

Отчёта по лабораторной работе №8

**Команды безусловного и условного переходов в Nasm.
Программирование ветвлений.**

Мокочунина Влада Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Самостоятельная работа	13
5	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	Создание каталога и файлов	7
3.2	Ввод текста	7
3.3	Запуск файла	8
3.4	Замена текста	8
3.5	Запуск файла	9
3.6	Изменение текста	9
3.7	Запуск файла	9
3.8	Ввод текста	10
3.9	Запуск файла	10
3.10	Создание файла	10
3.11	Файл листинга	11
3.12	Удаление операнда	12
3.13	Трансляция файла	12
3.14	Ошибка	12
4.1	Написание программы	13
4.2	Запуск программы	13
4.3	Написание программы	14
4.4	Запуск программы	14

Список таблиц

1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Задание

Изучить команды условного и безусловного переходов, приобрести навыки написания программ с использованием переходов и познакомиться с назначением и структурой файла листинга.

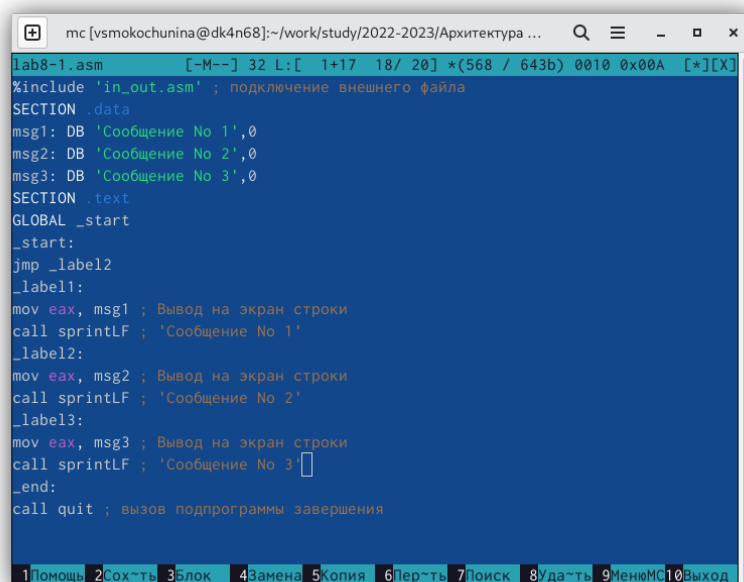
3 Выполнение лабораторной работы

1. Я создала каталог для программ лабораторной работы No 8, перешла в него и создала файл lab8-1.asm (рис. 3.1)

```
vsmokochunina@dk4n68 ~ $ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab08
vsmokochunina@dk4n68 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab08
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ touch lab8-1.asm
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$
```

Рис. 3.1: Создание каталога и файлов

2. Я ввела в файл текст из листинга 1



```
lab8-1.asm [-M--] 32 L: [ 1+17 18/ 20] *(568 / 643b) 0010 0x00A [*][X]
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение No 1'
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение No 2'
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение No 3'
_end:
call quit ; вызов подпрограммы завершения
```

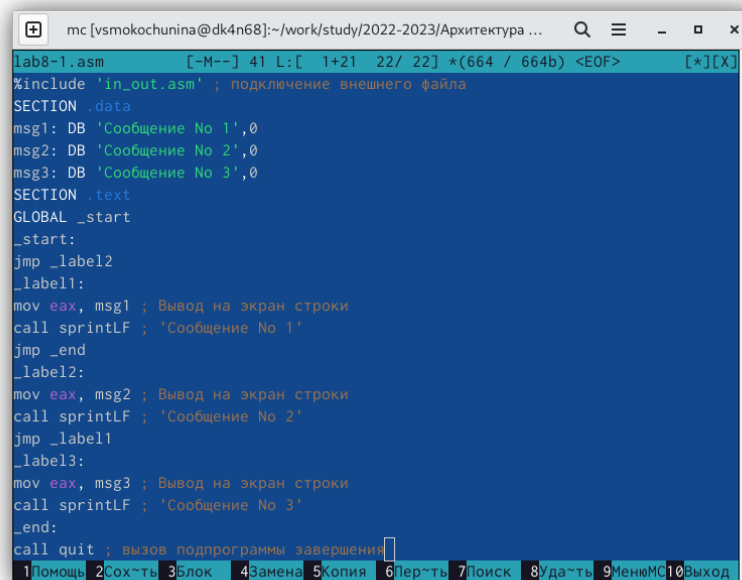
Рис. 3.2: Ввод текста

3. Я создала файл и запустила его

```
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ nasm -f elf lab8-1.asm
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ ld -m elf_i386 -o lab8-1 lab8-1.o
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ ./lab8-1
Сообщение No 2
Сообщение No 3
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$
```

Рис. 3.3: Запуск файла

4. Я изменила текст программы в соответствии с листингом 2



```
lab8-1.asm [-M--] 41 L:[ 1+21 22/ 22] *(664 / 664b) <EOF> [*][X]
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение No 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение No 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение No 3'
_end:
call quit ; вызов подпрограммы завершения
```

Рис. 3.4: Замена текста

5. Я создала файл и проверила его работу

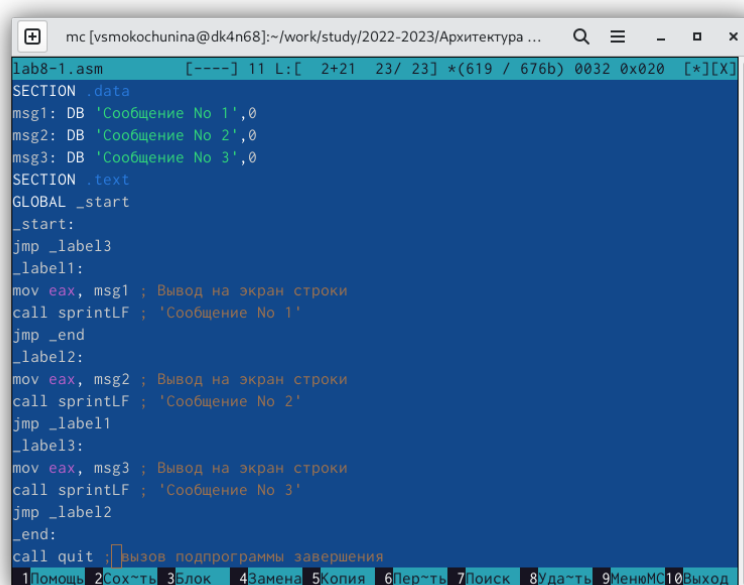

```

vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ nasm -f elf lab8-1.asm
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ ld -m elf_i386 -o lab8-1 lab8-1.o
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ ./lab8-1
Сообщение No 2
Сообщение No 1
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ 

```

Рис. 3.5: Запуск файла

6. Я изменила текст программы, чтобы она работала требуемым образом



```

lab8-1.asm [----] 11 L: [ 2+21 23/ 23] *(619 / 676b) 0032 0x020 [*][X]
SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label3
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение No 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение No 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение No 3'
jmp _label2
_end:
call quit ; вызов подпрограммы завершения
1 Помощь 2 Сох-ть 3 Блок 4 Замена 5 Копия 6 Пер-ть 7 Поиск 8 Уда-ть 9 МенюМС 10 Выход

```

Рис. 3.6: Изменение текста

7. Я создала и запустила файл

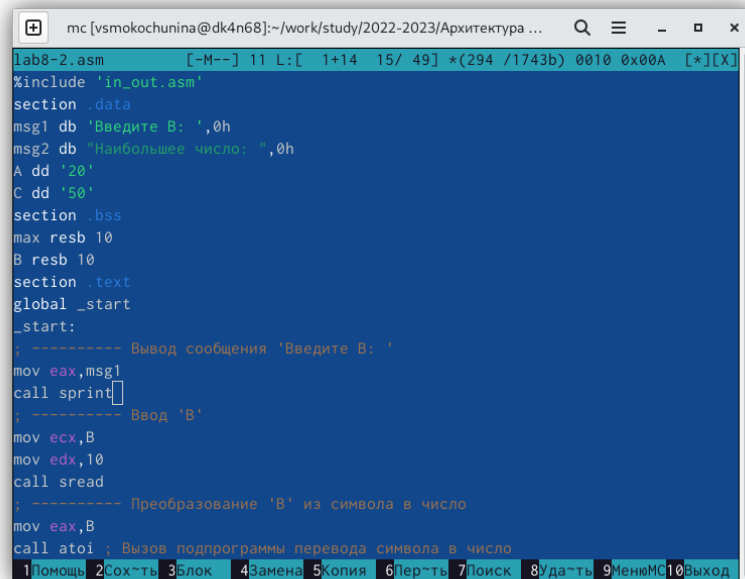
```

vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ nasm -f elf lab8-1.asm
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ ld -m elf_i386 -o lab8-1 lab8-1.o
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ ./lab8-1
Сообщение No 3
Сообщение No 2
Сообщение No 1
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ 

```

Рис. 3.7: Запуск файла

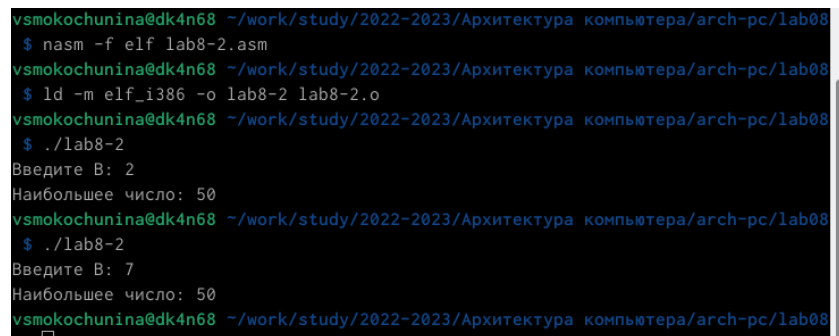
8. Я создала файл и ввела в него программу из листинга 3



```
lab8-2.asm [-M--] 11 L: [ 1+14 15/ 49] *(294 /1743b) 0010 0x00A [*][X]
#include 'in_out.asm'
section .data
msg1 db 'Введите B: ',0h
msg2 db "Наибольшее число: ",0h
A dd '20'
C dd '50'
section .bss
max resb 10
B resb 10
section .text
global _start
_start:
; ----- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ----- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
1Помощь 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Вда-ть 9МенюМС 0Выход
```

Рис. 3.8: Ввод текста

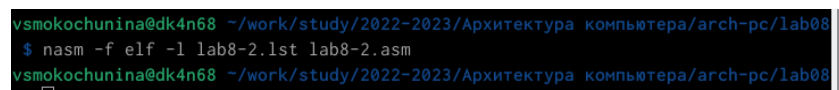
9. Я создала файл и проверила его работу



```
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ nasm -f elf lab8-2.asm
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ ld -m elf_i386 -o lab8-2 lab8-2.o
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ ./lab8-2
Введите B: 2
Наибольшее число: 50
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ ./lab8-2
Введите B: 7
Наибольшее число: 50
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
```

Рис. 3.9: Запуск файла

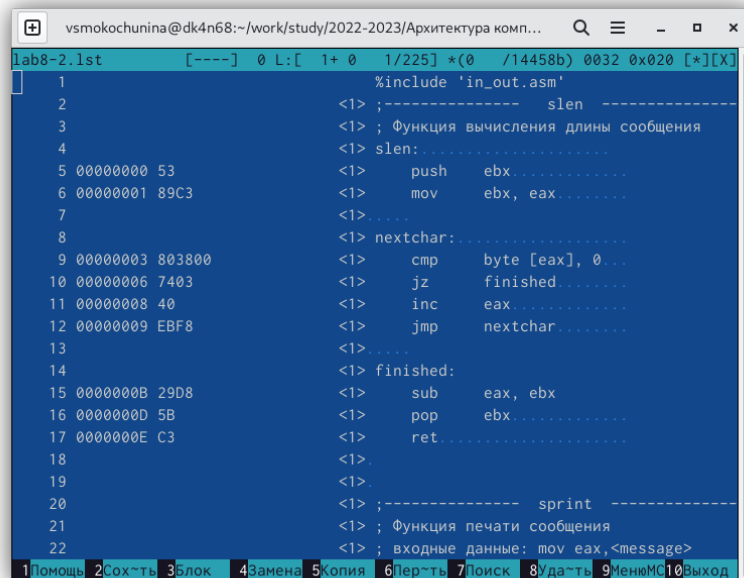
10. Я создала файл листинга для программы из файла lab8-2.asm



```
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ nasm -f elf -l lab8-2.lst lab8-2.asm
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
```

Рис. 3.10: Создание файла

11. Я открыла файл листинга, ознакомилась с форматом и содержимым

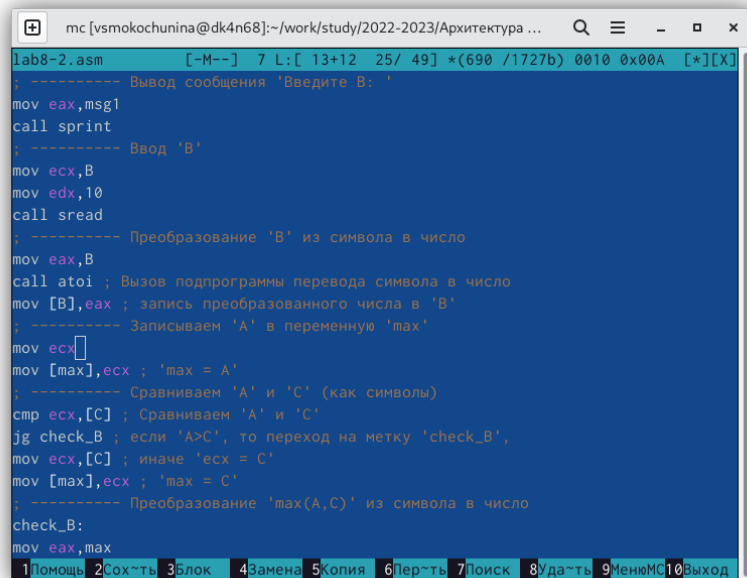


```
lab8-2.lst [----] 0 L: [ 1+ 0 1/225] *(0 /14458b) 0032 0x020 [*][X]
1      %include 'in_out.asm'
2      <1> ;----- slen -----
3      <1> ; Функция вычисления длины сообщения
4      <1> slen:
5      00000000 53      <1> push    ebx
6      00000001 89C3    <1> mov     ebx, eax
7      <1>
8      <1> nextchar:
9      00000003 803800    <1> cmp     byte [eax], 0
10     00000006 7403     <1> jz      finished
11     00000008 40      <1> inc     eax
12     00000009 EBF8     <1> jmp     nextchar
13     <1>
14     <1> finished:
15     0000000B 29D8     <1> sub     eax, ebx
16     0000000D 5B      <1> pop     ebx
17     0000000E C3      <1> ret
18     <1>
19     <1>
20     <1> ;----- sprint -----
21     <1> ; Функция печати сообщения
22     <1> ; входные данные: mov eax, <message>
```

Рис. 3.11: Файл листинга

строка 5: нули-адреса, 53-байты, справа push-имя ebx-операнда, те сохранить регистр,положить в стек строка 16: 0000000B-адрес, 5B-байты, рор-имя ebx- операнда, извлечения значения стека строка 12: 00000009-адрес, EBF8-байты, jmp-имя nextchar-операнда, получение символа

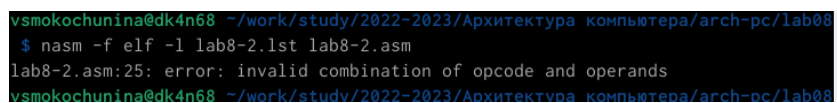
12. Я открыла файл с программой lab8-2.asm и удалила один операнд



```
lab8-2.asm [-M--] 7 L: [ 13+12 25/ 49] *(690 /1727b) 0010 0x00A [*][X]
; ----- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ----- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ----- Записываем 'A' в переменную 'max'
mov ecx[]
mov [max],ecx ; 'max = A'
; ----- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[C] ; Сравниваем 'A' и 'C'
jg check_B ; если 'A>C', то переход на метку 'check_B',
mov ecx,[C] ; иначе 'ecx = C'
mov [max],ecx ; 'max = C'
; ----- Преобразование 'max(A,C)' из символа в число
check_B:
mov eax,max
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюMC 10Выход
```

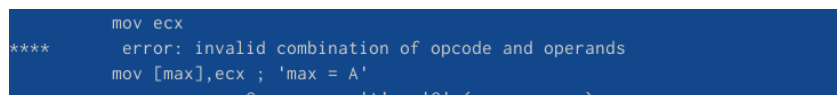
Рис. 3.12: Удаление операнда

13. Я выполнила трансляцию с получением файла листинга



```
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
$ nasm -f elf -l lab8-2.lst lab8-2.asm
lab8-2.asm:25: error: invalid combination of opcode and operands
vsmokochunina@dk4n68 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
```

Рис. 3.13: Трансляция файла



```
mov ecx
****      error: invalid combination of opcode and operands
mov [max],ecx ; 'max = A'
; ----- Сравниваем 'A' и 'C' (как символы)
```

Рис. 3.14: Ошибка

Файл не работает, тк в нем только одна операнда. А может работать только с двумя

4 Самостоятельная работа

1. Я написала программу для нахождения наименьшей из трех целочисленных переменных

[illegible]

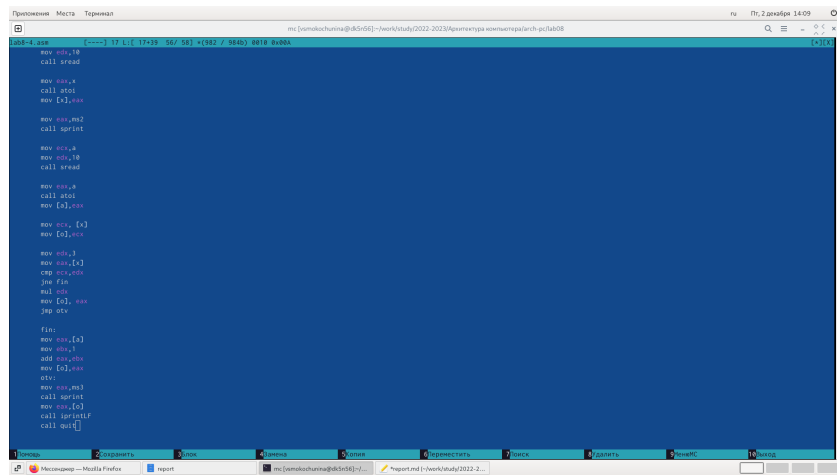
Рис. 4.1: Написание программы

- ## 2. Я создала и запустила файл

```
ysmkochunina@dk5n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08 $ nasm -f elf lab8-3.asm
ysmkochunina@dk5n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
ysmkochunina@dk5n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08 $ ./lab8-3
Результат: 5
ysmkochunina@dk5n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08 $
```

Рис. 4.2: Запуск программы

- ### 3. Я написала программу для вычисления функции



```
lab8-4.asm
mov esi,10
call read

mov esi,e
call atoi
mov [x],eax

mov esi,e2
call print

mov esi,e
mov esi,10
call read

mov esi,e
call atoi
mov [y],eax

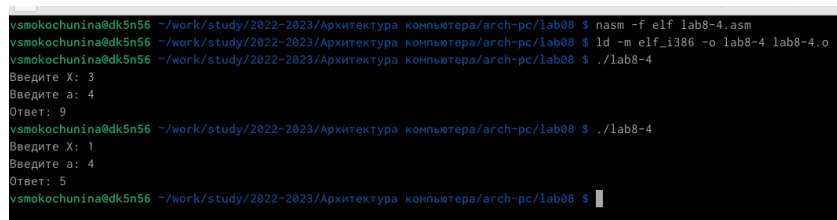
mov esi,[x]
mov esi,[y]
cmp esi,e2
jnz fin
mul esi
mov [z],eax
jmp fin

fin:
mov esi,[x]
mov esi,1
add esi,e2
mov [z],eax
ret

mov esi,e2
call printf
mov esi,[z]
call printf
call exit
```

Рис. 4.3: Написание программы

4. Я создала и запустила файл



```
vsmokochunina@dk5n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08 $ nasm -f elf lab8-4.asm
vsmokochunina@dk5n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08 $ ld -m elf_i386 -o lab8-4 lab8-4.o
vsmokochunina@dk5n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08 $ ./lab8-4
Введите X: 3
Введите a: 4
Ответ: 9
vsmokochunina@dk5n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08 $ ./lab8-4
Введите X: 1
Введите a: 4
Ответ: 5
vsmokochunina@dk5n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08 $
```

Рис. 4.4: Запуск программы

5 Выводы

Я изучила команды условного и безусловного переходов, приобрела навыки написания программ с использованием переходов и ознакомилась с назначением и структурой файлов листингов

Список литературы