

00 第五次作业要求

2018

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和三次作业的指导书。

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) 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。
- 4) 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 3.0 秒，开关门消耗 6.0 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电梯停靠在一层；
- 2) 电梯的状态定义：
 - a) **电梯运行状态**：从电梯启动时刻,到电梯运动停止时的运行状态,包括上行（UP）和下行（DOWN）状态。此状态电梯速度大于 0。（启动时刻，运行停止时刻）。
 - b) **电梯开关门状态**：电梯静止时，从门打开时刻（门开始动作），到门完全关闭时刻（门刚刚停止动作）时的状态。此状态电梯速度等于 0。[门打开时刻，门关闭时刻]。

- c) **电梯停留状态 (WFS 状态, Wait for Service)**: 电梯停在某层, 且门长时间处于关闭状态。此状态电梯速度等于 0, 且请求队列为空。(门关闭时刻, 门准备打开时刻或电梯准备启动时刻]。
- d) **STILL 状态**: 此时电梯处于运行速度为 0 的状态。[电梯运行停止时刻, 电梯准备启动时刻]。(注意 STILL 状态仅用于输出表达)
- 3) 一个楼层**同一时刻**只能发出**一个上行或下行请求**。电梯未到本楼层的时候, 某个请求按钮被按下变亮后, 再按不会产生实际效果(即不被响应)。但是发出上行请求后可以再发出下行请求, 反之亦可, 这视为两个不同的请求, 执行完一个后另一个仍需执行。在电梯到达某楼层处于开关门状态时, 该楼层的多个同向请求只认为是一个请求。当电梯关门动作完成后(含关门完毕时刻), 可以再产生新的上下行请求;
- 4) 电梯内的一个目标楼层按钮被按下后只能发出前往某个目标楼层的请求, 一旦发出某个目标楼层请求后, 在电梯到达该楼层并完成关门动作前(包括关门完毕时刻), 目标楼层与该按钮楼层相同的多个电梯内请求被认为是一个请求。当电梯关门结束后, 可以再发出任意目标楼层请求。
- 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。
- 6) 如果电梯同时收到了电梯内请求和楼层请求时, 则按照输入时的请求排列顺序执行。
- 7) 电梯系统在 FAFS 调度上采用 ALS_Schedule (A Little Smart Schedule) 调度策略:
- (1) 只要队列不为空, 每次都取出队列头请求来调度(同傻瓜调度策略);
 - (2) 电梯在运动过程中不能突然改变运动方向;
 - (3) 在调度电梯完成一个(或一组执行时间有重叠)请求的过程中, 电梯要响应所有满足“顺路捎带”条件的请求, 关于捎带的详细定义参考 2.4。

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, 引入电梯运动量均衡策略, 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，每行驶一层楼运动量记为 1。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，即当前处于 WFS（等待服务）状态的电梯，或者可以捎带该请求的电梯，则电梯调度系统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（如果有多个电梯同时满足上述条件，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）：则只能由该电梯响应其自身的电梯内请求，并按照该请求的产生时间顺序进行调度，此时无需考虑运动量均衡策略。在此基础上仍要考虑捎带的情况。

2.4 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) $(e.sta = UP \rightarrow 20 \geq e.n \geq e.e_n) \parallel (e.sta = DOWN \rightarrow 1 \leq e.n \leq e.e_n)$

注释：本段所述的可捎带条件可以这样理解：①电梯状态向上运行时，发出向上请求的楼层处于当前楼层和 20 层之间；②电梯状态向下运行时，发出向下请求的楼层处于当前楼层和 1 层之间。

- (2) 对于任意一个楼层请求 $r=(FR, n, dir, t)$ ，如果电梯当前是运动状态，则顺路捎带请求一定有：

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

$$(r.dir=DOWN \rightarrow (r.n \geq e.n) \&\& (r.n \leq e.e_n)))$$

注释：本段所述的可捎带条件可以这样理解：电梯状态向上或向下运行时，新楼层请求的运动方向与当前方向一致，而且新请求发生的楼层在当前所处楼层和目标楼层之间。

- (3) 对于任意一个电梯内运行请求 $r=(ER, n, t)$ ，如果是电梯当前运动状态下的顺路捎带请求，则一定有：

$$(e.sta=UP \rightarrow (20 \geq r.n \geq e.e_n)) \parallel$$

$$(e.sta=DOWN \rightarrow (1 \leq r.n \leq e.e_n))$$

注释：本段所述的可捎带状态可以这样理解：电梯内请求的目标楼层在电梯的前进方向上。

- (4) 对于 `e_sta = STILL` 状态分为 2 种情况。一是在电梯处于停留状态（WFS 状态），此种情况没有“顺路捎带”请求，因为此时请求队列为空；二是处于开关门状态时，若请求不是同质请求，则（**注意 STILL 状态仅用于输出表达，输入只有 UP/DOWN 两个状态**）

- (5) 可捎带请求的先决条件是非同质请求。

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。
- 3) **输入格式：(FR, floor, UP/DOWN)，即楼层请求；或(ER, #Elevator, floor)，电梯请求，注意需指明是发出请求的电梯编号。（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，所获得的时间（微秒）要转换为电梯系统时间（浮点数，单位为秒）（例：假设第一条请求发生的时间为 1234123ms，当前请求输入时间为 1236789ms，则当前时刻为 2.666s）。
- 5) 由于请求是从控制台输入，且请求时间自动取自系统时间，因此请求队列是按照时间先后进行排序的，不存在输入时间不按顺序的问题。
- 6) 一行可输入多条请求，请求之间用“;”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，该次输入的所有请求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（即使包含无效请求也不超过 10 条）。**
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）。请求之间使用“;”，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。
- 8) 需要退出程序运行时，要求输入“END”，程序结束运行。

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] / (#电梯号, 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] / (#电梯, 楼层, STILL, 累积运动量, t)`，其中 STILL 代表静止。（参考第三次作业指导书例 11）

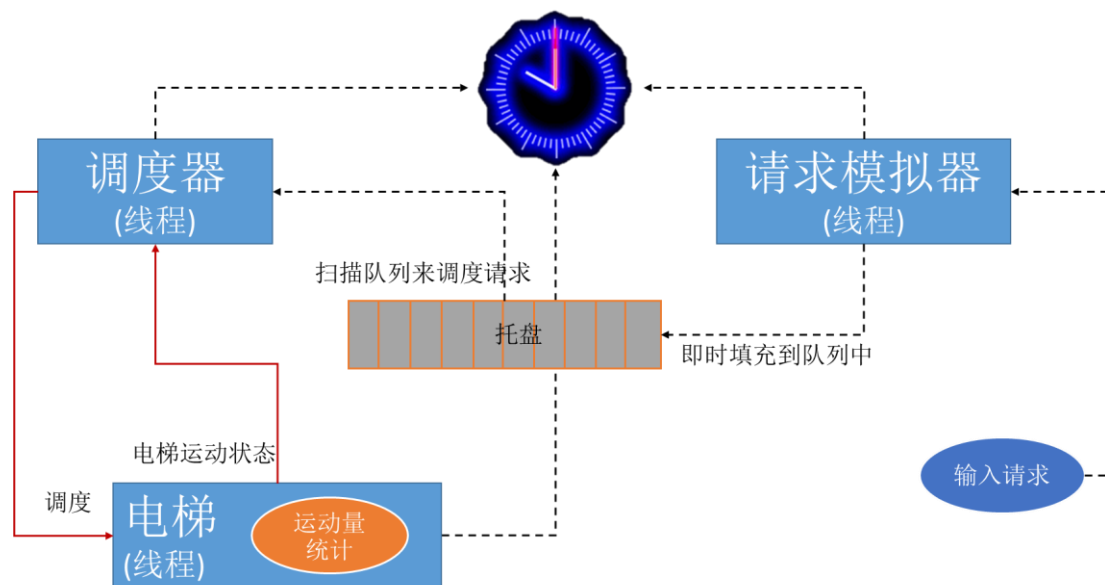
- iii. 如果一次停靠执行了多条请求，那么需要分行输出（参考第三次作业例 12）

- 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

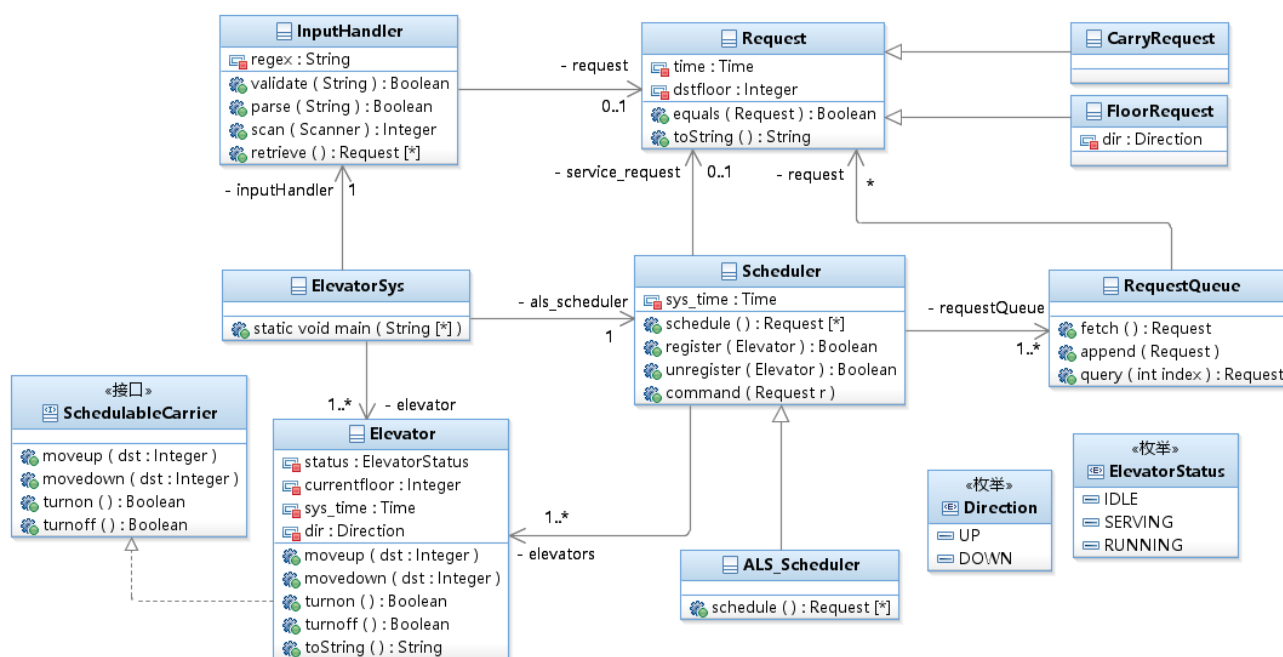
- 1) 设计请求序列，重点检查是否违背本次作业的调度规则；
- 2) 检查电梯系统在电梯调度和运载过程中的时间处理正确性；
- 3) 检查线程交互机制和共享对象的访问控制机制；
- 4) 检查是否使用继承机制来扩展调度器，并保留之前的调度策略；
- 5) 本次作业互测不支持针对无效输入格式的测试；

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），输出无效输入请求字符串后，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。**