同濟大學
TONGJI UNIVERSITY

进程管理项目——电梯调度设计方案

姓　　名：叶栩冰

学　　号：1953348

所 在 院 系：软件学院

任 课 教 师：张慧娟

二〇二一年五月

题目概述

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。其中，电梯包含以下功能：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；有数码显示器指示当前电梯状态；每层楼、每部电梯门口，有上行、下行按钮、数码显示。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。所有电梯初始状态都在第一层；每个电梯没有相应请求情况下，则应该在原地保持不动；并且需要自行设计电梯调度算法。设计项目要包含界面、算法、可行性等等。

开发环境

由于要对模拟电梯调度的多线程思想，基于进行电梯调度算法的设计，并实现 UI 界面来提高电梯调度算法的面向用户界。本项目选取 python 语言编写，利用 python 的 pyqt5 来实现 UI 用户交互，并利用 python 的 threading 线程与 qt 线程来实现电梯调度的模拟与算法设计。

项目架构设计

本项目在设计时将整体程序架构设计为三个主要的工具类，分别为电梯类(Elevator)、电梯集类(GroupOfElevator)和 UI 交互类(Ui_MainWindow)，他们分别实现单个电梯状态模拟、电梯之间的调度安排与乘客管理、和面向 UI 展示交互这三个主要功能。

下面为整体架构函数的成员项与意义：

Elevator

- name : 电梯名称
- status : 电梯运行状态(1 为上行，-1 为下行，0 为静止)
- power : 电梯电源状态
- floor : 电梯目前所处楼层

- elevator_num : 电梯编号 1-5
- door_open_button : 电梯开门按钮状态
- warning_status : 电梯报警按钮状态
- door_status : 电梯门状态
- stop_times : 电梯停留剩余时间
- target_status : 电梯目标接客楼层的目标去向(上行或下行)

Elevator 类为电梯运行状态类，这其中仅仅包含单个电梯在接受调度与显示状态时会用到的电梯状态属性，不包括任何函数，这是一个能够显示电梯楼层、运行方向、目标楼层、停止时间、目标楼层乘客的目标方向等信息的属性类，此属性类供 GroupOfElevator(线程类)进行调用来模拟多个电梯的并行调度。

GroupOfElevator(线程类)

- e1 = Elevator(1) : 一号电梯
- e2 = Elevator(2) : 二号电梯
- e3 = Elevator(3) : 三号电梯
- e4 = Elevator(4) : 四号电梯
- e5 = Elevator(5) : 五号电梯
- ele_task = [[] for i in range(5)] : 电梯任务楼层队列
- inside_task = [[] for i in range(5)] : 电梯内部目标楼层
- ele_group_task = [[0] * 20, [0] * 20] : 电梯外部按钮状态
- button_test = [[0] * 20, [0] * 20] : 电梯按钮显示队列
- ele = [self.e1, self.e2, self.e3, self.e4, self.e5] : 电梯组队列
- warning() : 警报函数
- step() : 模拟电梯线程函数
- tell_inside_button() : 内部电梯按钮判断函数
- arrange() : 外部电梯调度函数
- schedule() : 电梯状态与内部数组更新函数
- run() : 运行主函数

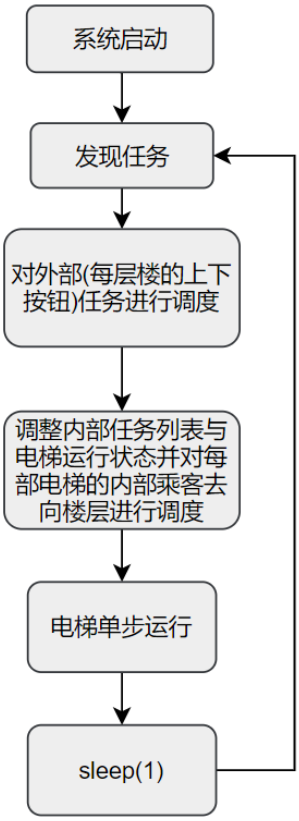
GroupOfElevator(线程类)是电梯集类，该类包含了电梯内外部目标队列、无不电

梯的目标停留楼层队列，并且负责调度电梯、更改内外部电梯数组、更行电梯运行列表与运行状态，此外，他还是一个线程类，该类的实体在 UI 类中构建，与 UI 的交互控制并行，依次来实现电梯的实时调度与任务分配。

电梯调度算法设计

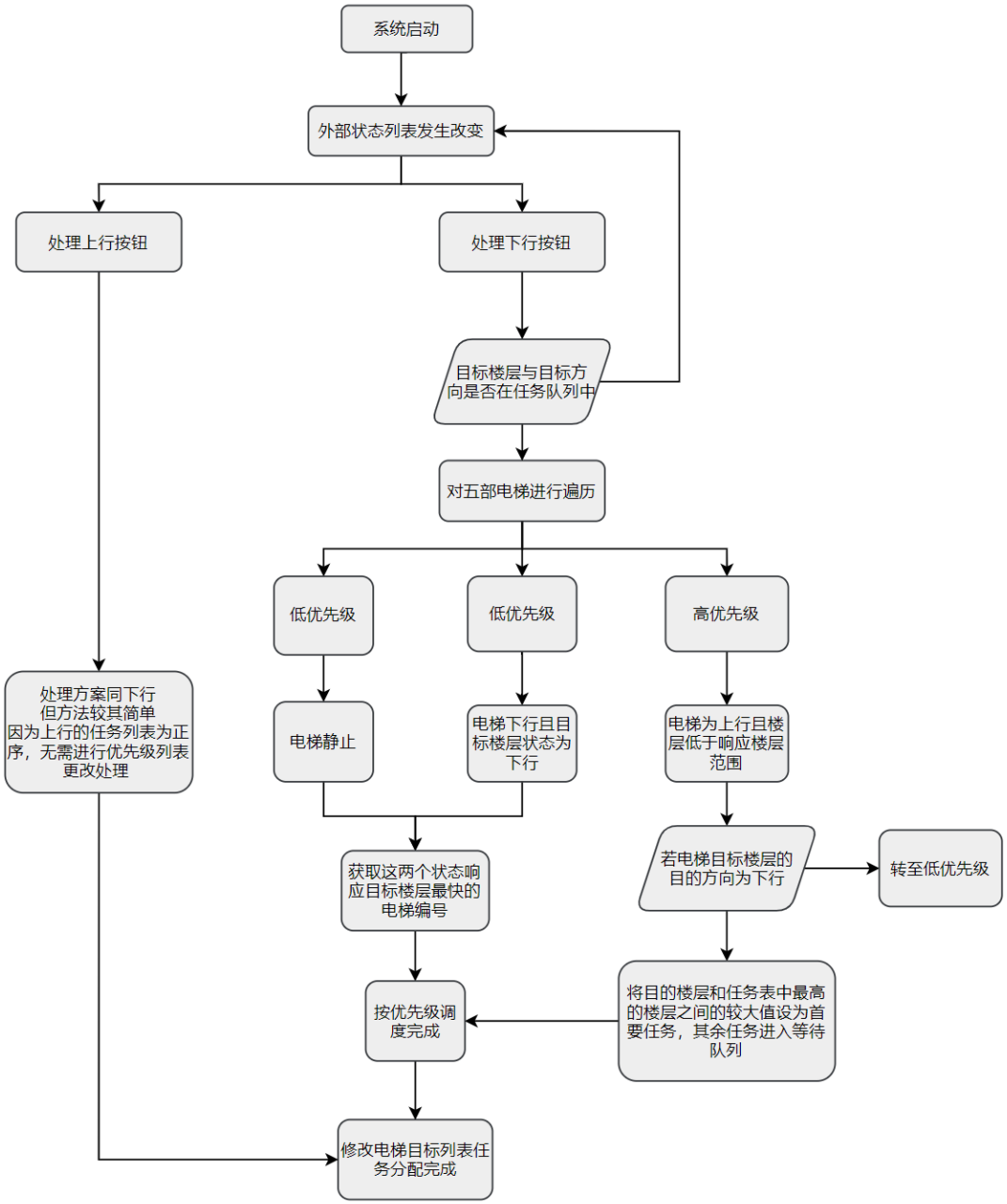
本项目的电梯调度算法结合 priority inversion 算法对电梯相应的优先级进行设计，旨在于模拟实际电梯乘客的合理化与人性化调度方案。主要利用到底电梯属性为 status 与 target_status，这两个属性分别代表电梯的运行状态与电梯所前往的目标楼层的乘客想去的楼层方向（上行或下行），基于这两个属性与楼层之间的关系对电梯进行合理调度。

在电梯运行线程函数中执行的为 run()函数：



其中 arrange 函数是对每层楼按下上下行键即外部乘客进行调度响应，而 schedule 函数则是结合 arrange 函数的结果与电梯内部乘客的需求进行电梯状态的更改与电梯目标队列的更改，并进行目标楼层分配。Step 函数则是电梯运行的基本函数，通过判断每个电梯被分配的任务列表与运行状态进行模拟楼层加减，设置 1s 的延时函数来模拟电梯的运行调度过程。这样的一个循环为电梯调度的单位架构，这也是模拟电梯调度的最小函数架构。下面来详细说明每一步函数所进行的调度方法与调度规则。

外部楼层调度函数(arrange)：



此函数用于相应电梯外乘客在每一楼层对电梯的调度需求，这也是调度算法的核心部分，由上流程图可知，这里分别处理上行电梯需求和下行电梯需求，实际实现的时候为两个函数，但在这两函数之间我依然插入了 schedule 函数。因为在分配完成任务后需要对其目标列表进行维护操作，防止二至之间存在由于未能及时更新列表而产生的错误。在外部列表发生状态变化时，我们会对该任务进行检查，

防止已经在任务列表中的任务由于乘客的重复按压按钮而导致任务冗余。完成检查任务后会对每一步电梯进行遍历，在这里我们为了最大程度满足电梯的合理化需求，会设置不同的优先级。目前以目标任务为下行为例，那么可以接受该任务的电梯可以具有以下几种运行状态：静止、向下运行且楼层高于需求楼层且目标楼层的目的方向为下行、向上运行且楼层低于需求楼层且目标楼层的目的方向为下行。我们会对这三种状态进行不同的处理，而且显然，为了电梯的最短路径合理化的需求，前两种电梯状态为低优先级，而状态三为需要处理的高优先级。面对高优先级状态需要对其任务楼层进行分析，要找到其任务楼层和需求楼层之间的最大值将其加入现有的任务列表，并将其余的楼层放入等待队列，等待本电梯的调用，详见流程图；而低优先级则为基本调度，选取处于这两种状态且可以最快达到需求楼层的电梯进行调度即可。上行电梯的处理方案基本同下行而且方法较其简单，因为上行的任务列表为正序，无需进行优先级列表更改处理。

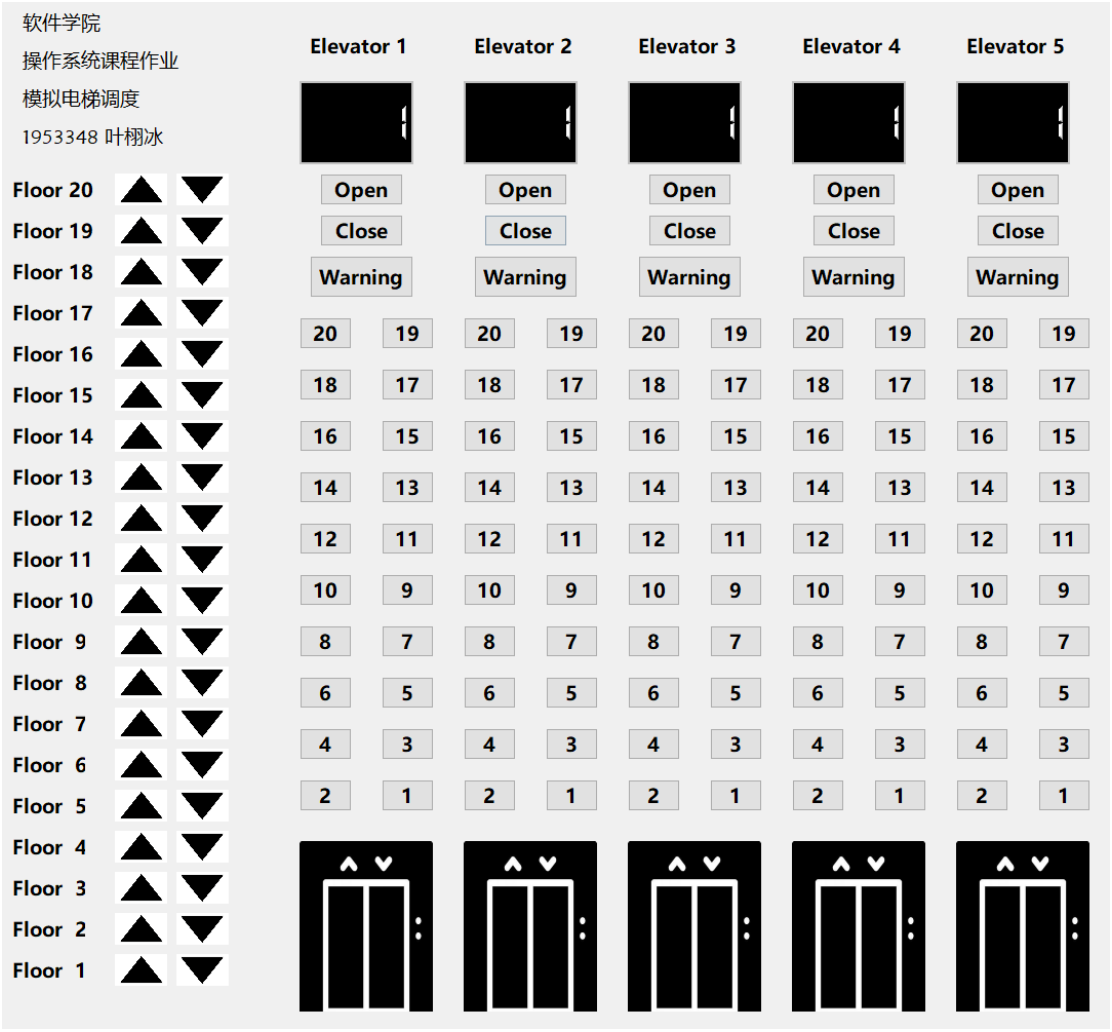
状态更新与维护函数 schedule():

该函数会基于每个电梯的任务列表对其状态进行更新，并且会完成电梯内部的需求，同时会包含内部需求判断函数 tell_inside_button()来判断，电梯在运行时电梯内部的需求按钮是否合理以及是否要被执行。由于其逻辑较为简单，仅仅是对电梯的状态进行更新与维护，在此不进行赘述，详见源代码 thread_construct.py。

UI 设计

项目选取 pyqt5 进行 UI 设计，PyQt5 是 Digia 的一套 Qt5 应用框架与 python 的结合，同时支持 2.x 和 3.x。Qt 库由 Riverbank Computing 开发，是最强大的 GUI 库之一。项目以 qt 类为主线程，电梯调度逻辑部分为另一线程，同时进行来实现同步调度的效果。由于项目界面中需要呈现每个电梯 20 个按钮与每层楼的全部按钮，因此界面较为繁杂，便将电梯所在楼层显示为 lcd 数码器，并设置电梯界面来展示其状态。

具体效果图如下：



上下行按钮与楼层按钮再按下后会有不同的点击触发效果，可运行可执行文件体验。