System and Device Programming

Examination Test – Programming Part
23 July 2018

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

Gaussian filtering is used to blur images and remove noise and detail. The process is the following one.
An image is a matrix of pixels. For each pixel a single integer value represents its gray tone value. Given a filter, Gaussian filtering consists of averaging the value of each pixel of the image with the adjacent pixels averaged throughout the filter values. For example, the following is the discrete representation of two Gaussian filters, the one of size (3×3) (left-hand size) and the other of size (5×5) (right-hand size).

1	2	1
2	4	2
1	2	1

1/16

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

1/273

If we consider the following image (left-hand side) of (6×6) pixels, its convolution with the (3×3) previous filter delivers the image on the right-hand side.

	1	2	3	4	5	6
1	15	20	25	25	15	10
2	20	15	50	30	20	15
3	20	50	55	60	30	20
4	20	15	65	30	15	30
5	15	20	30	20	25	30
6	20	25	15	20	10	15

1	2	3	4	5	6
17	21	27	25	17	13
20	28	38	35	23	17
24	35	48	43	28	22
20	31	42	36	26	25
18	23	28	25	22	23
20	21	20	18	17	18

where pixel on row 5 and column 2 is computed as: $\frac{20 \cdot 1 + 15 \cdot 2 + 65 \cdot 1 + 15 \cdot 2 + 20 \cdot 4 + 30 \cdot 2 + 20 \cdot 1 + 25 \cdot 2 + 15 \cdot 1}{16} = 23$

Write a Windows-32 application able to:

- Receive 3 file names (`name1`, `name2`, and `name3`) and 2 integer values (n and m) on the command line.
- Run two group of threads running in sequence (first group then second group) in a circular way.
  - The first group includes n threads. These threads read the image from the first file and convolves it with the filter of size (3×3). The order in which pixels are filtered is free, but each pixel has to be manipulated only once by any thread belonging to the group. The resulting image is saved in file `name2`. When all pixels of the image have been filtered, all threads belonging to the first group go to sleep until the are awakened from threads belonging to the second group.
  - The second group includes m threads. Those have to run when all pixels belonging to the image have been convolved with the first filter, and file `name2` stores the entire convolved image. Similarly to threads belonging to the first group, threads belonging to the second group have to filter each pixel of file `name2` with the filter of size (5×5). Again the order in which the pixel are filtered is not important, but all pixels have to be filtered once and only once by any thread belonging to the group. The image must be saved in file `name3`. When all pixels have been manipulated, all threads belonging to the second group have to go to sleep (until they are re-activated by the threads belonging to the first group).

Notice, that:

- The candidate must write the main program, including the generation of all threads. It is not necessary to deal with the process termination. The values n and m are generally smaller than the number of pixels in one image.
- As a first approximation, the candidate may suppose that two functions, e.g., `filter3` and `filter5`, are already implemented to compute the convolution. These functions, given a pixel coordinates within the file or within the matrix, would compute the convolution with the (3×3) or the (5×5) filter, respectively.
- All images have a fixed size, equal to $(640 \cdot 480)$ pixels. Each pixel is represented as a gray tone using one single 16-bit integer value. Thus, all files store images as sequences of $(640 \cdot 480)$ 16-bit integer values.
- As the program repeatedly manipulates the input file `name1` and it generates the output file `name3`, the candidate may suppose `name1` is written and `name3` is read by another process (which does not have to be implemented by the candidate) while the implemented process does not work on that file.