

Отчёт по лабораторной работе 7

Архитектура компьютеров и операционные системы

Тарасова Алина НКАбд 05-23

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 5 |
| 2 | Выполнение лабораторной работы | 6 |
| 2.1 | Реализация переходов в NASM | 6 |
| 2.2 | Изучение структуры файлы листинга | 13 |
| 2.3 | Самостоятельное задание | 17 |
| 3 | Выводы | 21 |

Список иллюстраций

| | | |
|------|---|----|
| 2.1 | Создан каталог | 6 |
| 2.2 | Программа lab7-1.asm | 7 |
| 2.3 | Запуск программы lab7-1.asm | 8 |
| 2.4 | Программа lab7-1.asm | 8 |
| 2.5 | Запуск программы lab7-1.asm | 9 |
| 2.6 | Программа lab7-1.asm | 10 |
| 2.7 | Запуск программы lab7-1.asm | 10 |
| 2.8 | Программа lab7-2.asm | 12 |
| 2.9 | Запуск программы lab7-2.asm | 13 |
| 2.10 | Файл листинга lab7-2 | 14 |
| 2.11 | Ошибка трансляции lab7-2 | 15 |
| 2.12 | Файл листинга с ошибкой lab7-2 | 16 |
| 2.13 | Программа lab7-task1.asm | 17 |
| 2.14 | Запуск программы lab7-task1.asm | 18 |
| 2.15 | Программа lab7-task2.asm | 19 |
| 2.16 | Запуск программы lab7-task2.asm | 20 |

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

2.1 Реализация переходов в NASM

Создаю каталог для программ лабораторной работы № 7 и файл lab7-1.asm.
(рис. [2.1])

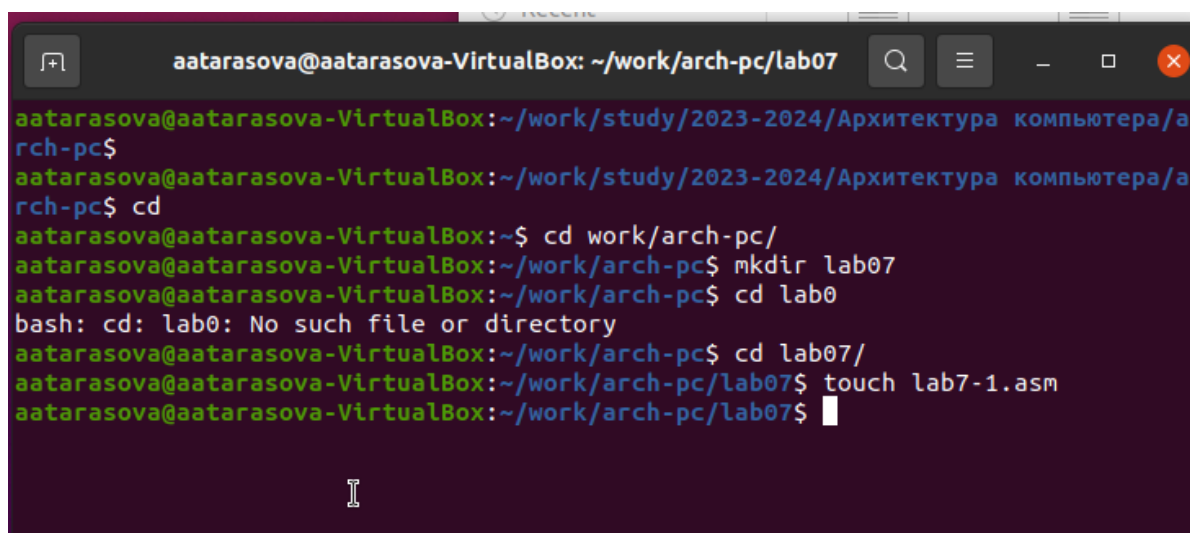
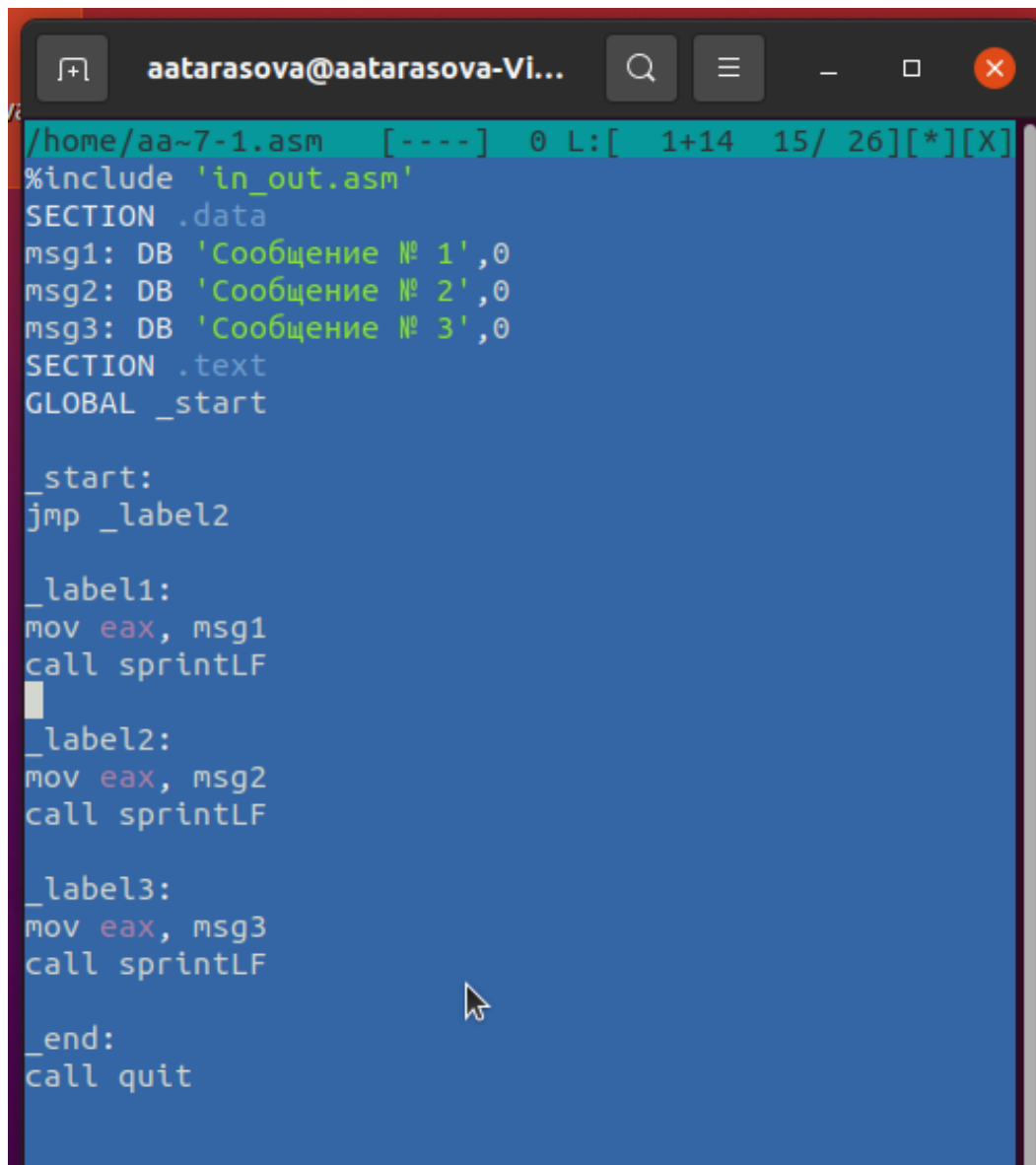
A screenshot of a terminal window titled 'aatarasova@aatarasova-VirtualBox: ~/work/arch-pc/lab07'. The terminal shows a series of commands and their outputs. The user starts at the prompt 'aatarasova@aatarasova-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc\$'. They enter 'cd', which changes the directory to '~/work/arch-pc/'. Then they enter 'mkdir lab07', which creates a new directory. Next, they enter 'cd lab0', which results in an error message 'bash: cd: lab0: No such file or directory'. They then enter 'cd lab07/', which successfully changes the directory to '~/work/arch-pc/lab07/'. Finally, they enter 'touch lab7-1.asm', which creates a new file. The terminal ends with the prompt 'aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07\$' and a cursor.

Рис. 2.1: Создан каталог

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. (рис. [2.2])



```
/home/aa~7-1.asm [ - - - ] 0 L: [ 1+14 15/ 26 ] [*][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
```

Рис. 2.2: Программа lab7-1.asm

Создаю исполняемый файл и запускаю его. (рис. [2.3])

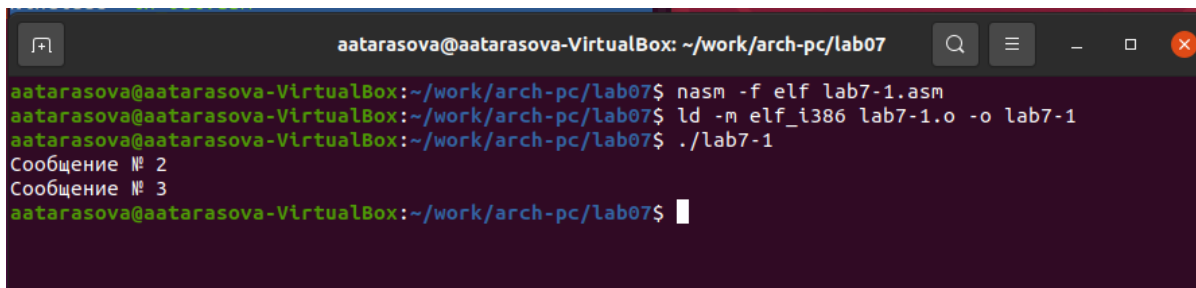
A terminal window titled 'aatarasova@aatarasova-VirtualBox: ~/work/arch-pc/lab07'. The prompt is 'aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07\$'. The user enters 'nasm -f elf lab7-1.asm', followed by 'ld -m elf_i386 lab7-1.o -o lab7-1', and finally './lab7-1'. The output shows 'Сообщение № 2' and 'Сообщение № 3' on separate lines. The prompt returns to 'aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07\$'.

Рис. 2.3: 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`).

Изменяю текст программы в соответствии с листингом 7.2. (рис. [2.4]) (рис. [2.5])

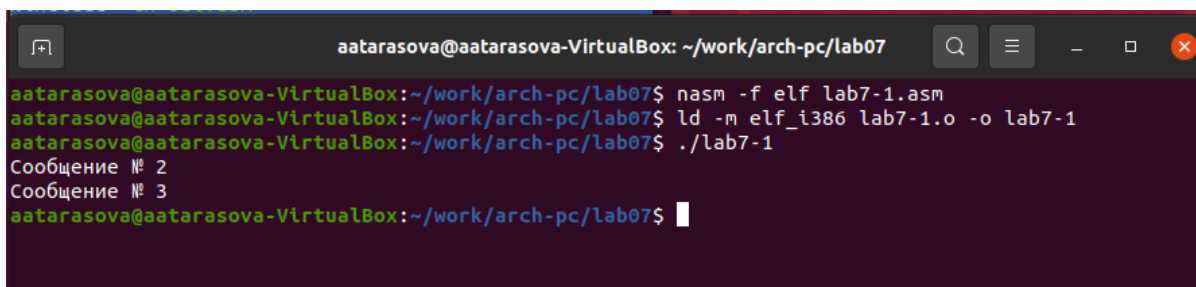
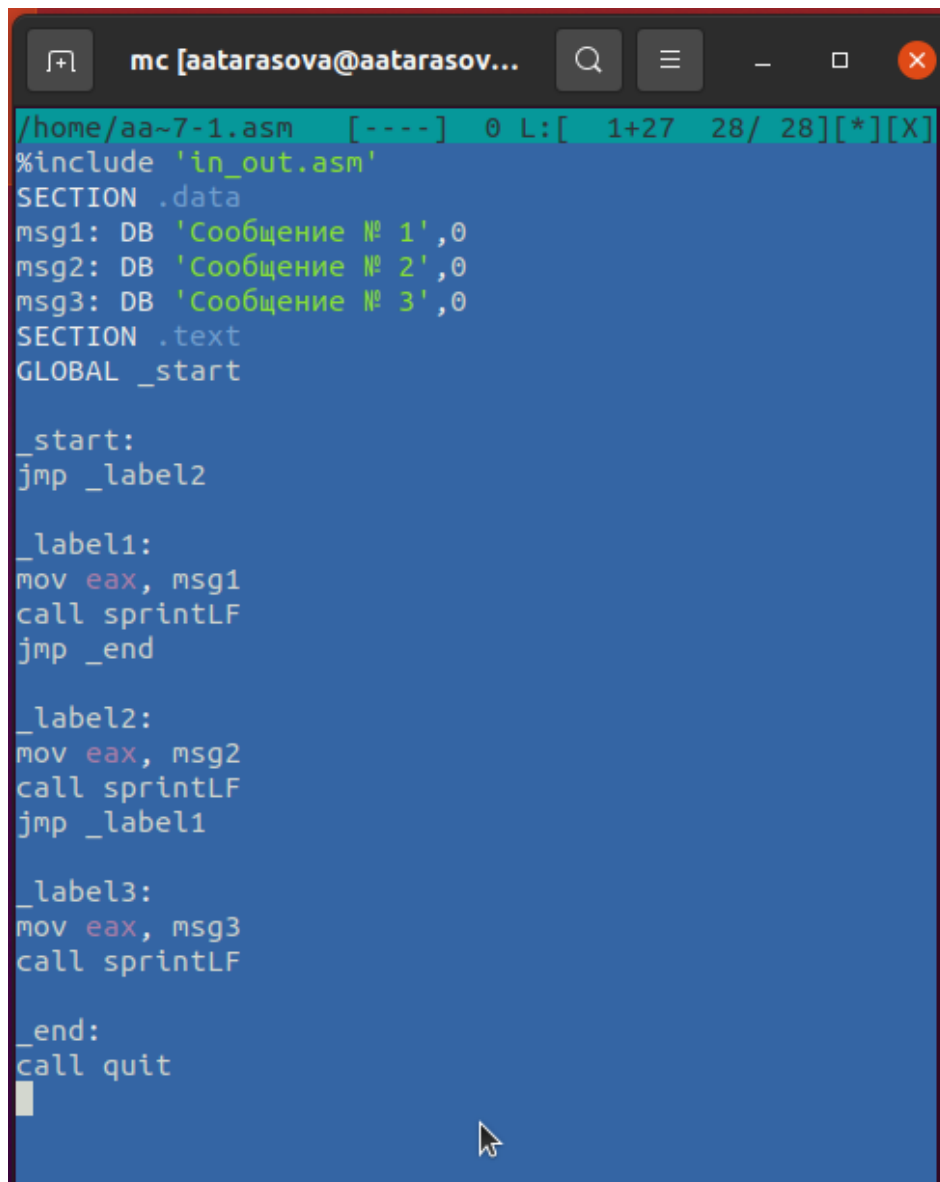
A terminal window titled 'aatarasova@aatarasova-VirtualBox: ~/work/arch-pc/lab07'. The prompt is 'aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07\$'. The user enters 'nasm -f elf lab7-1.asm', followed by 'ld -m elf_i386 lab7-1.o -o lab7-1', and finally './lab7-1'. The output shows 'Сообщение № 2' and 'Сообщение № 3' on separate lines. The prompt returns to 'aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07\$'.

Рис. 2.4: Программа lab7-1.asm



```
mc [aatarasova@aatarasov...  
/home/aa~7-1.asm [----] 0 L:[ 1+27 28/ 28][*][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
```

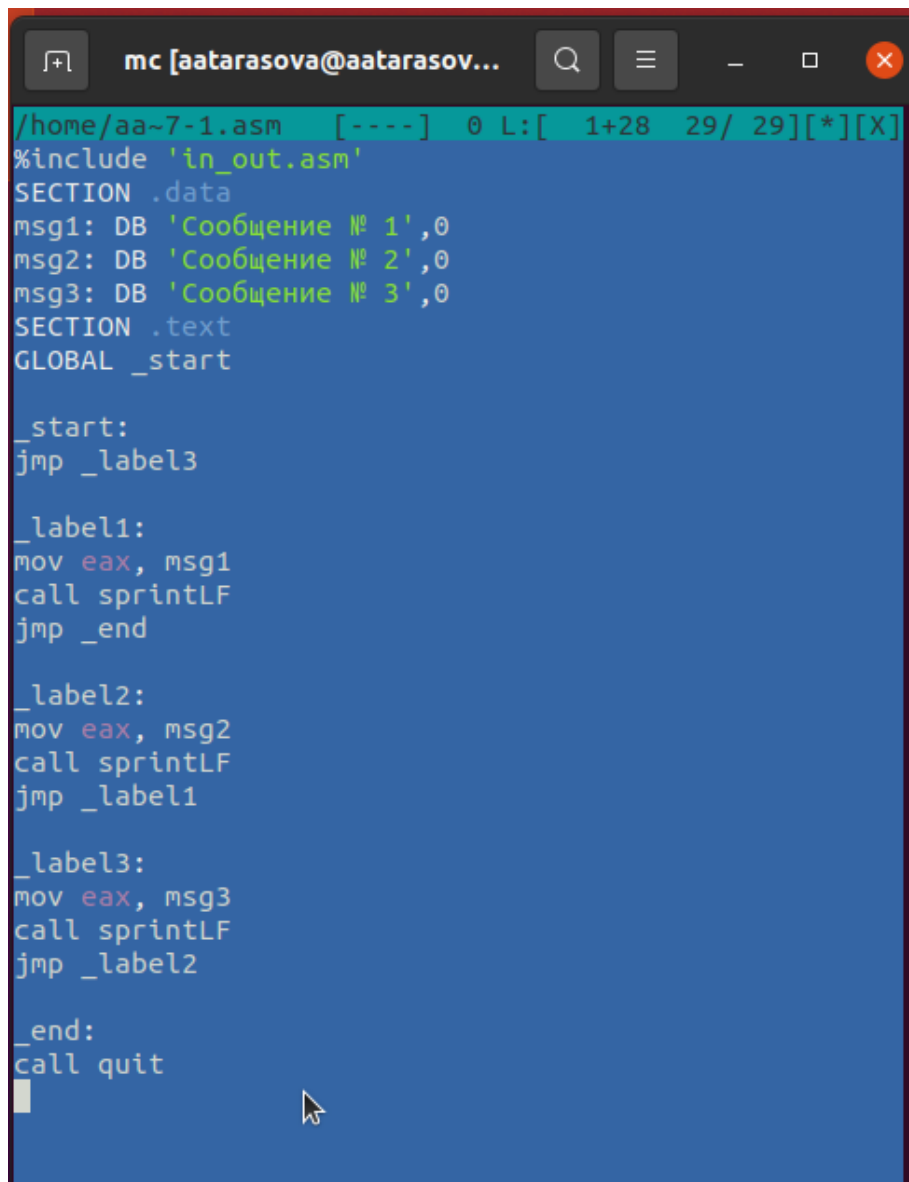
Рис. 2.5: Запуск программы lab7-1.asm

Изменила текст программы, чтобы вывод программы был следующим (рис. [2.6]) (рис. [2.7]):

Сообщение № 3

Сообщение № 2

Сообщение № 1



```
mc [aatarasova@aatarasov...
/home/aa~7-1.asm  [ - - - ]  0 L: [ 1+28 29/ 29][*][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

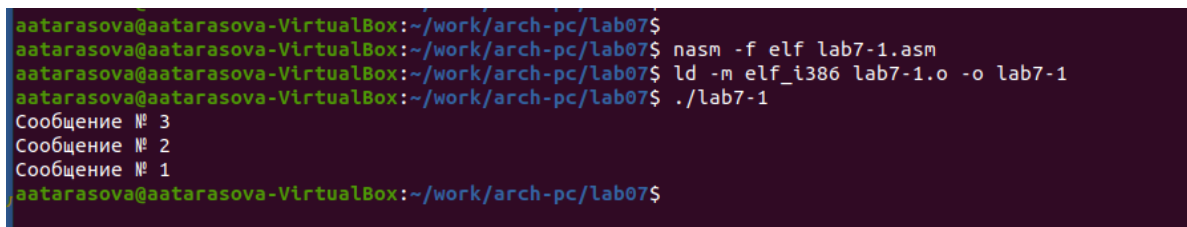
_label1:
mov eax, msg1
call sprintLF
jmp _end

_label2:
mov eax, msg2
call sprintLF
jmp _label1

_label3:
mov eax, msg3
call sprintLF
jmp _label2

_end:
call quit
```

Рис. 2.6: Программа lab7-1.asm



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

Рис. 2.7: Запуск программы lab7-1.asm

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: А, В и С. Значения для А и С задаются в программе, значение В вводится с клавиатуры.

Создала исполняемый файл и проверила его работу для разных значений В (рис. [2.8]) (рис. [2.9]).

```
mc [aatarasova@aatarasova-VirtualBox]:~/work/arch-pc/lab07
/home/aatar~/lab7-2.asm [----] 0 L:[ 8+23 31/ 50] *(666 /1057b) 0109 0x06D [*][X]
max resb 10
B resb 10
section .text
global _start
_start:
; ----- 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,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. 2.8: Программа lab7-2.asm

```
aatarasova@aatarasova-VirtualBox: ~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
aatarasova@aatarasova-VirtualBox: ~/work/arch-pc/lab07$ ld -m elf_i386 lab7-2.o -o lab7-2
aatarasova@aatarasova-VirtualBox: ~/work/arch-pc/lab07$ ./lab7-2
Введите В: 30
Наибольшее число: 50
aatarasova@aatarasova-VirtualBox: ~/work/arch-pc/lab07$ ./lab7-2
Введите В: 50
Наибольшее число: 50
aatarasova@aatarasova-VirtualBox: ~/work/arch-pc/lab07$ ./lab7-2
Введите В: 70
Наибольшее число: 70
aatarasova@aatarasova-VirtualBox: ~/work/arch-pc/lab07$
```

Рис. 2.9: Запуск программы lab7-2.asm

2.2 Изучение структуры файлы листинга

Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ `-l` и задав имя файла листинга в командной строке.

Создаю файл листинга для программы из файла lab7-2.asm (рис. [2.10])

```
mc [aatarasova@aatarasova-VirtualBox]:~/work/arch-pc/lab07
/home/aata~lab7-2.lst  [- - -]  0 L:[183+ 2 185/225] *(11284/13771b) 0032 0x020 [*][X]
 8 00000000 <res 0000000A>      max resb 10
 9 0000000A <res 0000000A>      B resb 10
10                                section .text
11                                global _start
12                                _start:
13                                ; ----- Вывод сообщения 'Введите В: '
14 000000E8 B8[00000000]      mov eax,msg1
15 000000ED E81DFFFFFF      call sprint
16                                ; ----- Ввод 'В'
17 000000F2 B9[0A000000]      mov ecx,B
18 000000F7 BA0A000000      mov edx,10
19 000000FC E842FFFFFF      call sread
20                                ; ----- Преобразование 'В' из символа в ч
21 00000101 B8[0A000000]      mov eax,B
22 00000106 E891FFFFFF      call atoi
23 0000010B A3[0A000000]      mov [B],eax
24                                ; ----- Записываем 'А' в переменную 'max'
25 00000110 8B0D[35000000]      mov ecx,[A].
26 00000116 890D[00000000]      mov [max],ecx
27                                ; ----- Сравниваем 'А' и 'С' (как символы
28 0000011C 3B0D[39000000]      cmp ecx,[C]
29 00000122 7F0C      jg check_B
30 00000124 8B0D[39000000]      mov ecx,[C].
31 0000012A 890D[00000000]      mov [max],ecx
32                                ; ----- Преобразование 'max(A,C)' из симв
33                                check_B:
34 00000130 B8[00000000]      mov eax,max
35 00000135 E862FFFFFF      call atoi
36 0000013A A3[00000000]      mov [max],eax
37                                ; ----- Сравниваем 'max(A,C)' и 'В' (как
38 0000013F 8B0D[00000000]      mov ecx,[max]
39 00000145 3B0D[0A000000]      cmp ecx,[B]
40 0000014B 7F0C      jg fin
41 0000014D 8B0D[0A000000]      mov ecx,[B]
42 00000153 890D[00000000]      mov [max],ecx
43                                ; ----- Вывод результата
44                                fin:
45 00000159 B8[13000000]      mov eax, msg2
46 0000015E E8ACFEFFFF      call sprint
47 00000163 A1[00000000]      mov eax,[max]
48 00000168 E819FFFFFF      call iprintLF
49 0000016D E869FFFFFF      call quit
```

Рис. 2.10: Файл листинга lab7-2

Ознакомимся с его форматом и содержимым.

строка 211

- 34 - номер строки
- 0000012E - адрес
- B8[00000000] - машинный код

- mov eax, max - код программы

строка 212

- 35 - номер строки
- 00000133 - адрес
- E864FFFFFF - машинный код
- call atoi - код программы

строка 213

- 36 - номер строки
- 00000138 - адрес
- A3[00000000] - машинный код
- mov [max], eax - код программы

Открыла файл с программой lab7-2.asm и в инструкции с двумя операндами удалила один операнд. Выполню трансляцию с получением файла листинга. (рис. [2.11]) (рис. [2.12])

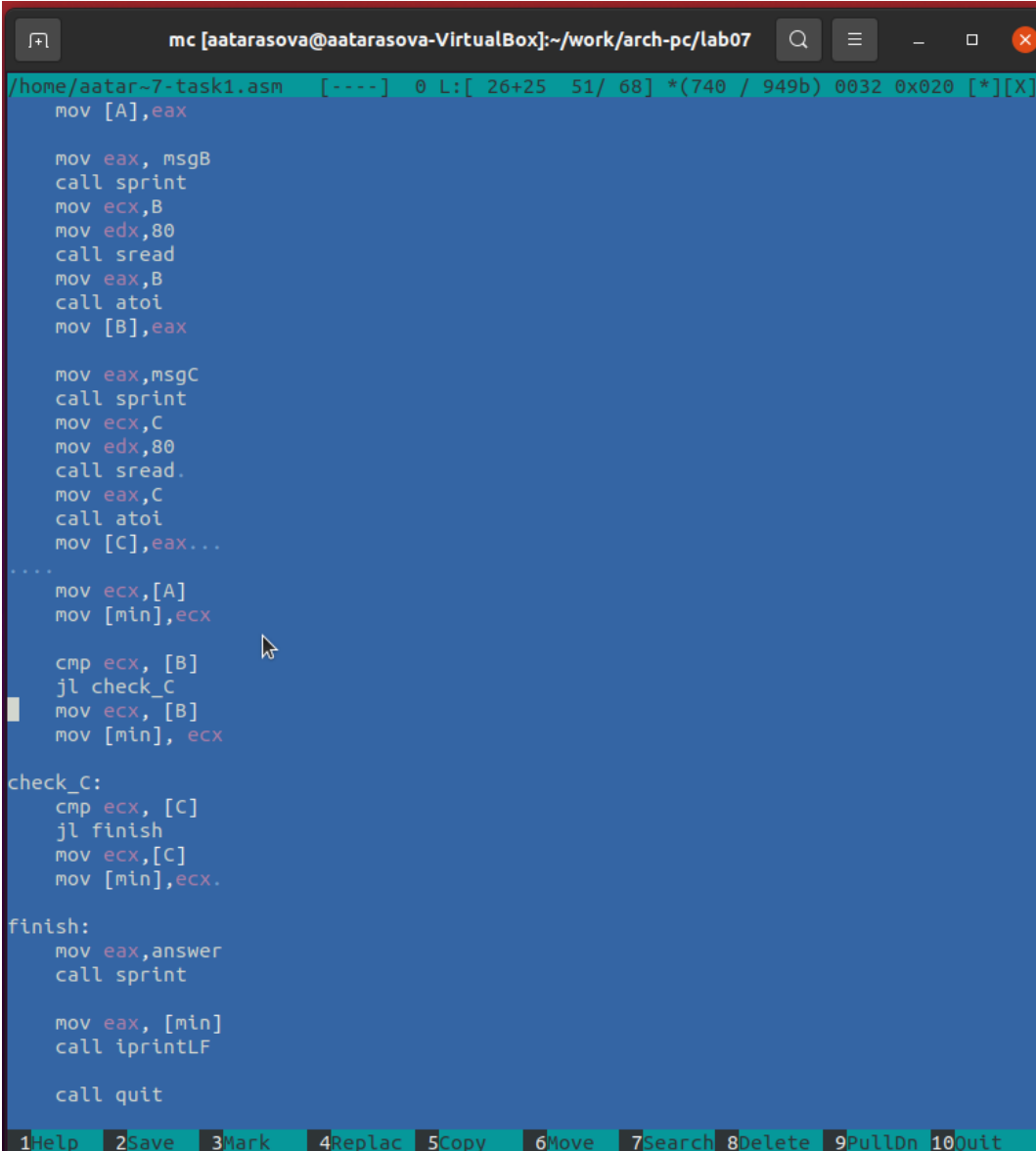
```
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm -l lab7-2.lst
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm -l lab7-2.lst
lab7-2.asm:29: error: symbol `check' undefined
lab7-2.asm:33: error: label `check_B' changed during code generation [-w+error=label-redef-late]
lab7-2.asm:44: error: label `fin' changed during code generation [-w+error=label-redef-late]
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 2.11: Ошибка трансляции lab7-2

2.3 Самостоятельное задание

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 (рис. [2.13]) (рис. [2.14])

для варианта 14 - 81, 22, 72



```
/home/aatar~7-task1.asm  [----]  0 L: [ 26+25  51/ 68] *(740 / 949b) 0032 0x020 [*][X]
mov [A],eax

mov eax, msgB
call sprint
mov ecx,B
mov edx,80
call sread
mov eax,B
call atoi
mov [B],eax

mov eax,msgC
call sprint
mov ecx,C
mov edx,80
call sread.
mov eax,C
call atoi
mov [C],eax...

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

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

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

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Рис. 2.13: Программа lab7-task1.asm

```

aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-task1.asm
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-task1.o -o lab7-task1
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$ ./lab7-task1
Input A: 81
Input B: 22
Input C: 72
Smallest: 22
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$

```

Рис. 2.14: Запуск программы lab7-task1.asm

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. (рис. [2.15]) (рис. [2.16])

для варианта 14

$$\begin{cases} 3a + 1, x < a \\ 3x + 1, x \geq a \end{cases}$$

При $x = 2, a = 3$ получается 10.

При $x = 4, a = 2$ получается 13.

```
mc [aatarasova@aatarasova-VirtualBox]:~/work/arch-pc/lab07
/home/aatar~7-task2.asm [----] 0 L: [ 11+ 4 15/ 53] *(207 / 733
SECTION .text
GLOBAL _start

start:
    mov eax,msgA
    call sprint
    mov ecx,A
    mov edx,80
    call sread
    mov eax,A
    call atoi.
    mov [A],eax

    mov eax,msgX
    call sprint
    mov ecx,X
    mov edx,80
    call sread.
    mov eax,X
    call atoi
    mov [X],eax...

    mov edx, [A]
    mov ebx, [X]
    cmp ebx, edx
    jl first
    jmp second

first:
    mov eax,[A]
    mov ebx, 3
    mul ebx
    add eax, 1
    call iprintLF.
    call quit

second:
    mov eax,[X]
    mov ebx, 3
    mul ebx
    add eax, 1
    call iprintLF.
    call quit
```

Рис. 2.15: Программа lab7-task2.asm

```
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$  
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-task2.asm  
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-task2.o -o lab7-task2  
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$ ./lab7-task2  
Input A: 3  
Input X: 2  
10  
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$ ./lab7-task2  
Input A: 2  
Input X: 4  
13  
aatarasova@aatarasova-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 2.16: Запуск программы lab7-task2.asm

3 Выводы

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