

Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Адмиральская Александра Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	22
	Список литературы	23

Список иллюстраций

4.1	Создание каталога lab06	8
4.2	Создание файла	8
4.3	Копирование файла	8
4.4	Программа в файле	9
4.5	Создание и запуск исполняемого файла	9
4.6	Изменение текста программы	10
4.7	Создание и запуск исполняемого файла	11
4.8	Создание файла	11
4.9	Ввод текста программы в файл lab6-2.asm	12
4.10	Создание и запуск исполняемого файла	12
4.11	Изменение текста программы	13
4.12	Запуск исполняемого файла	13
4.13	Изменение текста программы	14
4.14	Запуск измененного файла	14
4.15	Создание файла lab6-3.asm	15
4.16	Ввод текста программы в файл lab6-3.asm	15
4.17	Создание и запуск исполняемого файла	16
4.18	Изменение текста программы	16
4.19	Запуск исполняемого файла	17
4.20	Создание файла variant.asm	17
4.21	Ввод текста программы в файл variant.asm	18
4.22	Запуск исполняемого файла	19
4.23	Работа программы	19
4.24	Создание файла lab6-4.asm	20
4.25	Ввод текста программы в файл lab6-4.asm	20
4.26	Создание и запуск файла. Проверка его работы	21

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

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 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/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

Для начала создаем каталог для программ лабораторной работы № 6 и переходим в него (рис. 4.1).

```
aaadmiraljskaya@dk3n55 ~ $ mkdir ~/work/arch-pc/lab06
aaadmiraljskaya@dk3n55 ~ $ cd ~/work/arch-pc/lab06
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 4.1: Создание каталога lab06

Перейдя в каталог, создаем файл lab6-1.asm (рис. 4.2).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ touch lab6-1.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ls
lab6-1.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 4.2: Создание файла

Затем копируем в текущий каталог файл in_out.asm с помощью утилиты cp, т.к. он будет использоваться в других программах (рис. 4.3).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ cp ~/Загрузки/in_out.asm in_out.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ls
in_out.asm lab6-1.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 4.3: Копирование файла

Открываем созданный файл lab6-1.asm, вставляем в него программу вывода значения регистра eax (рис. 4.4).


```

lab6-1.asm [-M--
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintf
call quit

```

Рис. 4.4: Программа в файле

Создаем исполняемый файл и запускаем его. Программа выводит символ j (рис. 4.5).

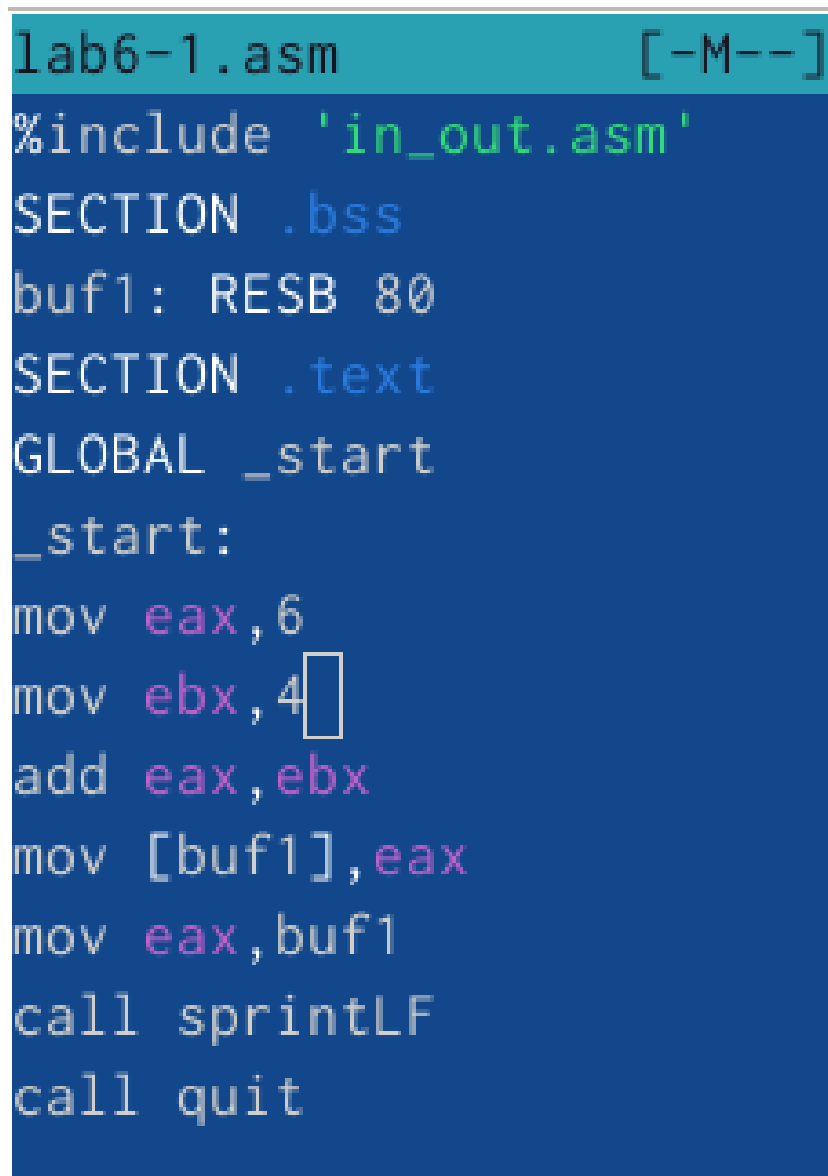
```

aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-1
j

```

Рис. 4.5: Создание и запуск исполняемого файла

Изменяем в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 4.6).



```
lab6-1.asm [-M--]
#include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov  eax,6
mov  ebx,4
add  eax,ebx
mov  [buf1],eax
mov  eax,buf1
call sprintf
call quit
```

Рис. 4.6: Изменение текста программы

Создаем новый исполняемый файл программы и запускаем его. Теперь программа выводит символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран (рис. 4.7).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-1

aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ █
```

Рис. 4.7: Создание и запуск исполняемого файла

Затем создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. 4.8).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-2.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ █
```

Рис. 4.8: Создание файла

Открываем созданный файл и вводим в него текст для вывода значения регистра eax (рис. 4.9).

```
lab6-2.asm [-M-
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 4.9: Ввод текста программы в файл lab6-2.asm

Создаем и запускаем исполняемый файл lab6-2. Программа выводит число 106 (рис. 4.10).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-2
bash: ./lab6-2: Нет такого файла или каталога
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-2
106
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 4.10: Создание и запуск исполняемого файла

Так же, как и в lab6-1.asm, заменяем в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 4.11).

```
lab6-2.asm [-M
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 4.11: Изменение текста программы

Создаем новый исполняемый файл программы и запускаем его. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10 (рис. 4.12).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-2
10
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 4.12: Запуск исполняемого файла

Далее открываем файл lab6-2.asm и меняем функцию iprintLF на iprint (рис. 4.13).

```
lab6-2.asm [-M
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис. 4.13: Изменение текста программы

Создаем и запускаем 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. 4.14).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-2
10aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 4.14: Запуск измененного файла

Следующим шагом создаем файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06` (рис. 4.15).

```

aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-3.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2 lab6-2.asm lab6-2.o lab6-3.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $

```

Рис. 4.15: Создание файла lab6-3.asm

Открываем созданный файл и вводим текст программы для вычисления значения выражения $f(x) = (5 * 2 + 3)/3$ (рис. 4.16).

```

lab6-3.asm      [-M--] 41 L:[ 1+28 29/ 29] *(1365/13
;-----
; Программа вычисления выражения
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

Рис. 4.16: Ввод текста программы в файл lab6-3.asm

Создаем исполняемый файл и запускаем его (рис. 4.17).

```

aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $

```

Рис. 4.17: Создание и запуск исполняемого файла

Затем в этом же файле изменяем текст программы для вычисления выражения $\square(\square) = (4 \square 6 + 2)/5$ (рис. 4.18).

```

lab6-3.asm      [BM--]  0 L:[ 1+11 12/ 29] *(383 ,
;-----
; Программа вычисления выражения
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=4
mov ebx,6 ; EBX=6
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

Рис. 4.18: Изменение текста программы

Создаем и запускаем исполняемый файл (рис. 4.19).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 4.19: Запуск исполняемого файла

Далее создаем файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. 4.20).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ls
in_out.asm  lab6-1.asm  lab6-2      lab6-2.o  lab6-3.asm  variant.asm
lab6-1      lab6-1.o    lab6-2.asm  lab6-3    lab6-3.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 4.20: Создание файла variant.asm

Вводим в этот файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 4.21).

```

variant.asm      [-M--]  9 L:[ 1+27 28/ 28]
;-----
; Программа вычисления варианта
;-----
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call iprintLF
call quit

```

Рис. 4.21: Ввод текста программы в файл variant.asm

Создаем и запускаем исполняемый файл. Программа просит ввести номер студенческого билета (рис. 4.22).

```

aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:

```

Рис. 4.22: Запуск исполняемого файла

Вводим номер своего студенческого билета с клавиатуры, программа вывела, что мой вариант - 7 (рис. 4.23).

```

aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132246806
Ваш вариант: 7

```

Рис. 4.23: Работа программы

Ответы на вопросы:

1) За вывод сообщения “Ваш вариант” отвечают строки кода:

```
mov eax,rem call sprint
```

2) Инструкция `mov esx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `esx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки, `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры

3) `Call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`

4) 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
```

5) При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`

6) Инструкция `inc edx` увеличивает значение регистра `edx` на 1

7) За вывод на экран результатов вычислений отвечают строки:

```
mov eax,edx call iprintLF
```

Приступим к выполнению заданий для самостоятельной работы. Создаем файл lab6-4.asm с помощью утилиты touch (рис. 4.24).

```
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ touch lab6-4.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ls
in_out.asm lab6-1.asm lab6-2 lab6-2.o lab6-3.asm lab6-4.asm variant.asm
lab6-1 lab6-1.o lab6-2.asm lab6-3 lab6-3.o variant variant.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 4.24: Создание файла lab6-4.asm

Открываем созданный файл клавишей F4. Вводим в него текст программы для вычисления значения выражения $5(x-1)^2$. Оно было под вариантом 7 (рис. 4.25).

```
lab6-4.asm [----] 41 L: [ 1+30 31/ 31] *(1281/1281b) <EOF>
;-----
; Программа вычисления выражения
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат ',0
SECTION .bss
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax, msg
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
add eax, -1; eax = eax-1 = x-1
mul eax; EAX=EAX*EAX = (x-1)*(x-1)
mov ebx, 5
mul ebx; EAX=EAX*EBX = 5*(x-1)*(x-1)
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
```

Рис. 4.25: Ввод текста программы в файл lab6-4.asm

Создаем и запускаем исполняемый файл. Проверяем его работу для значения

3 - на выводе 20 и для значения 5 - на выводе 80. Все верно (рис. 4.26).

```

aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x: 3
Результат 20
aaadmiraljskaya@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x: 5
Результат 80

```

Рис. 4.26: Создание и запуск файла. Проверка его работы

Текст программы для вычисления значения выражения $5(x-1)^2$:

```

;----- ; Программа вычисления выражения ;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data msg: DB
'Введите значение переменной x:',0 rem: DB 'Результат',0
SECTION .bss x: RESB 80 ; Переменная, значение которой будем вводить с клавиатуры
SECTION .text
GLOBAL _start _start: ; --- Вычисление выражения
mov eax, msg call sprint mov
ecx, x mov edx, 80 call sread mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x' add eax,-1; eax = eax-1 = x-1 mul eax; EAX=EAXEAX
= (x-1)(x-1) mov ebx,5 mul ebx; EAX=EAXEBX = 5(x-1)*(x-1) 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. O'Reilly Media, 2005. 354 с.