Отчет по лабораторной работе No.6

Дисциплины: Архитектура компьютера

Нджову Нелиа

Содержание

# 1 Цель работы

Целью данной работы является освоение арифметических инструкций на языке ассемблера NASM.

# 2 Задание

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

# 3 Выполнение лабораторной работы

**1. Символьные и численные данные в NASM**

Я создам каталог для программ лабораторных работ 6 с помощью команды mkdir, зайду в него с помощью команды cd и создам файл lab6-1.asm с помощью команды touch(рис 1)

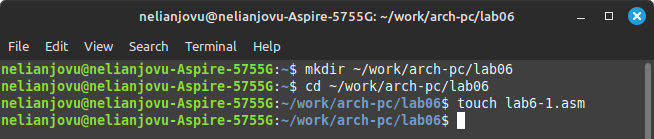


Рис 1

Я скопирую файл in\_out.asm в текущий каталог с помощью команды cp, потому что буду использовать его в программах(рис 2)

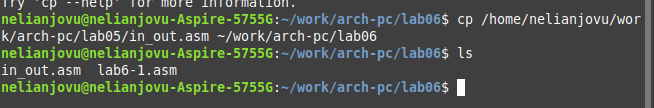


Рис 2

Открою созданный файл lab6-1.asm и скопирую в него программу вывода значения регистра eax(рис 3)

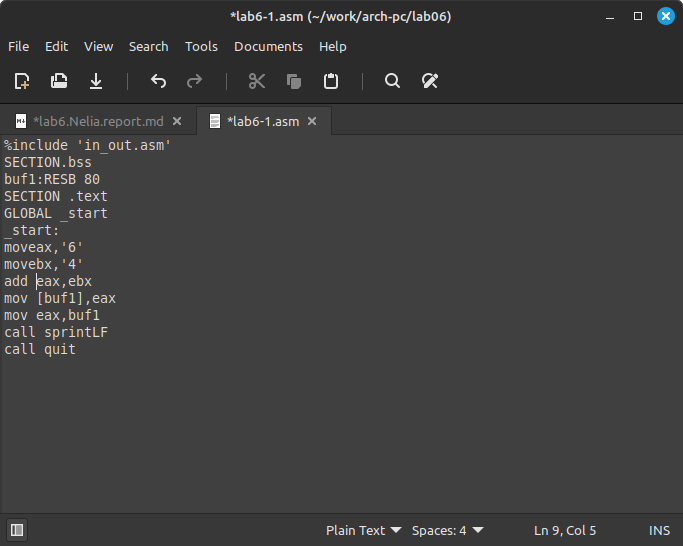


Рис 3

Я создам исполняемый файл программы и запущу его. Программа выведет символ j, поскольку программа выводит символ, соответствующий сумме ASCII двоичных кодов символов 4 и 6(рис 4)

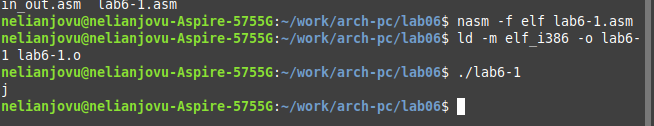


Рис 4

Я заменю символы ‘6’ и ‘4’ в тексте программы на цифры 6 и 4(рис 5)

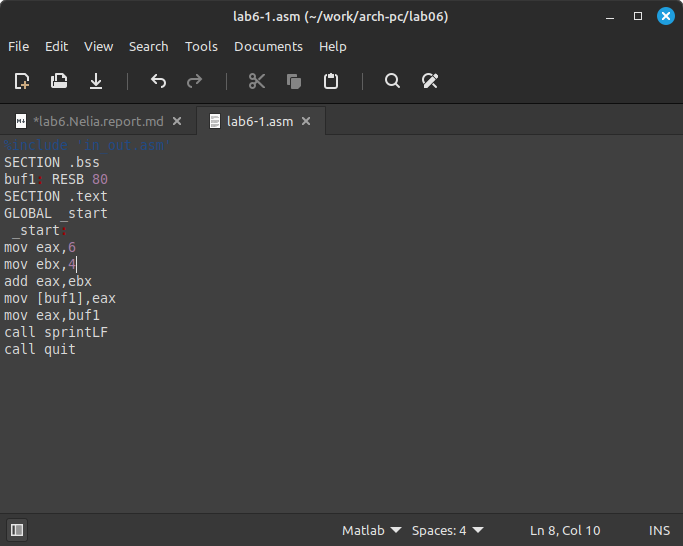


Рис 5

Я создам новый исполняемый файл программы и запущу его. Теперь отображается символ с кодом 10, это символ перевода строки(рис 6)

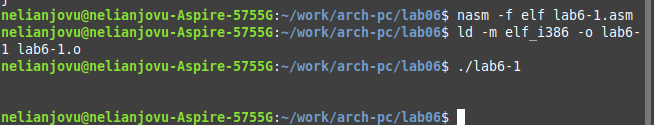


Рис 6

Я создам новый файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 с помощью команды touch(рис 7)

Рис 7

Рис 7

Я открою вновь созданный файл и скопирую в него заданный текст программы(рис 8)

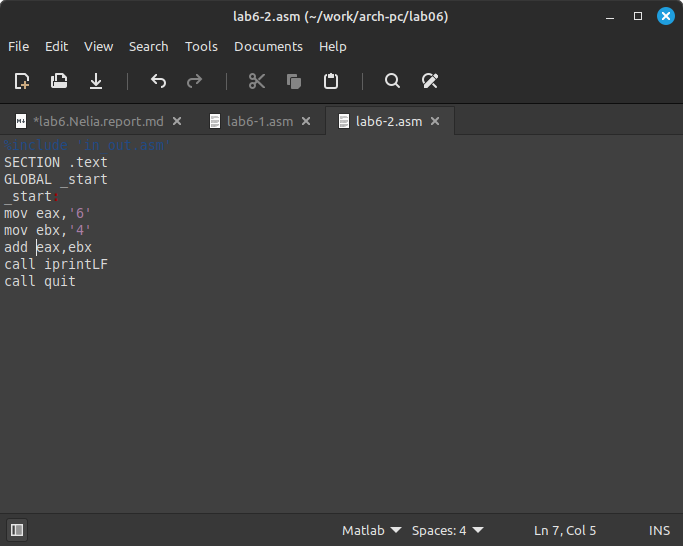


Рис 8

Я создаю и запускаю исполняемый файл lab6-2. Теперь на выходе будет число 106, поскольку функция iprintLF позволяет программе выводить точное число вместо символа ASCII(рис 9)

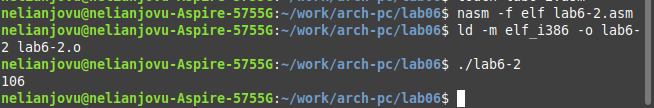


Рис 9

Теперь я заменю символы ‘6’ и ‘4’ в тексте программы на цифры 6 и 4(рис 10)

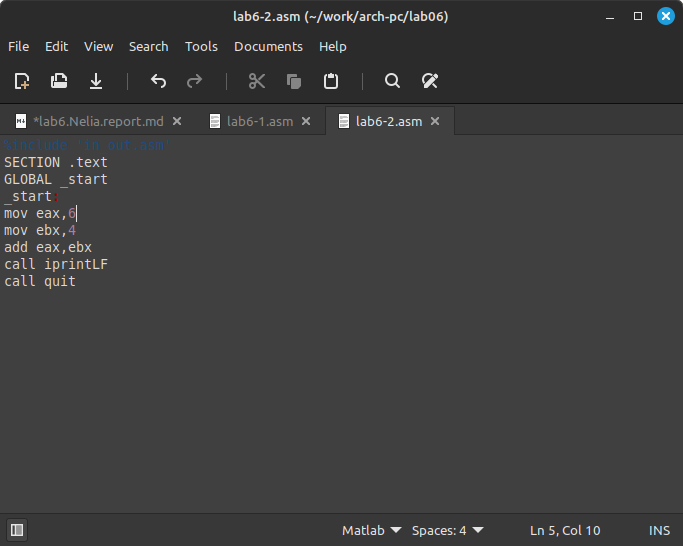


Рис 10

Как и ранее, я создаю и запускаю исполняемый файл lab6-2. Вывод равен 10, поскольку функция iprintLF позволяет программе выводить точное число вместо символа ASCII(рис 11)

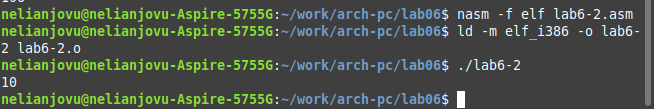


Рис 11

**2. Выполнение арифметических операций в NASM**

Я создам еще один файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 с помощью команды touch(рис 12)

Рис 12

Рис 12

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

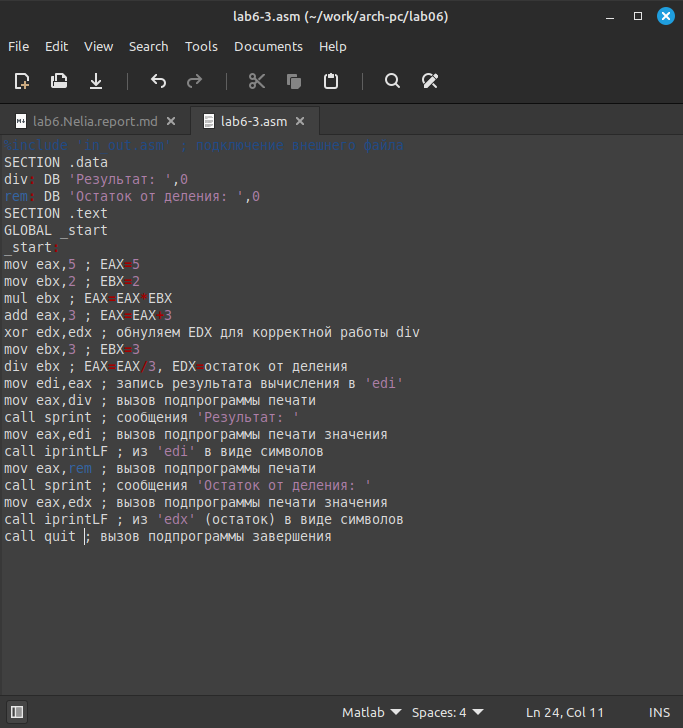


Рис 13

Создаю исполняемый файл и запускаю его(рис 14)

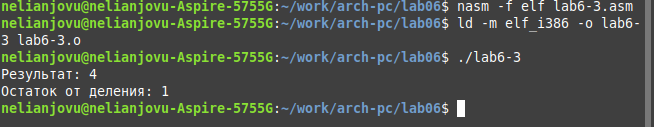


Рис 14

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

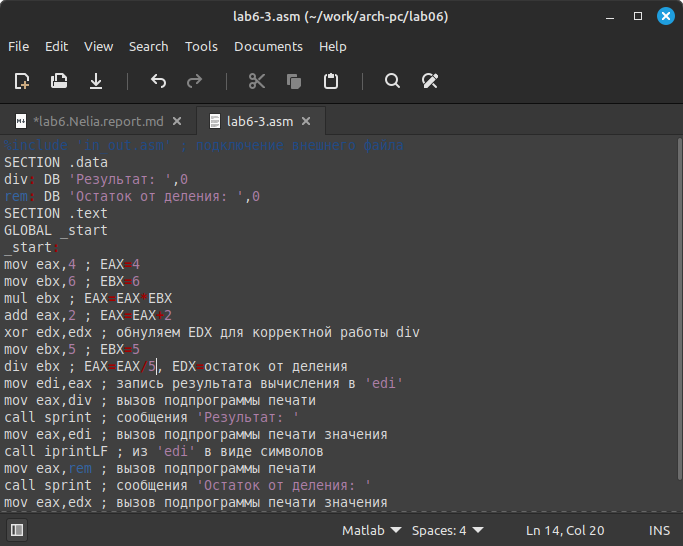


Рис 15

Теперь я создаю исполняемый файл и запускаю его(рис 16)

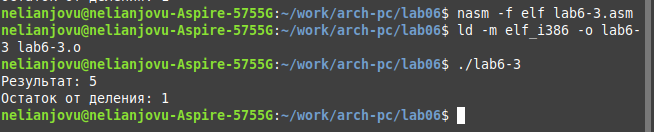


Рис 16

Я создам файл variant.asm в каталоге ~/work/arch-pc/lab06 с помощью команду touch(рис 17)

Рис 17

Рис 17

Скопирую текст программы в файл для расчета варианта задания по студенческому билету(рис 18)

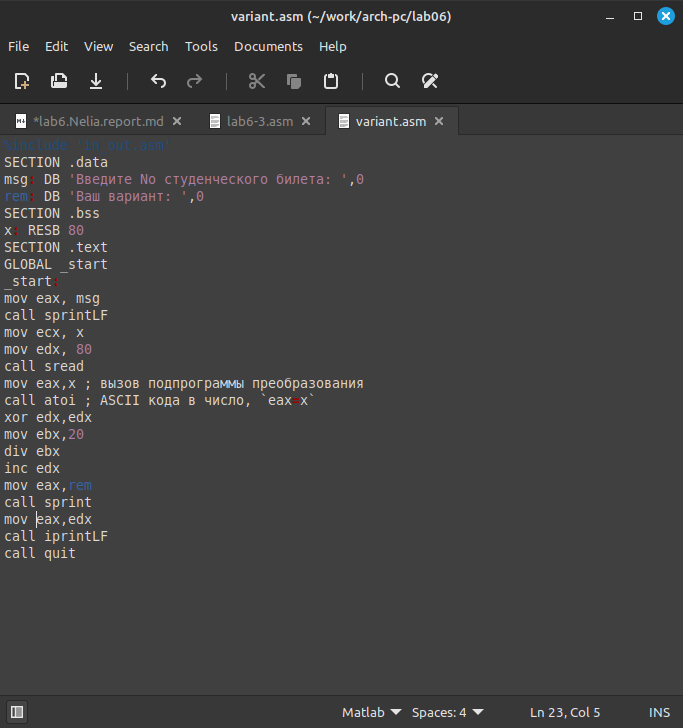


Рис 18

Я создам и запущу исполняемый файл. Ввожу с клавиатуры номер студенческого билета, программа показывает, что мой вариант-14(рис 19)

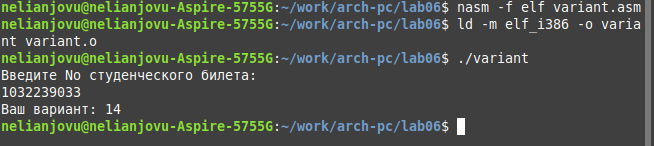


Рис 19

**2.1 Ответы на вопросы по программе**

1.mov eax,rem

call sprint

2.mov ecx,x — используется для помещения адреса входной строки x в регистр ecx.

mov edx,80 — записывает длину входной строки в регистр edx

call sread — вызов подпрограммы из внешнего файла, позволяющей ввести сообщение с клавиатуры

3.call atoi используется для преобразования ascii-кода символа в целое число и записи результата в регистр eax

4.xor edx,edx

mov ebx,20

div ebx

inc edx

5.Остаток от деления записывается в регистр edx

6.Инструкция inc edx увеличивает значение регистра edx на 1

7.mov eax,edx

call iprintLF

**3. Выполнение заданий для самостоятельной работы**

Я создам файл lab6-4.asm с помощью команды touch(рис 20)

Рис 20

Рис 20

Созданный файл открою для редактирования, введу в него текст программы для вычисления значения выражения (x/2 + 8)\*3(рис. 4.24). Это выражение было под вариантом 14(рис 21)

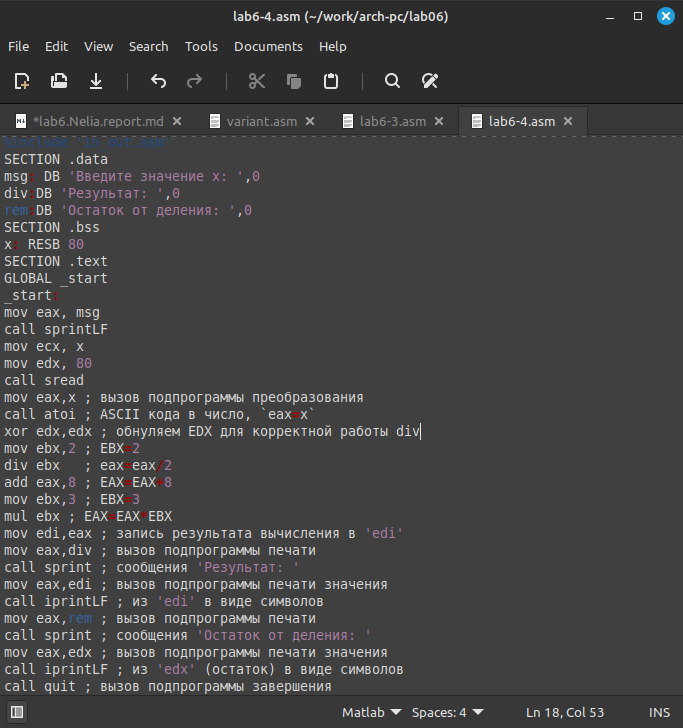


Рис 21

Я создам и запущу исполняемый файл. Когда вы вводите значение 1, выход 24(рис 22)

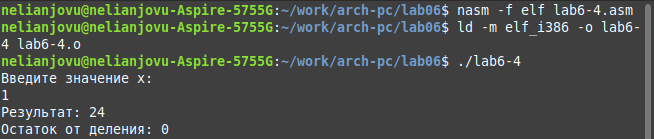


Рис 22

Я запускаю исполняемый файл еще раз, чтобы проверить работу программы с другим входным значением, на этот раз я буду использовать 4, а на выходе должно быть 30. Программа работала корректно(рис 23)

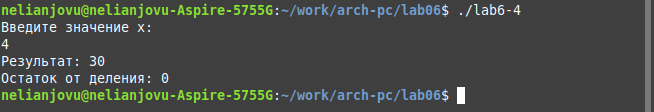


Рис 23

***Листинг 3.1. Программа для вычисления значения выражения (8*x - 6)/2**\*

%include 'in\_out.asm'  
SECTION .data  
msg: DB 'Введите значение х: ',0  
div:DB 'Результат: ',0  
rem:DB 'Остаток от деления: ',0  
SECTION .bss  
x: RESB 80  
SECTION .text  
GLOBAL \_start  
\_start:  
mov eax, msg  
call sprintLF  
mov ecx, x  
mov edx, 80  
call sread  
mov eax,x ; вызов подпрограммы преобразования  
call atoi ; ASCII кода в число, `eax=x`  
xor edx,edx ; обнуляем EDX для корректной работы div  
mov ebx,2 ; EBX=2  
div ebx ; eax=eax/2  
add eax,8 ; EAX=EAX+8  
mov ebx,3 ; EBX=3  
mul ebx ; EAX=EAX\*EBX  
mov edi,eax ; запись результата вычисления в 'edi'  
mov eax,div ; вызов подпрограммы печати  
call sprint ; сообщения 'Результат: '  
mov eax,edi ; вызов подпрограммы печати значения  
call iprintLF ; из 'edi' в виде символов  
mov eax,rem ; вызов подпрограммы печати  
call sprint ; сообщения 'Остаток от деления: '  
mov eax,edx ; вызов подпрограммы печати значения  
call iprintLF ; из 'edx' (остаток) в виде символов  
call quit ; вызов подпрограммы завершения

# 4 Выводы

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

# Список литературы

1. Архитектура ЭВМ
2. Таблица ASCII