x86-to-C interface programming project

Started: Jul 20 at 11:06pm

Quiz Instructions

**Remember the academic honor pledge that you signed.

**********READ MY ANNOUNCEMENT**********

General directions:

1.) Only one member will submit. The other member will be assigned the "[partner grade] C-assembly interface" assignment.

2.) Ignore the file upload button in the specification question.

3.) Submission via GitHub. Place your Github link in the next question. Make sure I can access your GitHub.

4.) Write the name of all group members in the Github documentation.

4.) Follow the directions found in the specifications.

5.) Take a screenshot of your project specification for reference purposes. **The first project specification is your project specification regardless of the attempts.**

6.) Don't forget to submit the peer evaluation.

Question 1 0 pts

Introduction

A car's peformance is usually measured using an acceleration test with a 1KM distance. Some cars are measured from a cold start (they start at 0KM/H), some are tested from a hot start (the car is running at a stable speed and then accelerates).

Factors that are needed to compute acceleration (m/s^2):

1. Initial Velocity (V_i) - starting speed.

2. Final Velocity (V_f) - final speed at the target distance (1KM).

3. Time (T) - the total time consumed to reach 1KM.

The formula for acceleration is:

Acceleration = $(V_f - V_i)/T$

Sample Computation:

V_i =62.5 KM/H

V_f = 0.0 KM/H

T = 10.1 s

Acceleration = (62.5KM/H - 0.0KM/H) / 10.1s

= (62.5KM/H) / 10.1s

= Convert KM/H to m/s

= ([62.5KM/H * 1000M/H] * 1m/3600s)/ 10.1s

= (17.36M/s) / 10.1s

= 1.7188 m/s^2

= convert to int (1.7188 m/s^2) = 2 m/s^2

Task

Implement a program that computes for the acceleration of multiple cars stored in a Y by 3 matrix. Where Y is the number of cars. All inputs are in single floating point.

The output acceleration for each will be converted into Integers.

Each row will indicate Initial Velocity (in KM/H), Final Velocity (in KM/H), and Time (in Seconds).

Sample Matrix:

0.0, 62.5, 10.1

60.0, 122.3, 5.5

30.0, 160.7, 7.8

*Required to use functional scalar SIMD registers

*Required to use functional scalar SIMD floating-point instructions

Input: Matrix Rows, single float matrix values

Example.

3

0.0, 62.5, 10.1

60.0, 122.3, 5.5

30.0, 160.7, 7.8

Output: Integer acceleration values (m/s^2)

Example.

2

3

5

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 converting velocity to m/s, calculating acceleration and converting the data type from the input single float into the output integer.

3) Time the asm function only for input Y size = {10, 100, 1000, 10000}. If 10000 is impossible, you may reduce it to the point your machine can support. You may use a random number generator to generate 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 input 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
Video	10
not following instructions	-10/instructions
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 B I U A T²