

# OO 第五次作业要求 v2.2

2017/3/31

## 1. 作业目标

本次作业是设计一套由 3 部电梯组成的多电梯调度系统，通过采用线程机制，在第三次作业所实现程序的基础上完成新的调度系统程序。

## 2. 作业内容和成果物

### 2.1 单电梯系统基本描述

此部分内容继承自第二次作业。

详细内容参考《OO 第二次作业要求 V8.1》中 2.1 节第 1)、第 2) 部分的说明。

### 2.2 多电梯系统的补充描述

- 1) 本电梯系统由 3 部完全一样的电梯组成。
- 2) 三部电梯的运行在 1~20 层（含两端）之间，每部电梯的楼层计数采用中国式计数，即 1 层显示为 1，2 层为 2，依次类推，直至顶层显示为 20。
- 3) 每个楼层都有电梯请求按钮（下面简称为楼层按钮），告知乘客想要去的楼层与发出请求楼层之间的方向关系。如果乘客按下了向上请求按钮，则表示想搭乘电梯去往上面的楼层；反之，则表示想搭乘电梯去往下面的楼层。规定最底层只有向上运行请求按钮，最顶层只有向下运行请求按钮，中间楼层同时有向上和向下两个方向请求按钮。
- 4) 一个电梯箱体（剩余文档简称电梯）内有对应每个楼层的目的地请求按钮（下面简称目标），一般显示相应楼层的编号。
- 5) 系统由一个调度器根据相应的策略安排这 3 部电梯如何响应楼层请求（FR 请求），发给某个电梯的 ER 请求则只能由这个电梯进行响应处理。
- 6) 为了提高测试时的系统状态的可观察性，设定电梯运行一层楼消耗 3 秒，开关门消耗 6 秒。

## 2.3 电梯基本运行规则设定

- 1) 程序运行开始或重置时设置所有 3 部电梯停靠在一层，电梯系统的时间点设为 0；
- 2) 中间楼层只有一个上行和一个下行按钮，1 层只有上行按钮，顶层只有下行按钮。
- 3) 一个楼层按钮同一时刻只能发出一个上行或下行请求，在该请求的响应未完成之前，即电梯到达该楼层并开关门完毕前（含开关门完毕时刻），系统视该楼层的其他同向请求为同一个请求，直接从输入中掉剔除不进行调度处理；
- 4) 一个电梯内目标楼层按钮发出的 ER 请求在响应完成之前，即到达该楼层并开关门完毕前（包括开关门完毕时刻），系统视该电梯去往该目标楼层的其他请求为相同请求，直接从输入中剔除掉不进行调度处理；
- 5) 所有请求按照时间上的先来先服务策略（First Arrived First Served, FAFS）作为基本调度原则，具体含义是，在没有其它策略时，按照 FAFS 来响应。

本次作业的电梯系统在 ALS\_Schedule (A Little Smart Schedule) 调度策略（详见第三次作业的具体规定）基础上，引入运动量均衡策略，即 3 部电梯的累积工作量要尽量均衡：

(a) 电梯“运动量”：即一部电梯行驶的楼层数。电梯的累积运动量，是指一部电梯从系统开始时刻到当前时刻累计行驶的楼层数。

(b) 针对任何一个楼层请求（FR 请求）：如果有电梯可以响应（即电梯当前不在响应某个请求的过程中，或者在响应一个请求过程中但可捎带），优先选择可以捎带的电梯来响应。如果有多部电梯可以捎带，则选择**累积运动量最小**的电梯（如果有多个满足该条件，则可随机选择）；如果没有可以捎带的电梯，则选择可以响应中的**累积运动量最小**的电梯来响应。如果没有可以响应的电梯，则一直等待直至有电梯能够响应。

(c) 针对任何一个电梯内请求（ER 请求）：按照所属电梯和该请求的产生时间顺序调度给该电梯进行处理，注意捎带策略仍然适用，但无法考虑运动量均衡策略。

## 2.3 关于“顺路”请求的补充说明

设电梯当前状态为  $e=(e\_n, sta, n)$ ，即当前所处楼层为  $e\_n$ ，运动状态为  $sta$ （包括 UP，DOWN，STILL 三种状态），当前运动的目标为楼层  $n$ ，则：

(1)  $(e.sta = UP \rightarrow 20 \geq e.n \geq e.e\_n) \ \&\& \ (e.sta = DOWN \rightarrow 1 \leq e.n \leq e.e\_n) \ \&\& \ (e.sta = STILL \rightarrow 1 \leq e.e\_n \leq 20)^1$

(2) 对于任意一个楼层请求  $r=(FR, n, dir, t)$ ，如果是电梯当前运动状态下的顺路请求，则一定有：

$(r.dir=e.sta) \ \&\& \ (r.dir=UP \rightarrow (r.n \leq e.n) \ \&\& \ (r.n \geq e.e\_n)) \ \&\& \ (r.dir=DOWN \rightarrow (r.n \geq e.n) \ \&\& \ (r.n \leq e.e\_n))$

(3) 对于任意一个电梯运载请求  $r=(ER, n, t)$ ，如果是电梯当前运动状态下的顺路请求，则一定有：

$(e.sta=UP \rightarrow (r.n > e.e\_n)) \ \&\& \ (e.sta=DOWN \rightarrow (r.n < e.e\_n))$

以上三条同第三次作业内容（但是修改了逻辑表达式的表达准确性）。

(4) 对于  $e.sta = STILL$  状态分为 3 种情况。一是在一定时间内电梯完成了所有请求而停在某一楼层，此种情况没有“顺路”请求；二是电梯在捎带响应某个请求中途停靠在该楼层静止时，这时视为电梯仍旧没有改变运动状态，对于满足(2)和(3)的条件时，同样是顺路请求；三是电梯在到达当前主请求目标楼层停靠静止（包含到达时刻，即到达时刻为静止状态）时，此种情况同样没有“顺路”请求（注意 STILL 状态仅用于输出表达，输入只有 UP/DOWN 两个状态）//注意：此部分内容在第三次作业指导书中将 STILL 状态分为 2 种情况。

## 3. 作业要求和限制

### 3.1 输入规范

- 1) 本次作业要求使用控制台进行输入符合规范的请求字符串。
- 2) 请求分为两类：一类是楼层请求，一类是电梯内请求。注意，本次作业不支持开门和关门请求的处理。
- 3) 输入格式：(FR, floor, UP/DOWN)，即楼层请求；或(ER, #Elevator, floor)，电梯请求，注意需指明是哪部电梯的请求。

---

<sup>1</sup> 符号“ $\rightarrow$ ”是逻辑蕴含的意思

(注, 电梯编号为 1, 2, 3, 如 (ER, #1, 5))

- 4) 请求事件中的时间  $t$  自动从系统获得, 要转换为电梯系统时间(浮点数)(例: 初始时刻时间为 1234123ms, 请求输入时间为 1236789ms, 则当前时刻为 2.666s)。
- 5) 由于是从控制台输入, 且请求时间自动取自系统时间, 因此请求队列是按照时间自排序的, 不存在输入时间不按顺序的问题。
- 6) 一行可输入多条请求, 请求间用 “;” 或 Tab 分割, 同一行的指令取相同的系统时间。当一次输入多条请求时, 该次输入的所有指令都采用同一个系统时间。本次作业规定一次最多输入 10 条请求。输入次数不限。
- 7) 所有字符均为英文状态下的字符(要求测试者保证)。请求之间可以使用空格和 tab 分隔, 这一点请在 readme 说明。请求内部元素之间可以有空格, 要求程序能够自动过滤。

## 4.2 输出规范

本次作业的执行结果要求输出到一个文本文件, 文件命名规则是 `result.txt`, 字符集采用英文 ASCII 码。

输出方法参考 <https://docs.oracle.com/javase/tutorial/essential/io/file.html>。

输出内容包括 2 类 3 种:

- 1) 对于无效请求、实质上的相同请求一定要输出(报告), 即使进行容错也要输出相应的字符串。

格式为: `st:INVALID [request, T]`, 或

`st:SAME [request, T]`

其中  $st$  是输出时的系统时间,  $request$  为输入的请求字符串,  $T$  是请求产生时刻的电梯系统时间。

- 2) 每个有效请求执行完毕的输出请求内容和请求执行结果, 分两种情况:

- i. 任何一部电梯停靠时要立刻输出停靠信息, 输出信息为输出时系统时间, 请求信息, 电梯执行效果:

格式为: `st:[request, T] / (#电梯, 楼层, UP/DOWN, 累积运动量, t)`

其中  $st$  是输出时的系统时间,  $[request, T]$  为有效请求的字符串和输入时的电梯系统时间, 用 “[]” 包含。中间使用 “/” 分割。后一部分是该请求的执

行效果，其中  $t$  是停靠时刻，即电梯刚到达目标楼层由运动转为静止状态，尚未执行开关门的电梯时间。

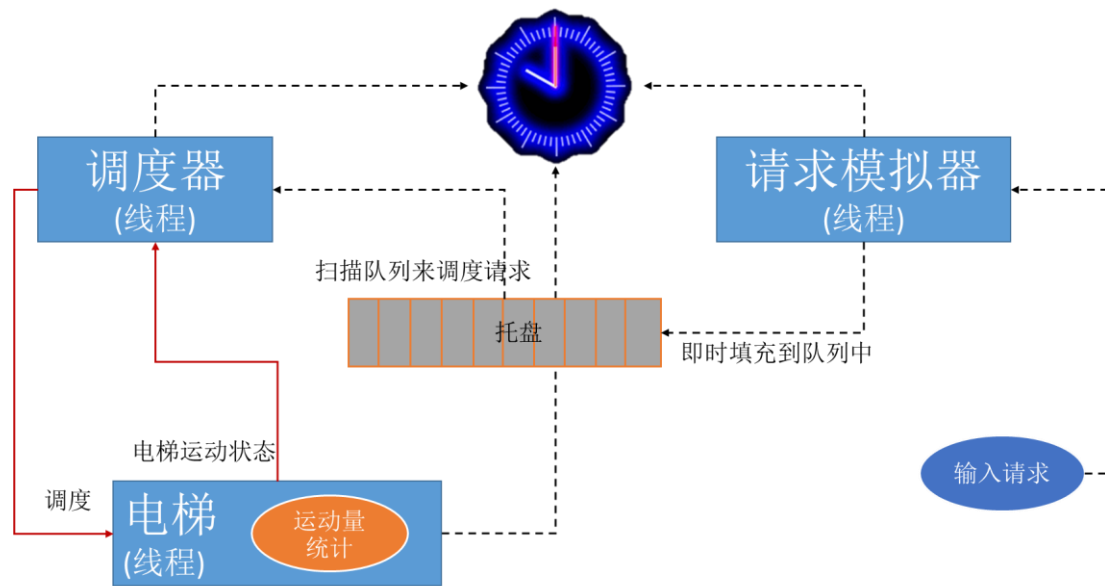
- ii. 同层请求时输出为:  $st:[request, T]/(\# \text{电梯}, \text{楼层}, \text{STILL}, \text{累积运} \\ \text{动量}, t)$  , 其中 STILL 代表静止。(参考第三次作业指导书例 11)
- iii. 如果一次停靠执行了多条请求，那么需要分行输出 (参考第三次作业例 12)

- 3) 输出格式要求所有字符为英文符号。一个请求和执行效果占用一行。
- 4) 时间处理方面,  $st$  的单位为毫秒, 直接取自系统时间;  $T$  和  $t$  的单位为秒, 支持一位小数, 取整方式自定, 但要满足电梯运行时间和停靠时间要求 (即处理后运行一层和停靠时间差要准确为 3.0S 和 6.0S), 不支持科学记数法。
- 5) 其他未规定的地方可由编程者自行决定。

## 4. 其它说明事项

### 4.1 设计要求

- 1) 使用继承机制, 重构代码, 保留前两次调度功能, 增加新的多电梯均衡调度方法完成响应请求。
- 2) 数据无效的请求 (如楼层超过20) 将被直接从输入请求序列中拿掉, 不影响对其他有效请求的调度处理。
- 3) 任何情况下, 程序都不应crash, 要正常结束 (exitcode=0)。
- 4) 使用多线程并发处理输入的请求。注意, 如果同步控制设计不当, 多线程运行时会产生各种莫名其妙的行为, 甚至在逻辑上无法解释。建议: 采用打印输出, 而非debug的办法来调试多线程程序, 否则调试时观察到的程序行为和实际运行的行为会有很大差异, 且无法解释。
- 5) 修复第三次作业的 bug, 并在 readme 中加以说明。



参考程序框架

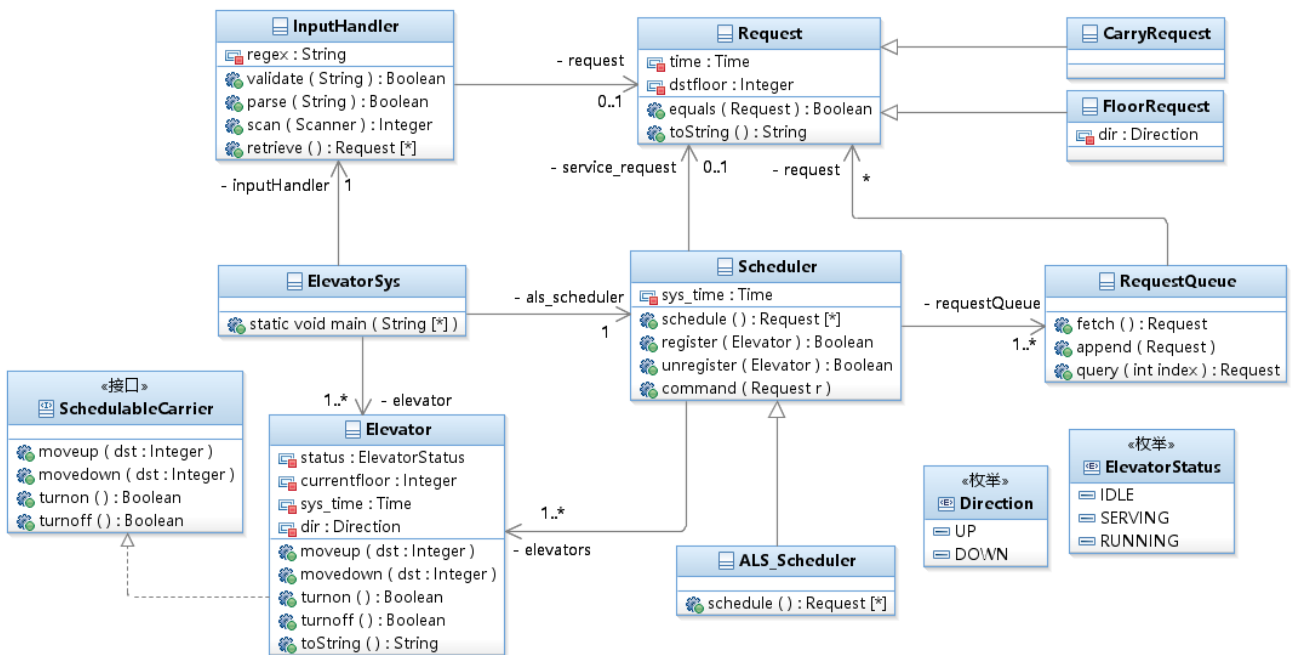
## 4.2 测试要求

- 1) 设计请求序列，重点检查是否违背本次作业的调度规则；
- 2) 检查电梯系统在电梯调度和运载过程中的时间处理正确性；
- 3) 检查线程交互机制和共享对象的访问控制机制；
- 4) 检查是否使用继承机制来扩展调度器，并保留之前的调度策略；
- 5) 本次作业互测不支持针对无效输入格式的测试；
- 6) 针对设计要求未满足情况，除了报告其导致的功能性 bug 外，还作为 incomplete 类 bug 来报告设计要求不满足，但写清楚那个要求不满足。

## 4.3 错误处理原则：

- 1) 遇到无效请求（包括格式或内容不符合要求的），输出无效输入请求串后，继续处理下一个输入请求直至结束。
- 2) 任何情况下，程序都不应 crash，要正常结束（exitcode=0）。

## 4.4 Tips



## 参考类图

## 5. 其他规定

- 1) 文档中**红色**和**粗体字**为重要提示或强制要求。
- 2) 必须提交全部工程文件。
- 3) 无效作业，以下四种情况视为无效作业。
  - (1) 程序不能编译和运行；
  - (2) 未使用 **Java** 语言；
  - (3) 未采用线程机制来实现调度器、电梯和请求模拟器；
  - (4) 虽然采用多线程，但是实际只构造了不足 3 部电梯。