

Отчёт по лабораторной работе 7

дисциплина: Архитектура компьютера

Алан Таймуразвич Кесаев

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	9
2.5	Программа lab7-1.asm	10
2.6	Запуск программы lab7-1.asm	11
2.7	Программа lab7-2.asm	12
2.8	Запуск программы lab7-2.asm	12
2.9	Файл листинга lab7-2	13
2.10	Ошибка трансляции lab7-2	14
2.11	Файл листинга с ошибкой lab7-2	15
2.12	Программа lab7-3.asm	16
2.13	Запуск программы lab7-3.asm	16
2.14	Программа lab7-4.asm	18
2.15	Запуск программы lab7-4.asm	19

Список таблиц

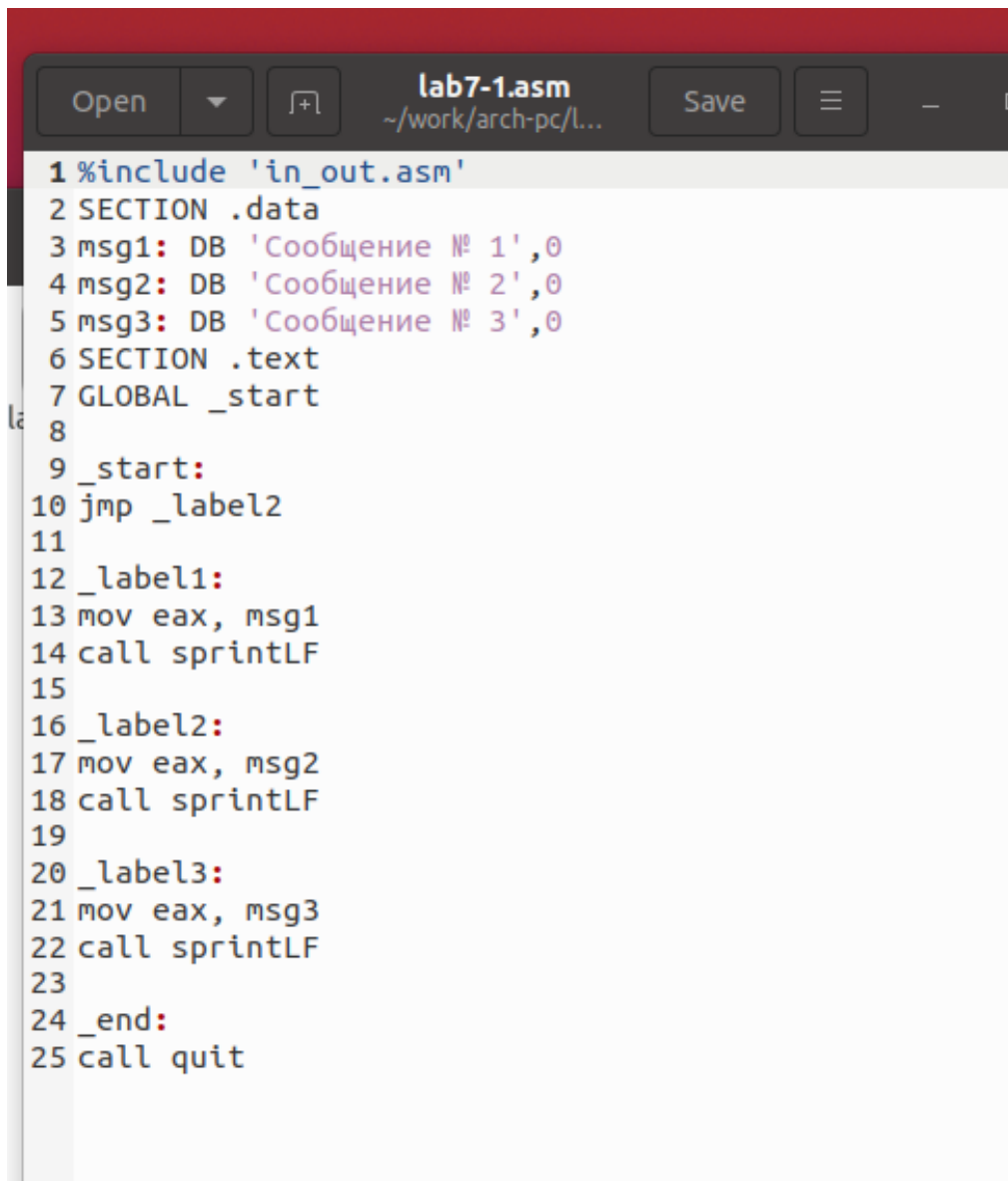
1 Цель работы

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`.

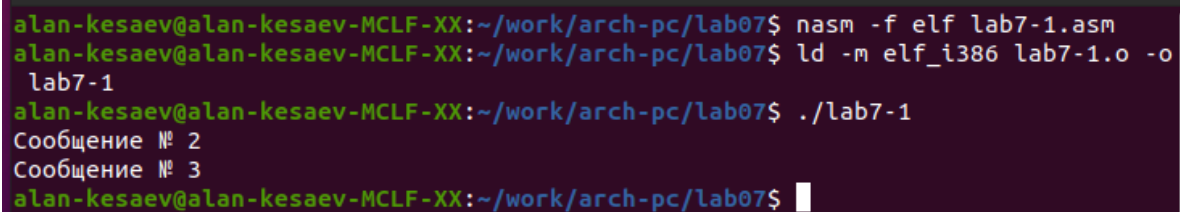
Написал в файл lab7-1.asm текст программы из листинга 7.1.



```
1 %include 'in_out.asm'
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
9 _start:
10 jmp _label2
11
12 _label1:
13 mov eax, msg1
14 call sprintLF
15
16 _label2:
17 mov eax, msg2
18 call sprintLF
19
20 _label3:
21 mov eax, msg3
22 call sprintLF
23
24 _end:
25 call quit
```

Рис. 2.1: Программа lab7-1.asm

Создал исполняемый файл и запустил его.

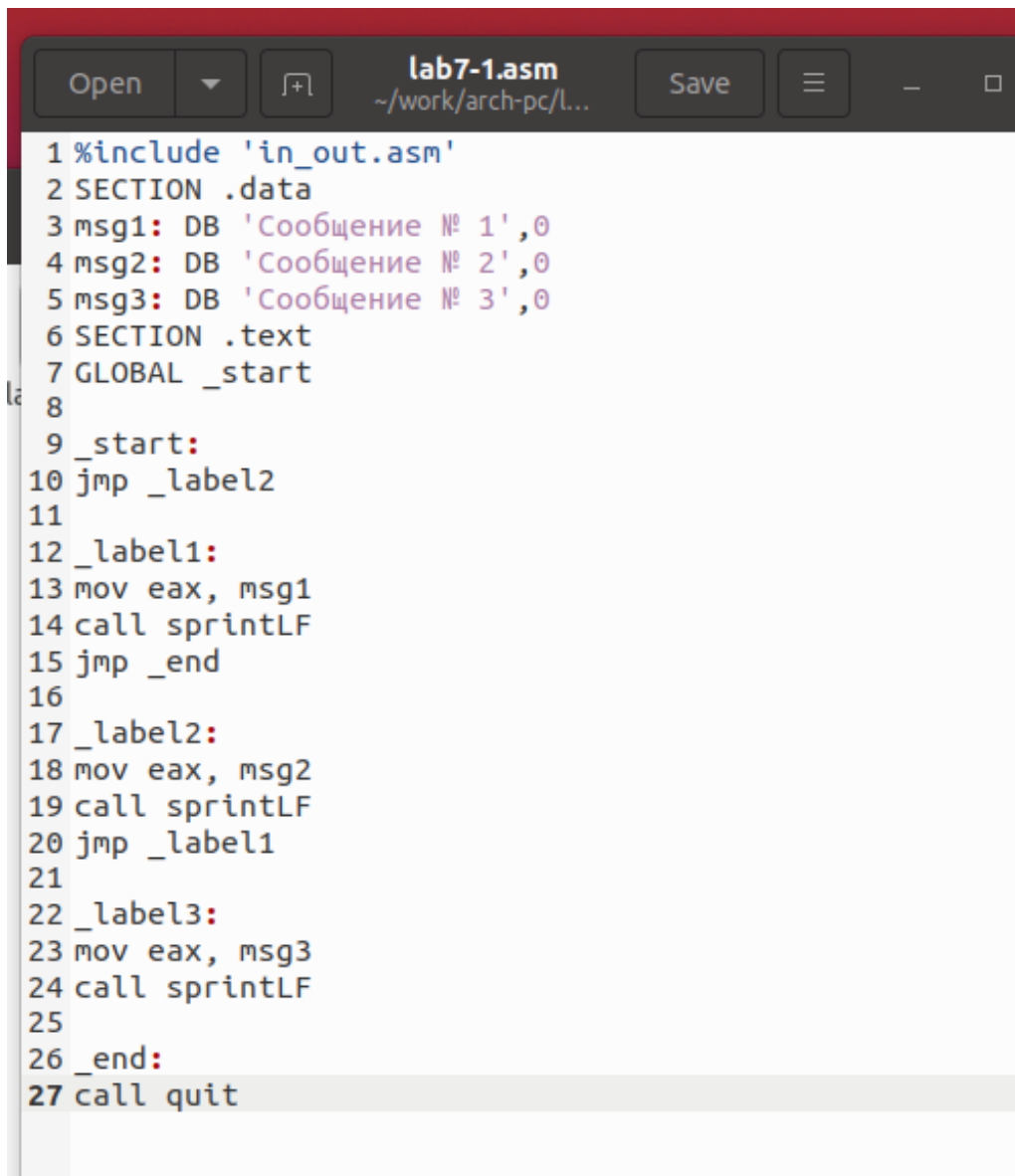
A terminal window with a dark purple background and green text. The prompt is 'alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07\$'. The first command is 'nasm -f elf lab7-1.asm'. The second command is 'ld -m elf_i386 lab7-1.o -o lab7-1'. The third command is './lab7-1'. The output shows 'Сообщение № 2' and 'Сообщение № 3' on separate lines. The prompt returns after the execution.

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

Рис. 2.2: Запуск программы lab7-1.asm

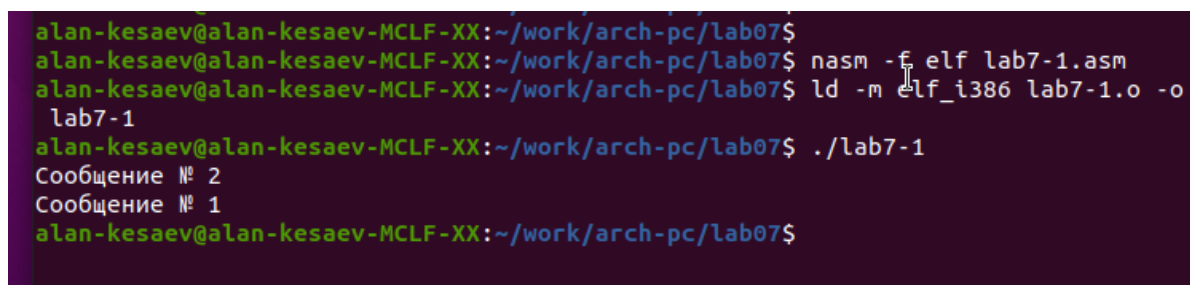
Инструкция `jmp` позволяет осуществлять переходы не только вперед но и назад. Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого в текст программы после вывода сообщения № 2 добавим инструкцию `jmp` с меткой `_label1` (т.е. 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.



```
1 %include 'in_out.asm'
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
9 _start:
10 jmp _label2
11
12 _label1:
13 mov eax, msg1
14 call sprintLF
15 jmp _end
16
17 _label2:
18 mov eax, msg2
19 call sprintLF
20 jmp _label1
21
22 _label3:
23 mov eax, msg3
24 call sprintLF
25
26 _end:
27 call quit
```

Рис. 2.3: Программа lab7-1.asm



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

```
1 %include 'in_out.asm'
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
9 _start:
10 jmp _label3
11
12 _label1:
13 mov eax, msg1
14 call sprintfLF
15 jmp _end
16
17 _label2:
18 mov eax, msg2
19 call sprintfLF
20 jmp _label1
21
22 _label3:
23 mov eax, msg3
24 call sprintfLF
25 jmp _label2
26
27 _end:
28 call quit
```

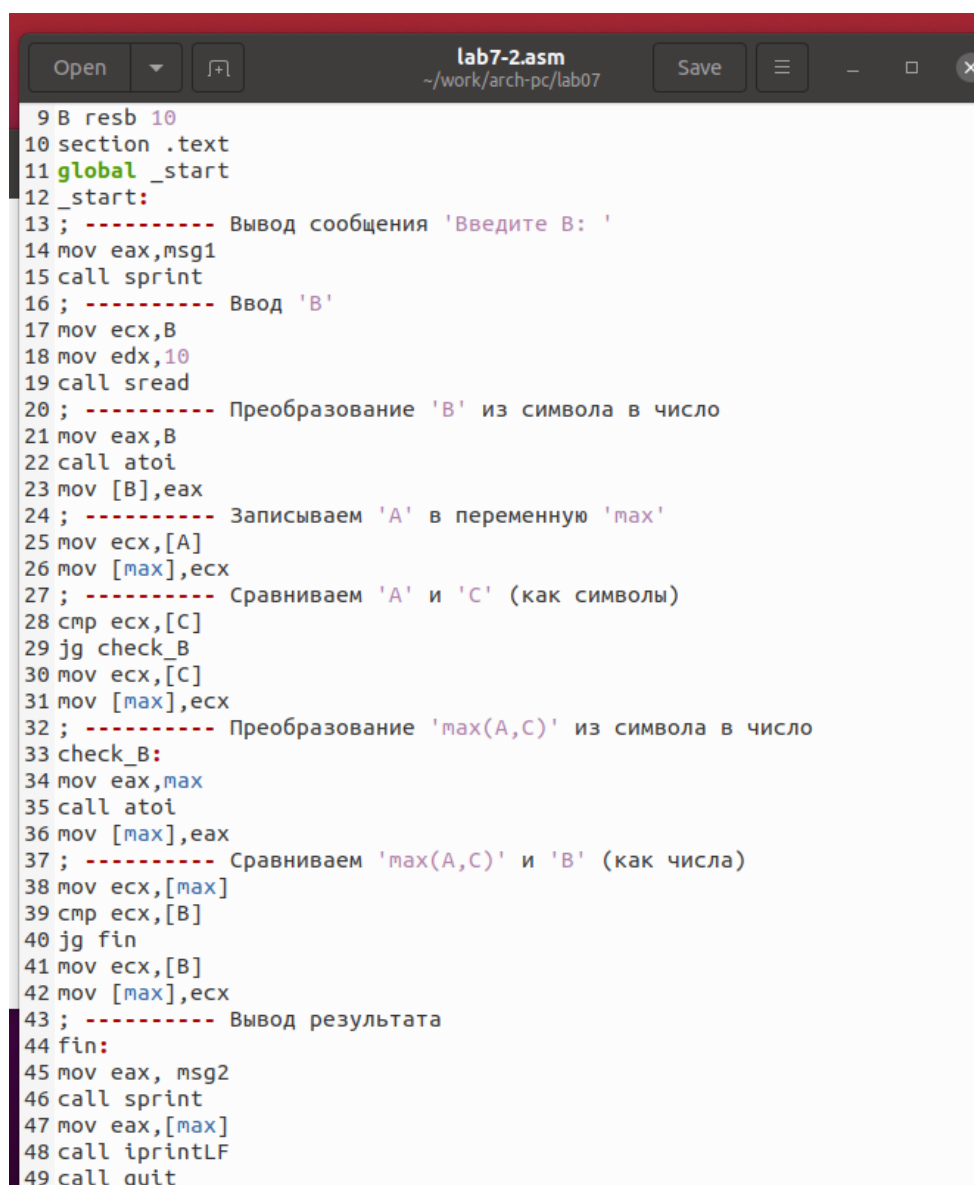
Рис. 2.5: Программа lab7-1.asm

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

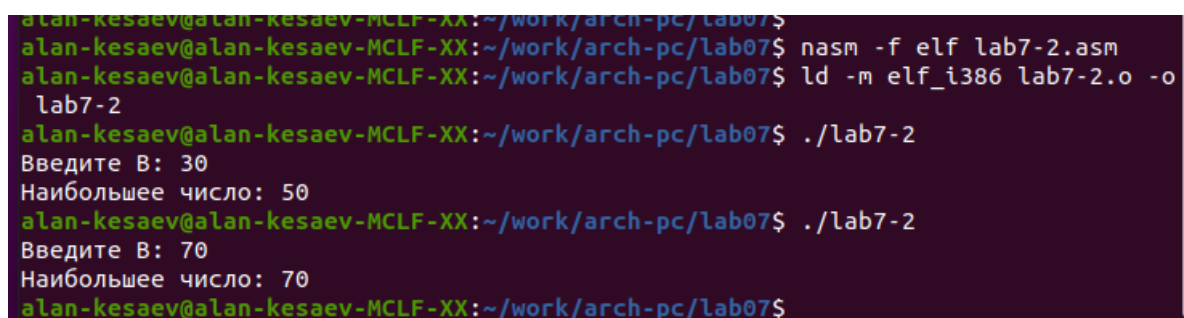
Создал исполняемый файл и проверил его работу для разных значений B.



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

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
23 mov [B],eax
24 ; ----- Записываем 'A' в переменную 'max'
25 mov ecx,[A]
26 mov [max],ecx
27 ; ----- Сравниваем 'A' и 'C' (как символы)
28 cmp ecx,[C]
29 jg check_B
30 mov ecx,[C]
31 mov [max],ecx
32 ; ----- Преобразование 'max(A,C)' из символа в число
33 check_B:
34 mov eax,max
35 call atoi
36 mov [max],eax
37 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
38 mov ecx,[max]
39 cmp ecx,[B]
40 jg fin
41 mov ecx,[B]
42 mov [max],ecx
43 ; ----- Вывод результата
44 fin:
45 mov eax, msg2
46 call sprint
47 mov eax,[max]
48 call iprintLF
49 call quit
```

Рис. 2.7: Программа lab7-2.asm

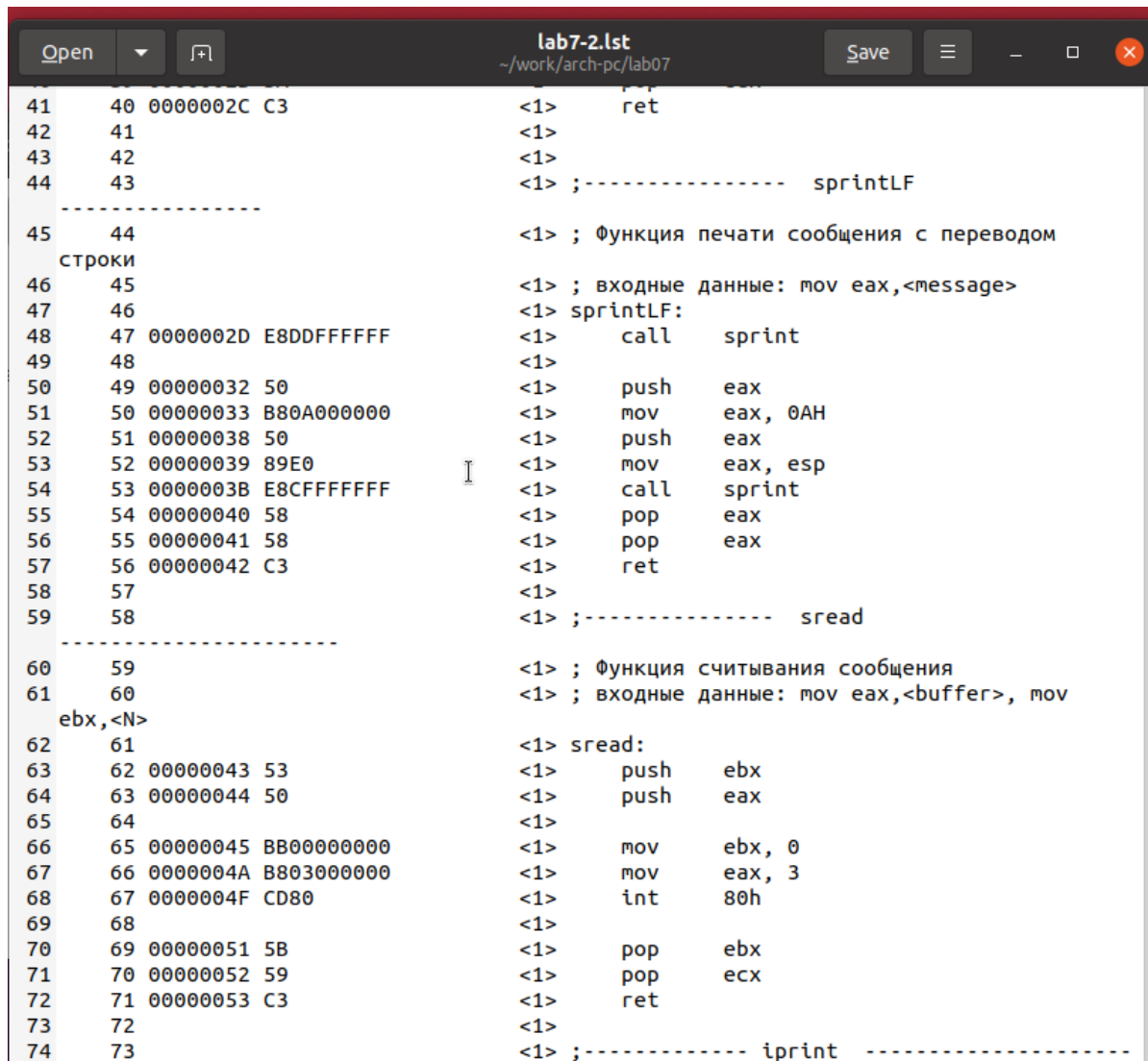


```
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-2.o -o
lab7-2
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 30
Наибольшее число: 50
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 70
Наибольшее число: 70
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/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.lst
~/work/arch-pc/lab07

41      40 0000002C C3          <1>      ret
42      41
43      42
44      43          <1> ;----- sprintLF
-----
45      44          <1> ; Функция печати сообщения с переводом
строки
46      45          <1> ; входные данные: mov eax,<message>
47      46          <1> sprintLF:
48      47 0000002D E8DDFFFFFF  <1>      call    sprint
49      48          <1>
50      49 00000032 50          <1>      push    eax
51      50 00000033 B80A000000  <1>      mov     eax, 0Ah
52      51 00000038 50          <1>      push    eax
53      52 00000039 89E0          <1>      mov     eax, esp
54      53 0000003B E8CFFFFFFF  <1>      call    sprint
55      54 00000040 58          <1>      pop     eax
56      55 00000041 58          <1>      pop     eax
57      56 00000042 C3          <1>      ret
58      57          <1>
59      58          <1> ;----- sread
-----
60      59          <1> ; Функция считывания сообщения
61      60          <1> ; входные данные: mov eax,<buffer>, mov
ebx,<N>
62      61          <1> sread:
63      62 00000043 53          <1>      push    ebx
64      63 00000044 50          <1>      push    eax
65      64          <1>
66      65 00000045 BB00000000  <1>      mov     ebx, 0
67      66 0000004A B803000000  <1>      mov     eax, 3
68      67 0000004F CD80          <1>      int     80h
69      68          <1>
70      69 00000051 5B          <1>      pop     ebx
71      70 00000052 59          <1>      pop     ecx
72      71 00000053 C3          <1>      ret
73      72          <1>
74      73          <1> ;----- iprint -----
```

Рис. 2.9: Файл листинга lab7-2

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

строка 34

- 34 - номер строки
- 00000130 - адрес
- B8[00000000] - машинный код
- mov eax, max - код программы

строка 35

- 35 - номер строки
- 00000135 - адрес
- E862FFFFFF - машинный код
- call atoi - код программы

строка 36

- 36 - номер строки
- 0000013A - адрес
- A3[00000000] - машинный код
- mov [max], eax - код программы

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

```
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm -l
lab7-2.lst
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm -l
lab7-2.lst
lab7-2.asm:18: error: invalid combination of opcode and operands
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$
```

Рис. 2.10: Ошибка трансляции lab7-2

```

lab7-2.lst
~/work/arch-pc/lab07
Open Save

178 4 00000025 D0B520D187D0B8D181-
179 4 0000002E D0BBD0BE3A2000
180 5 00000035 32300000 A dd '20'
181 6 00000039 35300000 C dd '50'
182 7 section .bss
183 8 00000000 <res 0000000A> max resb 10
184 9 0000000A <res 0000000A> B resb 10
185 10 section .text
186 11 global _start
187 12 _start:
188 13 ; ----- Вывод сообщения 'Введите B: '
189 14 000000E8 B8[00000000] mov eax,msg1
190 15 000000ED E81DFFFFFF call sprint
191 16 ; ----- Ввод 'B'
192 17 000000F2 B9[0A000000] mov ecx,B
193 18 mov edx,
194 18 ***** error: invalid combination of opcode and
operands
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)
204 28 00000117 3B0D[39000000] cmp ecx,[C]
205 29 0000011D 7F0C jg check_B
206 30 0000011F 8B0D[39000000] mov ecx,[C]
207 31 00000125 890D[00000000] mov [max],ecx
208 32 ; ----- Преобразование 'max(A,C)' из
символа в число
209 33 check_B:
210 34 0000012B B8[00000000] mov eax,max
211 35 00000130 E867FFFFFF call atoi
212 36 00000135 A3[00000000] mov [max],eax
213 37 ; ----- Создаем 'max(A,C)' и

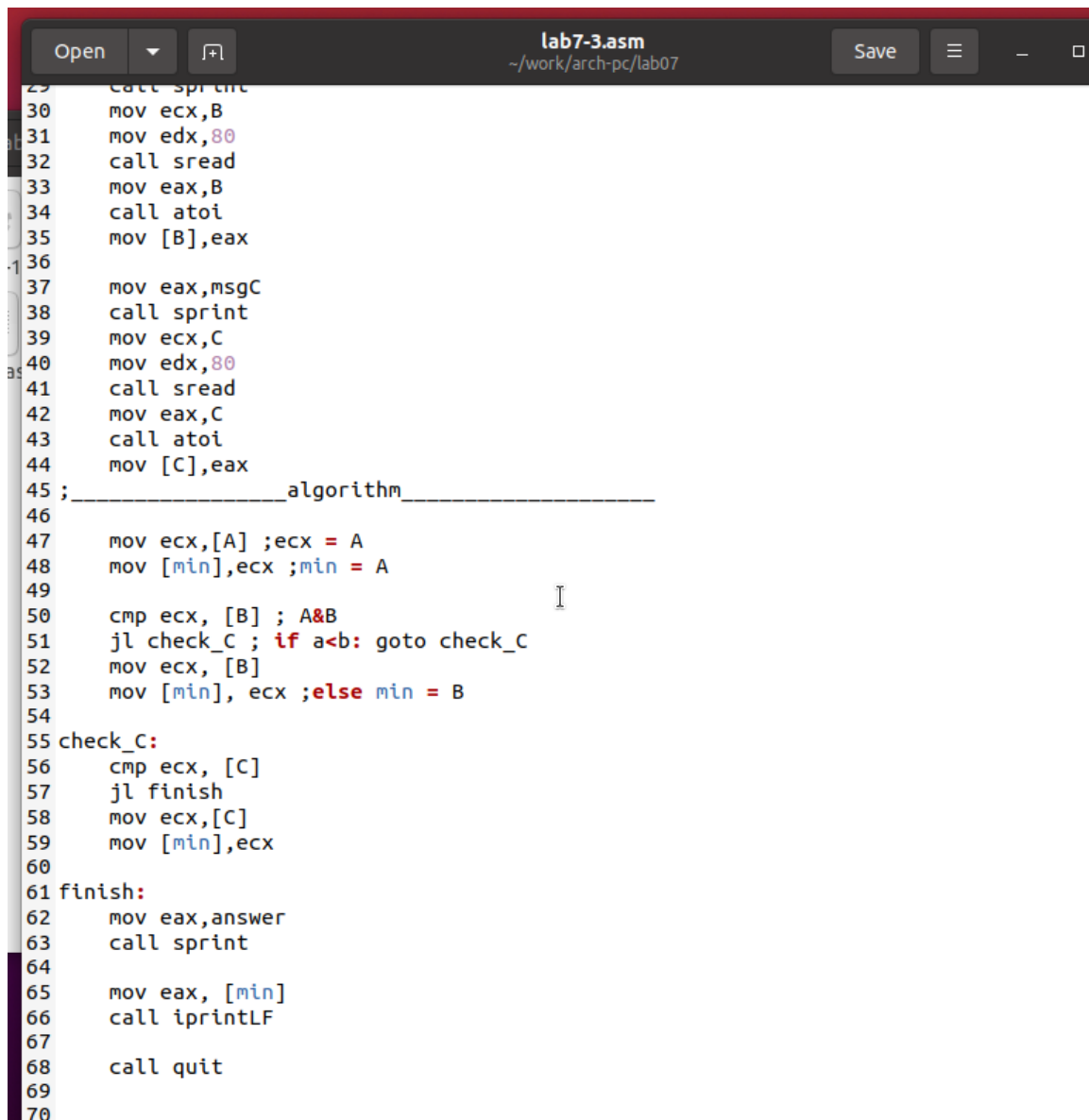
```

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

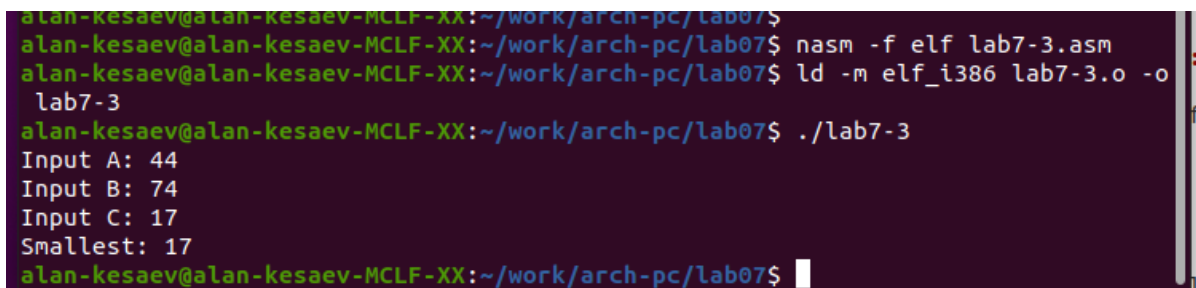
для варианта 16 - 44,74,17



```
lab7-3.asm
~/work/arch-pc/lab07

29  call sprint
30  mov ecx,B
31  mov edx,80
32  call sread
33  mov eax,B
34  call atoi
35  mov [B],eax
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 ;_____algorithm_____
46
47  mov ecx,[A] ;ecx = A
48  mov [min],ecx ;min = A
49
50  cmp ecx, [B] ; A&B
51  jl check_C ; if a<b: goto check_C
52  mov ecx, [B]
53  mov [min], ecx ;else min = B
54
55 check_C:
56  cmp ecx, [C]
57  jl finish
58  mov ecx,[C]
59  mov [min],ecx
60
61 finish:
62  mov eax,answer
63  call sprint
64
65  mov eax, [min]
66  call iprintLF
67
68  call quit
69
70
```

Рис. 2.12: Программа lab7-3.asm



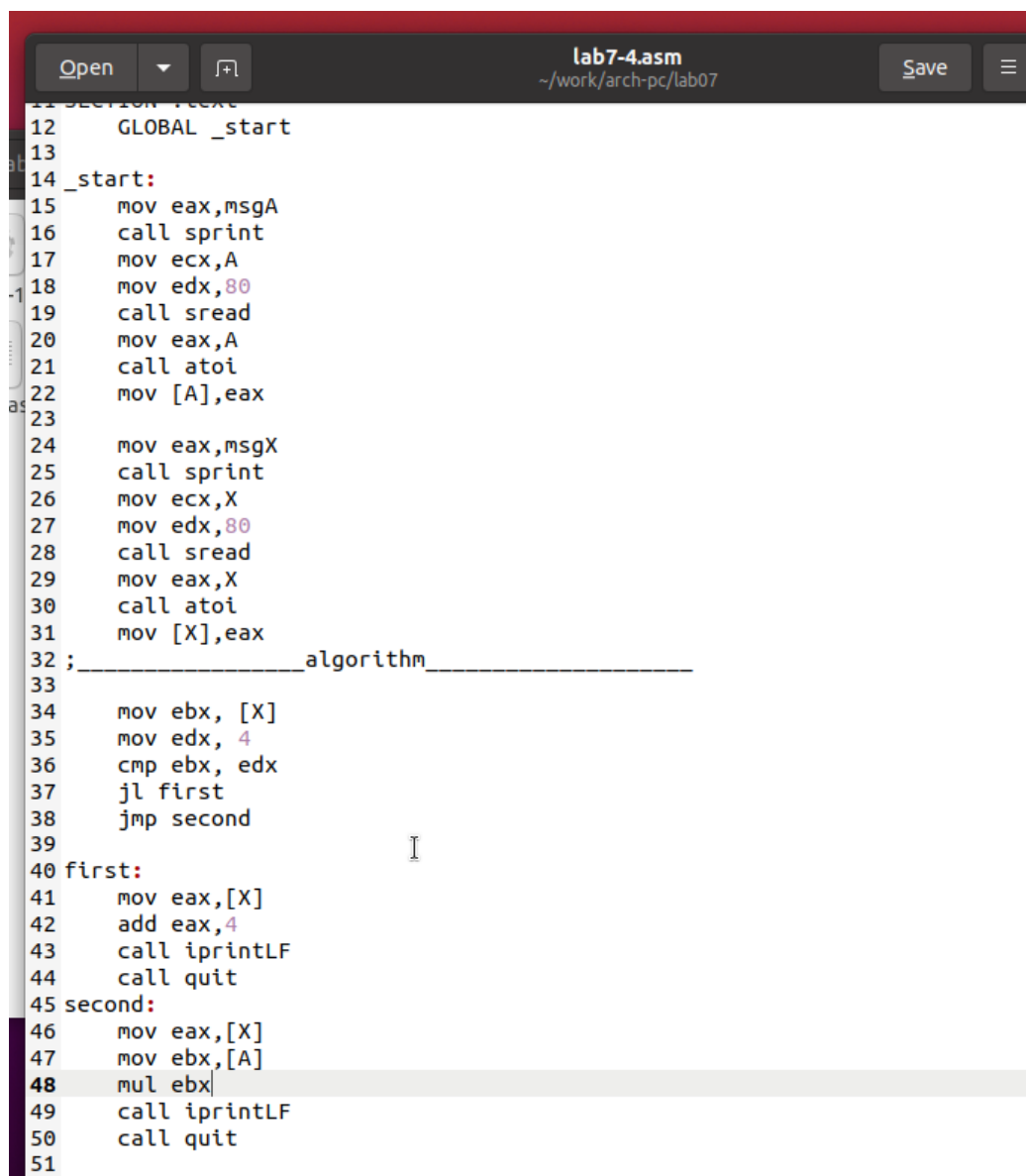
```
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-3.o -o
lab7-3
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ ./lab7-3
Input A: 44
Input B: 74
Input C: 17
Smallest: 17
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$
```

Рис. 2.13: Запуск программы lab7-3.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 16

$$\begin{cases} x + 4, x < 4 \\ ax, x \geq 4 \end{cases}$$



```
11 section .text
12     GLOBAL _start
13
14 _start:
15     mov eax,msgA
16     call sprint
17     mov ecx,A
18     mov edx,80
19     call sread
20     mov eax,A
21     call atoi
22     mov [A],eax
23
24     mov eax,msgX
25     call sprint
26     mov ecx,X
27     mov edx,80
28     call sread
29     mov eax,X
30     call atoi
31     mov [X],eax
32 ; ----- algorithm -----
33
34     mov ebx, [X]
35     mov edx, 4
36     cmp ebx, edx
37     jl first
38     jmp second
39
40 first:
41     mov eax,[X]
42     add eax,4
43     call iprintLF
44     call quit
45 second:
46     mov eax,[X]
47     mov ebx,[A]
48     mul ebx
49     call iprintLF
50     call quit
51
```

Рис. 2.14: Программа lab7-4.asm

```
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$  
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$  
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ nasm -f elf lab7-4.asm  
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-4.o -o  
lab7-4  
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ ./lab7-4  
Input A: 1  
Input X: 1  
5  
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$ ./lab7-4  
Input A: 1  
Input X: 7  
7  
alan-kesaev@alan-kesaev-MCLF-XX:~/work/arch-pc/lab07$
```

Рис. 2.15: Запуск программы lab7-4.asm

3 Выводы

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