

Отчет по лабораторной работе номер 8

Сафин Андрей Алексеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Самостоятельная работа	16
5	Выводы	22

Список иллюстраций

3.1	Создание lab08/lab8-1.asm	7
3.2	Текст lab8-1.asm	8
3.3	Выполнение lab8-1.asm	8
3.4	Текст lab8-1.asm (2)	9
3.5	Выполнение lab8-1.asm (2)	9
3.6	Текст lab8-1.asm (3)	10
3.7	Выполнение lab8-1.asm (3)	10
3.8	Текст lab8-2.asm	11
3.9	Текст lab8-2.asm (продолжение)	12
3.10	Выполнение lab8-2.asm	12
3.11	Листинг lab8-2.asm	13
3.12	Удаление операнда В в lab8-2.asm	14
3.13	Скриншот с созданием листингов	14
3.14	Новый листинг lab8-2.asm	15
4.1	Текст sr.asm	17
4.2	Выполнение sr.asm	18
4.3	Текст sr2.asm	19
4.4	Текст sr2.asm (продолжение)	20
4.5	Выполнение sr2.asm	21

Список таблиц

1 Цель работы

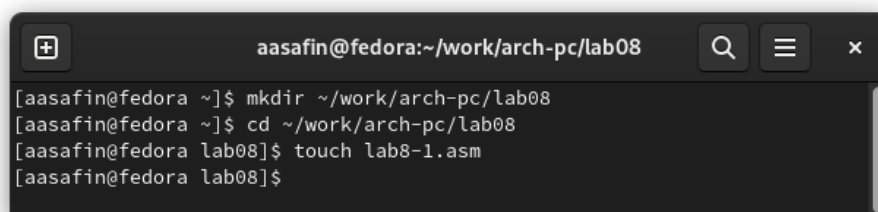
Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Задание

Составить описанные в лабораторной работе программы, использующие команды условного и безусловного переходов.

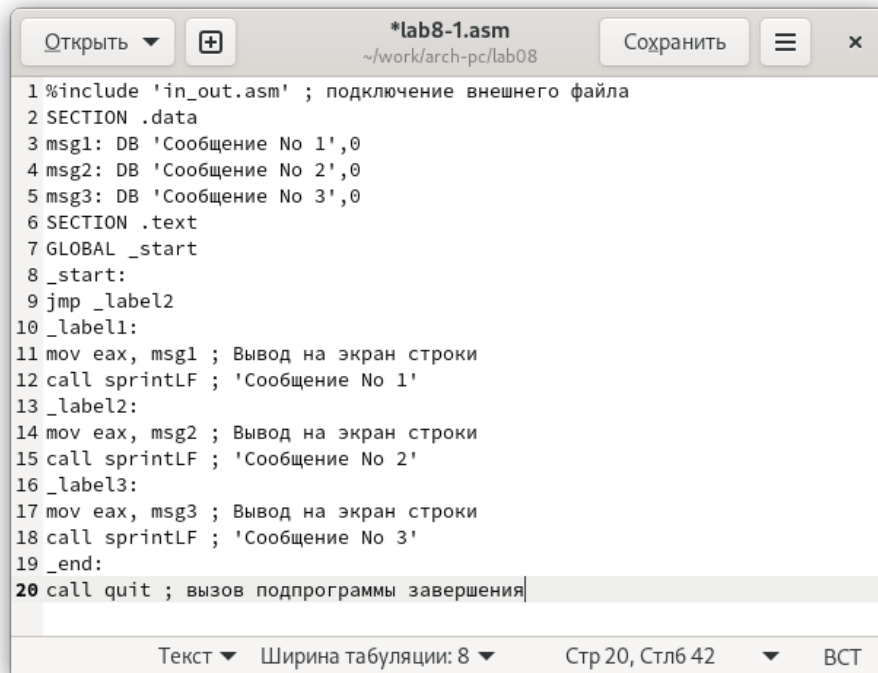
3 Выполнение лабораторной работы

1. Созданы каталог lab08 и файл lab8-1.asm в нем (рис. 3.1).
2. В файле записана программа из листинга 8.1 (рис. 3.2), выполняющая безусловный переход на секцию _label2 (рис. 3.3). Затем программа изменена так (рис. 3.4), чтоб переход затем выполнялся из _label2 в _label1, после чего программа приходила к завершению (рис. 3.5). Итоговый код в lab8-1.asm написан так (рис. 3.6), что программа идет в следующей последовательности: _label3 -> _label2 -> _label1 ->end (рис. 3.7).



```
aasafin@fedora:~/work/arch-pc/lab08
[aasafin@fedora ~]$ mkdir ~/work/arch-pc/lab08
[aasafin@fedora ~]$ cd ~/work/arch-pc/lab08
[aasafin@fedora lab08]$ touch lab8-1.asm
[aasafin@fedora lab08]$
```

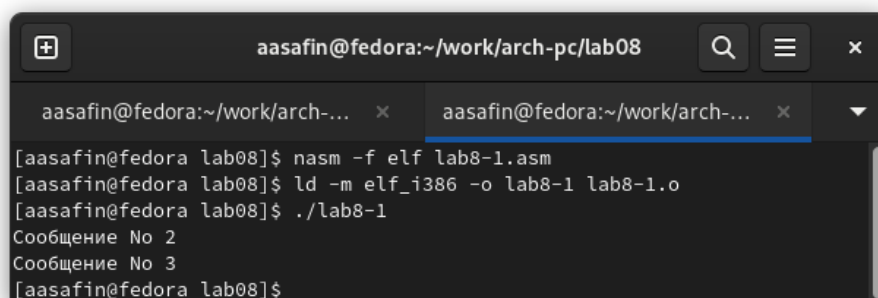
Рис. 3.1: Создание lab08/lab8-1.asm



```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение No 1',0
4 msg2: DB 'Сообщение No 2',0
5 msg3: DB 'Сообщение No 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 No 1'
13 _label2:
14 mov eax, msg2 ; Вывод на экран строки
15 call sprintLF ; 'Сообщение No 2'
16 _label3:
17 mov eax, msg3 ; Вывод на экран строки
18 call sprintLF ; 'Сообщение No 3'
19 _end:
20 call quit ; вызов подпрограммы завершения
```

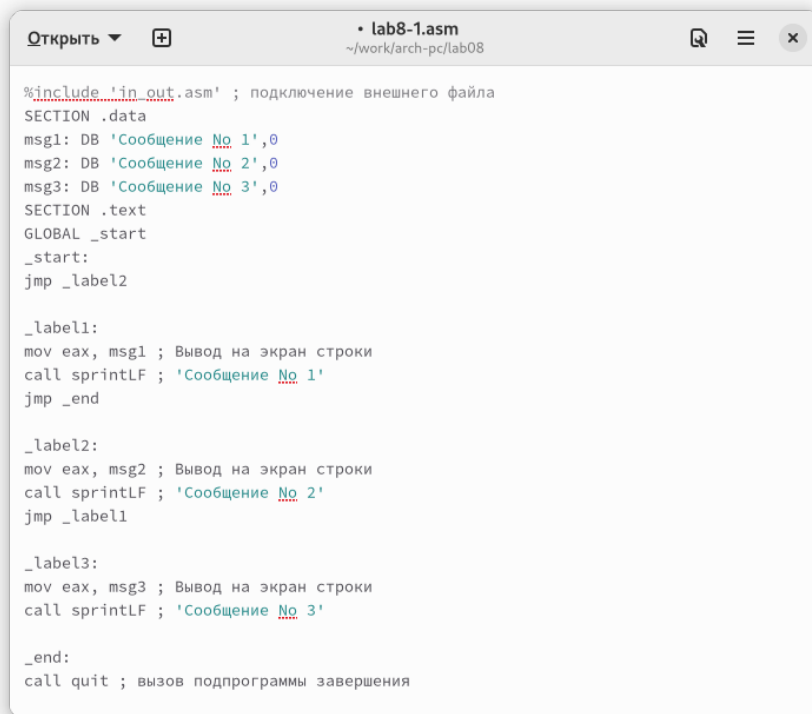
Текст Ширина табуляции: 8 Стр 20, Стлб 42 ВСТ

Рис. 3.2: Текст lab8-1.asm



```
aasafin@fedora:~/work/arch-pc/lab08
aasafin@fedora:~/work/arch-... x aasafin@fedora:~/work/arch-... x
[aasafin@fedora lab08]$ nasm -f elf lab8-1.asm
[aasafin@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[aasafin@fedora lab08]$ ./lab8-1
Сообщение No 2
Сообщение No 3
[aasafin@fedora lab08]$
```

Рис. 3.3: Выполнение lab8-1.asm



```
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2

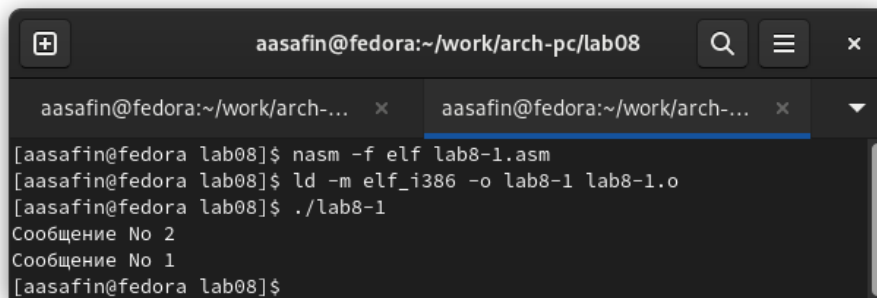
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение No 1'
jmp _end

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

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

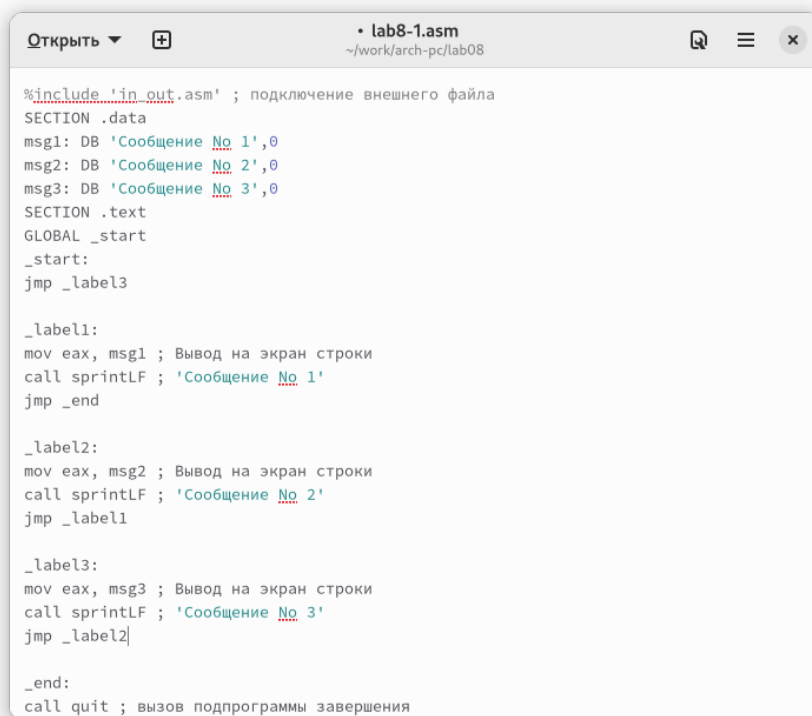
_end:
call quit ; вызов подпрограммы завершения
```

Рис. 3.4: Текст lab8-1.asm (2)



```
aasafin@fedora:~/work/arch-pc/lab08
aasafin@fedora lab08$ nasm -f elf lab8-1.asm
aasafin@fedora lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
aasafin@fedora lab08$ ./lab8-1
Сообщение No 2
Сообщение No 1
aasafin@fedora lab08$
```

Рис. 3.5: Выполнение lab8-1.asm (2)



```
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label3

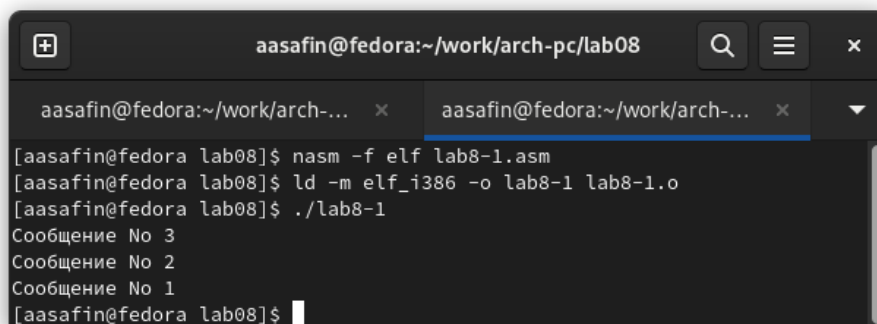
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение No 1'
jmp _end

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

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

_end:
call quit ; вызов подпрограммы завершения
```

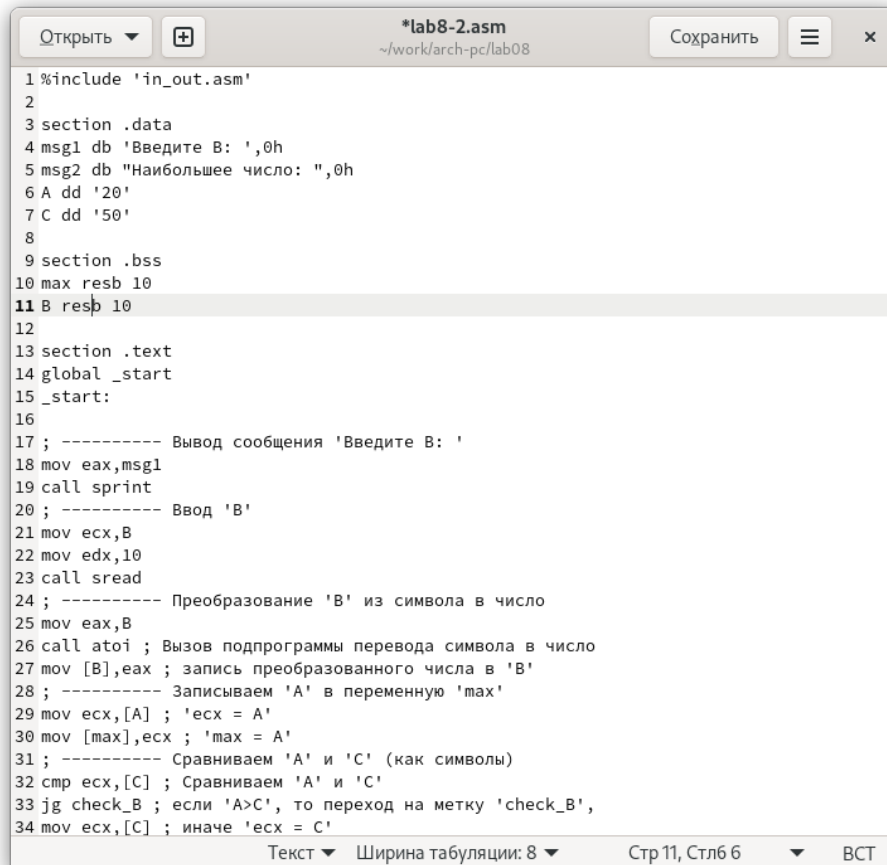
Рис. 3.6: Текст lab8-1.asm (3)



```
aasafin@fedora:~/work/arch-pc/lab08
aasafin@fedora lab08$ nasm -f elf lab8-1.asm
aasafin@fedora lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
aasafin@fedora lab08$ ./lab8-1
Сообщение No 3
Сообщение No 2
Сообщение No 1
aasafin@fedora lab08$
```

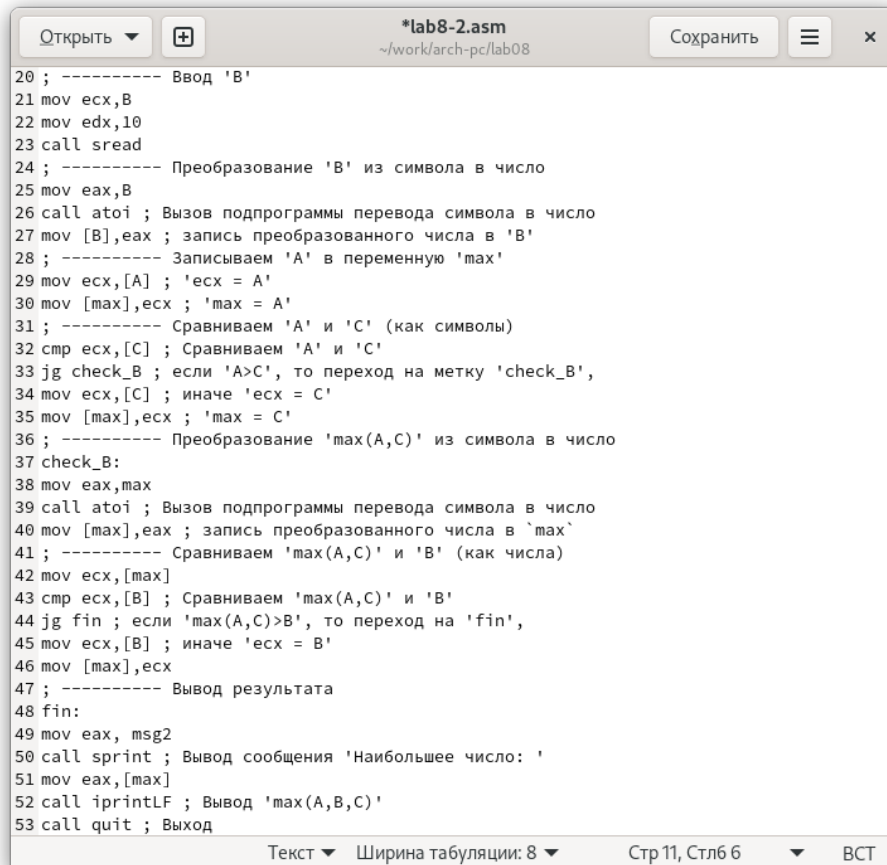
Рис. 3.7: Выполнение lab8-1.asm (3)

3. Создан файл lab8-2.asm, в него введена программа из листинга 8.3 (рис. 3.8-3.9), выполняющая сравнение чисел 20 и 50 с введенным, выводя максимальное (рис. 3.10).



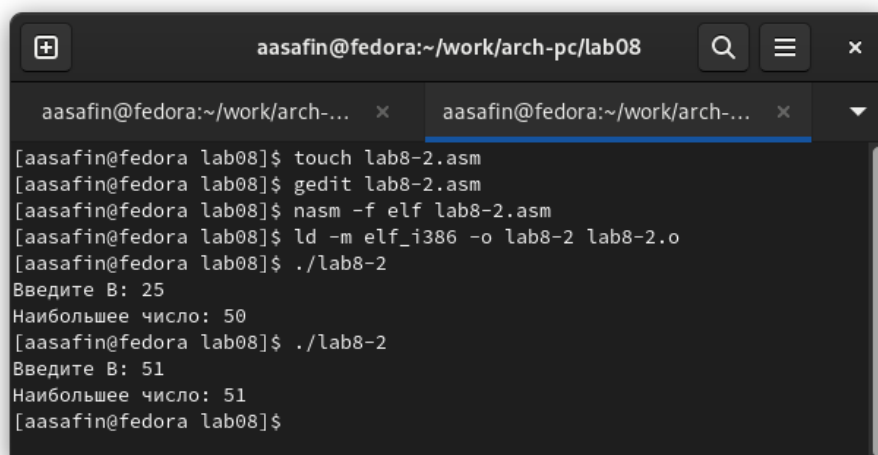
```
1 %include 'in_out.asm'
2
3 section .data
4 msg1 db 'Введите B: ',0h
5 msg2 db "Наибольшее число: ",0h
6 A dd '20'
7 C dd '50'
8
9 section .bss
10 max resb 10
11 B resb 10
12
13 section .text
14 global _start
15 _start:
16
17 ; ----- Вывод сообщения 'Введите B: '
18 mov eax,msg1
19 call sprint
20 ; ----- Ввод 'B'
21 mov ecx,B
22 mov edx,10
23 call sread
24 ; ----- Преобразование 'B' из символа в число
25 mov eax,B
26 call atoi ; Вызов подпрограммы перевода символа в число
27 mov [B],eax ; запись преобразованного числа в 'B'
28 ; ----- Записываем 'A' в переменную 'max'
29 mov ecx,[A] ; 'ecx = A'
30 mov [max],ecx ; 'max = A'
31 ; ----- Сравниваем 'A' и 'C' (как символы)
32 cmp ecx,[C] ; Сравниваем 'A' и 'C'
33 jg check_B ; если 'A>C', то переход на метку 'check_B',
34 mov ecx,[C] ; иначе 'ecx = C'
```

Рис. 3.8: Текст lab8-2.asm



```
20 ; ----- Ввод 'B'
21 mov ecx,B
22 mov edx,10
23 call sread
24 ; ----- Преобразование 'B' из символа в число
25 mov eax,B
26 call atoi ; Вызов подпрограммы перевода символа в число
27 mov [B],eax ; запись преобразованного числа в 'B'
28 ; ----- Записываем 'A' в переменную 'max'
29 mov ecx,[A] ; 'ecx = A'
30 mov [max],ecx ; 'max = A'
31 ; ----- Сравниваем 'A' и 'C' (как символы)
32 cmp ecx,[C] ; Сравниваем 'A' и 'C'
33 jg check_B ; если 'A>C', то переход на метку 'check_B',
34 mov ecx,[C] ; иначе 'ecx = C'
35 mov [max],ecx ; 'max = C'
36 ; ----- Преобразование 'max(A,C)' из символа в число
37 check_B:
38 mov eax,max
39 call atoi ; Вызов подпрограммы перевода символа в число
40 mov [max],eax ; запись преобразованного числа в 'max'
41 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
42 mov ecx,[max]
43 cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
44 jg fin ; если 'max(A,C)>B', то переход на 'fin',
45 mov ecx,[B] ; иначе 'ecx = B'
46 mov [max],ecx
47 ; ----- Вывод результата
48 fin:
49 mov eax, msg2
50 call sprint ; Вывод сообщения 'Наибольшее число: '
51 mov eax,[max]
52 call iprintLF ; Вывод 'max(A,B,C)'
53 call quit ; Выход
```

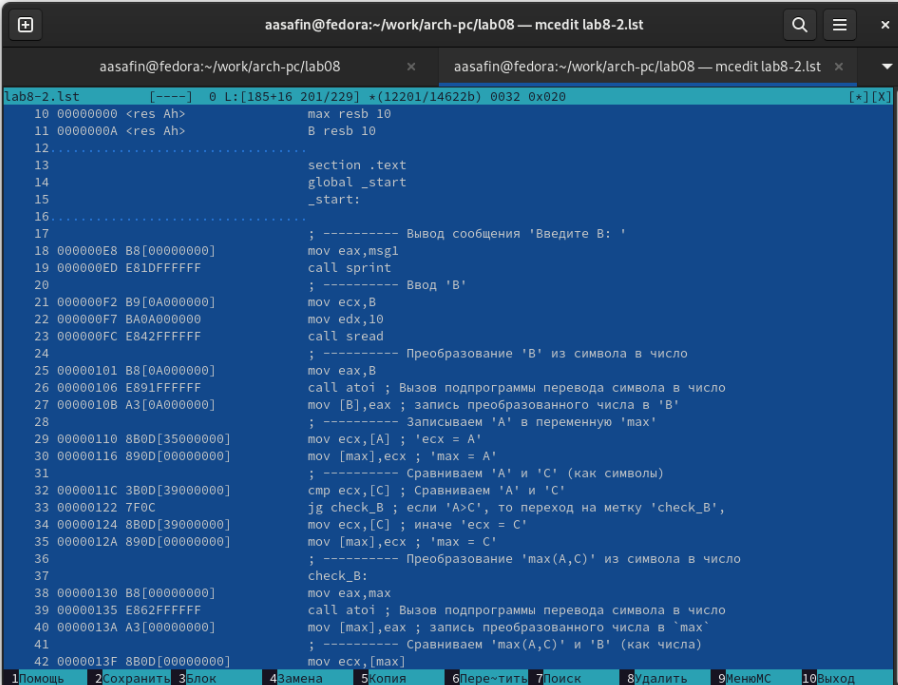
Рис. 3.9: Текст lab8-2.asm (продолжение)



```
aasafin@fedora:~/work/arch-pc/lab08
[aasafin@fedora lab08]$ touch lab8-2.asm
[aasafin@fedora lab08]$ gedit lab8-2.asm
[aasafin@fedora lab08]$ nasm -f elf lab8-2.asm
[aasafin@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[aasafin@fedora lab08]$ ./lab8-2
Введите B: 25
Наибольшее число: 50
[aasafin@fedora lab08]$ ./lab8-2
Введите B: 51
Наибольшее число: 51
[aasafin@fedora lab08]$
```

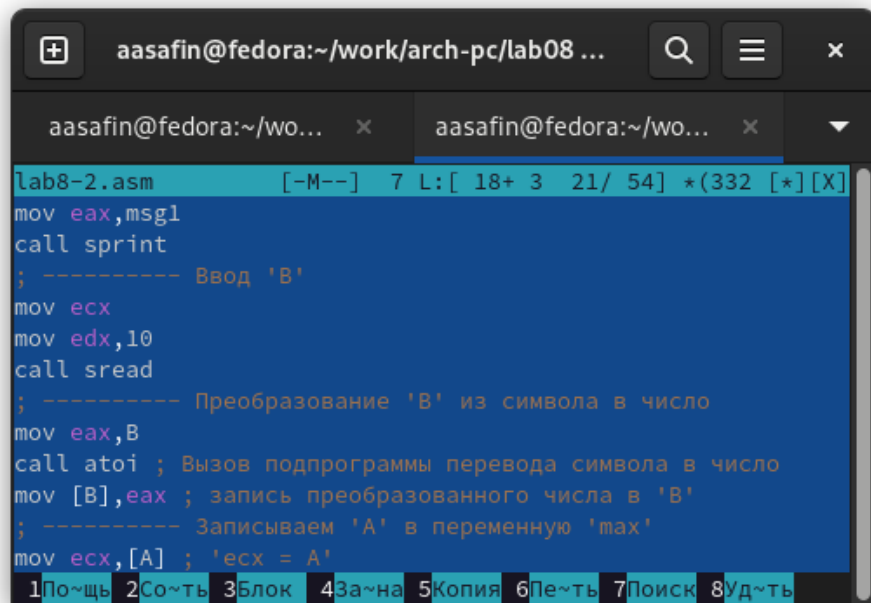
Рис. 3.10: Выполнение lab8-2.asm

4. Создан листинг программы lab8-2.asm (рис. 3.11). В рассматриваемом фрагменте 20-ая строка имеет лишь номер, так как в ней нет кода, есть лишь комментарий. В 21-ой и 22-ой строках помимо номера и самого текста файла отображаются адреса (значения отступа от начала секции в машинном коде, записанные в 16-ной системе - 000000F2 и 000000F7 соответственно), а также сам машинный код, переведенный в 16-ную систему счисления (B8[00000000] и E81DFFFFFF соответственно). При удалении операнда B из команды, соответствующей 21-ой строке листинга (рис. 3.12), после неё в новом листинге появляется сообщение об ошибке (рис. 3.14).



```
lab8-2.lst  [-----]  0 L:[185+16 201/229] *(12201/14622b) 0032 0x020  [*][X]
10 00000000 <res Ah>          max resb 10
11 0000000A <res Ah>          B resb 10
12 .....
13 .....                      section .text
14 .....                      global _start
15 .....                      _start:
16 .....
17 .....                      ; ----- Вывод сообщения 'Введите B: '
18 000000E8 B8[00000000]      mov eax,msg1
19 000000ED E81DFFFFFF      call sprint
20 .....                      ; ----- Ввод 'B'
21 000000F2 B9[0A000000]      mov ecx,B
22 000000F7 BA0A000000      mov edx,10
23 000000FC E842FFFFFF      call sread
24 .....                      ; ----- Преобразование 'B' из символа в число
25 00000101 B8[0A000000]      mov eax,B
26 00000106 E891FFFFFF      call atoi ; Вызов подпрограммы перевода символа в число
27 0000010B A3[0A000000]      mov [B],eax ; запись преобразованного числа в 'B'
28 .....                      ; ----- Записываем 'A' в переменную 'max'
29 00000110 8B0D[35000000]      mov ecx,[A] ; 'ecx = A'
30 00000116 890D[00000000]      mov [max],ecx ; 'max = A'
31 .....                      ; ----- Сравниваем 'A' и 'C' (как символы)
32 0000011C 3B0D[39000000]      cmp ecx,[C] ; Сравниваем 'A' и 'C'
33 00000122 7F0C              jg check_B ; если 'A>C', то переход на метку 'check_B',
34 00000124 8B0D[39000000]      mov ecx,[C] ; иначе 'ecx = C'
35 0000012A 890D[00000000]      mov [max],ecx ; 'max = C'
36 .....                      ; ----- Преобразование 'max(A,C)' из символа в число
37 .....                      check_B:
38 00000130 B8[00000000]      mov eax,max
39 00000135 E862FFFFFF      call atoi ; Вызов подпрограммы перевода символа в число
40 0000013A A3[00000000]      mov [max],eax ; запись преобразованного числа в 'max'
41 .....                      ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
42 0000013F 8B0D[00000000]      mov ecx,[max]
```

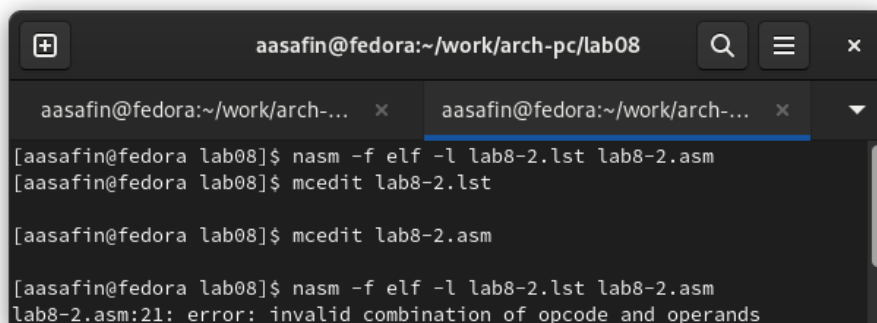
Рис. 3.11: Листинг lab8-2.asm



```
lab8-2.asm [-M--] 7 L: [ 18+ 3 21/ 54] *(332 [*] [X])
mov eax,msg1
call sprint
; ----- Ввод 'B'
mov ecx
mov edx,10
call sread
; ----- Преобразование 'B' из символа в число
mov eax,B
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'
```

1По~щъ 2Со~ть 3Блок 4За~на 5Копия 6Пе~ть 7Поиск 8Уд~ть

Рис. 3.12: Удаление операнда B в lab8-2.asm



```
aasafin@fedora:~/work/arch-pc/lab08
aasafin@fedora:~/work/arch-... x aasafin@fedora:~/work/arch-... x
[aasafin@fedora lab08]$ nasm -f elf -l lab8-2.lst lab8-2.asm
[aasafin@fedora lab08]$ mcedit lab8-2.lst
[aasafin@fedora lab08]$ mcedit lab8-2.asm
[aasafin@fedora lab08]$ nasm -f elf -l lab8-2.lst lab8-2.asm
lab8-2.asm:21: error: invalid combination of opcode and operands
```

Рис. 3.13: Скриншот с созданием листингов

```
lab8-2.lst [B---] 89 L:[188+15 203/230] *(12541/14711b) 0010 0x00A [*][X]
13                                     section .text
14                                     global _start
15                                     _start:
16                                     .....
17                                     ; ----- Вывод сообщения 'Введите B: '
18 000000E8 B8[00000000]      mov eax,msg1
19 000000ED E81DFFFFFF      call sprint
20                                     ; ----- Ввод 'B'
21                                     mov ecx
21 ***** error: invalid combination of opcode and operands
22 000000F2 BA0A000000      mov edx,10
23 000000F7 E847FFFFFF      call sread
24                                     ; ----- Преобразование 'B' из символа в число
25 000000FC B8[0A000000]      mov eax,B
26 00000101 E896FFFFFF      call atoi ; Вызов подпрограммы перевода символа в число
27 00000106 A3[0A000000]      mov [B],eax ; запись преобразованного числа в 'B'
28                                     ; ----- Записываем 'A' в переменную 'max'
29 0000010B 8B0D[35000000]      mov ecx,[A] ; 'ecx = A'
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перечитать 7Поиск 8Удалить 9МенюМС 10Выход

Рис. 3.14: Новый листинг lab8-2.asm

4 Самостоятельная работа

1. Создана программа, сравнивающая числа 84,32,77, соответствующие варианту, вычисленному по номеру студенческого билета (13), и выводящая наименьшее (рис. 4.1). Результат корректен (рис. 4.2).



```
%include 'in_out.asm'

section .data
msg db "Наименьшее число: ",0h
A dd 84
B dd 32
C dd 77

section .bss
max resb 10

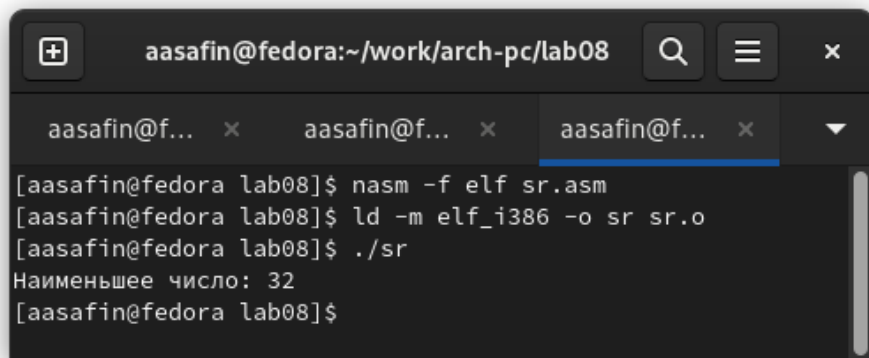
section .text
global _start
_start:

mov ecx,[A]
mov [max],ecx
cmp ecx,[C]
jl check_B
mov ecx,[C]
mov [max],ecx

check_B:
mov ecx,[max]
cmp ecx,[B]
jl fin
mov ecx,[B]
mov [max],ecx

fin:
mov eax,msg
call sprint
mov eax,[max]
call iprintLF
call quit
```

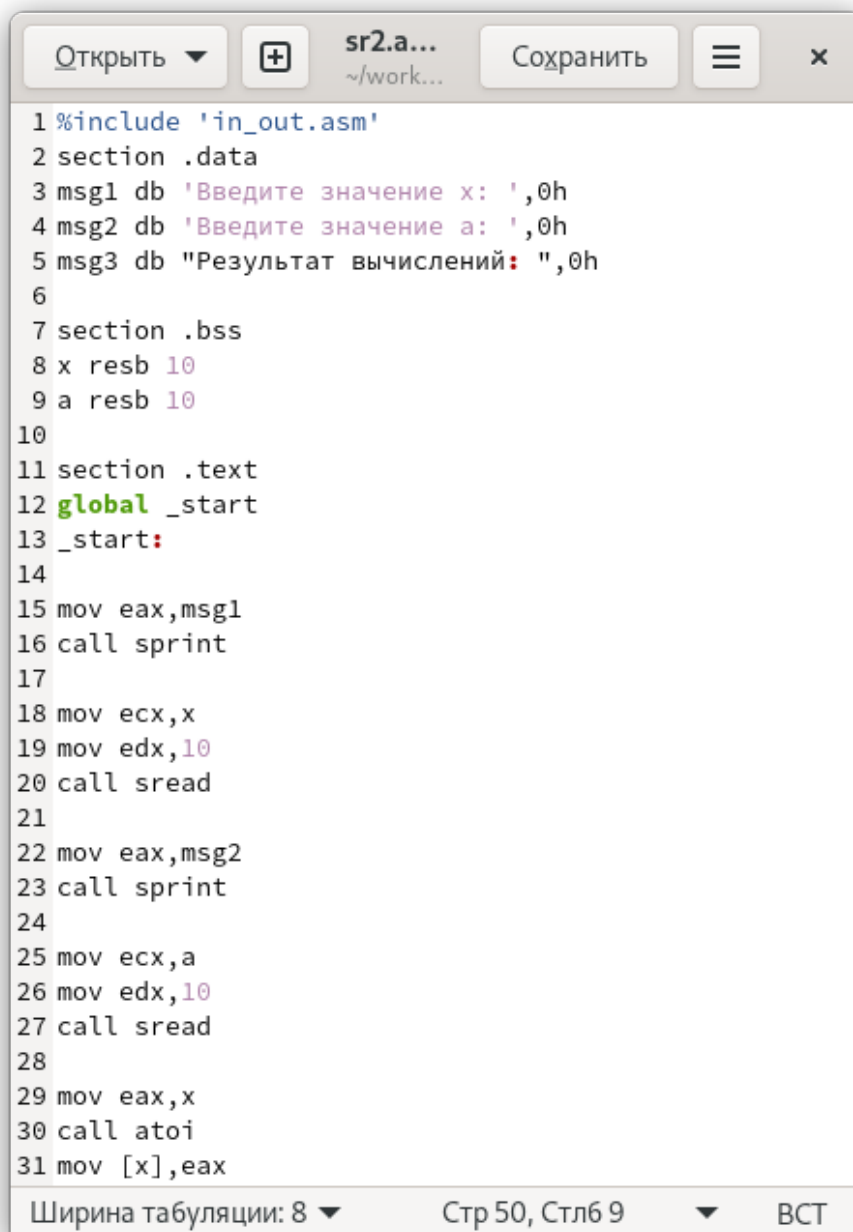
Рис. 4.1: Текст sr.asm

A terminal window with a dark background. The title bar shows 'aasafin@fedora:~/work/arch-pc/lab08'. There are three tabs, all labeled 'aasafin@f...'. The terminal content shows the following commands and output:

```
[aasafin@fedora lab08]$ nasm -f elf sr.asm
[aasafin@fedora lab08]$ ld -m elf_i386 -o sr sr.o
[aasafin@fedora lab08]$ ./sr
Наименьшее число: 32
[aasafin@fedora lab08]$
```

Рис. 4.2: Выполнение sr.asm

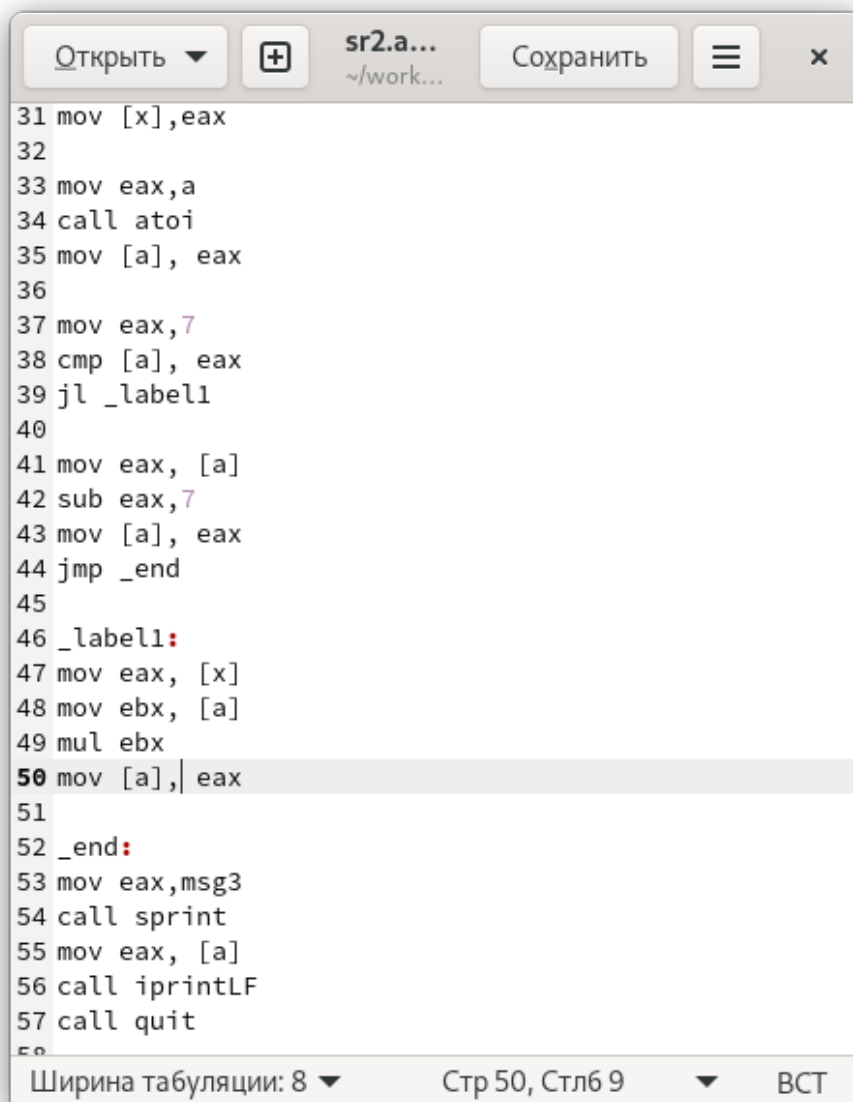
2. Создана программа, вычисляющая для введенных значений x и a результат функции, соответствующей варианту 13, и выводящая результат на экран (рис. 4.3-4.4). Результат для пробных значений корректен (рис. 4.5).



```
1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите значение x: ',0h
4 msg2 db 'Введите значение a: ',0h
5 msg3 db "Результат вычислений: ",0h
6
7 section .bss
8 x resb 10
9 a resb 10
10
11 section .text
12 global _start
13 _start:
14
15 mov eax,msg1
16 call sprint
17
18 mov ecx,x
19 mov edx,10
20 call sread
21
22 mov eax,msg2
23 call sprint
24
25 mov ecx,a
26 mov edx,10
27 call sread
28
29 mov eax,x
30 call atoi
31 mov [x],eax
```

Ширина табуляции: 8 Стр 50, Стлб 9 ВСТ

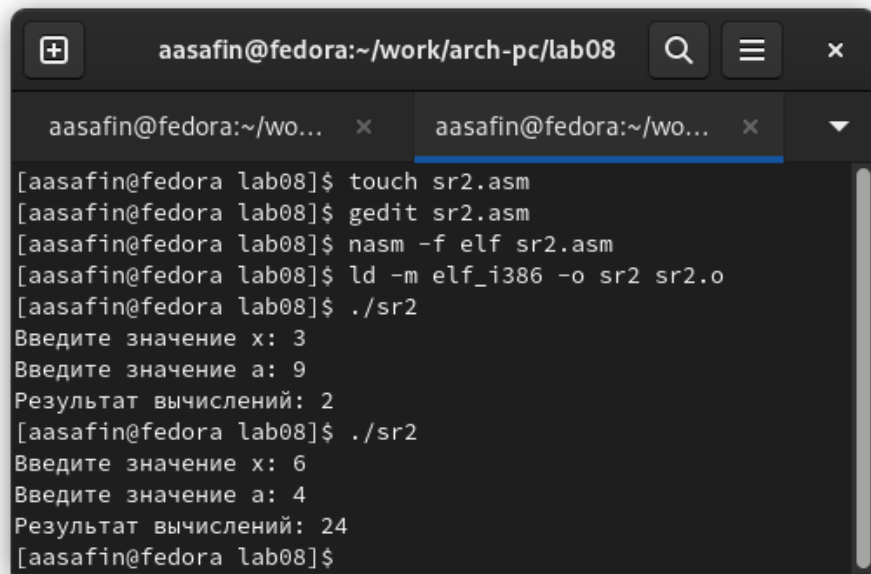
Рис. 4.3: Текст sr2.asm



```
31 mov [x],eax
32
33 mov eax,a
34 call atoi
35 mov [a], eax
36
37 mov eax,7
38 cmp [a], eax
39 jl _label1
40
41 mov eax, [a]
42 sub eax,7
43 mov [a], eax
44 jmp _end
45
46 _label1:
47 mov eax, [x]
48 mov ebx, [a]
49 mul ebx
50 mov [a],| eax
51
52 _end:
53 mov eax,msg3
54 call sprint
55 mov eax, [a]
56 call iprintLF
57 call quit
58
```

Ширина табуляции: 8 Стр 50, Стлб 9 ВСТ

Рис. 4.4: Текст sr2.asm (продолжение)

A terminal window titled 'aasafin@fedora:~/work/arch-pc/lab08' with search, menu, and close icons. It shows two tabs, both labeled 'aasafin@fedora:~/wo...'. The terminal output is as follows:

```
[aasafin@fedora lab08]$ touch sr2.asm
[aasafin@fedora lab08]$ gedit sr2.asm
[aasafin@fedora lab08]$ nasm -f elf sr2.asm
[aasafin@fedora lab08]$ ld -m elf_i386 -o sr2 sr2.o
[aasafin@fedora lab08]$ ./sr2
Введите значение x: 3
Введите значение a: 9
Результат вычислений: 2
[aasafin@fedora lab08]$ ./sr2
Введите значение x: 6
Введите значение a: 4
Результат вычислений: 24
[aasafin@fedora lab08]$
```

Рис. 4.5: Выполнение sr2.asm

5 Выводы

Требуемые программы созданы и выполнены корректно. Команды условного и безусловного перехода рассмотрены и применены в рамках самостоятельной работы.