

# **Отчёт по лабораторной работе №7**

**дисциплина: Архитектура компьютера**

**Попов Даниил Георгиевич**

# Содержание

<b>1)Цель работы</b>	<b>5</b>
<b>2)Задание</b>	<b>6</b>
<b>3)Выполнение лабораторной работы</b>	<b>7</b>
3.1)Создаем каталог для программ ЛБ7, и в нем создаем файл . . . . .	7
3.2)Открываем файл в Midnight Commander и заполняем его согласно листингу 1. Создаем исполняемый файл и проверяем . . . . .	8
3.3)Изменяем код в файле согласно листингу 2 и проверяем что изменилось . . . . .	8
3.4)Меняем код так, чтобы код работал полностью верно . . . . .	9
3.5)Создаем файл и заполняем его согласно листингу 3, проверяем работу файла . . .	10
3.6)Создаем файл листинга для lab7-2.asm . . . . .	10
3.7)Открываем созданный файл через Midnight Commander . . . . .	11
3.8)Удаляем операндум из файла, создаем файл листинга и смотрим ошибки . . . . .	12
<b>4)Задания для самостоятельной работы</b>	<b>14</b>
<b>5)Выводы</b>	<b>18</b>

# Список иллюстраций

1	Создаем . . . . .	7
2	Заполняем, проверяем . . . . .	8
3	заполняем, проверяем . . . . .	8
4	заполняем, проверяем . . . . .	10
5	создаем . . . . .	10

## **Список таблиц**

# **1)Цель работы**

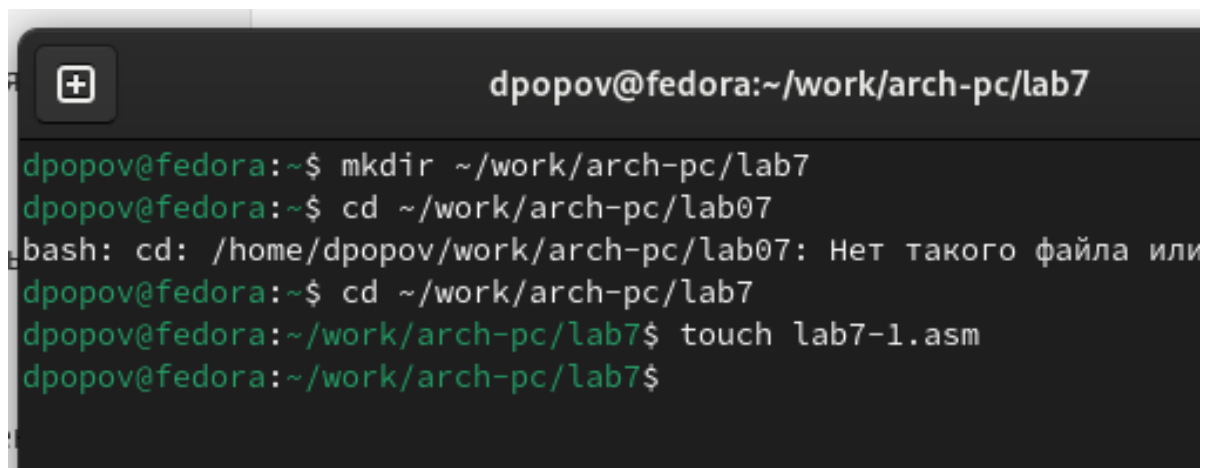
Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

## **2)Задание**

Написать программы для решения системы выражений.

### 3)Выполнение лабораторной работы

3.1)Создаем каталог для программ ЛБ7, и в нем создаем файл

A terminal window with a dark background and light text. The title bar shows the user 'dpopov@fedora' and the current directory '~/work/arch-pc/lab7'. The terminal shows a sequence of commands: 'mkdir ~/work/arch-pc/lab7', 'cd ~/work/arch-pc/lab07', an error message 'bash: cd: /home/dpopov/work/arch-pc/lab07: Нет такого файла или каталога', another 'cd ~/work/arch-pc/lab7', and finally 'touch lab7-1.asm'.

```
dpopov@fedora:~/work/arch-pc/lab7
dpopov@fedora:~$ mkdir ~/work/arch-pc/lab7
dpopov@fedora:~$ cd ~/work/arch-pc/lab07
bash: cd: /home/dpopov/work/arch-pc/lab07: Нет такого файла или каталога
dpopov@fedora:~$ cd ~/work/arch-pc/lab7
dpopov@fedora:~/work/arch-pc/lab7$ touch lab7-1.asm
dpopov@fedora:~/work/arch-pc/lab7$
```

Рис. 1: Создаем

**3.2)Открываем файл в Midnight Commander и заполняем его согласно листингу 1. Создаем исполняемый файл и проверяем**

```
dpopov@fedora:~/work/arch-pc/lab06$ mc
dpopov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab06-1 lab06-1.o
dpopov@fedora:~/work/arch-pc/lab06$ ./lab06-1
j
dpopov@fedora:~/work/arch-pc/lab06$
```

Рис. 2: Заполняем, проверяем

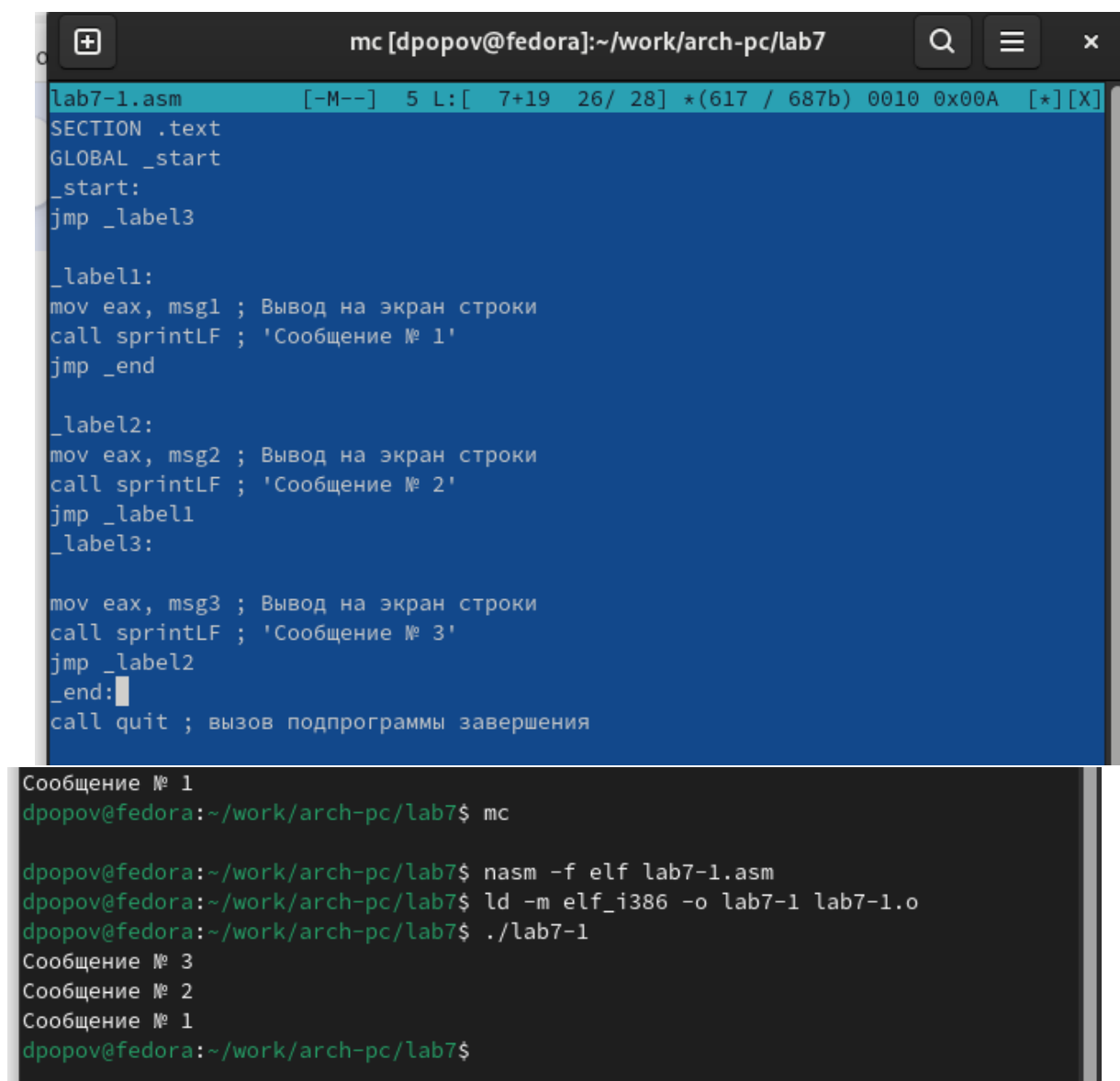
**3.3)Изменяем код в файле согласно листингу 2 и проверяем что изменилось**

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

Рис. 3: заполняем, проверяем



### 3.4) Меняем код так, чтобы код работал полностью верно



The image shows a terminal window with a dark background. The top part displays the assembly code for a file named `lab7-1.asm`. The code includes section directives, global declarations, and assembly instructions for printing messages. The bottom part shows the execution of the program, which prints three messages in Russian: "Сообщение № 1", "Сообщение № 2", and "Сообщение № 3".

```
mc [dporov@fedora]:~/work/arch-pc/lab7
lab7-1.asm [-M--] 5 L:[ 7+19 26/ 28] *(617 / 687b) 0010 0x00A [*][X]
SECTION .text
GLOBAL _start
_start:
jmp _label3

_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 1'
jmp _end

_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
jmp _label1

_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
jmp _label2

_end:
call quit ; вызов подпрограммы завершения

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

### 3.5) Создаем файл и заполняем его согласно листингу 3, проверяем работу файла

```
Сообщение № 1
dpopov@fedora:~/work/arch-pc/lab7$ touch lab7-2.asm
dpopov@fedora:~/work/arch-pc/lab7$ mc

dpopov@fedora:~/work/arch-pc/lab7$ nasm -f elf lab7-2.asm
dpopov@fedora:~/work/arch-pc/lab7$ ld -m elf_i386 lab7-2 lab7-2.o
ld: невозможно найти lab7-2: Нет такого файла или каталога
dpopov@fedora:~/work/arch-pc/lab7$ ld -m elf_i386 -o lab7-2 lab7-2.o
dpopov@fedora:~/work/arch-pc/lab7$ ./lab7-2
Введите В: 1
Наибольшее число: 50
dpopov@fedora:~/work/arch-pc/lab7$ ./lab7-2
Введите В: 100
Наибольшее число: 100
dpopov@fedora:~/work/arch-pc/lab7$ ./lab7-2
Введите В: 20
Наибольшее число: 50
dpopov@fedora:~/work/arch-pc/lab7$
```

Рис. 4: заполняем, проверяем

### 3.6) Создаем файл листинга для lab7-2.asm

```
dpopov@fedora:~/work/arch-pc/lab7$ nasm -f elf -l lab7-2.lst lab7-2.asm
dpopov@fedora:~/work/arch-pc/lab7$
```

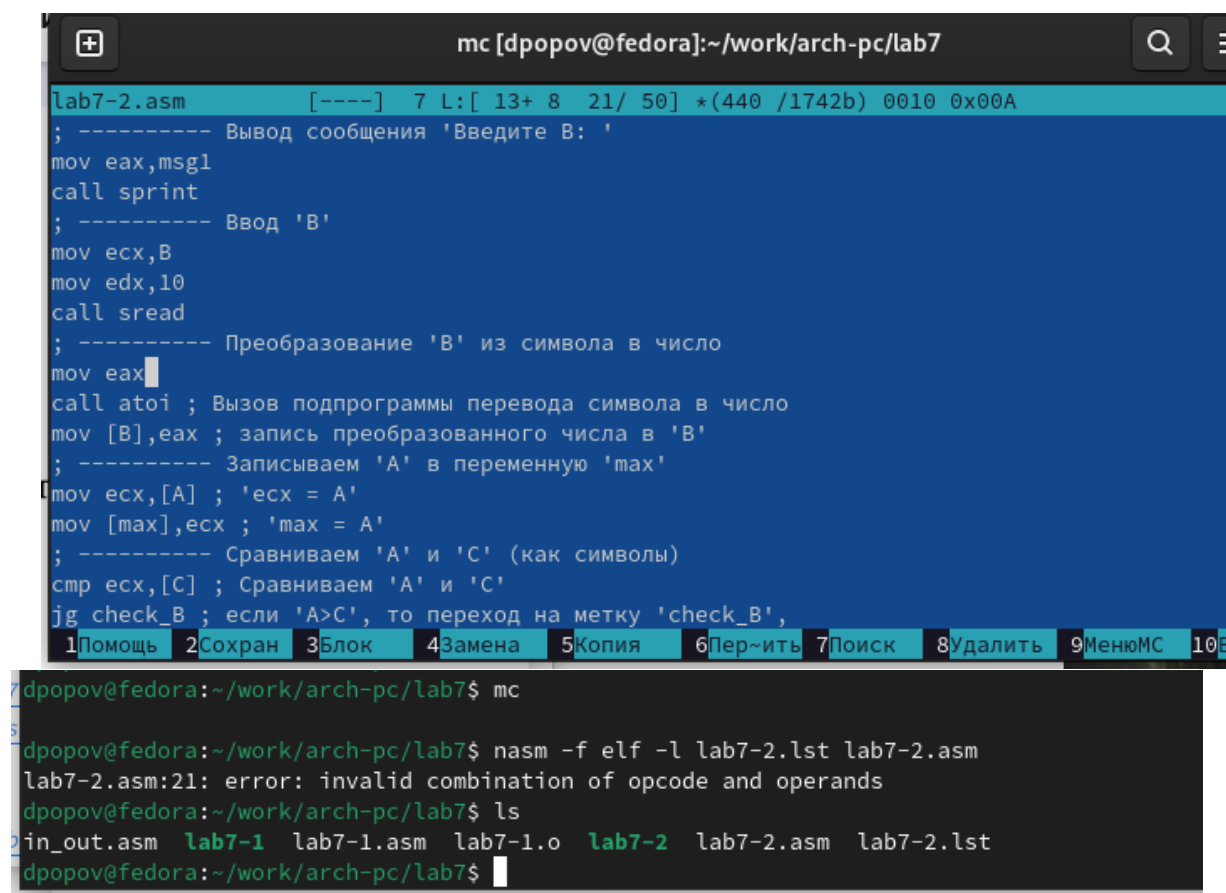
Рис. 5: создаем

### 3.7)Открываем созданный файл через Midnight Commander

```
mc [dpopov@fedora]:~/work/arch-pc/lab7
/home/dpopov/work/arch-pc/lab7/lab7-2.lst 2518/14458 17%
1                                     %include 'in_out.asm'
1                                     <1> ;----- slen -----
2                                     <1> ; Функция вычисления длины сообщения
3                                     <1> slen:
4 00000000 53                         <1>     push    ebx
5 00000001 89C3                       <1>     mov     ebx, eax
6                                     <1>
7                                     <1> nextchar:
8 00000003 803800                     <1>     cmp     byte [eax], 0
9 00000006 7403                       <1>     jz      finished
10 00000008 40                       <1>     inc     eax
11 00000009 EBF8                     <1>     jmp     nextchar
12                                     <1>
13                                     <1> finished:
14 0000000B 29D8                     <1>     sub     eax, ebx
15 0000000D 5B                       <1>     pop     ebx
16 0000000E C3                       <1>     ret
17                                     <1>
18                                     <1>
19                                     <1> ;----- sprint -----
20                                     <1> ; Функция печати сообщения
21                                     <1> ; входные данные: mov eax,<message>
22                                     <1> sprint:
23 0000000F 52                         <1>     push    edx
24 00000010 51                         <1>     push    ecx
25 00000011 53                         <1>     push    ebx
26 00000012 50                         <1>     push    eax
27 00000013 E8E8FFFFFF               <1>     call    slen
28                                     <1>
29 00000018 89C2                     <1>     mov     edx, eax
30 0000001A 58                       <1>     pop     eax
31                                     <1>
32 0000001B 89C1                     <1>     mov     ecx, eax
33 0000001D BB01000000               <1>     mov     ebx, 1
34 00000022 B804000000               <1>     mov     eax, 4
35 00000027 CD80                     <1>     int     80h
36                                     <1>
37 00000029 5B                       <1>     pop     ebx
38 0000002A 59                       <1>     pop     ecx
39 0000002B 5A                       <1>     pop     edx
40 0000002C C3                       <1>     ret
```

Изучаем файл Строка 33: 0000001D-адрес в сегменте кода, BB01000000-машинный код, mov ebx,1-присвоение переменной ebx значения 1 Строка 34: 00000022-адрес в сегменте кода, B804000000-машинный код, mov eax,4-присвоение переменной eax значения 4 Строка 35 00000027-адрес в сегменте кода, CD80-машинный код, int 80h- вызов ядра.

### 3.8) Удаляем операндум из файла, создаем файл листинга и смотрим ошибки



The screenshot shows a terminal window with a file editor. The editor displays assembly code for `lab7-2.asm`. The code includes comments in Russian and assembly instructions. A menu bar at the bottom of the editor shows options like 'Помощь', 'Сохран', 'Блок', etc. Below the editor, a terminal window shows the execution of the `mc` command, followed by `nasm -f elf -l lab7-2.lst lab7-2.asm`, which results in an error: `lab7-2.asm:21: error: invalid combination of opcode and operands`. The `ls` command is also executed, showing the files in the directory.

```
mc [dpopov@fedora]:~/work/arch-pc/lab7
lab7-2.asm [----] 7 L:[ 13+ 8 21/ 50] *(440 /1742b) 0010 0x00A
; ----- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ----- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'B' из символа в число
mov eax,
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ----- Записываем 'A' в переменную 'max'
mov ecx,[A] ; 'ecx = A'
mov [max],ecx ; 'max = A'
; ----- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[C] ; Сравниваем 'A' и 'C'
jg check_B ; если 'A>C', то переход на метку 'check_B',
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ить 7Поиск 8Удалить 9МенюМС 10Б

dpopov@fedora:~/work/arch-pc/lab7$ mc
dpopov@fedora:~/work/arch-pc/lab7$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:21: error: invalid combination of opcode and operands
dpopov@fedora:~/work/arch-pc/lab7$ ls
in_out.asm lab7-1 lab7-1.asm lab7-1.o lab7-2 lab7-2.asm lab7-2.lst
dpopov@fedora:~/work/arch-pc/lab7$
```

```

mc [dpopov@fedora]:~/work/arch-pc/lab7
/home/dpopov/work/arch-pc/lab7/lab7-2.lst 8968/14547
102 00000082 5A <1> pop edx
103 00000083 59 <1> pop ecx
104 00000084 58 <1> pop eax
105 00000085 C3 <1> ret
106 <1>
107 <1>
108 <1> ;----- iprintLF -----
109 <1> ; Функция вывода на экран чисел в
110 <1> ; входные данные: mov eax,<int>
111 <1> iprintLF:
112 00000086 E8C9FFFFFF <1> call iprint
113 <1>
114 0000008B 50 <1> push eax
115 0000008C B80A000000 <1> mov eax, 0Ah
116 00000091 50 <1> push eax
117 00000092 89E0 <1> mov eax, esp
118 00000094 E876FFFFFF <1> call sprint
119 00000099 58 <1> pop eax
120 0000009A 58 <1> pop eax
121 0000009B C3 <1> ret
122 <1>
123 <1> ;----- atoi -----
124 <1> ; Функция преобразования ascii-код
число
125 <1> ; входные данные: mov eax,<int>
126 <1> atoi:
127 0000009C 53 <1> push ebx
128 0000009D 51 <1> push ecx
129 0000009E 52 <1> push edx
130 0000009F 56 <1> push esi
131 000000A0 89C6 <1> mov esi, eax
132 000000A2 B800000000 <1> mov eax, 0
133 000000A7 B900000000 <1> mov ecx, 0
134 <1>
135 <1> .multiplyLoop:
136 000000AC 31DB <1> xor ebx, ebx
137 000000AE 8A1C0E <1> mov bl, [esi+ecx]
138 000000B1 80FB30 <1> cmp bl, 48
139 000000B4 7C14 <1> jl .finished
140 000000B6 80FB39 <1> cmp bl, 57
141 000000B9 7F0F <1> jg .finished
1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исх~ный 9

```

не смотря на ошибку файл создается

## 4)Задания для самостоятельной работы

ВАРИАНТ 10 ## 4.1)Задача 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. Создаем файл с помощью touch и пишем в нем код

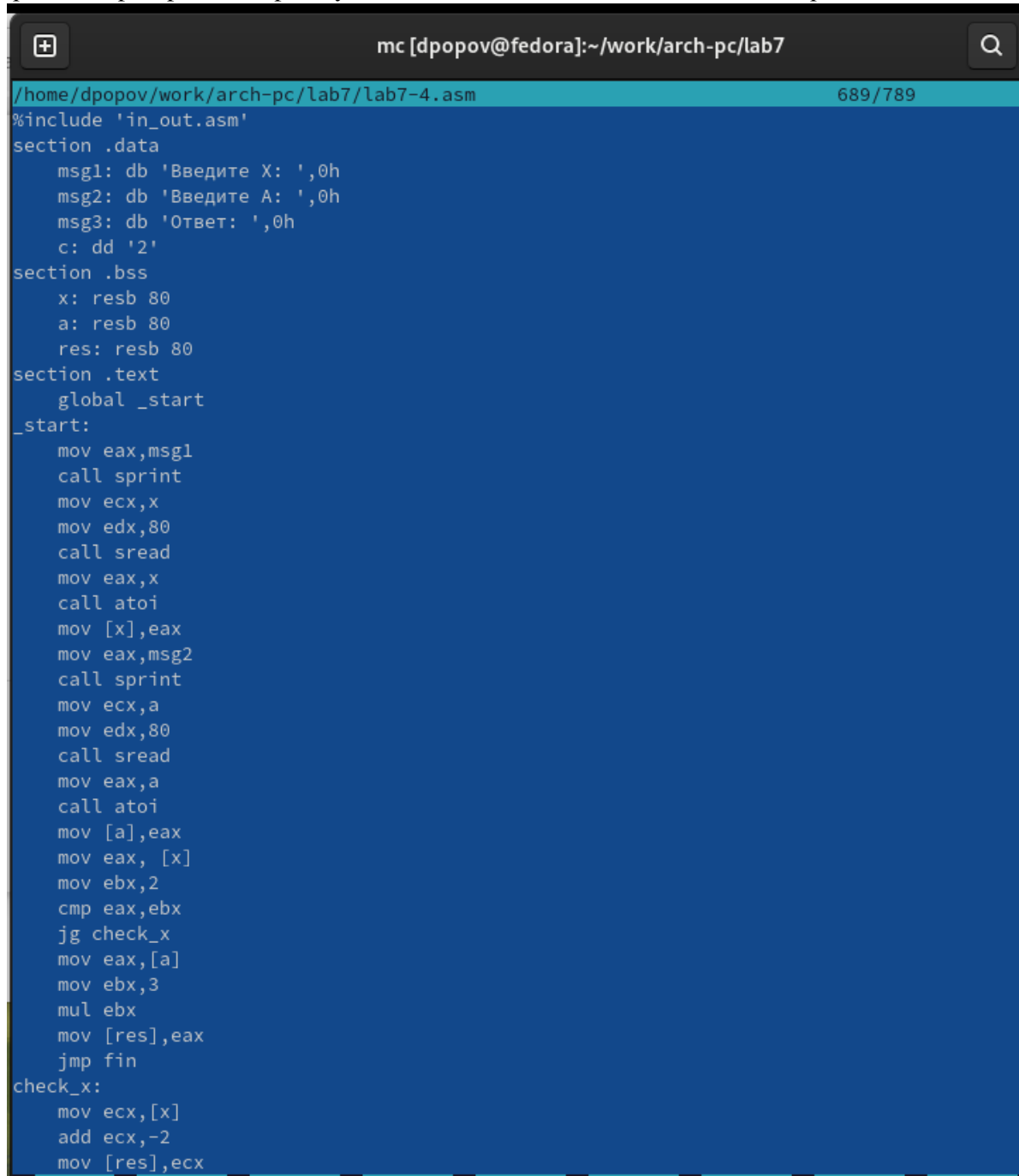
```
mc [dpopov@fedora]:~/work/arch-pc/lab7
lab7-3.asm [-M--] 13 L:[ 1+ 0 1/ 41] *(13 / 630b) 0111 0x06F
%include 'in_out.asm'
section .data
    msg1 DB '',0h
    msg2 DB '',0h
    A add '41'
    C add '35'
section .bss
    min resb 10
    B resb 10
section .text
    global _start
_start:
    mov eax,msg1
    call sprint
    mov ecx,B
    mov edx,10
    call sread
    mov eax,B
    call atoi
    mov [B],eax
    mov ecx,[A]
    mov [min], ecx
    cmp ecx,[C]
    jl check_B
    mov ecx,[C]
    mov [min],ecx
check_B:
    mov eax,min
    call atoi
    mov [min],eax
    mov ecx,[min]
    cmp ecx,[b]
    jl fin
    mov ecx,[B]
    mov [min],ecx
fin:
    mov eax, msg2
    call sprint
    mov eax,[min]
    call iprintLF
    call quit

dpopov@fedora:~/work/arch-pc/lab7$ mc

dpopov@fedora:~/work/arch-pc/lab7$ nasm -f elf lab7-3.asm
dpopov@fedora:~/work/arch-pc/lab7$ ld -m elf_i386 -o lab7-3 lab7-3.o
dpopov@fedora:~/work/arch-pc/lab7$ ./lab7-3
Введите B: 62
Наименьшее число: 35
dpopov@fedora:~/work/arch-pc/lab7$
```

##4.2)Задача 2 Напишите программу, которая для введенных с клавиатуры значе-

ний  $x$  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. Создаем файл и пишем код



```
mc [dpopov@fedora]:~/work/arch-pc/lab7
/home/dpopov/work/arch-pc/lab7/lab7-4.asm 689/789
#include 'in_out.asm'
section .data
    msg1: db 'Введите X: ',0h
    msg2: db 'Введите A: ',0h
    msg3: db 'Ответ: ',0h
    c: dd '2'
section .bss
    x: resb 80
    a: resb 80
    res: 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]
    mov ebx,2
    cmp eax,ebx
    jg check_x
    mov eax,[a]
    mov ebx,3
    mul ebx
    mov [res],eax
    jmp fin
check_x:
    mov ecx,[x]
    add ecx,-2
    mov [res],ecx
```



```

check_x:
    mov ecx,[x]
    add ecx,-2
    mov [res],ecx
    jmp fin
fin:
    mov eax,msg3
    call sprint
    mov eax,[res]
    call iprintLF
    call quit

```

1Помощь 2Разверн 3Выход 4Hex 5Перейти 6 7Поиск 8Исходный 9Формат 1

```

dpopov@fedora:~/work/arch-pc/lab7$ mc

dpopov@fedora:~/work/arch-pc/lab7$ nasm -f elf lab7-4.asm
dpopov@fedora:~/work/arch-pc/lab7$ ld -m elf_i386 -o lab7-4 lab7-4.o
dpopov@fedora:~/work/arch-pc/lab7$ ./lab7-4
Введите X: 3
Введите A: 1
Ответ: 1
dpopov@fedora:~/work/arch-pc/lab7$ mc

dpopov@fedora:~/work/arch-pc/lab7$ ./lab7-4
Введите X: 3
Введите A: 0
Ответ: 1
dpopov@fedora:~/work/arch-pc/lab7$ ./lab7-4
Введите X: 1
Введите A: 2
Ответ: 6
dpopov@fedora:~/work/arch-pc/lab7$

```

## **5)Выводы**

Мы познакомились с структурой файла листинга, изучили команды условного и безусловного перехода.