

Лабораторная работа №7.

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

Митрофанов Тимур Александрович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Задание для самостоятельной работы	14
4	Выводы	23

Список иллюстраций

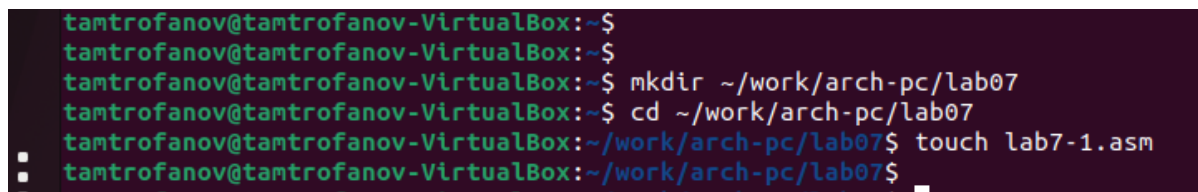
2.1	Создание каталога <i>lab07</i> и создания в нём файла <i>lab7-1.asm</i>	5
2.2	Добавление кода из листинга 7.1 в файл <i>lab7-1.asm</i>	6
2.3	Создание исполняемого файла и его запуск	6
2.4	Изменённый файл <i>lab7-1.asm</i>	7
2.5	Создание исполняемого файла и его запуск	7
2.6	Изменённый файл <i>lab7-1.asm</i>	8
2.7	Создание исполняемого файла и его запуск	8
2.8	Создание файла <i>lab7-2.asm</i>	9
2.9	Внесённый код в файл <i>lab7-2.asm</i>	10
2.10	Компеляция файла и проверка его работы	11
2.11	Создание файла листинга и проверка его наличия	11
2.12	Открытие файла при помощи <i>mcedit</i>	11
2.13	Три выбранные строки из файла	12
2.14	Изменение в файле <i>lab7-2.asm</i>	13
2.15	соответствующие изменения в созданном файле листинга	13
3.1	Создание файла <i>lav07-3.asm</i>	14
3.2	Изменённый код в файле <i>lav07-3.asm</i>	15
3.3	Создание исполняемого файла и проверка его работы	16
3.4	Создание файла <i>lav07-4.asm</i>	18
3.5	Код в файле <i>lav07-4.asm</i>	19
3.6	Создание исполняемого файла и проверка его работы	20

1 Цель работы

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

2 Выполнение лабораторной работы

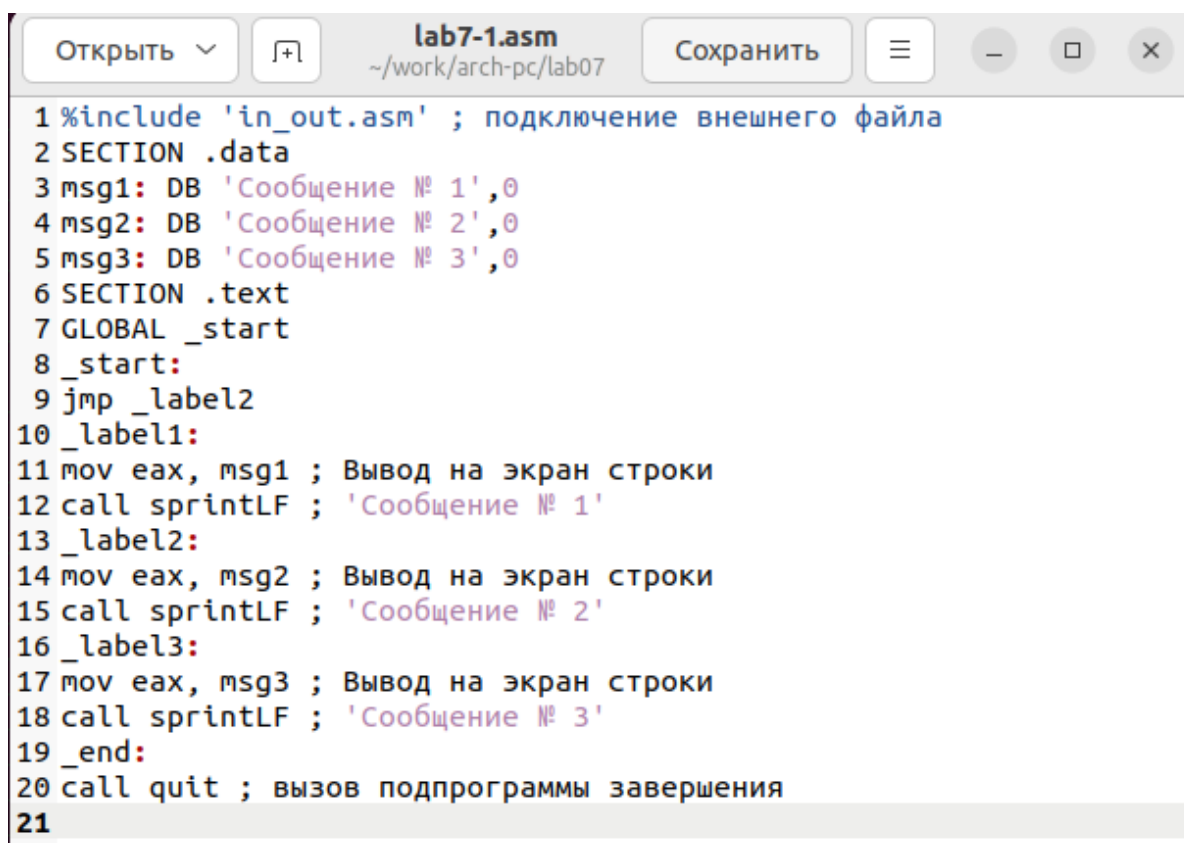
Создал подкаталог *lab07*, перешел в него и в нём создал файл *lab7-1.asm* (рис. 2.1).



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

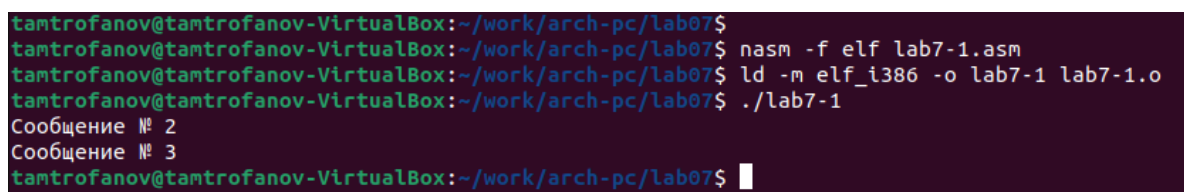
Рис. 2.1: Создание каталога *lab07* и создания в нём файла *lab7-1.asm*

Ввёл тест программы из листинга 7.1 в файл *lab7-1.asm* (рис. 2.2). Затем создал исполняемый файл и проверил его работу (рис. 2.3)



```
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
21
```

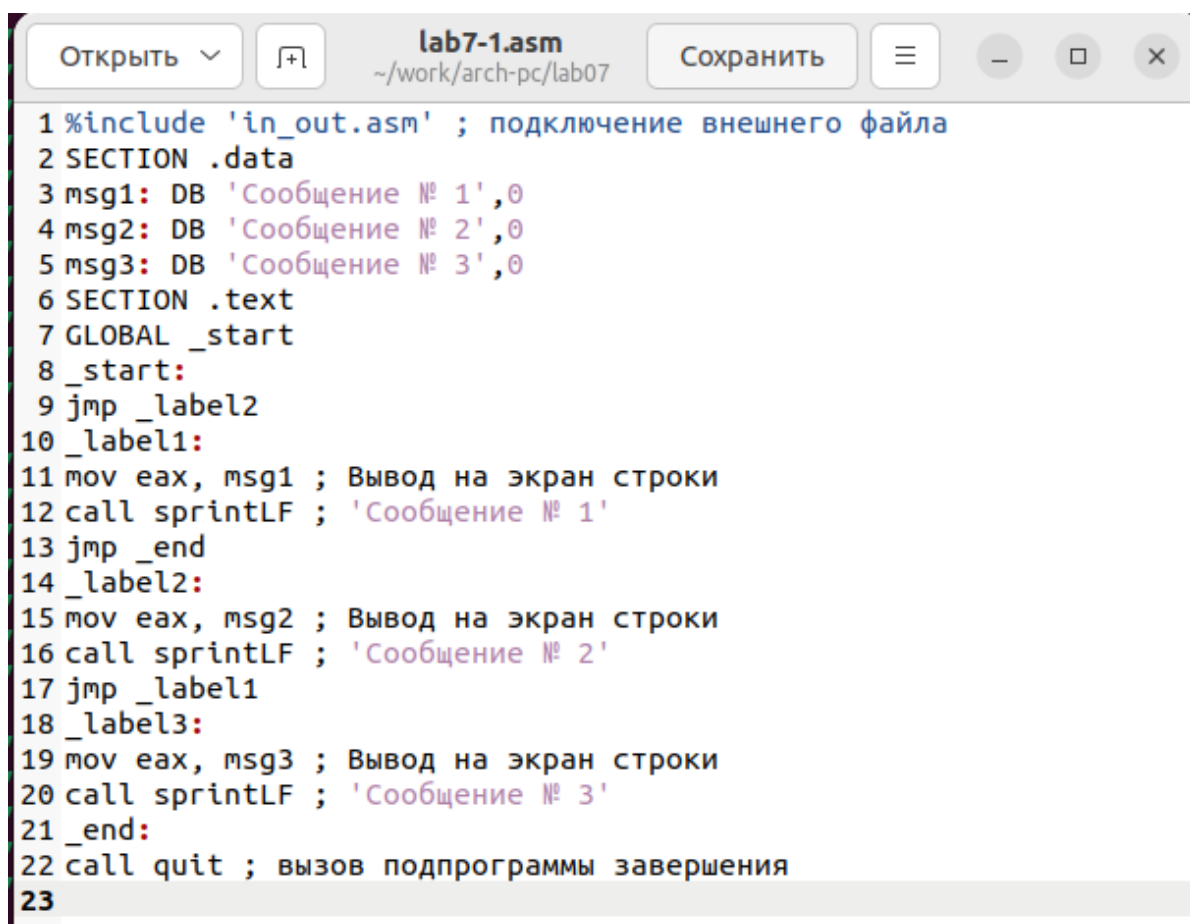
Рис. 2.2: Добавление кода из листинга 7.1 в файл *lab7-1.asm*



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

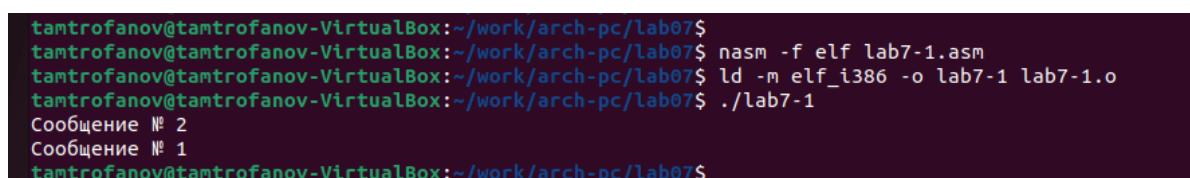
Рис. 2.3: Создание исполняемого файла и его запуск

Изменил код в файле *lab7-1.asm* в соответствии с листингом 7.2 (рис. 2.4). Повторно создал исполняемый файл и проверил его работу (рис. 2.5).



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

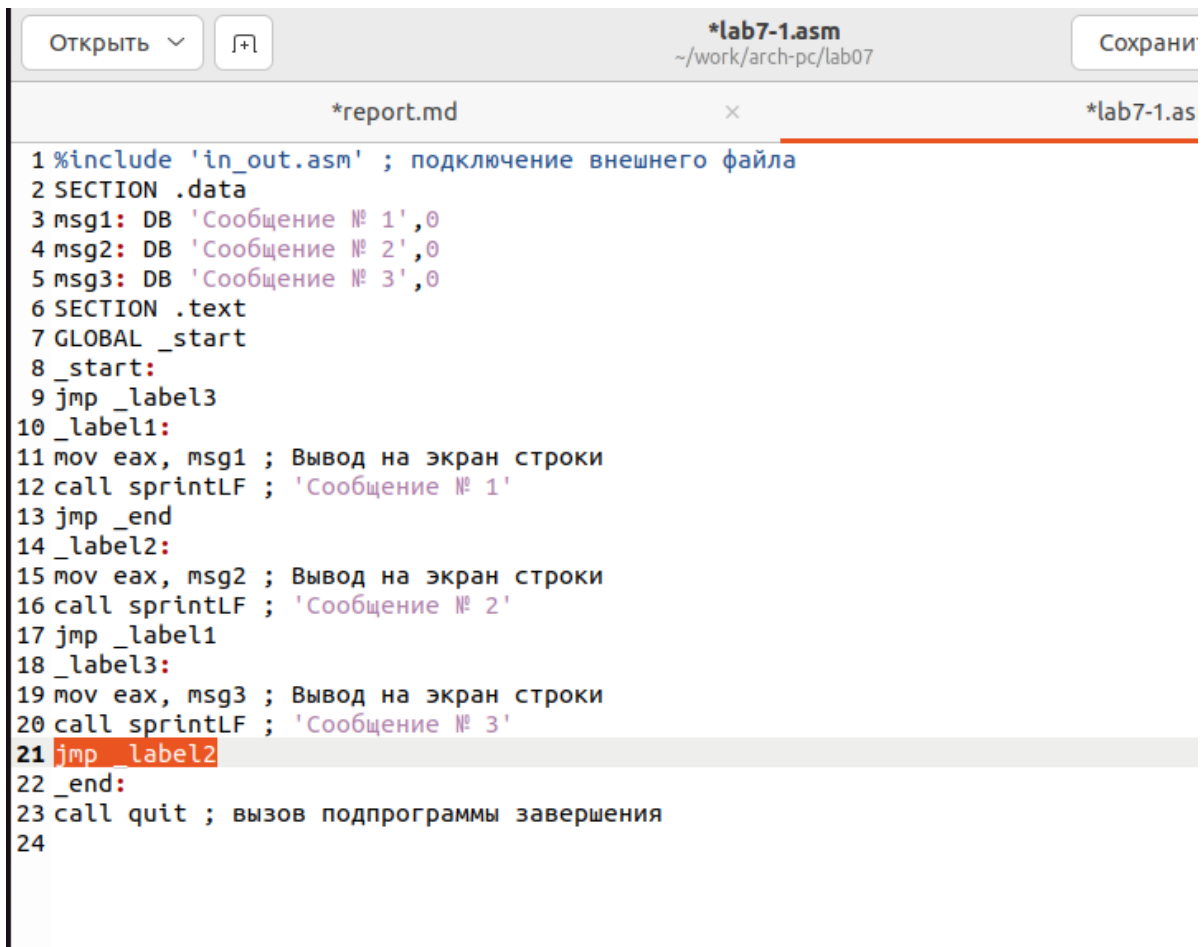
Рис. 2.4: Изменённый файл *lab7-1.asm*



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

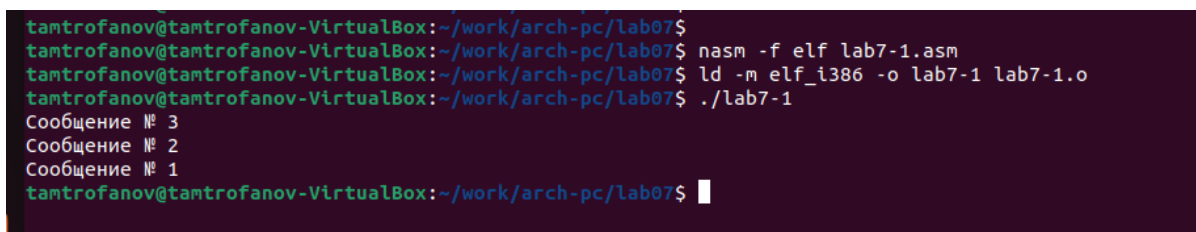
Рис. 2.5: Создание исполняемого файла и его запуск

Вновь изменил программу (рис. 2.6). Повторно создал исполняемый файл и проверил его работу (рис. 2.7).



```
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
24
```

Рис. 2.6: Изменённый файл *lab7-1.asm*




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

Рис. 2.7: Создание исполняемого файла и его запуск

Создал файл *lab7-2.asm* (рис. 2.8). Внес код из листинга 7.3 в файл *lab7-2.asm* (рис. 2.9). Затем создал исполняемый файл и проверил его для нескольких значений (рис. 2.10).


```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$  
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ touch lab7-2.asm  
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 2.8: Создание файла *lab7-2.asm*

Открыть ▾ 

lab7-2.asm
~/work/arch-pc/lab07

```
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'
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
44 fin:
45 mov eax, msg2
46 call sprint ; Вывод сообщения 'Наибольшее число: '
47 mov eax,[max]
48 call iprintLF ; Вывод 'max(A,B,C)'
49 call quit ; Выход
```

Рис. 2.9: Внесённый код в файл *lab7-2.asm*

```

tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 10
Наибольшее число: 50
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 20
Наибольшее число: 50
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 110
Наибольшее число: 110
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ █

```

Рис. 2.10: Компиляция файла и проверка его работы

Создал файл листинга для файла *lab7-2.asm* (рис. 2.11). Открыл его с помощью *mcedit* (рис. 2.12).

```

tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ ls
in_out.asm lab7-1 lab7-1.asm lab7-1.o lab7-2 lab7-2.asm lab7-2.lst lab7-2.o
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ █

```

Рис. 2.11: Создание файла листинга и проверка его наличия

```

tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ mcedit lab7-2.lst

```

Рис. 2.12: Открытие файла при помощи mcedit

Три строки выбранных мною следующее (рис. 2.13):

- 45 - строка, 00000159 адрес, B8[13000000] - машинный код, **mov eax, msg2** - исходный тест программы. В котором записывается адрес *msg2* в *EAX*.
- 46 - строка, 0000015E адрес, E8ACFEFFFF - машинный код, **call sprint** - исходный тест программы. В котором вызывается функция *sprint*. Она в свою очередь выводит сообщение содержащееся в переменной *msg2*
- 47 - строка, 00000163 адрес, A1[00000000] - машинный код, **mov eax,[max]** - исходный тест программы. В котором записывается адрес переменной *[max]* в *EAX*.

```
45 00000159 B8[13000000]      mov eax, msg2
46 0000015E E8ACFEFFFF      call sprint ; Вывод сообщения 'Наибольшее число: '
47 00000163 A1[00000000]      mov eax,[max]
```

Рис. 2.13: Три выбранные строки из файла

Убрал один операнд из строки 14 файла *lab7-2.asm* (рис. 2.14). И создал файл листинга в котором на строчке изменения появилась ошибка(рис. 2.15).

```

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
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
44 fin:
45 mov eax, msg2
46 call sprint ; Вывод сообщения 'Наибольшее число: '
47 mov eax,[max]
48 call iprintLF ; Вывод 'max(A,B,C)'
49 call quit ; Выход

```

Рис. 2.14: Изменение в файле *lab7-2.asm*

```

14                                     mov eax
14                                     error: invalid combination of opcode and operands
15 000000F8 F822FFFFFF call sprint

```

Рис. 2.15: соответствующие изменения в созданном файле листинга

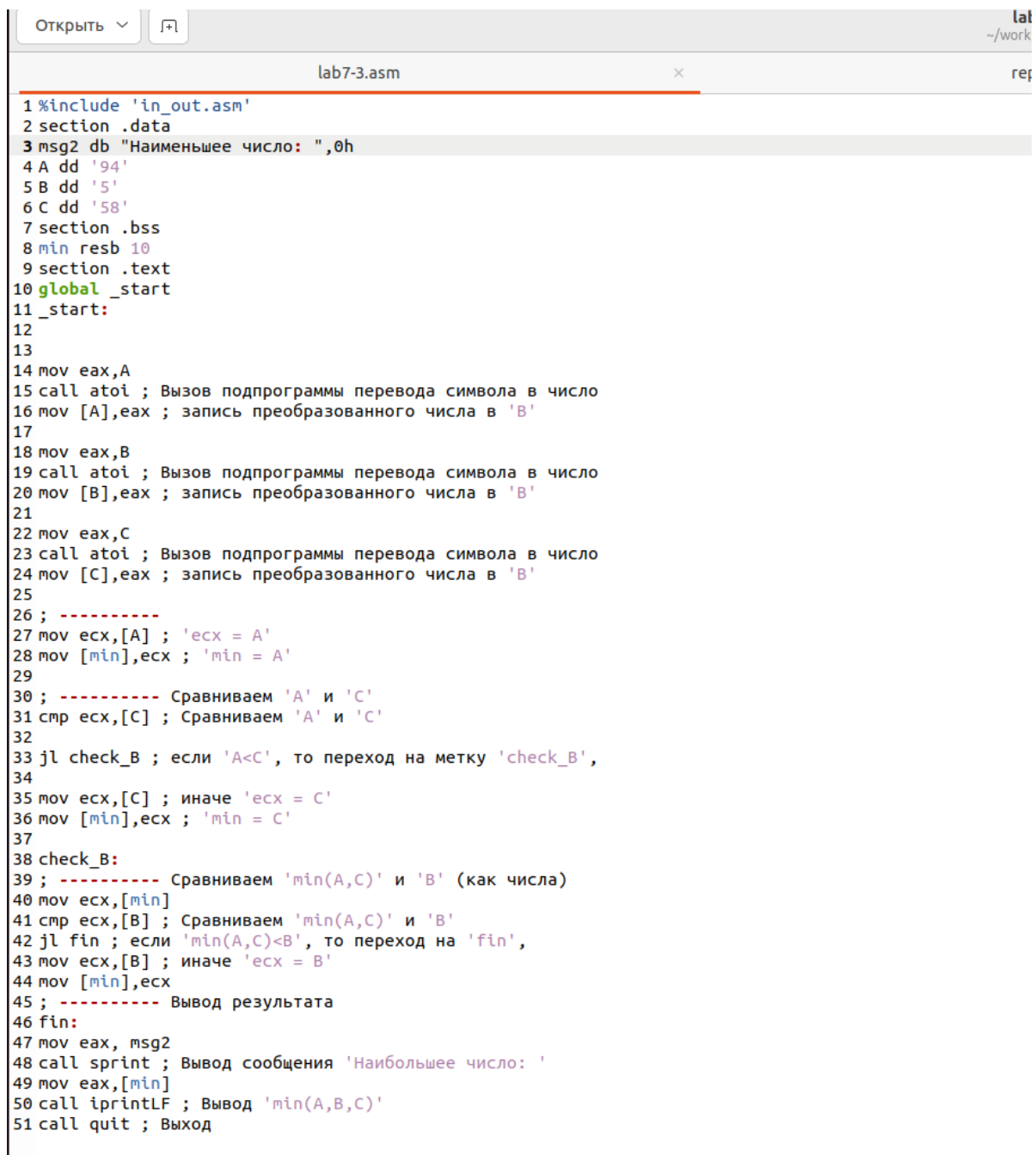
3 Задание для самостоятельной работы

Создаю новый файл *lav07-3.asm* (рис. 3.1). За основу программы взял предыдущий код из листинга 7.3. Внёс необходимые изменения в код (рис. 3.2). Затем скомпилировал код и проверил его работу (рис. 3.3).



```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ touch lab7-3.asm  
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 3.1: Создание файла *lav07-3.asm*



```
1 %include 'in_out.asm'
2 section .data
3 msg2 db "Наименьшее число: ",0h
4 A dd '94'
5 B dd '5'
6 C dd '58'
7 section .bss
8 min resb 10
9 section .text
10 global _start
11 _start:
12
13
14 mov eax,A
15 call atoi ; Вызов подпрограммы перевода символа в число
16 mov [A],eax ; запись преобразованного числа в 'B'
17
18 mov eax,B
19 call atoi ; Вызов подпрограммы перевода символа в число
20 mov [B],eax ; запись преобразованного числа в 'B'
21
22 mov eax,C
23 call atoi ; Вызов подпрограммы перевода символа в число
24 mov [C],eax ; запись преобразованного числа в 'B'
25
26 ; -----
27 mov ecx,[A] ; 'ecx = A'
28 mov [min],ecx ; 'min = A'
29
30 ; ----- Сравниваем 'A' и 'C'
31 cmp ecx,[C] ; Сравниваем 'A' и 'C'
32
33 jl check_B ; если 'A<C', то переход на метку 'check_B',
34
35 mov ecx,[C] ; иначе 'ecx = C'
36 mov [min],ecx ; 'min = C'
37
38 check_B:
39 ; ----- Сравниваем 'min(A,C)' и 'B' (как числа)
40 mov ecx,[min]
41 cmp ecx,[B] ; Сравниваем 'min(A,C)' и 'B'
42 jl fin ; если 'min(A,C)<B', то переход на 'fin',
43 mov ecx,[B] ; иначе 'ecx = B'
44 mov [min],ecx
45 ; ----- Вывод результата
46 fin:
47 mov eax, msg2
48 call sprint ; Вывод сообщения 'Наибольшее число: '
49 mov eax,[min]
50 call iprintLF ; Вывод 'min(A,B,C)'
51 call quit ; Выход
```

Рис. 3.2: Изменённый код в файле *lav07-3.asm*

```

tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ ./lab7-3
Наименьшее число: 5
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$

```

Рис. 3.3: Создание исполняемого файла и проверка его работы

КОД ПРОГРАММЫ

```

%include 'in_out.asm'

section .data
msg2 db "Наименьшее число: ",0h
A dd '94'
B dd '5'
C dd '58'

section .bss
min resb 10

section .text
global _start
_start:

mov eax,A
call atoi ; Вызов подпрограммы перевода символа в число
mov [A],eax ; запись преобразованного числа в 'B'

mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'

mov eax,C

```



```

call atoi ; Вызов подпрограммы перевода символа в число
mov [C],eax ; запись преобразованного числа в 'B'

; -----
mov ecx,[A] ; 'ecx = A'
mov [min],ecx ; 'min = A'

; ----- Сравниваем 'A' и 'C'
cmp ecx,[C] ; Сравниваем 'A' и 'C'

jl check_B ; если 'A<C', то переход на метку 'check_B',

mov ecx,[C] ; иначе 'ecx = C'
mov [min],ecx ; 'min = C'

check_B:
; ----- Сравниваем 'min(A,C)' и 'B' (как числа)
mov ecx,[min]
cmp ecx,[B] ; Сравниваем 'min(A,C)' и 'B'

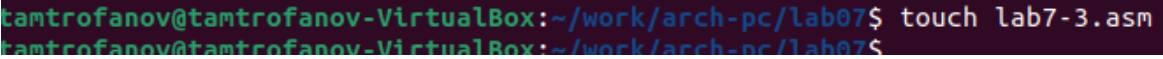
jl fin

mov ecx,[B] ; иначе 'ecx = B'
mov [min],ecx
; ----- Вывод результата
fin:
mov eax, msg2
call sprint ; Вывод сообщения 'Наибольшее число: '
mov eax,[min]

```

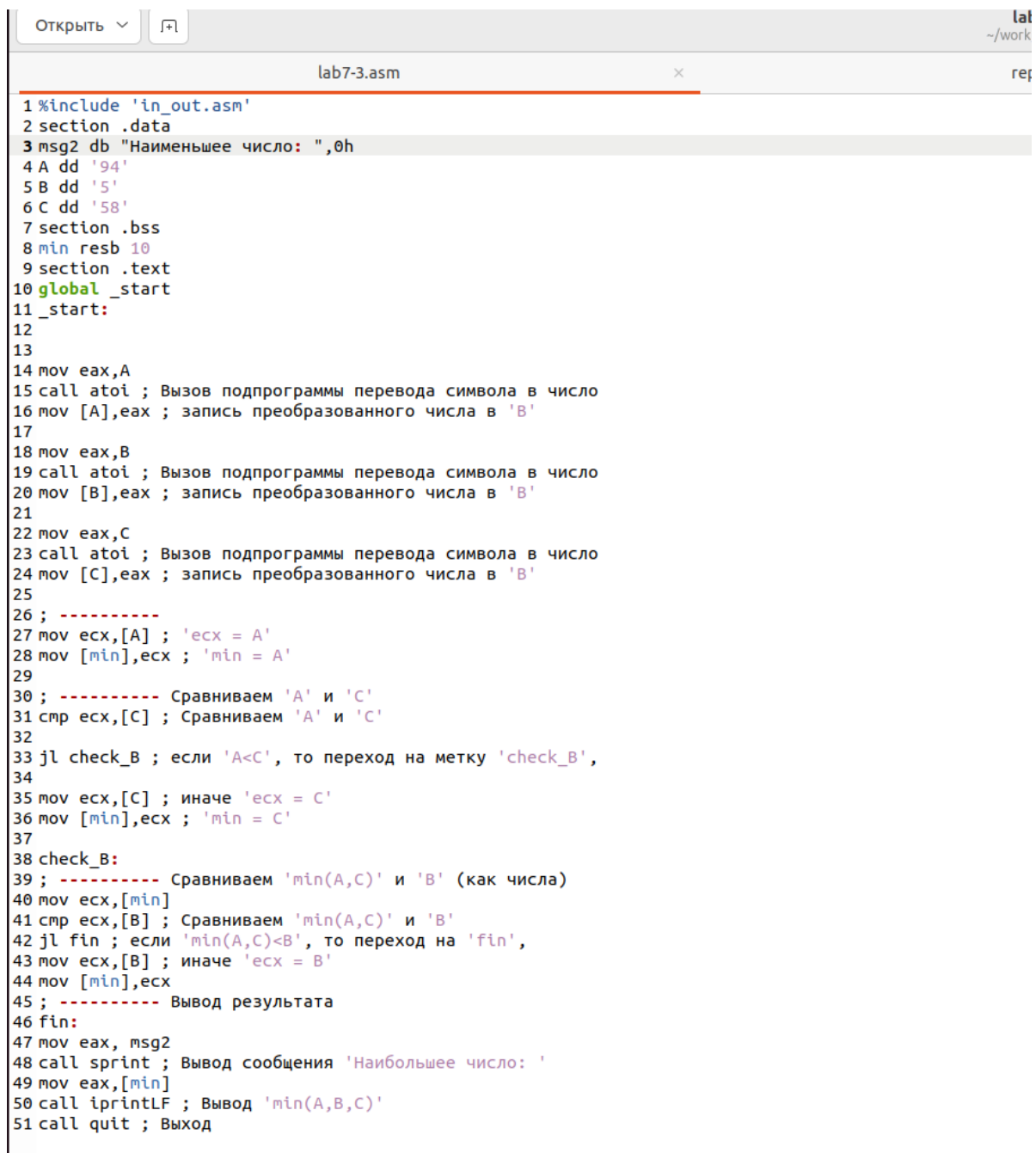
```
call iprintLF ; Вывод 'min(A,B,C)'  
call quit ; Выход
```

Создаю новый файл *lav07-4.asm* (рис. 3.4). Внёс код в файл (рис. 3.5). Затем скомпилировал код и проверил его работу (рис. 3.6).

A terminal window with a dark background and light green text. The prompt is 'tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07\$'. The command 'touch lab7-3.asm' has been entered and executed. The next line shows the prompt again: 'tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07\$'.

```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ touch lab7-3.asm  
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 3.4: Создание файла *lav07-4.asm*



```
1 %include 'in_out.asm'
2 section .data
3 msg2 db "Наименьшее число: ",0h
4 A dd '94'
5 B dd '5'
6 C dd '58'
7 section .bss
8 min resb 10
9 section .text
10 global _start
11 _start:
12
13
14 mov eax,A
15 call atoi ; Вызов подпрограммы перевода символа в число
16 mov [A],eax ; запись преобразованного числа в 'B'
17
18 mov eax,B
19 call atoi ; Вызов подпрограммы перевода символа в число
20 mov [B],eax ; запись преобразованного числа в 'B'
21
22 mov eax,C
23 call atoi ; Вызов подпрограммы перевода символа в число
24 mov [C],eax ; запись преобразованного числа в 'B'
25
26 ; -----
27 mov ecx,[A] ; 'ecx = A'
28 mov [min],ecx ; 'min = A'
29
30 ; ----- Сравниваем 'A' и 'C'
31 cmp ecx,[C] ; Сравниваем 'A' и 'C'
32
33 jl check_B ; если 'A<C', то переход на метку 'check_B',
34
35 mov ecx,[C] ; иначе 'ecx = C'
36 mov [min],ecx ; 'min = C'
37
38 check_B:
39 ; ----- Сравниваем 'min(A,C)' и 'B' (как числа)
40 mov ecx,[min]
41 cmp ecx,[B] ; Сравниваем 'min(A,C)' и 'B'
42 jl fin ; если 'min(A,C)<B', то переход на 'fin',
43 mov ecx,[B] ; иначе 'ecx = B'
44 mov [min],ecx
45 ; ----- Вывод результата
46 fin:
47 mov eax, msg2
48 call sprint ; Вывод сообщения 'Наибольшее число: '
49 mov eax,[min]
50 call iprintLF ; Вывод 'min(A,B,C)'
51 call quit ; Выход
```

Рис. 3.5: Код в файле *lav07-4.asm*

```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$ ./lab7-3
Наименьшее число: 5
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 3.6: Создание исполняемого файла и проверка его работы

КОД ПРОГРАММЫ

```
%include 'in_out.asm'

section .data
msg1 DB 'Введите x: ',0h
msg2 DB 'Введите a: ',0h
msg3: DB 'Ответ: ',0h


section .bss
x: RESB 80
a: RESB 80
r: RESB 80


section .text
global _start
_start:
mov eax,msg1
call sprint


mov ecx,x
mov edx,80
call sread


mov eax,x
```

```

call atoi
mov [x],eax

mov eax,msg2
call sprint

mov ecx,a
mov edx,80
call sread

mov eax,a
call atoi
mov [a],eax

mov eax, [x]
cmp eax, 3
je x_ravno_3

mov eax, [a]
add eax, 1
jmp res

x_ravno_3:
mov eax, [x]
imul eax,3

res:
mov [r],eax

```

```
fin:
mov  eax,msg3
call sprint
mov  eax,[r]
call iprintLF
call quit
```

4 Выводы

Сегодня я изучил команды условного и безусловного переходов. Приобрёл навыки написания программ с использованием переходов. Ознакомился с значением и структурой файла листинга.