

Отчёт по ИДЗ №3

ФИО: Сайфутдинов Рафаэль Рустамович

Группа: БПИ216

Вариант задания: 30

Условие задания:

30. Разработать программу численного интегрирования функции $y = a + b * x^3$ (задаётся действительными числами a, b) в определённом диапазоне целых (задаётся так же) методом трапеций (точность вычислений = 0.0001).

Для удобства будем считать, что действительные числа a и b могут лежать в диапазоне $[-100;100]$ (так же, как и целые границы)

Тесты, демонстрирующие проверку программ:

Подготовлено 5 тестов, покрывающих достаточное количество возможных результатов работы программы.

На вход подаются числа a и b , а затем целые значения левой и правой границы.

Тест №1:

Команда: `./main.exe < Tests/test1.in`

Входные данные: `1 1 -1 1`

Ожидаемые выходные данные: `2.000000`

Фактические выходные данные: `2.000000`

Тест №2:

Команда: `./main.exe < Tests/test2.in`

Входные данные: `3.1415 2 -5 2`

Ожидаемые выходные данные: `-282.509500`

Фактические выходные данные: `-282.509500`

Тест №3:

Команда: `./main.exe < Tests/test3.in`

Входные данные: `2.5 150 5 3`

Ожидаемые выходные данные: Числа a и b должны быть в диапазоне $[-100;100]$

Фактические выходные данные: Числа a и b должны быть в диапазоне $[-100;100]$

Тест №4:

Команда: `./main.exe < Tests/test4.in`

Входные данные: `1 1 3 1`

Ожидаемые выходные данные: Левая граница должна быть не больше правой

Фактические выходные данные: Левая граница должна быть не больше правой

Тест №5:

Команда: `./main.exe < Tests/test5.in`

Входные данные: `0 3 2 2`

Ожидаемые выходные данные: `0.000000`

Фактические выходные данные: `0.000000`

Исходные тексты программ на языке C можно посмотреть, открыв файлы [main.c](#), [read.c](#), [integrate.c](#) соответственно.

Тексты финальных программ на языке ассемблера, полученных после компиляции и с расширенными комментариями и оптимизацией можно посмотреть, открыв файл [main.S](#), [read.S](#), [integrate.S](#) соответственно.

Тексты программ на языке ассемблера, полученных после компиляции и с расширенными комментариями без оптимизации можно посмотреть, открыв файл [main_without_optimization.S](#), [read_without_optimization.S](#), [integrate_without_optimization.S](#) соответственно.

Текст программы на ассемблере без комментариев, полученный сразу после компиляции программы на языке C, можно посмотреть, открыв файл [main_before.S](#), [read_before.S](#), [integrate_before.S](#) соответственно.

(Я не стал вставлять сюда тексты программ, чтобы не «захламлять» PDF файл)

Информация, подтверждающая выполнение задания в соответствии требованиями на **оценку 6-7**:

- Программа на ассемблере была скомпилирована при помощи соответствующих **аргументов командной строки** (`gcc -S -masm=intel -fno-asynchronous-unwind-tables -fno-jump-tables -fno-stack-protector -fno-exceptions main.c -o main.s`), а также за счёт **ручного** редактирования исходного текста ассемблерного кода программы;
- Программа на ассемблере представлена в виде **трёх единиц компиляции**: `main.S`, `read.S`, `integrate.S`
- В реализованной программе **использованы функции** `read` и `integrate` **с передачей данных через параметры**;
- Добавлены исчерпывающие **комментарии**, поясняющие эквивалентное представление переменных в программе на C, а также описывающие передачу фактических параметров и перенос возвращаемого результата в функциях;
- Представлен **набор тестов**, на которых **корректно** отработала программа как на языке C, так и на ассемблере;
- Размер объектного файла на ассемблере, полученного после компиляции на C, **больше** объектного файла модифицированной программы, использующей регистры:
 - `main.O` – на 56 битов
 - `read.O` – на 80 битов
 - `integrate.O` – на 64 бита
- Максимально использованы регистры процессора вместо локальных переменных (и не только), чтобы увеличить производительность. Подробнее:
 - **Убраны** команды `por`, которые буквально «ничего не делают»;
 - Операция «зануления» регистра **оптимизирована** при помощи `xor`;

- **Использование** «нестираемых» 4-х байтовых **регистров** r12d, r13d, r14d вместо локальных int переменных left, right и других;
- **Использование регистров** xmm3, xmm4, xmm5, xmm6, xmm7 вместо локальных double переменных a, b, l, result и других;
- **Удаление** ненужных операций mov для перемещения значения из одного регистра на другой, которые используют сторонние регистры в качестве «посредника» (помечено комментарием «Оптимизировано»)
- Представлены **тестовые файлы** .in и .out