ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Мошаров Денис Максимович

Содержание

# 1 Цель работы

Научиться писать и анализировать ассемблерный код с арифметическими операциями и понять синтаксис. Работа поможет развить навыки низкоуровневого программирования и понимания работы процессора.

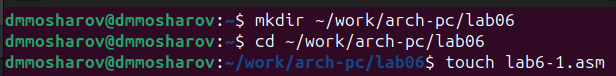
# 2 Задание

Написать несколько программ для вычислений.

# 3 Выполнение лабораторной работы

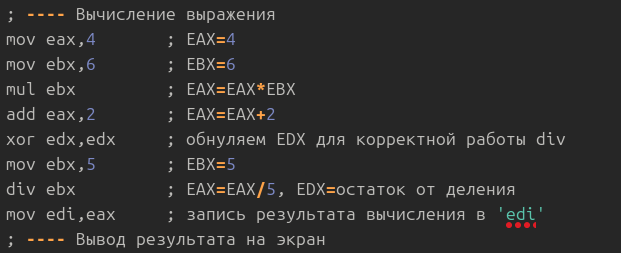
## 3.1 Порядок выполнения лабораторной работы

Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm

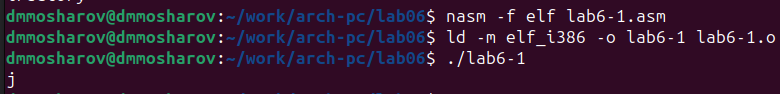


Создаём каталог

Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax



Заполняем программу

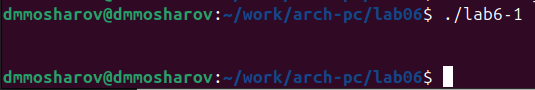


Результат

Изменим текст программы и вместо символов, запишем в регистры числа.

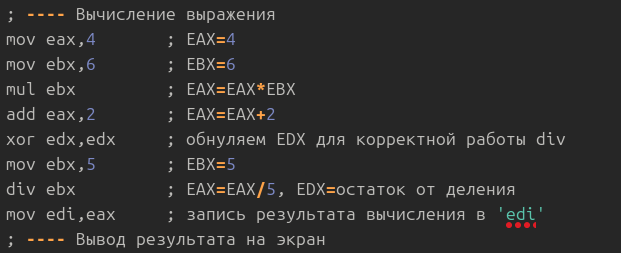
Подмена

Подмена



Результат

Преобразуем текст программы из Листинга 6.1 с использованием этих функций.



Заполняем

Результат

Результат

Изменим символы на числа

Подмена

Подмена

Результат

Результат

Замените функцию iprintLF на iprint. Создайте исполняемый файл и запустите его.

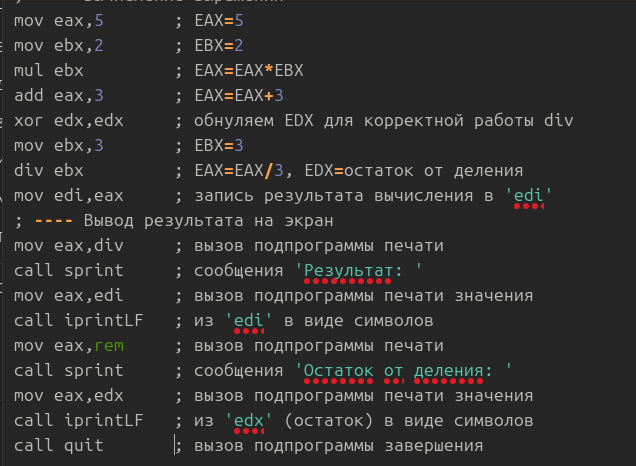
Меняем

Меняем

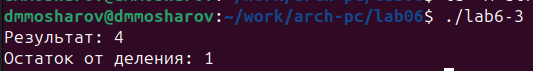
Результат

Результат

В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения f(x) = (5 \* 2+ 3)/3

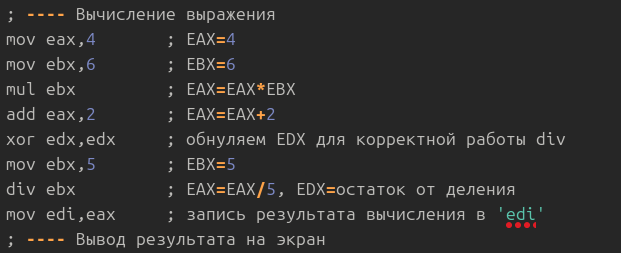


Пишем программу

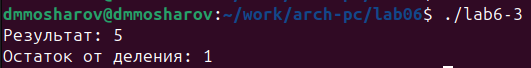


Результат

Измените текст программы для вычисления выражения f(x) = (4 \* 6 + 2)/5. Создайте исполняемый файл и проверьте его работу.

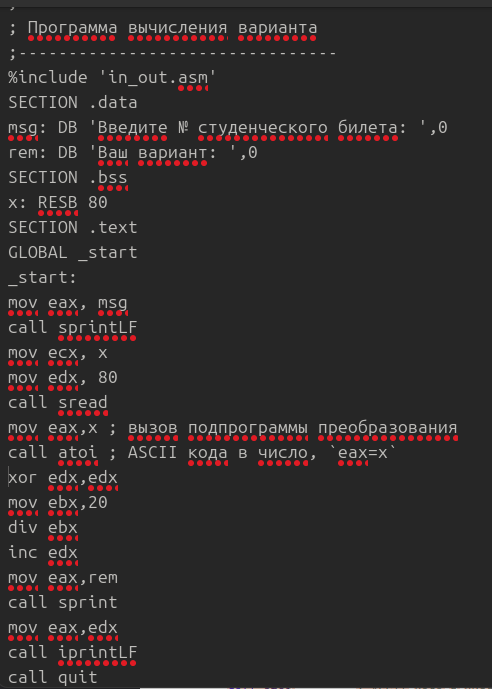


Меняем выражение

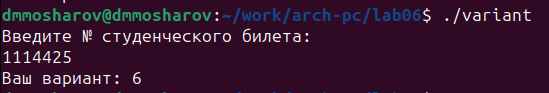


Результат

Рассмотрим программу вычисления варианта задания по номеру студенческого билета



Пишем программу



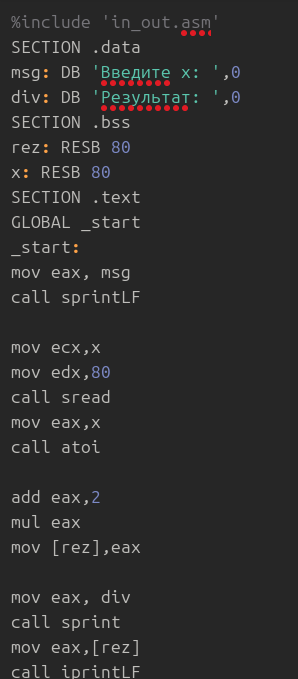
Результат

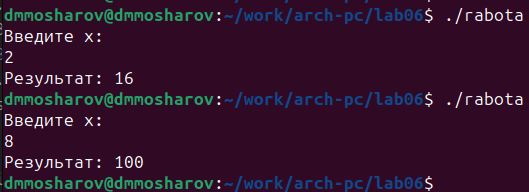
## 3.2 Ответы на вопросы

1. Строка “moveax.rem” и строка “call sprint” отвечают за вывод на экран сообщения ‘Ваш вариант:’.
2. Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре есх, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре edx. Затем вызывается процедура sread, которая выполняет чтение строки.
3. Инструкция “call atoi” используется для преобразования строки в целое число. Она принимает адрес строки в регистре еах и возвращает полученное число в регистре еaх. Строка “xoredx.edx” обнуляет регистр. edx перед выполнением деления. Строка “movebx,20” загружает значение 20 в регистр ebx. Строка “divebx” выполняет деление регистра еах на значение регистра ebx с сохранением частного в регистре еах и остатка в регистре edx,
4. Остаток от деления записывается в регистр edx.
5. Инструкция “inc edx” используется для увеличения значения в регистре edx на
6. В данном случае, она увеличивает остаток от деления на 1. 1з
7. Строка “moy eax.edx” передает значение остатка от деления в регистр eax. 36 Строка “call iprintLF” вызывает процедуруiprintLF для вывода значения на экран вместе с переводом строки.

## 3.3 Задание для самостоятельной работы

Написать программу вычисления выражения y = f(x). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x, вычислять заданное выражение в зависимости от введенного x, выводить результат вычислений. Создайте исполняемый файл и проверьте его работу для значений x1 и x2





# 4 Выводы

В работе были изучены арифметические операции в языке ассемблера NASM.Был рассмотрен синтаксис и были написаны и проанализированы программы на ассемблере, которые используют арифметические операции для решения различных задач.