一条路径的返回地址

输出：成功获得为BOOL\_TRUE,若失败或无下一条路径，返回BOOL\_FALSE;

流程：若当前g\_functionTemplate为空，返回失败

当当前路径为begin初始时，返回第一条路径。

否则，遍历所有可能的路径名称，当匹配status时，检查此时是否为最后一条路径，如果是，返回失败，否则，返回下一条路径。

* 1. Sys\_getNextItem函数，获得下一个项目

输入：status，当前路径，itemstatus,当前item,pNextItem，返回路径

输出：BOOL\_STATUS

流程：遍历所有path，找到匹配的，否则返回失败

在匹配的path中，获得item的数目；

如果数目为0，返回失败；

若为”begin”，则返回第一个item

找到匹配itemstatus的项目，如果成功，若找到的为该path最后一个，返回失败，否则，返回下一个item.

* 1. Sys\_getNumOfPath函数，返回路径个数。

输入：无

输出：路径个数最少为0

* 1. Sys\_getNumOfItem 函数，返回特定路径下item个数

输入：当前路径名称

输出：item个数，如果无匹配，返回-1

1. RegisterTable模块，核心模块，完成匹配命令与函数入口之间的映射。

三、用户使用方法：

1. 在RegisterTable.h中加入DECLARE\_FUNC(函数名)，完成函数声明，并在相应的实现文件中实现该函数，函数的返回值为BOOL\_STATUS，无参数。

2. 在Scheduler.c中的Sys\_Initial()中，通过IMPLEMENT\_FUNC(注册名,&函数名)完成函数注册，其中，注册名要与配置文件中的使用的关键字一直，注册名为真实函数名称。

3. 在Configuration.txt中实现逻辑。