

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 7**

*дисциплина: Архитектура компьютера*

Студент: Юсуф Абдулнасир Абубакар

Группа: НКАбд - 06 - 25

**МОСКВА**

2025 г.

# СОДЕРЖАНИЕ

1.

- Цель работы
- Задание
- Теоретическое введение
- Выполнение лабораторной работы
- Задание для самостоятельной работы
- Выводы

## 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

### 2.1 Основные задания

1. Создать программу с использованием инструкции безусловного перехода jmp
2. Создать программу с использованием инструкций условного перехода cmp и jg
3. Изучить структуру файла листинга
4. Создать файл листинга и проанализировать его содержимое

### 2.2 Задания для самостоятельной работы

1. Написать программу нахождения наименьшей из 3 целочисленных переменных
2. Написать программу вычисления значения функции  $f(x)$  для введенных  $x$  и  $a$

## 3 Теоретическое введение

### 3.1 Команды переходов

- **Безусловный переход** - выполнение передачи управления в определенную точку программы без каких-либо условий (инструкция jmp)

- **Условный переход** - выполнение или не выполнение перехода в зависимости от проверки условия (инструкции je, jne, jg, jl и др.)

## 3.2 Регистр флагов

Флаги отражают результат выполнения арифметических инструкций:

- CF - флаг переноса
- ZF - флаг нуля
- SF - флаг знака
- OF - флаг переполнения

## 3.3 Инструкция cmp

Инструкция cmp сравнивает два операнда и устанавливает флаги в зависимости от результата сравнения.

## 4 Выполнение лабораторной работы

**Код программы:**

```
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07    lab7-1.asm *
```

```
GNU nano 7.2
%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

^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

## Компиляция и выполнение: и Результат выполнения:

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

**Вывод:** Программа демонстрирует работу безусловных переходов jmp, изменяя порядок выполнения инструкций и пропуская вывод первого сообщения.

## 4.2 Программа с условными переходами (lab7-2.asm)

### Основной код программы:

The screenshot shows the assembly code for the program. The code includes sections for data and bss, and defines symbols A, C, msg1, and msg2. It includes a start label with code to print a message and read input, and a section to convert the input character to a number using atoi. The menu bar at the bottom includes options like Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line.

```
GNU nano 7.2                               lab7-2.asm *
%include 'in_out.asm'
section .data
msg1 db 'Введите В: ',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:
; --- Вывод сообщения 'Введите В: '
    mov eax, msg1
    call sprint
; --- Ввод 'В'
    mov ecx, B
    mov edx, 10
    call sread
; --- Преобразование 'В' из символа в число
    mov eax, B
    call atoi
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

### Тестирование программы

Тест 1 - В=30:

The terminal window shows the command nano lab7-2.asm, followed by nasm -f elf lab7-2.asm, ld -m elf\_i386 lab7-2.o -o lab7-2, and finally ./lab7-2. The user inputs '30' when prompted for 'Введите В:' and the program outputs 'Наибольшее число: 50'.

```
сообщение № 1
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nano lab7-2.asm
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-2.o
-o lab7-2
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 30
Наибольшее число: 50
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
```

## Тест 2 - В=60:

```
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 60
Наибольшее число: 60
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ █
```

**Вывод:** Программа корректно находит наибольшее число среди А, В, С используя условные переходы cmp и jg.

## 4.3 Файл листинга

### Создание файла листинга:

```
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ cat lab7-2.lst
```

### Структура файла листинга:

```
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ cat lab7-2.lst
1                                %include 'in_out.asm'
1                                <1> ; in_out.asm
2                                <1> ; Assembly library for input/output oper
ations
3                                <1>
4                                <1> section .text
5                                <1>
6                                <1> ; -----
-----
7                                <1> ; String length calculation
8                                <1> ; eax = pointer to string
9                                <1> ; returns: eax = length
10                               <1> ; -----
11                               <1> slen:
12 00000000 53                  <1>     push    ebx
13 00000001 89C3                <1>     mov     ebx, eax
14
15                               <1> .next:
16 00000003 803800              <1>     cmp     byte [eax], 0
17 00000006 7403                <1>     jz      .finished
18 00000008 40                  <1>     inc     eax
19 00000009 EBF8                <1>     jmp     .next
20
21                               <1> .finished:
22 0000000B 29D8                <1>     sub     eax, ebx
23 0000000D 5B                  <1>     pop     ebx
```

Файл листинга содержит:

- Номер строки
- Адрес в сегменте
- Машинный код
- Исходный текст программы

## 4.4 Анализ ошибок

**Ошибка при удалении операнда:**

```
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ cp lab7-2.asm lab7-2-er
g.asm
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nano lab7-2-errg.asm
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2-e
rrg.asm lab7-2-errg.asm
lab7-2-error.asm:15: error: invalid combination of opcode and operands
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$
```

**Вывод:** Транслятор NASM обнаруживает синтаксические ошибки и не создает файл листинга при их наличии

## 5 Задание для самостоятельной работы

### 5.1 Программа поиска наименьшего числа (smallest.asm)

**Код программы:** (см. файл smallest.asm)

**Результат выполнения:**

```
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf smallest.as
m
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 smallest
.o -o smallest
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ./smallest
Наименьшее число: 17
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$
```

**Вывод:** Программа корректно находит наименьшее число среди a=17, b=23, c=45 используя сравнения и условные переходы.

### 5.2 Программа вычисления функции f(x) (function.asm)

**Функция по варианту 1:**

$$f(x) = \begin{cases} 2a - x, & \text{если } x < a \\ 8, & \text{если } x \geq a \end{cases}$$

**Тестирование программы:**

Тест 1 - x=1, a=2:

```
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nano function.asm
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf function.asm
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 function
.o -o function
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ./function
Введите x: 1
Введите a: 2
Результат f(x): 3
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$
```

## Тест 2 - x=2, a=1:

```
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nano function.asm
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf function.asm
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 function
.o -o function
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ./function
Введите x: 1
Введите a: 2
Результат f(x): 3
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$ ./function
Введите x: 2
Введите a: 1
Результат f(x): 8
aayusuf@yusufabdulnasir-VirtualBox:~/work/arch-pc/lab07$
```

**Вывод:** Программа правильно вычисляет значение кусочной функции в зависимости от условия  $x < a$  или  $x \geq a$ .

## 6 Выводы

В ходе лабораторной работы №7 были успешно выполнены все поставленные задачи:

- **Изучены команды переходов** - освоены безусловные переходы (jmp) и условные переходы (cmp, jg, jl и др.)

- **Приобретены практические навыки** написания программ с использованием переходов
- **Изучена работа с флагами** процессора и их влияние на условные переходы
- **Освоена структура файла листинга** и его использование для отладки программ
- **Созданы и протестированы 4 программы** на языке ассемблера NASM
- **Все программы успешно компилируются** и работают корректно