COSC2430 Homework 4: Queue

1. Introduction

Operating systems have a large number of processes including CPU bursts interspersed between I/O requests. To maximize the performance, a process is restricted to run no longer than a quantum time and stays in the queue waiting for the other process to run.

You will create a C++ program to implement a basic scheduling system. The program will need to calculate the waiting time, start time, and end time of each process.

The purpose of this homework is to get students familiar with Queue. Please name the folder on the server as “hw4”, and start on the homework early.
SECTION 3 HAS ALL THE IMPORTANT INFORMATION YOU NEED
There will also be an outline of what the program should be doing at each step of the simulation

2. Input and Output

a. Input file

The input file contains:

i. Max time allowed in queue
ii. Number of total processes
iii. Number of servers
iv. List of execution time of each job

The number of processes and servers will not exceed 100.

b. Output file

The output is a single text file. Please follow the format below:

Max time: *value*
Number of process: *value*
Number of servers: *value*

Process: *value*
Execution time: *value*
Waiting time: *value*
Start time: *value*
End time: *value*

c. Examples

i. Example 1

input41.txt
2 3 2
3 1 5

output41.txt
Max time: 2
Number of processes: 3
Number of servers: 2

Process: 1
Execution time: 3
Wait Time: 0
Start Time: 0
End Time: 3

Process: 2
Execution time: 1
Wait Time: 0
Start Time: 0
End Time: 1

Process: 3
Execution time: 5
Wait Time: 1
Start Time: 1
End Time: 6

Linux Command:
./queue “input=input41.txt;output=output41.txt”
./queue input=input41.txt output=output41.txt

ii. **Example 2**

input42.txt
3 2 4
9 3

output42.txt
Max time: 3
Number of processes: 2
Number of servers: 4

Process: 1
Execution time: 9
Wait Time: 0
Start Time: 0
End Time: 9

Process: 2
Execution time: 3

Wait Time: 0
Start Time: 0
End Time: 3

Linux Command:
./queue “input=input42.txt;output=output42.txt”
./queue input=input42.txt output=output42.txt

iii. **Example 3**

input43.txt
7 26 3
66 4 3 2 4 5 33 33 44 5 2 4 2 2 4 2 5 44 33 33 5 4 2 3 4 66

output43.txt
Max time: 7
Number of processes: 26
Number of servers: 3

Process: 1
Execution time: 66
Wait Time: 77
Start Time: 0
End Time: 143

Process: 2
Execution time: 4
Wait Time: 0
Start Time: 0
End Time: 4

Process: 3
Execution time: 3
Wait Time: 0
Start Time: 0
End Time: 3

Process: 4
Execution time: 2
Wait Time: 3
Start Time: 3
End Time: 5

Process: 5
Execution time: 4
Wait Time: 4

Start Time: 4
End Time: 8

Process: 6
Execution time: 5
Wait Time: 5
Start Time: 5
End Time: 10

Process: 7
Execution time: 33
Wait Time: 67
Start Time: 7
End Time: 100

Process: 8
Execution time: 33
Wait Time: 70
Start Time: 8
End Time: 103

Process: 9
Execution time: 44
Wait Time: 79
Start Time: 10
End Time: 123

Process: 10
Execution time: 5
Wait Time: 14
Start Time: 14
End Time: 19

Process: 11
Execution time: 2
Wait Time: 15
Start Time: 15
End Time: 17

Process: 12
Execution time: 4
Wait Time: 17
Start Time: 17
End Time: 21

Process: 13
Execution time: 2
Wait Time: 17
Start Time: 17
End Time: 19

Process: 14
Execution time: 2
Wait Time: 19
Start Time: 19
End Time: 21

Process: 15
Execution time: 4
Wait Time: 19
Start Time: 19
End Time: 23

Process: 16
Execution time: 2
Wait Time: 21
Start Time: 21
End Time: 23

Process: 17
Execution time: 5
Wait Time: 21
Start Time: 21
End Time: 26

Process: 18
Execution time: 44
Wait Time: 80
Start Time: 23
End Time: 124

Process: 19
Execution time: 33
Wait Time: 75
Start Time: 23
End Time: 108

Process: 20

Execution time: 33
Wait Time: 79
Start Time: 26
End Time: 112

Process: 21
Execution time: 5
Wait Time: 30
Start Time: 30
End Time: 35

Process: 22
Execution time: 4
Wait Time: 30
Start Time: 30
End Time: 34

Process: 23
Execution time: 2
Wait Time: 33
Start Time: 33
End Time: 35

Process: 24
Execution time: 3
Wait Time: 34
Start Time: 34
End Time: 37

Process: 25
Execution time: 4
Wait Time: 35
Start Time: 35
End Time: 39

Process: 26
Execution time: 66
Wait Time: 81
Start Time: 35
End Time: 147

Linux Command:
./queue “input=input43.txt;output=output43.txt”
./queue input=input43.txt output=output43.txt

3. The Rules and Operations

You must implement Queue in this homework. Please read and trace with the explanation to understand the problem.

- If the process has 0 execution time, there will be no waiting time on other processes, and it will be immediately completed. The queue will switch to the next process.
- If all the servers are available, the next process in the queue will always start with the lowest number server.
- If all the servers finish at the same time, the lowest number server will eject the process first for it to be pushed back to the queue.

Let's look into example 41:

2 3 2
3 1 5

- The first line provides:
  - A maximum time: 2 seconds - the time each process will execute on a server.
  - The total number of processes: 3 - total processes need to be executed.
  - The number of servers: 2 - number of servers that can execute processes parallely.

- The second line provides:
  - List of execution time for each process: process #1 is 3 seconds, process #2 is 1 seconds, and process #3 is 5 seconds.

- The timeline in second will be demonstrated below:

TIME: 0
Server #1 is available
Server #2 is available
Putting Process #1 into Server #1 (Process #1 starts)
Server #1 is no longer available

TIME: 0
Server #1 isn't available
Server #2 is available
Putting Process #2 into Server #2 (Process #2 starts)
Server #2 is no longer available

TIME: 1
Server #1 isn't available
Server #2 isn't available
Putting Process #2 into COMPLETED (Process #2 ends)
Server #2 is now available

TIME: 1
Server #1 isn't available
Server #2 is available
Putting Process #3 into Server #2 (Process #3 starts)
Server #2 is no longer available

TIME: 2
Server #1 isn't available
Server #2 isn't available
Process #1 reaches max time. Putting Process #1 back into queue
Server #1 is now available

TIME: 2
Server #1 is available
Server #2 isn't available
Putting Process #1 into Server #1
Server #1 is no longer available

TIME: 3
Server #1 isn't available
Server #2 isn't available
Putting Process #1 into COMPLETED (Process #1 ends)
Server #1 is now available

TIME: 3
Server #1 is available
Server #2 isn't available
Process #3 reaches max time. Putting Process #3 back into queue
Server #2 is now available

TIME: 3
Server #1 is available
Server #2 is available
Putting Process #3 into Server #1
Server #1 is no longer available

TIME: 4
Server #1 isn't available
Server #2 is available

TIME: 5
Server #1 isn't available
Server #2 is available

Process #3 reaches max time. Putting Process #3 back into queue
Server #1 is now available

TIME: 5
Server #1 is available
Server #2 is available
Putting Process #3 into Server #1
Server #1 is no longer available

TIME: 6
Server #1 isn't available
Server #2 is available
Putting Process #3 into COMPLETED (Process #3 ends)
Server #1 is now available

Result:
Process 1 starts at 0, ends at 3, and waits 0
Process 2 starts at 0, ends at 1, and waits 0
Process 3 starts at 1, ends at 6, and waits 1 (the waiting time is when the process waits in the queue and does not execute starting from 0).

4. Requirements

Do **NOT** use the Queue in STL. Please create the Queue manually.
The main C++ problem will become the executable to be tested by the TAs. The result file should be written to another text file (output file), provided with the command line.

Homework is individual. Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources. Code that is copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc, will be treated as copy) will be detected and result in ”0” in this homework. The limit is 50% similarity. Here are some previous homework which have been found to copy each other (the main function has been deleted).

5. Turn in your homework

Homework 4 needs to be turned in to our Linux server, follow the link here
https://rizk.netlify.app/courses/cosc2430/2_resources/

Make sure to create a folder under your root directory, name it hw4 (name need to be lower case), only copy your code to this folder, no testcase or other files needed.

PS: This document may have typos, if you think something illogical, please email TAs for confirmation. Always update the newest version of the instruction from the drive every time you work on your program.