

# **Отчет по лабораторной работе №7**

**Архитектура компьютеров и операционные системы**

Никита Сергеевич Кокшаров

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
2.1	Реализация переходов в NASM . . . . .	5
2.2	Изучение структуры файла листинга . . . . .	9
2.3	Задания для самостоятельной работы . . . . .	10
<b>3</b>	<b>Выводы</b>	<b>15</b>

## Список иллюстраций

2.1	Создание lab7-1.asm . . . . .	5
2.2	Код в lab7-1.asm . . . . .	5
2.3	Запуск lab7-1 . . . . .	6
2.4	Изменение кода в lab7-1.asm . . . . .	6
2.5	Запуск измененного lab7-1 . . . . .	6
2.6	Код в lab7-1.asm, написанный под запрашиваемый вывод . . . . .	7
2.7	Запуск lab7-1 с получением запрошенного вывода . . . . .	7
2.8	Код в lab7-2.asm . . . . .	8
2.9	Запуск lab7-2 . . . . .	8
2.10	Файл lab7-2.lst . . . . .	9
2.11	Указание на ошибку в lab7-2.asm . . . . .	10
2.12	Указание на ошибку в lab7-2.lst . . . . .	10
2.13	Код в lab7-3.asm . . . . .	11
2.14	Запуск lab7-3 . . . . .	11
2.15	Основная часть программы в lab7-3.lst . . . . .	12
2.16	Код в lab7-4.asm . . . . .	13
2.17	Запуск lab7-4 . . . . .	13
2.18	Основная часть программы в lab7-4.lst . . . . .	14

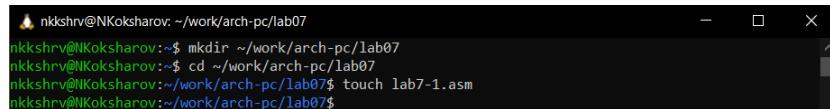
# 1 Цель работы

Целью работы является изучение команд условного и безусловного переходов, приобретение навыков написания программ с использованием переходов и знакомство с назначением и структурой файла листинга.

## 2 Выполнение лабораторной работы

### 2.1 Реализация переходов в NASM

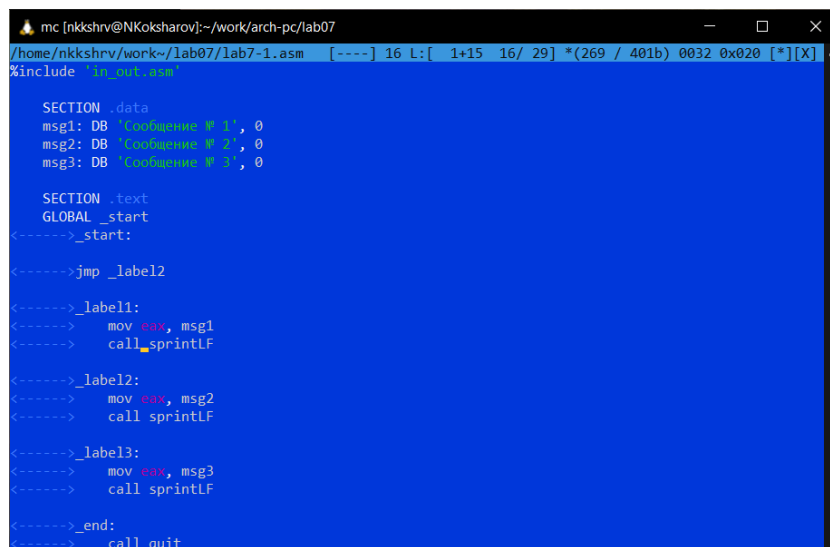
Создаю файл lab7-1.asm в новой директории lab07 (рис. 2.1).



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

Рис. 2.1: Создание lab7-1.asm

Пишу код программы из листинга 7.1 (рис. 2.2)



```
mc [nkkshrv@NKKoksharov:~/work/arch-pc/lab07]
/home/nkkshrv/work/~lab07/lab7-1.asm [----] 16 L: [ 1+15 16/ 29] *(269 / 401b) 0032 0x020 [*][X] ^
%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 sprintf

<----->_label2:
<----->  mov eax, msg2
<----->  call sprintf

<----->_label3:
<----->  mov eax, msg3
<----->  call sprintf

<----->_end:
<----->  call quit
```

Рис. 2.2: Код в lab7-1.asm

Создаю исполняемый файл и запускаю его (рис. 2.3)

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

Рис. 2.3: Запуск lab7-1

Изменяю код программы (рис. 2.4)

```
mc [nkkshrv@NKKoksharov:~/work/arch-pc/lab07
/home/nkkshrv/work~/lab07/lab7-1.asm [----] 13 L: [ 1+27 28/ 31] *(415 / 432b) 0010 0x00A [*][X]
#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 sprintf
<----->    jmp _end

<----->_label2:
<----->    mov eax, msg2
<----->    call sprintf
<----->    jmp _label1

<----->_label3:
<----->    mov eax, msg3
<----->    call sprintf

<----->_end:
<----->    call quit
```

Рис. 2.4: Изменение кода в lab7-1.asm

Создаю исполняемый файл и запускаю его (рис. 2.5)

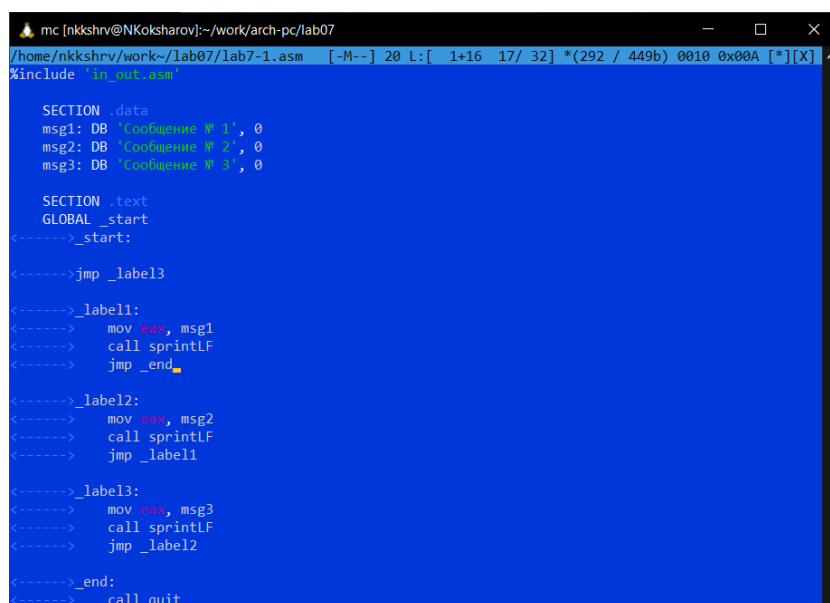
```
nkkshrv@NKKoksharov: ~/work/arch-pc/lab07
nkkshrv@NKKoksharov:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
nkkshrv@NKKoksharov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
nkkshrv@NKKoksharov:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
nkkshrv@NKKoksharov:~/work/arch-pc/lab07$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

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

Рис. 2.5: Запуск измененного lab7-1

Разница вывода наглядна на изображении.

Изменяю код программы в соответствии с листингом 7.2, делаю так, чтобы вывод был следующим: Сообщение №3, Сообщение №2, Сообщение №1 (рис. 2.6)



```
mc [nkkshrv@NKKoksharov]:~/work/arch-pc/lab07
/home/nkkshrv/work/~lab07/lab7-1.asm [-M--] 20 L:[ 1+16 17/ 32] *(292 / 449b) 0010 0x00A [*][X] ^
%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

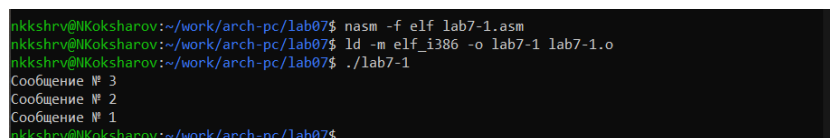
<----->_label2:
<----->mov eax, msg2
<----->call sprintf
<----->jmp _label1

<----->_label3:
<----->mov eax, msg3
<----->call sprintf
<----->jmp _label2

<----->_end:
<----->call quit
```

Рис. 2.6: Код в lab7-1.asm, написанный под запрашиваемый вывод

Создаю исполняемый файл и запускаю его (рис. 2.7)



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

Рис. 2.7: Запуск lab7-1 с получением запрошенного вывода

Создаю lab7-2.asm, пишу туда код из листинга 7.3 (рис. 2.8)

```
mc [nkshrv@NKoksharov:~/work/arch-pc/lab07]
/home/nkshrv/work/~lab07/lab7-2.asm [----] 0 L: [ 1+49 50/ 50] *(672 / 672b) <EOF> [*][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:

    mov eax, msg1
    call sprint

    mov ecx, B
    mov edx, 10
    call sread

    mov eax, B
    call atoi
    mov [B], eax

    mov ecx, [C]
    jg check_B
    mov ecx, [C]
    mov [max], ecx

check_B:
    mov eax, max
    call atoi
    mov [max], eax

    mov ecx, [max]
    cmp ecx, [B]
    jg fin
    mov ecx, [B]
    mov [max], ecx

fin:
    mov eax, msg2
    call sprint
    mov eax, [max]
    call iprintlF
    call quit

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Рис. 2.8: Код в lab7-2.asm

Создаю исполняемый файл и запускаю его, проверяю для разных значений B (рис. 2.9)

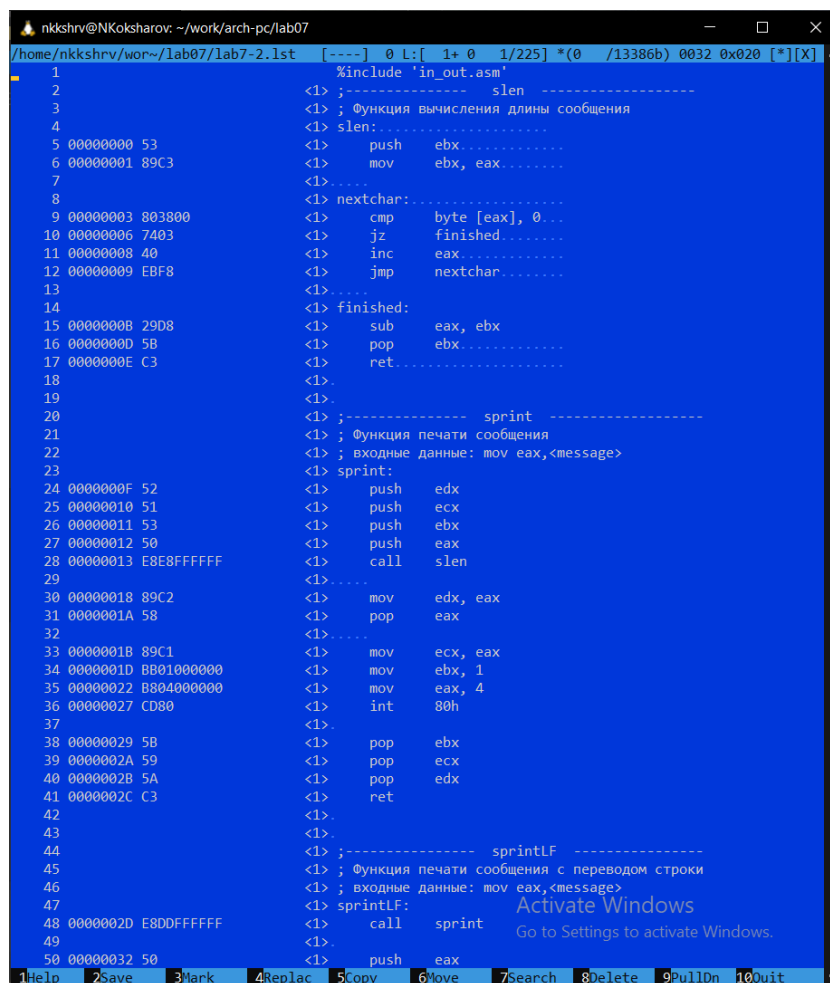
```
nkshrv@NKoksharov: ~/work/arch-pc/lab07
nkshrv@NKoksharov:~/work/arch-pc/lab07$ mc
nkshrv@NKoksharov:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
nkshrv@NKoksharov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
nkshrv@NKoksharov:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 10
Наибольшее число: 50
nkshrv@NKoksharov:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 60
Наибольшее число: 60
nkshrv@NKoksharov:~/work/arch-pc/lab07$
```

Рис. 2.9: Запуск lab7-2



## 2.2 Изучение структуры файла листинга

Создаю файл листинга и открываю его (рис. 2.10)



```
1 %include 'in_out.asm'
2 ;----- slen -----
3 ; Функция вычисления длины сообщения
4 slen:
5     push    ebx
6     mov     ebx, eax
7     nextchar:
8     cmp     byte [eax], 0
9     jz      finished
10    inc     eax
11    jmp     nextchar
12 finished:
13    sub     eax, ebx
14    pop     ebx
15    ret
16
17 ;----- sprint -----
18 ; Функция печати сообщения
19 ; входные данные: mov eax, <message>
20 sprint:
21    push    edx
22    push    ecx
23    push    ebx
24    push    eax
25    call    slen
26    mov     edx, eax
27    pop     eax
28    mov     ecx, eax
29    mov     ebx, 1
30    mov     eax, 4
31    int     80h
32    pop     ebx
33    pop     ecx
34    pop     edx
35    ret
36
37 ;----- sprintf -----
38 ; Функция печати сообщения с переводом строки
39 ; входные данные: mov eax, <message>
40 sprintf:
41    call    sprint
42    push    eax
```

Рис. 2.10: Файл lab7-2.lst

Строка 17 (основной программы): `mov eax, msg1` - в регистр `eax` передается содержимое переменной `msg1`

Строка 29 (основной программы): `jg check_B` - условный переход к `check_B`, если `ecx > [C]`

Строка 40 (основной программы): `jf fin` - условный переход к `fin`, если `ecx > [B]`

При удалении в 50-й строчки одного из операндов и создания листинга измененного файла терминал указывает на ошибку в коде (рис. 2.11)

```
nkkshrv@Nkoksharov:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:47: error: invalid combination of opcode and operands
nkkshrv@Nkoksharov:~/work/arch-pc/lab07$ ls
```

Рис. 2.11: Указание на ошибку в lab7-2.asm

В файле листинга также указана ошибка (рис. 2.12)

```
44                                     fin:
45 0000014D B8[13000000]             mov eax, msg2
46 00000152 E8B8FEFFFF             call sprint
47                                     mov [max]
47                                     error: invalid combination of opcode and operands
48 00000157 E82AFFFFFF             call iprintLF
49 0000015C E87AFFFFFF             call quit
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Рис. 2.12: Указание на ошибку в lab7-2.lst

## 2.3 Задания для самостоятельной работы

Мой вариант: №18

Пишу код для задачи под номером 1 (рис. 2.13)

```

1 %include 'in_out.asm'
2
3 section .data
4     msg db 'Наибольшее число: ', 0h
5     A dd '83'
6     B dd '73'
7     C dd '30'
8
9 section .bss
10     max resb 10
11
12 section .text
13     global _start
14 _start:
15
16     mov eax, B
17     call atoi
18     mov [B], eax
19
20     mov ecx, [A]
21     mov [max], ecx
22
23     cmp ecx, [C]
24     jg check_B
25     mov ecx, [C]
26     mov [max], ecx
27
28 check_B:
29     mov eax, max
30     call atoi
31     mov [max], eax
32
33     mov ecx, [max]
34     cmp ecx, [B]
35     jg fin
36     mov ecx, [B]
37     mov [max], ecx
38
39 fin:
40     mov eax, msg
41     call sprint
42     mov eax, [max]
43     call iprintLF
44     call quit

```

Рис. 2.13: Код в lab7-3.asm

Создаю исполняемый файл и запускаю его (рис. 2.14)

```

nkkshrv@Nkoksharov:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
nkkshrv@Nkoksharov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
nkkshrv@Nkoksharov:~/work/arch-pc/lab07$ ./lab7-3
Наибольшее число: 83
nkkshrv@Nkoksharov:~/work/arch-pc/lab07$

```

Рис. 2.14: Запуск lab7-3

Результат верный.

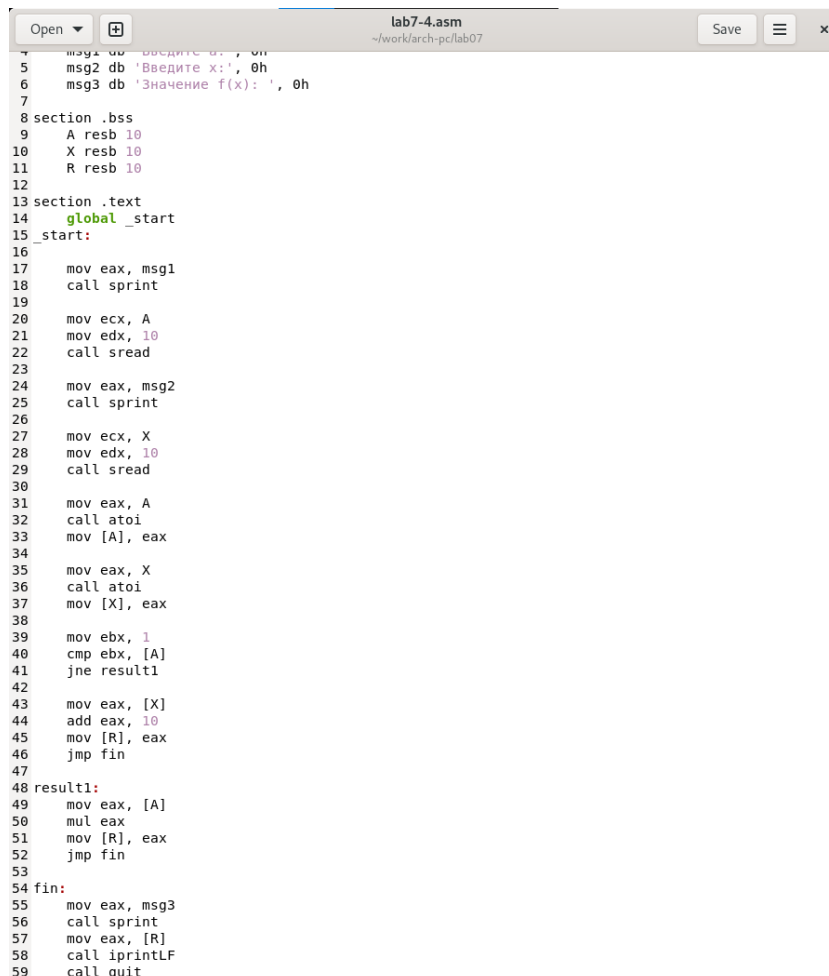
```

173 3
174 4 00000000 D09D00B0D0B8D0B1D0- section .data
175 4 00000009 BED0BB018CD188D0BE- msg db 'Наибольшее число: ', 0h
176 4 00000012 D0B520D187D0B8D181-
177 4 0000001B D0BB00BE3A2000
178 5 00000022 38330000 A dd '83'
179 6 00000026 37330000 B dd '73'
180 7 0000002A 33300000 C dd '30'
181 8
182 9 section .bss
183 10 00000000 <res Ah> max resb 10
184 11
185 12 section .text
186 13 global _start
187 14 _start:
188 15
189 16 000000E8 B8[26000000] mov eax, B
190 17 000000ED E8AAFFFFFF call atoi
191 18 000000F2 A3[26000000] mov [B], eax
192 19
193 20 000000F7 8B0D[22000000] mov ecx, [A]
194 21 000000FD 890D[00000000] mov [max], ecx
195 22
196 23 00000103 3B0D[2A000000] cmp ecx, [C]
197 24 00000109 7F0C jg check_B
198 25 0000010B 8B0D[2A000000] mov ecx, [C]
199 26 00000111 890D[00000000] mov [max], ecx
200 27
201 28 check_B:
202 29 00000117 B8[00000000] mov eax, max
203 30 0000011C E87BFFFFFF call atoi
204 31 00000121 A3[00000000] mov [max], eax
205 32
206 33 00000126 8B0D[00000000] mov ecx, [max]
207 34 0000012C 3B0D[26000000] cmp ecx, [B]
208 35 00000132 7F0C jg fin
209 36 00000134 8B0D[26000000] mov ecx, [B]
210 37 0000013A 890D[00000000] mov [max], ecx
211 38
212 39 fin:
213 40 00000140 B8[00000000] mov eax, msg
214 41 00000145 E8C5FFFFFF call sprint
215 42 0000014A A1[00000000] mov eax, [max]
216 43 0000014F E832FFFFFF call iprintLF
217 44 00000154 E882FFFFFF call quit

```

Рис. 2.15: Основная часть программы в lab7-3.lst

Пишу код для задачи под номером 2 (рис. 2.16)

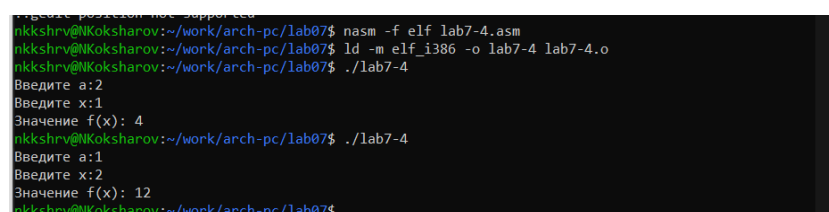


```
lab7-4.asm
~/work/arch-pc/lab07

4 msg1 db "Введите a: ", 0h
5 msg2 db "Введите x: ", 0h
6 msg3 db "Значение f(x): ", 0h
7
8 section .bss
9     A resb 10
10    X resb 10
11    R resb 10
12
13 section .text
14     global _start
15 _start:
16
17     mov eax, msg1
18     call sprint
19
20     mov ecx, A
21     mov edx, 10
22     call sread
23
24     mov eax, msg2
25     call sprint
26
27     mov ecx, X
28     mov edx, 10
29     call sread
30
31     mov eax, A
32     call atoi
33     mov [A], eax
34
35     mov eax, X
36     call atoi
37     mov [X], eax
38
39     mov ebx, 1
40     cmp ebx, [A]
41     jne result1
42
43     mov eax, [X]
44     add eax, 10
45     mov [R], eax
46     jmp fin
47
48 result1:
49     mov eax, [A]
50     mul eax
51     mov [R], eax
52     jmp fin
53
54 fin:
55     mov eax, msg3
56     call sprint
57     mov eax, [R]
58     call iprintLF
59     call quit
```

Рис. 2.16: Код в lab7-4.asm

Создаю исполняемый файл и запускаю его (рис. 2.17)



```
nkshrv@Nkoksharov:~/work/arch-pc/lab07$ nasm -f elf lab7-4.asm
nkshrv@Nkoksharov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-4 lab7-4.o
nkshrv@Nkoksharov:~/work/arch-pc/lab07$ ./lab7-4
Введите a:2
Введите x:1
Значение f(x): 4
nkshrv@Nkoksharov:~/work/arch-pc/lab07$ ./lab7-4
Введите a:1
Введите x:2
Значение f(x): 12
nkshrv@Nkoksharov:~/work/arch-pc/lab07$
```

Рис. 2.17: Запуск lab7-4

Для данных  $x_1, a_1, x_2, a_2$  результат верный.

```

182      8
183      9 00000000 <res Ah>
184     10 0000000A <res Ah>
185     11 00000014 <res Ah>
186     12
187     13
188     14
189     15
190     16
191     17 000000E8 B8[00000000]
192     18 000000ED E81DFFFFFF
193     19
194     20 000000F2 B9[00000000]
195     21 000000F7 BA0A000000
196     22 000000FC E842FFFFFF
197     23
198     24 00000101 B8[12000000]
199     25 00000106 E804FFFFFF
200     26
201     27 0000010B B9[0A000000]
202     28 00000110 BA0A000000
203     29 00000115 E829FFFFFF
204     30
205     31 0000011A B8[00000000]
206     32 0000011F E878FFFFFF
207     33 00000124 A3[00000000]
208     34
209     35 00000129 B8[0A000000]
210     36 0000012E E869FFFFFF
211     37 00000133 A3[0A000000]
212     38
213     39 00000138 BB01000000
214     40 0000013D 3B1D[00000000]
215     41 00000143 750F
216     42
217     43 00000145 A1[0A000000]
218     44 0000014A 83C00A
219     45 0000014D A3[14000000]
220     46 00000152 EB0E
221     47
222     48
223     49 00000154 A1[00000000]
224     50 00000159 F7E0
225     51 0000015B A3[14000000]
226     52 00000160 EB00
227     53
228     54
229     55 00000162 B8[24000000]
230     56 00000167 E8A3FFFFFF
231     57 0000016C A1[14000000]
232     58 00000171 E810FFFFFF
233     59 00000176 E860FFFFFF

```

```

section .bss
    A resb 10
    X resb 10
    R resb 10

section .text
    global _start
_start:

    mov eax, msg1
    call sprint

    mov ecx, A
    mov edx, 10
    call sread

    mov eax, msg2
    call sprint

    mov ecx, X
    mov edx, 10
    call sread

    mov eax, A
    call atoi
    mov [A], eax

    mov eax, X
    call atoi
    mov [X], eax

    mov ebx, 1
    cmp ebx, [A]
    jne result1

    mov eax, [X]
    add eax, 10
    mov [R], eax
    jmp fin

result1:
    mov eax, [A]
    mul eax
    mov [R], eax
    jmp fin

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

```

Activate Windows  
Go to Settings to activate Windows.

Рис. 2.18: Основная часть программы в lab7-4.lst

## **3 Выводы**

При выполнении лабораторной работы я изучил команды условного и безусловного переходов, приобрел навыки написания программ с использованием переходов и ознакомился с назначением и структурой файла листинга.