TASK SCHEDULER IN C++

Patka Zsolt-András
-2018-

Contents

1. Task

```
Implementing task scheduling algorithms with an object oriented programming
mindset.
Algorithms implemented: FCFS (First come, first served), RR (Round Robin).
Possible expansions: SJF (Shortest Job First), SRTF (Shortest Remaining
Time First).

Example input: From a text file
RR 2         The name of the scheduling alg., time slice (In the case of RR)
4                                                    Task to be scheduled
0 5                                    First task: starts at 0, 5 CPU burst
4 5                                   Second task: starts at 4, 5 CPU burst
5 3                                    Third task: starts at 5, 3 CPU burst
6 1                                    Forth task: starts at 6, 1 CPU burst

Example output: In the form of a Gantt diagramm, to the stdout.
```

Tasks	Start, CPU burst													
A	0, 5	x	x			x	x						x		
B	4, 5			x	x			x	x					x	
C	5, 3									x	x				x
D	6, 1											x			
Timestep		1	2	3	4	5	6	7	8	9	10	11	12	13	14

2. Aim of the project

The aim of this project is to implement scheduling algorithms. The user may specify multiple scheduling tasks in the input file. The program will write the output for these scheduling tasks to stdout, one after the other.
The program validates user inputs, in the case of incorrect inputs, the program exits with an error message.
The maximum number of tasks and the maximum CPU burst value is going to be specified. If these requirements are not met, then the program exists with an error message
Testing: the program was tested for a wide range of inputs, among these there were incorrect inputs as well.

3. Plan

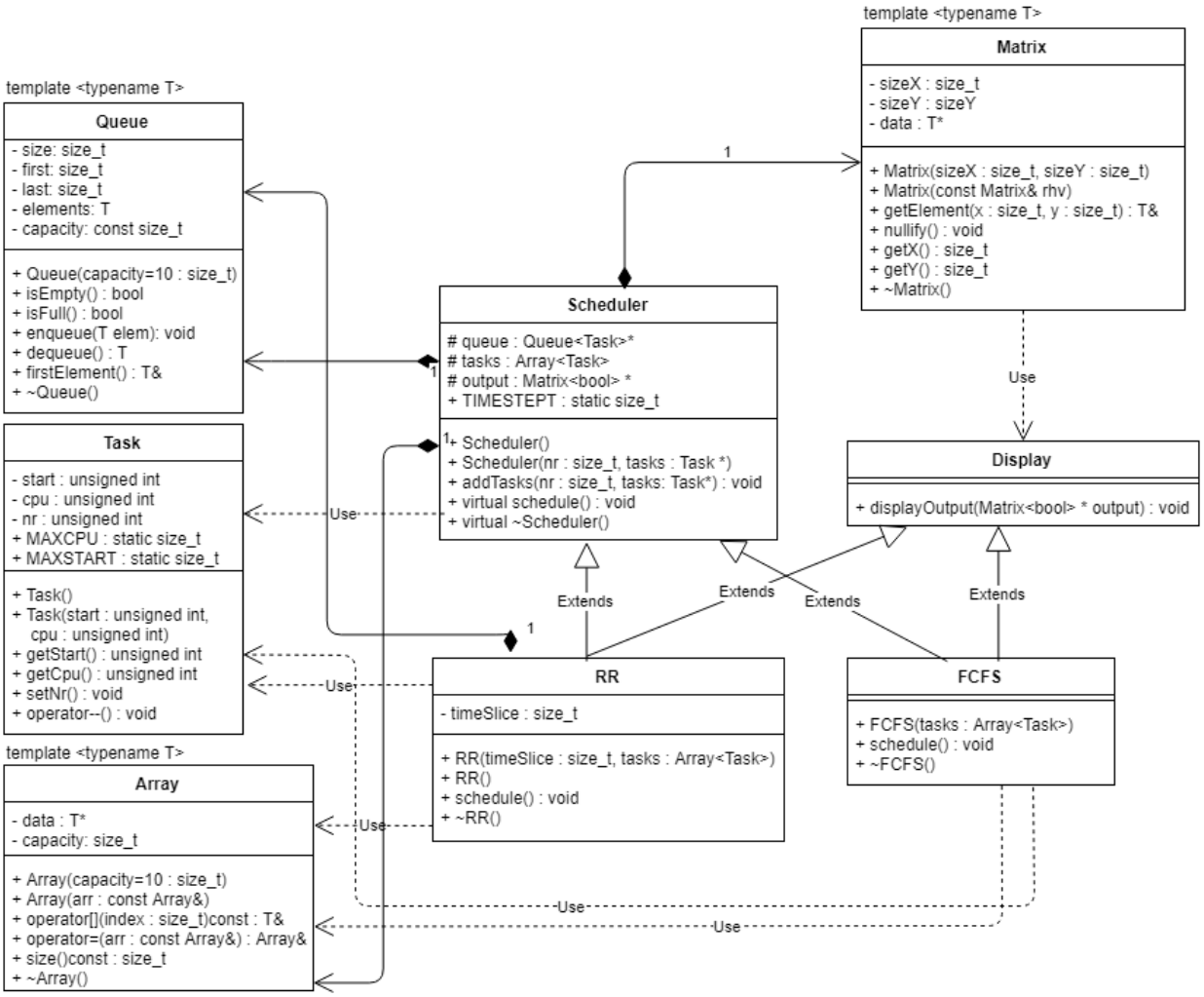
Multiple classes and a test program was needed to be planned.

3.1. UML diagram

Three generic collections are needed:

- Queue: FIFO data structure, needed for the scheduling algorithms
- Array: Simple generic array, used for storing Task objects
- Matrix: Generic two dimensional array represented in memory as a one dimensional array.

In the Task class the tasks's start time and cpu burst are stored.
The Scheduler class is the base class of all of the classes which implement a scheduling algorithm.
The Display class is also the base class of all other scheduling classes, this class displays the output. This class can be exchanged for another one (which also works with a Matrix<bool> object) if the way the output is displayed needs to be changed.
The FCFS class implements the First Come First Served algorithm.
The RR class implements the Round Robin algorithm. This class has an aditional data member: time slice, this is needed for implementing the algorithm.



3.2. Algorithms

3.2.1. FCFS (First Come First Served)

Runs the tasks in the order in which they arrive. When a task arrives it goes to the end of the queue. Runs the task at the front of the queue then dequeues it.

3.2.2. RR (Round Robin)

Only lets the task run for a certain period of time, then it goes into waiting state and the next task gets run.

3.2.3. Queue

A queue implemented with the help of a circular array. If we get to the end of the array but there is still place in the queue, then we put the next element to the beginning of the array.

3.2.4. Matrix

Two dimensional array implemented as a one dimensional array. It can be indexed the same way as a traditional two dimensional array
Index conversion: t[i][j] -> t[i*row_num + j]

3.2.5. Test program's algorithms

The test program reads the scheduler type and the tasks from an input file. After that it runs the algorithms and displays the output to stdout.