

Отчет по лабораторной работе №8

дисциплина: Архитектура компьютера

Колобова Елизавета Андреевна гр. НММбд-01

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Задание для самостоятельной работы	17
3	Выводы	22

Список иллюстраций

2.1	Рис. 1. Создание каталога для лаб. работы №8 и файла lab8-1.asm .	6
2.2	Рис. 2. Ввод текста из листинга 8.1	7
2.3	Рис. 3. Компоновка и запуск файла lab8-1	8
2.4	Рис. 4. Изменение текста программы	8
2.5	Рис. 5. Компоновка и запуск измененного файла	9
2.6	Рис. 6. Изменение текста программы	9
2.7	Рис. 7. Компоновка и запуск измененного файла	10
2.8	Рис. 8. Ввод текста из листинга 8.3	11
2.9	Рис. 9. Ввод текста из листинга 8.3	12
2.10	Рис. 10. Компоновка и запуск файла	12
2.11	Рис. 11. Создание файла листинга	13
2.12	Рис. 12. Окно файла листинга	13
2.13	Рис. 13. Окно файла листинга	14
2.14	Рис. 14. Окно файла листинга	15
2.15	Рис. 15. Окно файла листинга	16
2.16	Рис. 16. Окно файла листинга	16
2.17	Рис. 17. Удаление операнда	17
2.18	Рис. 18. Удаление операнда	17
2.19	Рис. 19. Создание листинга	17
2.20	Рис. 20. Текст программы	18
2.21	Рис. 21. Компоновка и запуск файла	18
2.22	Рис. 22. Текст программы	19
2.23	Рис. 23. Текст программы	20
2.24	Рис. 24. Компоновка и запуск файла	21

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.

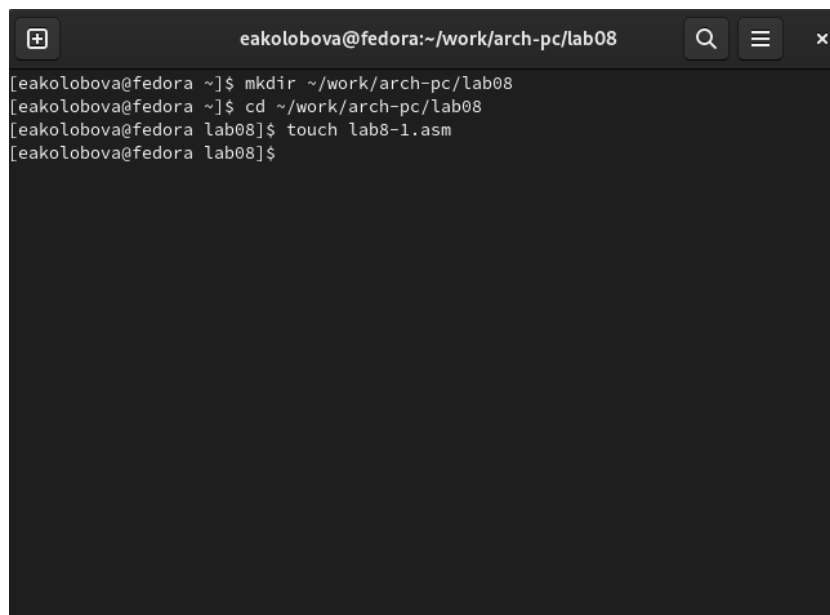
2 Выполнение лабораторной работы

1. Создаем каталог для программ лабораторной работы No 8, переходим в него и создаем файл lab8-1.asm (рис. 2.1):

```
mkdir ~/work/arch-pc/lab08
```

```
cd ~/work/arch-pc/lab08
```

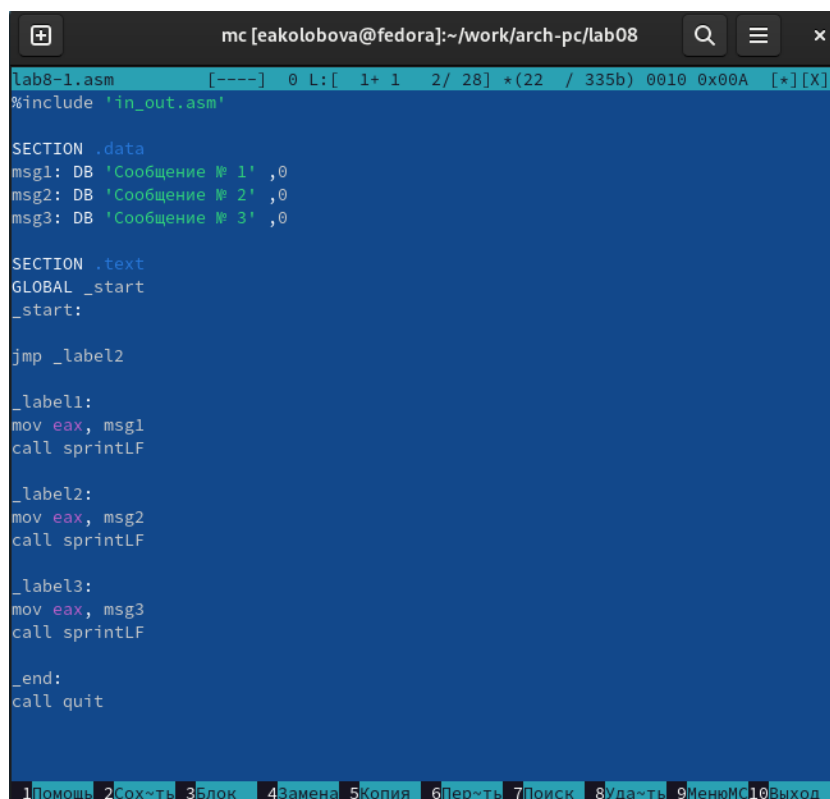
```
touch lab8-1.asm
```

A screenshot of a terminal window with a dark background. The title bar at the top shows the user 'eakolobova' on a 'fedora' machine, in the directory '~/work/arch-pc/lab08'. There are search, menu, and close icons on the right. The terminal shows four lines of commands and their outputs: 1. '[eakolobova@fedora ~]\$ mkdir ~/work/arch-pc/lab08' followed by a new line. 2. '[eakolobova@fedora ~]\$ cd ~/work/arch-pc/lab08' followed by a new line. 3. '[eakolobova@fedora lab08]\$ touch lab8-1.asm' followed by a new line. 4. '[eakolobova@fedora lab08]\$' followed by a new line.

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

Рис. 2.1: Рис. 1. Создание каталога для лаб. работы №8 и файла lab8-1.asm

2. Рассмотрим пример программы с использованием инструкции jmp. Введем в файл lab8-1.asm текст программы из листинга 8.1. (рис. 2.2).



```
lab8-1.asm  [----]  0 L: [ 1+ 1 2/ 28] *(22 / 335b) 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 sprintLF

_label2:
mov eax, msg2
call sprintLF

_label3:
mov eax, msg3
call sprintLF

_end:
call quit
```

1Помощь 2Сох-ты 3Блок 4Замена 5Копия 6Пер-ты 7Поиск 8Уда-ты 9МенюMC10Выход

Рис. 2.2: Рис. 2. Ввод текста из листинга 8.1

3. Создаем исполняемый файл и запускаем его. (рис. 2.3)

```
nasm -f elf lab8-1.asm
ld -m elf_i386 -o lab8-1 lab7-8.o
./lab8-1
```

Ожидаемый результат:

```
user@dk4n31:~$ ./lab8-1
Сообщение No 2
Сообщение No 3
user@dk4n31:~$
```

```

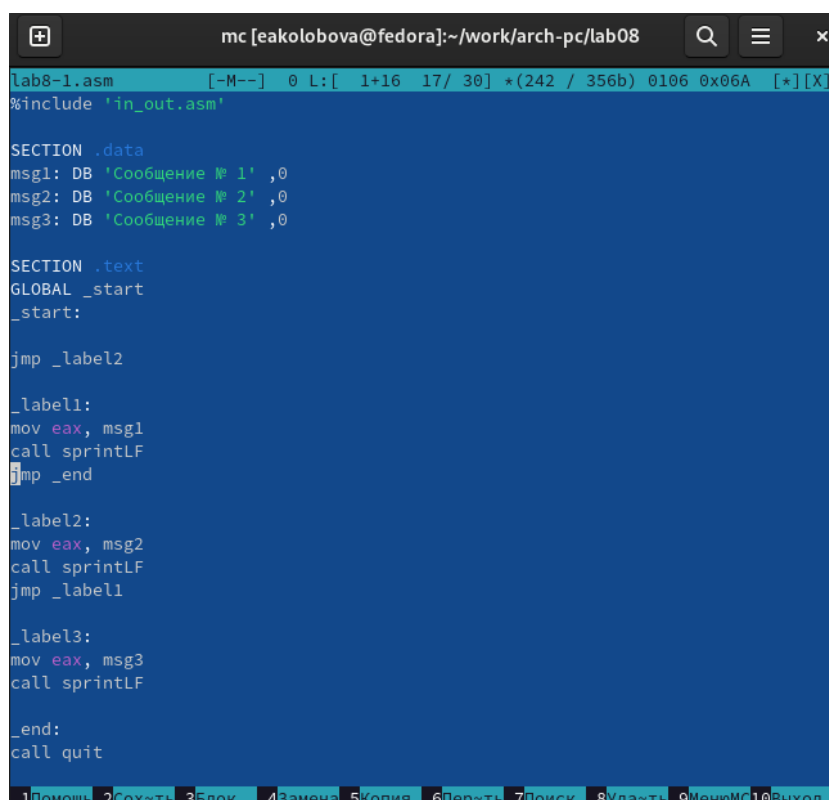
[eakolobova@fedora lab08]$ nasm -f elf lab8-1.asm
[eakolobova@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[eakolobova@fedora lab08]$ ./lab8-1
Сообщение № 2
Сообщение № 3
[eakolobova@fedora lab08]$

```

Рис. 2.3: Рис. 3. Компоновка и запуск файла lab8-1

4. Далее изменим текст программы таким образом, чтобы она выводила сначала 'Сообщение No 2', потом 'Сообщение No 1' и завершала работу. Для этого в текст программы после вывода сообщения No 2 добавим инструкцию `jmp` с меткой `_label1` и после вывода сообщения No 1 добавим инструкцию `jmp` с меткой `_end`. Изменим текст программы в соответствии с листингом 8.2 и вместо символов, запишем в регистры числа. (рис. 2.4):

Создаем исполняемый файл и запускаем его. (рис. 2.5)



```

lab8-1.asm  [-M--]  0  L: [ 1+16 17/ 30] *(242 / 356b) 0106 0x06A  [*][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 sprintLF
jmp _end

_label2:
mov eax, msg2
call sprintLF
jmp _label1

_label3:
mov eax, msg3
call sprintLF

_end:
call quit

```

Рис. 2.4: Рис. 4. Изменение текста программы


```

[eakolobova@fedora lab08]$ nasm -f elf lab8-1.asm
[eakolobova@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[eakolobova@fedora lab08]$ ./lab8-1
Сообщение № 2
Сообщение № 1
[eakolobova@fedora lab08]$

```

Рис. 2.5: Рис. 5. Компоновка и запуск измененного файла

5. Изменим текст программы добавив или изменив инструкции `jmp`, чтобы вывод программы был следующим (рис. 2.6, 2.7):

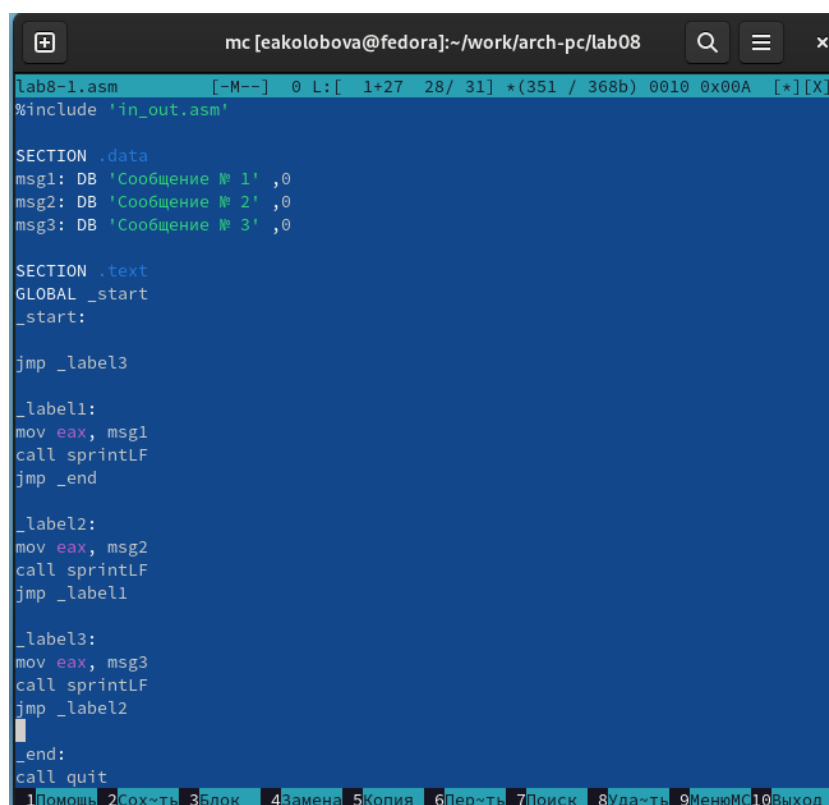
```
user@dk4n31:~$ ./lab8-1
```

```
Сообщение No 3
```

```
Сообщение No 2
```

```
Сообщение No 1
```

```
user@dk4n31:~$
```



```

lab8-1.asm  [-M--]  0 L: [ 1+27 28/ 31] *(351 / 368b) 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: Рис. 6. Изменение текста программы

```
[eakolobova@fedora lab08]$ nasm -f elf lab8-1.asm
[eakolobova@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[eakolobova@fedora lab08]$ ./lab8-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
```

Рис. 2.7: Рис. 7. Компоновка и запуск измененного файла

5. Рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: А, В и С. Значения для А и С задаются в программе, значение В вводится с клавиатуры. Создаем файл lab8-2.asm в каталоге ~/work/arch-рс/lab08. и вводим в него текст программы из листинга 8.3. (рис. 2.8, 2.9, 2.10)

```
lab8-2.asm [----] 27 L: [ 1+ 3 4/ 52] *(111 / 557b) 0034 0x022 [*] [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,[A]
mov [max],ecx

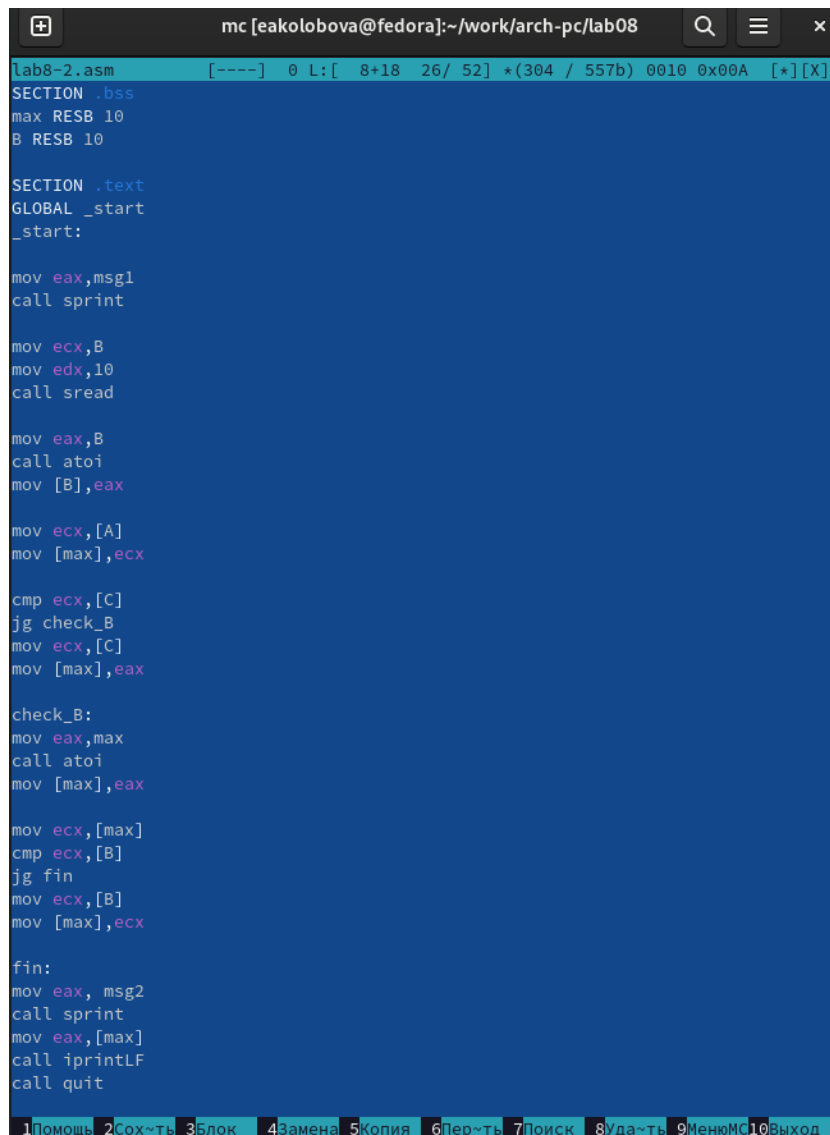
cmp ecx,[C]
jg check_B
mov ecx,[C]
mov [max],eax

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

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

1Помощь 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход
```

Рис. 2.8: Рис. 8. Ввод текста из листинга 8.3



```
lab8-2.asm  [----]  0 L: [ 8+18 26/ 52] *(304 / 557b) 0010 0x00A  [*][X]
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,[A]
mov [max],ecx

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

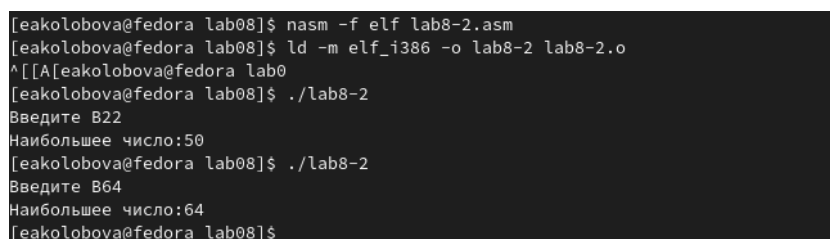
check_B:
mov eax,max
call atoi
mov [max],eax

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
```

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

Рис. 2.9: Рис. 9. Ввод текста из листинга 8.3



```
[eakolobova@fedora lab08]$ nasm -f elf lab8-2.asm
[eakolobova@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
^[[A[eakolobova@fedora lab0
[eakolobova@fedora lab08]$ ./lab8-2
Введите B22
Наибольшее число:50
[eakolobova@fedora lab08]$ ./lab8-2
Введите B64
Наибольшее число:64
[eakolobova@fedora lab08]$
```

Рис. 2.10: Рис. 10. Компоновка и запуск файла

6. Создадим файл листинга для программы из файла lab8-2.asm (рис. 2.11)

```
nasm -f elf -l lab8-2.lst lab8-2.asm
```

Откроем файл листинга lab8-2.lst с помощью любого текстового редактора:

```
mcedit lab8-2.lst
```

(рис. 2.12, 2.13, 2.14, 2.15)

```
[eakolobova@fedora lab08]$ nasm -f elf -l lab8-2.lst lab8-2.asm
[eakolobova@fedora lab08]$ mcedit lab8-2.lst
```

Рис. 2.11: Рис. 11. Создание файла листинга

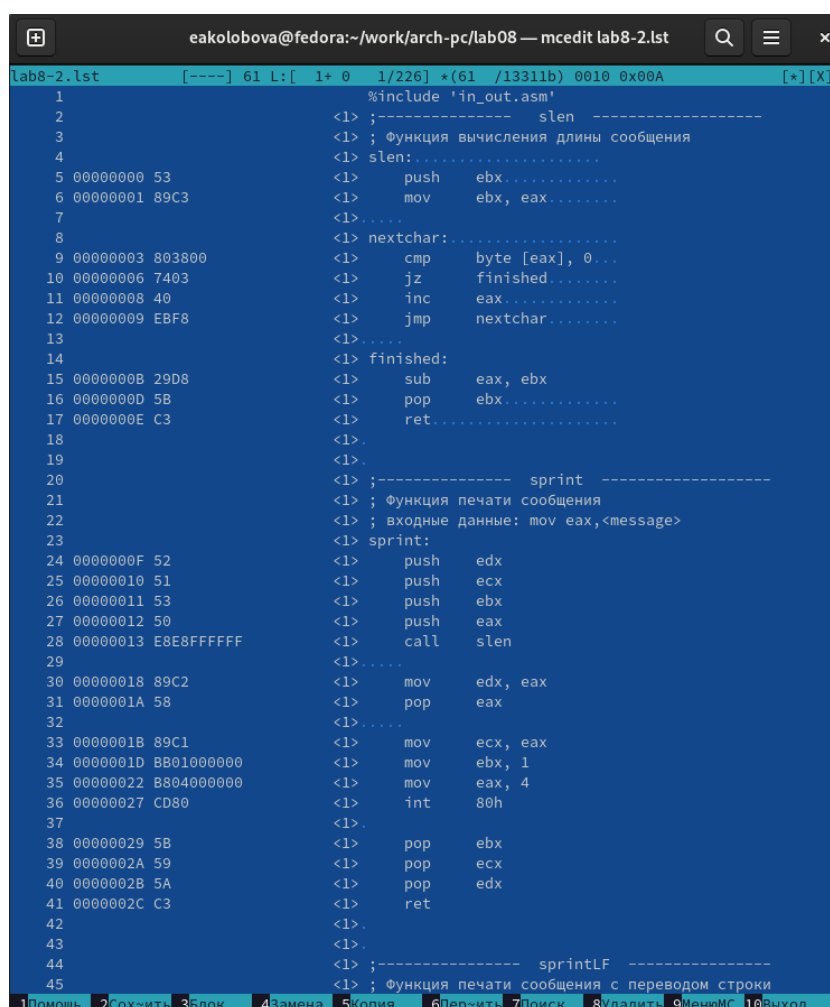


Рис. 2.12: Рис. 12. Окно файла листинга

```

eakolobova@fedora:~/work/arch-pc/lab08 — mcedit lab8-2.lst
lab8-2.lst  [----] 75 L: [ 45+ 0 45/226] *(2790/13311b) 1076 0x434 [*][X]
45      <I> ; Функция печати сообщения с переводом строки
46      <I> ; входные данные: mov eax,<message>
47      <I> sprintf:
48 0000002D E8DDFFFFFF <I> call sprint
49      <I> .
50 00000032 50 <I> push eax
51 00000033 B80A000000 <I> mov eax, 0AH
52 00000038 50 <I> push eax
53 00000039 89E0 <I> mov eax, esp
54 0000003B E8CFFFFFFF <I> call sprintf
55 00000040 58 <I> pop eax
56 00000041 58 <I> pop eax
57 00000042 C3 <I> ret
58      <I> .
59      <I> ;----- sread -----
60      <I> ; Функция считывания сообщения
61      <I> ; входные данные: mov eax,<buffer>, mov ebx,<N>
62      <I> sread:
63 00000043 53 <I> push ebx
64 00000044 50 <I> push eax
65      <I> .
66 00000045 B800000000 <I> mov ebx, 0
67 0000004A B803000000 <I> mov eax, 3
68 0000004F CD80 <I> int 80h
69      <I> .
70 00000051 5B <I> pop ebx
71 00000052 59 <I> pop ecx
72 00000053 C3 <I> ret
73      <I> .
74      <I> ;----- iprint -----
75      <I> ; Функция вывода на экран чисел в формате ASCII
76      <I> ; входные данные: mov eax,<int>
77      <I> iprint:
78 00000054 50 <I> push eax
79 00000055 51 <I> push ecx
80 00000056 52 <I> push edx
81 00000057 56 <I> push esi
82 00000058 B900000000 <I> mov ecx, 0
83      <I> .
84      <I> divideLoop:
85 0000005D 41 <I> inc ecx
86 0000005E BA00000000 <I> mov edx, 0
87 00000063 BE0A000000 <I> mov esi, 10
88 00000068 F7FE <I> idiv esi
89 0000006A 83C230 <I> add edx, 48

```

Рис. 2.13: Рис. 13. Окно файла листинга

```

eakolobova@fedora:~/work/arch-pc/lab08 — mcedit lab8-2.lst
lab8-2.lst  [----] 61 L: [ 89+ 0 89/226] *(5554/13311b) 0010 0x00A [*][X]
89 0000006A 83C230 <1> add edx, 48..
90 0000006D 52 <1> push edx...
91 0000006E 83F800 <1> cmp eax, 0...
92 00000071 75EA <1> jnz divideLoop..
93 <1> .
94 <1> printLoop:
95 00000073 49 <1> dec ecx.....
96 00000074 89E0 <1> mov eax, esp..
97 00000076 E894FFFFFF <1> call sprint...
98 0000007B 58 <1> pop eax.....
99 0000007C 83F900 <1> cmp ecx, 0...
100 0000007F 75F2 <1> jnz printLoop..
101 <1> .
102 00000081 5E <1> pop esi...
103 00000082 5A <1> pop edx...
104 00000083 59 <1> pop ecx...
105 00000084 58 <1> pop eax.....
106 00000085 C3 <1> ret
107 <1> .
108 <1> .
109 <1> ;----- iprintLF -----
110 <1> ; Функция вывода на экран чисел в формате ASCII
111 <1> ; входные данные: mov eax,<int>
112 <1> iprintLF:
113 00000086 E8C9FFFFFF <1> call iprint.....
114 <1> .
115 0000008B 50 <1> push eax.....
116 0000008C B80A000000 <1> mov eax, 0Ah.....
117 00000091 50 <1> push eax.....
118 00000092 89E0 <1> mov eax, esp.....
119 00000094 E876FFFFFF <1> call sprint.....
120 00000099 58 <1> pop eax.....
121 0000009A 58 <1> pop eax.....
122 0000009B C3 <1> ret
123 <1> .
124 <1> ;----- atoi -----
125 <1> ; Функция преобразования ascii-код символа в целое ч
126 <1> ; входные данные: mov eax,<int>
127 <1> atoi:
128 0000009C 53 <1> push ebx.....
129 0000009D 51 <1> push ecx.....
130 0000009E 52 <1> push edx.....
131 0000009F 56 <1> push esi.....
132 000000A0 89C6 <1> mov esi, eax.....
133 000000A2 B800000000 <1> mov eax, 0.....
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 2.14: Рис. 14. Окно файла листинга

```

eakolobova@fedora:~/work/arch-pc/lab08 — mcedit lab8-2.lst
lab8-2.lst  [----] 41 L: [135+ 0 135/226] *(8530/13311b) 0010 0x00A [*][X]
135 <I> .
136 <I> .multiplyLoop:
137 000000AC 31DB <I> xor ebx, ebx.....
138 000000AE 8A1C0E <I> mov bl, [esi+ecx]
139 000000B1 80FB30 <I> cmp bl, 48.
140 000000B4 7C14 <I> jl .finished.
141 000000B6 80FB39 <I> cmp bl, 57..
142 000000B9 7F0F <I> jg .finished.
143 <I> ..
144 000000BB 80EB30 <I> sub bl, 48.
145 000000BE 01D8 <I> add eax, ebx
146 000000C0 BB0A000000 <I> mov ebx, 10..
147 000000C5 F7E3 <I> mul ebx..
148 000000C7 41 <I> inc ecx...
149 000000C8 EBE2 <I> jmp .multiplyLoop..
150 <I> ..
151 <I> .finished:
152 000000CA 83F900 <I> cmp ecx, 0..
153 000000CD 7407 <I> je .restore...
154 000000CF BB0A000000 <I> mov ebx, 10..
155 000000D4 F7F3 <I> div ebx.....
156 <I> ..
157 <I> .restore:
158 000000D6 5E <I> pop esi...
159 000000D7 5A <I> pop edx....
160 000000D8 59 <I> pop ecx..
161 000000D9 5B <I> pop ebx..
162 000000DA C3 <I> ret
163 <I> ..
164 <I> ..
165 <I> ;----- quit -----
166 <I> ; Функция завершения программы
167 <I> quit:
168 000000DB 8B00000000 <I> mov ebx, 0.....
169 000000DE B801000000 <I> mov eax, 1.....
170 000000E5 CD80 <I> int 80h
171 000000E7 C3 <I> ret
2 SECTION .data
3 00000000 D092D0B2D0B5D0B4D0- msg1 DB 'Введите B' ,0h
3 00000009 B8D182D0B520D09200.
4 00000012 D09DD0B0D0B8D0B1D0- msg2 DB "Наибольшее число:" ,0h
4 0000001B 8ED0BBD18CD188D0B5-
4 00000024 D0B520D187D0B8D181-
4 0000002D D0BBD0BE3A00.....
5 00000033 32300000 A DD '20'
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 2.15: Рис. 15. Окно файла листинга

7. Подробно объясним содержимое трёх строк файла листинга 19, 20, 21, представленных на рис. 2.16 : 19, 20 и 21 - номера соотв. строк 000000F2, 000000F7, 000000FC - адреса B9(0A000000], BA0A000000, E842FFFFFF - машинные кода, ассемблированные исходные строки в виде шестнадцатеричных последовательностей `mov ecx,B`; `mov edx,10`; `call sread` - исходный текст программы

```

19 000000F2 B9[0A000000] mov ecx,B
20 000000F7 BA0A000000 mov edx,10
21 000000FC E842FFFFFF call sread

```

Рис. 2.16: Рис. 16. Окно файла листинга

8. Откроем файл с программой lab8-2.asm и в выделенной на рис. инструк-

ции с двумя операндами удалим один операнд. Выполните трансляцию с получением файла листинга, (рис. 2.17, 2.18):

```
nasm -f elf -l lab8-2.lst lab8-2.asm
```

В данном случае файлы не создаются, а в терминале выводится ошибка - рис. 2.19



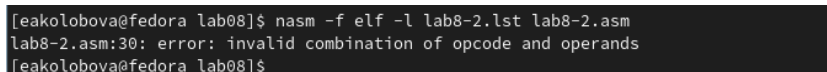
```
cmp ecx, [C]
jg check_B
mov ecx, [C]
mov [max], ecx
```

Рис. 2.17: Рис. 17. Удаление операнда



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

Рис. 2.18: Рис. 18. Удаление операнда



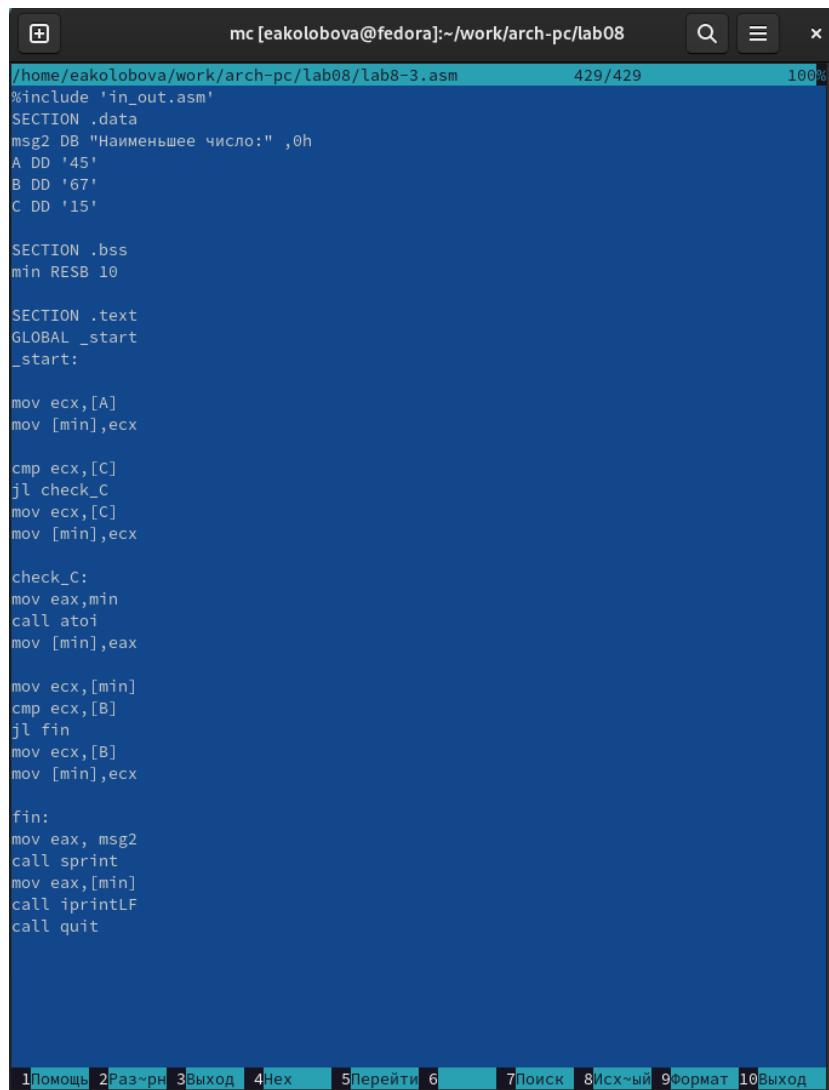
```
[eakolobova@fedora lab08]$ nasm -f elf -l lab8-2.lst lab8-2.asm
lab8-2.asm:30: error: invalid combination of opcode and operands
[eakolobova@fedora lab08]$
```

Рис. 2.19: Рис. 19. Создание листинга

2.1 Задание для самостоятельной работы

В ходе выполнения работы № 7 был получен вариант 7, соответствующие значения: 45, 67, 15 в табл. 8.5 выражение в табл. 8.6: $f(x) = 6a$, $x=a$; $x+a$, $x \neq a$; $(x_1, a_1) = (1, 1)$, $(x_2, a_2) = (2, 1)$

1. Написать программу нахождения наименьшей из 3 целочисленных переменных $a = 45$, $b = 67$ и $c = 15$. (рис. 2.20, 2.21)



```
mc [eakolobova@fedora]:~/work/arch-pc/lab08
/home/eakolobova/work/arch-pc/lab08/lab8-3.asm 429/429 100%
#include 'in_out.asm'
SECTION .data
msg2 DB "Наименьшее число:" ,0h
A DD '45'
B DD '67'
C DD '15'

SECTION .bss
min RESB 10

SECTION .text
GLOBAL _start
_start:

mov ecx,[A]
mov [min],ecx

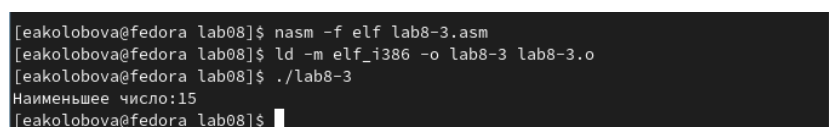
cmp ecx,[C]
jl check_C
mov ecx,[C]
mov [min],ecx

check_C:
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
```

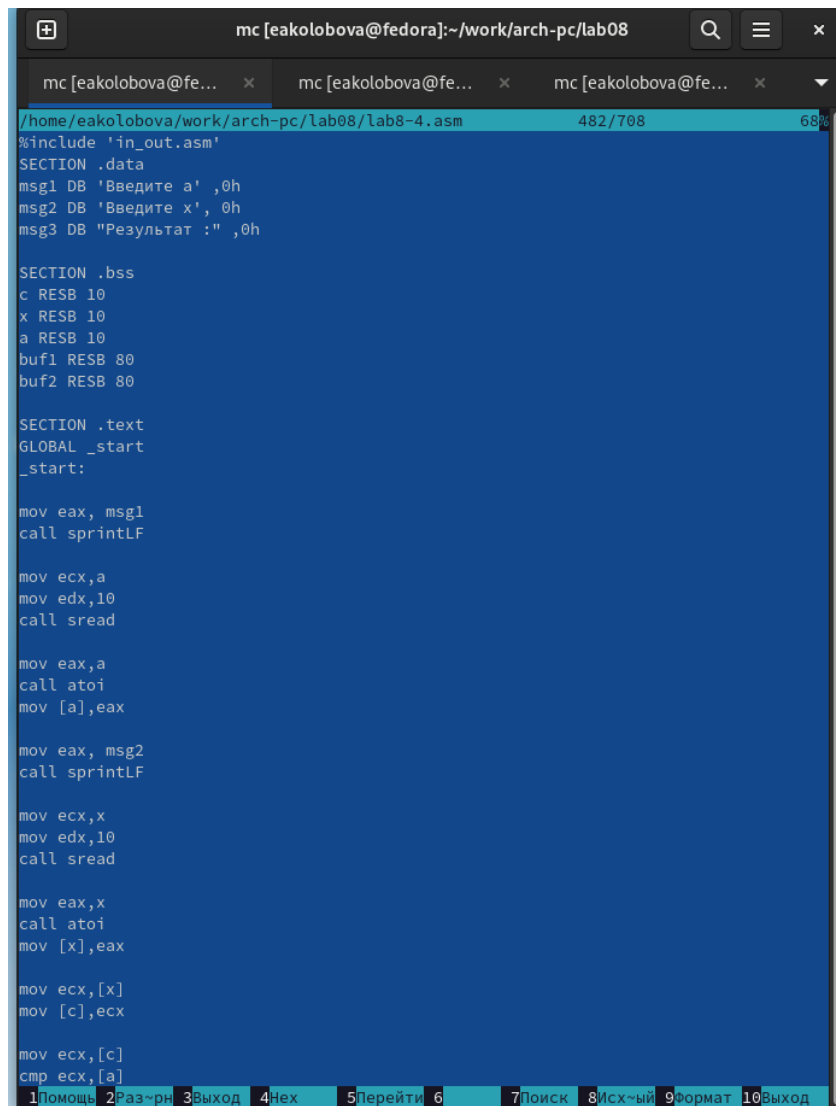
Рис. 2.20: Рис. 20. Текст программы



```
[eakolobova@fedora lab08]$ nasm -f elf lab8-3.asm
[eakolobova@fedora lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[eakolobova@fedora lab08]$ ./lab8-3
Наименьшее число:15
[eakolobova@fedora lab08]$
```

Рис. 2.21: Рис. 21. Компоновка и запуск файла

2. Написать программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x) = \begin{cases} a, & x=a; \\ x+a, & x \neq a; \end{cases}$ и выводит результат вычислений. (рис. 2.22, 2.23, 2.24)



The screenshot shows a text editor window titled "mc [eakolobova@fedora]:~/work/arch-pc/lab08". The file being edited is "/home/eakolobova/work/arch-pc/lab08/lab8-4.asm", which is 482/708 bytes in size. The code is written in assembly language and includes sections for data, bss, and text. The text section contains instructions for reading two integers, converting them to decimal, and comparing them. The bottom status bar shows a menu with options: 1Помощь, 2Развернуть, 3Выход, 4Нех, 5Перейти, 6, 7Поиск, 8Исходный, 9Формат, 10Выход.

```
%include 'in_out.asm'
SECTION .data
msg1 DB 'Введите a' ,0h
msg2 DB 'Введите x' ,0h
msg3 DB "Результат :" ,0h

SECTION .bss
c RESB 10
x RESB 10
a RESB 10
buf1 RESB 80
buf2 RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg1
call sprintf

mov ecx, a
mov edx, 10
call sread

mov eax, a
call atoi
mov [a], eax

mov eax, msg2
call sprintf

mov ecx, x
mov edx, 10
call sread

mov eax, x
call atoi
mov [x], eax

mov ecx, [x]
mov [c], ecx

mov ecx, [c]
cmp ecx, [a]
```

Рис. 2.22: Рис. 22. Текст программы

```
mc [eakolobova@fedora]:~/work/arch-pc/lab08
/home/eakolobova/work/arch-pc/lab08/lab8-4.asm 708/708 100%
call sread

mov eax,a
call atoi
mov [a],eax

mov eax, msg2
call sprintf

mov ecx,x
mov edx,10
call sread

mov eax,x
call atoi
mov [x],eax

mov ecx,[x]
mov [c],ecx

mov ecx,[c]
cmp ecx,[a]
jne uneq
mov eax,[a]
mov ebx,6
mul ebx
mov [buf1],eax
mov ecx,[buf1]
jmp fin

uneq:
mov eax,[a]
mov ebx,[x]
add eax,ebx
mov [buf2],eax
mov ecx,[buf2]
jmp fin

fin:
mov eax, msg3
call sprintf
mov eax,ecx
call iprintLF
call quit
```

Рис. 2.23: Рис. 23. Текст программы

```
[eakolobova@fedora lab08]$ nasm -f elf -g lab8-4.asm
[eakolobova@fedora lab08]$ ld -m elf_i386 -o lab8-4 lab8-4.o
[eakolobova@fedora lab08]$ ./lab8-4
Введите a
3
Введите x
3
Результат :18
[eakolobova@fedora lab08]$ mc
[eakolobova@fedora lab08]$ ./lab8-4
Введите a
1
Введите x
1
Результат :6
[eakolobova@fedora lab08]$ ./lab8-4
Введите a
1
Введите x
2
Результат :3
[eakolobova@fedora lab08]$
```

Рис. 2.24: Рис. 24. Компоновка и запуск файла

Ссылка на репозиторий: https://github.com/eakolobova/study_2022-2023_arch-pc/tree/master/labs/lab08/report

3 Выводы

Результатом проведенной работы является освоение команд условного и безусловного переходов и приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.