Отчёт по лабораторной работе №6

Дисциплина: архитектура компьютера

Арфонос Дмитрий

Содержание

# 1 Цель работы

Цель данной лабораторной работы - освоение арифметческих инструкций низкоуровневого 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

**1**

С помощью утилиты mkdir создаю директорию lab06, перехожу в нее и создаю файл для работы. (рис. [[1](#fig:001)])

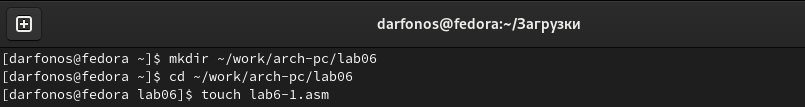


Figure 1: Создание директории

**2**

Копирую в текущий каталог файл in\_out.asm из загрузок с помощью утилиты cp, т.к. он будет использоваться в других программах (рис. [[2](#fig:002)]).

Figure 2: Создание копии файла для дальнейшей работы

Figure 2: Создание копии файла для дальнейшей работы

**3**

Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax (рис. [[3](#fig:003)]).



Figure 3: Редактирование файла

**4**

Создаю исполняемый файл программы и запускаю его (рис. [[4](#fig:004)]). Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6.

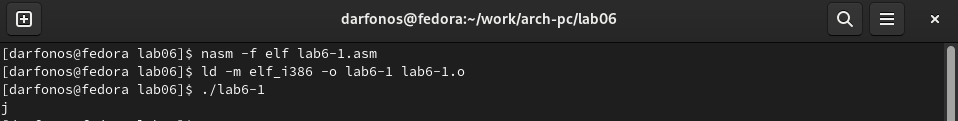


Figure 4: Запуск исполняемого файла

**5**

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. [[5](#fig:005)]).



Figure 5: Редактирование файла

**6**

Создаю новый исполняемый файл программы и запускаю его (рис. [[6](#fig:006)]). Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.

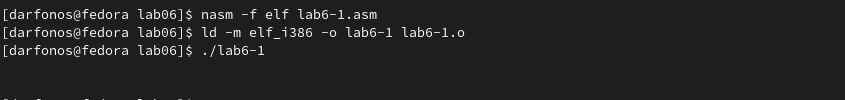


Figure 6: Запуск исполняемого файла

**7**

Создаю новый файл lab6-2.asm с помощью утилиты touch (рис. [[7](#fig:007)]).

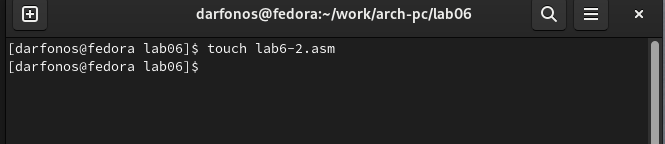


Figure 7: Создание файла

**8**

Ввожу в файл текст другой программы для вывода значения регистра eax (рис. [[8](#fig:008)]).

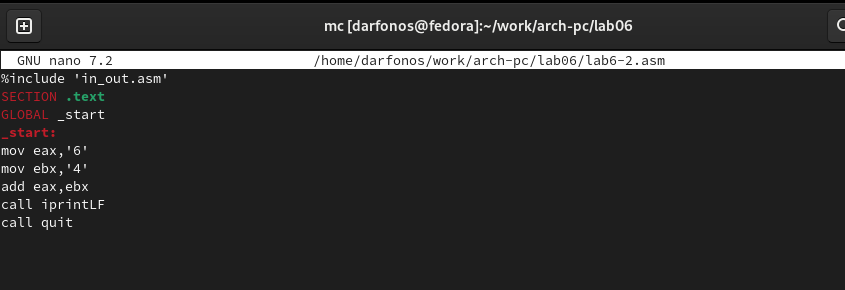


Figure 8: Редактирование файла

**9**

Создаю и запускаю исполняемый файл lab6-2 (рис. [[9](#fig:009)]). 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”.

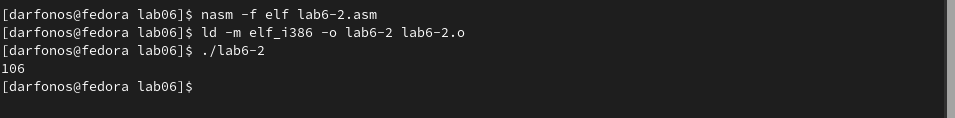


Figure 9: Запуск исполняемого файла

**10**

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. [[10](#fig:010)]).



Figure 10: Редактирование файла

**11**

Создаю и запускаю новый исполняемый файл (рис. [[11](#fig:011)]).. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.

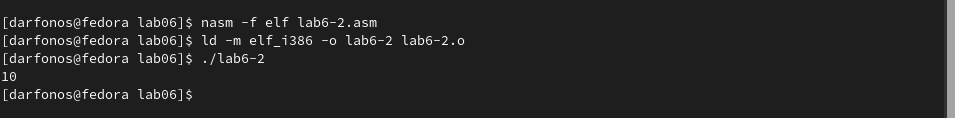


Figure 11: Запуск исполняемого файла

**12**

Заменяю в тексте программы функцию iprintLF на iprint (рис. [[12](#fig:012)]).

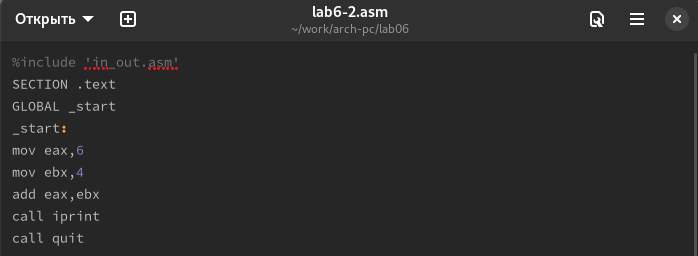


Figure 12: Редактирование файла

**13**

Создаю и запускаю новый исполняемый файл (рис. [[13](#fig:013)]). 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.

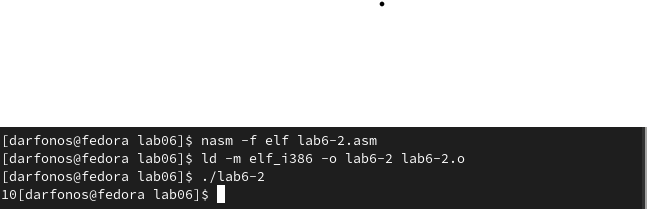


Figure 13: Запуск исполняемого файла

## 3.2 Выполнение арифметических операций в NASM

**14**

Создаю файл lab6-3.asm с помощью утилиты touch (рис. [[14](#fig:014)]).



Figure 14: Создание файла

**15**

Ввожу в созданный файл текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис. [[15](#fig:015)]).

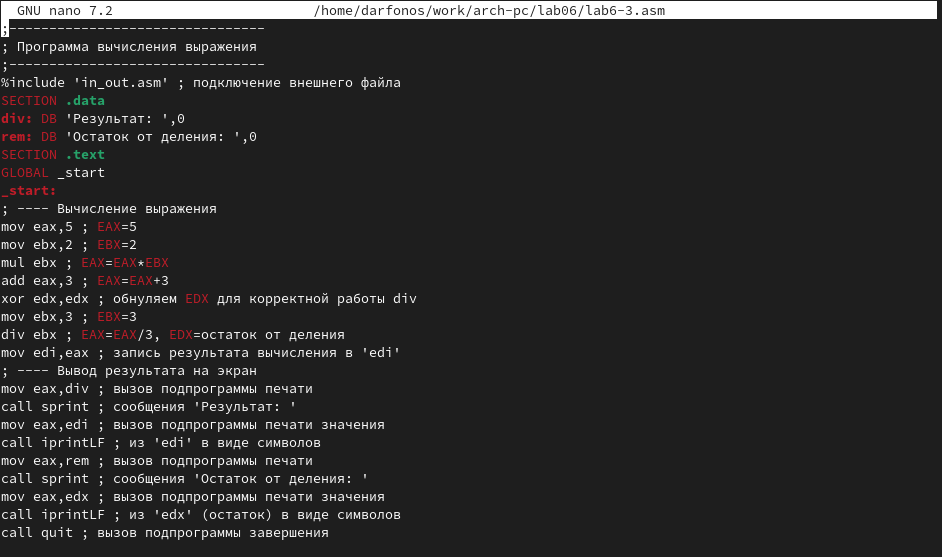


Figure 15: Редактирование файла

**16**

Создаю исполняемый файл и запускаю его (рис. [[16](#fig:016)]).

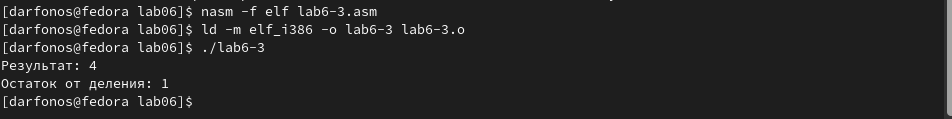


Figure 16: Запуск исполняемого файла

**17**

Изменяю программу так, чтобы она вычисляла значение выражения f(x) = (4 \* 6 + 2)/5 (рис. [[17](#fig:017)]).

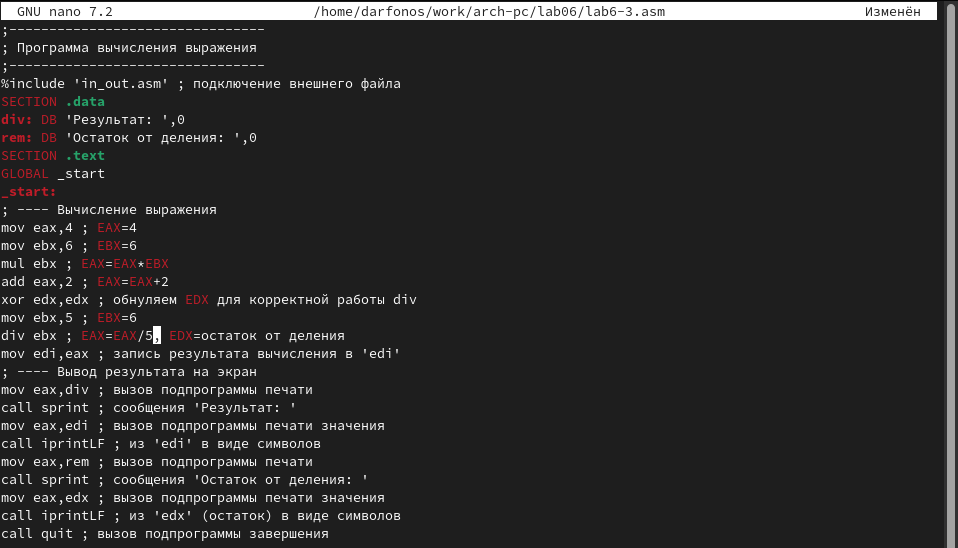


Figure 17: Изменение программы

**18**

Создаю и запускаю новый исполняемый файл (рис. [[18](#fig:018)]). Программа отработала верно.

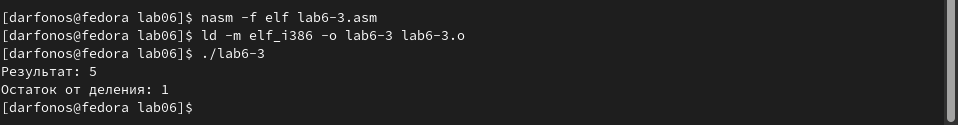


Figure 18: Запуск исполняемого файла

**19**

Создаю файл variant.asm с помощью утилиты touch (рис. [[19](#fig:019)]).

Figure 19: Создание файла

Figure 19: Создание файла

**20**

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. [[20](#fig:020)]).



Figure 20: Редактирование файла

**21**

Создаю и запускаю исполняемый файл (рис. [[21](#fig:021)]). Ввожу номер своего студ. билета “1032235421” с клавиатуры, программа вывела, что мой вариант - 2.

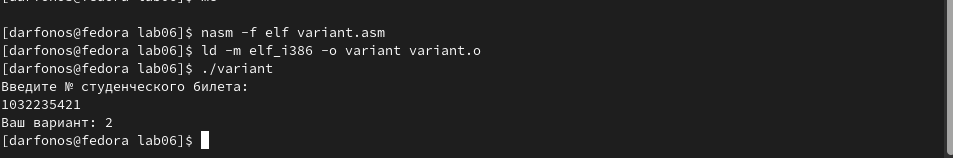


Figure 21: Запуск исполняемого файла

### 3.2.1 Ответы на вопросы по программе

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

mov eax,rem  
call sprint

1. 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
2. call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax
3. 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

## 3.3 Выполнение заданий для самостоятельной работы

**Шаг 1**

Создаю файл lab6-4.asm с помощью утилиты touch (рис. [[22](#fig:022)]).

Figure 22: Создание файла

Figure 22: Создание файла

**Шаг 2**

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения 2 варианта: (рис. [[23](#fig:023)]). *f(x) = (3+12x)5*

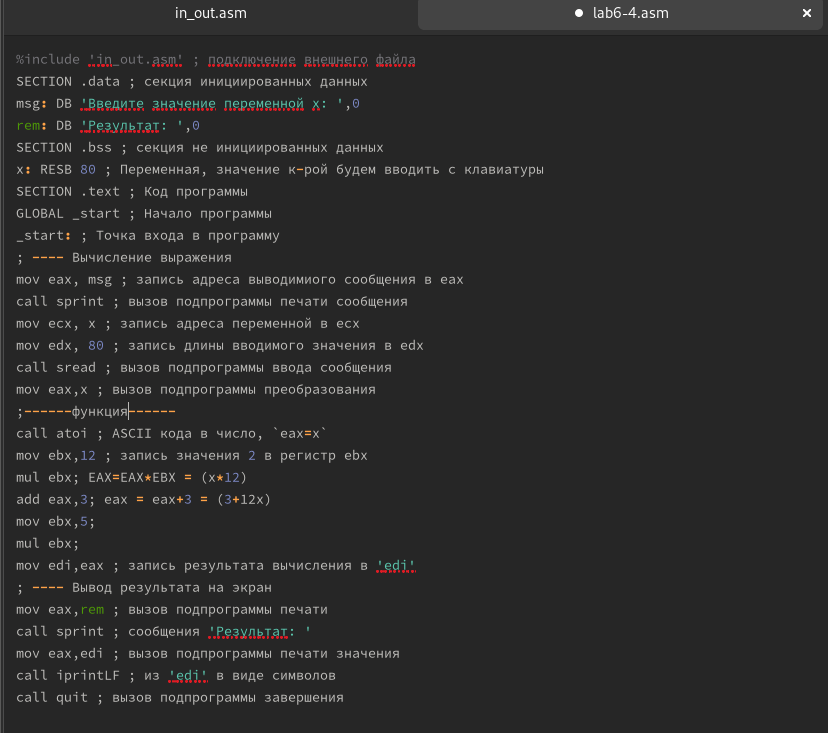


Figure 23: Написание программы

**Шаг 3**

1. Создаю и запускаю исполняемый файл при вводе двух значений (рис. [[24](#fig:024)]).

* x = 1, f(x) = 75
* x = 6, f(x) = 375

Программа отработала верно.

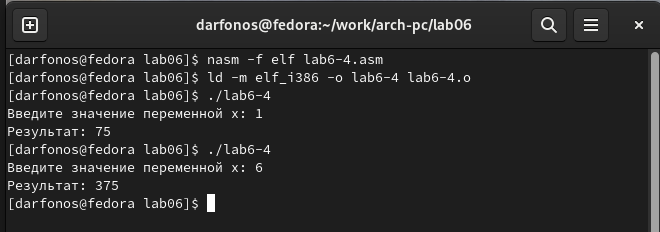


Figure 24: Запуск исполняемого файла

**Текст программы для вычисления значения выражения f(x)=(3+12x)5**

%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  
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  
;------функция------  
call atoi ; ASCII кода в число, `eax=x`  
mov ebx,12 ; запись значения 2 в регистр ebx  
mul ebx; EAX=EAX\*EBX = (x\*12)  
add eax,3 ; eax = eax+3 = (3+12x)  
mov ebx,5 ; ebx =5  
mul ebx ; EAX=EAX\*EBX = (3+12x)\*5  
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 iprintLF ; из 'edi' в виде символов  
call quit ; вызов подпрограммы завершения

# 4 Выводы

При выполнении данной лабораторной работы я освоил арифметические инструкции: сложение, вычитание, умножение и деление на языке ассемблер NASM.