Лабораторная работа №7

Дисциплина: Архитектура компьютера

Апареев Дмитрий Андреевич

Содержание

# 1 Цель работы

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

# 2 Задание

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

# 3 Выполнение лабораторной работы

С помощью утилиты mkdir создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №7 (рис. 1). Перехожу в созданный каталог с помощью утилиты cd (рис. 2)

Рис. 1: 1

Рис. 1: 1

Рис. 2: 1.1

Рис. 2: 1.1

С помощью утилиты touch создаю файл lab7-1.asm (рис. 3).

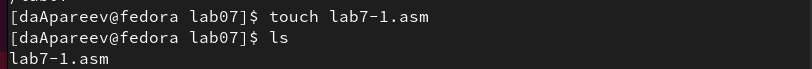


Рис. 3: 2

Копирую в текущий каталог файл in\_out.asm с помощью утилиты cp (рис. 4).

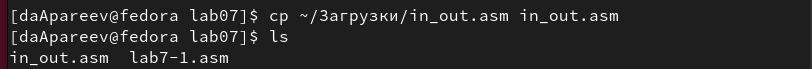


Рис. 4: 3

Открываю созданный файл lab7-1.asm, вставляю в него программу вывода значения регистра eax (рис. 5).



Рис. 5: 4

Создаю исполняемый файл программы и запускаю его. Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6. (рис. 6)

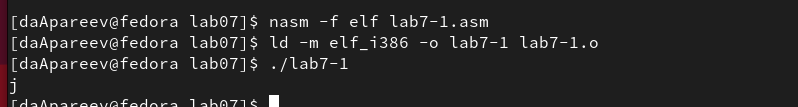


Рис. 6: 5

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 7).



Рис. 7: 6

Создаю новый исполняемый файл программы и запускаю его. Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.(рис. 8)

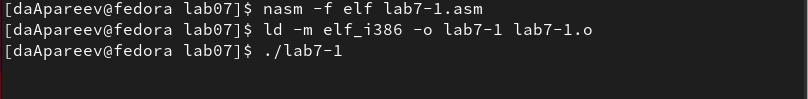


Рис. 8: 7

Создаю новый файл lab7-2.asm с помощью утилиты touch (рис. 9).

Рис. 9: 8

Рис. 9: 8

Ввожу в файл текст другой программы для вывода значения регистра eax (рис. **¿fig:009?**). !(image/9.png){ #fig:009 width=70% }

Создаю и запускаю исполняемый файл lab7-2. Теперь вывод число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”. (рис. 10).

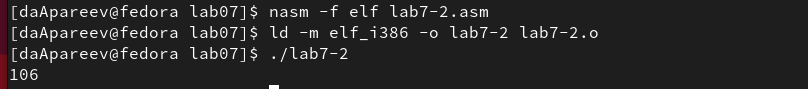


Рис. 10: 10

Заменяю в тексте программы в файле lab7-2.asm символы “6” и “4” на числа 6 и 4 (рис. 11).

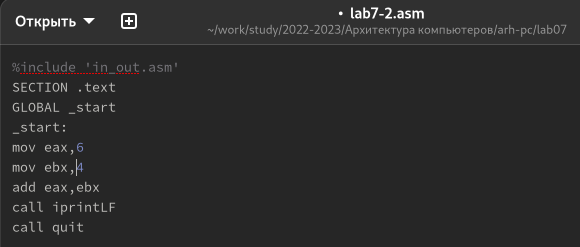


Рис. 11: 11

Создаю и запускаю новый исполняемый файл. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10. (рис. 12)

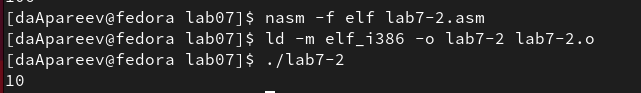


Рис. 12: 12

Заменяю в тексте программы функцию iprintLF на iprint (рис. 13).

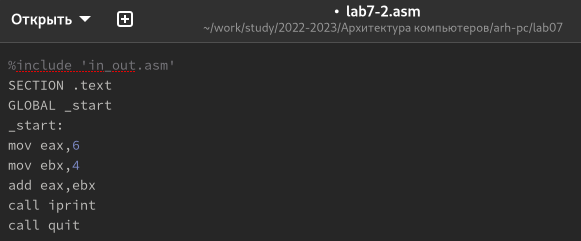


Рис. 13: 13

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

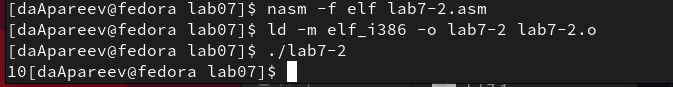


Рис. 14: 14

Создаю файл lab7-3.asm с помощью утилиты touch (рис. 15).

Рис. 15: 15

Рис. 15: 15

Ввожу в созданный файл текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис. 16).

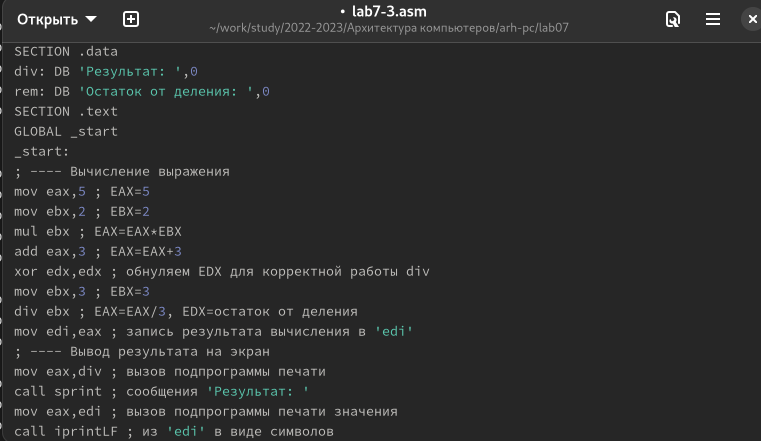


Рис. 16: 16

Создаю исполняемый файл и запускаю его (рис. 17).

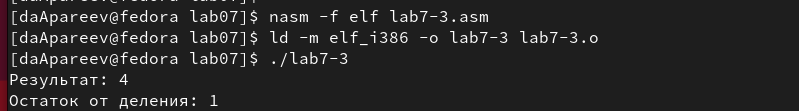


Рис. 17: 17

Изменяю программу так, чтобы она вычисляла значение выражения f(x) = (4 \* 6 + 2)/5 (рис. 18).

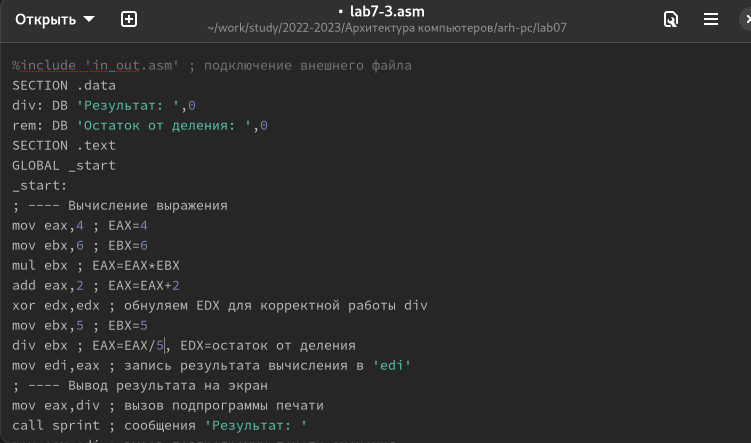


Рис. 18: 18

Создаю и запускаю новый исполняемый файл (рис. 19).

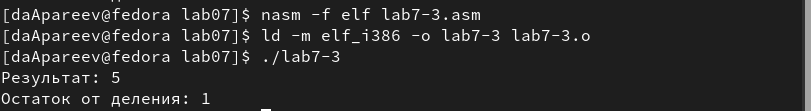


Рис. 19: 19

Создаю файл variant.asm с помощью утилиты touch (рис. 20).

Рис. 20: 20

Рис. 20: 20

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 21).

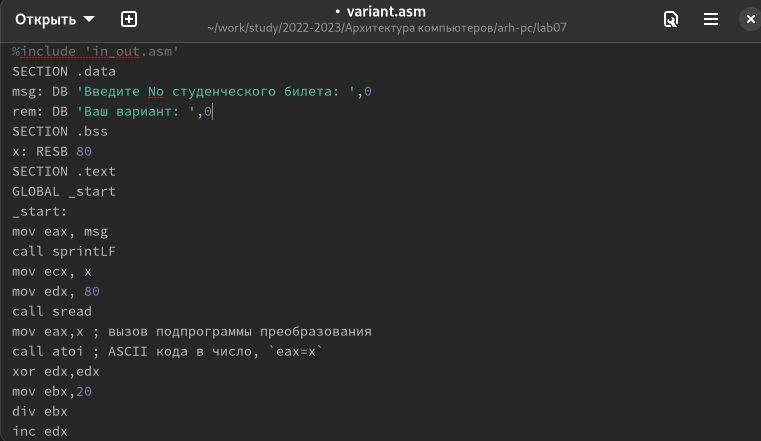


Рис. 21: 21

Создаю и запускаю исполняемый файл. Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант 6. (рис. 22)

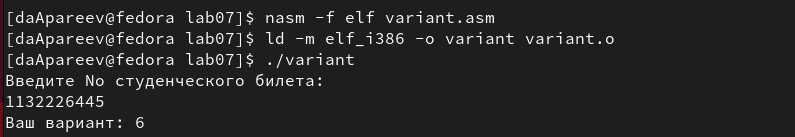


Рис. 22: 22

## 3.1 Ответы на вопросы

1. За вывод сообщения “Ваш вариант” отвечают строки кода: mov eax,rem call sprint
2. Инструкция mov ecx, x используется, чтобы положить адрес вводимой строки x в регистр ecx mov edx, 80 - запись в регистр edx длины вводимой строки call sread - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax
4. За вычисления варианта отвечают строки: xor edx,edx ; обнуление edx для корректной работы div mov ebx,20 ; ebx = 20 div ebx ; eax = eax/20, edx - остаток от деления inc edx ; edx = edx + 1
5. При выполнении инструкции div ebx остаток от деления записывается в регистр edx
6. Инструкция inc edx увеличивает значение регистра edx на 1
7. За вывод на экран результатов вычислений отвечают строки: mov eax,edx call iprintLF

Открываю файл report.md с помощью любого текстового редактора gedit. Компилирую файл с отчетом. Загружаю отчет на GitHub.

# 4 Задания для самостоятельной работы

Создаю файл lab7-4.asm с помощью утилиты touch (рис. 23).

Рис. 23: 23

Рис. 23: 23

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения x^3 / 2 + 1 (рис. 24).

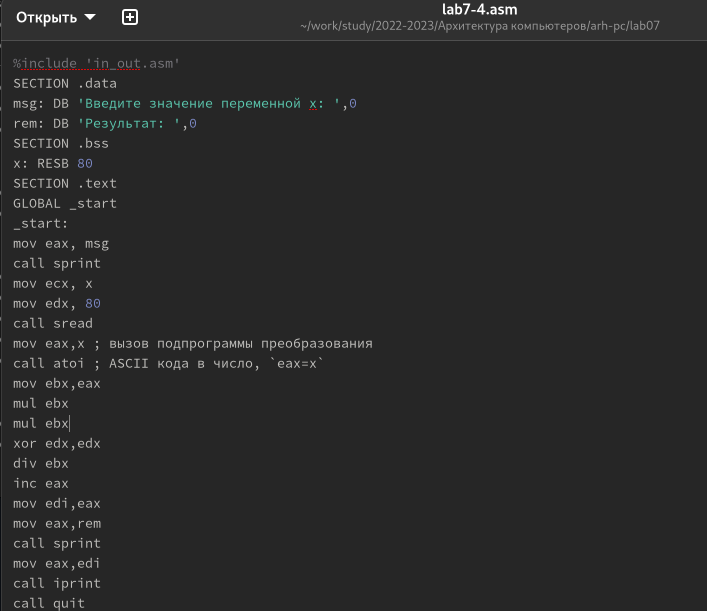


Рис. 24: 24

Создаю и запускаю исполняемый файл (рис. 25)

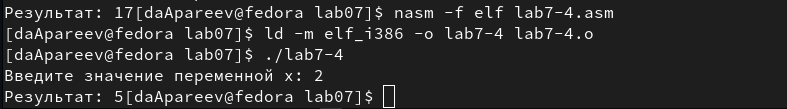


Рис. 25: 25

Добавляю файлы в git , Сохраняю файлы в git , Отправляю файлы на сервер

# 5 Вывод

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