Отчет по лабораторной работе №7

Дисциплина архитектура компьютера

Извекова Мария Петровна

Содержание

# 1 Цель работы

Освоение арифметических 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. Адрес операнда предоставляет место, где хранятся данные, подлежащие обработке. Это могут быть данные хранящиеся в регистре или в ячейке памяти. Далее рассмотрены все существующие способы задания адреса хранения операндов – способы адресации. Существует три основных способа адресации: • Регистровая адресация – операнды хранятся в регистрах и в команде используются имена этих регистров, например: mov ax,bx. • Непосредственная адресация – значение операнда задается непосредственно в команде, Например: mov ax,2. • Адресация памяти– операнд задает адрес в памяти. В команде указывается символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию.

# 4 Выполнение лабораторной работы

## 4.1 Символьные и численные данные в NASM

1. Переходим в каталог для программ Лабораторной работы №7, создаем файл lab7-1.asm и с помощью функции gedit открываем этот файл для редактирования (рис. 1)

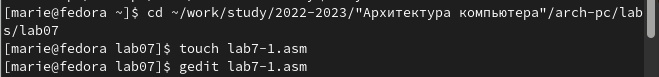


Рис. 1: рис. 1

1. В этот файл вставляем текст из файла in\_out.asm для вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax.(рис. 2)

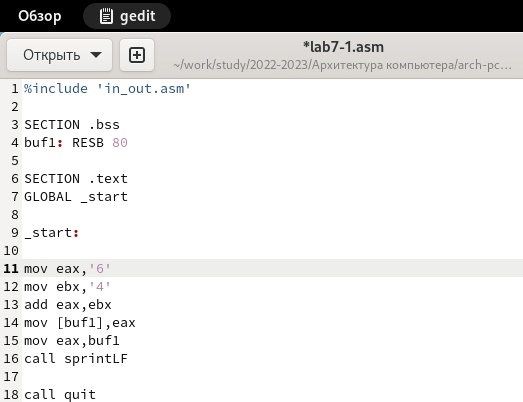


Рис. 2: рис. 2

1. Запускаем файл с помощью следующих команд (рис. 3) выводим значение. Мы получаем значение j.

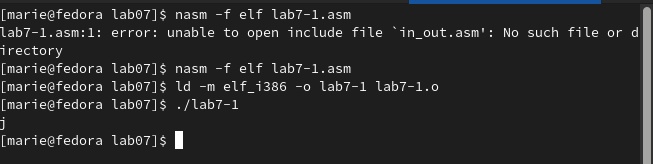


Рис. 3: рис. 3

1. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправляем текст программы из рисунка 2 следующим образом: замените строкит mov eax,‘6’, mov ebx,‘4’ на строки mov eax,6 , mov ebx,4 (рис. 4)

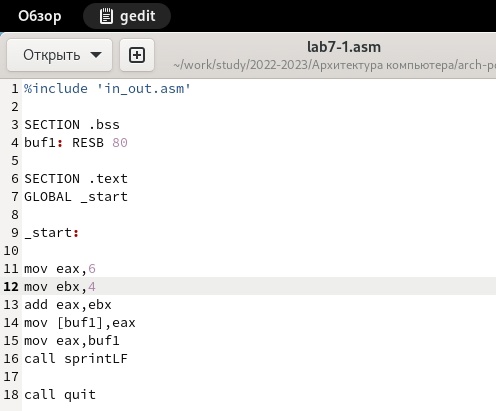


Рис. 4: рис. 4

1. Создаем файл и запускаем его с помощью команд. Выводится символ с кодом 10, который не отображается на экране. (рис. 5)

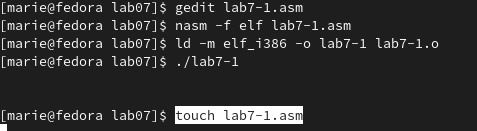


Рис. 5: рис. 5

1. Создаем новый файл lab7-2.asm и вставляем исправленный текст из рисунка 2. Создаем файл и запускаем его. В данном случае выводится цифра 106, так как функция iprintLF позволяет вывести число, а не символ, кодом которого является это число.(рис. 6 - 7)



Рис. 6: рис. 6

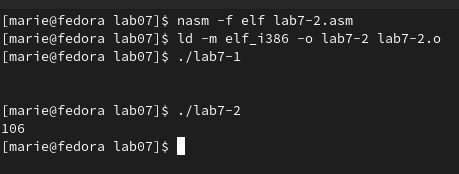


Рис. 7: рис. 7

1. Аналогично предыдущему примеру изменяем символы на числа. Заменяем строки mov eax,‘6’, mov ebx,‘4’ на строки mov eax,6, mov ebx,4 . Создаем исполняемый файл и запускаем его. (рис. 8)



Рис. 8: рис. 8

1. В данном случае выводится цифра 10 (рис. 9)

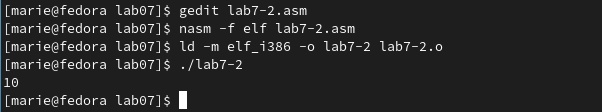


Рис. 9: рис. 9

1. Заменяем функцию iprintLF на iprint. Создаем исполняемый файл и запускаем его. (рис. 10 - 11)

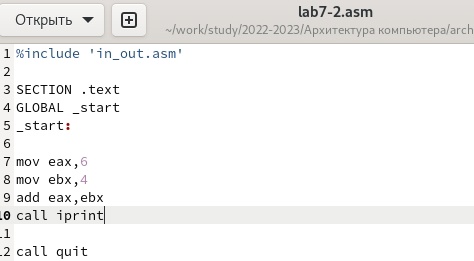


Рис. 10: рис. 10

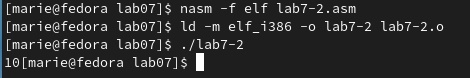


Рис. 11: рис. 11

## 4.2 Выполнение арифметических операций в NASM

1. В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения 𝑓(𝑥) = (5 ∗ 2 + 3)/3. Создаем файл lab7-3.asm в каталоге ~/work/arch-pc/lab07 и открываем его с помощью редактора (рис. 12)

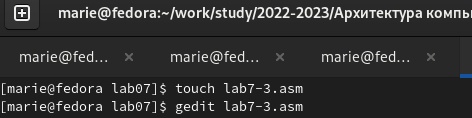


Рис. 12: рис. 12

1. В этот файл вставляем следующий текст (рис. 13)

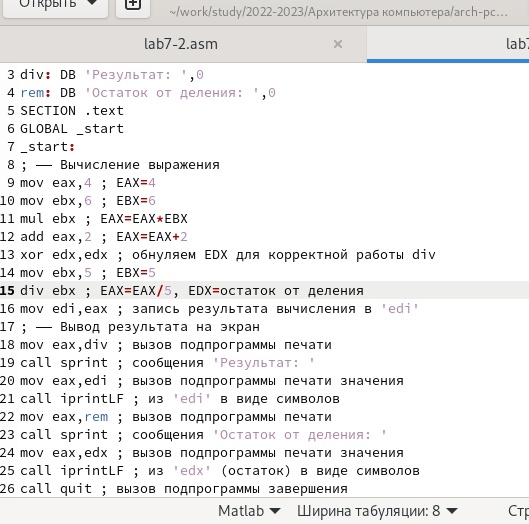


Рис. 13: рис. 13

1. Создаем файл и запускаем его. Результат работы следующий: (рис. 14)

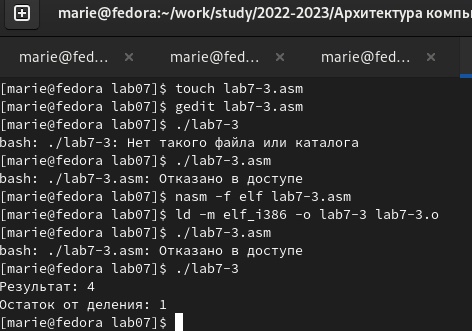


Рис. 14: рис. 14

1. Изменяем текст программы для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5. Создаем исполняемый файл и проверяем его работу. (рис. 15 - 16)

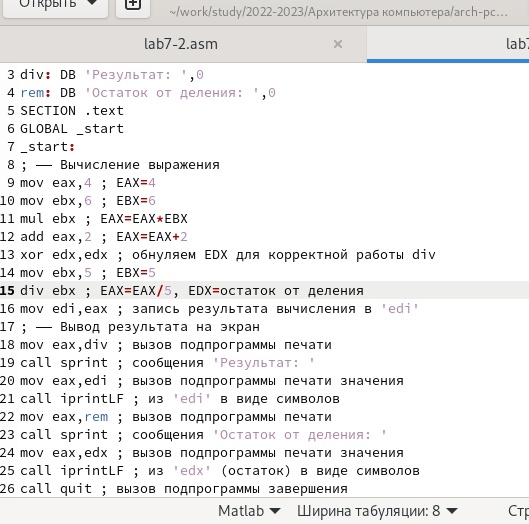


Рис. 15: рис. 15

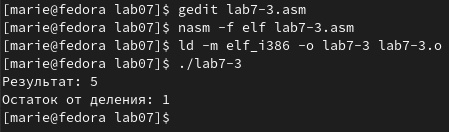


Рис. 16: рис. 16

1. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета. Создаем файл variant.asm в каталоге ~/work/arch-pc/lab07 с помощью команды touch. В этот файл вводим следующий текст (рис. 17)

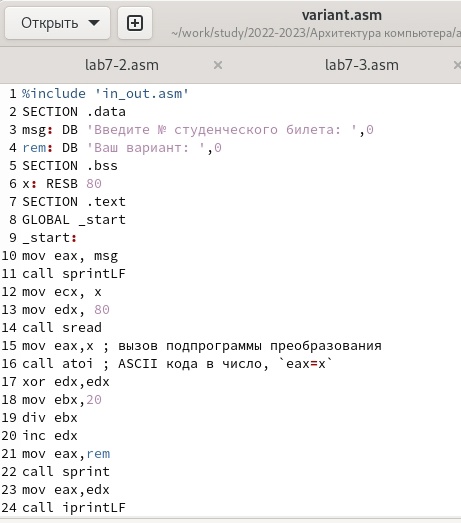


Рис. 17: рис. 17

1. Проверяем работу, ввожу номер студенческого билета, он выводит номер варианта(рис. 18).

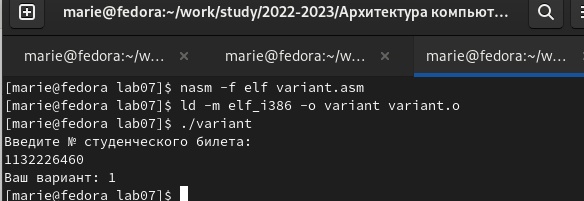


Рис. 18: рис. 18

## 4.3 Ответы на вопросы по программе

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

1. При выполнении инструкции div ebx остаток от деления записывается в регистр edx
2. Инструкция inc edx увеличивает значение регистра edx на 1
3. За вывод на экран результатов вычислений отвечают строки:

mov eax,edx call iprintLF

## 4.4 Самостоятельная работа

Вариант 1, вычисление (10 + 2𝑥)/3

1. создаем файл lab4-1.asm, открываем его с помощью редактора gedit и вставляем следующий текст (рис. 19 - 20)

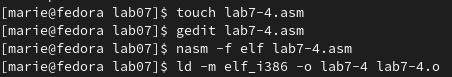


Рис. 19: рис. 19

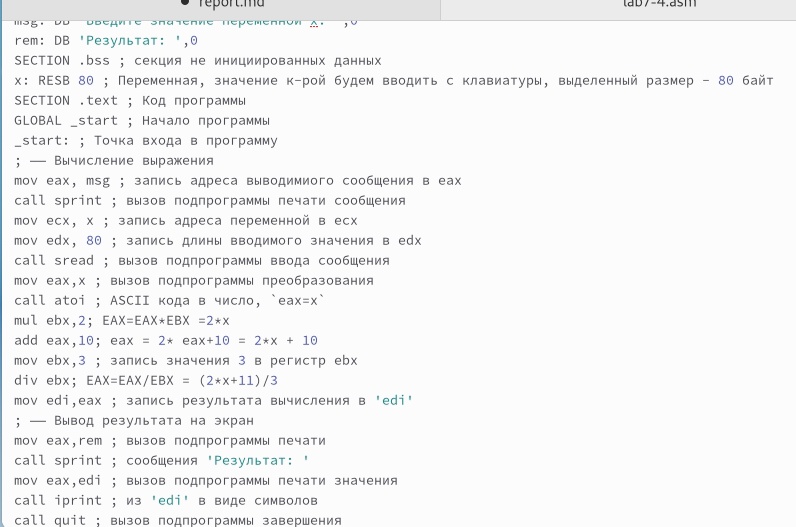


Рис. 20: рис. 20

1. Создаем и запускаем файл. Получаем следующие результаты: (рис. 21 - 22)

Рис. 21: рис. 21

Рис. 21: рис. 21

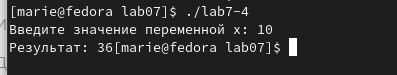


Рис. 22: рис. 22

# 5 Выводы

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

# Список литературы

1. Лабораторная работа №7
2. Таблица ASCII