操作系统进程管理项目

——电梯模拟系统

1652670_齐旭晨

目录

一、 项目概述

1. 基本任务

a) 某一层楼 20 层，有五部互联的电梯。基于线程思想，编写一个电梯调度程序。

2. 功能描述

a) 电梯应有一些按键，如：数字键、关门键、开门键、上行键、下行键、报警键等
b) 有数码显示器指示当前电梯状态；
c) 每层楼、每部电梯门口，有上行、下行按钮、数码显示。
d) 五部电梯相互联结，即当一个电梯按钮按下去时，其它电梯相应按钮同时点亮，表示也按下去了。

3. 电梯调度算法

a) 所有电梯初始状态都在第一层；
b) 每个电梯没有相应请求情况下，则应该在原地保持不动；
c) 电梯调度算法自行设计。

4. 需求分析

a) 当一部电梯接收到请求后需要合理安排路线，处理请求；同时实现警报、修复、开门、关门的操作
b) 用户交互界面接收到用户的请求信号后，需要将信息反馈给任务分配器
c) 分配器接收任务后择优将请求分别分配给电梯

5. 具体实现

a) 电梯能接受到的命令分为两类

二、 开发环境

1. **IDE**
   Microsoft Visual Studio Community 2017

2. **编程语言**
   C#

3. **版本控制**
   Github

三、 源码说明

1. **Elevator 类**

   a) **变量一览：**

变量名	变量类型	变量功能
eStatus	public int	标识电梯状态 1：向上运行 2：停靠 3：向下运行
toDeal	public int	存储待处理请求数
WhetherStop	public int	标识是否需要延长或减少停靠时间
currentFloor	public int	存储电梯当前所处楼层
inTarget	public int [20]	存储电梯内部请求楼层
outTarget	public int [2] [20]	存储发出了外部请求的楼层，[0][x]表示 x 层向上请求 ;[1][x]表示 x 层向下请求
inLightStatus	public int [20]	标识电梯内部按钮的状态（是否需要亮灯）

   b) **核心功能**
   模拟单部电梯，存储各项基本信息

2. **ElebatorController 类**

   a) **变量一览**

ifNormal	bool	标记当前电梯是否能正常工作
eControl	Elevator	绑定电梯类

elevatorText	TextBox	绑定可视化界面中的电梯模块

b) **主要功能：**

电梯控制器，每台电梯都对应一个 ElevatorController 控制器对象；
接收主界面类传递来的信息；
在该类中完成电梯的内部调度、方向设立、可视化电梯模块移动等重要功能。

c) **Run (object t) 函数**

主要功能：

调用 ElevatorController 类中的大部分函数，以此完成电梯的运行、停靠、可视化移动、出现故障、维修等重要功能

功能详解：

当变量 ifNormal 为 true 时表示电梯运行正常，此时调用 SetTarget ()函数，为电梯找出当前的目标。

以函数 WhetherStop () 作为判断条件，如果当前抵达楼层为当前目的地楼层，就做停靠，并调用相关函数，完成停靠动作。不是目的地就继续运行

d) **SetTarget ()**

主要功能：

根据当前的运动状态与待处理请求的楼层和类型作为条件，分配最优路线，并存储最优目标为 currentTarget

调用 SetDirection（）函数，设置当前电梯方向状态

e) **功能详解**

如果电梯向上运行，就在当前目标与当前楼层之间由上至下检索，如果其中某楼层有内部请求或外部向上的请求，就将当前目标更新为该楼层

如果电梯向下运行，就在当前目标与当前楼层之间由下至上检索，如有内部请求或外部向下的请求，就更新当前目标

如果电梯处于停靠状态，就在当前楼层与顶层之间搜索最近的内部或外部向上请求并记录距离。如无请求，则将距离变量保持为初始值 999。然后在当前楼层与底层之间搜索最近的内部或外部向下请求并记录距离，如无，同样将距离保持为 999。检索之后比较向上或向下的距离，择近设立 currentTarget。

调用 SetDirection() 函数，由此改变当前运行方向，详情见下。

f) **Move() 函数**

主要功能

根据当前的运动状态信息，控制绑定当前电梯的 TextBox 控件在主界面移动

实时更新当前电梯楼层

g) **WhetherStop() 函数**

主要功能

判断当前所处楼层是否为当前的目标楼层

h) ElevatorStop() 函数

主要功能

标识当前请求为已完成状态

更新数据

i) ErrorCallBack() 函数

主要功能

显示报错层 Label

修改变量 ifNormal 以停用电梯

j) RepairCallBack() 函数

主要功能

隐藏报错层

修改变量 ifNormal 以重新启动电梯

3. 主界面类：MainWindow()

	变量名	变量类型	变量功能
prvate	toAssigh	int	存储仍未被分配的请求数量
public	currentElevator	Int	标识当前控制的电梯是哪一部
	timer	DispatchTimer	计时器
	elevator	Elevator[5]	存放电梯控制器实例
	ourRequestStatus	Int[40]	存放当前的外部电梯状态

a) 核心函数

i. Button_Click 一类

Button_Click_inRequest 用于响应电梯内部楼层按键

Button_Click_outRequest 用于响应外部的上下按键

Button_Click_Error 响应故障按键

Button_Click_Repair 响应修复按键

Button_Click_Open 响应开门按键

Button_Click_Close 响应关门按键

ii. 请求响应

具体思路：

内部按键直接修改 Elevator.Controller.eContol.inTarget[]数组

外部请求分向上按键和向下按键进行调度：对于向上按键，如在其下的楼层找到了

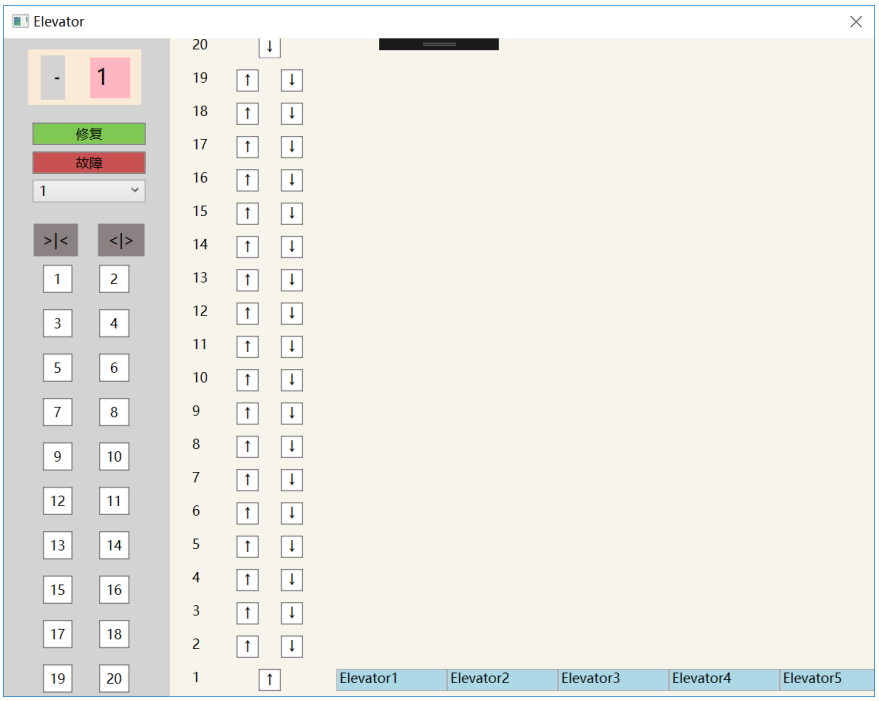
向上的电梯, 则直接分配, 如无则在停靠电梯中找到最近的一部 ;向下的按键同理。

四、 多线程实现

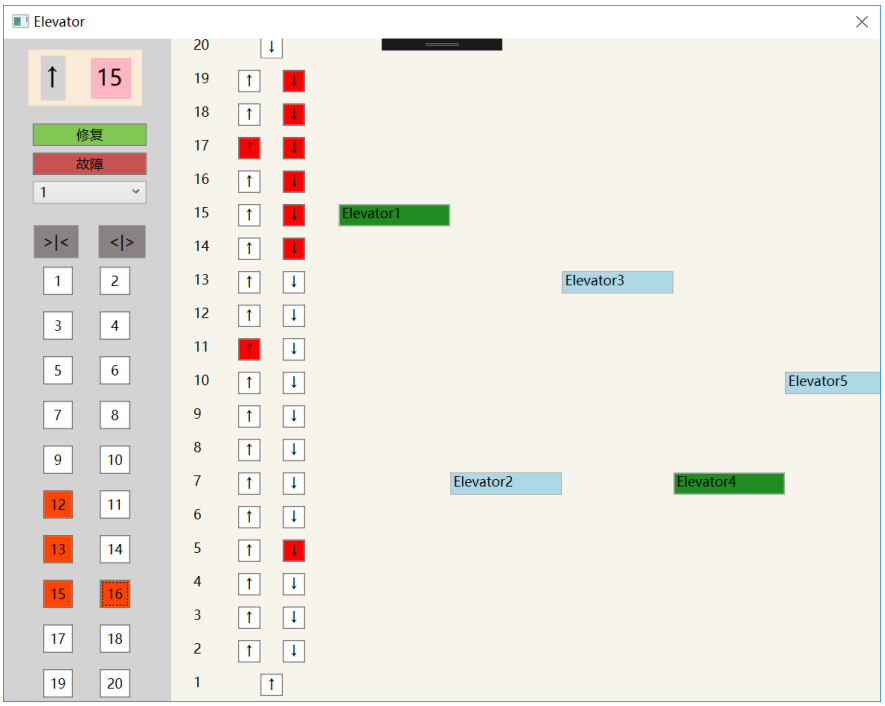
分别给五部电梯创建五个线程，与主线程同步进行，

五、 运行实例

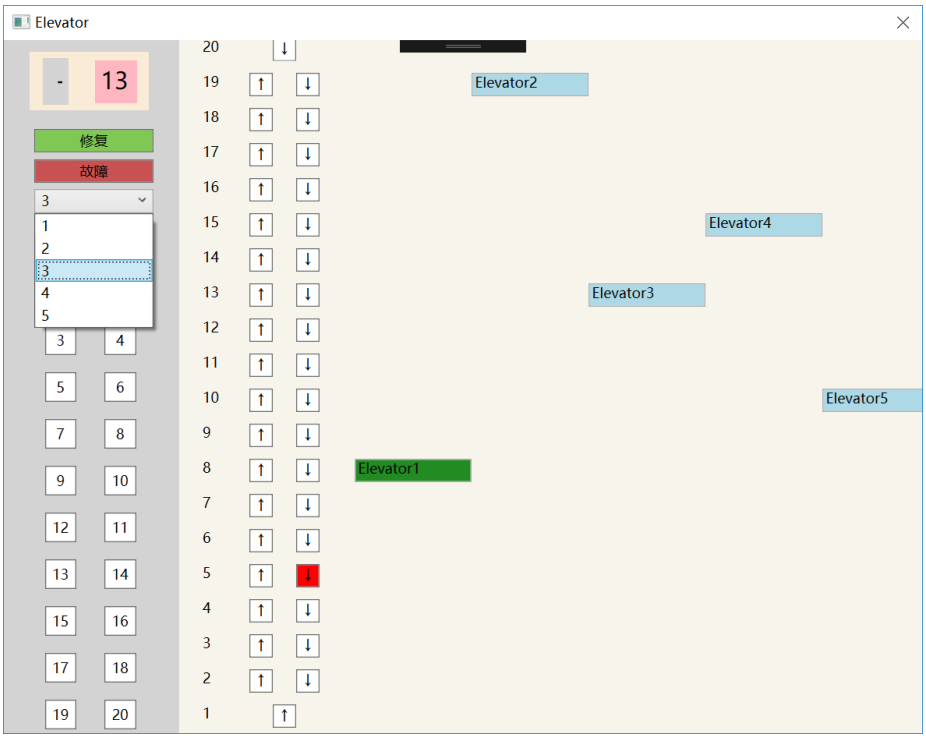
a) 初始界面



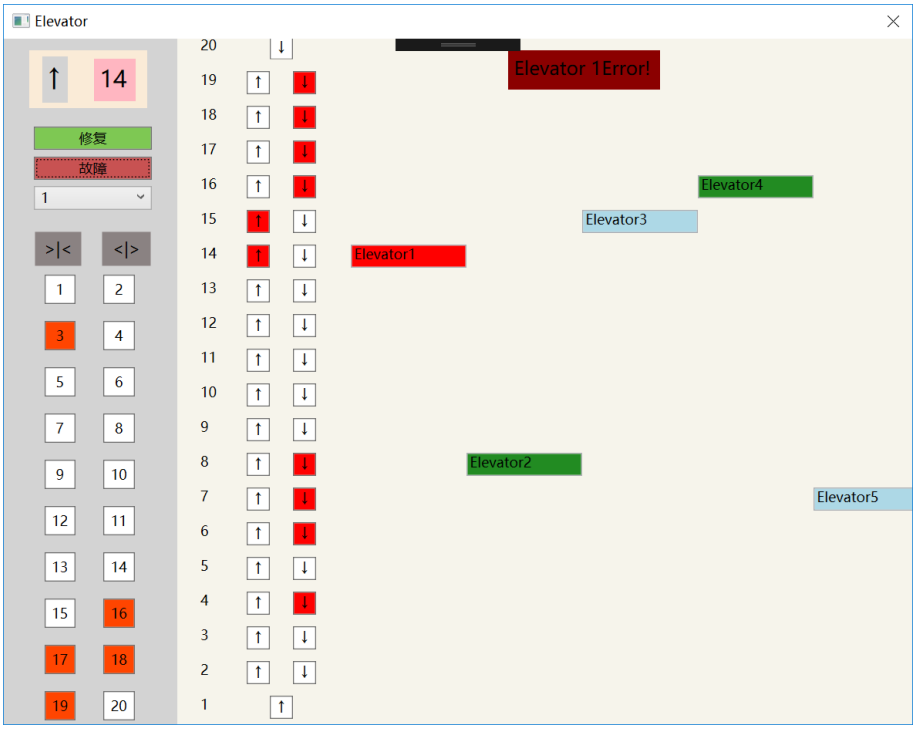
b) 运行界面



c) 电梯切换


Elevator
-
13
修复
故障
3
1
2
3
4
5
Elevator2
Elevator4
Elevator3
Elevator5
Elevator1

d) 故障界面


Elevator
↑
14
修复
故障
1
>|<
<|>
Elevator 1Error!
Elevator4
Elevator3
Elevator1
Elevator2
Elevator5