Отчет по лабораторной работе №6

Дисциплина: архитектура компьютера

Гончарь Анастасия Александровна

Содержание

# 1 Цель работы

Целью данной лабораторной работы является освоение арифметческих 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 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 1 приведено краткое описание стандартных каталогов Unix.

Таблица 1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Более подробно про Unix см. в [1–4].

# 4 Выполнение лабораторной работы

## 4.1 Символьные и численные данные в NASM

Сначала с помощью mkdir создаю директорию, в которой буду создавать файлы с программами (рис. 1). Перехожу в созданный каталог с помощью утилиты cd.

Рис. 1: Создание директории

Рис. 1: Создание директории

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

Рис. 2: Создание файла

Рис. 2: Создание файла

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

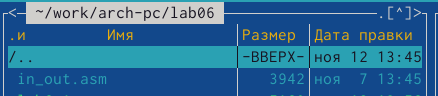


Рис. 3: Создание копии файла

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

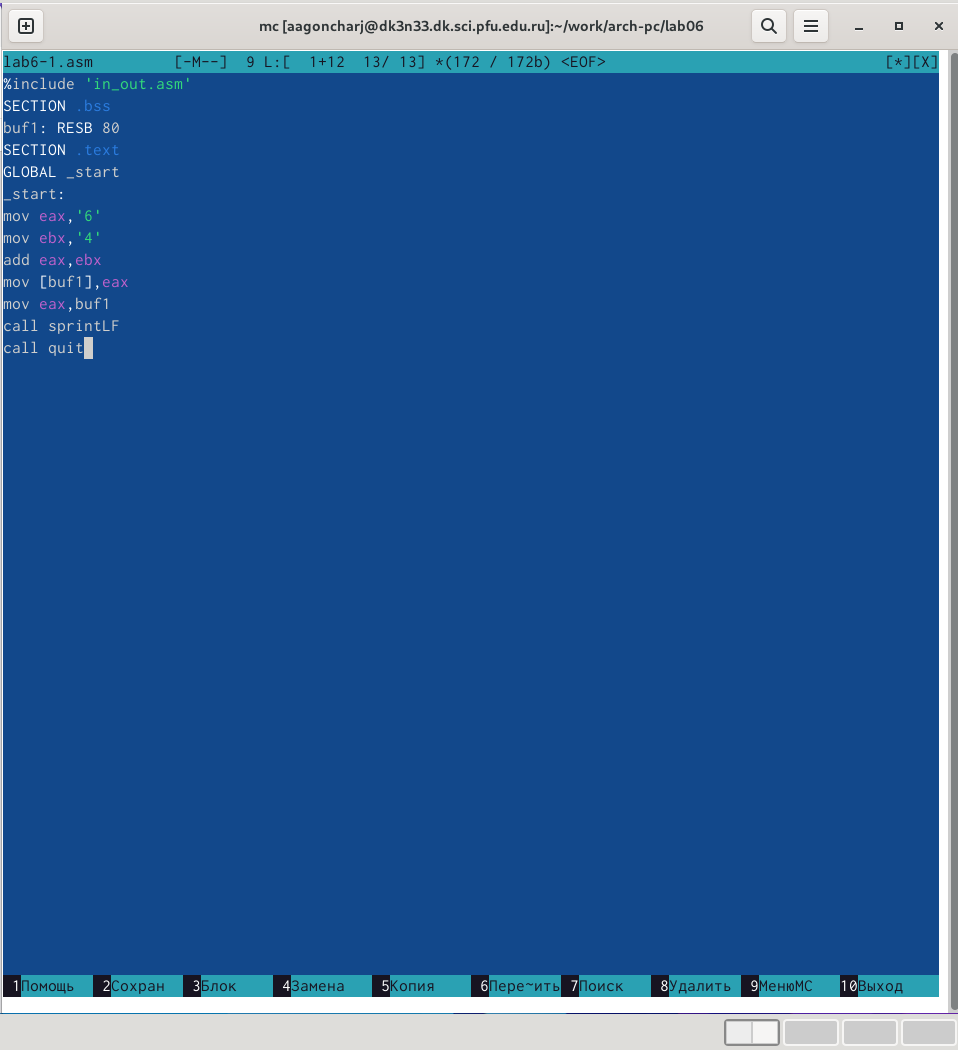


Рис. 4: Редактирование файла

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

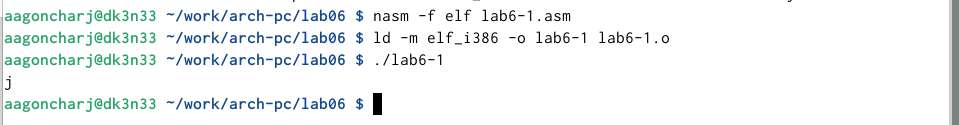


Рис. 5: Запуск исполняемого файла

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

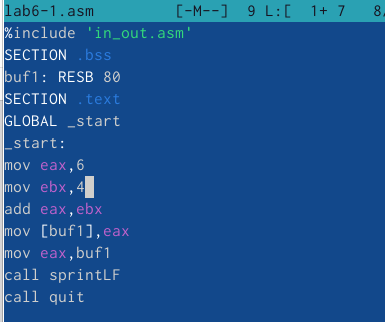


Рис. 6: Редактирование файла

Создаю новый исполняемый файл программы и запускаю его. (рис. 7)

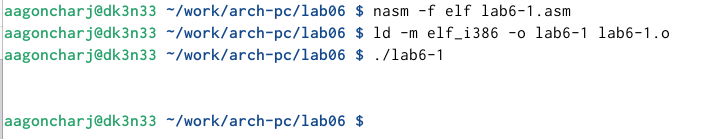


Рис. 7: Запуск исполняемого файла

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

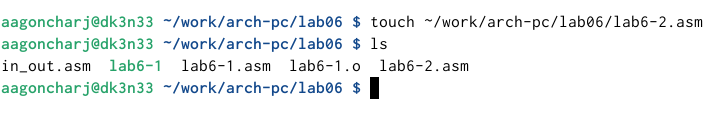


Рис. 8: Создание файла

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

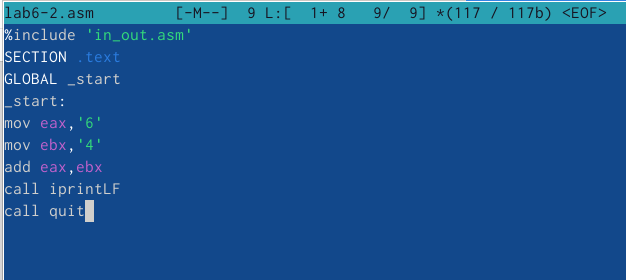


Рис. 9: Редактирование файла

Создаю и запускаю исполняемый файл lab6-2 (рис. 10). Теперь выводится число 106.

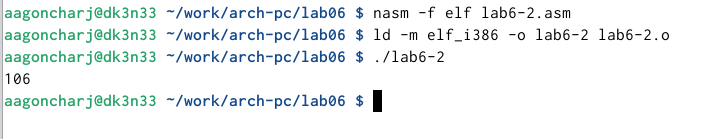


Рис. 10: Запуск исполняемого файла

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

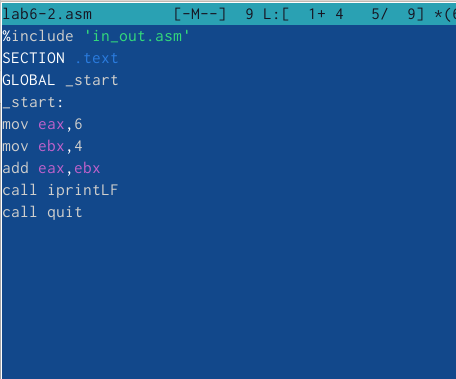


Рис. 11: Редактирование файла

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

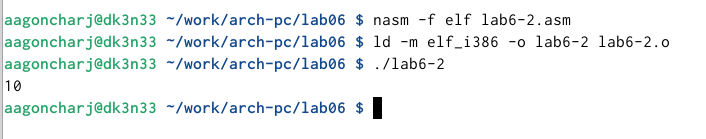


Рис. 12: Запуск исполняемого файла

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



Рис. 13: Редактирование файла

Создаю и запускаю новый исполняемый файл. 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. (рис. 14)

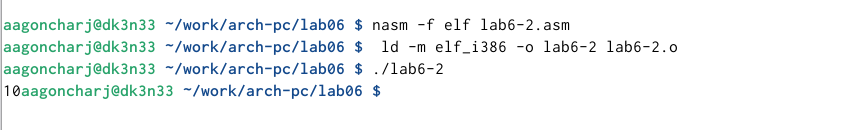


Рис. 14: Запуск исполняемого файла

## 4.2 Выполнение арифметических операций в NASM

Создаю файл lab7-3.asm с помощью утилиты touch (рис. 15).

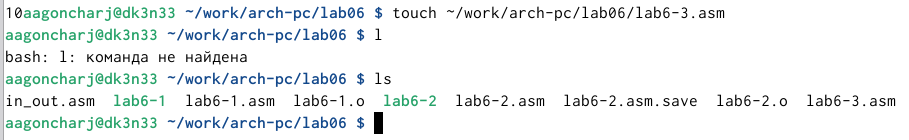


Рис. 15: Создание файла

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

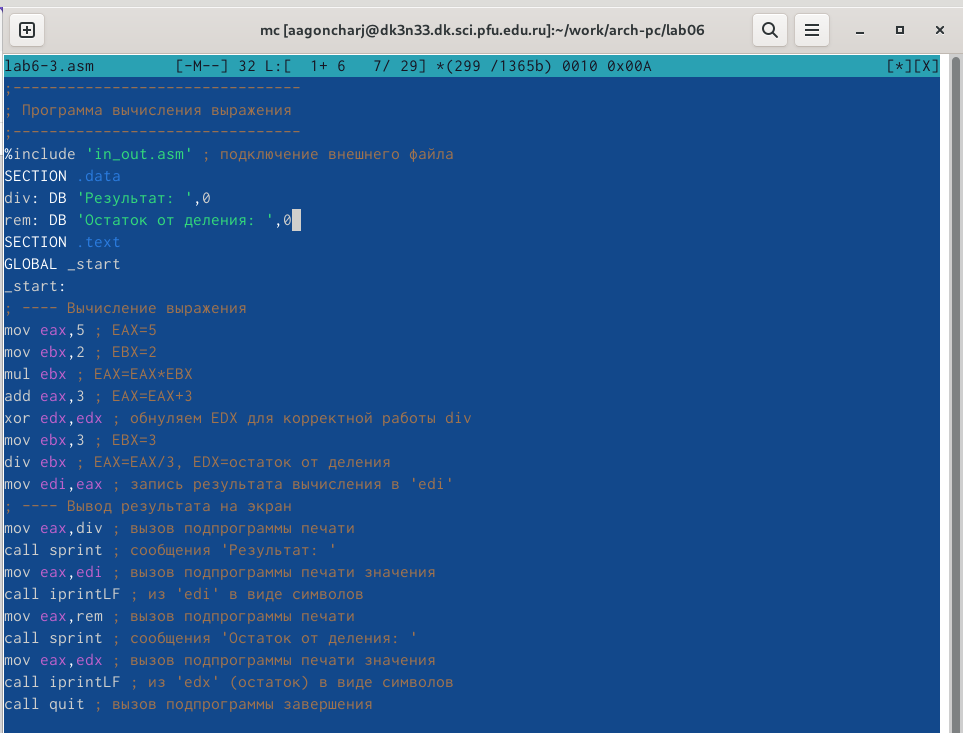


Рис. 16: Редактирование файла

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

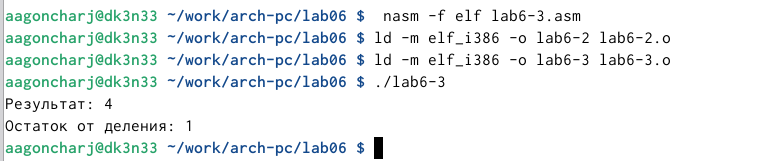


Рис. 17: Запуск исполняемого файла

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

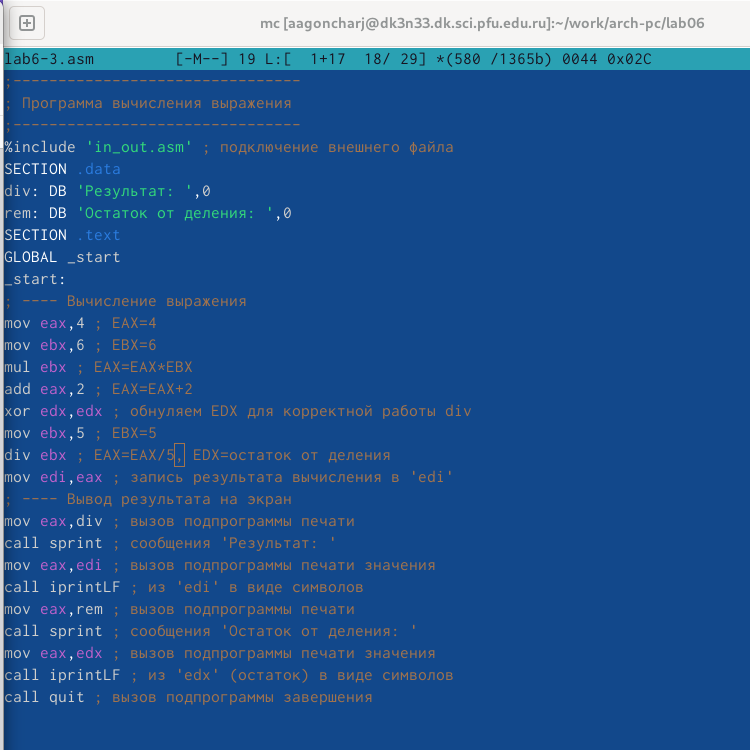


Рис. 18: Изменение программы

Создаю и запускаю новый исполняемый файл (рис. 19).

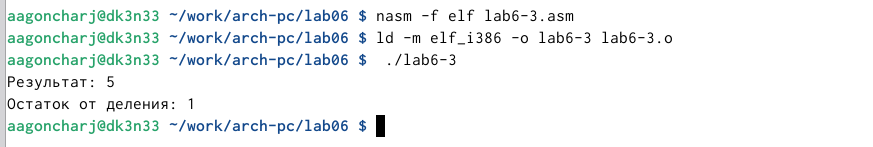


Рис. 19: Запуск исполняемого файла

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

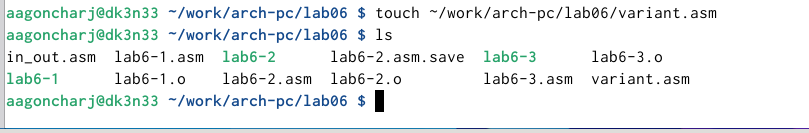


Рис. 20: Создание файла

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

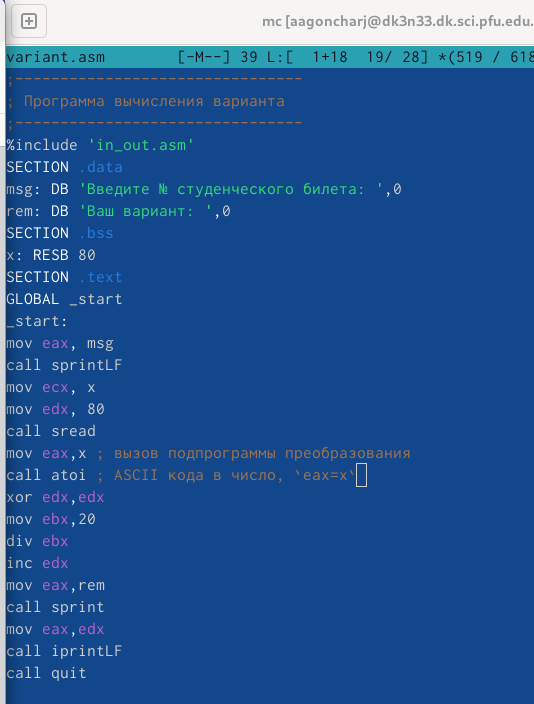


Рис. 21: Редактирование файла

Создаю и запускаю исполняемый файл. Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 15. (рис. 22)

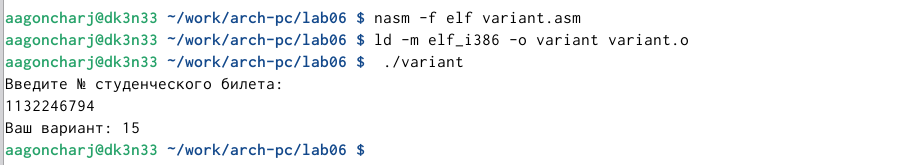


Рис. 22: Запуск исполняемого файла

### 4.2.1 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

## 4.3 Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm с помощью touch (рис. 23).

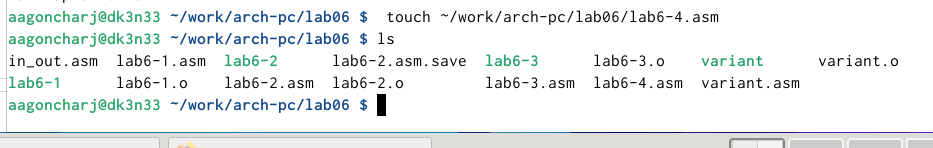


Рис. 23: Создание файла

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения (x+5)^2-3 из варианта 15 (рис. 24).

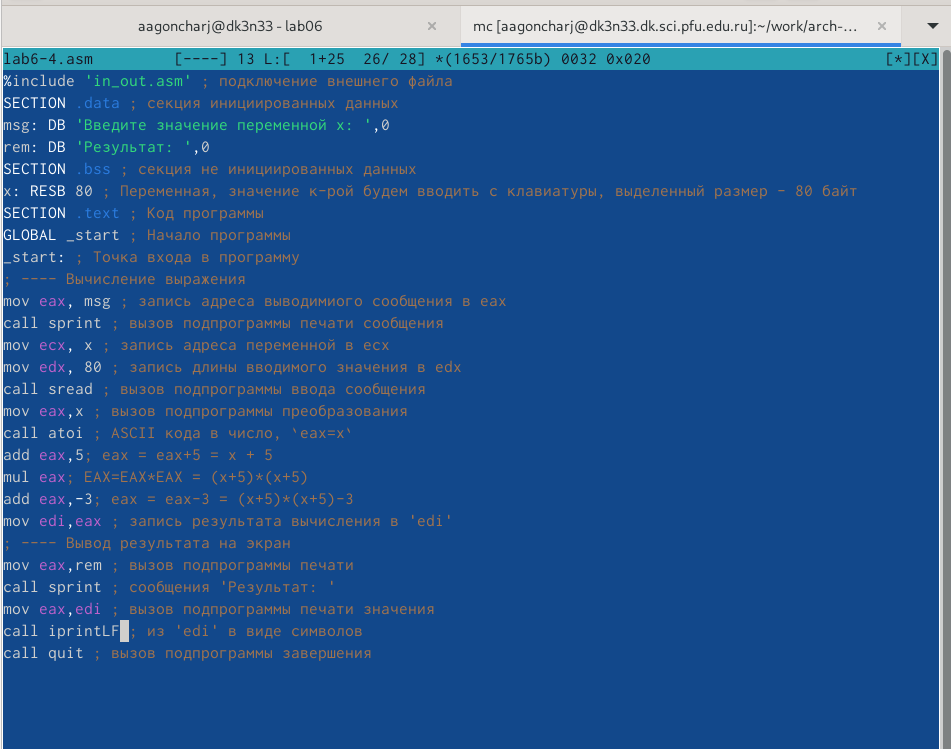


Рис. 24: Написание программы

Создаю и запускаю исполняемый файл. При вводе значения 1 выводится число 33, при вводе значения 5 - 97, что явлется верным. (рис. 25)

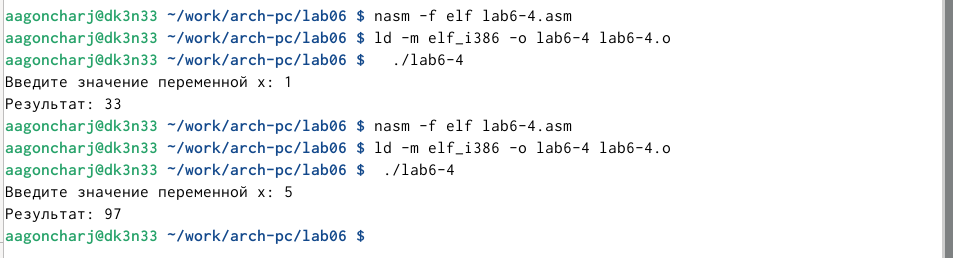


Рис. 25: Запуск исполняемого файла

### 4.3.1 Листинг программы для вычисления значения выражения (x+5)^2-3

%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 >  
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`  
add eax,5; eax = eax+5 = x + 5  
mul eax; EAX=EAX\*EAX = (x+5)\*(x+5)  
add eax,-3; eax = eax-3 = (x+5)\*(x+5)-3  
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

# 5 Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

# Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

2. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.