Отчёт по лабораторной работе №6

Дисциплина: архитектура компьютера

Горобцова Арина Романовна

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 Выполнение лабораторной работы

## 3.1 Символьные и численные данные в NASM

С помощью утилиты mkdir создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №6. Перехожу в созданный каталог с помощью утилиты cd и создаю файл touch lab6-1.asm

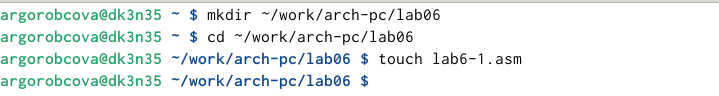


Рис. 1: Создание директории и файла

Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax (рис. 2).

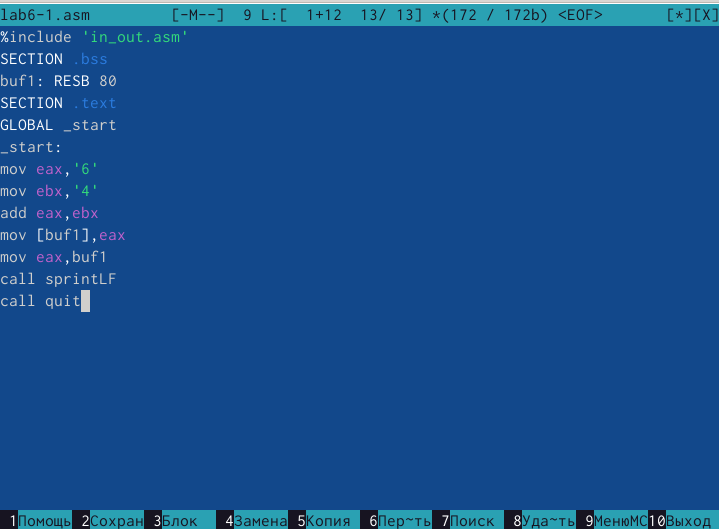


Рис. 2: Редактирование файла

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. 3).

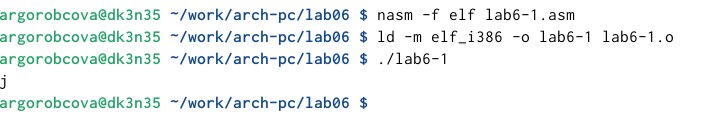


Рис. 3: Запуск исполняемого файла

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

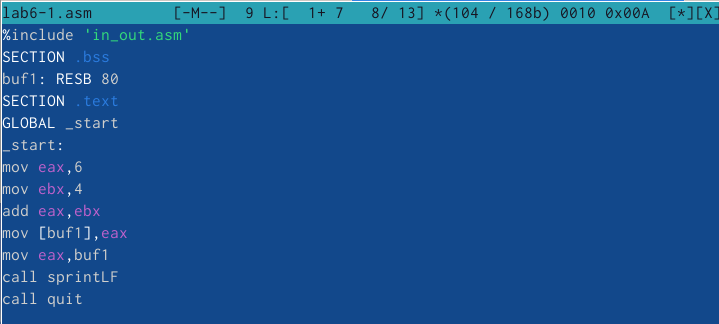


Рис. 4: Редактирование файла

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. 5).

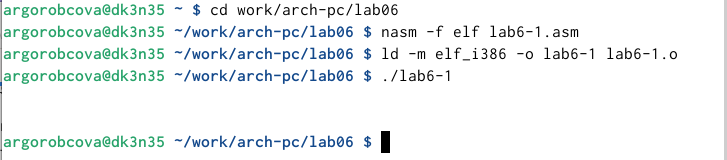


Рис. 5: Запуск исполняемого файла

Создаю новый файл lab6-2.asm с помощью утилиты touch (рис. 6).

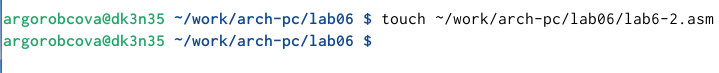


Рис. 6: Создание файла

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы (рис. 7).

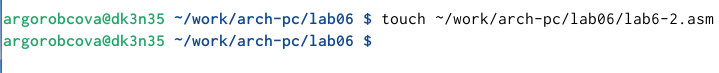


Рис. 7: Открытие файла для просмотра

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

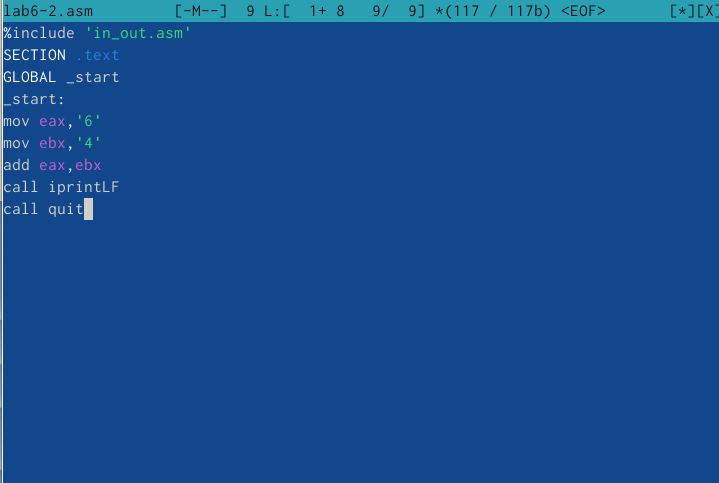


Рис. 8: Редактирование файла

Создаю и запускаю исполняемый файл lab6-2. 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. 9).

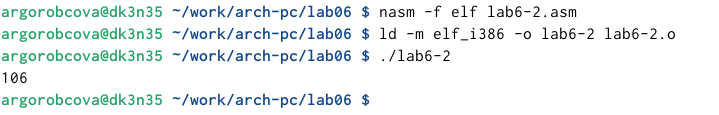


Рис. 9: Запуск исполняемого файла

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 10).

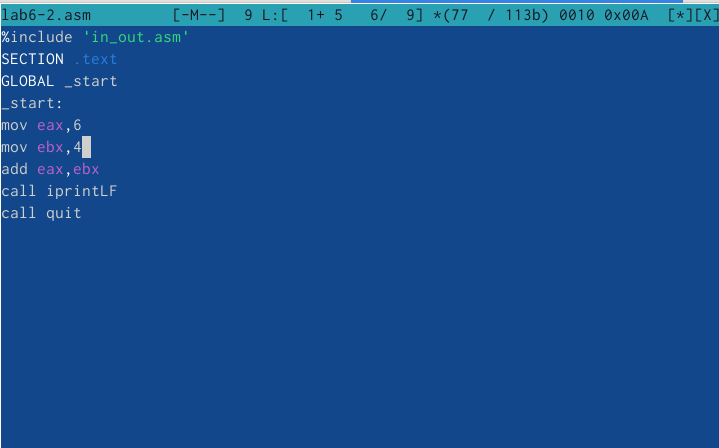


Рис. 10: Редактирование файла

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

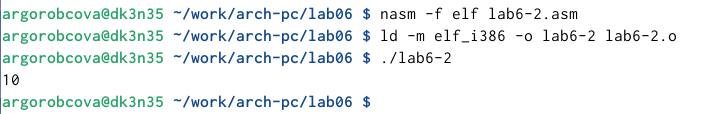


Рис. 11: Запуск исполняемого файла

Заменяю в тексте программы функцию iprintLF на iprint (рис. 12).

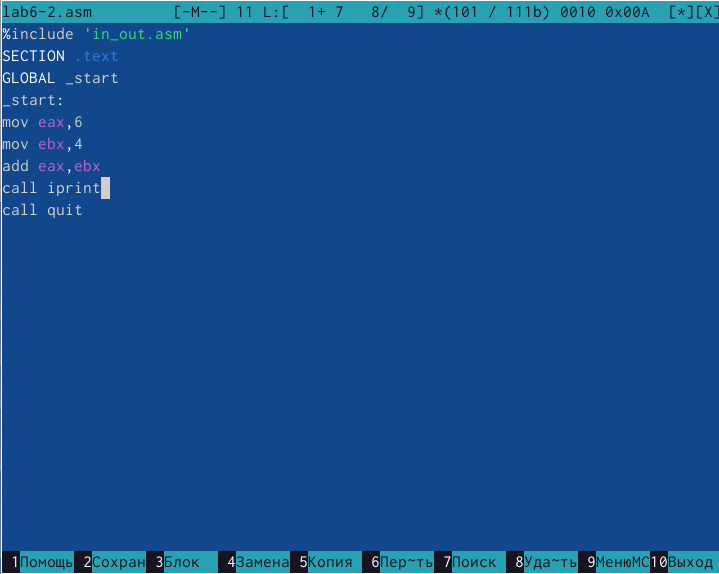


Рис. 12: Редактирование файла

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. 13).

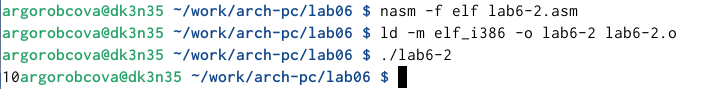


Рис. 13: Запуск исполняемого файла

## 3.2 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm с помощью утилиты touch (рис. 14).

Рис. 14: Создание файла

Рис. 14: Создание файла

Ввожу в созданный файл текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис. 15).

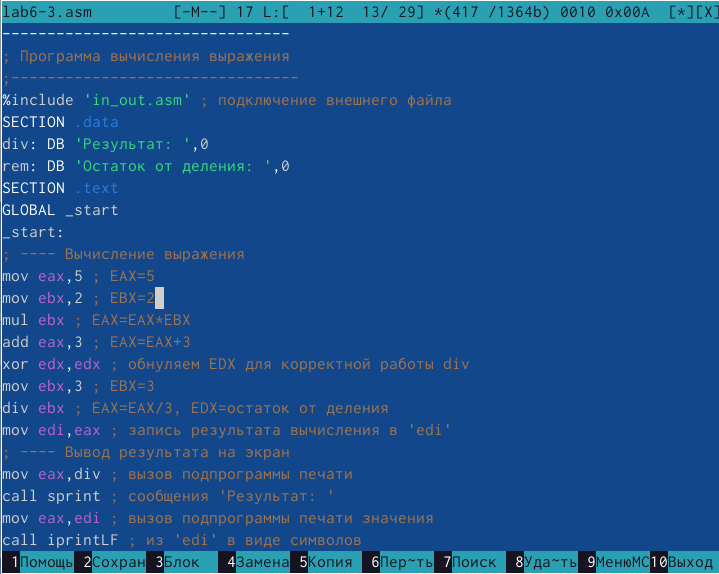


Рис. 15: Редактирование файла

Создаю исполняемый файл и запускаю его (рис. 16).

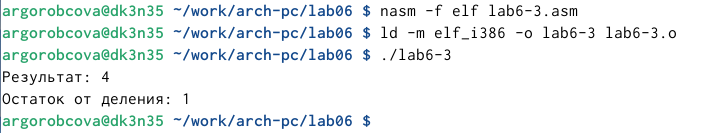


Рис. 16: Запуск исполняемого файла

Изменяю программу так, чтобы она вычисляла значение выражения f(x) = (4 \* 6 + 2)/5 (рис. 17).

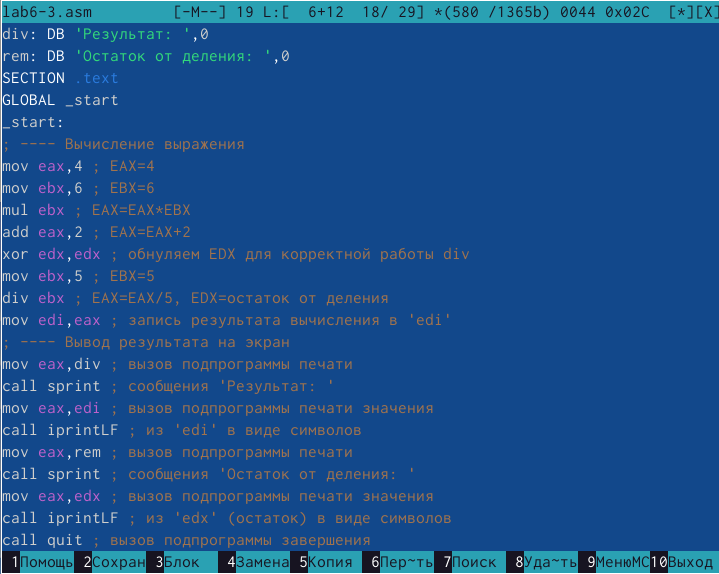


Рис. 17: Изменение программы

Создаю и запускаю новый исполняемый файл. Я посчитала для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно (рис. 18).

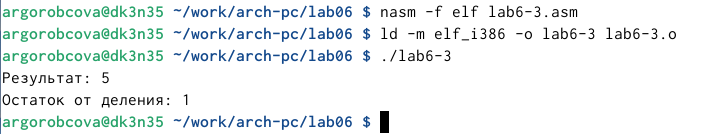


Рис. 18: Запуск исполняемого файла

Создаю файл variant.asm с помощью утилиты touch (рис. 19).

Рис. 19: Создание файла

Рис. 19: Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 20).

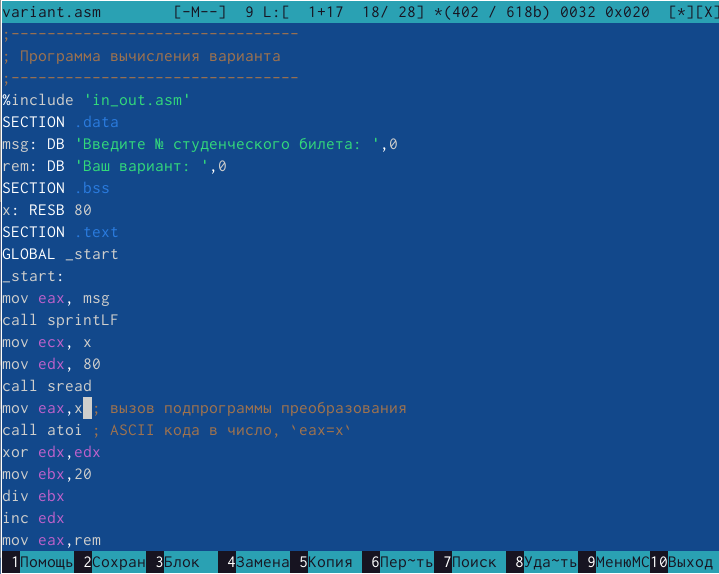


Рис. 20: Редактирование файла

Создаю и запускаю исполняемый файл. Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 20. (рис. 21).

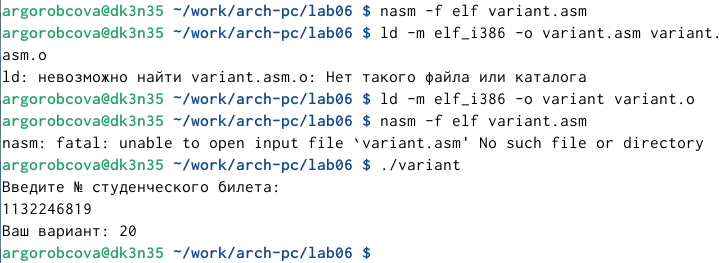


Рис. 21: Запуск исполняемого файла

## 3.3 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

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

## 3.4 Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm с помощью утилиты touch (рис. 22).

Рис. 22: Создание файла

Рис. 22: Создание файла

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения x^3\*1/3+21. Это выражение было под вариантом 20. (рис. 23).

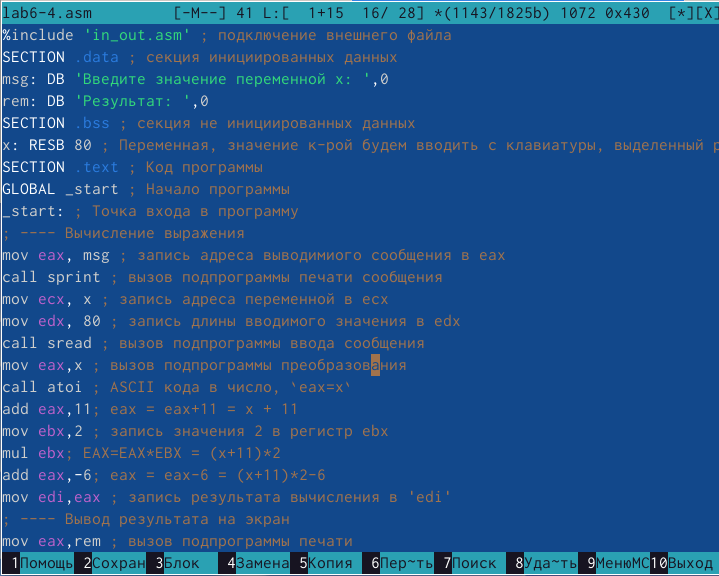


Рис. 23: Написание программы

Создаю и запускаю исполняемый файл. При вводе значения 21, вывод - 48. (рис. 24).

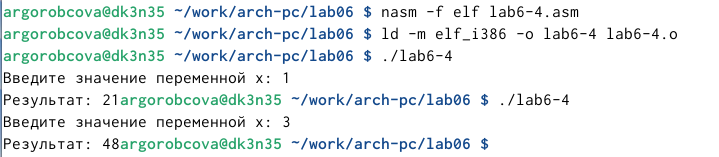


Рис. 24: Запуск исполняемого файла

Листинг 4.1. Программа для вычисления значения выражения (11 + x) \* 2 − 6.

%include ‘in\_out.asm’ ; подключение внешнего файла

SECTION .data ; секция инициированных данных

msg: DB ‘Введите значение переменной х:’,0

rem: DB ‘Результат:’,0

SECTION .bss ; секция не инициированных данных

x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенный размер - 80 байт

SECTION .text ; Код программы

GLOBAL \_start ; Начало программы

\_start: ; Точка входа в программу

; —- Вычисление выражения

mov eax, msg ; запись адреса выводимиого сообщения в eax

call sprint ; вызов подпрограммы печати сообщения

mov ecx, x ; запись адреса переменной в ecx

mov edx, 80 ; запись длины вводимого значения в edx

call sread ; вызов подпрограммы ввода сообщения

mov eax,x ; вызов подпрограммы преобразования

call atoi ; ASCII кода в число, eax=x

add eax,11; eax = eax+11 = x + 11

mov ebx,2 ; запись значения 2 в регистр ebx

mul ebx; EAX=EAX*EBX = (x+11)*2

add eax,-6; eax = eax-6 = (x+11)\*2-6

mov edi,eax ; запись результата вычисления в ‘edi’

; —- Вывод результата на экран

mov eax,rem ; вызов подпрограммы печати

call sprint ; сообщения ‘Результат:’

mov eax,edi ; вызов подпрограммы печати значения

call iprint ; из ‘edi’ в виде символов

call quit ; вызов подпрограммы завершения

# 4 Выводы

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