Лабораторная работа №7

Щетинин Даниил Николаевич

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

Написать программу вычисления функции, ознакомиться с Арифметическими операциями используя язык ассемблера NASM

# 3 Выполнение лабораторной работы

**Шаг 1**

Создаём каталог для программам лабораторной работы No 7, и в нём создаём файл lab7-1.asm и переносим туда файл in\_out.asm, копируем в него текст листинга 7.1, для того, чтобы ознакомиться с принципом вывода переменных на экран. Создаём исполняемый файл и получаем результат, символ ‘j’, т.к. код полученного нами элемента является кодом не для числа 10, а для символа ‘j’

(рис. 1)

**Шаг 2**

Изменим текст нашего файла, заменив

mov eax,'6'  
mov ebx,'4

на

mov eax,6  
mov ebx,4

Получим 10 символ ASCII, который соответствует символу ‘LF,/n’, который не отображается на экране.

(рис. 2)

**Шаг 3**

Создадим следующий файл, lab7-2.asm, где мы используем код из файла in\_out.asm для преобразования ASCII символов в числа. Копируем код из листинга 7.2, запустим исполняемый файл, :

(рис. 3)

Получим число 106, сумму кодов символов ‘4’ и ‘6’

проделаем аналогичную работу с прошлым примером, заменив

mov eax,'6'  
mov ebx,'4

на

mov eax,6  
mov ebx,4

и, запустив файл, получим число 10, сумму 4 и 6.

(рис. 4)

Также заменим в исходном файле функцию iprintLF на iprint, и увидим, что из-за этого строка не переводится после результата.

(рис. 5)

**Шаг 4**

В новый файл lab7-3.asm введём текст из листинга 7.3, для нахождения выражения ((5∗2)+3)/3

Создаём файл и получаем ответ:

(рис. 6)

Изменим код файла для вычисления выражения ((4∗6)+2)/5

(рис. 7)

Результат:

(рис. 8)

**Шаг 5**

вводим в variant.asm текст программы, проверяем его работу:

(рис. 9)

Убедимся в правильности результата, разделим 1132226495 на 20 на калькуляторе, получим 56611324 целых, 15 в остатке, добавим к остатку 1, это равняется полученному номеру варианта

# 4 Ответы на вопросы 1-7

1. переменная rem отвечает за сообщение ‘Ваш вариант:’

rem: DB 'Ваш вариант: ',0

,

mov eax, msg  
  
call sprintLF

Отвечает за вывод сообщения ‘ваш вариант: [вариант]’

Данные функции используются для ввода значения переменной Х с клавиатуры и сохранения введенных данных

Данная инструкция используется для преобразования кода символа ASCII в число

mov ebx,20  
div ebx  
inc edx

В регистр edx

Для увеличения значения edx на 1.

Результат вычислений = номер варианта

mov eax,edx  
call iprintLF

# 5 Задание для самостоятельной работы

**Шаг 1**

Создадим файл var16.asm, в которой решим выражение (10\*x-5)^2, в зависимости от различных Х. За основу возьмём Variant.asm

Рассмотрим принцип работы нашего кода (Также см. Комментарии в самом коде)

mov eax, msg   
call sprintLF   
mov ecx, x   
mov edx, 80   
call sread   
mov eax,x  
call atoi

отвечает за запись переменной Х

Далее, умножим Х (eax) на 10 (ebx)

вычтем из Х (eax) 5

и умножим на Х (eax)

Запишем результат в edi, выведем на экран с помощью iprintLF

(рис. 10)

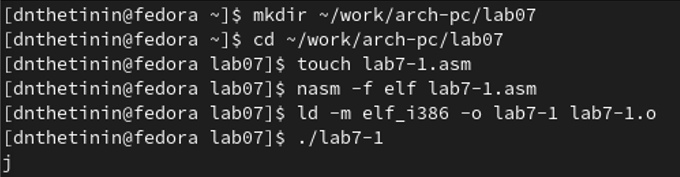


Рис. 1: lab7-1

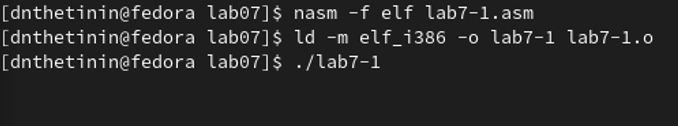


Рис. 2: lab7-1

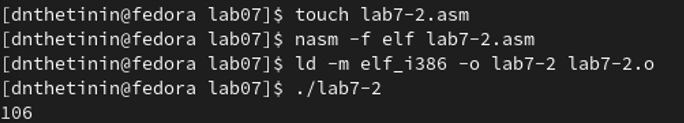


Рис. 3: lab7-2

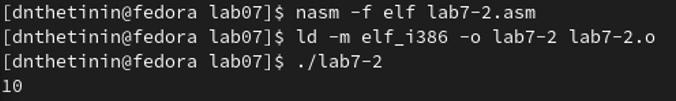


Рис. 4: lab7-2

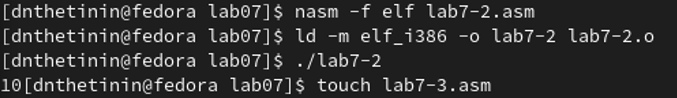


Рис. 5: lab7-2

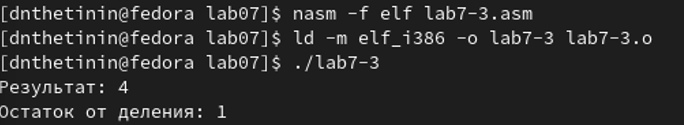


Рис. 6: работа файла lab7-3

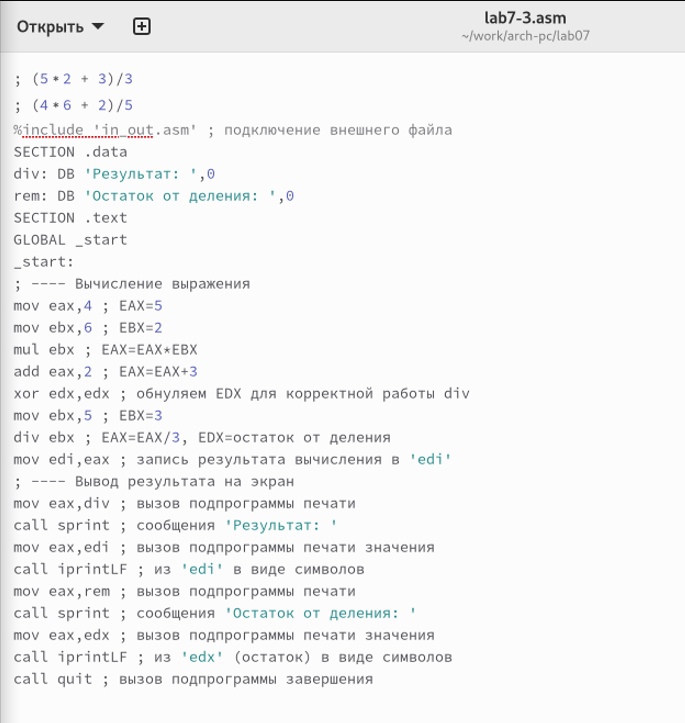


Рис. 7: Код новой lab7-3.asm

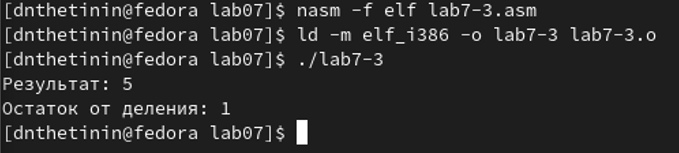


Рис. 8: Результат новой lab7-3

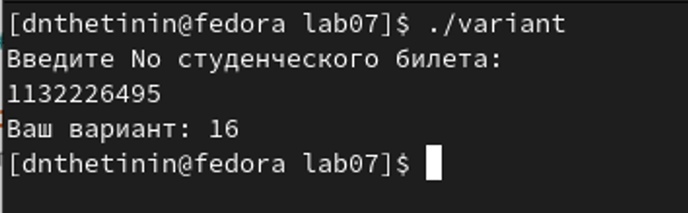


Рис. 9: Полученный вариант

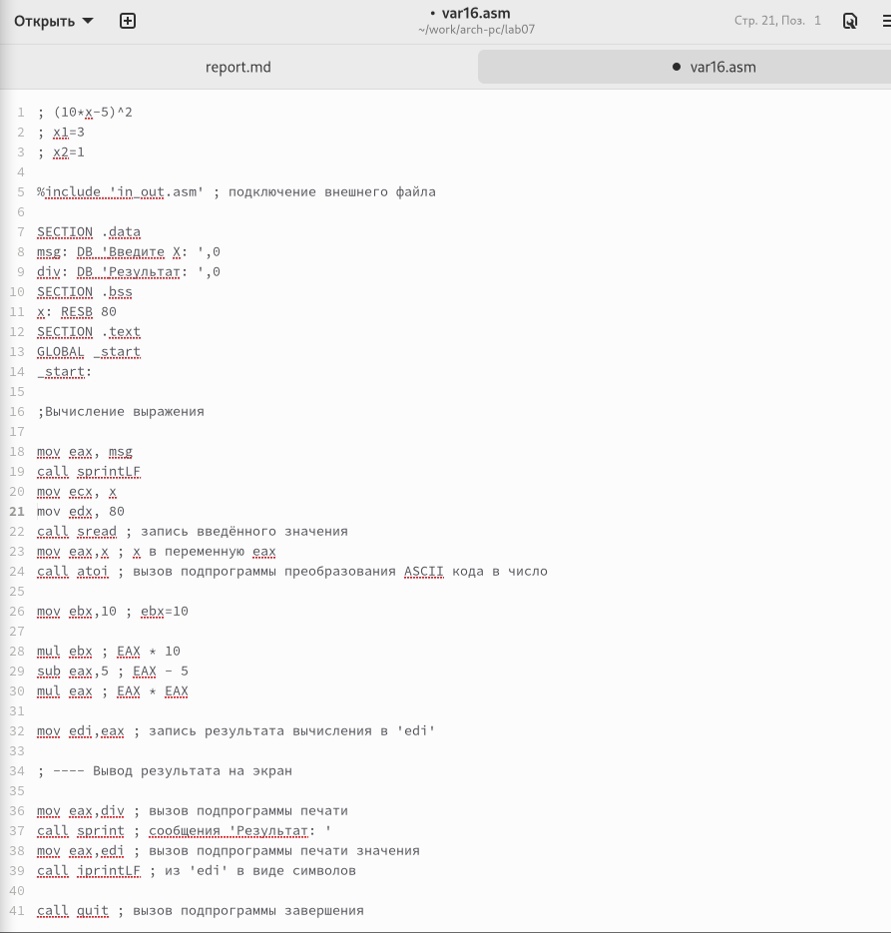


Рис. 10: код var16

# 6 Выводы

Я смог написать базовую программу вычисления функции на языке ассемблера NASM