

# 处理机管理——电梯调度 设计方案

1851201 周子龙

## 一、 问题背景

在某栋 20 层的居民住宅楼中，一共有五部联网的电梯，现在需要为调度电梯的处理器设计一套算法，使得该处理器能够合理的响应各个楼层发出的请求，并合理的分配可支配的电梯，在合理的时间能到达各楼层。

## 二、 问题分析

### a) 电梯

在实际情况中，电梯里面需要设置必要的功能键，包括：

- i. 数字键：电梯中的人可以指定该电梯需要停靠的楼层
- ii. 开门键：在到达某一楼层后，电梯中的人可以延长开门时间
- iii. 关门键：如果没有用户按下该键，电梯将在最后一次感应到有人通过后的的一定时间内自动关门，如果用户按下该键，将立即关门
- iv. 显示面板：显示当前电梯的楼层位置和运行方向
- v. 报警键：按下后，将连接至调度中心，并联系相关工作人员

### b) 楼层

为了使得各个楼层的居民可以和电梯进行交互，在各个楼层中需要为每一个电梯设置必要的按键包括：

- i. 上下行键：确定该楼层的居民上、下行的需求
- ii. 显示面板：显示某一部电梯的当前位置和运行方向

## 三、 设计约束

### a) 各楼层中的电梯按钮是相互连接的

即当一个电梯按钮按下后，认为该楼层的所有电梯均按下了该按钮

### b) 所有电梯初始状态在第一层

### c) 当没有楼层发出请求时，所有电梯应当保持当前位置不动

### d) 电梯应当能在任何时候发出警报

### e) 电梯内部开门按钮仅能在电梯处于空闲状态或者开门状态激活

## 四、 电梯调度的逻辑

电梯调度算法已有许多相关的研究，基于本问题的实际背景情况，我们不需要为楼层设置不一样的优先级，综合“饥饿”等现象，我们采用类似于磁盘寻道的 LOOK 算法来实现对电梯的调度。

LOOK算法基于以下两个原则：

a) 最短距离优先原则

当多个电梯均处于空闲状态，或者同行（同上、同下）状态时，优先调度距离发出请求的楼层最近的电梯

b) 同向原则

当某一电梯处于运行状态时，只会响应在它运行方向上的楼层的同向请求，当同一方向上的请求均已完成后，该电梯才会相应另一方向的请求。如电梯 A 位于 10 楼且处于上行状态，那么电梯 A 只会响应 10 楼以上的上行请求。

## 五、 实现

a) 电梯的停靠楼层

为了实现楼层对电梯的调度，以及电梯内部的按键对电梯的调度，我们应当在每一部电梯内部储存该电梯的内部需求，以及楼层的需求。由于我们假设在一个楼层发出的请求视为对所有电梯同时发出请求，我们仅需要保留一份外部的请求数据，以及每一个电梯内部自己的请求数据。

b) 电梯的调度

在宏观上，我们为每一个电梯赋予一个实例，由主线程创建，并进行调度。在主线程中，应当可以对各个电梯（子线程）的某一时刻的数据进行访问，以获取该电梯的信息进行规划。

在主线程规划完毕后，我们为相应的电梯发出停靠的请求，并将该请求添加到该部电梯的带停靠楼层中。

基于如上分析，我们可以通过一个共享区存放各个电梯的停靠数据，在主线程调度完毕后，直接修改相应的电椅数据即可。

c) 电梯内部请求

在实际中，电梯应当无条件执行所有来自电梯内部的请求，即所有在内部按下的楼层，该电梯均需要停靠。故除了外部请求之外，我

们还需要在内部存放一个内部按键。

d) 开门和关门

我们为开门设置一个资源信号量，正常情况下（用户没有按下开关门）时在开门时释放，在关门时获取，如果用户在开门后按下开门按键，则临时消耗该资源，仅有当按下关门键再次释放该资源后才能正常关门。

## 六、 开发环境

系统环境：Mac OS 10.14.5 Visual Studio Code

语言：python

图形化界面：Tkinter

## 七、 运行

本程序于 python3.7.2 macOS Mojave 10.14.6 的环境下的执行情况如下图(主程序位于 GUI.py 中)：

Elevator Simulator by Zilong Zhou 1851201

Elevator0

SOS

-

1

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

Elevator1

SOS

-

1

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

Elevator2

SOS

-

1

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

Elevator3

SOS

-

1

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

Elevator4

SOS

-

1

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

20

△

▽

19

△

▽

18

△

▽

17

△

▽

16

△

▽

15

△

▽

14

△

▽

13

△

▽

12

△

▽

11

△

▽

10

△

▽

9

△

▽

8

△

▽

7

△

▽

6

△

▽

5

△

▽

4

△

▽

3

△

▽

2

△

▽

1

△

▽

Schedule Detile:

elevator schedule detile will be shown here

Elevator Simulator by Zilong Zhou 1851201

Elevator0

SOS

↑

3

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

Elevator1

SOS

↑

6

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

Elevator2

SOS

-

1

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

Elevator3

SOS

-

1

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

Elevator4

SOS

-

1

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

20

△

▽

19

△

▽

18

△

▽

17

△

▽

16

△

▽

15

△

▽

14

△

▽

13

△

▽

12

△

▽

11

△

▽

10

△

▽

9

△

▽

8

△

▽

7

△

▽

6

△

▽

5

△

▽

4

△

▽

3

△

▽

2

△

▽

1

△

▽

Schedule Detile:

schedule elevator0 to 9

9▽ is push from outside

schedule elevator1 to 12

12△ is push from outside

schedule elevator1 to 12

elevator1 is push 12 from inside

elevator schedule detile will be shown here