

# **Отчёта по лабораторной работе 8**

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

Хусейнов Шухрат Наимжонович

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	23

## Список иллюстраций

3.1	Файл lab8-1.asm: . . . . .	8
3.2	Программа lab8-1.asm: . . . . .	9
3.3	Файл lab8-1.asm: . . . . .	10
3.4	Программа lab8-1.asm: . . . . .	11
3.5	Файл lab8-1.asm . . . . .	12
3.6	Программа lab8-1.asm . . . . .	13
3.7	Файл lab8-2.asm . . . . .	14
3.8	Программа lab8-2.asm . . . . .	15
3.9	Файл листинга lab8-2 . . . . .	16
3.10	ошибка трансляции lab8-2 . . . . .	17
3.11	файл листинга с ошибкой lab8-2 . . . . .	18
3.12	Файл lab8-3.asm . . . . .	19
3.13	Программа lab8-3.asm . . . . .	20
3.14	Файл lab8-4.asm . . . . .	21
3.15	Программа lab8-4.asm . . . . .	22

## **Список таблиц**

# 1 Цель работы

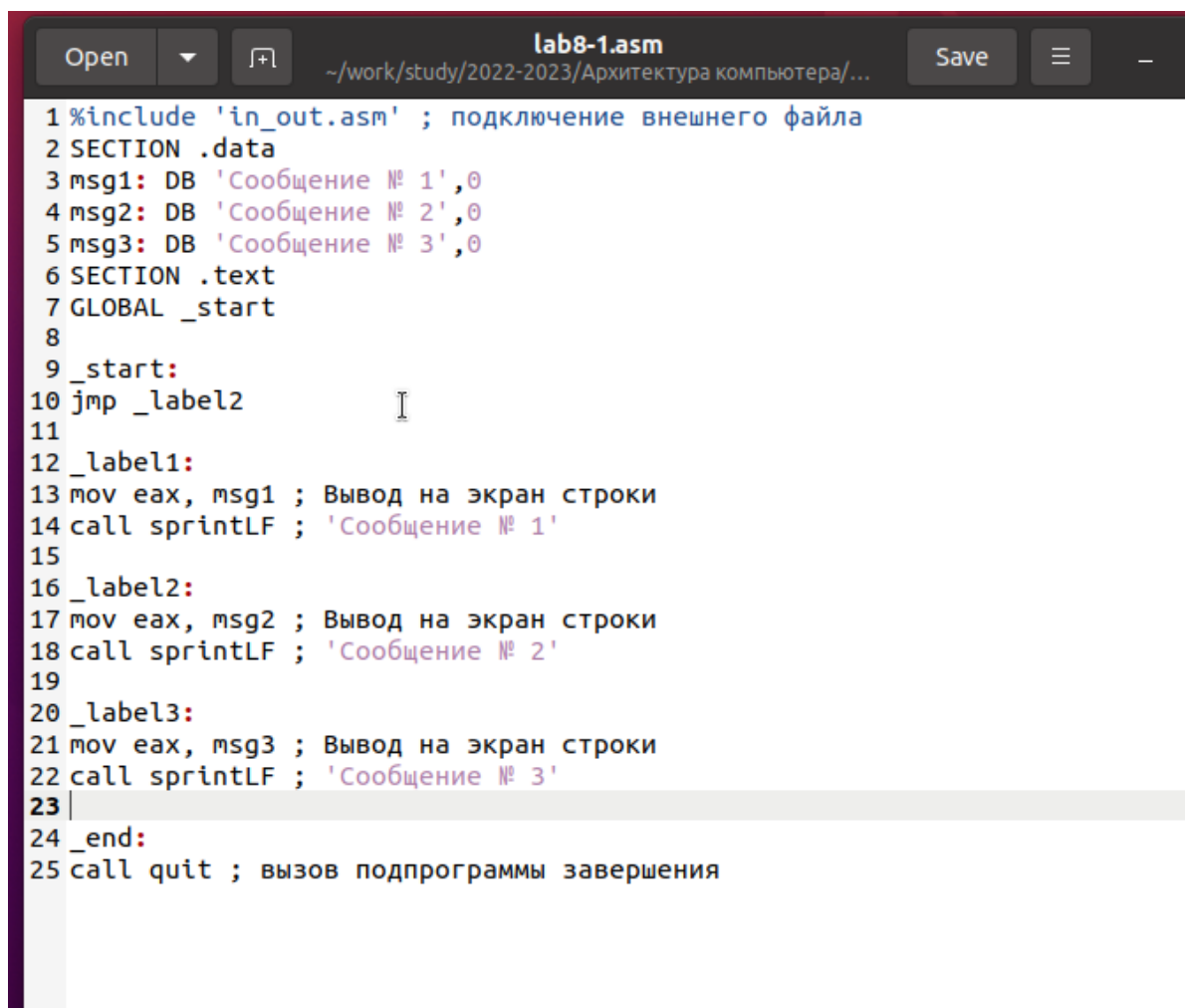
Целью работы является изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

## 2 Задание

1. Изучите примеры программ.
2. Изучите файл листинга.
3. Напишите программу нахождения наименьшей из 3 целочисленных переменных  $a, b$  и  $c$ . Значения переменных выбрать из табл. 8.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу
4. Напишите программу, которая для введенных с клавиатуры значений  $x$  и  $a$  вычисляет значение заданной функции  $f(x)$  и выводит результат вычислений. Вид функции  $f(x)$  выбрать из таблицы 8.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений  $X$  и  $a$  из 8.6.

### 3 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 8, перейдите в него и создайте файл lab8-1.asm
2. Инструкция jmp в NASM используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции jmp. Введите в файл lab8-1.asm текст программы из листинга 8.1. (рис. 3.1)

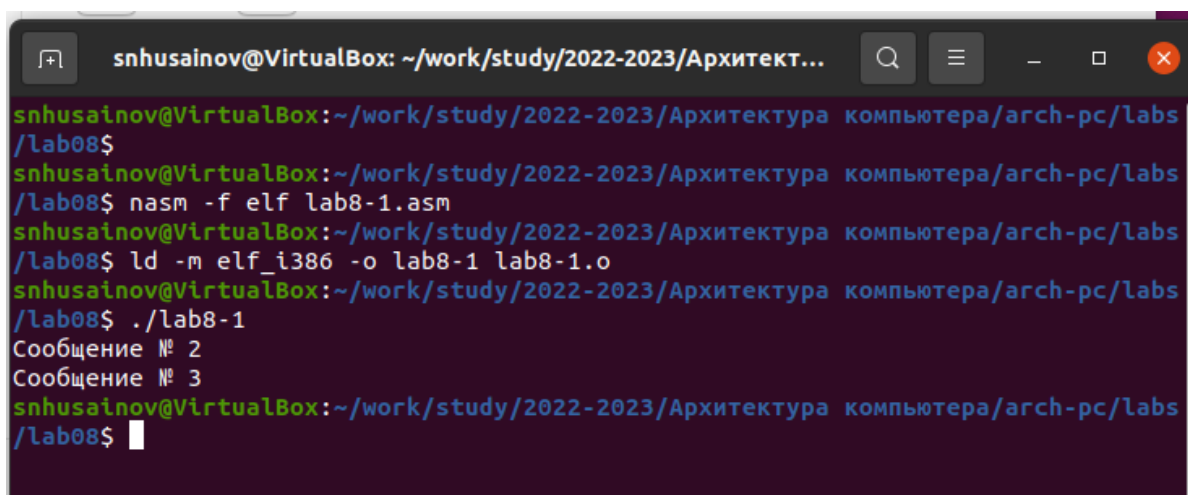


```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8
9 _start:
10 jmp _label2
11
12 _label1:
13 mov eax, msg1 ; Вывод на экран строки
14 call sprintf ; 'Сообщение № 1'
15
16 _label2:
17 mov eax, msg2 ; Вывод на экран строки
18 call sprintf ; 'Сообщение № 2'
19
20 _label3:
21 mov eax, msg3 ; Вывод на экран строки
22 call sprintf ; 'Сообщение № 3'
23
24 _end:
25 call quit ; вызов подпрограммы завершения
```

Рис. 3.1: Файл lab8-1.asm:

Создайте исполняемый файл и запустите его. (рис. 3.2)

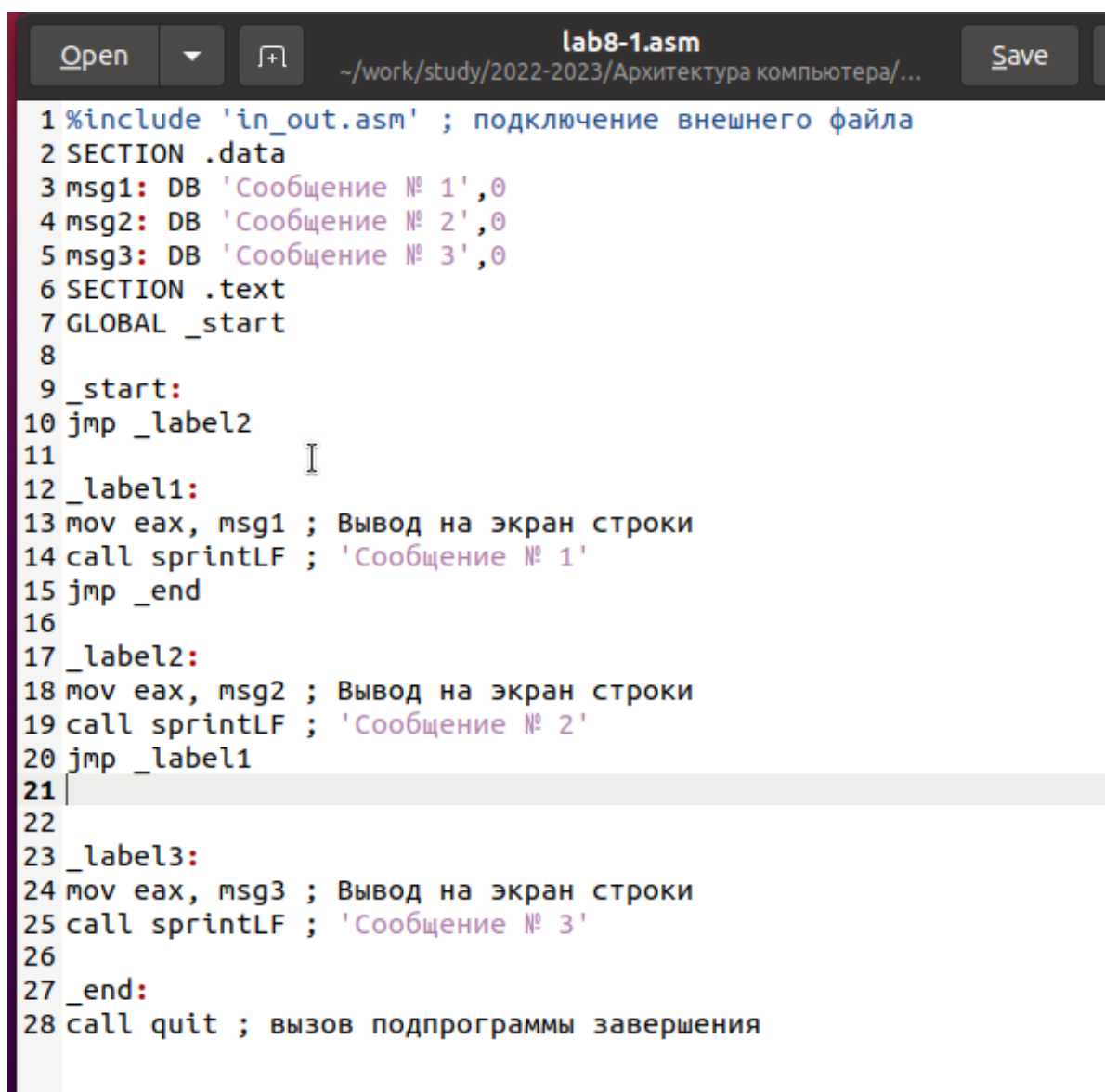




```
snhusainov@VirtualBox: ~/work/study/2022-2023/Архитект...
snhusainov@VirtualBox:~/work/study/2022-2023/Архитект.../arch-pc/labs/lab08$
snhusainov@VirtualBox:~/work/study/2022-2023/Архитект.../arch-pc/labs/lab08$ nasm -f elf lab8-1.asm
snhusainov@VirtualBox:~/work/study/2022-2023/Архитект.../arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
snhusainov@VirtualBox:~/work/study/2022-2023/Архитект.../arch-pc/labs/lab08$ ./lab8-1
Сообщение № 2
Сообщение № 3
snhusainov@VirtualBox:~/work/study/2022-2023/Архитект.../arch-pc/labs/lab08$
```

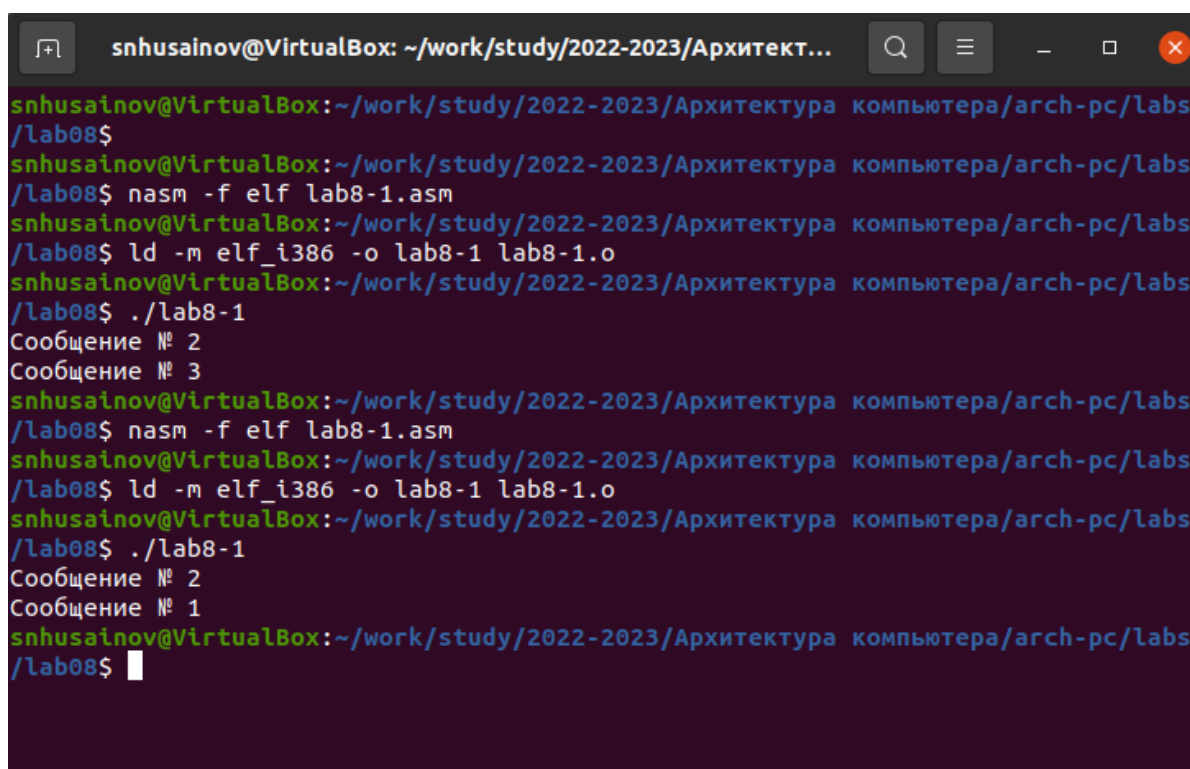
Рис. 3.2: Программа lab8-1.asm:

Инструкция `jmp` позволяет осуществлять переходы не только вперед но и назад. Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого в текст программы после вывода сообщения № 2 добавим инструкцию `jmp` с меткой `_label1` (т.е. переход к инструкциям вывода сообщения № 1) и после вывода сообщения № 1 добавим инструкцию `jmp` с меткой `_end` (т.е. переход к инструкции `call quit`). Измените текст программы в соответствии с листингом 8.2. (рис. 3.3, 3.4)



```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8
9 _start:
10 jmp _label2
11
12 _label1:
13 mov eax, msg1 ; Вывод на экран строки
14 call sprintf ; 'Сообщение № 1'
15 jmp _end
16
17 _label2:
18 mov eax, msg2 ; Вывод на экран строки
19 call sprintf ; 'Сообщение № 2'
20 jmp _label1
21
22
23 _label3:
24 mov eax, msg3 ; Вывод на экран строки
25 call sprintf ; 'Сообщение № 3'
26
27 _end:
28 call quit ; вызов подпрограммы завершения
```

Рис. 3.3: Файл lab8-1.asm:



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

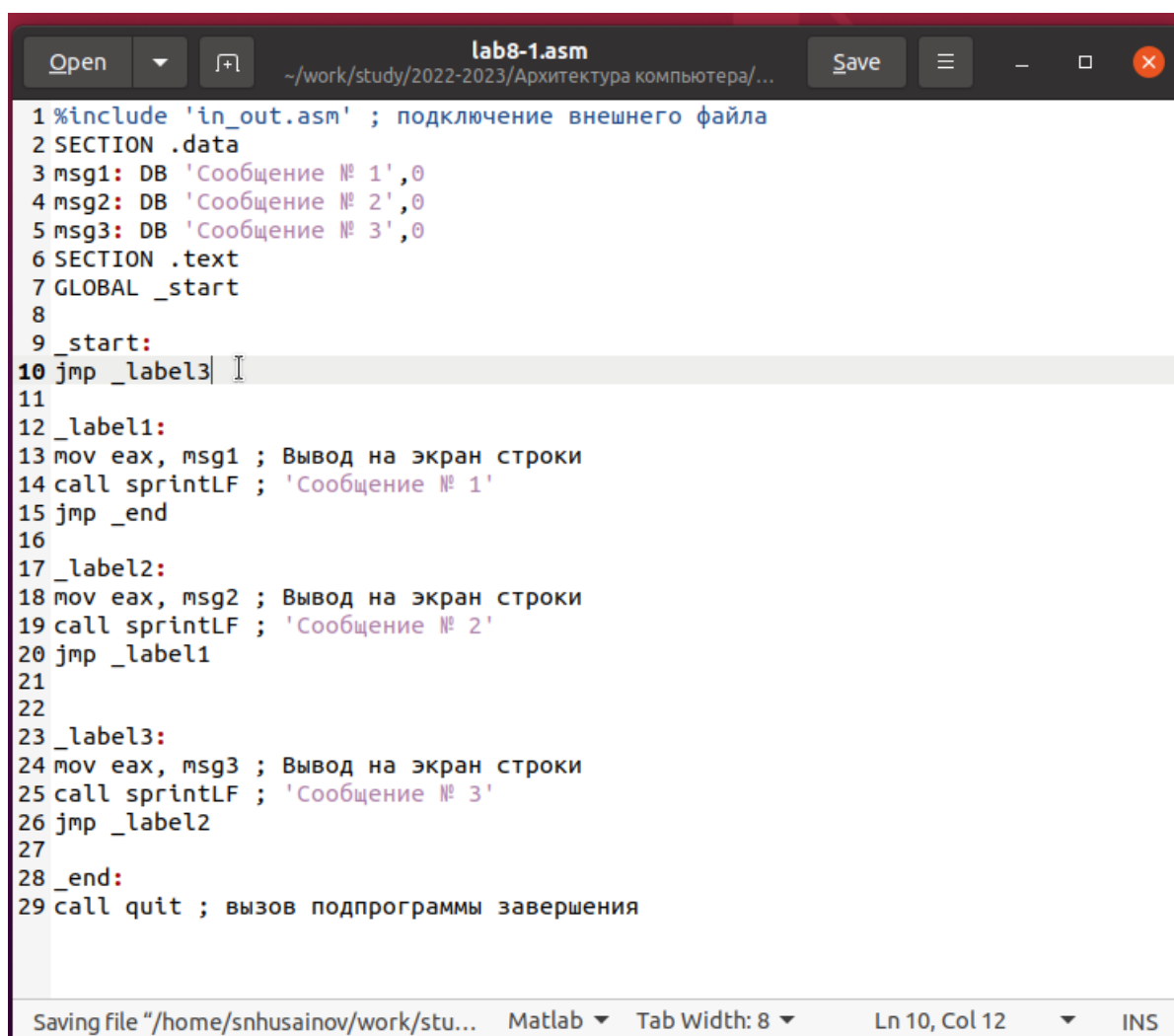
Рис. 3.4: Программа lab8-1.asm:

Измените текст программы добавив или изменив инструкции `jmp`, чтобы вывод программы был следующим (рис. 3.5, 3.6):

Сообщение № 3

Сообщение № 2

Сообщение № 1



```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8
9 _start:
10 jmp _label3
11
12 _label1:
13 mov eax, msg1 ; Вывод на экран строки
14 call sprintf ; 'Сообщение № 1'
15 jmp _end
16
17 _label2:
18 mov eax, msg2 ; Вывод на экран строки
19 call sprintf ; 'Сообщение № 2'
20 jmp _label1
21
22
23 _label3:
24 mov eax, msg3 ; Вывод на экран строки
25 call sprintf ; 'Сообщение № 3'
26 jmp _label2
27
28 _end:
29 call quit ; вызов подпрограммы завершения
```

Saving file "/home/snhusainov/work/stu... Matlab Tab Width: 8 Ln 10, Col 12 INS

Рис. 3.5: Файл lab8-1.asm

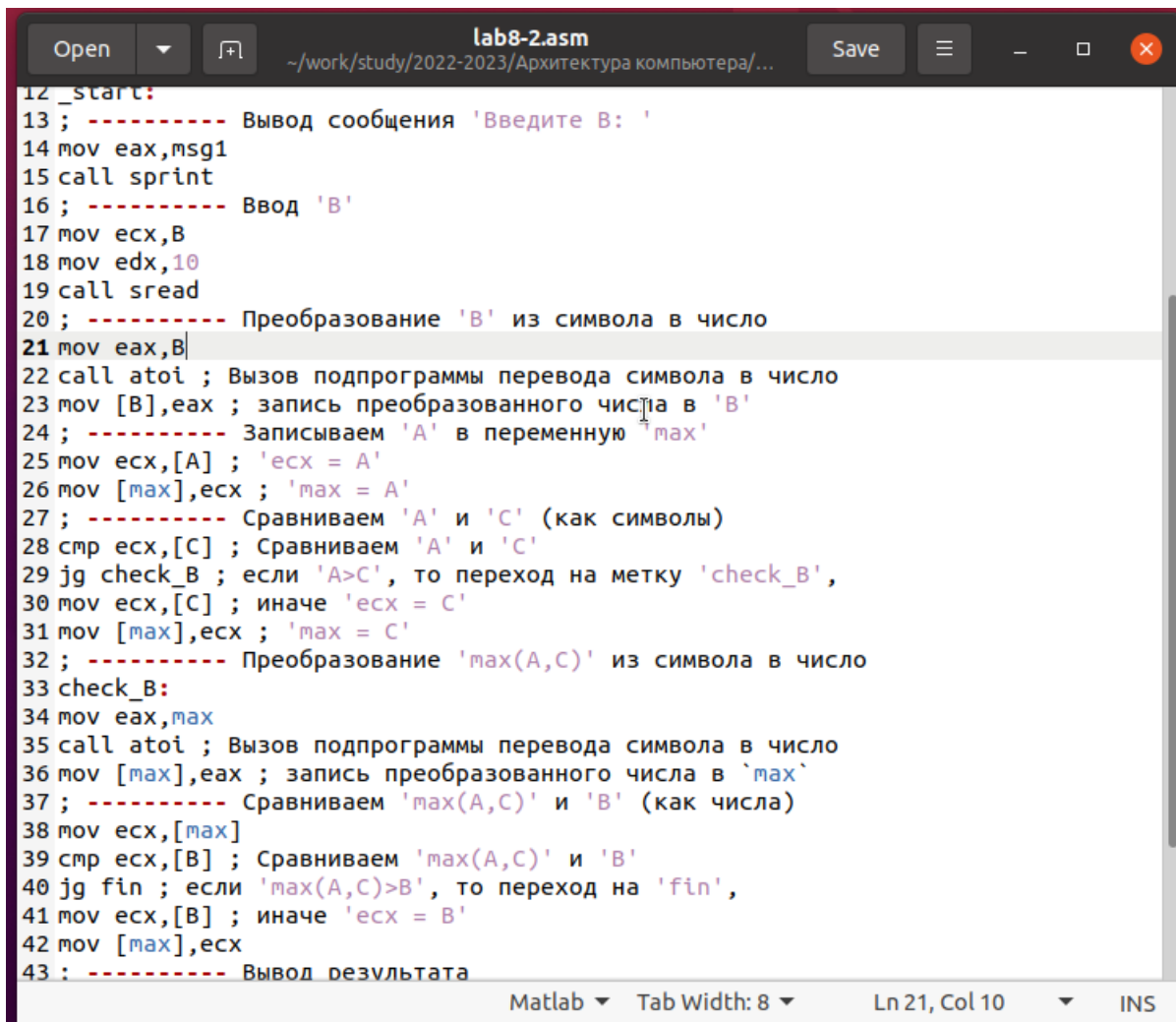
```

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

```

Рис. 3.6: Программа lab8-1.asm

3. Использование инструкции `jmp` приводит к переходу в любом случае. Однако, часто при написании программ необходимо использовать условные переходы, т.е. переход должен происходить если выполнено какое-либо условие. В качестве примера рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: А, В и С. Значения для А и С задаются в программе, значение В вводится с клавиатуры. Создайте исполняемый файл и проверьте его работу для разных значений В. (рис. 3.7, 3.8)



```
12 _start:
13 ; ----- Вывод сообщения 'Введите B: '
14 mov eax,msg1
15 call sprint
16 ; ----- Ввод 'B'
17 mov ecx,B
18 mov edx,10
19 call sread
20 ; ----- Преобразование 'B' из символа в число
21 mov eax,B
22 call atoi ; Вызов подпрограммы перевода символа в число
23 mov [B],eax ; запись преобразованного числа в 'B'
24 ; ----- Записываем 'A' в переменную 'max'
25 mov ecx,[A] ; 'ecx = A'
26 mov [max],ecx ; 'max = A'
27 ; ----- Сравниваем 'A' и 'C' (как символы)
28 cmp ecx,[C] ; Сравниваем 'A' и 'C'
29 jg check_B ; если 'A>C', то переход на метку 'check_B',
30 mov ecx,[C] ; иначе 'ecx = C'
31 mov [max],ecx ; 'max = C'
32 ; ----- Преобразование 'max(A,C)' из символа в число
33 check_B:
34 mov eax,max
35 call atoi ; Вызов подпрограммы перевода символа в число
36 mov [max],eax ; запись преобразованного числа в 'max'
37 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
38 mov ecx,[max]
39 cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
40 jg fin ; если 'max(A,C)>B', то переход на 'fin',
41 mov ecx,[B] ; иначе 'ecx = B'
42 mov [max],ecx
43 ; ----- Вывод результата
```

Matlab Tab Width: 8 Ln 21, Col 10 INS

Рис. 3.7: Файл lab8-2.asm

```

snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ nasm -f elf lab8-2.asm
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ ./lab8-2
Введите B: 54
Наибольшее число: 54
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ 62
62: command not found
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ ./lab8-2
Введите B: 10
Наибольшее число: 50
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ █

```

Рис. 3.8: Программа lab8-2.asm

4. Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ -l и задав имя файла листинга в командной строке. Создайте файл листинга для программы из файла lab8-2.asm (рис. 3.9)

```

lab8-2.lst
~/work/study/2022-2023/Архитектура компьютера/...
Save

33 32 0000001B 89C1 <1> mov ecx, eax
34 33 0000001D BB01000000 <1> mov ebx, 1
35 34 00000022 B804000000 <1> mov eax, 4
36 35 00000027 CD80 <1> int 80h
37 36 <1>
38 37 00000029 5B <1> pop ebx
39 38 0000002A 59 <1> pop ecx
40 39 0000002B 5A <1> pop edx
41 40 0000002C C3 <1> ret
42 41 <1>
43 42 <1>
44 43 <1> ;----- sprintLF
-----
45 44 <1> ; Функция печати сообщения с
переводом строки
46 45 <1> ; входные данные: mov eax,<message>
47 46 <1> sprintLF:
48 47 0000002D E8DDFFFFFF <1> call sprint
49 48 <1>
50 49 00000032 50 <1> push eax
51 50 00000033 B80A000000 <1> mov eax, 0AH
52 51 00000038 50 <1> push eax
53 52 00000039 89E0 <1> mov eax, esp
54 53 0000003B E8CFFFFFFF <1> call sprint
55 54 00000040 58 <1> pop eax
56 55 00000041 58 <1> pop eax
57 56 00000042 C3 <1> ret
58 57 <1>
59 58 <1> ;----- sread
-----
60 59 <1> ; Функция считывания сообщения
61 60 <1> ; входные данные: mov eax, <buffer>

```

Рис. 3.9: Файл листинга lab8-2

Внимательно ознакомиться с его форматом и содержимым. Подробно объяснить содержимое трёх строк файла листинга по выбору.

строка 47

- 47 - номер строки
- 0000002D - адрес
- E8DDFFFFFF - машинный код
- call sprint - код программы



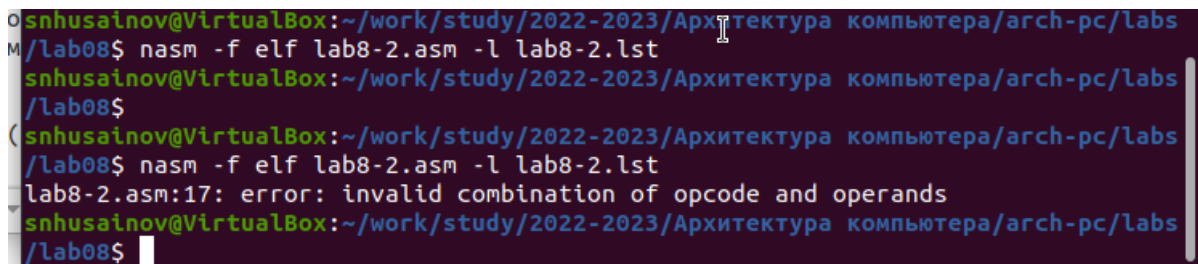
строка 49

- 49 - номер строки
- 00000032 - адрес
- 50 - машинный код
- push eax - код программы

строка 50

- 50 - номер строки
- 00000033 - адрес
- B80A000000 - машинный код
- mov eax, 0AH - код программы

Откройте файл с программой lab8-2.asm и в любой инструкции с двумя операндами удалите один операнд. Выполните трансляцию с получением файла листинга (рис. 3.10, 3.11)



```
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/ lab08$ nasm -f elf lab8-2.asm -l lab8-2.lst
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/ lab08$
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/ lab08$ nasm -f elf lab8-2.asm -l lab8-2.lst
lab8-2.asm:17: error: invalid combination of opcode and operands
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/ lab08$
```

Рис. 3.10: ошибка трансляции lab8-2

```

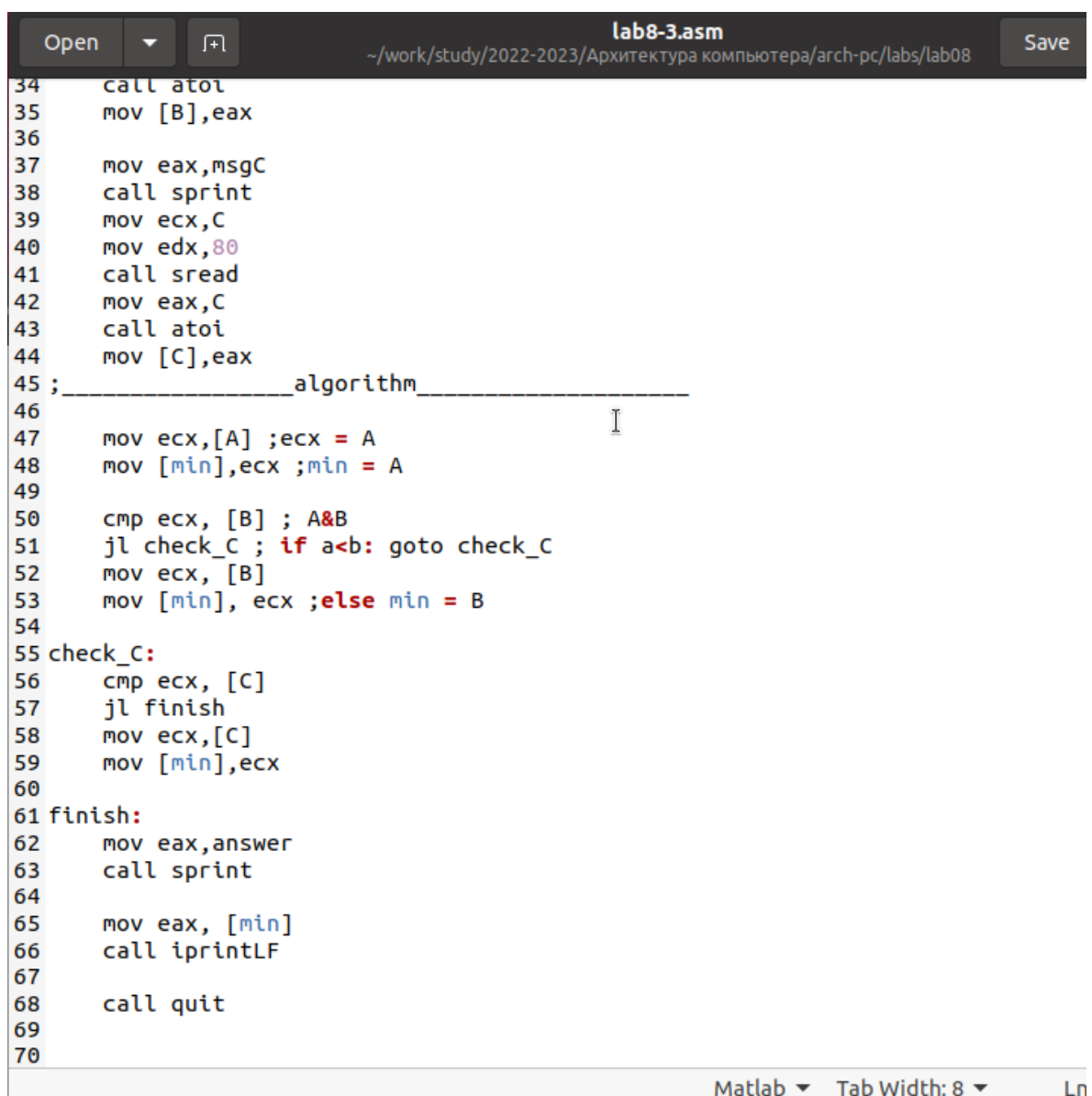
lab8-2.lst
~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab08
lab8-2.asm
lab8-2.lst
184      9 0000000A <res 0000000A>      B resb 10
185     10                                section .text
186     11                                global _start
187     12                                _start:
188     13                                ; ----- Вывод сообщения 'Введите B: '
189     14 000000E8 B8[00000000]      mov eax,msg1
190     15 000000ED E81DFFFFFF      call sprint
191     16                                ; ----- Ввод 'B'
192     17                                mov ecx
193     17                                *****
194     18 000000F2 BA0A000000      error: invalid combination of opcode and operands
195     19 000000F7 E847FFFFFF      mov edx,10
196     20                                call sread
197     21 000000FC B8[0A000000]      ; ----- Преобразование 'B' из символа в число
198     22 00000101 E896FFFFFF      mov eax,B
199                                call atoi ; Вызов подпрограммы перевода символа в
200                                число
201     23 00000106 A3[0A000000]      mov [B],eax ; запись преобразованного числа в 'B'
202     24                                ; ----- Записываем 'A' в переменную 'max'
203     25 0000010B 8B0D[35000000]      mov ecx,[A] ; 'ecx = A'
204     26 00000111 890D[00000000]      mov [max],ecx ; 'max = A'
205     27                                ; ----- Сравниваем 'A' и 'C' (как символы)
206     28 00000117 3B0D[39000000]      cmp ecx,[C] ; Сравниваем 'A' и 'C'
207     29 0000011D 7F0C      jg check_B ; если 'A>C', то переход на метку
208                                'check_B',
209     30 0000011F 8B0D[39000000]      mov ecx,[C] ; иначе 'ecx = C'
210     31 00000125 890D[00000000]      mov [max],ecx ; 'max = C'
211     32                                ; ----- Преобразование 'max(A,C)' из символа в
212                                число
213     33                                check_B:
214     34 0000012B B8[00000000]      mov eax,max
215     35 00000130 E867FFFFFF      call atoi ; Вызов подпрограммы перевода символа в
216                                число
217     36 00000135 A3[00000000]      mov [max],eax ; запись преобразованного числа в 'max'
218     37                                ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)

```

Рис. 3.11: файл листинга с ошибкой lab8-2

5. Напишите программу нахождения наименьшей из 3 целочисленных переменных a,b и c. Значения переменных выбрать из табл. 8.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу (рис. 3.12,3.13)

для варианта 5 - 54, 62, 87



```
34  call atoi
35  mov [B],eax
36
37  mov eax,msgC
38  call sprint
39  mov ecx,C
40  mov edx,80
41  call sread
42  mov eax,C
43  call atoi
44  mov [C],eax
45 ;-----algorithm-----
46
47  mov ecx,[A] ;ecx = A
48  mov [min],ecx ;min = A
49
50  cmp ecx, [B] ; A&B
51  jl check_C ; if a<b: goto check_C
52  mov ecx, [B]
53  mov [min], ecx ;else min = B
54
55 check_C:
56  cmp ecx, [C]
57  jl finish
58  mov ecx,[C]
59  mov [min],ecx
60
61 finish:
62  mov eax,answer
63  call sprint
64
65  mov eax, [min]
66  call iprintLF
67
68  call quit
69
70
```

Рис. 3.12: Файл lab8-3.asm

```

snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ nasm -f elf lab8-3.asm
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ ./lab8-3
Input A: 54
Input B: 62
Input C: 87
Smallest: 54
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs

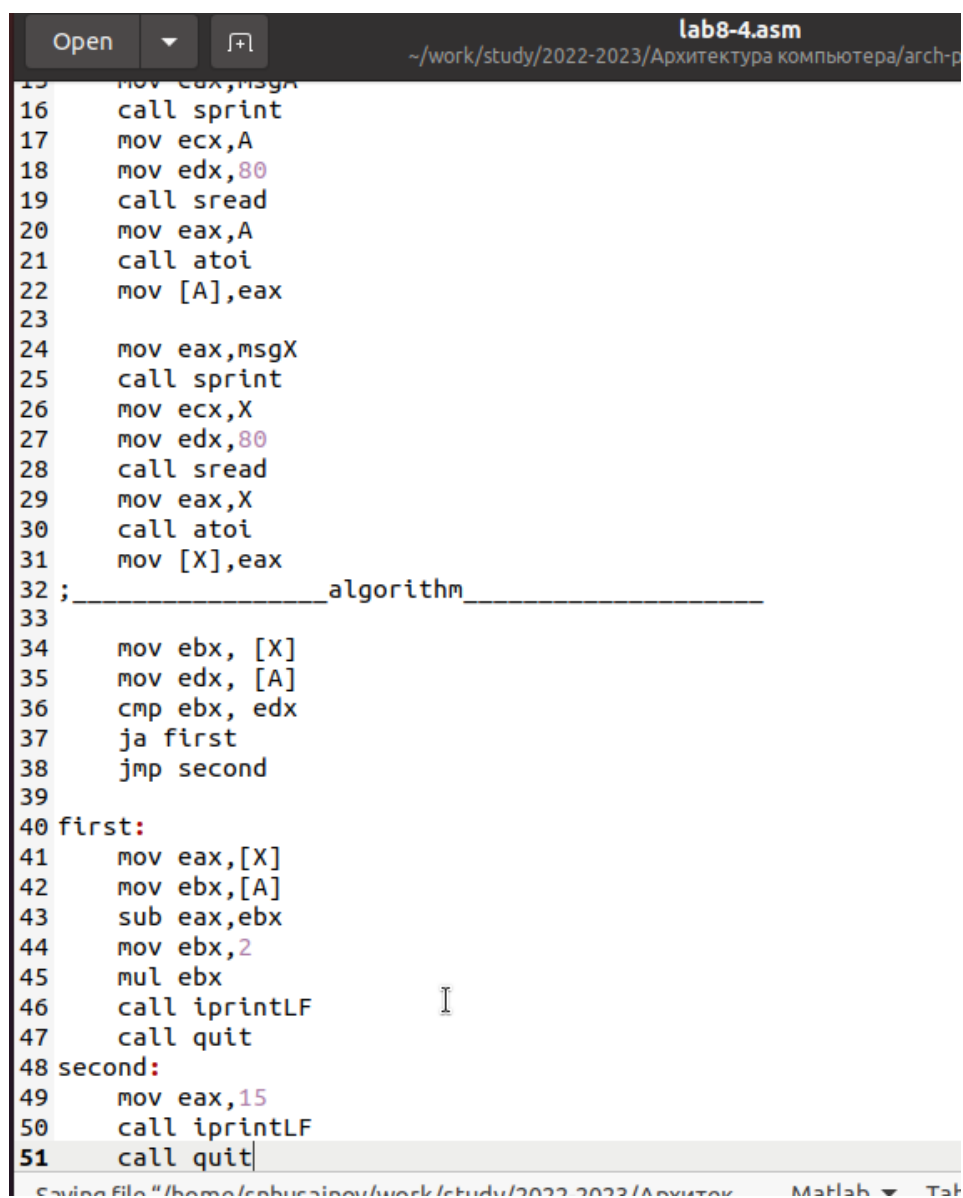
```

Рис. 3.13: Программа lab8-3.asm

6. Напишите программу, которая для введенных с клавиатуры значений  $x$  и  $a$  вычисляет значение заданной функции  $f(x)$  и выводит результат вычислений. Вид функции  $f(x)$  выбрать из таблицы 8.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений  $x$  и  $a$  из 8.6. (рис. 3.14,3.15)

для варианта 5

$$\begin{cases} 2(x - a) + 1, x > a \\ 15, x \leq a \end{cases}$$



```
lab8-4.asm
~/work/study/2022-2023/Архитектура компьютера/arch-p
15  mov ecx,msgX
16  call sprint
17  mov ecx,A
18  mov edx,80
19  call sread
20  mov eax,A
21  call atoi
22  mov [A],eax
23
24  mov eax,msgX
25  call sprint
26  mov ecx,X
27  mov edx,80
28  call sread
29  mov eax,X
30  call atoi
31  mov [X],eax
32 ; _____algorithm_____
33
34  mov ebx, [X]
35  mov edx, [A]
36  cmp ebx, edx
37  ja first
38  jmp second
39
40 first:
41  mov eax,[X]
42  mov ebx,[A]
43  sub eax,ebx
44  mov ebx,2
45  mul ebx
46  call iprintLF
47  call quit
48 second:
49  mov eax,15
50  call iprintLF
51  call quit
```

Рис. 3.14: Файл lab8-4.asm

```
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ nasm -f elf lab8-4.asm
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ ./lab8-4
Input A: 2
Input X: 1
15
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$ ./lab8-4
Input A: 1
Input X: 2
2
snhusainov@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs
/lab08$
```

Рис. 3.15: Программа lab8-4.asm

## 4 Выводы

Изучили команды условного и безусловного переходов, познакомились с фалом листинга.