

x86-to-C interface programming project


Started: Jul 21 at 12:27pm

Quiz Instructions

**Remember the academic honor pledge that you signed.

General directions:

- 1.) Submission is by pair, your lab partner. Required to only have 1 submission, deduction for multiple submissions.
- 2.) Submission via a zip file that contains 1) c file 2) asm file 3) text file that contains the assemble, compile, link, execute commands, and 4) screenshot that displays that your project is working properly 5) screenshot of the specification.
- 3.) Follow the directions found in the specifications.
- 5.) Take a screenshot of your project specification. **The first project specification is your project specification regardless of the attempts.**


Question 1 0 pts
Introduction <p>A grayscale image is internally represented as a 2 dimensional array. Size is defined by the number of pixels in a row (width), and number of pixels in a column (height). Each cell represents the pixel value. Pixel values in grayscale are represented as white, black, and shades of gray using different numerical data type representation depending on the library used.</p> <p>Some Image Processing Libraries represent grayscale images as having single float based pixel values between 0 to 1: 0 for black, 1 for white, 0.25 lighter gray, 0.75 darker gray, and all other gray values in between.</p> <p>Image =</p> <p>0.25, 0.35, 0.45, 0.33</p> <p>0.55, 0.65, 0.75, 0.33</p>

0.85, 0.95, 0.15, 0.33

Some Libraries represent using 8 bit unsigned integer. 0 for black, 255 for white, 64 lighter gray, 191 darker gray, and all other gray values in between.

Image =

64, 89, 114, 84

140, 166, 191, 84

216, 242, 38, 84

To enable using different image processing libraries simultaneously, conversion from one format to the other has to be done. **Perform Mapping from *single precision* float based grayscale to uint8 based integer representation.** Each pixel is calculated using ratio and proportion equation

$$\frac{f}{i} = \frac{1}{255}$$

Where f is the float based pixel value and i is the integer based pixel value

Task

Implement a program that converts the grayscale image representation from float based to integer based using C and x86 assembly language. Implement the function **imgCvtGrayFloatToInt()**.

*Required to use functional scalar SIMD registers

*Required to use functional scalar SIMD floating-point instructions

Input: height, width, single float pixel values

Example.

3 4

0.25 0.35 0.45 0.33

0.55 0.65 0.75 0.33

0.85 0.95 0.15 0.33

Output: Integer pixel values

Example.

64, 89, 114, 84

140, 166, 191, 84

216, 242, 38, 84

Note:

- 1.) C is responsible for: collecting the inputs, allocating memory spaces for the images, and printing the outputs.
- 2.) Function implemented in assembly is responsible for calculating and converting the data type from the input single float pixels into the output int pixels.
- 3.) Time the asm function only for input image size $width*height = \{10*10, 100*100, \text{ and } 1000*1000\}$. If 1000*1000 is impossible, you may reduce it to the point your machine can support. You may use a random number generator to generate pixel values for the input.
- 4.) You must run at least 30 times to get the average execution time.
- 5.) For the data, you may initialize each pixel with the same or different random value.
- 6.) You will need to check the correctness of your output.
- 7.) Output in GitHub (make sure that I can access your Github):
 - a.) Github readme containing the following (C and x86-64):
 - i.) execution time and short analysis of the performance
 - ii.) Take a screenshot of the program output with the correctness check.
 - iii.) short videos (5-10mins) showing your source code, compilation, and execution of the C and x86-64 program
 - b.) Submit all files needed to run your project. (source code: C, x86-64, and all other required files) for others to load and execute your program.

Rubric:

C main program with initialization and correct call/passing parameters to C and x86-64	25
Correct output (x86-64)	45
Performance result	20

Videos

10

not following instructions

-10/
instruction

Note: No usage of functional scalar SIMD registers and scalar SIMD instructions, function not in assembly

grade = 0

Upload

Choose a File



Question 2 90 pts

Place your GitHub link here.

Edit View Insert Format Tools Table

12pt

Paragraph

p



0 words

</>

+

-



Quiz saved at 5:11pm

Submit Quiz