

# Домашнее задание 3 по архитектуре вычислительных систем

Вариант 33

Выполнил Чугунов Андрей БПИ213

# Условие

Разработать программу, определяющую корень уравнения  $x^3 - 0.5x^2 + 0.2x - 4 = 0$  методом половинного деления с точностью от 0,001 до 0,00000001 в диапазоне [1;3]. Если диапазон некорректен, то подобрать корректный диапазон.

# Реализация

С помощью стороннего сервиса я нашел корень уравнения, который примерно равен 1.7. Это значит, что диапазон корректный. Точность в программе обозначена за "e", а ее значение считывается с клавиатуры с помощью `scanf`. Выводить будем 8 знаков после запятой.

# Состав репозитория

В репозитории находятся 2 папки: programs и tests. В папке programs находятся 3 программы:

1. main.c - си программа
2. main.s - откомпилированная си программа: gcc -masm=intel -fno-asynchronous-unwind-tables -fno-jump-tables -fno-stack-protector -fno-exceptions ./main.c -S -o ./main.s.
3. refactor.s - отредактированная программа, которая была получена так: gcc -masm=intel -fno-asynchronous-unwind-tables -fno-jump-tables -fno-stack-protector -fno-exceptions ./main.c -S -o ./refactor.s. Данная программа прокомментирована.

В папке test-results находится скриншот с тестами и результатами

# Сравнение программ

Используемые опции компиляции описаны в "Состав репозитория".

Программа refactor.s была получена путем ручного редактирования кода. Почти все переменные, которые изначально хранились на стеке, теперь хранятся в регистрах. Одна локальная переменная (t) хранится в памяти. Таким образом было использовано множество регистров, а количество обращений в память сильно уменьшилось. Также стало меньше строк, из 143 получилось 95. Размер refactor.o = 4.61 KB; main.o = 5.02 KB. В качестве изменений можно отметить секцию .rodata. Туда были добавлены нужные константы, тогда как в исходной программе компилятор расположил их по другому.

# Тестовые прогоны

Всего было проведено 4 теста.

1.  $e=0.000000001$
2.  $e=0.001$
3.  $e=0.00001$
4.  $e=0.0000005$

На все входные данные программы дали одинаковый результат, из чего можно сделать вывод, что refactor.s работает корректно.

# Результаты работы

Для оценки 6 баллов нужно:

1. Программа на си - main.c
2. Асемблерная программа с использованием локальных переменных и функций - refactor.s/main.s
3. Рефакторинг асемблерной программы за счет максимального использования регистров - refactor.s
4. Отчет с результатами тестовых прогонов - папка tests и "Тестовые прогоны"
5. Сопоставление размеров программ - описано в "Сравнение программ"
6. Прокомментированная последняя версия программы - refactor.s прокомментирована
7. Отобразить информацию об изменениях в отчете - описано в "Сравнение программ"