

进程管理项目——电梯调度 设计文档

1953608 吴英豪

指导老师：张慧娟老师

进程管理项目——电梯调度 设计文档

一、调度算法

1. 算法介绍

2. 数据结构及相关参数

二、使用说明

1. 按键及页面介绍

2. 使用说明

三、开发环境

一、调度算法

1. 算法介绍

(1) 思路分析：

- 每一部电梯均有以下的属性：当前运行状态（空闲、上行、下行，int），报警情况（是否损坏，bool），是否需要切换运行状态（bool）。
- 使用Qt内置的QTimer类作为定时器，使用线程的思想，定时监听每一部电梯的运行状态，并根据当前运行情况（电梯上行、下行，按楼层按钮，按报警按钮等）使用调度算法实现电梯调度。

(2) 调度算法设计：

- 正常 <--> 报警

所有电梯初始状态下均为可运行状态。进行调度时首先判断电梯损坏情况，若已经损坏（按下报警键）则不调度此电梯，若恢复正常运行（再次按下报警键）则考虑调度此电梯。

```
if(!_isDamage[0]) //若0号电梯损坏，直接返回
    return;
```

- **运行（上行、下行）-->运行（上行、下行）**

- a. 当电梯为上行状态时，若需要到达更高的楼层，则当前运行状态保持不变（上行）；若已经到达顶层（20楼），则切换运行方向为下行。
- b. 当电梯为下行状态时，若需要到达更低的楼层，则当前运行状态保持不变（下行）；若已经到达底层（1楼），则切换运行方向为上行。

```
//以上行->上、下行为例
floor=min(floor+1,FLOORS);
if(floor==FLOORS) //FLOORS = 20
{
    _elevator[0]->setStatus(DOWN); //由上升变成下降
}
```

- **运行（上行、下行）-->停止（空闲）**

电梯从运行状态上行转换为停止状态（空闲）需要同时满足以下三点：

- a. 没有按下当前楼层以上直至顶楼的电梯内部按钮
- b. 没有按下当前楼层以上直至顶楼的电梯外部上行按钮
- c. 需要切换运行状态，没有按下当前楼层以上直至顶楼的电梯外部下行按钮

同理，电梯从下行状态转换为停止状态（空闲）需要同时满足以下三点：

- a. 没有按下当前楼层以下直至一楼的电梯内部按钮
- b. 没有按下当前楼层以下直至一楼的电梯外部下行按钮
- c. 需要切换运行状态，没有按下当前楼层以下直至一楼的电梯外部上行按钮

- **停止（空闲）-->运行（上行、下行）**

不妨做如下定义：

- a. 定义系统使电梯从停止（空闲）到运行状态（上行、下行）的过程为“唤醒”。
- b. 若唤醒正在运行的电梯（上行或下行），不改变当前电梯的运行状态。

则唤醒空闲电梯需要考虑以下几个方面：

- **电梯内部按钮唤醒（空闲-->上行）**

当按下电梯内部等于或高于电梯停靠楼层的按钮时，电梯切换为上行。

- **电梯内部按钮唤醒（空闲-->下行）**

当按下电梯外部低于电梯停靠楼层的按钮时，电梯切换为下行。

（PS..此处优先考虑电梯内部按钮唤醒电梯的情况）

- 电梯外部按钮唤醒（空闲-->上行）

需要唤醒以下几种电梯中距离按下按钮楼层最近的一个电梯：

- i. 上行状态且低于按钮楼层的电梯
- ii. 空闲且低于按钮楼层的电梯，运行状态:空闲->上行
- iii. 空闲且高于按钮楼层的电梯，运行状态：空闲->上行，需要切换运行状态

- 电梯内部按钮唤醒（空闲-->下行）

同理，需要唤醒以下几种电梯中距离按下按钮楼层最近的一个电梯：

- i. 下行状态且高于按钮楼层的电梯
- ii. 空闲且高于按钮楼层的电梯，运行状态:空闲->下行
- iii. 空闲且低于按钮楼层的电梯，运行状态：空闲->下行，需要切换运行状态

2. 数据结构及相关参数

（1）相关参数

```
#define UP 1      //上行状态
#define DOWN 2    //下行状态
#define FREE 3    //空闲（停止状态）
#define ELEVATOR_NUM 5 //电梯总数
#define MAX_FLOORS 21 //最大楼层数，选择21是为了使数组下标与楼层对齐

bool _extend; //是否需要切换运行状态 false-
              //不切换 true--切换
```

（2）相关数据结构

```
//将电梯封装为一个类
//1. 使用数组存储5个电梯
Container* _elevator[ELEVATOR_NUM]; //5部电梯

//2. 对每一个电梯，用bool数组存储电梯内部的20个按钮
//true表示电梯内对应楼层按钮被按下，false表示没有按下按钮
bool _Floors[MAX_FLOORS]; //需要到达的楼层

//3. 使用bool数组存储电梯外部的上下行按钮
//true表示对应楼层的上（下）行按钮被按下，false表示没有按下按钮
bool _upwaitFloors[MAX_FLOORS]; //上升等待队列
bool _downwaitFloors[MAX_FLOORS]; //下降等待队列
```

//4. 使用bool数组存储电梯是否损坏

//true表示损坏(按下报警键)、false表示修复(再次按下报警键)

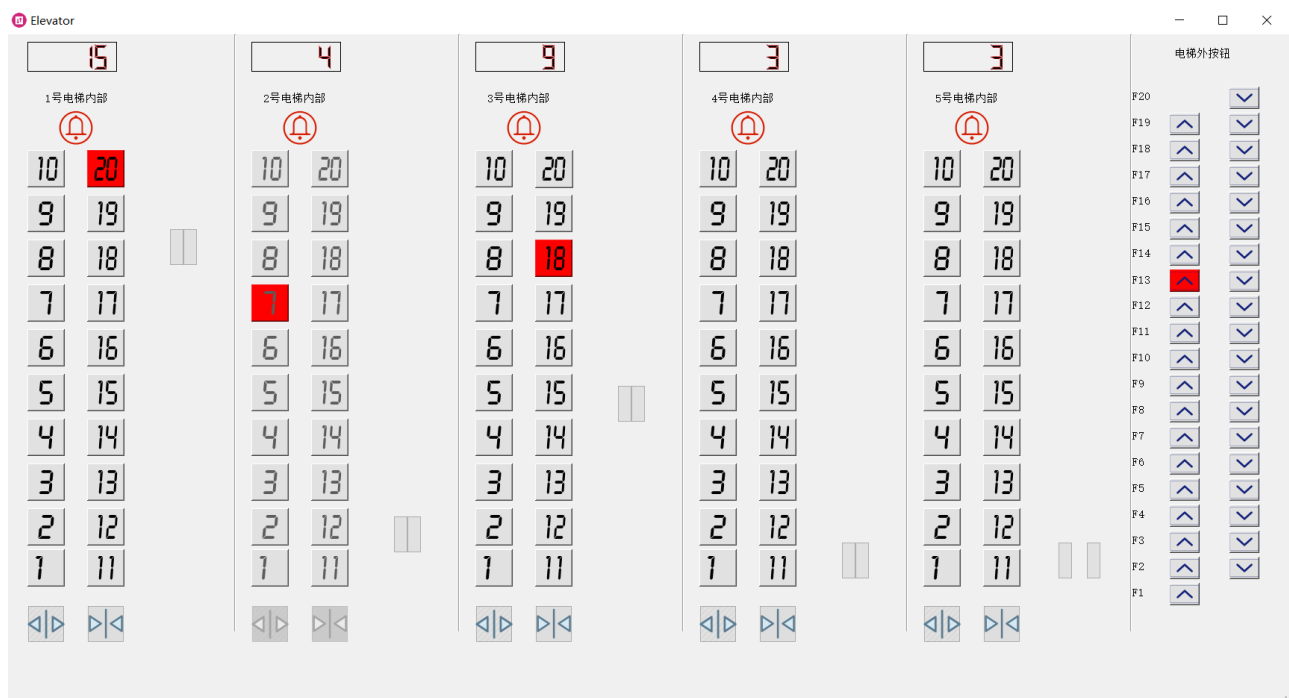
bool _isDamage[ELEVATOR_NUM]{false}; //表示5部电梯的损坏情况

二、使用说明

具体实现结果请见压缩包中的.exe可执行文件和源代码文件夹

1. 按键及页面介绍

- 五部电梯均使用竖线隔开，每一部电梯内部均有20个按钮，表示20层楼。
- 电梯内部按钮上方是警报按钮，第一次按下警报键电梯将暂停运行（此时不能按该电梯的任何按钮），再一次按下报警键电梯将恢复运行。
- 电梯内部下方是开关门键，只有当电梯停靠或空闲时按下开门键才能开门；只有电梯的门为开时才能关门。
- 页面右侧为电梯外每一层楼的按键，分为上行按键和下行按键（其中一楼只有上行键、二十楼只有下行键）。



2. 使用说明

- 初始情况下所有的电梯均停靠在1楼，当有用户在电梯外按下上行、下行键之后，会有电梯进行响应并运行指定楼层；电梯到达后将会开门，用户进入电梯，按下电梯内的按钮，电梯运行到指定楼层，随后开电梯门，用户出电梯，若无其他调度电梯停靠在当前楼层。

- 电梯开门会持续一段时间，可以按电梯内部关门键关门，若不按下关门键，电梯将自动关门。
- 时间设计：

电梯每上（下）一层楼需要0.85秒的时间。在不按关门按钮的情况下，电梯到达指定楼层停靠的时长为4秒。

三、开发环境

- 开发环境：Windows 10
- 开发软件：qt-opensource-windows-x86-5.14.2（QtCreator）
- 开发语言：C++、XML