Отчет по лабораторной работе № 6

Дисциплина: архитектура компьютера

Выслоух Алиса Александровна

Содержание

# 1 Цель работы

Цель данной лабораторной работы - освоение арифметческих инструкций языка ассемблера NASM.

# 2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

# 3 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов, и адрес операнда указывает на место хранения данных, подлежащих обработке. Эти данные могут находиться как в регистрах, так и в ячейках памяти.Регистровая адресация подразумевает, что операнды хранятся в регистрах, а в команде используются их имена. Например: mov ax, bx. В случае непосредственной адресации значение операнда задается прямо в команде, как в примере: mov ax, 2 . Адресация памяти предполагает, что операнд указывает адрес в памяти. В команде указывается символическое обозначение ячейки памяти, с которой нужно выполнять операцию.Ввод информации с клавиатуры и вывод на экран производятся в символьном формате. Кодирование этих данных осуществляется на основе таблицы символов ASCII. ASCII (American Standard Code for Information Interchange) представляет собой стандарт, по которому каждый символ кодируется одним байтом. Важно отметить, что в NASM нет инструкции, которая могла бы непосредственно выводить числа (не в символьном представлении). Поэтому, чтобы отобразить число на экране, его нужно сначала преобразовать в соответствующие коды ASCII. Если попытаться вывести число напрямую, экран воспримет его не как число, а как последовательность ASCII-символов; каждый байт числа будет интерпретирован как отдельный ASCII-символ.Аналогичная ситуация наблюдается при вводе данных с клавиатуры: введенные символы будут интерпретированы как текст, что затруднит выполнение арифметических операций с ними. Для решения этой проблемы необходимо осуществлять преобразование ASCII-символов в числа и наоборот.

# 4 Выполнение лабораторной работы

## 4.1 Символьные и численные данные в NASM

Создаю папку lab6 и перехожу в нее. (рис. 1).

Рис. 1: Создание папки.

Рис. 1: Создание папки.

С помощью утилиты touch создаю файл lab6-1.asm (рис. 2).

Рис. 2: Создание файла.

Рис. 2: Создание файла.

Открываю созданный файл lab7-1.asm, вставляю в него программу вывода значения регистра eax (рис. 3).

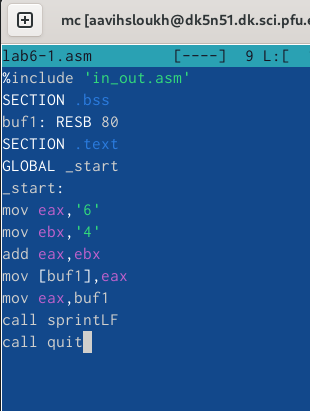


Рис. 3: Открытие файла и вставка.

Создаю исполняемый файл программы и запускаю его (рис. 4). Вывод программы: символ j.

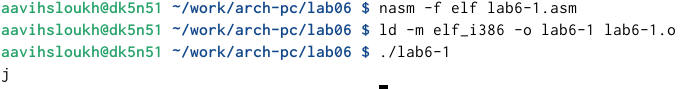


Рис. 4: Создание и запуск файла.

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 5).

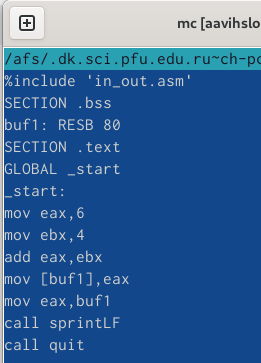


Рис. 5: Изменение файла.

Создаю новый исполняемый файл программы и запускаю его (рис. 6). Теперь вывелся символ перевода строки.

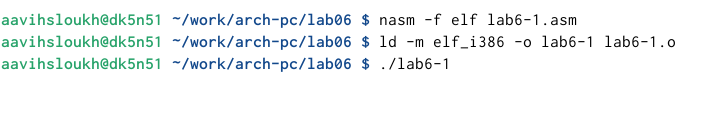


Рис. 6: Создание и запуск файла.

Создаю новый файл lab6-2.asm с помощью утилиты touch (рис. 7).

Рис. 7: Создание файла.

Рис. 7: Создание файла.

Ввожу в файл текст другойпрограммы для вывода значения регистра eax (рис. 8).

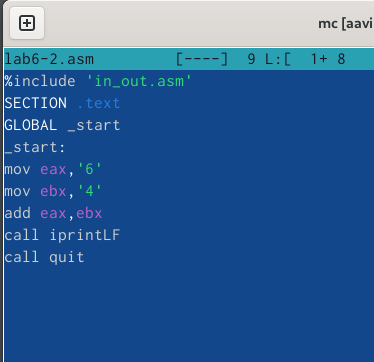


Рис. 8: Ввод текста в файл.

Создаю и запускаю исполняемый файл lab7-2 (рис. 9). Теперь вывод число 106.

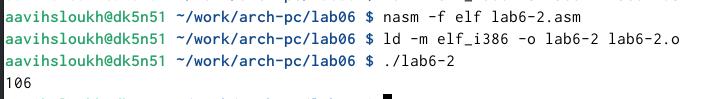


Рис. 9: Создание и запуск файла.

Заменяю в тексте программы в файле lab7-2.asm символы “6” и “4” на числа 6 и 4 (рис. 10).

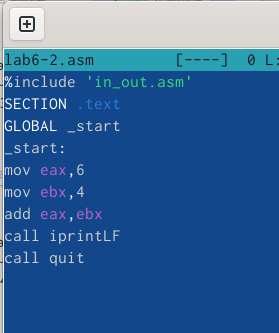


Рис. 10: Изменение файла.

Создаю и запускаю новый исполняемый файл (рис. 11).. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.

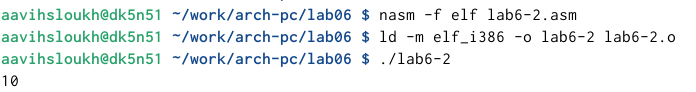


Рис. 11: Создание и запуск файла.

Заменяю в тексте программы функцию iprintLF на iprint. Вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией iprintLF, а iprint не добавляет к выводу символ переноса строки, в отличие от iprintLF.

## 4.2 Выполнение арифметических операций в NASM

Создаю файл lab7-3.asm с помощью утилиты touch (рис. 12).

Рис. 12: Создание файла.

Рис. 12: Создание файла.

Ввожу в созданный файл текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис. 13).

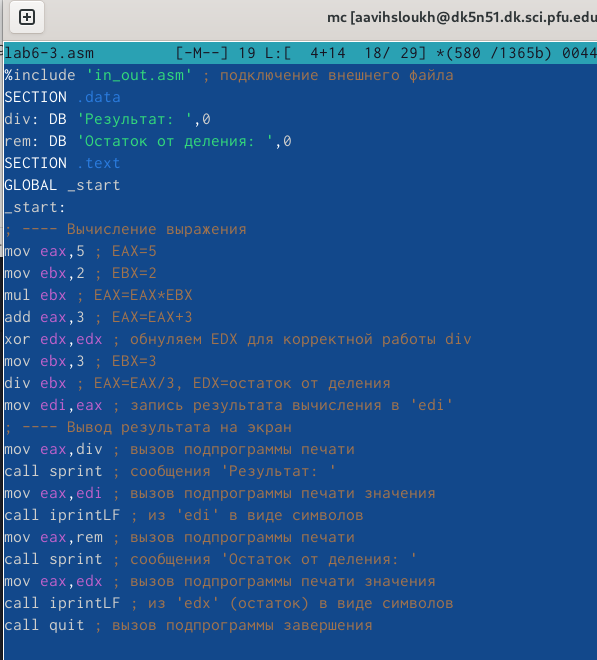


Рис. 13: Ввод текста.

Создаю исполняемый файл и запускаю его (рис. 14).

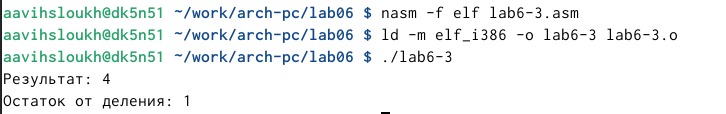


Рис. 14: Создание и запуск файла.

Изменяю программу так, чтобы она вычисляла значение выражения f(x) = (4 \* 6 + 2)/5 (рис. 15).

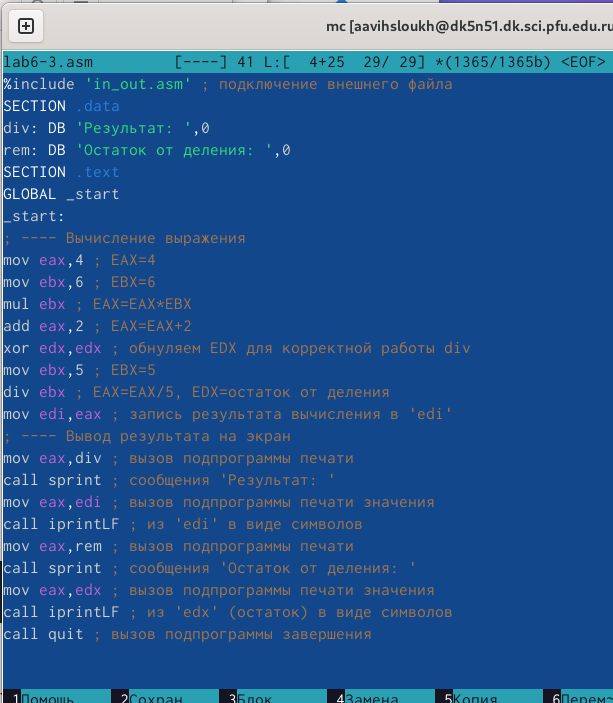


Рис. 15: Изменение программы.

Создаю и запускаю новый исполняемый файл (рис. 16). Я посчитала для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно.

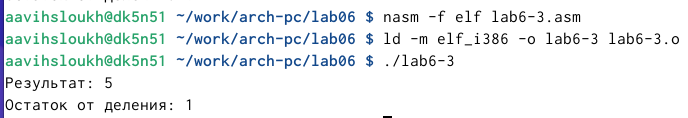


Рис. 16: Создание и запуск файла.

Создаю файл variant.asm с помощью утилиты touch (рис. 17).

Рис. 17: Создание файла.

Рис. 17: Создание файла.

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 18).

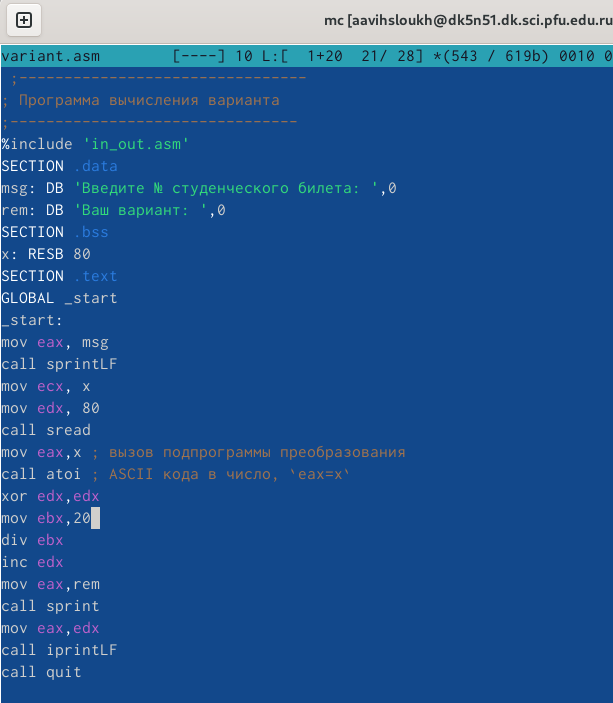


Рис. 18: Ввод текста.

Создаю и запускаю исполняемый файл (рис. 19). Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 19.

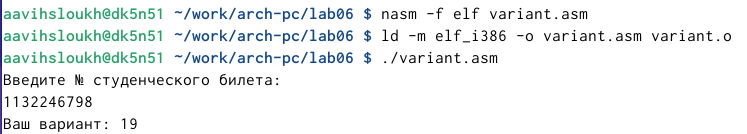


Рис. 19: Создание и запуск файла.

## 4.3 Выполнение заданий для самостоятельной работы

Создаю файл lab7-4.asm с помощью утилиты touch (рис. 20).

Рис. 20: Создание файла.

Рис. 20: Создание файла.

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения (11 + x) \* 2 - 6 (рис. 21). Это выражение было под вариантом 8.

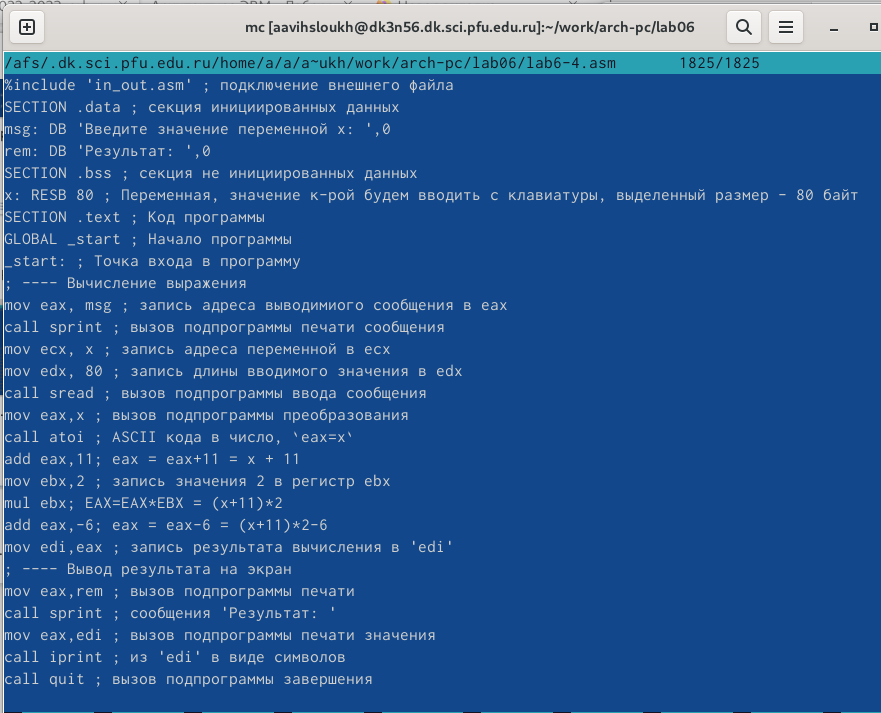


Рис. 21: Ввод текста программы.

Создаю и запускаю исполняемый файл (рис. 22). При вводе значения 2, вывод 20.

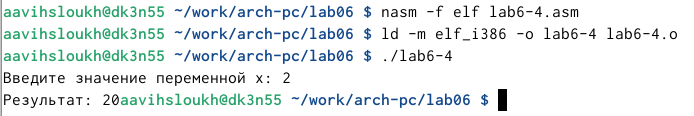


Рис. 22: Создание и запуск исполняемого файла.

# 5 Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.