

# **Лабораторная работа №7**

**Команды безусловного и условного переходов в Nasm.  
Программирование ветвлений.**

Жукова Арина Александровна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Реализация переходов в NASM . . . . .	6
2.2	Изучение структуры файлы листинга . . . . .	12
2.3	Задания для самостоятельной работы . . . . .	13
<b>3</b>	<b>Выводы</b>	<b>19</b>
	<b>Список литературы</b>	<b>20</b>

## Список иллюстраций

2.1	Создание файла . . . . .	6
2.2	Ввод текста программы . . . . .	7
2.3	Создание и проверка работы исполняемого файла . . . . .	8
2.4	Редактирование текста программы . . . . .	8
2.5	Проверка работы программы . . . . .	9
2.6	Редактирование текста программы . . . . .	10
2.7	Проверка работы программы . . . . .	11
2.8	Создание файла . . . . .	11
2.9	Проверка работы программы . . . . .	11
2.10	Создание файла листинга . . . . .	12
2.11	Строки файла листинга . . . . .	12
2.12	Результат работы программы . . . . .	15
2.13	Результат работы программы . . . . .	18

## **Список таблиц**

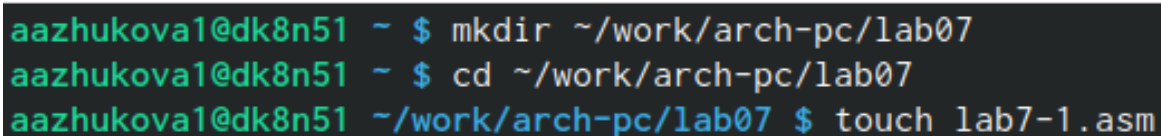
# 1 Цель работы

Цель данной лабораторной работы изучить команды условного и безусловного переходов, приобрести навыки написания программ с использованием переходов, познакомиться с назначением и структурой файла листинга.

## 2 Выполнение лабораторной работы

### 2.1 Реализация переходов в NASM

1. Создадим файлы lab7-1.asm в каталоге для программ лабораторной работы №7 (рис. 2.1).

A screenshot of a terminal window with a dark background. It shows three lines of text: the first line is 'aazhukova1@dk8n51 ~ \$ mkdir ~/work/arch-pc/lab07', the second line is 'aazhukova1@dk8n51 ~ \$ cd ~/work/arch-pc/lab07', and the third line is 'aazhukova1@dk8n51 ~/work/arch-pc/lab07 \$ touch lab7-1.asm'. The text is colored in a way that matches a standard terminal theme, with green for the prompt and red for the command.

```
aazhukova1@dk8n51 ~ $ mkdir ~/work/arch-pc/lab07
aazhukova1@dk8n51 ~ $ cd ~/work/arch-pc/lab07
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ touch lab7-1.asm
```

Рис. 2.1: Создание файла

2. Введём в файл lab7-1.asm текст программы из листинга 7.1 (рис. 2.2).

```
lab7-1.asm      [-M--] 14 L:[ 1+26 2
%include      'in_out.asm'

SECTION .data
msg1:  DB  'Сообщение № 1',0
msg2:  DB  'Сообщение № 2',0
msg3:  DB  'Сообщение № 3',0

SECTION .data
GLOBAL _start
_start:

    jmp _label2

_label1:
    mov  eax, msg1
    call sprintf

_label2:
    mov  eax, msg2
    call sprintf

_label3:
    mov  eax, msg3
    call sprintf

_end:
    call quit
```

Рис. 2.2: Ввод текста программы

Создаём исполняемый файл и проверяем его работу (рис. 2.3).

```

aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 3

```

Рис. 2.3: Создание и проверка работы исполняемого файла

Изменение текста программы файла так, чтобы программа сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу (рис. 2.4).

```

lab7-1.asm [-M--] 14 L:[ 1+26 27/ 27]
#include 'in_out.asm'

SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0

SECTION .data
GLOBAL _start
_start:

_label1:
    mov    eax, msg1
    call   sprintf

_label2:
    mov    eax, msg2
    call   sprintf

    jmp    _label1

_label3:
    mov    eax, msg3
    call   sprintf

_end:
    call   quit

```

Рис. 2.4: Редактирование текста программы



Создание исполняемого файла и проверка его работы (рис. 2.5).

```
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 1
```

Рис. 2.5: Проверка работы программы

Изменим текст программы так, чтобы программа сначала ‘Сообщение № 3’, ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу (рис. 2.6).

```
/afs/.dk.sci.pfu.edu.ru/home/a/a/aazhukova1
%include 'in_out.asm'

SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0

SECTION .data
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: Редактирование текста программы

Создание исполняемого файла и проверка его работы (рис. 2.7).

```

aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1

```

Рис. 2.7: Проверка работы программы

3. Создадим файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Введём текст программы из листинга 7.3 в lab7-2.asm (рис. 2.8).

```

aazhukova1@dk8n51 ~ $ mkdir ~/work/arch-pc/lab07
aazhukova1@dk8n51 ~ $ cd ~/work/arch-pc/lab07
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ touch lab7-1.asm

```

Рис. 2.8: Создание файла

Проверим работу программы, введя различные значения В (рис. 2.9).

```

aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./lab7-2
Введите В: 2
Наибольшее число: 50
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./lab7-2
Введите В: 4
Наибольшее число: 50
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./lab7-2
Введите В: 123
Наибольшее число: 123
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./lab7-2
Введите В: 3478
Наибольшее число: 3478

```

Рис. 2.9: Проверка работы программы

## 2.2 Изучение структуры файлы листинга

4. Создадим файл листинга для программы из файла lab7-2.asm, указав ключ -l и задав имя файла листинга в командной строке (рис. 2.10).

```
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ mcedit lab7-2.lst

aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ls
in_out.asm  lab7-1  lab7-1.asm  lab7-1.o  lab7-2  lab7-2.asm  lab7-2.lst  lab7-2.o
```

Рис. 2.10: Создание файла листинга

Опишем приведенные на рисунке 2.11 строки (рис. 2.11).

32	0000001B	89C1	<1>	mov	ecx, eax
33	0000001D	BB01000000	<1>	mov	ebx, 1
34	00000022	B804000000	<1>	mov	eax, 4

Рис. 2.11: Строки файла листинга

- 1) 32 - номер строки файла листинга, 0000001B - адрес (смещение машинного кода от начала текущего сегмента), 89C1 - машинный код, представляющая собой ассемблированную исходную строку в виде шестнадцатеричной последовательности, mov ecx, eax - исходный тест программы (приравнивающий значение ecx к eax).
- 2) 33 - номер строки файла листинга, 0000001D - адрес (смещение машинного кода от начала текущего сегмента), 8BB01000000 - машинный код, представляющая собой ассемблированную исходную строку в виде шестнадцатеричной последовательности, mov ebx, 1 - исходный тест программы (приравнивающий значение ebx к 1).
- 3) 34 - номер строки файла листинга, 00000022 - адрес (смещение машинного кода от начала текущего сегмента), 8B804000000 - машинный код, представ-

ляющая собой ассемблированную исходную строку в виде шестнадцатеричной последовательности, mov ebx, 1 - исходный тест программы (приравнивающий значение eax к 4).

## 2.3 Задания для самостоятельной работы

1. Программа нахождения наименьшей из 3 целочисленных переменных a, b и c.

```
%include 'in_out.asm'

section .data
    msg db "Наибольшее число: ",0h
    msgA db "Введите a: ",0h
    msgB db "Введите b: ",0h
    msgC db "Введите c: ",0h

section .bss
    max resb 10
    A resb 10
    B resb 10
    C resb 10

section .text
    global _start
_start:
; ----- Ввод A, B, C -----
    mov eax,msgA
    call sprint

    mov eax,A
    mov edx,10
    call sread
```

```

mov eax,msgB
call sprint

mov ebx,B
mov edx,10
call sread

mov eax,msgC
call sprint

mov ecx,C
mov edx,10
call sread
; ----- Преобразование -----
mov eax,B
call atoi
mov [B],eax
; ----- Записываем A в переменную max -----
mov ecx,[A]
mov [max],ecx ; max = A
; ----- Сравниваем A и C -----
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]
jg fin
mov ecx, [B]
mov [max], ecx

```

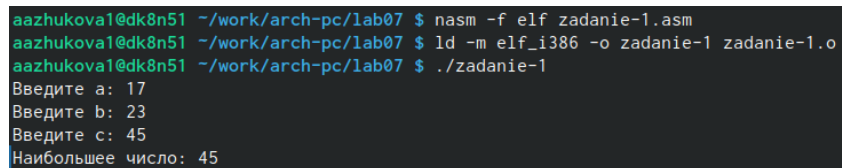
fin:

```

mov eax, msg
call sprint
mov eax, [max]
call iprintLF
call quit

```

Результат (рис. 2.12).



```

aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ nasm -f elf zadanie-1.asm
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o zadanie-1 zadanie-1.o
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./zadanie-1
Введите a: 17
Введите b: 23
Введите c: 45
Наибольшее число: 45

```

Рис. 2.12: Результат работы программы

2. Программа, которая для введенных с клавиатуры значений  $x$  и  $a$  вычисляет значение заданной функции  $f(x)$  и выводит результат вычислений ( $2a - x$ ,  $x < a$ ,  $8$ ,  $x \geq a$ ).

```

#include 'in_out.asm'

section .data
    result_msg db "Результат: ", 0h

```

```

    msgA db "Введите a: ",0h
    msgX db "Введите x: ",0h
section .bss
    a    resb 10
    x    resb 10
    result resd 1
section      .text
    global _start
_start:
; ----- Ввод A -----
    mov eax,msgA
    call sprint

    mov ecx,a
    mov edx,10
    call sread

; ----- Преобразование A -----
    mov eax,a
    call atoi
    mov [a],eax

; ----- Ввод X -----
    mov eax,msgX
    call sprint

    mov ecx,x
    mov edx,10
    call sread

```



; ----- Преобразование X -----

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

; ----- Сравнение -----

```
mov eax, [a]
cmp eax, [x]
jge less_A

mov eax, result_msg
call sprint
mov eax,8
call iprintLF
call quit
```

less\_A:

```
mov eax,[a]
mov ebx,2
mul ebx
sub eax, [x]
mov [result], eax

mov eax, result_msg
call sprint
mov eax, [result]
call iprintLF
```

`call quit`

Результат (рис. 2.13).

```
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ nasm -f elf zadanie-2.asm
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o zadanie-2 zadanie-2.o
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./zadanie-2
Введите а: 2
Введите х: 1
Результат: 3
aazhukova1@dk8n51 ~/work/arch-pc/lab07 $ ./zadanie-2
Введите а: 1
Введите х: 2
Результат: 8
```

Рис. 2.13: Результат работы программы

## 3 Выводы

Нами были изучены команды условного и безусловного переходов, приобретены навыки написания программ с использованием переходов, мы познакомились с назначением и структурой файла листинга.

## **Список литературы**