一、 实验目的

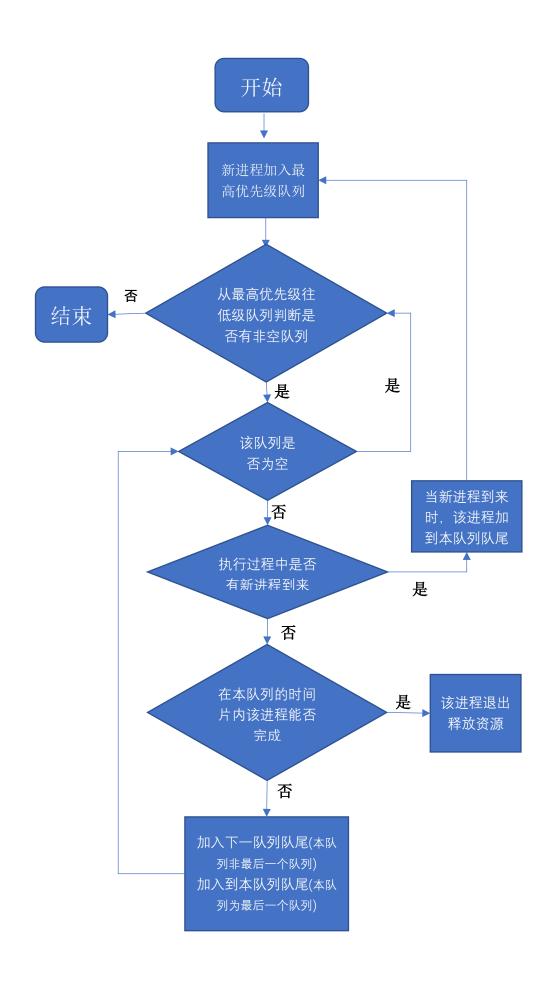
使用 C++模拟多级反馈队列调度算法

二、 数据结构及说明

定义一个 PCB 的结构体,代表每个进程,里面包含了进程名、到达时间、需要服务时间、剩余服务时间、完成时间、周转时间、带权周转时间

```
struct PCB {
    string name;
    int arrive_time = 0;
    int need_time = 0;
    int left_time = 0;
    int finish_time = 0;
    int turnaround_time = 0;
    double turn_weight_time = 0.0;
    };
    定义一个 Queue 结构体,代表一个就绪队列,包含进程队列、时间片
struct Queue {
    queue<PCB> process;
    int time = 0;//时间片
    };
    定义一个 Queue 的结构体数组,即多级队列
```

三、流程图



四、 实验结果

🥘 result.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

进程名	服务时间	到达时间	完成时间	周转时间	带权周转时间	
E	3	7	17	10	3.33	
\mathbf{C}	6	5	30	25	4.17	
D	5	6	32	26	5.20	
\mathbf{A}	10	0	38	38	3.80	
\mathbf{F}	12	10	51	41	3.42	
В	60	4	96	92	1.53	

初始数据:

```
PCB A = { "A",0,10,10,0,0,0.0 };

PCB B = { "B",4,60,60,0,0.0 };

PCB C = { "C",5,6,6,0,0,0.0 };

PCB D = { "D",6,5,5,0,0,0.0 };

PCB E = { "E",7,3,3,0,0,0.0 };

PCB F = { "F",10,12,12,0,0,0.0 };
```

初始化每个进程依次为: 进程名、到达时间、服务时间、剩余服务时间、完成时间、周转时间、带权周转时间

五、 结果分析

第0队列,时间片:1

进程没有执行完毕,需要添加到下一队列队尾,进程名:A 剩余服务时间:9

第1队列,时间片:2

进程没有执行完毕,需要添加到下一队列队尾,进程名:A 剩余服务时间:7

第2队列,时间片:4

进程没有执行完毕,需要添加到本队列队尾,进程名:A剩余服务时间:6

第 0 队列,时间片:1

进程没有执行完毕,需要添加到下一队列队尾,进程名:B剩余服务时间:59

第0队列,时间片:1

进程没有执行完毕,需要添加到下一队列队尾,进程名:C剩余服务时间:5

第0队列,时间片:1

进程没有执行完毕,需要添加到下一队列队尾,进程名:D 剩余服务时间: 4

第0队列,时间片:1

进程没有执行完毕,需要添加到下一队列队尾,进程名:E 剩余服务时间: 2 第1队列,时间片:2 进程没有执行完毕,需要添加到下一队列队尾,进程名:B 剩余服务时间:57 第0队列,时间片:1 进程没有执行完毕,需要添加到下一队列队尾,进程名:F剩余服务时间:11 第1队列,时间片:2 进程没有执行完毕,需要添加到下一队列队尾,进程名:C 剩余服务时间: 3 第1队列,时间片:2 进程没有执行完毕,需要添加到下一队列队尾,进程名:D 剩余服务时间: 2 此时时间: 17 服务完成并退出: E 第1队列,时间片:2 进程没有执行完毕,需要添加到下一队列队尾,进程名:F剩余服务时间:9 第2队列,时间片:4 进程没有执行完毕,需要添加到下一队列队尾,进程名:A 剩余服务时间:2 第2队列,时间片:4 进程没有执行完毕,需要添加到下一队列队尾,进程名:B剩余服务时间:53 此时时间: 30 服务完成并退出: C 此时时间: 32 服务完成并退出: D 第2队列,时间片:4 进程没有执行完毕,需要添加到下一队列队尾,进程名:F剩余服务时间:5 此时时间: 38 服务完成并退出: A 第3队列,时间片:8 进程没有执行完毕,需要添加到下一队列队尾,进程名:B 剩余服务时间: 45 -____

此时时间: 51

服务完成并退出: F

第4队列,时间片:16

进程没有执行完毕,需要添加到本队列队尾,进程名:B 剩余服务时间: 29

第4队列,时间片:16

进程没有执行完毕,需要添加到本队列队尾,进程名:B 剩余服务时间:13

此时时间: 96

服务完成并退出: B
