Отчёт по лабораторной работе № 6

Дисциплина: Архитектура компьютера

Мутале Чали

Содержание

[1 Цель работы 1](#_Toc182606479)

[2 Задание 1](#_Toc182606480)

[3 Выполнение лабораторной работы 1](#_Toc182606481)

[4 Выполнение самостоятельной работы 8](#_Toc182606482)

[5 Выводы 11](#_Toc182606483)

[Список литературы 11](#_Toc182606484)

# 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. Ответы на вопросы по программе

# 3 Выполнение лабораторной работы

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

Создаю каталог для программам лабораторной работы № 6, перехожу в него и создаю файл lab6-1.asm:

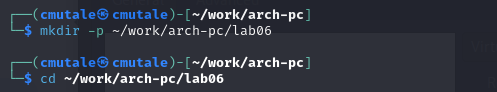


Рис 1

Копирую в текущий каталог файл in\_out.asm,потому что он будет использоваться в других программах:

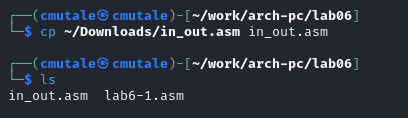


Рис 2

Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax:

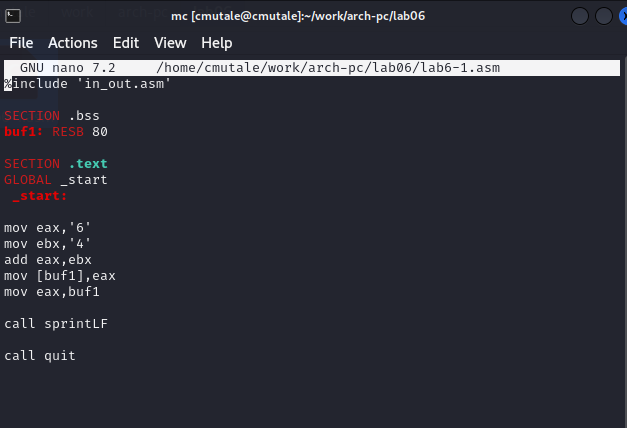


Рис 3

Создаю исполняемый файл программы и запускаю его. Программа выводит символ ‘j’, потому что это сумма кодов символов 6 и 4 по системе ASCII:

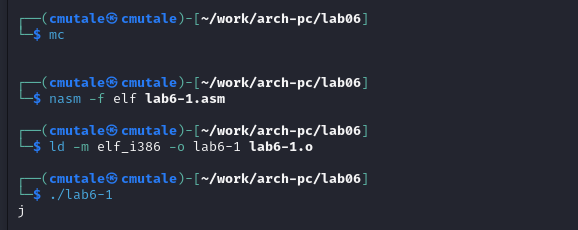


Рис 4

Изменяю текст программы и вместо символов, пишу в регистры числа:

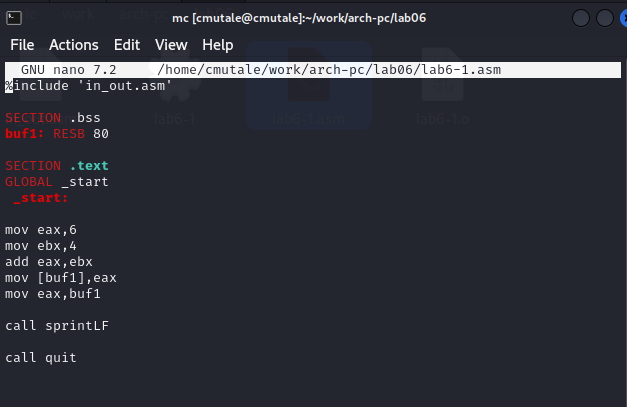


Рис 5

Создаю исполняемый файл программы и запускаю его. Теперь она выводит новую строку (код символа 10):

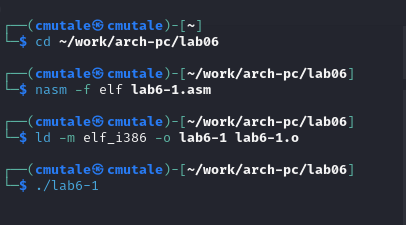


Рис 6

Создаю новый файл lab6-2.asm:

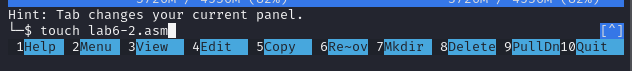


Рис 7

Открываю созданный файл lab6-2.asm, вставляю в него программу вывода значения регистра eax:

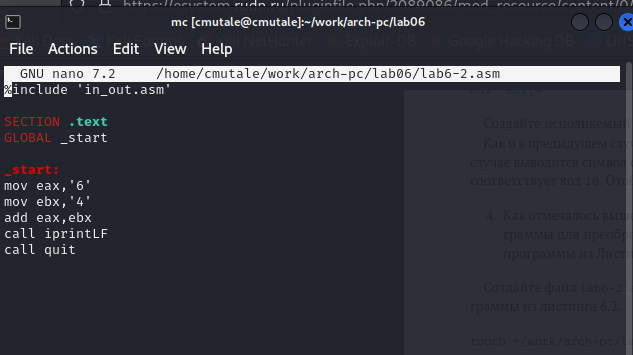


Рис 8

Создаю исполняемый файл программы и запускаю его. Она выводит сумму кодов символ 6(54) и символ 4(52):

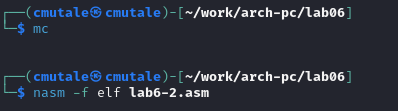


Рис 9

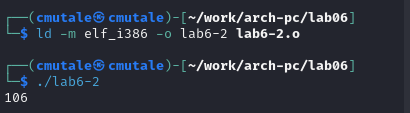


Рис 10

Изменяю текст программы и вместо символов, пишу в регистры числа:

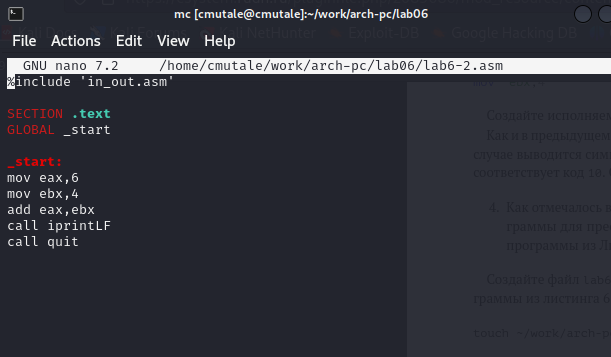


Рис 11

Создаю исполняемый файл программы и запускаю его. Теперь она выводит сумму чисел 6 и 4:

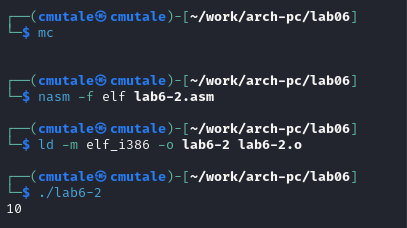


Рис 12

Изменяю iprintLF в iprint:

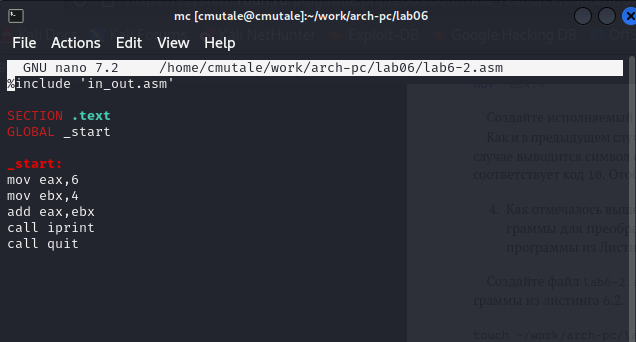


Рис 13

Создаю испольняемый файл и запускаю его:

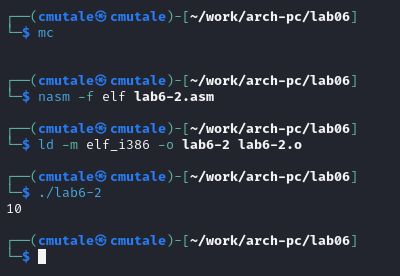


Рис 14

**2. Выполнение арифметических операций в NASM**

Создаю новый файл lab6-3.asm:



Рис 15

Вставляю в него программу для вычисления выражения f(x) = (5 \* 2 + 3)/3 :

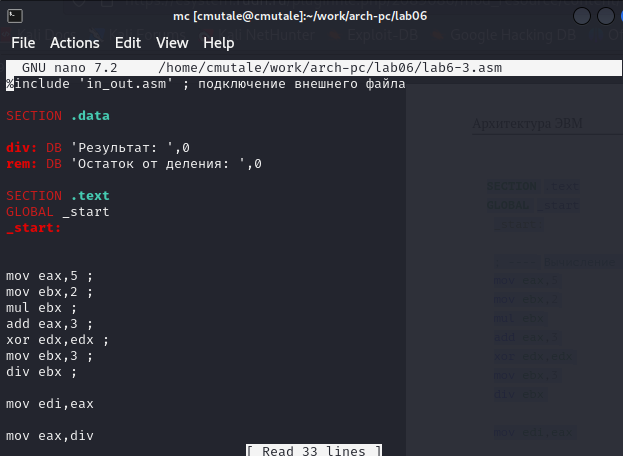


Рис 16

Создаю исполняемый файл и запускаю его:

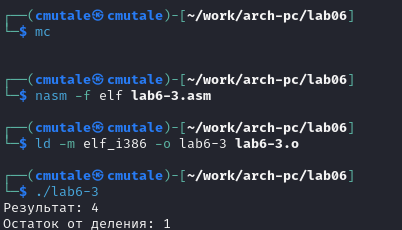


Рис 17

Изменяю текст программы для вычисления выражения f(x)=(4 \* 6 + 2)/5:

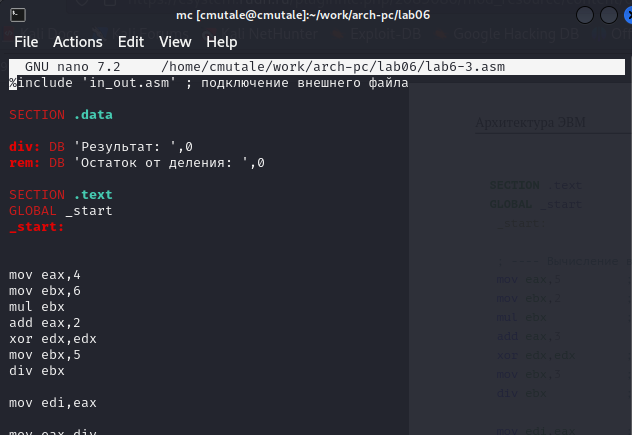


Рис 18

Создаю исполняемый файл и запускаю его:

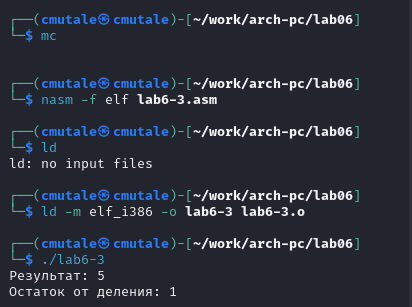


Рис 19

Создаю файл variant.asm:



Рис 20

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета:

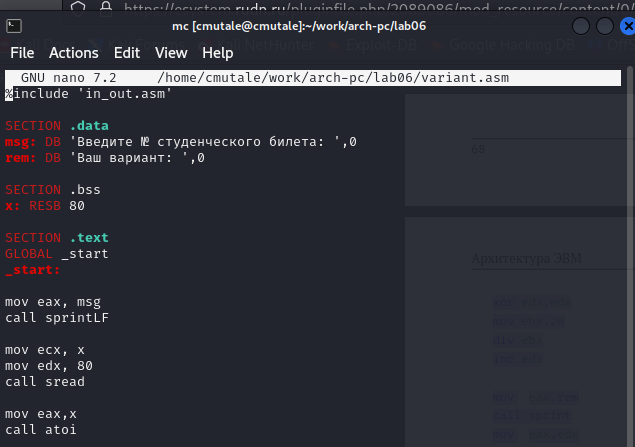


Рис 21

Создаю исполняемый файл и запускаю его. Ввожу номер своего студенческого билета, программа выводит, вариант 10:



Рис 22

**3. Ответы на вопросы по программе**

1. За вывод сообщения “Ваш вариант” отвечают строки кода:

mov eax,rem  
call sprint

1. 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 ообщения с клавиатуры
2. call atoi используется для вызова подпрограммы из внешнего файла, кото рая преобразует ascii-код символа в целое число и записывает результат в регистр eax
3. За вычисления варианта отвечают строки:

xor edx,edx   
mov ebx,20   
div ebx   
inc edx

1. Остаток от деления записывается в регистр edx.
2. Инструкция inc edx используется для увеличения значения регистра edx на 1.
3. За вывод на экран результатов вычислений отвечают строки:

mov eax,edx  
call iprintLF

# 4 Выполнение самостоятельной работы

Создаю файл task.asm:



Рис 23

В него пишу программу для вычисления выражения f(x)=5(x + 18) - 28. Она берет входное значение ” х “, прибавляет его к 18, умножает результат сложения на 5 а потом вычитает 28:

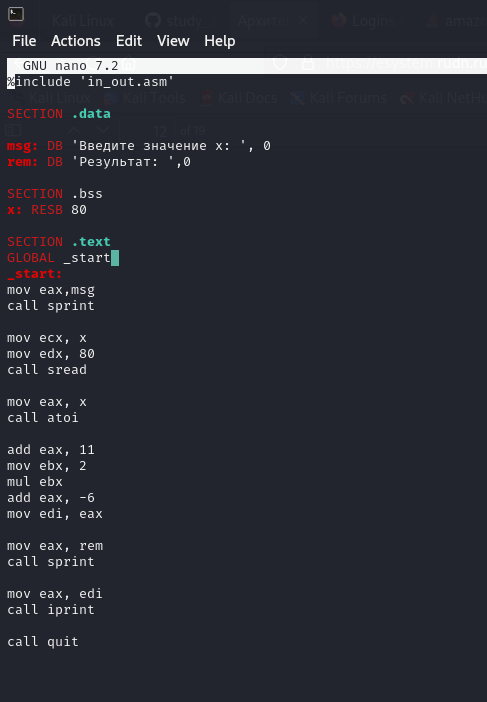


Рис 24

Создаю исполняемый файл и запускаю его. При x = 2, она выводит 72. При x = 3, она выводит 77:



Рис 25

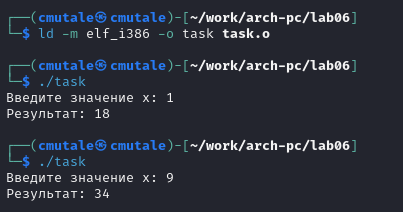


Рис 26

Код Программы:

%include 'in\_out.asm'   
  
SECTION .data   
msg: DB 'Введите значение переменной х: ',0  
rem: DB 'Результат: ',0  
  
SECTION .bss   
x: RESB 80   
SECTION .text   
GLOBAL \_start   
\_start:   
mov eax, msg   
call sprint  
   
mov ecx, x   
mov edx, 80   
call sread   
  
mov eax,x   
call atoi   
  
add eax,11

mov ebx,2  
mul ebx  
add eax,-6  
mov edi,eax   
  
mov eax,rem   
call sprint   
  
mov eax,edi   
call iprint   
call quit

# 5 Выводы

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

# Список литературы

[Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089086/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%966.%20%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8%20%D0%B2%20NASM..pdf) :::