Отчёт по лабораторной работе №6

Арифметические операция в NASM

Новикова Анастасия Андреевна

Содержание

# 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. Выполнение заданий для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Символьные и численные данные в NASM

С помощью команды mkdir создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №6 (рис. 1). Перехожу в созданный каталог с помощью команды cd.

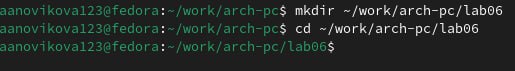


Рис. 1: Создание директории

С помощью команды touch создаю файл lab6-1.asm (рис. 2).

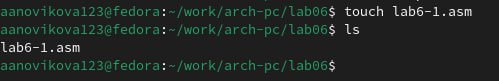


Рис. 2: Создание файла

Копирую в текущий каталог файл in\_out.asm с помощью команды cp, т.к. он будет использоваться в других программах (рис. 3).

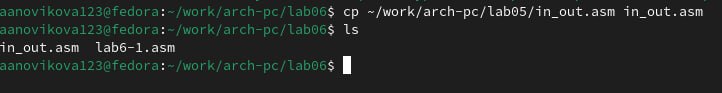


Рис. 3: Создание копии файла

Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax (рис. 4).

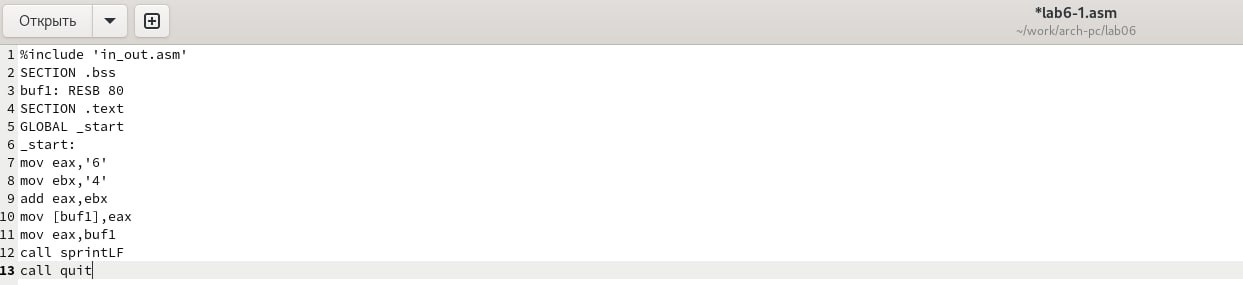


Рис. 4: Редактирование файла

Создаю исполняемый файл программы и запускаю его (рис. 5). Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6.

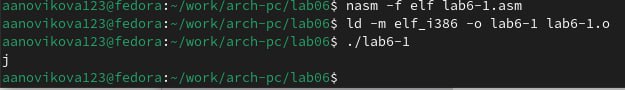


Рис. 5: Запуск исполняемого файла

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

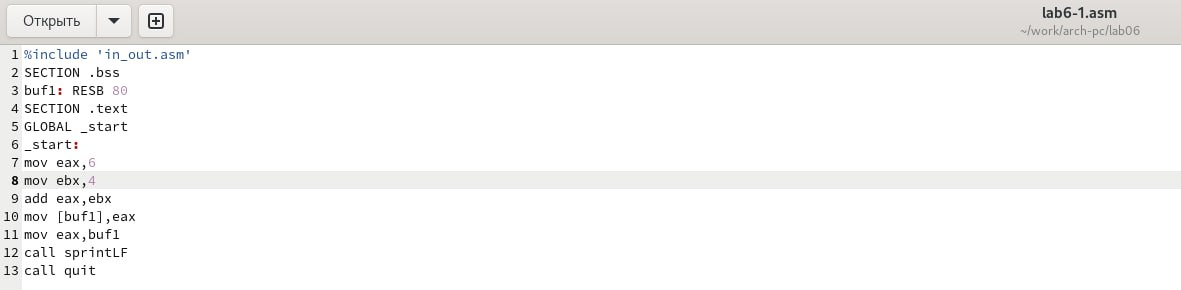


Рис. 6: Редактирование файла

Создаю новый исполняемый файл программы и запускаю его (рис. 7). Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.

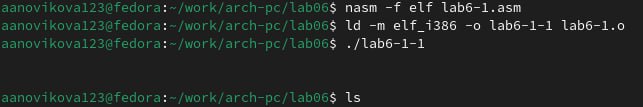


Рис. 7: Запуск исполняемого файла

Создаю новый файл lab6-2.asm с помощью команды touch (рис. 8).

Создание файла

Рис. 8: Создание файла

Ввожу в файл текст другой программы для вывода значения регистра eax (рис. 9).

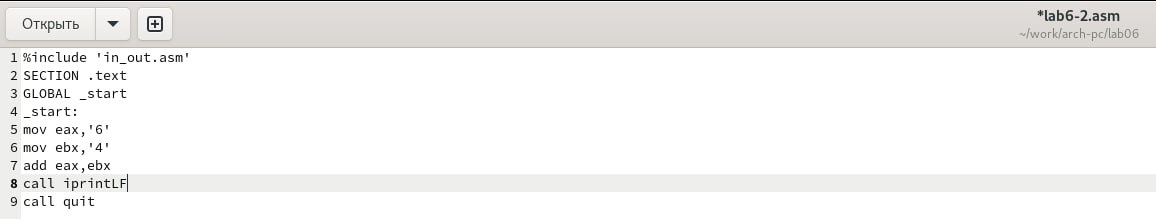


Рис. 9: Редактирование файла

Создаю и запускаю исполняемый файл lab6-2 (рис. 10). Теперь выводится число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4” (54 + 52 = 106).

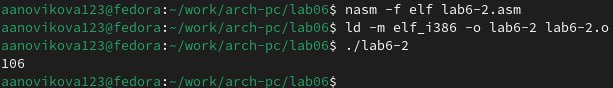


Рис. 10: Запуск исполняемого файла

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 11).

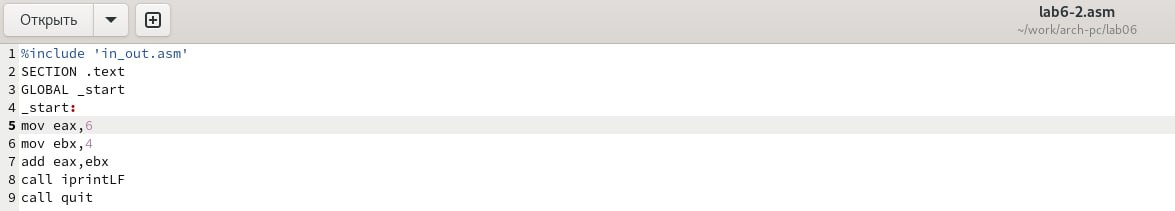


Рис. 11: Редактирование файла

Создаю и запускаю новый исполняемый файл (рис. 12). Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.

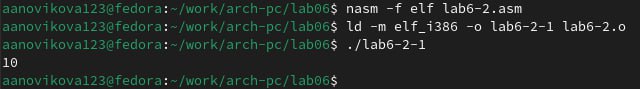


Рис. 12: Запуск исполняемого файла

Заменяю в тексте программы функцию iprintLF на iprint (рис. 13).

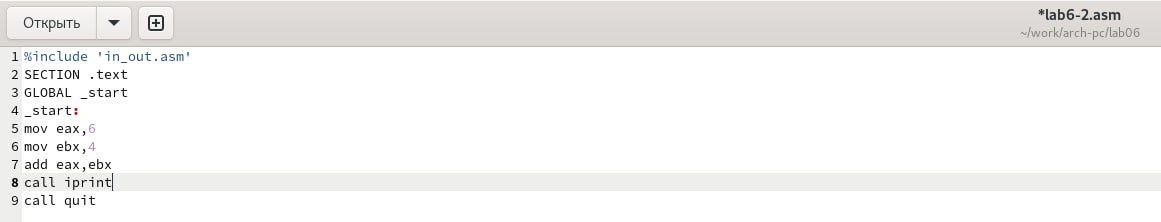


Рис. 13: Редактирование файла

Создаю и запускаю новый исполняемый файл (рис. 14). Вывод значения не изменился, но курсор остался на той же строке. То есть функция iprintLF переводит курсор на следующую строку, а iprint не добавляет к выводу символ переноса строки.

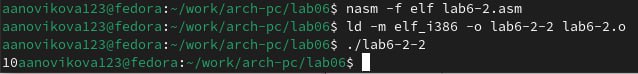


Рис. 14: Запуск исполняемого файла

## 3.2 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm с помощью команды touch (рис. 15).

Создание файла

Рис. 15: Создание файла

Ввожу в созданный файл текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис. 16).

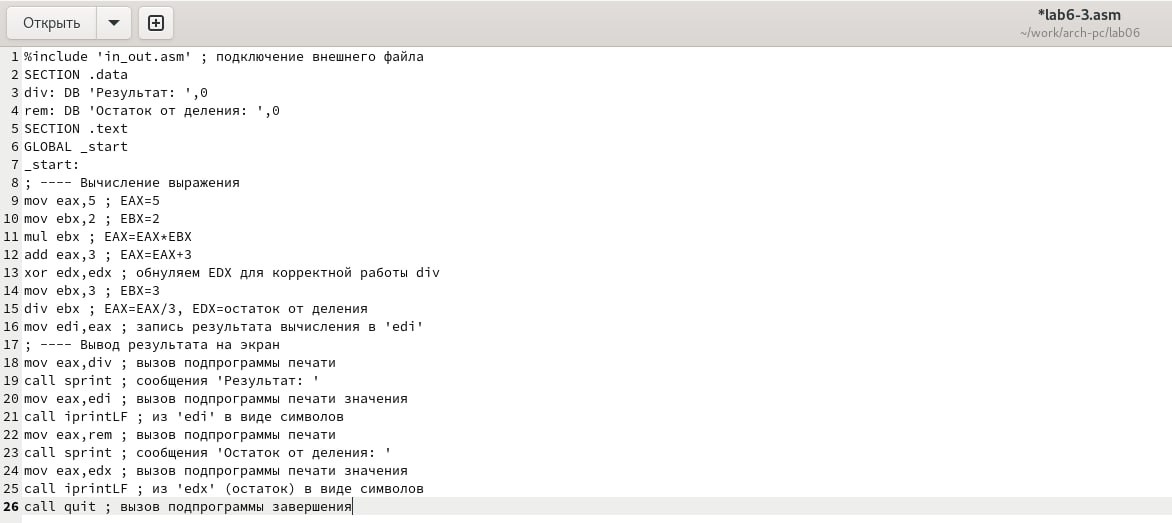


Рис. 16: Редактирование файла

Создаю исполняемый файл и запускаю его (рис. 17).

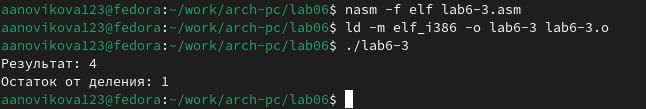


Рис. 17: Запуск исполняемого файла

Изменяю программу так, чтобы она вычисляла значение выражения f(x) = (4 \* 6 + 2)/5 (рис. 18).

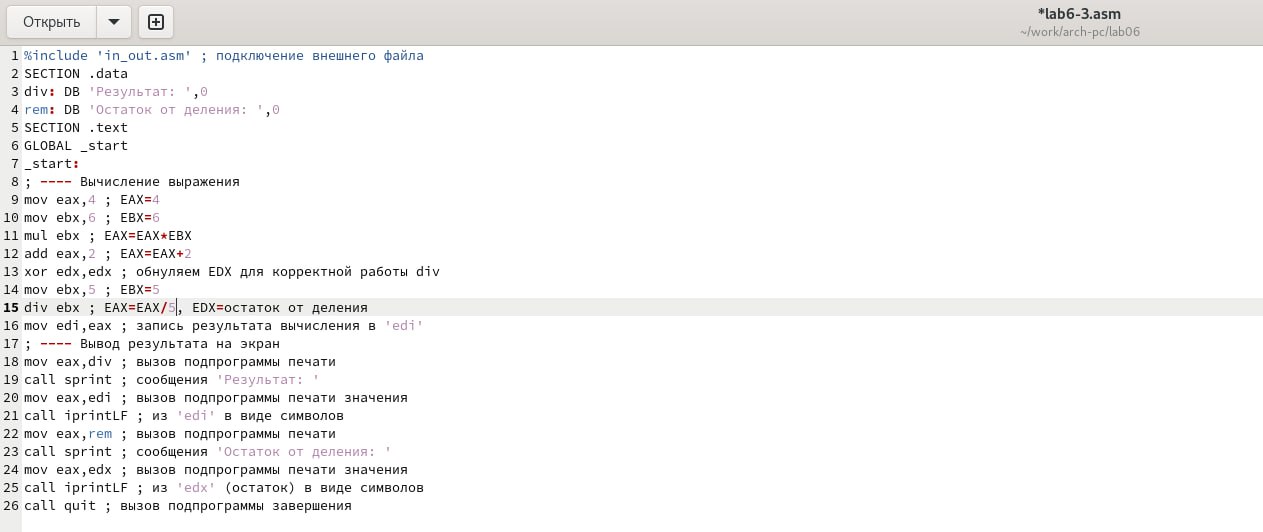


Рис. 18: Изменение программы

Создаю и запускаю новый исполняемый файл (рис. 19). Я посчитала для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно.

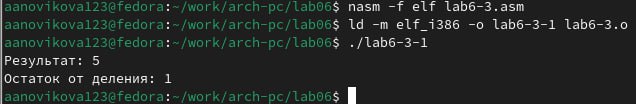


Рис. 19: Запуск исполняемого файла

Создаю файл variant.asm с помощью команды touch (рис. 20).

Создание файла

Рис. 20: Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 21).



Рис. 21: Редактирование файла

Создаю и запускаю исполняемый файл (рис. 22). Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 20.

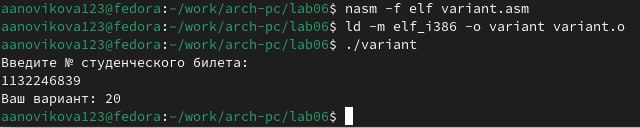


Рис. 22: Запуск исполняемого файла

Я проверила аналитически, действительно, 1132246839/20 + 1 = 20. Остаток от деления 1132246839 на 20 равен 19, 19 + 1 = 20.

### 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

Создаю файл lab6-zadanie.asm с помощью команды touch (рис. 23).

Создание файла

Рис. 23: Создание файла

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения x^3 \* 1/3 + 21 (рис. 24). Это выражение было под вариантом 20.

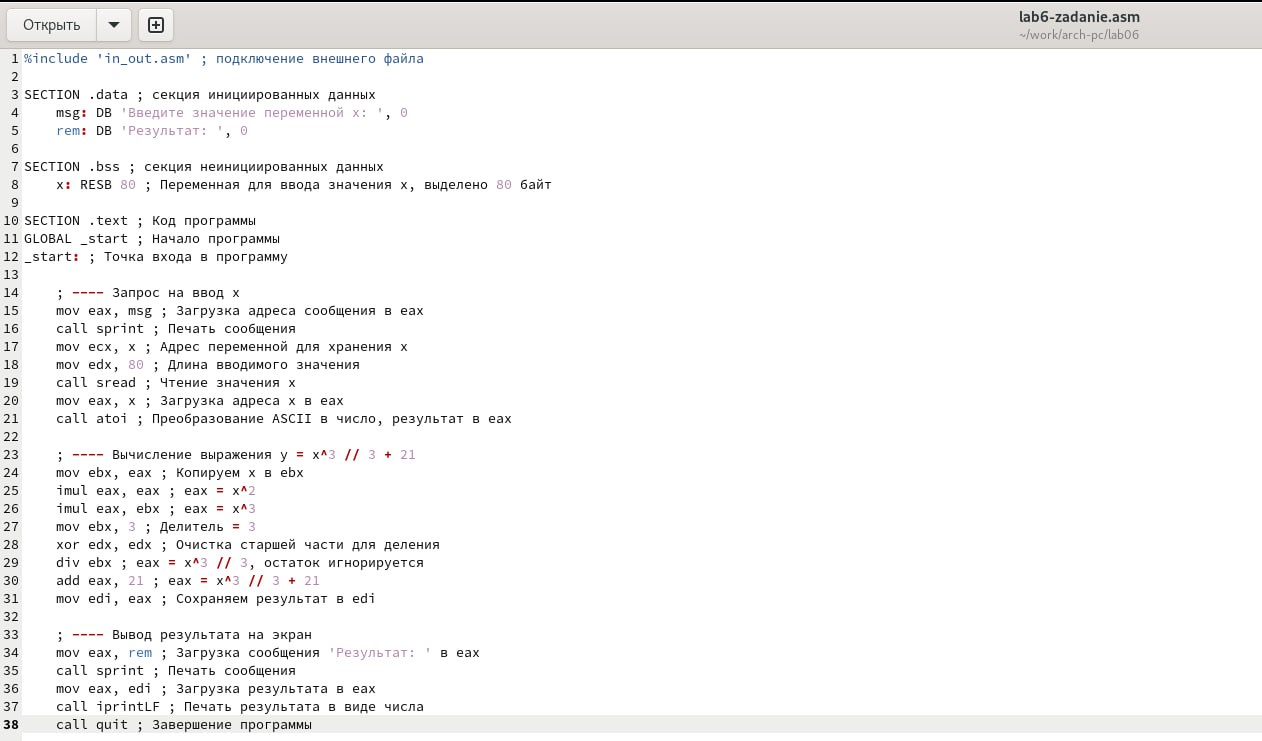


Рис. 24: Написание программы

Создаю и запускаю исполняемый файл (рис. 25). При вводе значения 1, вывод - 21. Так как 1^3 = 1. Остаток от деления на 3 равен 0. Значит, 0 + 21 = 21. Программа отработала верно.

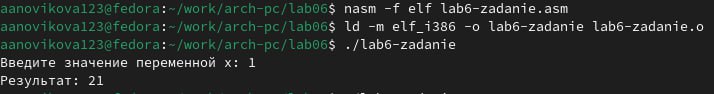


Рис. 25: Запуск исполняемого файла

Провожу еще один запуск исполняемого файла для проверки работы программы с другим значением на входе (рис. 26). Программа отработала верно. 3^3 = 27. 27/3 = 9. 9 + 21 = 30.

Запуск исполняемого файла

Рис. 26: Запуск исполняемого файла

**Листинг 4.1. Программа для вычисления значения выражения x^3 \* 1/3 + 21.**

%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 неинициированных данных  
 x: RESB 80 ; Переменная для ввода значения x, выделено 80 байт  
  
SECTION .text ; Код программы  
GLOBAL \_start ; Начало программы  
\_start: ; Точка входа в программу  
  
 ; ---- Запрос на ввод x  
 mov eax, msg ; Загрузка адреса сообщения в eax  
 call sprint ; Печать сообщения  
 mov ecx, x ; Адрес переменной для хранения x  
 mov edx, 80 ; Длина вводимого значения  
 call sread ; Чтение значения x  
 mov eax, x ; Загрузка адреса x в eax  
 call atoi ; Преобразование ASCII в число, результат в eax  
  
 ; ---- Вычисление выражения y = x^3 // 3 + 21  
 mov ebx, eax ; Копируем x в ebx  
 imul eax, eax ; eax = x^2  
 imul eax, ebx ; eax = x^3  
 mov ebx, 3 ; Делитель = 3  
 xor edx, edx ; Очистка старшей части для деления  
 div ebx ; eax = x^3 // 3, остаток игнорируется  
 add eax, 21 ; eax = x^3 // 3 + 21  
 mov edi, eax ; Сохраняем результат в edi  
  
 ; ---- Вывод результата на экран  
 mov eax, rem ; Загрузка сообщения 'Результат: ' в eax  
 call sprint ; Печать сообщения  
 mov eax, edi ; Загрузка результата в eax  
 call iprintLF ; Печать результата в виде числа  
 call quit ; Завершение программы

# 4 Выводы

При выполнении данной лабораторной работы были освоены арифметические инструкции языка ассемблера NASM.