

Отчёт по лабораторной работе 7

Архитектура компьютера

Душаев Азимбек Юсуфович НКАбд-02-23

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	20

Список иллюстраций

2.1	Программа lab7-1.asm	7
2.2	Запуск программы lab7-1.asm	8
2.3	Программа lab7-1.asm	9
2.4	Запуск программы lab7-1.asm	10
2.5	Программа lab7-1.asm	11
2.6	Запуск программы lab7-1.asm	12
2.7	Программа lab7-2.asm	13
2.8	Запуск программы lab7-2.asm	14
2.9	Файл листинга lab7-2	15
2.10	Ошибка трансляции lab7-2	16
2.11	Файл листинга с ошибкой lab7-2	17
2.12	Программа lab7-p1.asm	18
2.13	Запуск программы lab7-p1.asm	19

Список таблиц

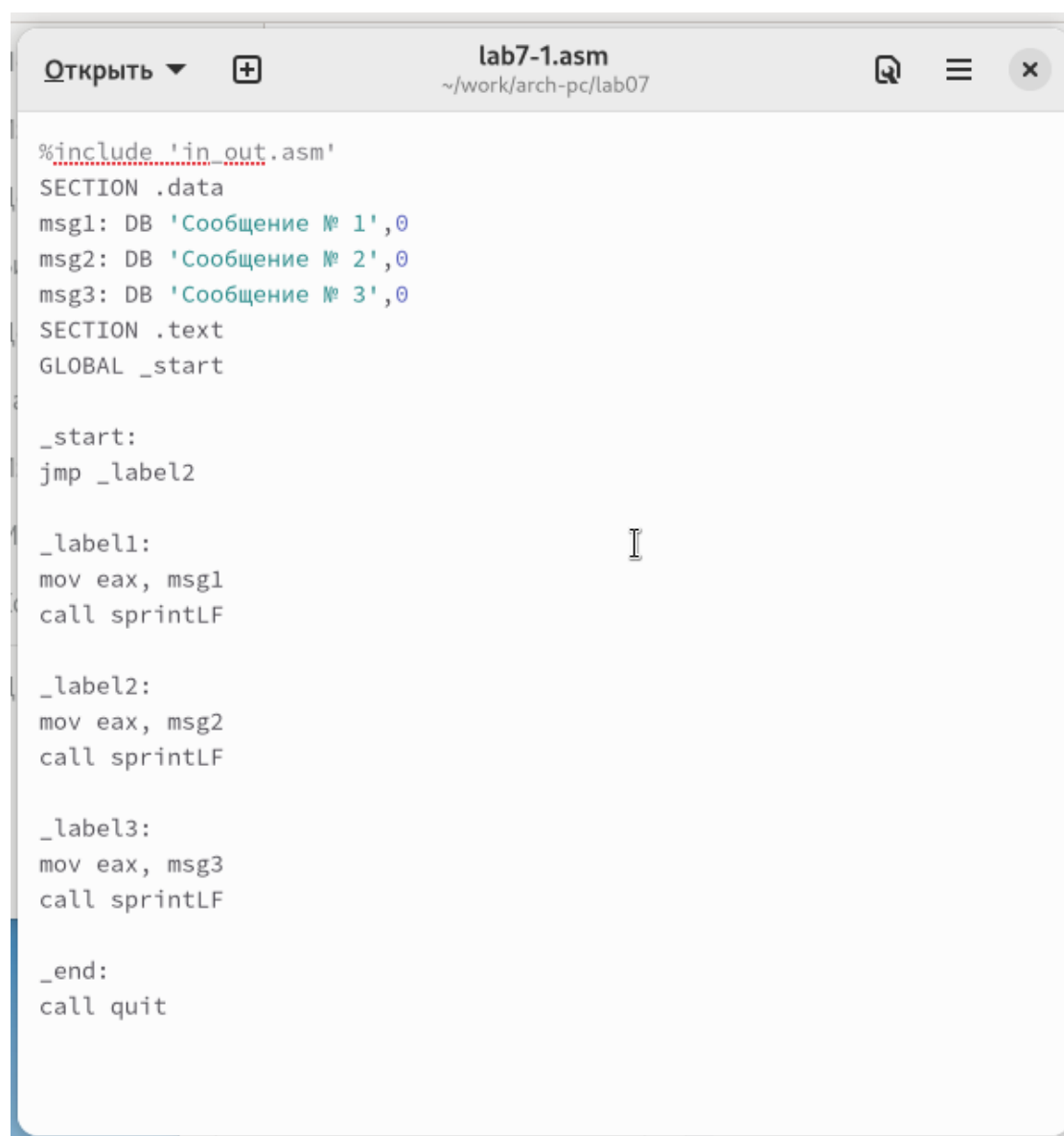
1 Цель работы

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

2 Выполнение лабораторной работы

1. Создал каталог для программам лабораторной работы № 7 и файл lab7-1.asm
2. Инструкция `jmp` в NASM используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции `jmp`.

Написал в файл lab7-1.asm текст программы из листинга 7.1.



```
%include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start

_start:
jmp _label2

_label1:
mov eax, msg1
call sprintLF

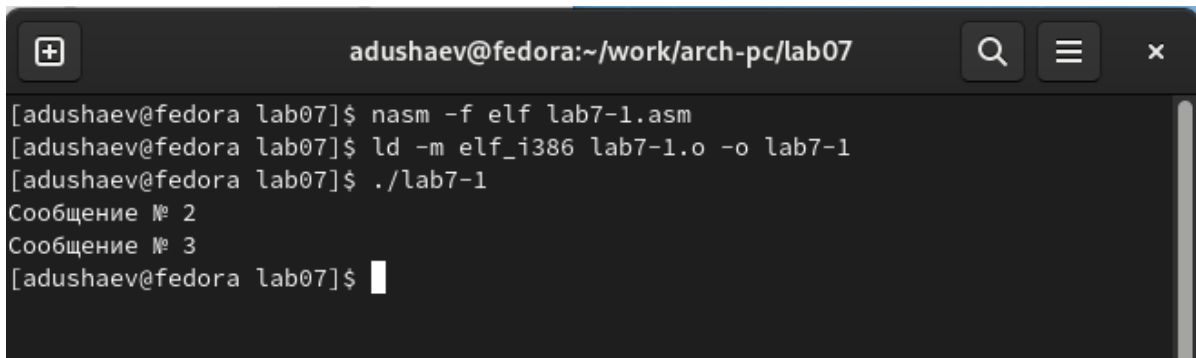
_label2:
mov eax, msg2
call sprintLF

_label3:
mov eax, msg3
call sprintLF

_end:
call quit
```

Рис. 2.1: Программа lab7-1.asm

Создал исполняемый файл и запустил его.

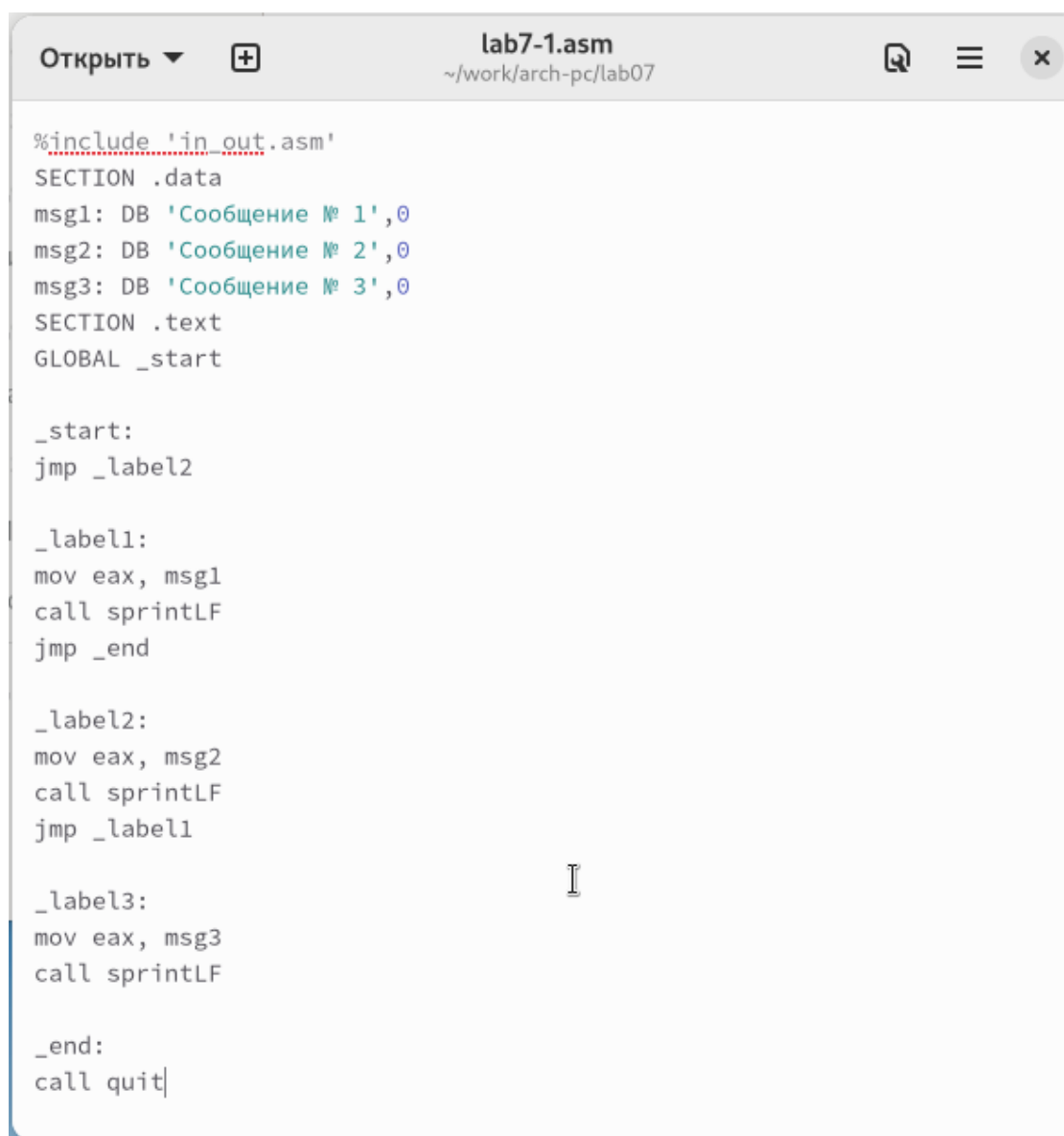
A terminal window with a dark background. The title bar shows the user 'adushaev@fedora' and the directory '~/work/arch-pc/lab07'. The terminal contains the following text:

```
[adushaev@fedora lab07]$ nasm -f elf lab7-1.asm
[adushaev@fedora lab07]$ ld -m elf_i386 lab7-1.o -o lab7-1
[adushaev@fedora lab07]$ ./lab7-1
Сообщение № 2
Сообщение № 3
[adushaev@fedora lab07]$
```

Рис. 2.2: Запуск программы lab7-1.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`).

Изменил текст программы в соответствии с листингом 7.2.



```
Открыть ▾ + lab7-1.asm ~/.work/arch-pc/lab07
%include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start

_start:
jmp _label2

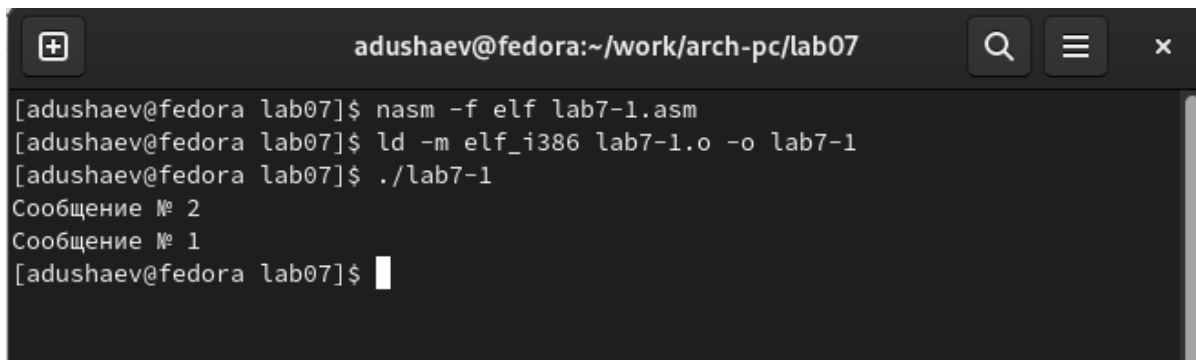
_label1:
mov eax, msg1
call sprintLF
jmp _end

_label2:
mov eax, msg2
call sprintLF
jmp _label1

_label3:
mov eax, msg3
call sprintLF

_end:
call quit
```

Рис. 2.3: Программа lab7-1.asm

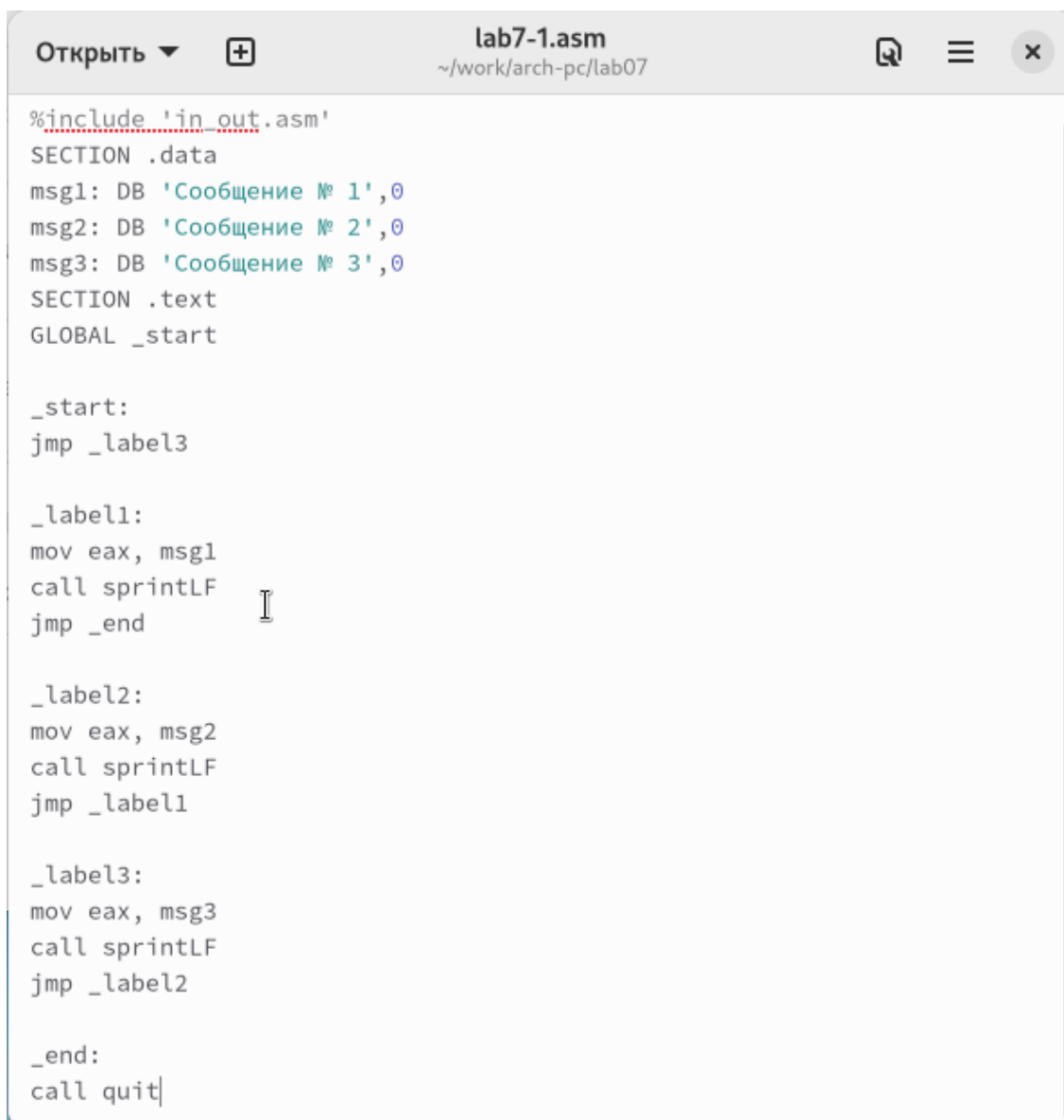
A terminal window with a dark background and light text. The title bar at the top reads 'adushaev@fedora:~/work/arch-pc/lab07'. The terminal content shows the following commands and output:

```
[adushaev@fedora lab07]$ nasm -f elf lab7-1.asm
[adushaev@fedora lab07]$ ld -m elf_i386 lab7-1.o -o lab7-1
[adushaev@fedora lab07]$ ./lab7-1
Сообщение № 2
Сообщение № 1
[adushaev@fedora lab07]$
```

Рис. 2.4: Запуск программы lab7-1.asm

Изменил текст программы, изменив инструкции `jmp`, чтобы вывод программы был следующим:

Сообщение № 3
Сообщение № 2
Сообщение № 1



```
Открыть ▾ + lab7-1.asm ~\work\arch-pc\lab07
%include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start

_start:
jmp _label3

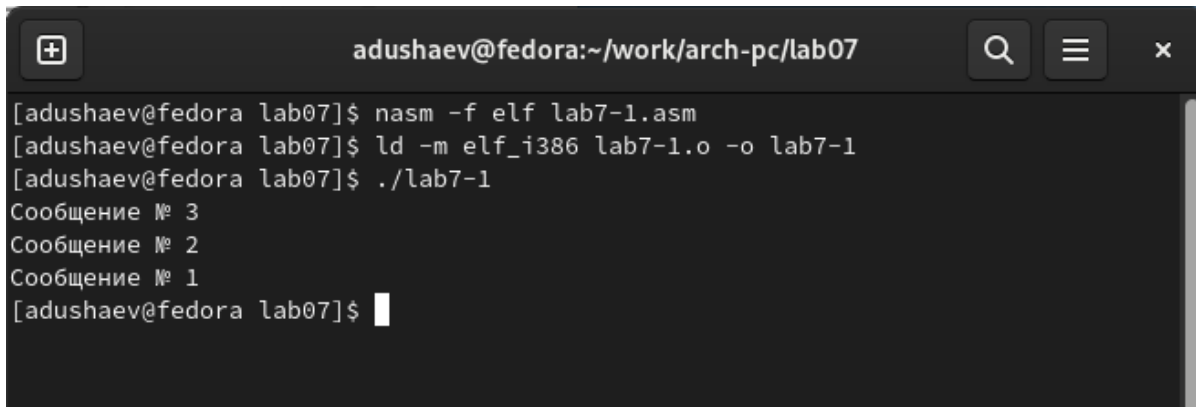
_label1:
mov eax, msg1
call sprintf
jmp _end

_label2:
mov eax, msg2
call sprintf
jmp _label1

_label3:
mov eax, msg3
call sprintf
jmp _label2

_end:
call quit
```

Рис. 2.5: Программа lab7-1.asm

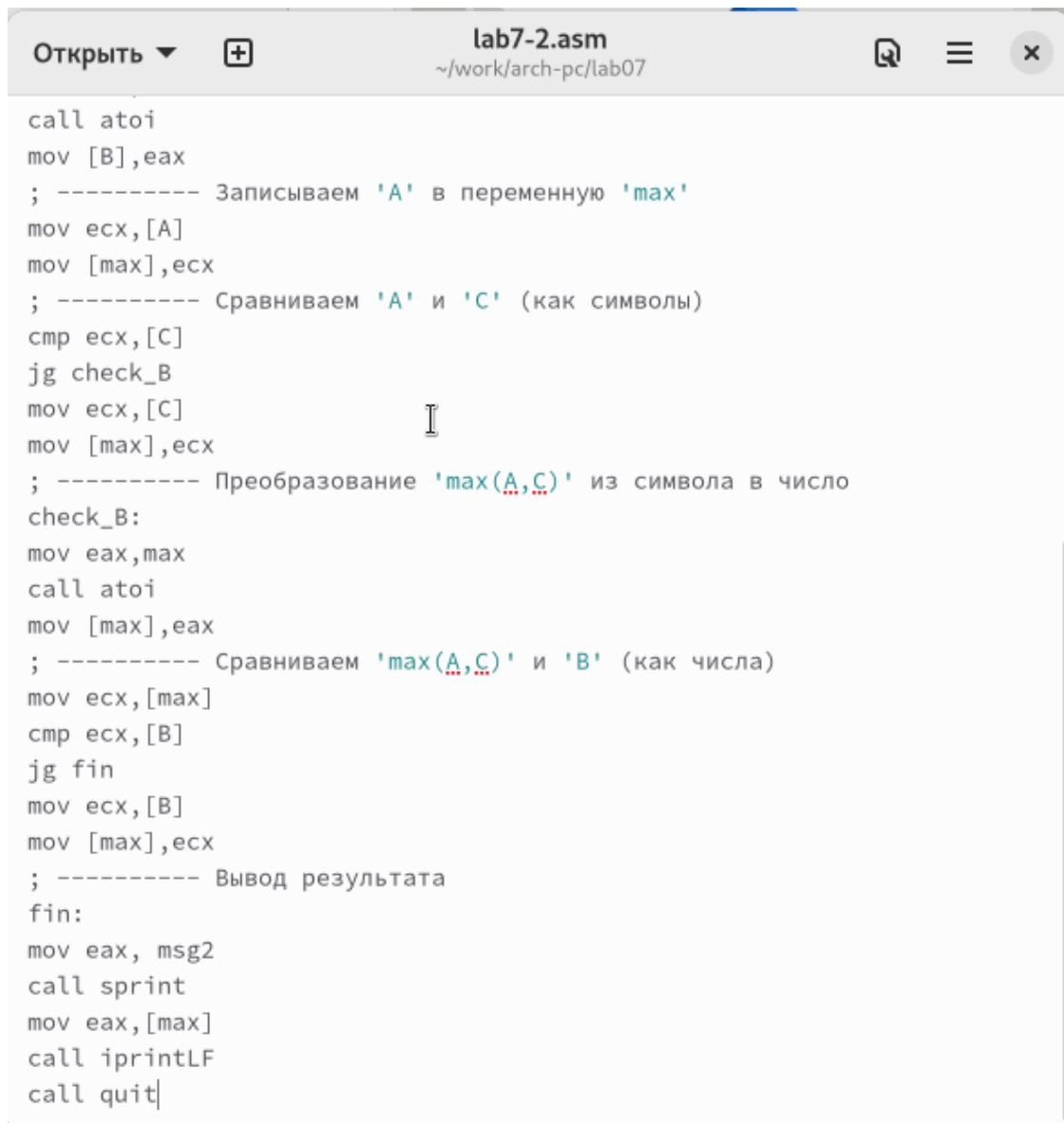
A terminal window with a dark background. The title bar shows the user 'adushaev@fedora' and the directory '~/work/arch-pc/lab07'. The terminal contains the following text:

```
[adushaev@fedora lab07]$ nasm -f elf lab7-1.asm
[adushaev@fedora lab07]$ ld -m elf_i386 lab7-1.o -o lab7-1
[adushaev@fedora lab07]$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
[adushaev@fedora lab07]$
```

Рис. 2.6: Запуск программы lab7-1.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: A, B и C. Значения для A и C задаются в программе, значение B вводится с клавиатуры.

Создал исполняемый файл и проверил его работу для разных значений B.



```
call atoi
mov [B],eax
; ----- Записываем 'A' в переменную 'max'
mov ecx,[A]
mov [max],ecx
; ----- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[C]
jg check_B
mov ecx,[C]
mov [max],ecx
; ----- Преобразование 'max(A,C)' из символа в число
check_B:
mov eax,max
call atoi
mov [max],eax
; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
mov ecx,[max]
cmp ecx,[B]
jg fin
mov ecx,[B]
mov [max],ecx
; ----- Вывод результата
fin:
mov eax, msg2
call sprint
mov eax,[max]
call iprintLF
call quit
```

Рис. 2.7: Программа lab7-2.asm

```
[adushaev@fedora lab07]$  
[adushaev@fedora lab07]$ nasm -f elf lab7-2.asm  
[adushaev@fedora lab07]$ ld -m elf_i386 lab7-2.o -o lab7-2  
[adushaev@fedora lab07]$ ./lab7-2  
Введите B: 40  
Наибольшее число: 50  
[adushaev@fedora lab07]$ ./lab7-2  
Введите B: 30  
Наибольшее число: 50  
[adushaev@fedora lab07]$ ./lab7-2  
Введите B: 70  
Наибольшее число: 70  
[adushaev@fedora lab07]$
```

Рис. 2.8: Запуск программы lab7-2.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.

Создал файл листинга для программы из файла lab7-2.asm

```
lab7-2.asm lab7-2.lst
49 49 <1>
50 50 00000032 50 <1> push eax
51 51 00000033 B80A000000 <1> mov eax, 0AH
52 52 00000038 50 <1> push eax
53 53 00000039 89E0 <1> mov eax, esp
54 54 0000003B E8CFFFFFFF <1> call sprint
55 55 00000040 58 <1> pop eax
56 56 00000041 58 <1> pop eax
57 57 00000042 C3 <1> ret
58 58 <1>
59 59 <1> ;----- sread -----
60 60 <1> ; Функция считывания сообщения
61 61 <1> ; входные данные: mov eax,<buffer>, mov ebx,<N>
62 62 <1> sread:
63 63 00000043 53 <1> push ebx
64 64 00000044 50 <1> push eax
65 65 <1>
66 66 00000045 BB00000000 <1> mov ebx, 0
67 67 0000004A B803000000 <1> mov eax, 3
68 68 0000004F CD80 <1> int 80h
69 69 <1>
70 70 00000051 5B <1> pop ebx
71 71 00000052 59 <1> pop ecx
72 72 00000053 C3 <1> ret
73 73 <1>
74 74 <1> ;----- iprint -----
75 75 <1> ; Функция вывода на экран чисел в формате ASCII
```

Рис. 2.9: Файл листинга lab7-2

Внимательно ознакомился с его форматом и содержимым. Подробно объясню содержимое трёх строк файла листинга по выбору.

*строка 70

- 70 - номер строки
- 00000051 - адрес
- 5B - машинный код
- pop ebx - код программы

*строка 71

- 71 - номер строки

- 00000052 - адрес

- 59 - машинный код

- pop ecx - код программы

*строка 72

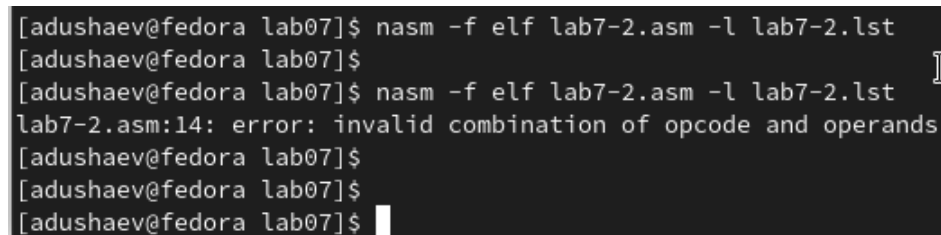
- 72 - номер строки

- 00000053 - адрес

- C3 - машинный код

- ret - код программы

Открыл файл с программой lab7-2.asm и в инструкции с двумя операндами удалил один операнд. Выполнил трансляцию с получением файла листинга.



```
[adushaev@fedora lab07]$ nasm -f elf lab7-2.asm -l lab7-2.lst
[adushaev@fedora lab07]$
[adushaev@fedora lab07]$ nasm -f elf lab7-2.asm -l lab7-2.lst
lab7-2.asm:14: error: invalid combination of opcode and operands
[adushaev@fedora lab07]$
[adushaev@fedora lab07]$
[adushaev@fedora lab07]$
```

Рис. 2.10: Ошибка трансляции lab7-2


```
lab7-2.lst
~/work/arch-pc/lab07
Стр. 1, Поз. 1

lab7-2.asm
lab7-2.lst

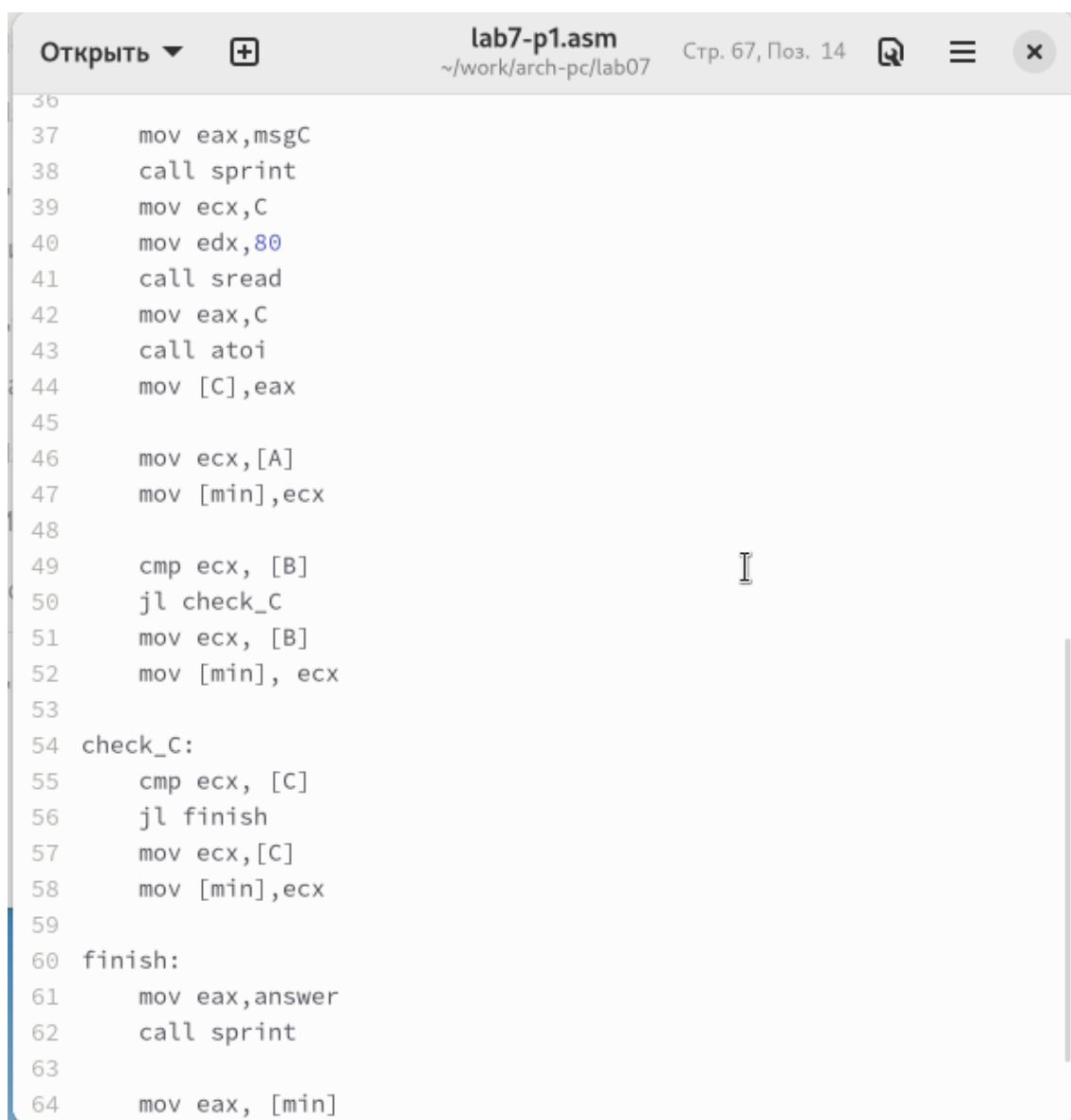
187 12 _start:
188 13 ; ----- Вывод сообщения 'Введите B: '
189 14 mov eax,
190 14 *****
error: invalid combination of opcode and operands
191 15 000000E8 E822FFFFFF call sprintf
192 16 ; ----- Ввод 'B'
193 17 000000ED B9[0A000000] mov ecx,B
194 18 000000F2 BA0A000000 mov edx,10
195 19 000000F7 E847FFFFFF call sread
196 20 ; ----- Преобразование 'B' из символа в число
197 21 000000FC B8[0A000000] mov eax,B
198 22 00000101 E896FFFFFF call atoi
199 23 00000106 A3[0A000000] mov [B],eax
200 24 ; ----- Записываем 'A' в переменную 'max'
201 25 0000010B 8B0D[35000000] mov ecx,[A]
202 26 00000111 890D[00000000] mov [max],ecx
203 27 ; ----- Сравниваем 'A' и 'C' (как символы)
```

Рис. 2.11: Файл листинга с ошибкой lab7-2

Объектный файл не смог создаться из-за ошибки. Но получился листинг, где выделено место ошибки.

5. 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 № 6. Создайте исполняемый файл и проверьте его работу

для варианта 5 - 54, 62, 87



```
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
46     mov ecx,[A]
47     mov [min],ecx
48
49     cmp ecx, [B]
50     jl check_C
51     mov ecx, [B]
52     mov [min], ecx
53
54 check_C:
55     cmp ecx, [C]
56     jl finish
57     mov ecx,[C]
58     mov [min],ecx
59
60 finish:
61     mov eax,answer
62     call sprint
63
64     mov eax, [min]
```

Рис. 2.12: Программа lab7-p1.asm

```

[adushaev@fedora lab07]$
[adushaev@fedora lab07]$ nasm -f elf lab7-p1.asm
[adushaev@fedora lab07]$ ld -m elf_i386 lab7-p1.o -o lab7-p1
[adushaev@fedora lab07]$ ./lab7-p1
Input A: 54
Input B: 62
Input C: 87
Smallest: 54
[adushaev@fedora lab07]$
[adushaev@fedora lab07]$

```

Рис. 2.13: Запуск программы lab7-p1.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 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.

для варианта 12

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

[Программа lab7-p2.asm](image/14.png){ #fig:014 width=70%, height=70% }

Подставляем $x = 1, a = 2$ и $x = 2, a = 1$

[Запуск программы lab7-p2.asm](image/15.png){ #fig:015 width=70%, height=70%

}

3 Выводы

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