

电梯模拟



姓名：王昱

学号：PB21030814

一.需求分析

• 问题描述

- 设计一个电梯模拟系统，模拟某校五层教学楼的电梯系统，共五层电梯，第一层是本垒层。
- 模拟时钟决定每个活动体的动作发生时刻和顺序，从0开始时间单位为0.1s。系统在某个模拟瞬间处理有待完成的事情，然后把时钟推进到某个动作预定要发生的下一时刻。
- 人和电梯的动作耗时情况(t 为时间单位):有人进出时，电梯每隔 $40t$ 测试一次；电梯关门开门需要 $20t$ ；乘客进出电梯 $25t$ 。
- 最终按时序显示系统状态变化过程:发生的全部人和电梯的动作序列。

• 采用的数据结构

- 为模拟乘客进电梯时先进后出的实际情况，在电梯内采用栈这一数据结构存储乘客信息。
- 为模拟乘客等待电梯先进先出的实际情况，在电梯外采用队列这一数据结构存储等待乘客的信息。

二.设计思路

- 本工程包含的项目文件:head.h|customer.c|elevator.c|operate.c
 - head.h文件主要包括了一些结构体定义、全局变量定义以及一些枚举类型的定义
 - customer.c文件主要包括有关乘客的动作
 - elevator.c文件主要包括有关电梯的动作
 - operate.c文件是对电梯系统模拟的“驱动文件”

- 下面列举一些重要的结构体、枚举类型、函数

```
typedef struct customer
{
    int ID;           //标记乘客
    int InFloor;      //进入了哪层楼
    int OutFloor;     //将要去哪层楼
    int GiveupTime;   //能容忍的等待时间
    int InterTime;    //下一个人出现的时间间隔
} Customer, *CustomerPtr;
```

乘客的结构体定义，这里的ID为了方便直接采用数字表示

//电梯里的乘客栈

```
typedef struct
{
    CustomerPtr base;
    CustomerPtr top;
    int stacksize;
} CustomerStack, *CustomerStackPtr;
```

//电梯门前的等待队列

```
typedef struct CustomerQueueNode
{
    Customer data;
    struct CustomerQueueNode *next;
} CustomerQueueNode, *CustomerQueuePtr;
```

```
typedef struct
{
    CustomerQueuePtr front; //队头指针
    CustomerQueuePtr rear;  //队尾指针
    int CustomerNumber;
} CustomerWaitQueue, *CustomerWaitQueuePtr;
```

```

typedef struct
{
    int Floor;
    //电梯当前位置
    int D1;
    //电梯无人进出则D1为0
    int D2;
    //电梯在某层停300t以上则D2为0
    int D3;
    //电梯门开并且无人进出电梯则D3等于0
    enum ElevatorState State;
    //电梯的当前状态(going_up/going_down/idle)
    int CallCar[MAXFLOOR + 1];
    //电梯内按钮,有人按下按钮置为1, 否则置为0
    int CustomerNumber;
    CustomerStackPtr CustomerStack[MAXFLOOR + 1];
    //乘客栈
    enum ElevatorActivity activity;
    //电梯的几个活动
    int activity_timer;
    //活动计时器
    int customer_in_out_timer;
    //乘客进出时间计时器
    int StateChange;
    //判断状态是否需要改变
} Elevator, *ElevatorPtr;

```

电梯的结构体定义，里面用到的两个枚举类型如下

```

enum ElevatorState
{
    going_up = 1,
    going_down = -1,
    idle = 0
};
enum ElevatorActivity
{
    OPENING, //正在开门
    OPENED,  //已经开门
    CLOSING, //正在关门
    CLOSED,  //已经关门
    ACCELERATING,
    SLOWING,
    MOVING,
    IDLE
};

```

电梯中有CallCar以及CallUp、CallDown，其中CallCar数组模拟的是电梯里面的按钮，所以放在电梯结构体中；CallUp、CallDown数组模拟的是电梯外面的按钮，所以定义了全局变量来模拟。

电梯的状态和活动都是依据课本[实现提示]部分完成的。

。与乘客栈有关操作的函数

```

bool InitStack(CustomerStackPtr S);
void DestroyStack(CustomerStackPtr S);
bool StackEmpty(CustomerStackPtr S);
void Push(CustomerStackPtr S, Customer e);
CustomerPtr Pop(CustomerStackPtr S);

```

- 与乘客等待队列有关操作的函数

```
void InitQueue(CustomerWaitQueuePtr Q);  
void DestroyQueue(CustomerWaitQueuePtr Q);  
void EnQueue(CustomerWaitQueuePtr Q, Customer e);  
Customer DeQueue(CustomerWaitQueuePtr Q);  
bool QueueEmpty(CustomerWaitQueuePtr Q);
```

这些函数的实现都较为基础，在此不再赘述。

三.具体实现

- 本工程的最核心的部分是电梯各种活动的转变，实际上它就相当于一个有限状态机(FSM)，用C语言实现FSM实际上也就是用switch语句实现，与VHDL语言中的Verilog类似。

```

void ChangeElevatorActivity(void)
{
    int cnt;          //用于计数
    int signal = 1; //用在MOVING中
    switch (FiveFloorElevator->activity)
    {
    case OPENING:
    {
        //开门时置D1和D2非0值
        FiveFloorElevator->D1 = 1;
        FiveFloorElevator->D2 = 1;
        FiveFloorElevator->activity = OPENED;
        FiveFloorElevator->activity_timer = door_test_time;
        DisplayInfo(ELEVATOR_OPENED, 0, 0, 0, FiveFloorElevator->Floor);
        break;
    }
    case OPENED:
    {
        if (NoCustomerInOut())
        {
            //无人进出D1为0,D3为0
            FiveFloorElevator->D1 = 0;
            FiveFloorElevator->D3 = 0;
            FiveFloorElevator->activity = CLOSING;
            FiveFloorElevator->activity_timer = open_close_door_time;
            DisplayInfo(ELEVATOR_CLOSING, 0, 0, 0, FiveFloorElevator->Floor);
        }
        else
        {
            //有人进出时电梯每隔40t测试一次
            FiveFloorElevator->activity_timer = door_test_time;
        }
        break;
    }
}

```

```

}
case CLOSING:
{
    if (FiveFloorElevator->D1 == 0)
    {
        FiveFloorElevator->D3 = 1;
        FiveFloorElevator->activity = CLOSED;
        DisplayInfo(ELEVATOR_CLOSED, 0, 0, 0, FiveFloorElevator->Floor);
    }
    else
    { //在关门期间有人到来
        FiveFloorElevator->D3 = 1;
        FiveFloorElevator->activity = OPENING;
        FiveFloorElevator->activity_timer = open_close_door_time;
        DisplayInfo(ELEVATOR_OPENING, 0, 0, 0, FiveFloorElevator->Floor);
    }
    break;
}
//门已经关闭
case CLOSED:
{ //准备移动
    FiveFloorElevator->CallCar[FiveFloorElevator->Floor] = 0;
    if (!NoCustomerInOut())
    {
        FiveFloorElevator->D3 = 1;
        FiveFloorElevator->activity = OPENING;
        FiveFloorElevator->activity_timer = open_close_door_time;
        DisplayInfo(ELEVATOR_OPENING, 0, 0, 0, FiveFloorElevator->Floor);
        break;
    }
    if (FiveFloorElevator->State == going_up)
    {
        if (CustomerUpQueue[FiveFloorElevator->Floor]->CustomerNumber)
            CallUp[FiveFloorElevator->Floor] = 1;
    }
}

```



```

else
    CallUp[FiveFloorElevator->Floor] = 0;

//如果有CallCar[cnt]成立则加速上升
for (cnt = FiveFloorElevator->Floor + 1; cnt <= MAXFLOOR; cnt++)
{
    if (FiveFloorElevator->CallCar[cnt])
    {
        FiveFloorElevator->activity = ACCELERATING;
        FiveFloorElevator->activity_timer = accelerate_time;
        DisplayInfo(ELEVATOR_ACCELERATING, 0, 0, 0, FiveFloorElev
        break;
    }
}
//没有则停下
if (cnt == MAXFLOOR + 1)
{
    FiveFloorElevator->activity = IDLE;
    FiveFloorElevator->activity_timer = elevator_max_wait_time;
    FiveFloorElevator->State = idle;
    DisplayInfo(ELEVATOR_IDLE, 0, 0, 0, FiveFloorElevator->Floor)
}
}
else if (FiveFloorElevator->State == going_down)
{
    if (CustomerDownQueue[FiveFloorElevator->Floor]->CustomerNumber)
        CallDown[FiveFloorElevator->Floor] = 1;
    else
        CallDown[FiveFloorElevator->Floor] = 0;
    for (cnt = FiveFloorElevator->Floor - 1; cnt >= 0; cnt--)
    {
        if (FiveFloorElevator->CallCar[cnt])
        {
            FiveFloorElevator->activity = ACCELERATING;

```

```

        FiveFloorElevator->activity_timer = accelerate_time;
        DisplayInfo(ELEVATOR_ACCELERATING, 0, 0, 0, FiveFloorElev
        break;
    }
}
if (cnt == -1)
{
    FiveFloorElevator->activity = IDLE;
    FiveFloorElevator->activity_timer = elevator_max_wait_time;
    FiveFloorElevator->State = idle;
    DisplayInfo(ELEVATOR_IDLE, 0, 0, 0, FiveFloorElevator->Floor)
}
}
else
    ;
break;
}
case ACCELERATING:
{
    FiveFloorElevator->activity = MOVING;
    if (FiveFloorElevator->State == going_up)
        FiveFloorElevator->activity_timer = up_wait_time;
    else
        FiveFloorElevator->activity_timer = down_wait_time;
    DisplayInfo(ELEVATOR_MOVING, 0, 0, 0, FiveFloorElevator->Floor);
    break;
}
case MOVING:
{
    if (FiveFloorElevator->State == going_up)
    {
        if (FiveFloorElevator->Floor < MAXFLOOR)
        {
            ++(FiveFloorElevator->Floor);

```

```

if (FiveFloorElevator->Floor == MAXFLOOR)
{
    FiveFloorElevator->StateChange = 1; //状态需要改变
    FiveFloorElevator->activity = SLOWING;
    FiveFloorElevator->activity_timer = up_slow_time;
    DisplayInfo(ELEVATOR_SLOWING, 0, 0, 0, FiveFloorElevator->Floor);
}
else
{
    // signal默认值是1
    for (int i = FiveFloorElevator->Floor + 1; i <= MAXFLOOR; i++)
    {
        if (!(CallUp[i] == 0 && CallDown[i] == 0 && FiveFloorElevator->Floor == i))
        {
            signal = 0;
            break;
        }
    }
    signal = 1;
    if (CallUp[FiveFloorElevator->Floor] == 1 || FiveFloorElevator->Floor == MAXFLOOR)
    {
        if (CallDown[FiveFloorElevator->Floor] == 1 && signal == 0)
        {
            FiveFloorElevator->StateChange = 1; //状态需要改变
        }
        FiveFloorElevator->activity = SLOWING;
        FiveFloorElevator->activity_timer = up_slow_time;
        DisplayInfo(ELEVATOR_SLOWING, 0, 0, 0, FiveFloorElevator->Floor);
    }
    else
    {
        FiveFloorElevator->activity_timer = up_wait_time;
        DisplayInfo(ELEVATOR_MOVING, 0, 0, 0, FiveFloorElevator->Floor);
    }
}

```

```

        }
    }
    //下面是已经到达最高层的情况
    else
    {
        FiveFloorElevator->StateChange = 1; //状态需要改变
        FiveFloorElevator->activity = SLOWING;
        FiveFloorElevator->activity_timer = up_slow_time;
        DisplayInfo(ELEVATOR_SLOWING, 0, 0, 0, FiveFloorElevator->Flo
    }
}
else if (FiveFloorElevator->State == going_down)
{
    if (FiveFloorElevator->Floor > 0)
    {
        --(FiveFloorElevator->Floor);
        if (FiveFloorElevator->Floor == 0)
        {
            FiveFloorElevator->StateChange = 1; //状态需要改变
            FiveFloorElevator->activity = SLOWING;
            FiveFloorElevator->activity_timer = down_slow_time;
            DisplayInfo(ELEVATOR_SLOWING, 0, 0, 0, FiveFloorElevator->
        }

        else
        {
            for (int i = FiveFloorElevator->Floor - 1; i >= 0; i--)
            {
                if (!(CallUp[i] == 0 && CallDown[i] == 0 && FiveFloor
                {
                    signal = 0;
                    break;
                }
            }
            signal = 1;
        }
    }
}

```

```

    }
    if (CallDown[FiveFloorElevator->Floor] == 1 || FiveFloorE
    {
        if (CallUp[FiveFloorElevator->Floor] == 1 && signal)
        {
            FiveFloorElevator->StateChange = 1; //状态需要改变
        }
        FiveFloorElevator->activity = SLOWING;
        FiveFloorElevator->activity_timer = down_slow_time;
        DisplayInfo(ELEVATOR_SLOWING, 0, 0, 0, FiveFloorEleva
    }
    else
    {
        FiveFloorElevator->activity_timer = down_wait_time;
        DisplayInfo(ELEVATOR_MOVING, 0, 0, 0, FiveFloorEleva
    }
}
//下面是电梯到达最低层的情况
else
{
    FiveFloorElevator->StateChange = 1; //状态需要改变
    FiveFloorElevator->activity = SLOWING;
    FiveFloorElevator->activity_timer = down_slow_time;
    DisplayInfo(ELEVATOR_SLOWING, 0, 0, 0, FiveFloorElevator->Flo
}
}
else
;
break;
}
case SLOWING:
{
    //进到SLOWING下一个活动就是OPENING

```

```

FiveFloorElevator->CallCar[FiveFloorElevator->Floor] = 0;
FiveFloorElevator->activity = OPENING;
FiveFloorElevator->activity_timer = open_close_door_time;
DisplayInfo(ELEVATOR_OPENING, 0, 0, 0, FiveFloorElevator->Floor);
if (FiveFloorElevator->State == going_up)
{
    if (FiveFloorElevator->StateChange) //状态需要改变
        FiveFloorElevator->State = going_down;
    FiveFloorElevator->StateChange = 0;
}
else if (FiveFloorElevator->State == going_down)
{
    if (FiveFloorElevator->StateChange)
        FiveFloorElevator->State = going_up;
    FiveFloorElevator->StateChange = 0;
}
else
    ;
}
case IDLE:
{
    if (FiveFloorElevator->Floor == 1)
    {
        //在一层的话保持IDLE等300t
        FiveFloorElevator->activity_timer = elevator_max_wait_time;
    }
    else
    {
        //在别的层需要返回到一层IDLE
        if (FiveFloorElevator->Floor > 1)
            FiveFloorElevator->State = going_down;
        else
            FiveFloorElevator->State = going_up;
        FiveFloorElevator->CallCar[1] = 1;
    }
}

```

```

        FiveFloorElevator->activity = ACCELERATING;
        FiveFloorElevator->activity_timer = accelerate_time;
        DisplayInfo(ELEVATOR_ACCELERATING, 0, 0, 0, FiveFloorElevator->F)
    }
    break;
}
}
}
}

```

这里的DisplayInfo函数实际上是根据现在状态输出对应信息(人和乘客的信息)的函数。其中最复杂的是case MOVING部分，我分别讨论了电梯目前已经在最高层、最低层的情况，因为在这时电梯不可能继续上升(最高层)或者继续下降(最低层)，所以电梯的状态一定会发生改变。

同时，在进入case SLOWING这个活动之前要根据课本[实现提示]部分给的判断条件来判断电梯的状态(StateChange)是否要发生改变。在一开始写代码时我没有进行判断，而是让电梯状态发生改变，导致运行出来的结果不符合实际和[实现提示]给的提示。

其他的代码由于相对来说较容易，故不再赘述。

四.调试分析

- 本次实验遇到的难解决的Bug如下：
 - 电梯只有五层(0-4)但是在一开始运行的时候出现了电梯楼层大于4/小于0的情况
 - 电梯状态在不该改变的时候发生了改变(e.g.电梯里载着去更高楼层的乘客，但是电梯在某层楼停下开门之后进来目的地在更低楼层的乘客)

- 本次实验调试的困难之处：

- 由于每次都是随机数生成的数据，导致一次结果正确不能保证程序的正确性，需要多测很多次。而且测不同数据时出的bug也不一样，增加了调试的困难程度。
- 由于电梯的状态和活动较多，而且活动的时间较长，即使我在修改了活动时间(每个活动变为时间变为1/2)的情况下，打断点的方式也不能很快地找出bug。
- 最终我采用了直接用纸和笔模拟程序输出的结果来检验程序的正确性，根据bug可能出现的原因去找相对应的代码来做修改。

五.测试结果

Current time is 0
Customer1 comes to floor1 and wants to go to floor4
Other Info About Customer:Give up time:474 Inter time: 63
Current time is 0
FiveFloorElevator is changing IDLE to OPENING in floor1
Current time is 20
FiveFloorElevator door is open in floor1
Current time is 21
Customer1 enters into the FiveFloorElevator in floor1
Current time is 61
FiveFloorElevator is closing the door in floor1
Current time is 64
Customer2 comes to floor3 and wants to go to floor0
Other Info About Customer:Give up time:462 Inter time: 104
Current time is 82
FiveFloorElevator door is closed in floor1
Current time is 83
FiveFloorElevator is speeding up and going up from floor1
Current time is 99
FiveFloorElevator is moving from floor1 to floor2
Current time is 151
FiveFloorElevator is moving from floor2 to floor3
Current time is 169
Customer3 comes to floor4 and wants to go to floor1
Other Info About Customer:Give up time:328 Inter time: 86
Current time is 203
FiveFloorElevator is moving from floor3 to floor4
Current time is 255
FiveFloorElevator is slowing in floor4
Current time is 256
Customer4 comes to floor2 and wants to go to floor4
Other Info About Customer:Give up time:309 Inter time: 67
Current time is 270

FiveFloorElevator is opening the door in floor4
Current time is 270
FiveFloorElevator is speeding up and going down from floor4
Current time is 286
FiveFloorElevator is moving from floor4 to floor3
Current time is 324
Customer5 comes to floor3 and wants to go to floor1
Other Info About Customer:Give up time:486 Inter time: 58
Current time is 348
FiveFloorElevator is slowing in floor3
Current time is 372
FiveFloorElevator is opening the door in floor3
Current time is 372
FiveFloorElevator is speeding up and going down from floor3
Current time is 383
Customer6 comes to floor4 and wants to go to floor0
Other Info About Customer:Give up time:485 Inter time: 87
Current time is 388
FiveFloorElevator is moving from floor3 to floor2
Current time is 450
FiveFloorElevator is moving from floor2 to floor1
Current time is 471
Customer7 comes to floor1 and wants to go to floor0
Other Info About Customer:Give up time:372 Inter time: 58
Current time is 497
Customer3 gives up in floor4
Current time is 512
FiveFloorElevator is slowing in floor1
Current time is 526
Customer2 gives up in floor3
Current time is 530
Customer8 comes to floor3 and wants to go to floor1
Other Info About Customer:Give up time:314 Inter time: 102
Current time is 536

FiveFloorElevator is opening the door in floor1
Current time is 565
Customer4 gives up in floor2
Current time is 633
Customer9 comes to floor1 and wants to go to floor3
Other Info About Customer:Give up time:301 Inter time: 92
Current time is 726
Customer10 comes to floor3 and wants to go to floor2
Other Info About Customer:Give up time:460 Inter time: 64
Current time is 791
Customer11 comes to floor1 and wants to go to floor4
Other Info About Customer:Give up time:419 Inter time: 107
Current time is 810
Customer5 gives up in floor3
Current time is 837
FiveFloorElevator door is open in floor1
Current time is 838
Customer7 enters into the FiveFloorElevator in floor1
Current time is 844
Customer8 gives up in floor3
Current time is 868
Customer6 gives up in floor4
Current time is 878
FiveFloorElevator is closing the door in floor1
Current time is 899
Customer12 comes to floor3 and wants to go to floor1
Other Info About Customer:Give up time:327 Inter time: 106
Current time is 899
FiveFloorElevator door is closed in floor1
Current time is 900
FiveFloorElevator is speeding up and going down from floor1
Current time is 916
FiveFloorElevator is moving from floor1 to floor0
Current time is 934

Customer9 gives up in floor1
Current time is 978
FiveFloorElevator is slowing in floor0
Current time is 1002
FiveFloorElevator is opening the door in floor0
Current time is 1002
FiveFloorElevator is speeding up and going up from floor0
Current time is 1006
Customer13 comes to floor0 and wants to go to floor3
Other Info About Customer:Give up time:345 Inter time: 64
Current time is 1018
FiveFloorElevator is moving from floor0 to floor1
Current time is 1070
FiveFloorElevator is slowing in floor1
Current time is 1071
Customer14 comes to floor0 and wants to go to floor3
Other Info About Customer:Give up time:464 Inter time: 88
Current time is 1085
FiveFloorElevator is opening the door in floor1
Current time is 1160
Customer15 comes to floor2 and wants to go to floor1
Other Info About Customer:Give up time:361 Inter time: 57
Current time is 1186
Customer10 gives up in floor3
Current time is 1210
Customer11 gives up in floor1
Current time is 1218
Customer16 comes to floor3 and wants to go to floor2
Other Info About Customer:Give up time:485 Inter time: 79
Current time is 1226
Customer12 gives up in floor3
Current time is 1298
Customer17 comes to floor4 and wants to go to floor1
Other Info About Customer:Give up time:337 Inter time: 90

Current time is 1351
Customer13 gives up in floor0
Current time is 1386
FiveFloorElevator door is open in floor1
Current time is 1389
Customer18 comes to floor3 and wants to go to floor2
Other Info About Customer:Give up time:376 Inter time: 68
Current time is 1427
FiveFloorElevator is closing the door in floor1
Current time is 1448
FiveFloorElevator door is closed in floor1
Current time is 1449
FiveFloorElevator is idle in floor1
Current time is 1450
FiveFloorElevator is changing IDLE to ACCELERATING in floor1
Current time is 1458
Customer19 comes to floor3 and wants to go to floor0
Other Info About Customer:Give up time:424 Inter time: 64
Current time is 1465
FiveFloorElevator is moving from floor1 to floor2
Current time is 1517
FiveFloorElevator is slowing in floor2
Current time is 1521
Customer15 gives up in floor2
Current time is 1523
Customer20 comes to floor4 and wants to go to floor1
Other Info About Customer:Give up time:363 Inter time: 58
Current time is 1532
FiveFloorElevator is opening the door in floor2
Current time is 1532
FiveFloorElevator is speeding up and going down from floor2
Current time is 1535
Customer14 gives up in floor0
Current time is 1548

FiveFloorElevator is moving from floor2 to floor1
Current time is 1582
Customer21 comes to floor4 and wants to go to floor2
Other Info About Customer:Give up time:443 Inter time: 75
Current time is 1610
FiveFloorElevator is slowing in floor1
Current time is 1634
FiveFloorElevator is opening the door in floor1
Current time is 1635
Customer17 gives up in floor4
Current time is 1658
Customer22 comes to floor3 and wants to go to floor4
Other Info About Customer:Give up time:458 Inter time: 79
Current time is 1703
Customer16 gives up in floor3
Current time is 1738
Customer23 comes to floor0 and wants to go to floor2
Other Info About Customer:Give up time:385 Inter time: 98
Current time is 1765
Customer18 gives up in floor3
Current time is 1837
Customer24 comes to floor1 and wants to go to floor3
Other Info About Customer:Give up time:479 Inter time: 86
Current time is 1882
Customer19 gives up in floor3
Current time is 1886
Customer20 gives up in floor4
Current time is 1924
Customer25 comes to floor1 and wants to go to floor3
Other Info About Customer:Give up time:332 Inter time: 68
Current time is 1935
FiveFloorElevator door is open in floor1
Current time is 1976
FiveFloorElevator is closing the door in floor1

Current time is 1993
Customer26 comes to floor2 and wants to go to floor1
Other Info About Customer:Give up time:374 Inter time: 69
Current time is 1997
FiveFloorElevator door is closed in floor1
Current time is 1998
FiveFloorElevator is idle in floor1
Current time is 1999
FiveFloorElevator is changing IDLE to OPENING in floor1
Press any key to continue . . .

六.总结与收获

- 通过本次实验熟练地掌握了栈和队列的相关操作
- 学会了用C语言实现一个FSM
- 对于大工程的实现，有效地将不同功能写在不同的文件中，清晰易懂
- 千行代码不仅锻炼了写代码的能力，也锻炼了debug的能力