

Шаблон отчёта по лабораторной работе

6

Разанацуа Сара Естэлл

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Ответы на вопросы по программе	15
4	Выводы	17
	Список литературы	18

Список иллюстраций

3.1	создание директории	7
3.2	создание файла	7
3.3	создание копии файла	7
3.4	редактирование файла	8
3.5	запуск исполняемого файла	8
3.6	редактирование файла	9
3.7	запуск исполняемого файла	9
3.8	создание копии файла	9
3.9	редактирование файла	10
3.10	запуск исполняемого файла	10
3.11	редактирование файла	10
3.12	запуск исполняемого файла	11
3.13	запуск исполняемого файла	11
3.14	редактирование файла	12
3.15	запуск исполняемого файла	12
3.16	изменение программы	13
3.17	запуск исполняемого файла	13
3.18	создание копии файла	13
3.19	редактирование файла	14
3.20	запуск исполняемого файла	14
3.21	запуск исполняемого файла	16

Список таблиц

1 Цель работы

- Цель данной лабораторной 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

- 1) Символьные и численные данные в NASM
- С помощью утилиты `mkdir` создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №6 (рис. [3.1]). Перехожу в созданный каталог с помощью утилиты `cd`.(рис. 3.1).

```
serazanacua@dk8n52 ~ $ mkdir ~/work/arch-pc/lab06
serazanacua@dk8n52 ~ $ cd ~/work/arch-pc/lab06
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ touch lab6-1.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $
```

Рис. 3.1: создание директории

- С помощью утилиты `touch` создаю файл `lab6-1.asm`.(рис. 3.2).

```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ touch lab6-1.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ls
lab6-1.asm
```

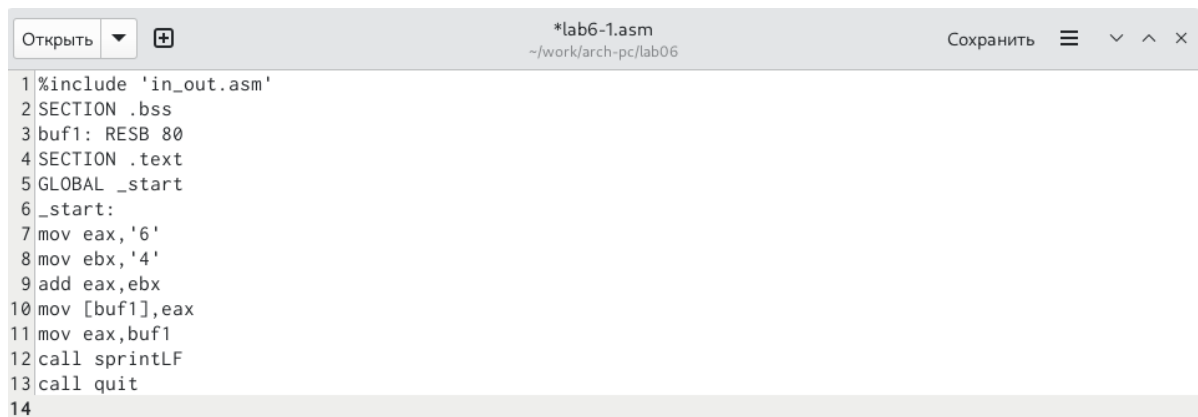
Рис. 3.2: создание файла

- Копирую в текущий каталог файл `in_out.asm` с помощью утилиты `cp`, т.к. он будет использоваться в других программах (рис. 3.3).

```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ cp ~/Зарпюзки/in_out.asm in_out.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ls
in_out.asm  lab6-1.asm
```

Рис. 3.3: создание копии файла

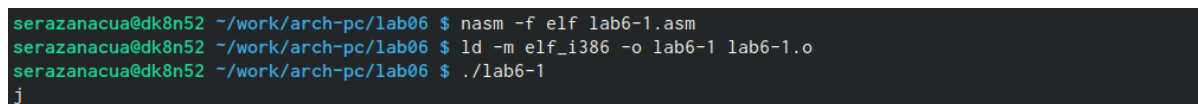
- Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax (рис. 3.4).



```
Открыть ▼ + *lab6-1.asm ~/work/arch-pc/lab06 Сохранить ≡ ▾ ^ ×
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax, '6'
8 mov ebx, '4'
9 add eax, ebx
10 mov [buf1], eax
11 mov eax, buf1
12 call sprintf
13 call quit
14
```

Рис. 3.4: редактирование файла

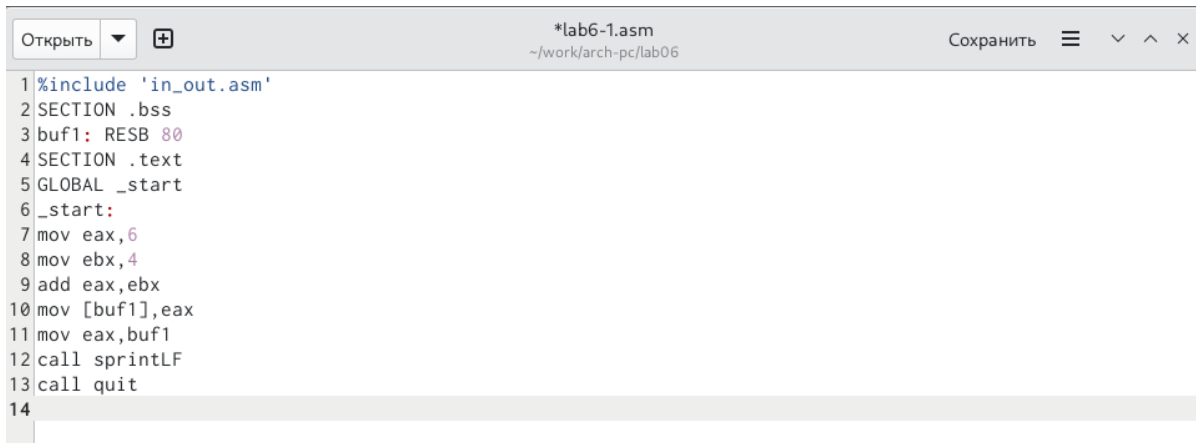
- Создаю исполняемый файл программы и запускаю его (рис. [3.5]). Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6. (рис. 3.5).



```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 3.5: запуск исполняемого файла

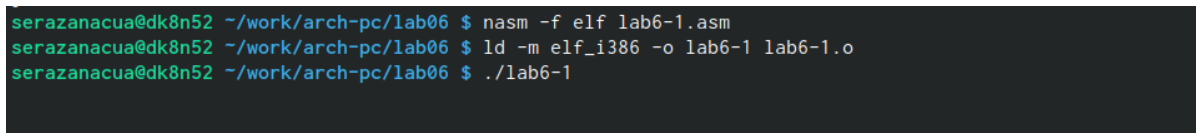
- Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 3.6).



```
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax, 6
8 mov ebx, 4
9 add eax, ebx
10 mov [buf1], eax
11 mov eax, buf1
12 call sprintf
13 call quit
14
```

Рис. 3.6: редактирование файла

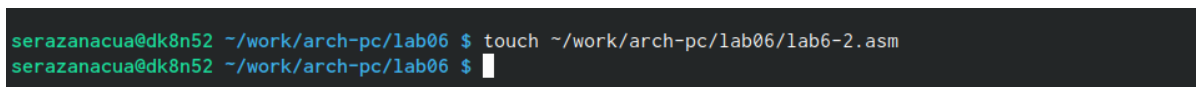
- Создаю новый исполняемый файл программы и запускаю его (рис. [3.7]). Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.(рис. 3.7).



```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./lab6-1
```

Рис. 3.7: запуск исполняемого файла

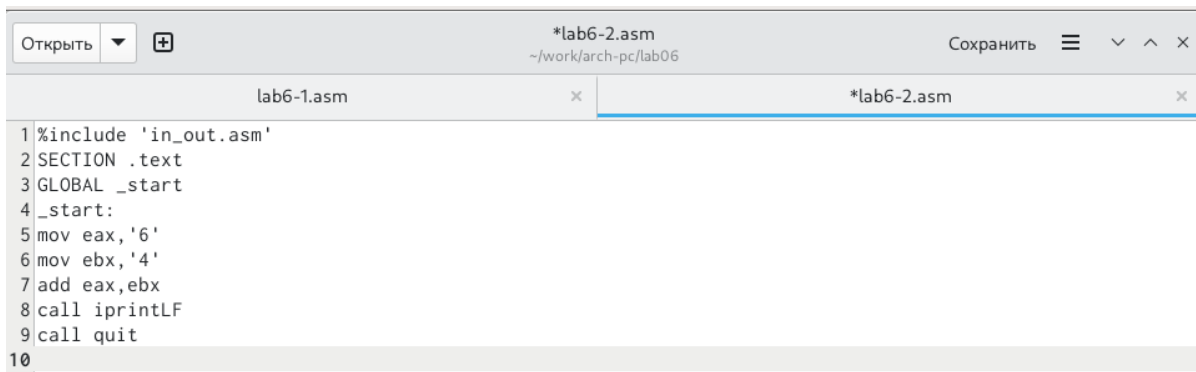
- Создаю новый файл lab6-2.asm с помощью утилиты touch (рис. 3.8).



```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-2.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $
```

Рис. 3.8: создание копии файла

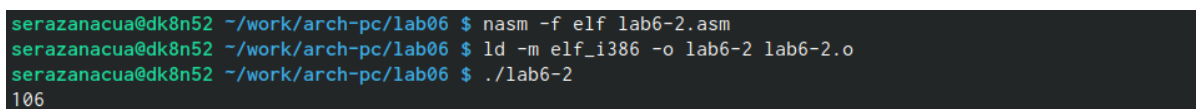
- Ввожу в файл текст другой программы для вывода значения регистра eax (рис. 3.9).



```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax, '6'
6 mov ebx, '4'
7 add eax, ebx
8 call iprintLF
9 call quit
10
```

Рис. 3.9: редактирование файла

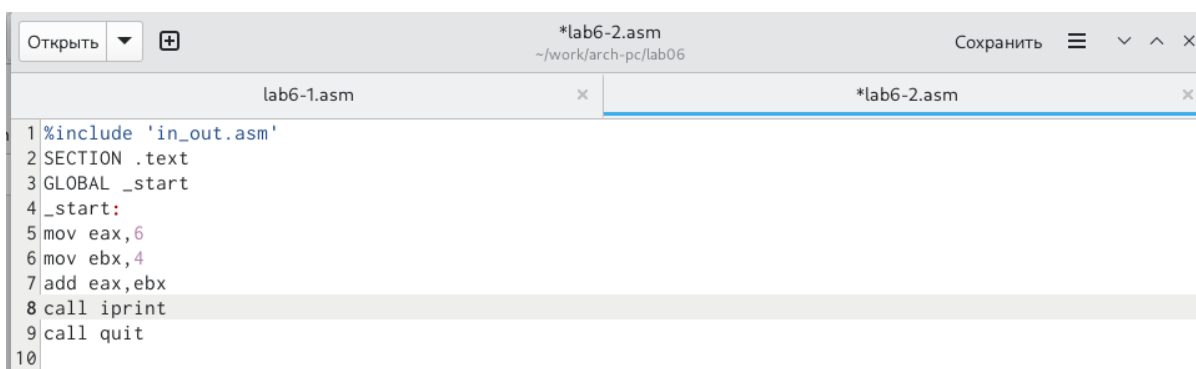
- Создаю и запускаю исполняемый файл lab6-2 (рис. [3.10]). Теперь вывод число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”. (рис. 3.10).



```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./lab6-2
106
```

Рис. 3.10: запуск исполняемого файла

- Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 3.11).



```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax, 6
6 mov ebx, 4
7 add eax, ebx
8 call iprint
9 call quit
10
```

Рис. 3.11: редактирование файла

-Создаю и запускаю новый исполняемый файл (рис. [3.12]).. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10. (рис. 3.12).

```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./lab6-2
10
```

Рис. 3.12: запуск исполняемого файла

- Заменяю в тексте программы функцию `iprintLF` на `iprint`
- Создаю и запускаю новый исполняемый файл (рис. [3.13]). 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. 3.13).

```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./lab6-2
10serazanacua@dk8n52 ~/work/arch-pc/lab06 $
```

Рис. 3.13: запуск исполняемого файла

- 2) Выполнение арифметических операций в NASM
- Создаю файл `lab6-3.asm` с помощью утилиты `touch` . Ввожу в созданный файл текст программы для вычисления значения выражения $f(x) = (5 * 2 + 3)/3$ (рис. 3.14).

```
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax,5
9 mov ebx,2
10 mul ebx
11 add eax,3
12 xor edx,edx
13 mov ebx,3
14 div ebx
15 mov edi,eax
16 mov eax,div
17 call sprint
18 mov eax,edi
19 call iprintLF
20 mov eax,rem
21 call sprint
22 mov eax,edx
23 call iprintLF
24 call quit
```

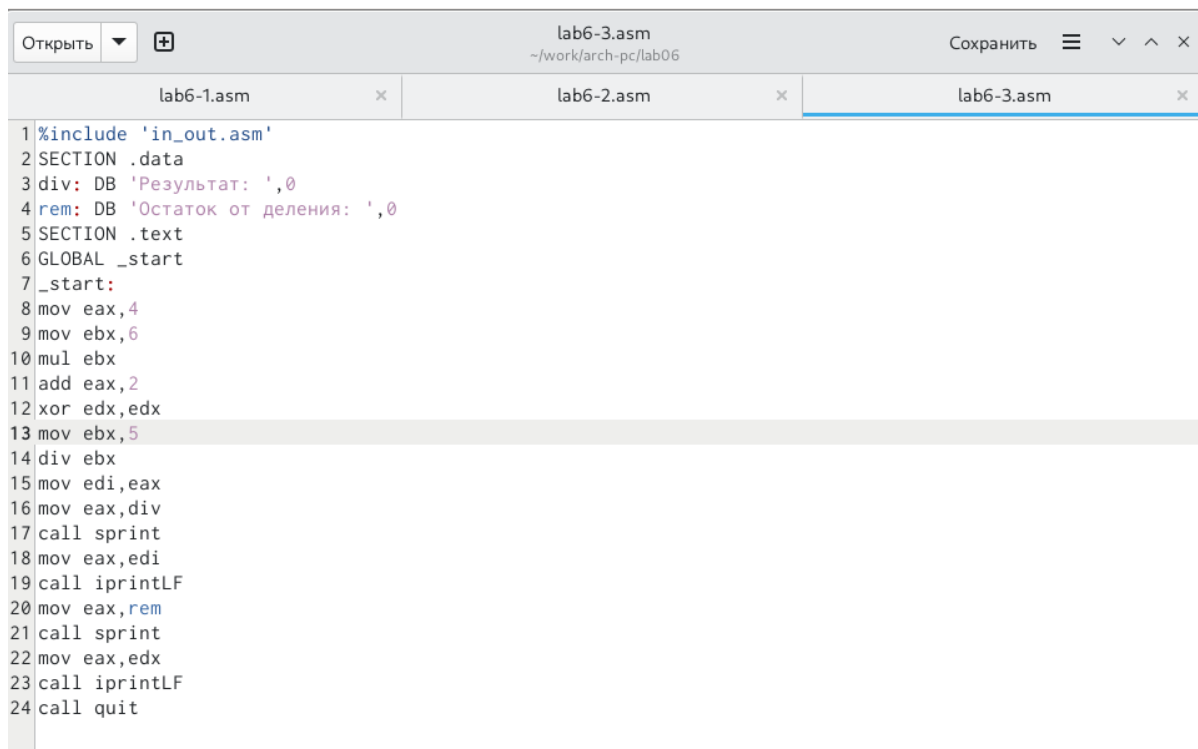
Рис. 3.14: редактирование файла

- Создаю исполняемый файл и запускаю его (рис. 3.15).

```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 3.15: запуск исполняемого файла

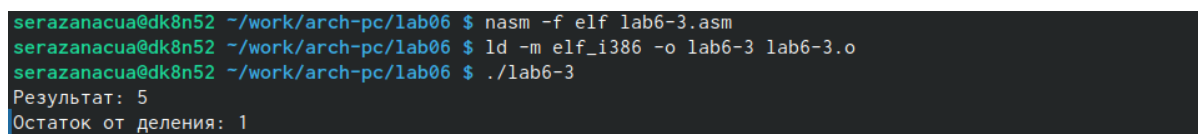
- Изменяю программу так, чтобы она вычисляла значение выражения $f(x) = (4 * 6 + 2)/5$ (рис. 3.16).



```
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax,4
9 mov ebx,6
10 mul ebx
11 add eax,2
12 xor edx,edx
13 mov ebx,5
14 div ebx
15 mov edi,eax
16 mov eax,div
17 call sprint
18 mov eax,edi
19 call iprintLF
20 mov eax,rem
21 call sprint
22 mov eax,edx
23 call iprintLF
24 call quit
```

Рис. 3.16: изменение программы

- Создаю и запускаю новый исполняемый файл (рис. [3.17]). Я посчитала для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно. (рис. 3.17).



```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 3.17: запуск исполняемого файла

- Создаю файл variant.asm с помощью утилиты touch. (рис. 3.18).



```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
serazanacua@dk8n52 ~/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  lab6-3.asm  lab6-3.o  variant.asm
```

Рис. 3.18: создание копии файла

- Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета.(рис. 3.19).

```

1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите No студенческого билета: ',0
4 rem: DB 'Ваш вариант: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintf
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax, x
16 call atoi
17 xor edx, edx
18 mov ebx, 20
19 div ebx
20 inc edx
21 mov eax, rem
22 call sprintf
23 mov eax, edx
24 call iprintLF
25 call quit

```

Рис. 3.19: редактирование файла

- Создаю и запускаю исполняемый файл (рис. [3.20]). Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 15. рис. 3.20).

```

serazanacua@dk8n52 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./variant
Введите No студенческого билета:
1032225834
Ваш вариант: 15

```

Рис. 3.20: запуск исполняемого файла

3.1 Ответы на вопросы по программе

- 1) За вывод сообщения “Ваш вариант” отвечают строки кода:
 - `mov eax,rem`
 - `call sprint`
- 2) 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
- 3) `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`
- 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`

- 3) Выполнение заданий для самостоятельной работы
- Создаю и запускаю исполняемый файл. рис. 3.21).

```
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x:
Введите значение переменной x:
3
Результат: 625
serazanacua@dk8n52 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x:
Введите значение переменной x:
1
Результат: 25
serazanacua@dk8n52 ~/work/arch-pc/lab06 $
```

Рис. 3.21: запуск исполняемого файла

4 Выводы

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

Список литературы