

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8**

дисциплина:   *Архитектура компьютера*

Студент:

Гольденгорин Виталий Борисович

Группа:

НММ-01-2022

**МОСКВА**

2022 г.

# Содержание

<b>1</b>	<b>Цель работы...</b>	<b>4</b>
<b>2</b>	<b>Задание...</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение...</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы...</b>	<b>7</b>
<b>5</b>	<b>Выводы...</b>	<b>13</b>

# Список иллюстраций

Рис. 4.1...	7
Рис. 4.2...	7
Рис. 4.3...	7
Рис. 4.4...	8
Рис. 4.5...	8
Рис. 4.6...	8
Рис. 4.7...	9
Рис. 4.8...	9
Рис. 4.9...	10
Рис. 4.10...	11
Рис. 4.11...	12
Рис. 4.12...	12
Рис. 4.13...	12
Рис. 4.14...	12

# 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

1. Создайте файл lab8-1.asm.
2. Введите в файл lab8-1.asm текст программы из листинга 8.1.  
Создайте исполняемый файл и запустите его. Измените текст программы в соответствии с листингом 8.2. Создайте исполняемый файл и проверьте его работу. Измените текст программы добавив или изменив инструкции jmp.
3. Создайте файл lab8-2.asm. Внимательно изучите текст программы из листинга 8.3 и введите в lab8-2.asm. Создайте исполняемый файл и проверьте его работу.
4. Создайте файл листинга для программы из файла lab8-2.asm.  
Откройте файл листинга lab8-2.lst с помощью любого текстового редактора. Откройте файл с программой lab8-2.asm и в любой инструкции с двумя операндами удалите один операнд. Выполните трансляцию с получением файла листинга.

### **3 Теоретическое введение**

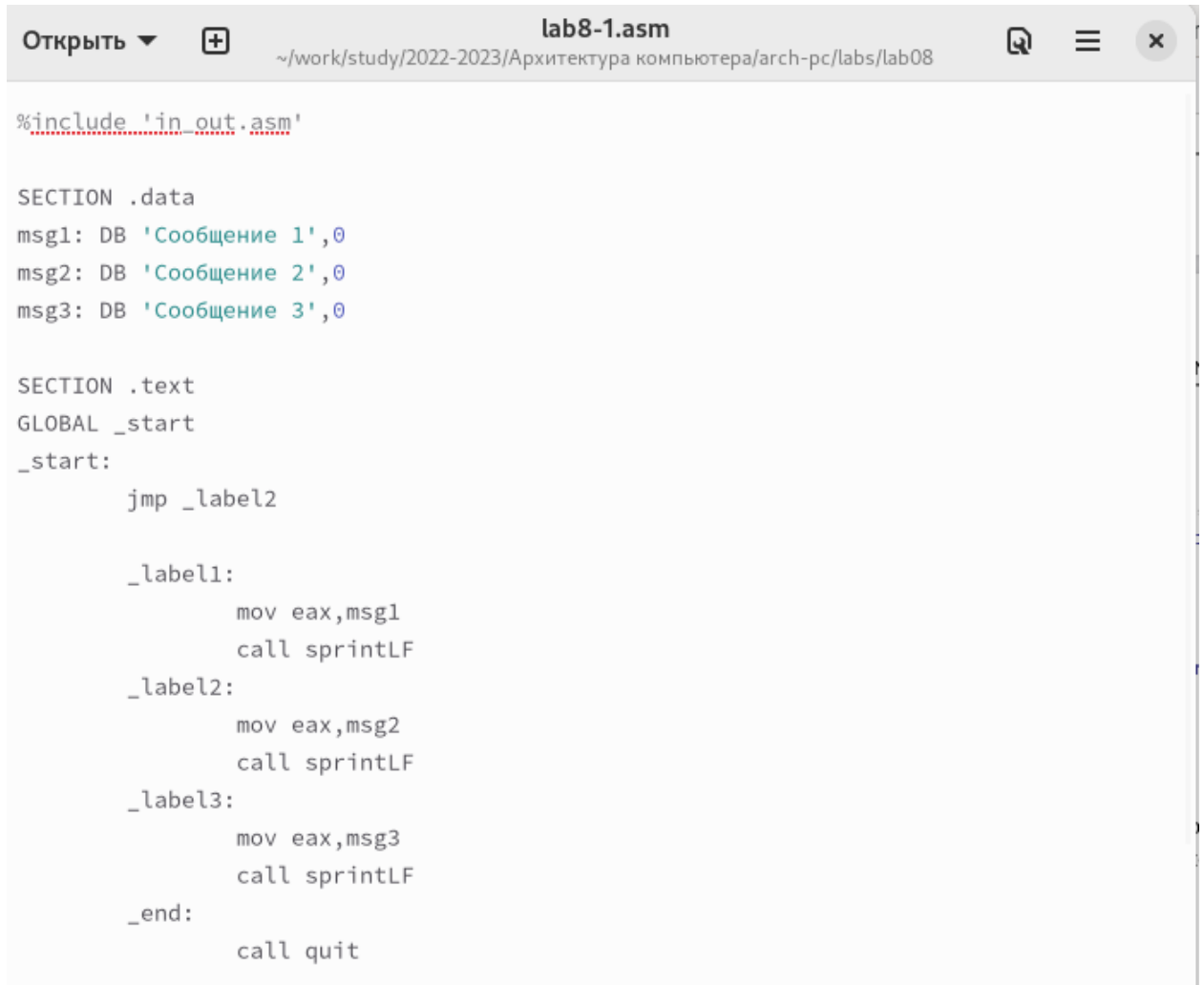
- условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия.
- безусловный переход – выполнение передачи управления в определенную точку программы без каких-либо условий.

## 4 Выполнение лабораторной работы

Сначала создаю файл lab8-1.asm в lab08. Потом ввожу текст программы из листинга 8.1. Создаю исполняемый файл и запускаю его.

```
[vitaliybg@fedora lab08]$ touch lab8-1.asm
```

Рис. 4.1: Создание файла lab8-1.asm



```
Открыть ▾ + lab8-1.asm
~/.work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab08

%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. 4.2: Текст программы lab8-1.asm

```
[vitaliybg@fedora lab08]$ nasm -f elf lab8-1.asm
[vitaliybg@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[vitaliybg@fedora lab08]$ ./lab8-1
Сообщение 2
Сообщение 3
```

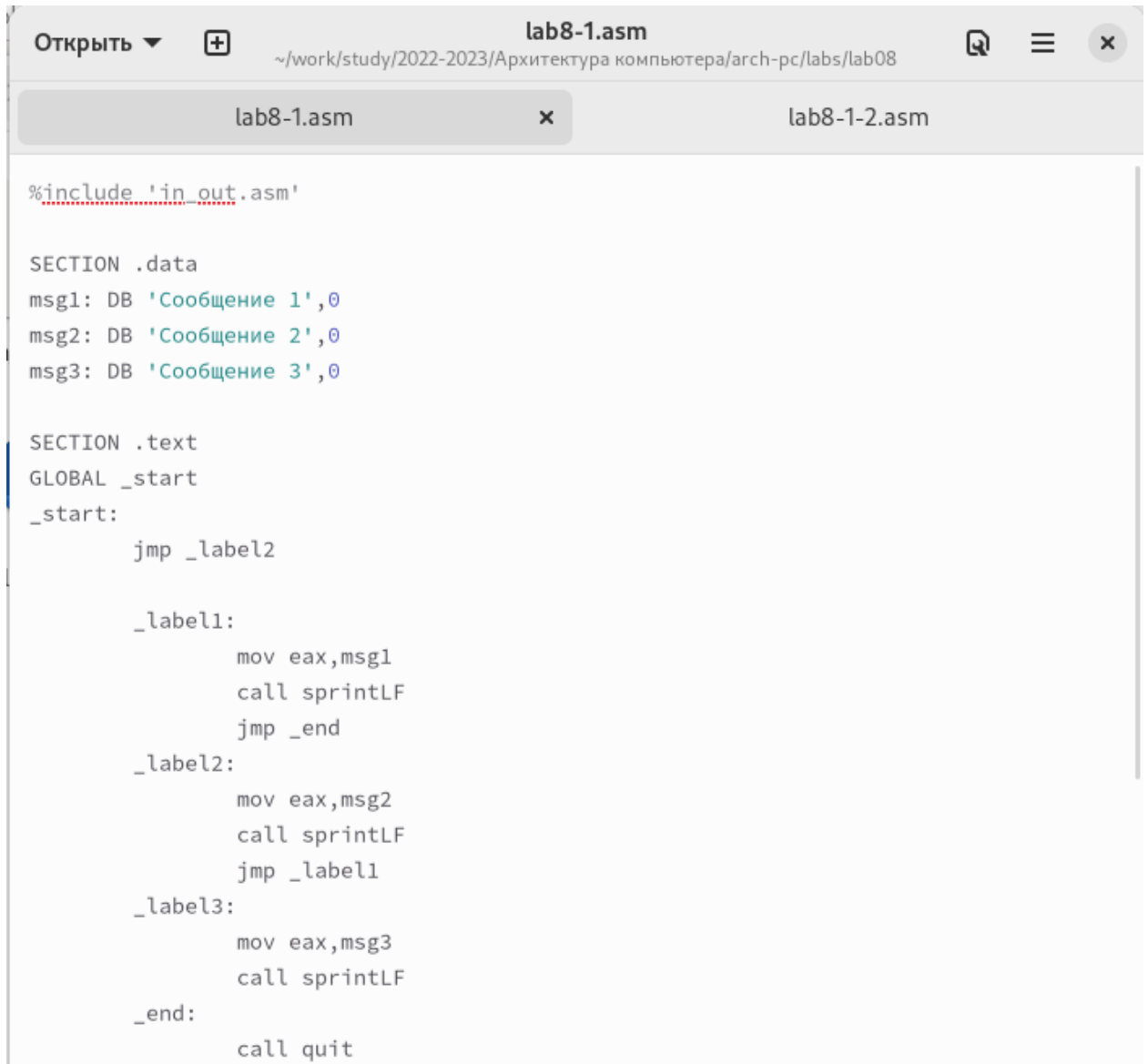
Рис. 4.3: Запуск исполняемого файла

После изменения `jmp _label2` на `jmp label_3` получаю новый результат.

```
[vitaliybg@fedora lab08]$ ./lab8-1-2
Сообщение 3
```

Рис. 4.4: Результат изменений

Изменяю текст программы `lab8-1.asm` под листинг 8.2 и запускаю его.



```
lab8-1.asm
~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab08

lab8-1.asm x lab8-1-2.asm

%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
```

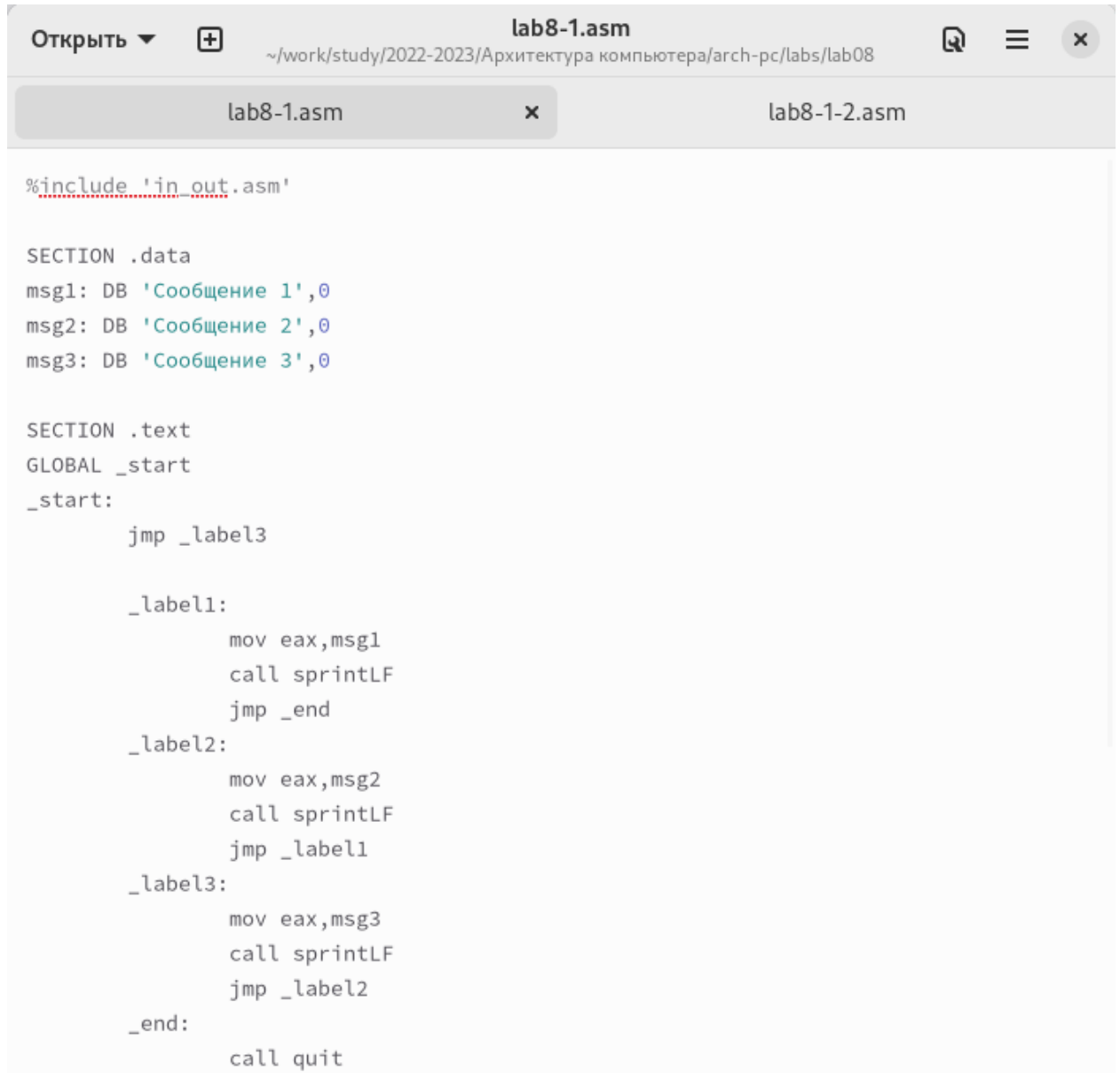
Рис. 4.5: Текст программы `lab8-1.asm`

```
[vitaliybg@fedora lab08]$ ./lab8-1
Сообщение 2
Сообщение 1
```

Рис. 4.6: Результат запуска



Опять изменяю текст программы, чтобы выводило сообщение 3, 2, 1.



```
%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

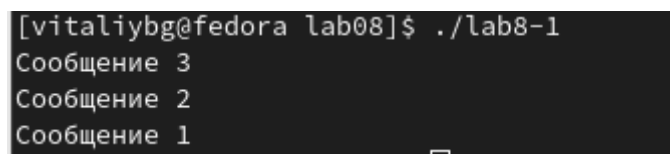
_label1:
    mov eax,msg1
    call sprintLF
    jmp _end

_label2:
    mov eax,msg2
    call sprintLF
    jmp _label1

_label3:
    mov eax,msg3
    call sprintLF
    jmp _label2

_end:
    call quit
```

Рис. 4.7: Текст программы после изменений



```
[vitaliybg@fedora lab08]$ ./lab8-1
Сообщение 3
Сообщение 2
Сообщение 1
```

Рис. 4.8: Результат изменений

Создаю файл lab8-2.asm в lab08. Ввожу текст программы листинга 8.3 и запускаю исполняемый файл, который я создал.

```
[vitaliybg@fedora lab08]$ touch lab8-2.asm
```

Рис. 4.9: Создание файла lab8-2.asm

```

%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],ecx

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

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

```

Рис. 4.10: Текст программы lab8-2.asm

```
[vitaliybg@fedora lab08]$ nasm -f elf lab8-2.asm
[vitaliybg@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[vitaliybg@fedora lab08]$ ./lab8-2
Введите B: 10
Наибольшее число: 50
[vitaliybg@fedora lab08]$
```

Рис. 4.11: Результат программы при вводе B=10

Создаю файл листинга для программы из файла lab8-2.asm. Потом открываю файл листинга lab8-2.lst с помощью mcedit.

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

Рис. 4.12: Создание листинга lab8-2.asm

```
lab8-2.lst  [----]  0  L:[ 91+10 101/223]  *(6207/13241b)  0032  0x020  [*] [X]
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 ASC
111 <1> ; входные данные: mov eax,<int>
112 <1> iprintLF:
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 4.13: Листинг lab8-2.lst

Если в любой инструкции с двумя операндами удалить один операнд, то выводит следующее:

```
[vitaliybg@fedora lab08]$ nasm -f elf -l lab8-2.lst lab8-2.asm
lab8-2.asm:13: error: invalid combination of opcode and operands
```

Рис. 4.14: Результат изменений

## **5 Выводы**

После выполнения работы, я пришел к следующему выводу: работа учит командам условного и безусловного переходов и как их использовать. Также учит структуру листинга.