

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина: Архитектура компьютера

Студент: Татур Платон Андреевич

Группа: НКАбд-05-25

МОСКВА

2025 г.

7.1. Цель работы

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

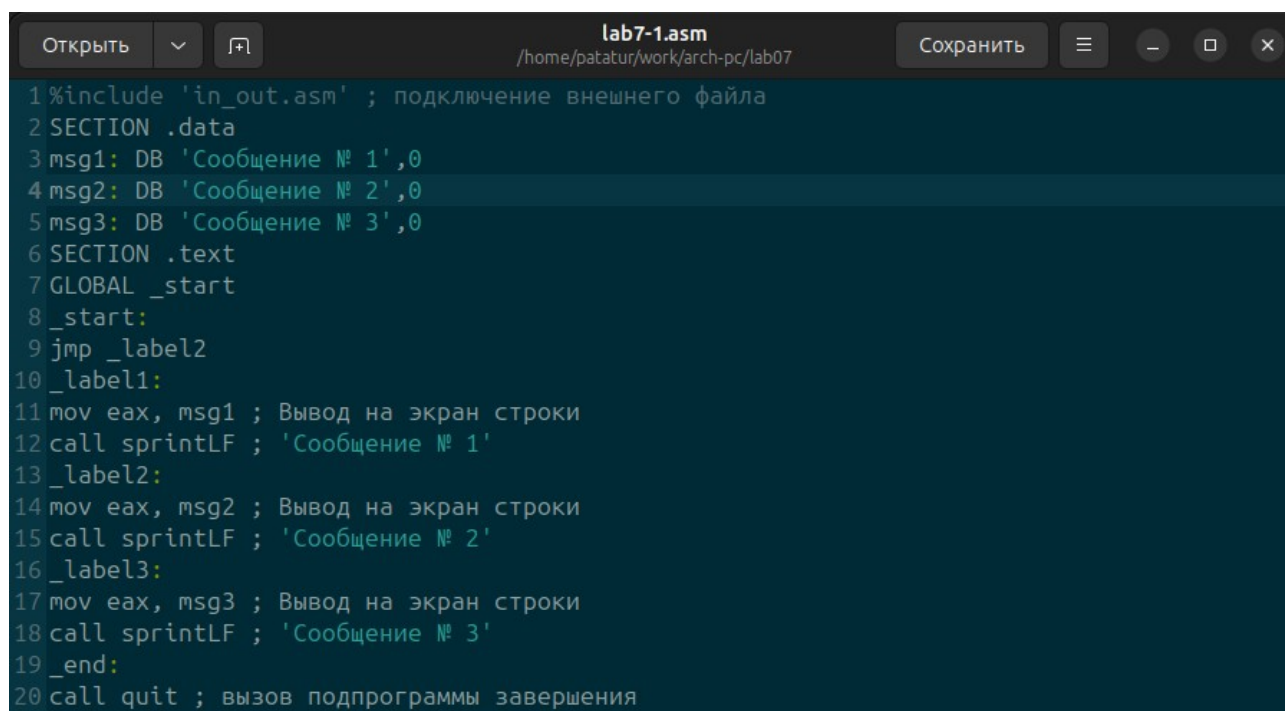
7.3. Порядок выполнения лабораторной работы

7.3.1. Реализация переходов в NASM

```
patatur@platonpc:~$ mkdir ~/work/arch-pc/lab07
patatur@platonpc:~$ cd ~/work/arch-pc/lab07
patatur@platonpc:~/work/arch-pc/lab07$ touch lab7-1.asm
```

Рис 1

1) Создал каталог для программ лабораторной работы № 7, перешел в него и создал файл lab7-1.asm (см Рис 1)



```
lab7-1.asm
/home/patatur/work/arch-pc/lab07
Открыть  Сохранить

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 _start:
9 jmp _label2
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintLF ; 'Сообщение № 1'
13 _label2:
14 mov eax, msg2 ; Вывод на экран строки
15 call sprintLF ; 'Сообщение № 2'
16 _label3:
17 mov eax, msg3 ; Вывод на экран строки
18 call sprintLF ; 'Сообщение № 3'
19 _end:
20 call quit ; вызов подпрограммы завершения
```

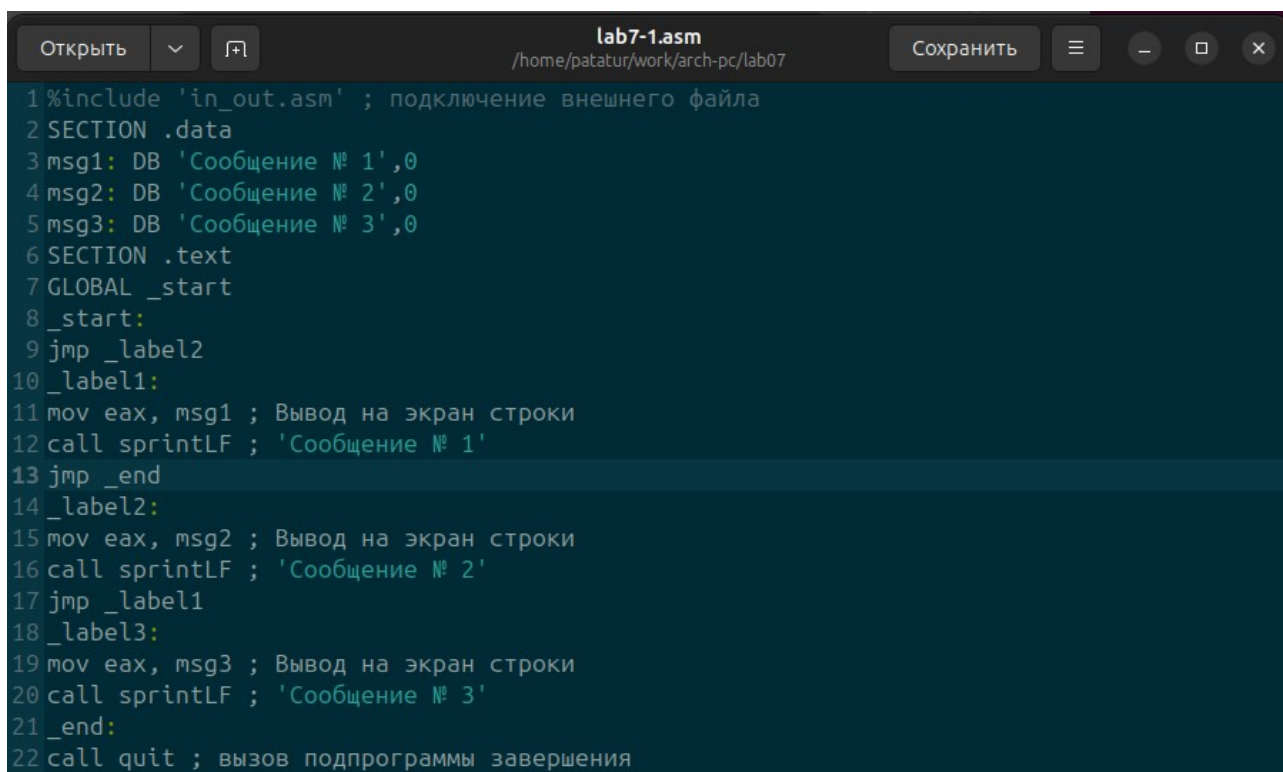
Рис 2.1

```
patatur@platonpc:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
patatur@platonpc:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
patatur@platonpc:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
```

Рис 2.2

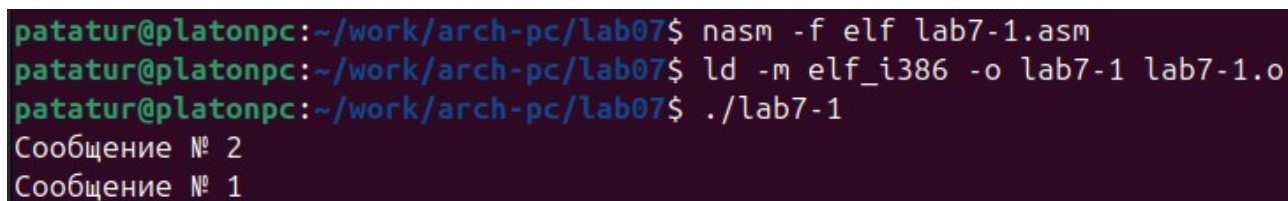
1) Ввел в файл lab7-1.asm текст программы из листинга 7.1 (см Рис 2.1)

2) Создал исполняемый файл и запустил его (см Рис 2.2)



```
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 _start:
9 jmp _label2
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 jmp _end
14 _label2:
15 mov eax, msg2 ; Вывод на экран строки
16 call sprintf ; 'Сообщение № 2'
17 jmp _label1
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение № 3'
21 _end:
22 call quit ; вызов подпрограммы завершения
```

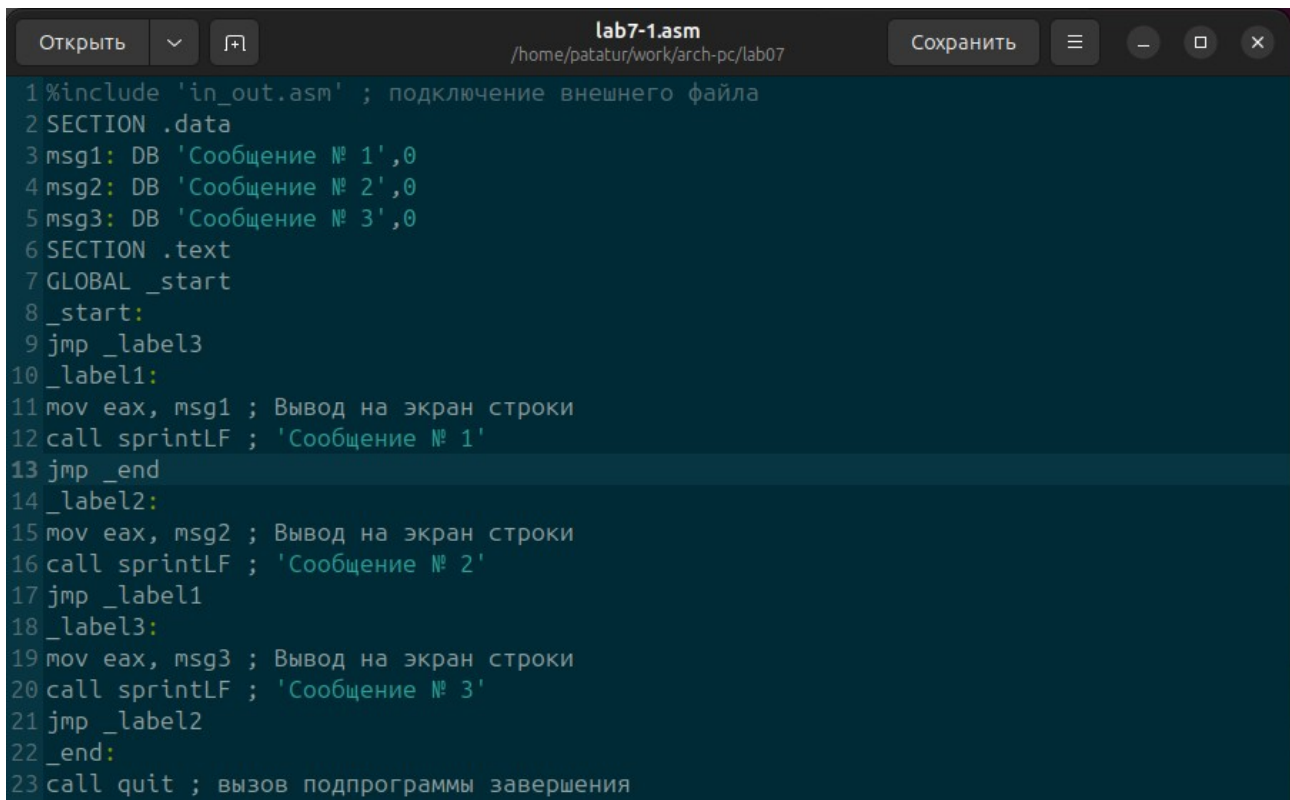
Рис 2.3



```
patatur@platonpc:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
patatur@platonpc:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
patatur@platonpc:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
```

Рис 2.4

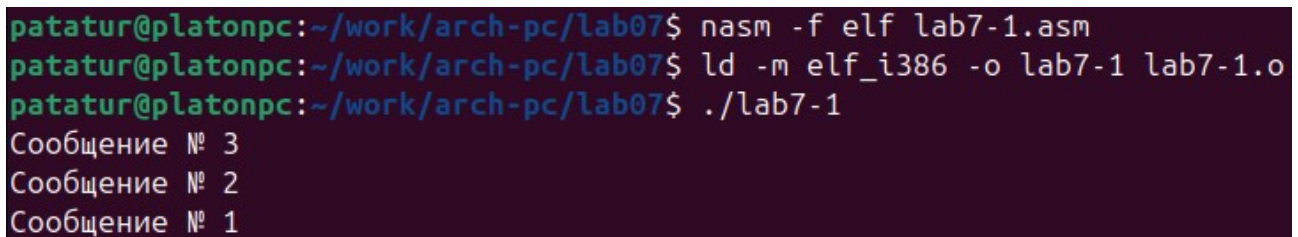
- 1) Изменил текст программы в соответствии с листингом 7.2 (см Рис 2.3)
 - 2) Создал исполняемый файл и запустил его (см Рис 2.4)
-



```
lab7-1.asm
/home/patatur/work/arch-pc/lab07
Сохранить

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 _start:
9 jmp _label3
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 jmp _end
14 _label2:
15 mov eax, msg2 ; Вывод на экран строки
16 call sprintf ; 'Сообщение № 2'
17 jmp _label1
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение № 3'
21 jmp _label2
22 _end:
23 call quit ; вызов подпрограммы завершения
```

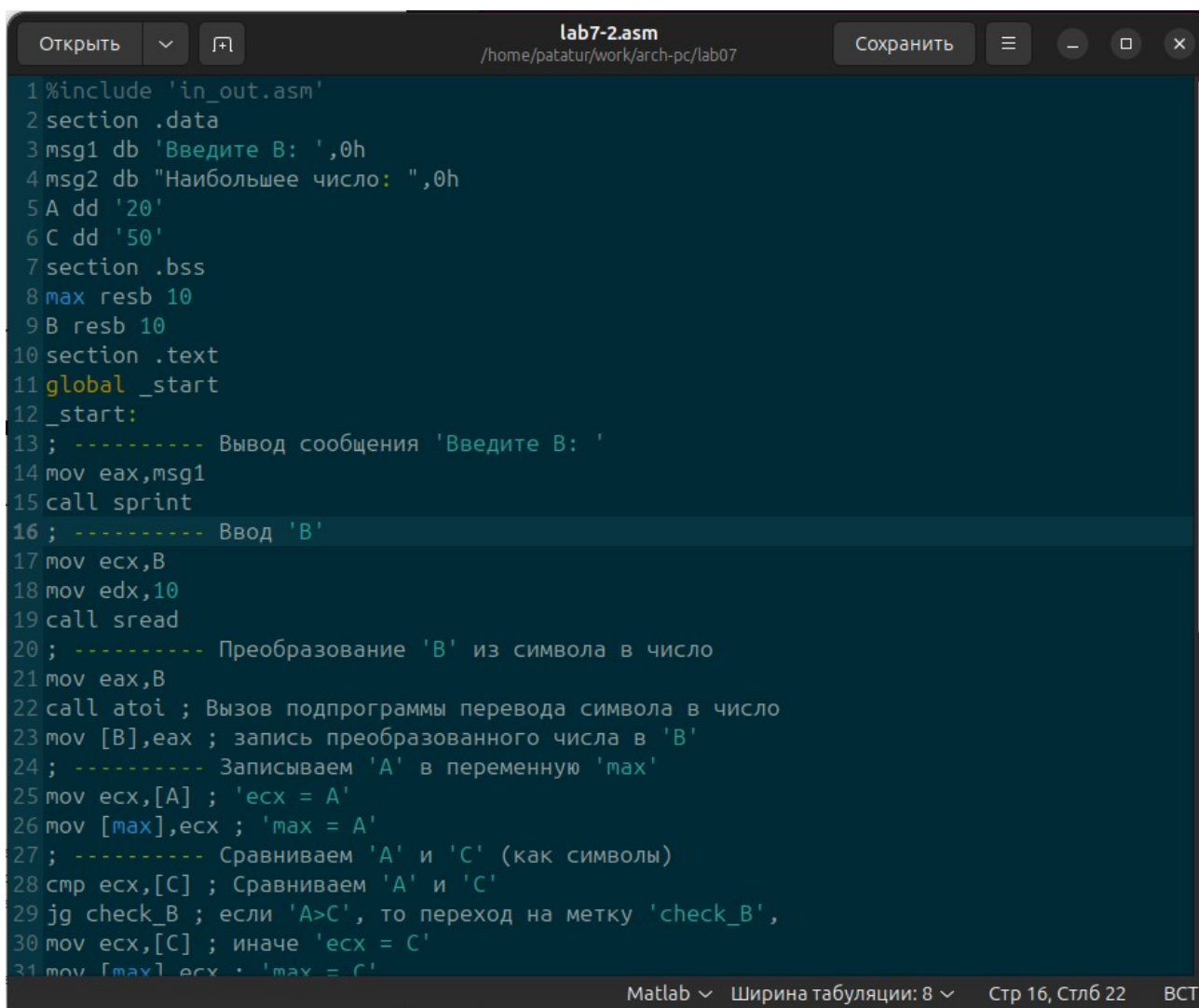
Рис 2.5



```
patatur@platonpc:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
patatur@platonpc:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
patatur@platonpc:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
```

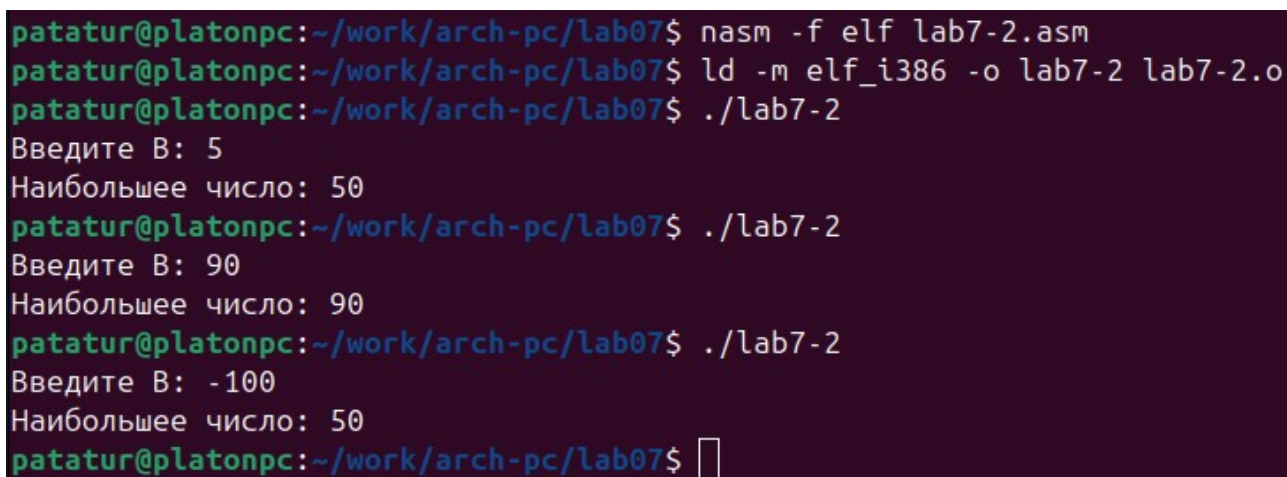
Рис 2.6

1) Изменил текст программы добавив и изменив инструкции jmp, чтобы вывод программы был как указанный в тексте лабораторной работы (см Рис 2.5 и Рис 2.6)



```
1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите B: ',0h
4 msg2 db "Наибольшее число: ",0h
5 A dd '20'
6 C dd '50'
7 section .bss
8 max resb 10
9 B resb 10
10 section .text
11 global _start
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'
```

Рис 3.1



```
patatur@platonpc:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
patatur@platonpc:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
patatur@platonpc:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 5
Наибольшее число: 50
patatur@platonpc:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 90
Наибольшее число: 90
patatur@platonpc:~/work/arch-pc/lab07$ ./lab7-2
Введите B: -100
Наибольшее число: 50
patatur@platonpc:~/work/arch-pc/lab07$
```

Рис 3.2

- 1) Создал файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и ввел в lab7-2.asm текст листинга 7.3 (см Рис 3.1)
 - 2) Создал исполняемый файл и проверил его работу для разных значений B
-

7.3.2. Изучение структуры файлы листинга

```
patatur@platonpc:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
patatur@platonpc:~/work/arch-pc/lab07$ gedit lab7-2.lst
```

Рис 4.1

```
32 0000001B 89C1          <1>      mov     ecx, eax
33 0000001D BB01000000      <1>      mov     ebx, 1
34 00000022 B804000000      <1>      mov     eax, 4
```

Рис 4.2

- 1) Создал файл листинга для программы из файла lab7-2.asm и открыл его через редактор gedit (см Рис 4.1)
- 2) Объяснение строк из файла lab7-2.lst (Рис 4.2):

- а) Первые двк цифры в строках – это номера этих строк
- б) Далее идет восьмизначный адрес
- с) Затем идет машинный код в шестнадцатичной системе счисления
- д) После символов “<1>” идет исходный текст программы

```
; ----- Вывод сообщения 'Введите В: '
mov eax,msg1
call sprint
```

Рис 4.3

```
; ----- Вывод сообщения 'Введите В: '
mov eax
call sprint
```

Рис 4.4

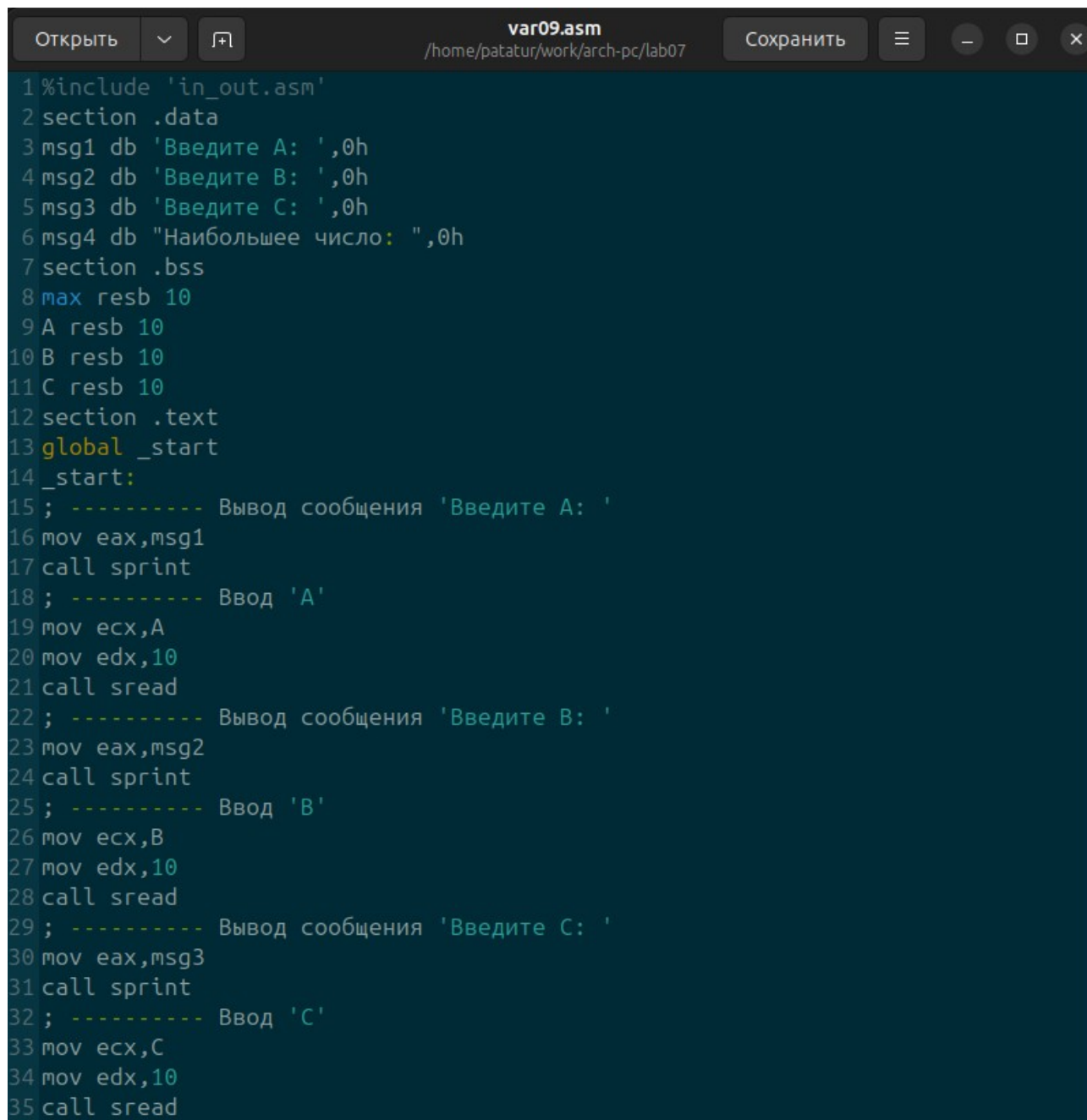
```
13          ; ----- Вывод сообщения
'Введите В: '
14          mov eax
14          *****
error: invalid combination of
opcode and operands
```

Рис 4.5

- 1) Открыл в редакторе программу lab7-2.asm и удалил в строке операнд msg1 (см Рис 4.3 и Рис 4.4)
 - 2) Выполнил трансляцию с получением файла листинга
- В этом случае в файле листинг четко видно ошибку (см Рис 4.5)
-

7.4. Задание для самостоятельной работы

1. Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и c . Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу.]



```
1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите A: ',0h
4 msg2 db 'Введите B: ',0h
5 msg3 db 'Введите C: ',0h
6 msg4 db "Наибольшее число: ",0h
7 section .bss
8 max resb 10
9 A resb 10
10 B resb 10
11 C resb 10
12 section .text
13 global _start
14 _start:
15 ; ----- Вывод сообщения 'Введите A: '
16 mov eax,msg1
17 call sprint
18 ; ----- Ввод 'A'
19 mov ecx,A
20 mov edx,10
21 call sread
22 ; ----- Вывод сообщения 'Введите B: '
23 mov eax,msg2
24 call sprint
25 ; ----- Ввод 'B'
26 mov ecx,B
27 mov edx,10
28 call sread
29 ; ----- Вывод сообщения 'Введите C: '
30 mov eax,msg3
31 call sprint
32 ; ----- Ввод 'C'
33 mov ecx,C
34 mov edx,10
35 call sread
```

Рис 5.1


```

37
38 ; ----- Преобразование 'B' из символа в число
39 mov eax,B
40 call atoi ; Вызов подпрограммы перевода символа в число
41 mov [B],eax ; запись преобразованного числа в 'B'
42
43
44 ; ----- Записываем 'A' в переменную 'max'
45 mov ecx,[A] ; 'ecx = A'
46 mov [max],ecx ; 'max = A'
47 ; ----- Сравниваем 'A' и 'C' (как символы)
48 cmp ecx,[C] ; Сравниваем 'A' и 'C'
49 jg check_B ; если 'A>C', то переход на метку 'check_B',
50 mov ecx,[C] ; иначе 'ecx = C'
51 mov [max],ecx ; 'max = C'
52 ; ----- Преобразование 'max(A,C)' из символа в число
53 check_B:
54 mov eax,max
55 call atoi ; Вызов подпрограммы перевода символа в число
56 mov [max],eax ; запись преобразованного числа в 'max'
57 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
58 mov ecx,[max]
59 cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
60 jg fin ; если 'max(A,C)>B', то переход на 'fin',
61 mov ecx,[B] ; иначе 'ecx = B'
62 mov [max],ecx
63 ; ----- Вывод результата
64 fin:
65 mov eax,msg4
66 call sprintf ; Вывод сообщения 'Наибольшее число: '
67 mov eax,[max]
68 call fprintf ; Вывод 'max(A,B,C)'
69 call quit ; Выход

```

Рис 5.2

```
patatur@platonpc:~/work/arch-pc/lab07$ nasm -f elf var09.asm
patatur@platonpc:~/work/arch-pc/lab07$ ld -m elf_i386 -o var09 var09.o
patatur@platonpc:~/work/arch-pc/lab07$
patatur@platonpc:~/work/arch-pc/lab07$ ./var09
Введите A: 1
Введите B: 2
Введите C: 3
Наибольшее число: 3
patatur@platonpc:~/work/arch-pc/lab07$ ./var09
Введите A: 24
Введите B: 98
Введите C: 15
Наибольшее число: 98
patatur@platonpc:~/work/arch-pc/lab07$ █
```

Рис 5.3

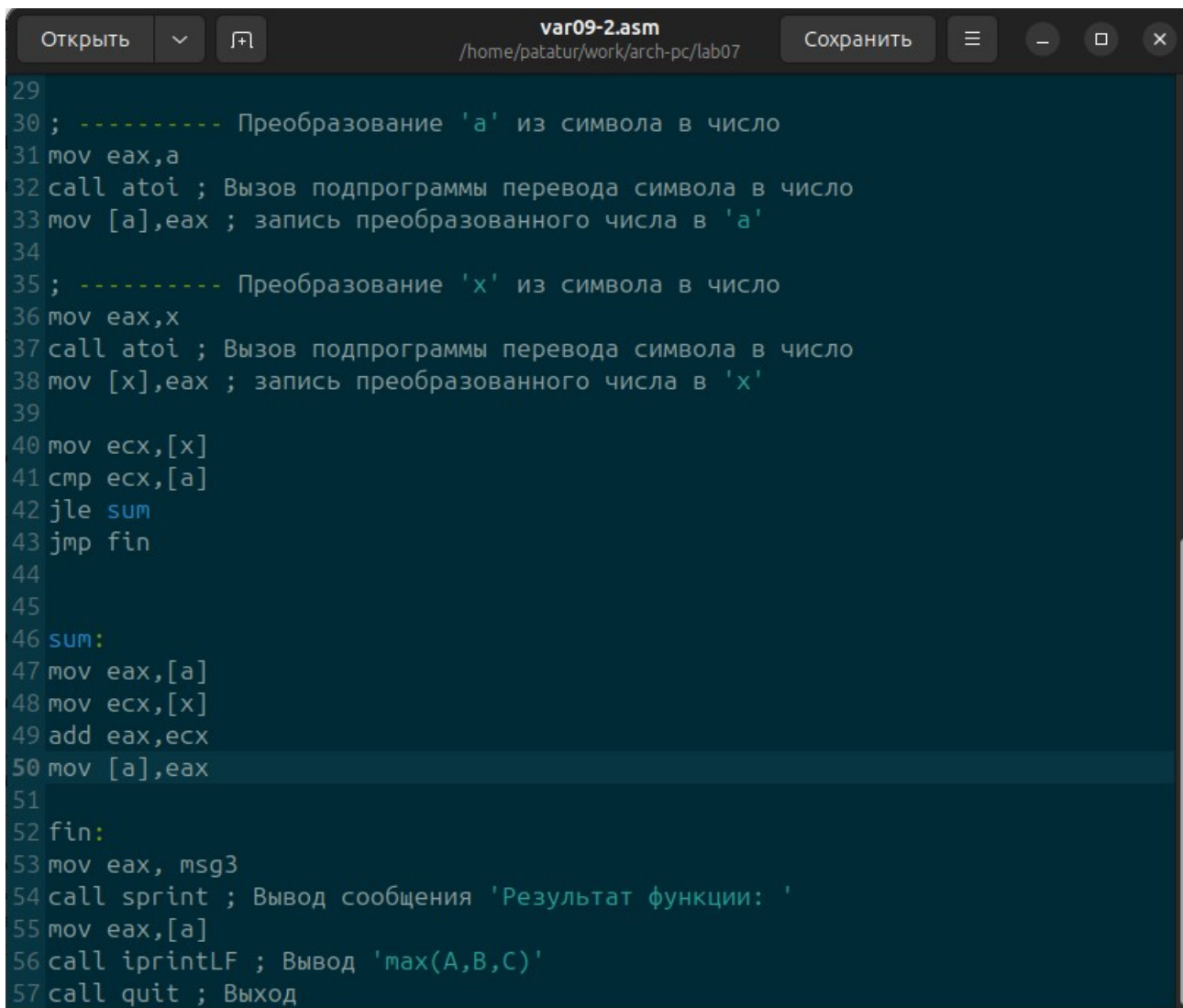
- 1) Создал файл var09.asm и написал программу (см Рис 5.1 и Рис 5.2)
- 2) Создал исполняемый файл и проверил работу программы, взяв значения для проверки из таблицы 7.5 под номеров варианта 9 (см Рис 5.3)

2. 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 7.6 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 7.6.

```
Открыть  var09-2.asm  Сохранить
/home/patatur/work/arch-pc/lab07

1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите a: ',0h
4 msg2 db 'Введите x: ',0h
5 msg3 db "Результат функции: ",0h
6
7 section .bss
8 a resb 10
9 x resb 10
10
11 section .text
12 global _start
13 _start:
14 ; ----- Вывод сообщения 'Введите a: '
15 mov eax,msg1
16 call sprint
17 ; ----- Ввод 'a'
18 mov ecx,a
19 mov edx,10
20 call sread
21
22 ; ----- Вывод сообщения 'Введите x: '
23 mov eax,msg2
24 call sprint
25 ; ----- Ввод 'x'
26 mov ecx,x
27 mov edx,10
28 call sread
29
```

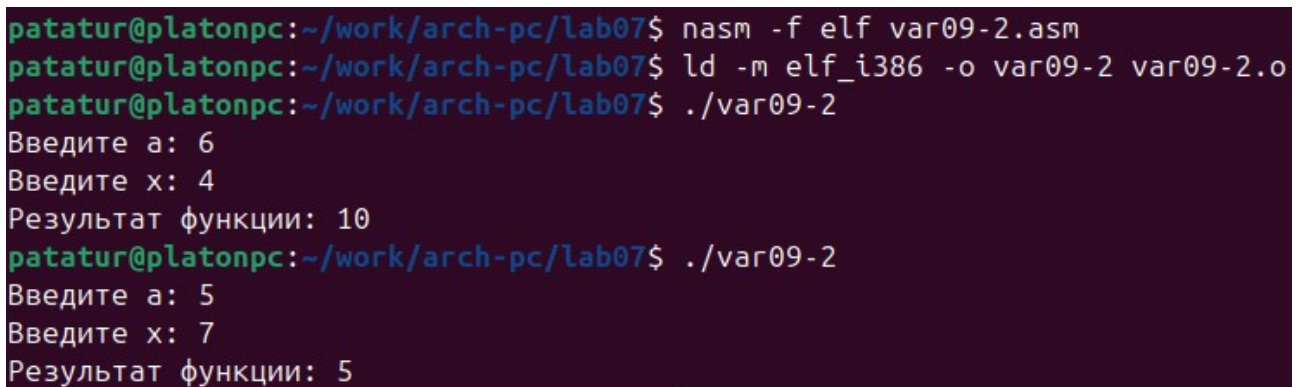
Рис 6.1



```
var09-2.asm
/home/patatur/work/arch-pc/lab07
Открыть
Сохранить

29
30 ; ----- Преобразование 'a' из символа в число
31 mov eax,a
32 call atoi ; Вызов подпрограммы перевода символа в число
33 mov [a],eax ; запись преобразованного числа в 'a'
34
35 ; ----- Преобразование 'x' из символа в число
36 mov eax,x
37 call atoi ; Вызов подпрограммы перевода символа в число
38 mov [x],eax ; запись преобразованного числа в 'x'
39
40 mov ecx,[x]
41 cmp ecx,[a]
42 jle sum
43 jmp fin
44
45
46 sum:
47 mov eax,[a]
48 mov ecx,[x]
49 add eax,ecx
50 mov [a],eax
51
52 fin:
53 mov eax, msg3
54 call sprint ; Вывод сообщения 'Результат функции: '
55 mov eax,[a]
56 call iprintLF ; Вывод 'max(A,B,C)'
57 call quit ; Выход
```

Рис 6.2



```
patatur@platonpc:~/work/arch-pc/lab07$ nasm -f elf var09-2.asm
patatur@platonpc:~/work/arch-pc/lab07$ ld -m elf_i386 -o var09-2 var09-2.o
patatur@platonpc:~/work/arch-pc/lab07$ ./var09-2
Введите a: 6
Введите x: 4
Результат функции: 10
patatur@platonpc:~/work/arch-pc/lab07$ ./var09-2
Введите a: 5
Введите x: 7
Результат функции: 5
```

Рис 6.3

- 1) Создал файл var09-2.asm открыл его и написал текст программы (см Рис 6.1 и Рис 6.2)
 - 2) Создал исполняемый файл и проверил его работу для значений x и a из 7.6 под именем варианта 9 (см Рис 6.3)
-

Вывод

В ходе выполнения лабораторной работы я изучил команды условного и безусловного переходов. Приобрел навыки написания программ с использованием переходов. Ознакомился с назначением и структурой файла листинга.