Лабораторная работа №6

Арифметические операции в NASM

Глобин Никита Анатольевич

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

Символьные и численные данные в NASM  
Выполнение арифметических операций в NASM  
Ответы на вопросы  
Задание для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Символьные и численные данные в NASM

1. переходим в катал lab06 и создаём там файл lab6-1.asm (рис. 1).

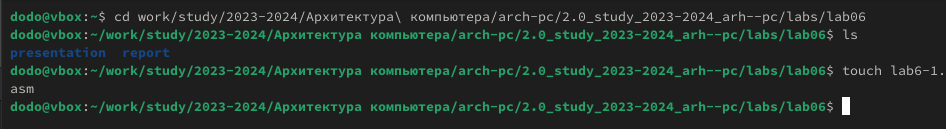


Рис. 1: photo 1

переписываем код программы из ТУИС (рис. 2).

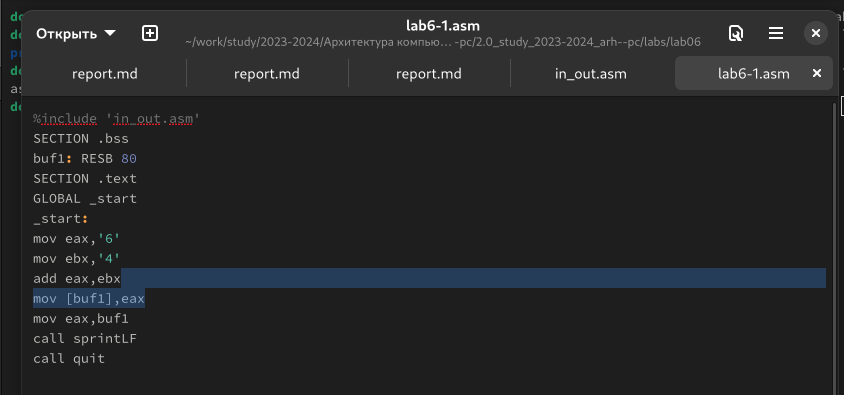


Рис. 2: photo 2

компилируем и запускаем (рис. 3).

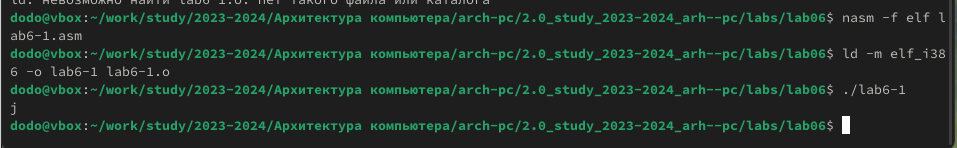


Рис. 3: photo 3

1. Далее изменим текст программы и вместо символов, запишем в регистры числа. (рис. 4).

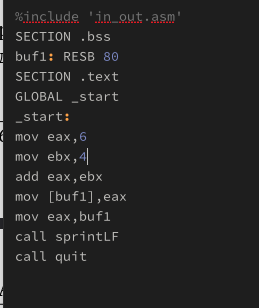


Рис. 4: photo 4

компилируем и запускаем (рис. 5).

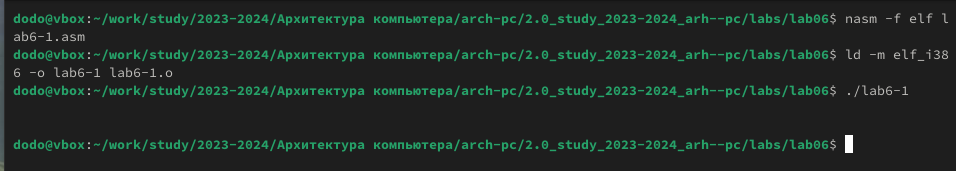


Рис. 5: photo 5

1. создаём новый файл lab6-2 (рис. 6).

photo 6

Рис. 6: photo 6

переписываем код программы из ТУИС (рис. 7).

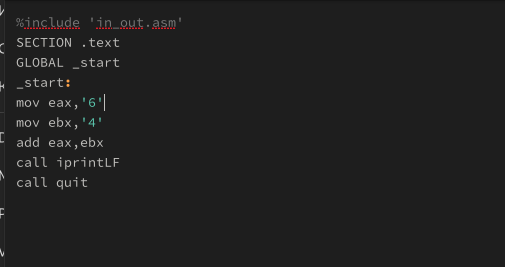


Рис. 7: photo 7

компилируем и запускаем (рис. 8).

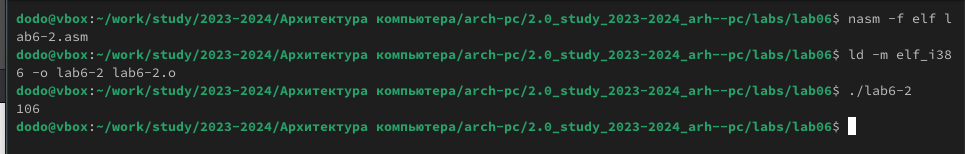


Рис. 8: photo 8

1. Аналогично предыдущему примеру изменим символы на числа (рис. 9).

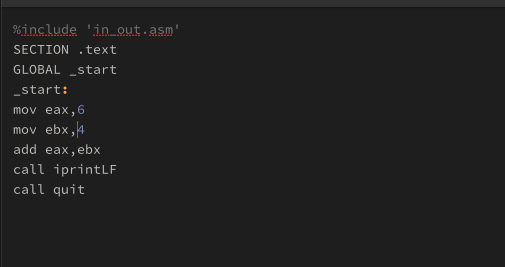


Рис. 9: photo 9

компилируем и запускаем (рис. 10).

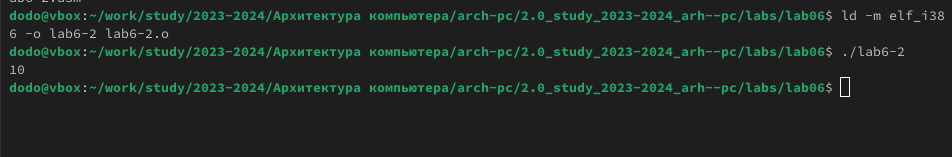


Рис. 10: photo 10

1. Заменим функцию iprintLF на iprint и выведи это (рис. 11).

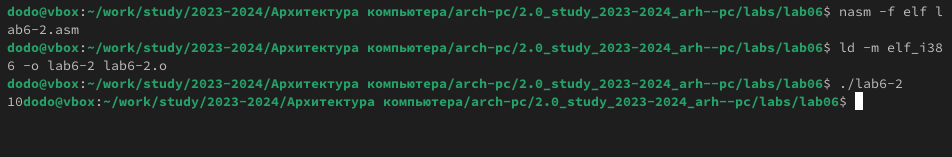


Рис. 11: photo 11

теперь ответ не пишется в отдельной строке.

## 3.2 Выполнение арифметических операций в NASM

1. Создаём файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 12).

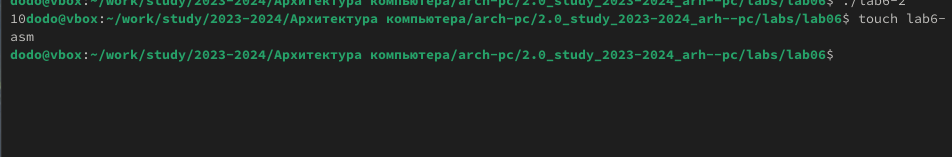


Рис. 12: photo 12

1. переписываем код программы из ТУИС (рис. 13).

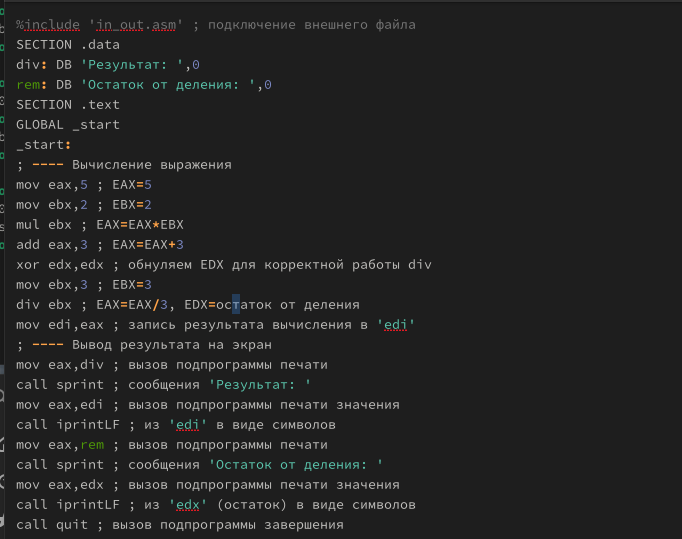


Рис. 13: photo 13

1. компилируем и запускаем (рис. 14).

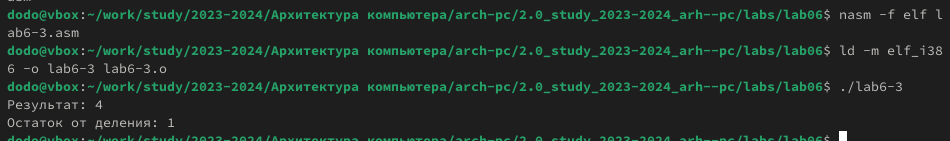


Рис. 14: photo 14

1. Изменяем текст программы для вычисления выражения f(x) = (4 \* 6 + 2)/5 (рис. 15).

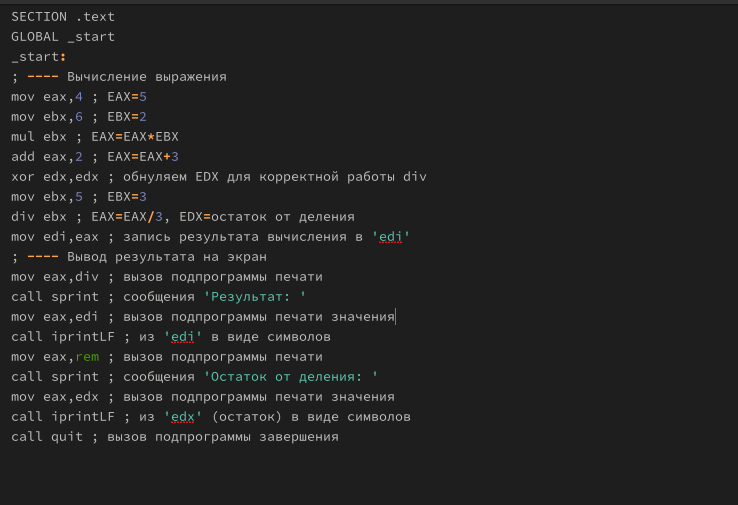


Рис. 15: photo 15

1. компилируем и запускаем (рис. 16).

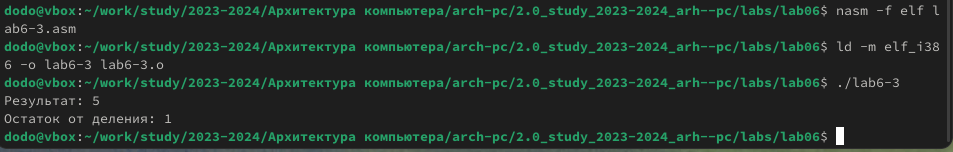


Рис. 16: photo 16

1. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму (рис. 17).

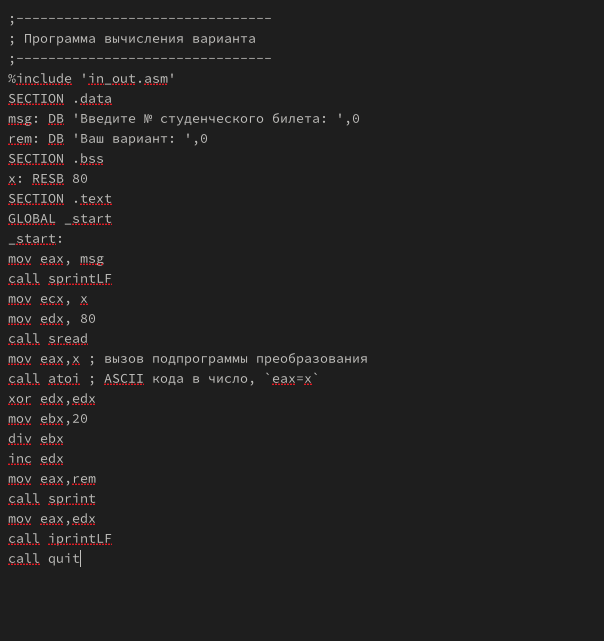


Рис. 17: photo 17

1. компилируем и запускаем (рис. 18).



Рис. 18: photo 18

## 3.3 Ответы на вопросы

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? Строки:

mov eax, rem ; Загрузка адреса сообщения “Ваш вариант:” в регистр EAX call sprint ; Вызов подпрограммы для вывода сообщения

1. Для чего используются следующие инструкции? mov ecx, x Устанавливает указатель на буфер x, куда будет записан введённый текст.

mov edx, 80 Указывает максимальную длину ввода (80 байт).

call sread Вызывает подпрограмму для считывания строки из ввода (с клавиатуры) и записи её в буфер x.

1. Для чего используется инструкция call atoi? Она преобразует строку (символы ASCII) из буфера x в число и записывает результат в регистр eax. Это необходимо для выполнения арифметических операций с введёнными данными.
2. Какие строки листинга 6.4 отвечают за вычисление варианта? Строки:

xor edx, edx ; Обнуление регистра EDX mov ebx, 20 ; Загрузка делителя (20) в EBX div ebx ; Деление: результат в EAX, остаток в EDX inc edx ; Увеличение остатка на 1 для получения номера варианта

1. В какой регистр записывается остаток от деления при выполнении инструкции div ebx? Остаток от деления записывается в регистр EDX.
2. Для чего используется инструкция inc edx? Инструкция inc edx увеличивает остаток от деления на единицу. Это необходимо, чтобы результат вычисления соответствовал формуле:
3. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений? Строки:

mov eax, rem ; Загрузка адреса сообщения “Ваш вариант:” в EAX call sprint ; Вывод сообщения mov eax, edx ; Загрузка номера варианта из EDX в EAX call iprintLF ; Вывод номера варианта с переводом строки

## 3.4 Задание для самостоятельной работы

1. создаём файл (рис. 19).

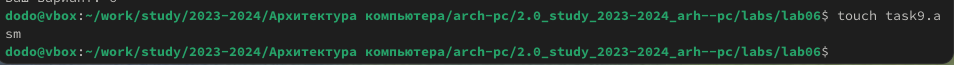


Рис. 19: photo 19

1. пишем в нём код (рис. 20).

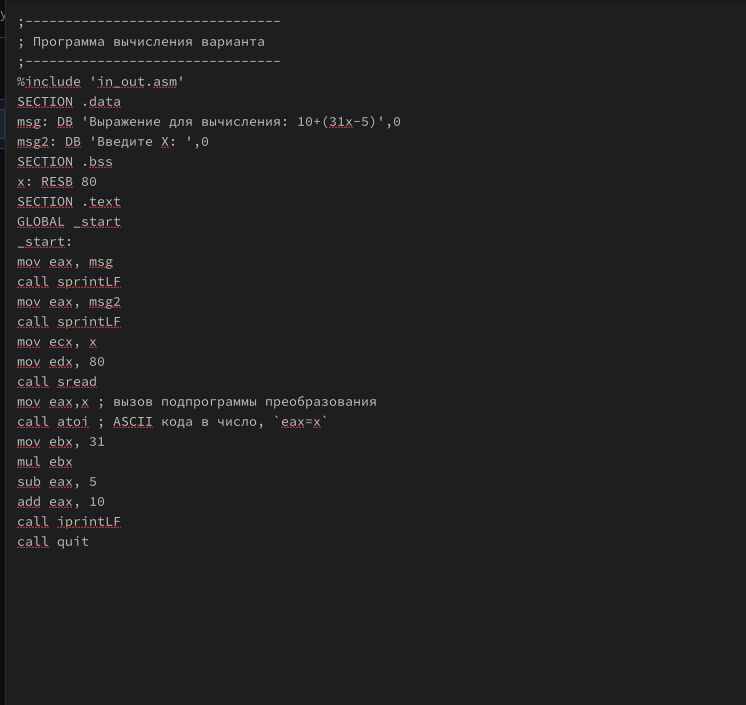


Рис. 20: photo 20

1. компилируем и запускаем (рис. 21).

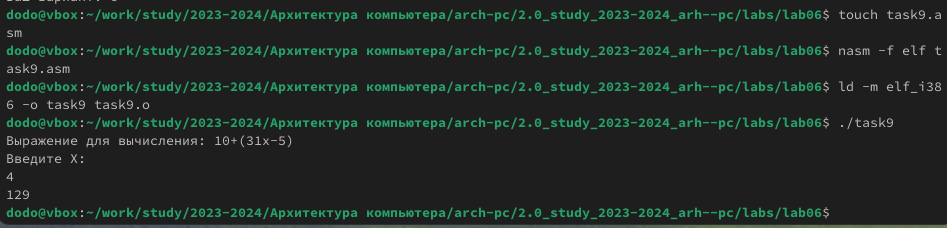


Рис. 21: photo 21

## 3.5 Выводы

В результате выполнения лабораторной работы были достигнуты следующие результаты, соответствующие поставленной цели:

1. Освоение арифметических инструкций NASM: Были изучены основные команды языка ассемблера для выполнения арифметических операций: сложение (add), вычитание (sub), умножение (mul и imul), деление (div и idiv), а также инкремент и декремент (inc, dec).
2. Работа с числовыми и символьными данными: Изучены методы преобразования данных между символьным и числовым представлением с использованием ASCII-кодов и специальных подпрограмм (atoi, iprint и др.). Это позволило корректно вводить данные с клавиатуры и выводить результаты на экран.
3. Практическое применение арифметических операций: Было успешно реализовано вычисление сложных арифметических выражений на языке NASM с учётом правил адресации данных и особенностей использования регистров.
4. Закрепление навыков программирования на ассемблере: Выполнение лабораторной работы позволило на практике закрепить теоретические знания по архитектуре ЭВМ и программированию на языке ассемблера NASM.

# Список литературы