<公交车调度>概要设计

版本号：v0.0

编制时间：2022/4/22

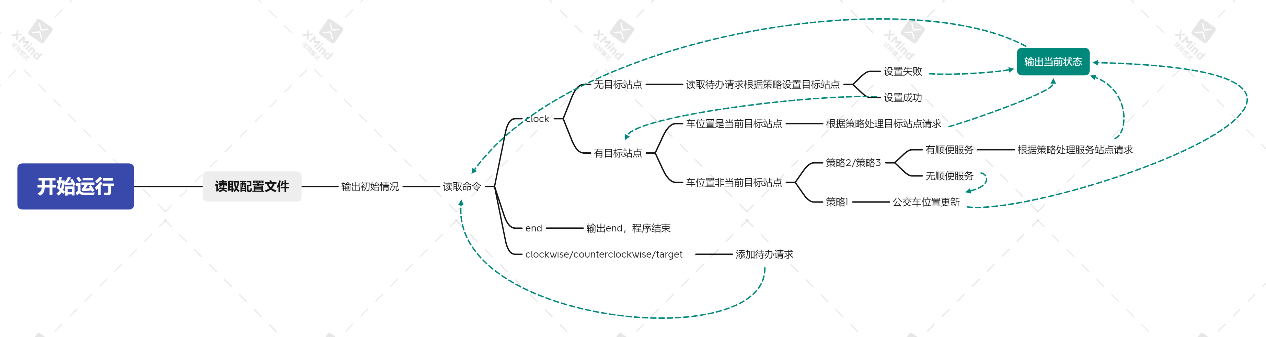
编制人员：吴超然 游俊轩

1. 输入、输出设计

输入：根据“dict.dic”文件依次读取字符串用字符串匹配判断配置文件所更改的变量。

输出：详见oj要求。

1. 算法设计

【策略1】

先来先服务，将上下车请求都用链表存储，根据顺序判断目标站点，到达站点后检查后续节点是否为同一站点，若相同则一并完成，每次输出时，遍历链表来检查请求状态。

【策略2】

最短寻找时间，根据时间对链表遍历取最小值(时间相同时顺时针请求优先,target取最短时间)，更新小车状态， 路过站点时检查链表内是否有同方向、无方向的该站点请求，若有，则停，到达目标时，遍历该链表找到所有该站点请求，每次输出时，遍历链表来检查请求状态。

【策略3】

顺便服务策略，同样将请求用链表存储，行驶方向确定后，每次检查最短完成时间，根据最短完成时间更新方向，到达站点后，将所有该站点请求完成(无视方向、上下车)

1. 数据结构设计
2. 宏定义

#define MAX\_NUM (10)//站点最大个数

#define clockwise (0)//顺时针

#define counter\_clockwise (1)//逆时针

#define get\_off (2)//下车,以上三个定义表示站台请求方向状态

#define FCFS (0)//先来先服务策略

#define SSTF (1)//最短寻找时间优先策略

#define SCAN (2)//顺便服务策略

#define dis\_max 99//最大距离,作为反方向距离计算使用

1. 全局变量设置

int sta\_dis=2;//站点距离

int sta\_num=5;//站点数

int length=10;//总长度

int strategy=FCFS;//使用策略

station\* head = NULL;//链表头

int status[MAX\_NUM][3]//输出时更新，表示站点的请求状态

car bus//记录公交车状态

1. 结构体定义

struct \_CAR

{

int id;//该作业中车仅有一辆

int flag;//判断是否需要顺便停车

int pos;//车辆位置

int direction;//车辆行驶方向

int target = -1;//标记当前车的目标站点

};

typedef struct \_CAR car;

struct \_REQUEST

{

int id;//表示请求顺序

int direction;//表示方向:clockwise/counterclockwise/get\_off

int distance;//表示该站与车距离,若反向则为dis\_max

struct \_REQUEST\* next;

}

typedef \_REQUEST request;

1. 系统模块划分

|  |  |  |
| --- | --- | --- |
| 模块名称 | 模块说明 | 函数名 |
| head\_for\_car.h | 设置程序所需的全局变量、数据结构 | 无 |
| input.c | 读取“dict.dic”文件，更改变量 | extern void input() |
| output.c | 输出公交车、车站当前状态 | extern void output() |
| change.c | 更新公交车、车站状态 | extern void change() |
| pushList.c | 读取命令录入为结构体形式并存入链表 | extern void pushList() |
| stg1.c | 根据策略1判断车的目标站台 | extern void stg1() |
| stg2.c | 根据策略2判断车的目标站台 | extern void stg2() |
| stg3.c | 根据策略3判断车的目标站台 | extern void stg3() |
| pass2.c | 根据策略2判断车的顺便停车情况 | extern void pass2() |
| pass3.c | 根据策略3判断车的顺便停车情况 | extern void pass3() |
| s1\_serve.c | 根据策略1处理目标站台的请求 | extern void s1\_serve() |
| s2\_serve.c | 根据策略2处理目标站台的请求 | extern void s2\_serve() |
| s3\_serve.c | 根据策略3处理目标站台的请求 | extern void s3\_serve() |
| ps2\_serve.c | 根据策略2处理顺便停车站台的请求 | extern void ps2\_serve() |
| ps3\_serve.c | 根据策略3处理顺便停车站台的请求 | extern void ps3\_serve() |