

# **Отчёт по лабораторной работе 7**

**Архитектура компьютеров**

Наурузова А.М. НПИбд-03-24

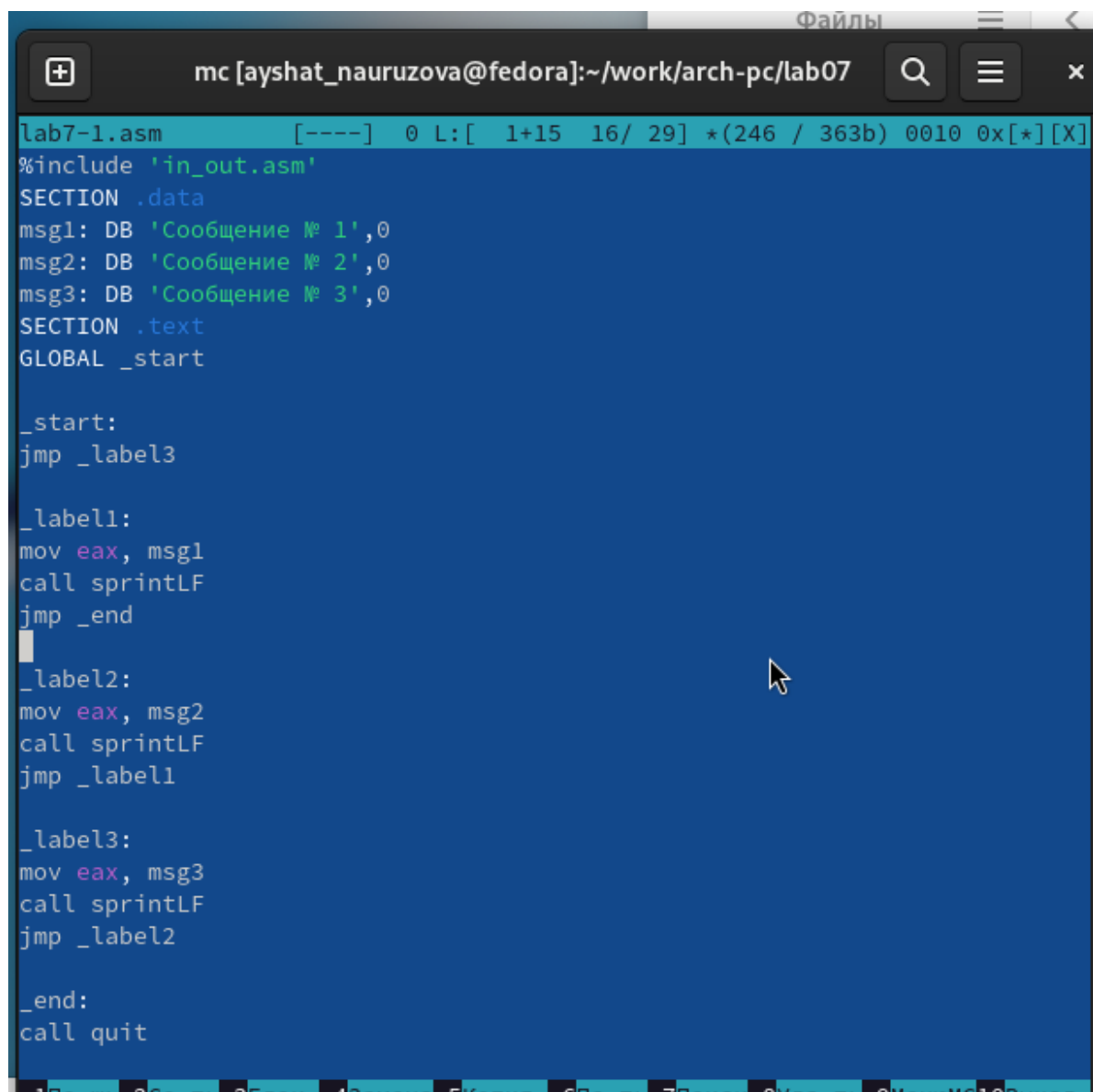
# Содержание

0.1	Самостоятельное задание . . . . .	9
<b>1</b>	<b>Выводы</b>	<b>12</b>

## Список иллюстраций

1	Программа в файле lab7-1.asm . . . . .	4
2	Запуск программы lab7-1.asm . . . . .	5
3	Программа в файле lab7-2.asm . . . . .	6
4	Запуск программы lab7-2.asm . . . . .	6
5	Файл листинга lab7-2 . . . . .	7
6	Ошибка трансляции lab7-2 . . . . .	8
7	Файл листинга с ошибкой lab7-2 . . . . .	8
8	Программа в файле prog1.asm . . . . .	9
9	Запуск программы prog1.asm . . . . .	10
10	Программа в файле prog2.asm . . . . .	11
11	Запуск программы prog2.asm . . . . .	11

# Список таблиц



```
lab7-1.asm [----] 0 L:[ 1+15 16/ 29] *(246 / 363b) 0010 0x[*][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
```

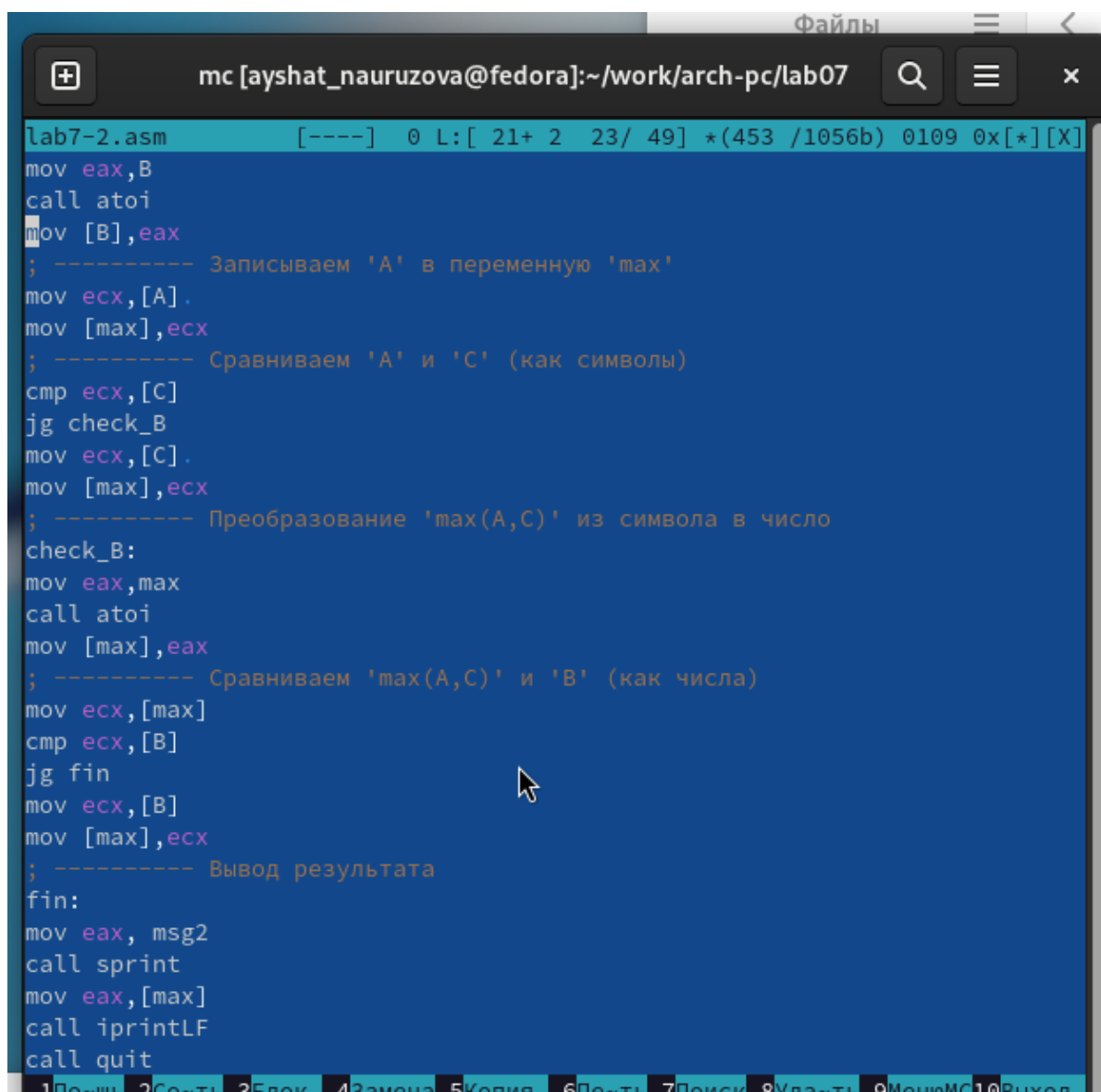
Рис. 1: Программа в файле lab7-1.asm

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

Рис. 2: Запуск программы lab7-1.asm

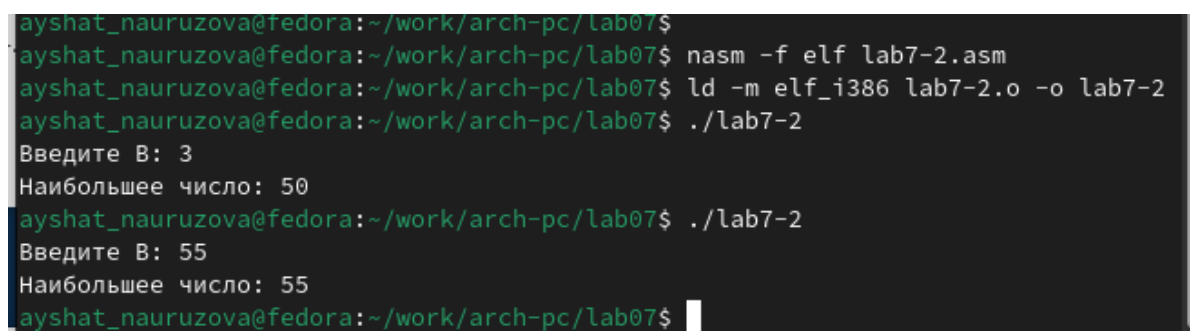
Использование инструкции `jmp` приводит к переходу в любом случае. Однако часто при написании программ требуется использовать условные переходы, когда переход должен происходить при выполнении какого-либо условия. В качестве примера рассмотрим программу, которая находит и выводит на экран наибольшую из 3 целочисленных переменных: А, В и С. Значения для А и С заданы в программе, значение В вводится с клавиатуры.

Я создала исполняемый файл и проверила его работу для разных значений В (рис. 3) (рис. 4).



```
lab7-2.asm  [----]  0 L:[ 21+ 2  23/ 49] *(453 /1056b) 0109 0x[*] [X]
mov  eax,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. 3: Программа в файле lab7-2.asm

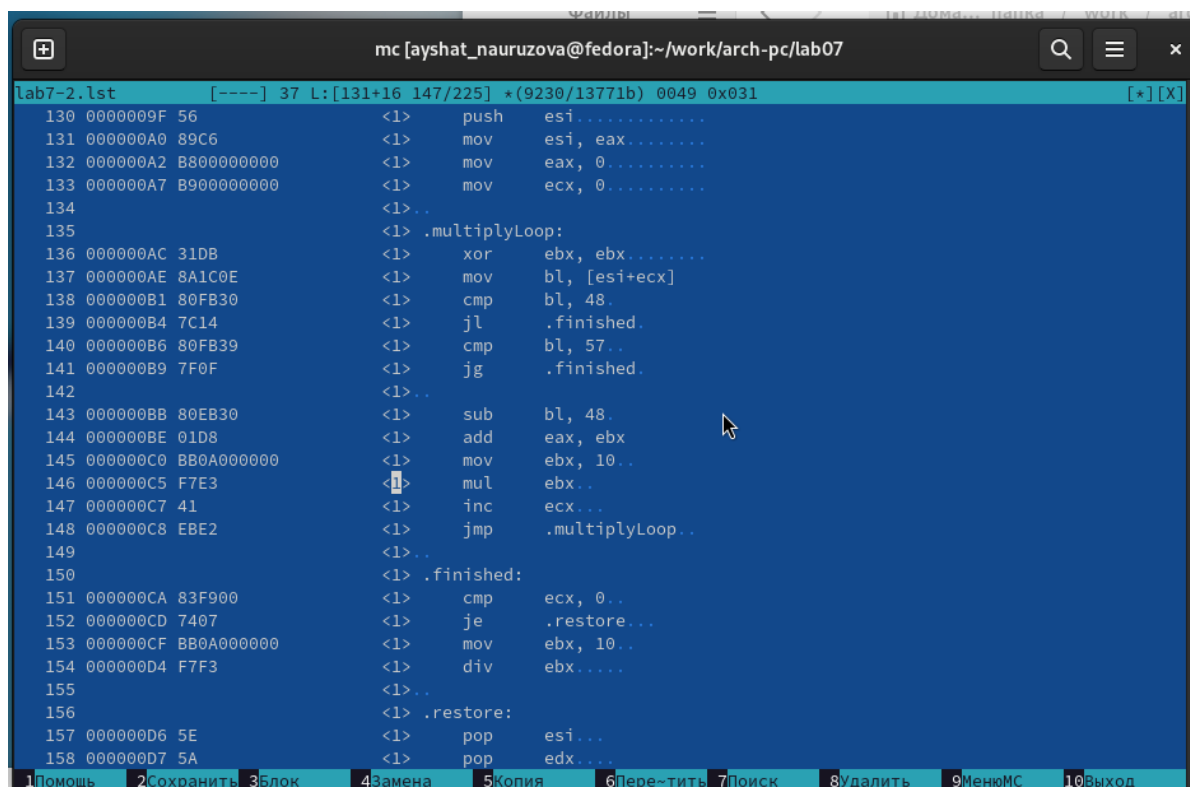


```
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-2.o -o lab7-2
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 3
Наибольшее число: 50
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 55
Наибольшее число: 55
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$
```

Рис. 4: Запуск программы lab7-2.asm

Обычно nasm создает в результате ассемблирования только объектный файл. Чтобы получить файл листинга, нужно указать ключ -l и задать имя файла листинга в командной строке.

Я создала файл листинга для программы из файла lab7-2.asm (рис. 5)



```
lab7-2.lst [----] 37 L:[131+16 147/225] *(9230/13771b) 0049 0x031 [*][X]
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.
142 <1>..
143 000000BB 80EB30 <1> sub bl, 48.
144 000000BE 01D8 <1> add eax, ebx
145 000000C0 BB0A000000 <1> mov ebx, 10..
146 000000C5 F7E3 <1> mul ebx..
147 000000C7 41 <1> inc ecx..
148 000000C8 EBE2 <1> jmp .multiplyLoop..
149 <1>..
150 <1> .finished:
151 000000CA 83F900 <1> cmp ecx, 0..
152 000000CD 7407 <1> je .restore...
153 000000CF BB0A000000 <1> mov ebx, 10..
154 000000D4 F7F3 <1> div ebx....
155 <1>..
156 <1> .restore:
157 000000D6 5E <1> pop esi...
158 000000D7 5A <1> pop edx...
```

Рис. 5: Файл листинга lab7-2

Внимательно ознакомилась с его форматом и содержимым. Подробно объяснила содержимое трёх строк файла листинга.

строка 189: - 14 — номер строки в подпрограмме - 000000E8 — адрес - B8[00000000] — машинный код - mov eax,msg1 — код программы — перекладывает msg1 в eax

строка 190: - 15 — номер строки в подпрограмме - 000000ED — адрес - E81DFFFFFF — машинный код - call sprint — код программы — вызов подпрограммы печати

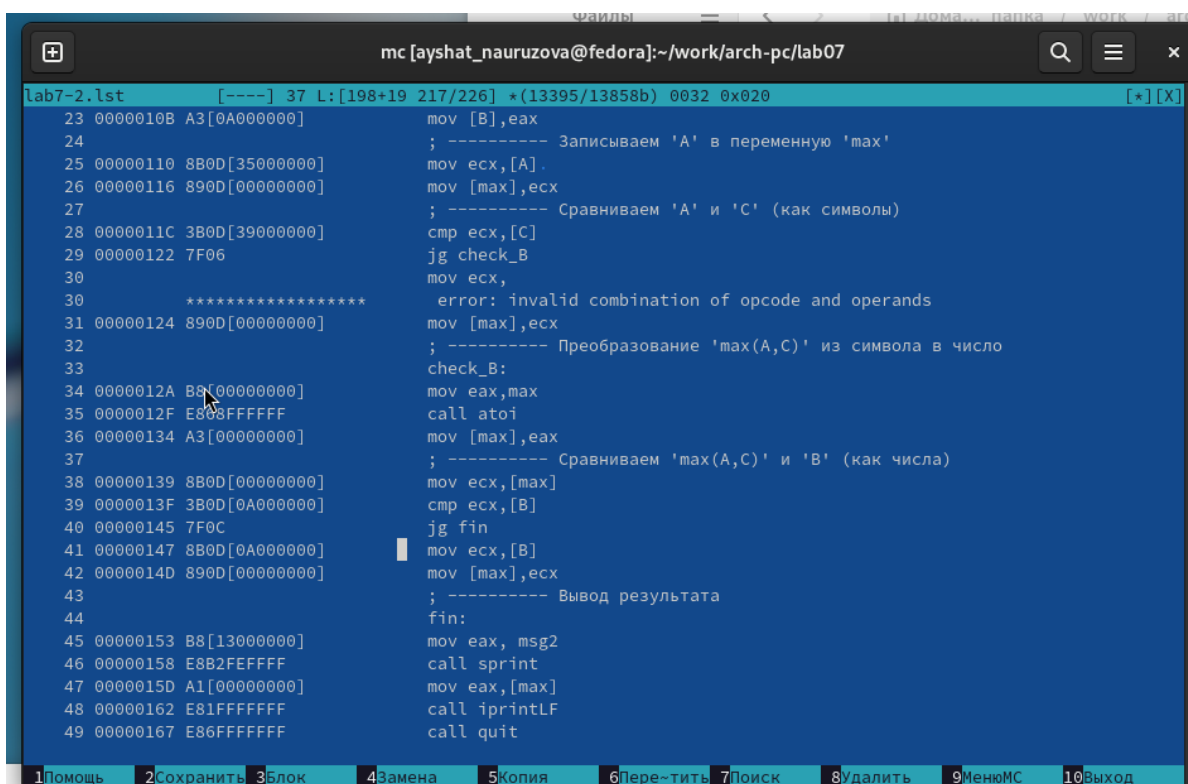
строка 192: - 17 — номер строки в подпрограмме - 000000F2 — адрес -

B9[0A000000] — машинный код - mov ecx,B — код программы — перекладывает B в eax

Я открыла файл с программой lab7-2.asm и в инструкции с двумя операндами удалила один операнд. Выполнила трансляцию с получением файла листинга. (рис. 6) (рис. 7)

```
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$  
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm -l lab7-2.lst  
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm -l lab7-2.lst  
lab7-2.asm:30: error: invalid combination of opcode and operands  
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$
```

Рис. 6: Ошибка трансляции lab7-2



```
lab7-2.lst [----] 37 L:[198+19 217/226] *(13395/13858b) 0032 0x020 [*][X]  
23 0000010B A3[0A000000] mov [B],eax  
24 ; ----- Записываем 'A' в переменную 'max'  
25 00000110 8B0D[35000000] mov ecx,[A]  
26 00000116 890D[00000000] mov [max],ecx  
27 ; ----- Сравниваем 'A' и 'C' (как символы)  
28 0000011C 3B0D[39000000] cmp ecx,[C]  
29 00000122 7F06 jg check_B  
30 mov ecx,  
30 ***** error: invalid combination of opcode and operands  
31 00000124 890D[00000000] mov [max],ecx  
32 ; ----- Преобразование 'max(A,C)' из символа в число  
33 check_B:  
34 0000012A B8[00000000] mov eax,max  
35 0000012F E808FFFFFF call atoi  
36 00000134 A3[00000000] mov [max],eax  
37 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)  
38 00000139 8B0D[00000000] mov ecx,[max]  
39 0000013F 3B0D[0A000000] cmp ecx,[B]  
40 00000145 7F0C jg fin  
41 00000147 8B0D[0A000000] mov ecx,[B]  
42 0000014D 890D[00000000] mov [max],ecx  
43 ; ----- Вывод результата  
44 fin:  
45 00000153 B8[13000000] mov eax,msg2  
46 00000158 E8B2FFFFFF call sprint  
47 0000015D A1[00000000] mov eax,[max]  
48 00000162 E81FFFFFFF call iprintLF  
49 00000167 E86FFFFFFF call quit  
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Пере-тить 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 7: Файл листинга с ошибкой lab7-2

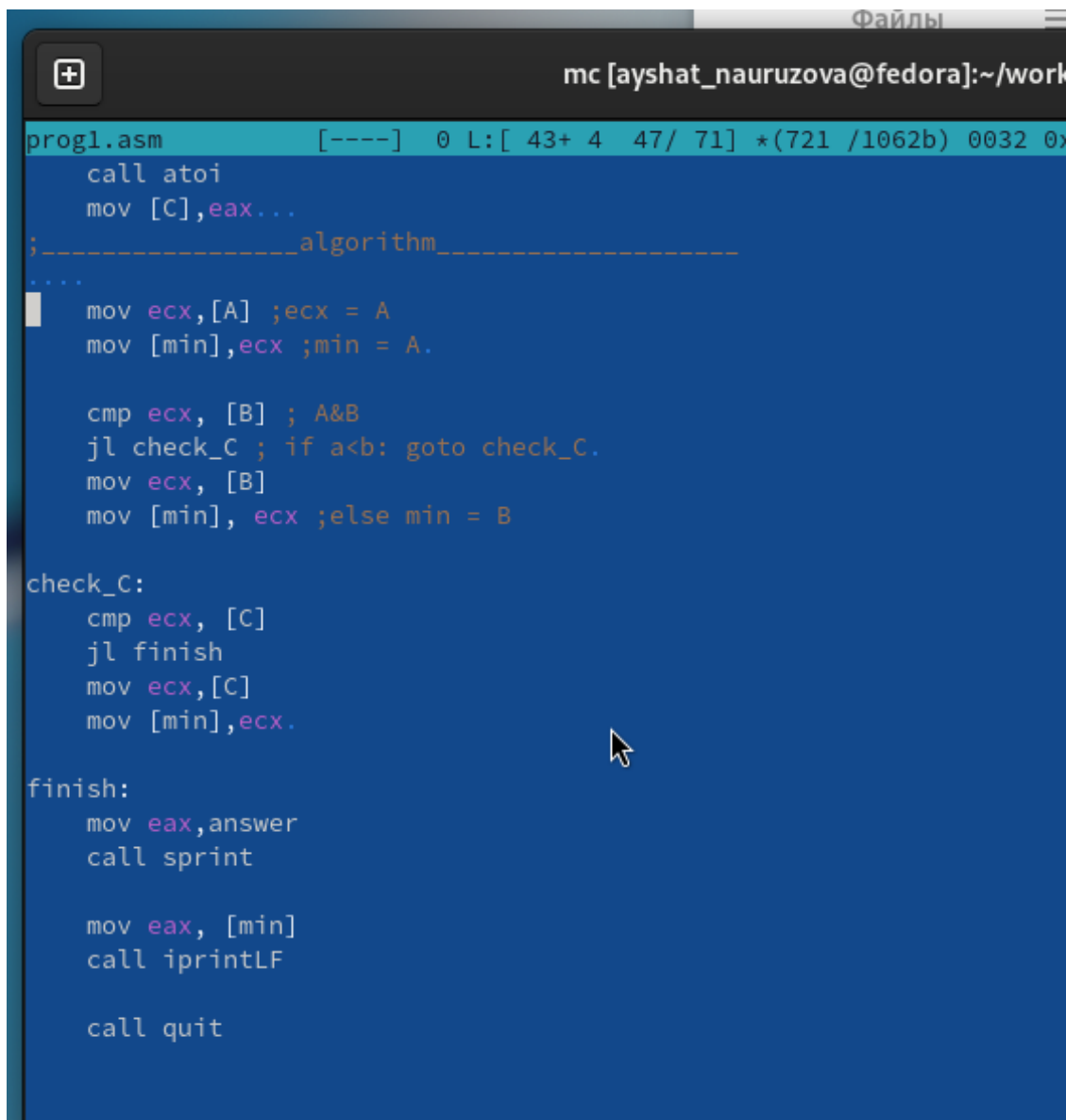
Объектный файл не смог создаться из-за ошибки. Однако я получила листинг, в котором выделено место ошибки.



## 0.1 Самостоятельное задание

Напиши программу нахождения наименьшей из 3 целочисленных переменных а, b и с. Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 6. Создаю исполняемый файл и проверяю его работу (рис. 8) (рис. 9).

для варианта 9 — 24,98,15



```
mc [ayshat_nauruzova@fedora]:~/work
prog1.asm [----] 0 L: [ 43+ 4 47/ 71] *(721 /1062b) 0032 0x
    call atoi
    mov [C],eax...
;-----algorithm-----
....
    mov ecx,[A] ;ecx = A
    mov [min],ecx ;min = A.

    cmp ecx, [B] ; A&B
    jl check_C ; if a<b: goto check_C.
    mov ecx, [B]
    mov [min], ecx ;else min = B

check_C:
    cmp ecx, [C]
    jl finish
    mov ecx,[C]
    mov [min],ecx.

finish:
    mov eax,answer
    call sprint

    mov eax, [min]
    call iprintLF

    call quit
```

Рис. 8: Программа в файле prog1.asm

```
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$  
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ nasm -f elf prog1.asm  
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 prog1.o -o prog1  
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ ./prog1  
Input A: 6  
Input B: 4  
Input C: 1  
Smallest: 1  
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ ./prog1  
Input A: 24  
Input B: 98  
Input C: 15  
Smallest: 15  
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$
```

Рис. 9: Запуск программы prog1.asm

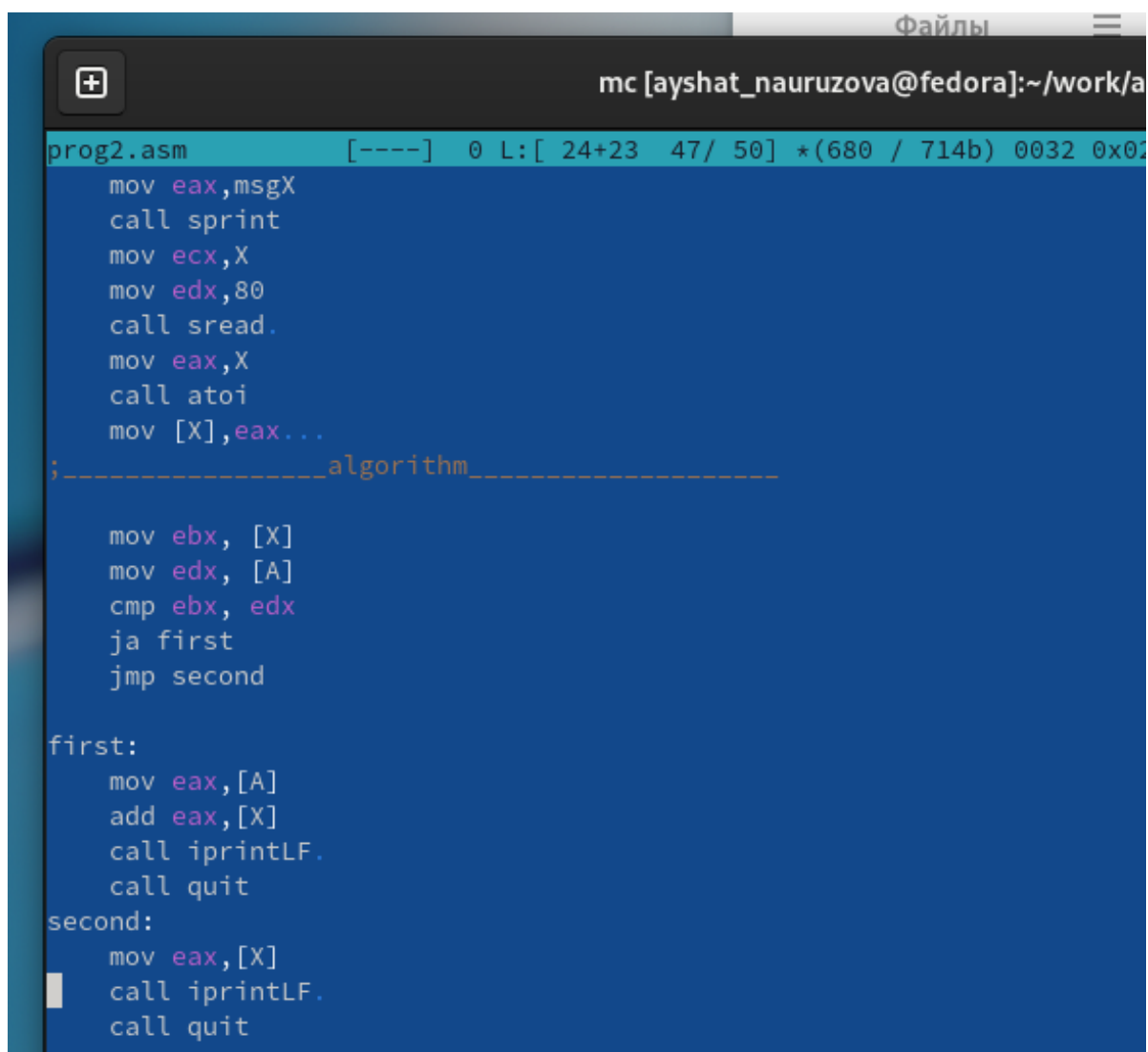
Теперь пишу программу, которая для введенных с клавиатуры значений  $x$  и  $a$  вычисляет значение заданной функции  $f(x)$  и выводит результат вычислений. Вид функции  $f(x)$  выбираю из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создаю исполняемый файл и проверяю его работу для значений  $X$  и  $a$  из 7.6. (рис. 10) (рис. 11).

для варианта 9:

$$\begin{cases} a + x, x > a \\ x, x \leq a \end{cases}$$

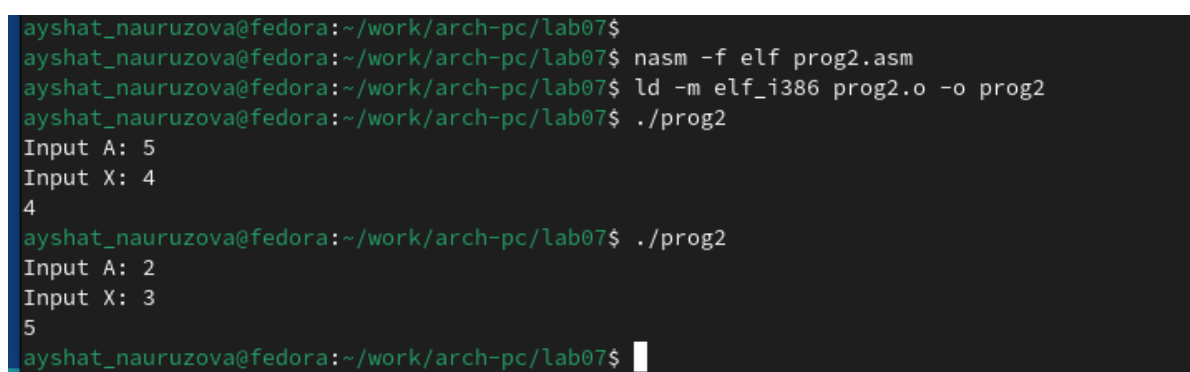
Если подставить  $x = 4, a = 5$  получается 4.

Если подставить  $x = 3, a = 2$  получается  $3 + 2 = 5$ .



```
mc [ayshat_nauruzova@fedora]:~/work/a
prog2.asm [----] 0 L: [ 24+23 47/ 50] *(680 / 714b) 0032 0x02
mov eax,msgX
call sprint
mov ecx,X
mov edx,80
call sread.
mov eax,X
call atoi
mov [X],eax...
;-----algorithm-----
mov ebx, [X]
mov edx, [A]
cmp ebx, edx
ja first
jmp second
first:
mov eax,[A]
add eax,[X]
call iprintLF.
call quit
second:
mov eax,[X]
call iprintLF.
call quit
```

Рис. 10: Программа в файле prog2.asm



```
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ nasm -f elf prog2.asm
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 prog2.o -o prog2
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ ./prog2
Input A: 5
Input X: 4
4
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$ ./prog2
Input A: 2
Input X: 3
5
ayshat_nauruzova@fedora:~/work/arch-pc/lab07$
```

Рис. 11: Запуск программы prog2.asm

# 1 Выводы

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