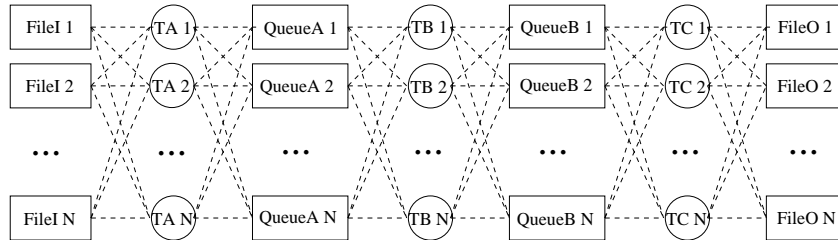


System and Device Programming

Examination Test – Programming Part 01 July 2016

Examination Time: 1h 45min. Evaluation. 18 marks.
Textbooks and/or course material allowed.

An office automation system is organized as represented by the following picture.



The Windows 32 system simulating the system is composed by:

- N input files FileI_i and N output files FileO_i .
- N threads TA_i , N threads TB_i and N threads TC_i .
- N FIFO queue QueueA_i and N FIFO queue QueueB_i .

where $i = 1, \dots, N$.

Every input file FileI_i (with $i = 1, \dots, N$) stores, in binary form, an undefined number of records. Each record includes a single 32-bit integer value n , and a sequence of n characters.

Each queue has a fixed length of size S .

Each one of the N threads TA_i (with $i = 1, \dots, N$) proceeds as follows:

1. It awaits for a random number of seconds (from 1 to 10).
2. It reads from one of the input files FileI_i (randomly selected) the next (not yet read) record.
3. It manipulates the record string by erasing all non alphabetic characters.
For example the record `23 123ab;-CaAbB56bcC??c(C)` will become `12 abCaAbBbcCcC`
4. It awaits for a random number of seconds (from 1 to 10).
5. It enqueues the same data read from the input file (but with the string manipulated) in a randomly selected QueueA FIFO queue.
6. It restarts the entire process from point (1).

Each thread TB_i (with $i = 1, \dots, N$), behaves **exactly** like thread TA_i with three main differences:

- It reads its input from a randomly selected QueueA_i FIFO queue (and not from a file).
- It manipulates the record string by transforming all lower case letters into upper case letters.
For example the record `12 abCaAbBbcCcC` will become `12 ABCAABBBCCCC`.
- It enqueues its output data on a randomly selected QueueB_i FIFO queue.

Each thread TC_i (with $i = 1, \dots, N$), behaves **exactly** like thread TA_i with three main differences:

- It reads its input from a randomly selected QueueB_i FIFO queue (and not from a file).
- It manipulates the record string by ordering all letters in ascending order.
For example the record `12 ABCAABBBCCCC` will become `12 AAABBBBCCCC`.
- It writes its output data on a randomly selected FileO_i file (and not to a queue).

The integer value N is received by the application on the command line. Input and output files have pre-defined names as indicated (i.e., FileI_i and FileO_i , where $i = 1, \dots, N$). The application has to synchronize all threads and perform all file and queue accesses in a proper way, i.e., with mutual exclusion whenever necessary. When all input files FileI_i have been manipulated all threads TA_i , TB_i and TC_i have to stop following a proper and clean procedure.

The way in which the program is organized, modularized and presented will be subject to evaluation. Concurrency has to be maximized.