Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Козина Дарья Александровна

Содержание

Список иллюстраций

Список таблиц

# Цель работы

Освоить арифметические инструкции языка ассемблера NASM.

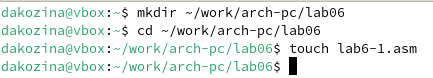
# Задание

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

# Выполнение лабораторной работы

## Символьные и численные данные в NASM

С помощью команды mkdir создадим каталог для выполнения лабораторной работы № 6. С помощью команды touch в созданном каталоге создадим файл lab6-1.asm (рис. [-@fig:001]).



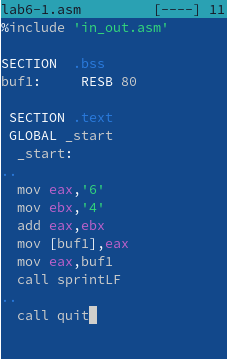
Создание каталога и файла

Для того, чтобы программы корректно работали, копирую файл in\_out.asm из каталога, созданного при выполнении прошлой лабораторной работы в только что созданный каталог (рис. [-@fig:002]).

Копирование файла

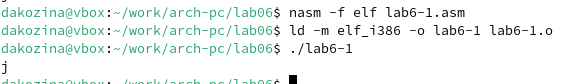
Копирование файла

Введем программу в файл lab6-1.asm с помощью редактора mcedit (рис. [-@fig:003]).



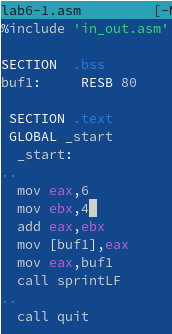
Редактирование файла

Создадим исполняемый файл и запустим его (рис. [-@fig:004]).



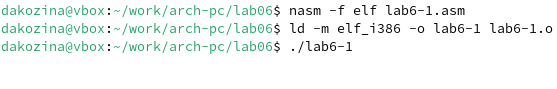
Создание и запуск файла

Изменим текст программы. В регистр вместо символов ‘6’ и ‘4’ запишем числа 6 и 4 (рис. [-@fig:005]).



Изменение программы

Создадим исполняемый файл и запустим его (рис. [-@fig:006]). Теперь выводится символ с кодом 10. Этот символ не вывелся на экран.



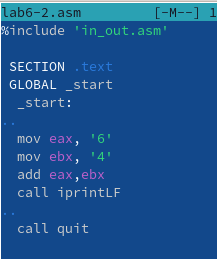
Создание и запуск файла

С помощью команды touch создадим файл lab6-2.asm (рис. [-@fig:007]).

Создание файла

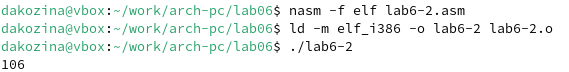
Создание файла

Введем текст программы в созданный файл с помощью редактора mcedit (рис. [-@fig:008]).



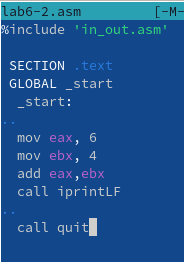
Редактирование файла

Создадим исполняемый файл и запустим его (рис. [-@fig:009]).



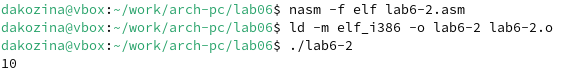
Создание и запуск файла

Изменим текст программы. В регистр вместо символов ‘6’ и ‘4’ запишем числа 6 и 4 (рис. [-@fig:010]).



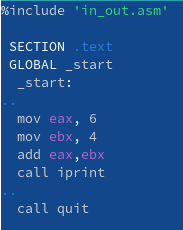
Изменение программы

Создадим исполняемый файл и запустим его (рис. [-@fig:011]). Выводится число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит сложение символов ‘6’ и ‘4’.



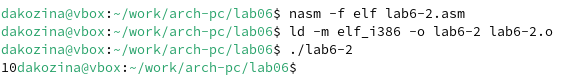
Создание и запуск файла

Заменим функцию iprintLF на iprint (рис. [-@fig:012]).



Замена функции

Создадим исполняемый файл и запустим его (рис. [-@fig:013]). Вывод не изменился. Символ переноса строки не отображался, когда программа исполнялась с функцией iprintLF, а iprint не добавляет к выводу символ переноса строки, в отличие от iprintLF.



Создание и запуск файла

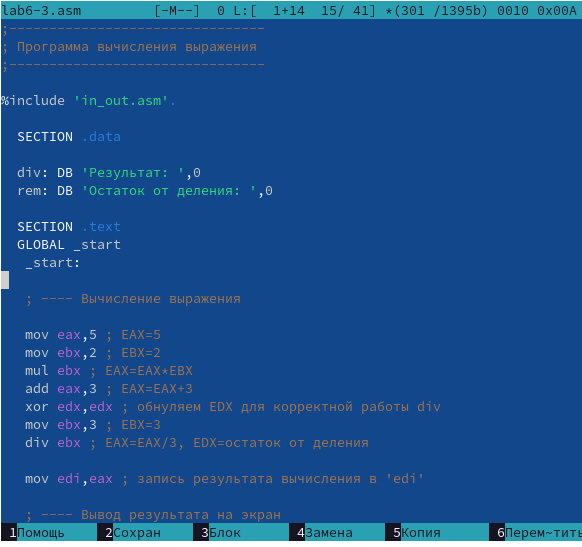
## Выполнение арифметических операций в NASM

С помощью команды touch создадим файл lab6-3.asm (рис. [-@fig:014]).

Создание файла

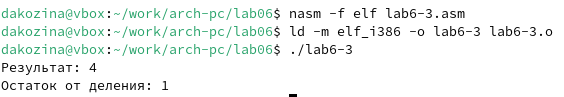
Создание файла

Введем текст программы вычисления выражения f(x) = (5\*2 + 3)/3 в созданный файл с помощью редактора mcedit (рис. [-@fig:015]).



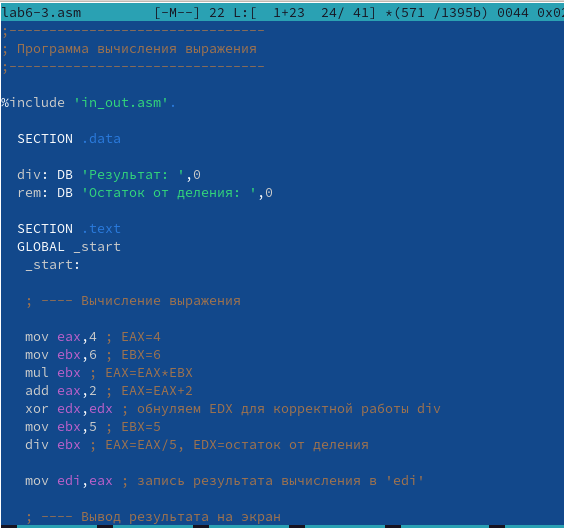
Редактирование файла

Создадим исполняемый файл и запустим его (рис. [-@fig:016]).



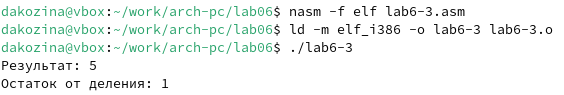
Создание и запуск файла

Изменим текст программы в редакторе mcedit таким образом, чтобы она вычислила f(x) = (4\*6 + 2)/5 (рис. [-@fig:017]).



Редактирование программы

Создадим исполняемый файл и запустим его (рис. [-@fig:018]).



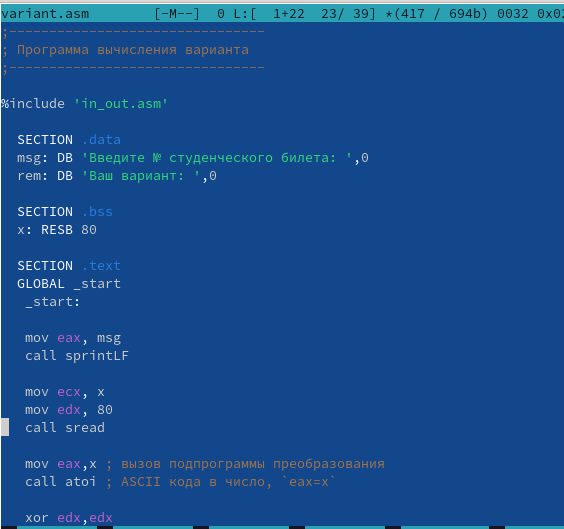
Создание и запуск файла

С помощью команды touch создадим файл variant.asm (рис. [-@fig:019]).

Создание файла

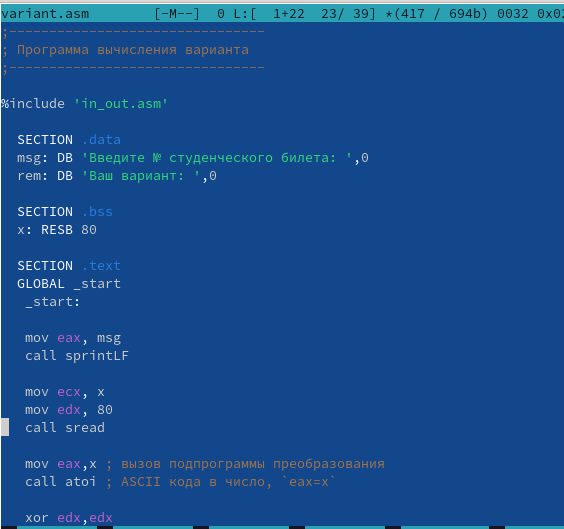
Создание файла

Введем текст программы вычисления варианта задания по номеру студенческого билета в созданный файл с помощью редактора mcedit (рис. [-@fig:020]).



Редактирование файла

Создадим исполняемый файл и запустим его (рис. [-@fig:021]). Программа вывела, что мой вариант - 12.



Создание и запуск файла

## Ответы на вопросы

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;

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;

mov eax,edx  
call iprintLF

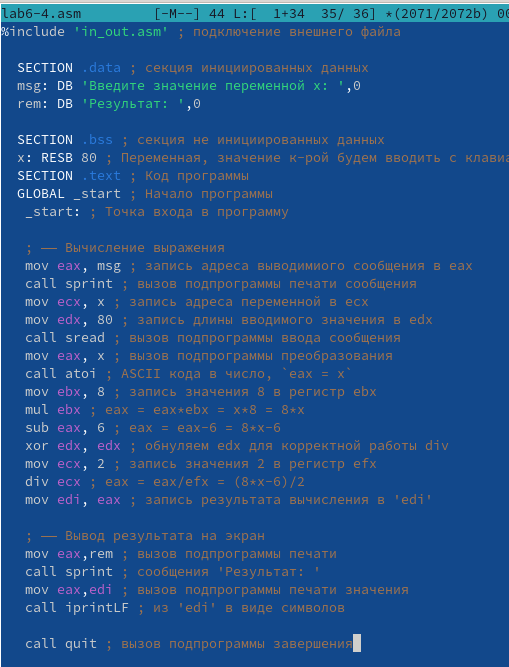
# Выполнение заданий для самостоятельной работы

С помощью команды touch создадим файл lab6-4.asm (рис. [-@fig:022]).

Создание файла

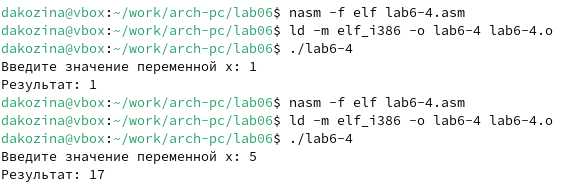
Создание файла

Введем текст программы вычисления выражения, которое соответствует варианту по номеру моего студенческого билета, а именно f(x) = (8\*x - 6)/2 с помощью редактора mcedit (рис. [-@fig:023]).



Редактирование файла

Создадим исполняемый файл и запустим его (рис. [-@fig:024]).



Создание и запуск файла

**Листинг программы**

%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`  
 mov ebx, 8 ; запись значения 8 в регистр ebx  
 mul ebx ; eax = eax\*ebx = x\*8 = 8\*x  
 sub eax, 6 ; eax = eax-6 = 8\*x-6  
 xor edx, edx ; обнуляем edx для корректной работы div  
 mov ecx, 2 ; запись значения 2 в регистр efx  
 div ecx ; eax = eax/efx = (8\*x-6)/2  
 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

# Выводы

В ходе лабораторной работы мы освоили арифметические инструкции языка ассемблера NASM.