

# **Отчёта по лабораторной работе №7**

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

Газизянов Владислав Альбертович

# Содержание

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Цель работы</b>                           | <b>4</b>  |
| <b>2</b> | <b>Задание</b>                               | <b>5</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b>        | <b>6</b>  |
| 3.1      | Реализация переходов в NASM . . . . .        | 6         |
| 3.2      | Изучение структуры файлы листинга . . . . .  | 10        |
| 3.3      | Задание для самостоятельной работы . . . . . | 12        |
| <b>4</b> | <b>Выводы</b>                                | <b>17</b> |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 3.1  | Создаем каталог с помощью команды <code>mkdir</code> и файл с помощью команды <code>touch</code> . . . . . | 6  |
| 3.2  | Заполняем файл . . . . .   | 7  |
| 3.3  | Запускаем файл и смотрим на его работу . . . . .   | 7  |
| 3.4  | Изменяем файл . . . . .  | 8  |
| 3.5  | Запускаем файл и смотрим на его работу . . . . .   | 8  |
| 3.6  | Создаем файл командой <code>touch</code> . . . . .   | 9  |
| 3.7  | Заполняем файл . . . . .   | 10 |
| 3.8  | Смотрим на работу программ . . . . .   | 10 |
| 3.9  | Изучаем файл . . . . .   | 11 |
| 3.10 | Удаляем операндум из файла . . . . .   | 11 |
| 3.11 | Транслируем файл . . . . .   | 11 |
| 3.12 | Создаем файл командой <code>touch</code> . . . . .   | 12 |
| 3.13 | Пишем программу . . . . .  | 13 |
| 3.14 | Смотрим на работу программы(всё верно) . . . . .   | 14 |
| 3.15 | Создаем файл командой <code>touch</code> . . . . .   | 14 |
| 3.16 | Пишем программу . . . . .  | 15 |
| 3.17 | Проверяем работу программы . . . . .   | 15 |
| 3.18 | Проверяем работу программы . . . . .   | 16 |

# 1 Цель работы

Освоить условного и безусловного перехода. Ознакомиться с назначением и структурой файла листинга.

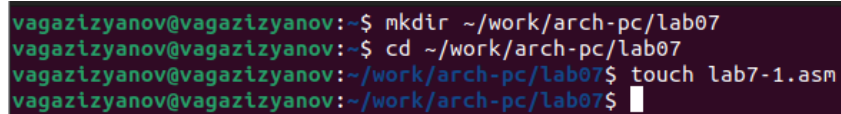
## 2 Задание

Написать программы для решения системы выражений.

## 3 Выполнение лабораторной работы

### 3.1 Реализация переходов в NASM

Создаем каталог для программ ЛБ7, и в нем создаем файл (рис. 3.2).



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

Рис. 3.1: Создаем каталог с помощью команды `mkdir` и файл с помощью команды `touch`

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 7.1 (рис. ??).

```

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label2
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 _label2:
14 mov eax, msg2 ; Вывод на экран строки
15 call sprintf ; 'Сообщение № 2'
16 _label3:
17 mov eax, msg3 ; Вывод на экран строки
18 call sprintf ; 'Сообщение № 3'
19 _end:
20 call quit ; вызов подпрограммы завершения

```

Рис. 3.2: Заполняем файл

Создаем исполняемый файл и запускаем его (рис. 3.3).

```

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

```

Рис. 3.3: Запускаем файл и смотрим на его работу

Снова открываем файл для редактирования и изменяем его в соответствии с листингом 7.2 (рис. 3.4).

```

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label3
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 jmp _end
14 _label2:
15 mov eax, msg2 ; Вывод на экран строки
16 call sprintf ; 'Сообщение № 2'
17 jmp _label1
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение № 3'
21 jmp _label2
22 _end:
23 call quit ; вызов подпрограммы завершения

```

Рис. 3.4: Изменяем файл

Создаем исполняемый файл и запускаем его (рис. 3.5).

```

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

```

Рис. 3.5: Запускаем файл и смотрим на его работу

Создаем новый файл (рис. 3.6).





Рис. 3.6: Создаем файл командой touch

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 7.3 (рис. 3.7).

```

1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите B: ',0h
4 msg2 db "Наибольшее число: ",0h
5 A dd '20'
6 C dd '50'
7 section .bss
8 max resb 10
9 B resb 10
10 section .text
11 global _start
12 _start:
13 ; ----- Вывод сообщения 'Введите B: '
14 mov eax,msg1
15 call sprint
16 ; ----- Ввод 'B'
17 mov ecx,B
18 mov edx,10
19 call sread
20 ; ----- Преобразование 'B' из символа в число
21 mov eax,B
22 call atoi ; Вызов подпрограммы перевода символа в число
23 mov [B],eax ; запись преобразованного числа в 'B'
24 ; ----- Записываем 'A' в переменную 'max'
25 mov ecx,[A] ; 'ecx = A'
26 mov [max],ecx ; 'max = A'
27 ; ----- Сравниваем 'A' и 'C' (как символы)
28 cmp ecx,[C] ; Сравниваем 'A' и 'C'
29 jg check_B ; если 'A>C', то переход на метку 'check_B',
30 mov ecx,[C] ; иначе 'ecx = C'
31 mov [max],ecx ; 'max = C'
32 ; ----- Преобразование 'max(A,C)' из символа в число
33 check_B:
34 mov eax,max
35 call atoi ; Вызов подпрограммы перевода символа в число
36 mov [max],eax ; запись преобразованного числа в 'max'
37 ; ----- Вывод сообщения 'Наибольшее число: '
38 mov eax,msg2
39 call sprint
40 ; ----- Вывод 'max'
41 mov ecx,[max]
42 mov edx,10
43 call sread
44 ; ----- Завершение программы
45 mov eax,0
46 int3

```

Рис. 3.7: Заполняем файл

Создаем исполняемый файл и проверяем его работу, вводя разные значения B (рис. 3.8).

```

vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 23
Наибольшее число: 50

```

Рис. 3.8: Смотрим на работу программ

## 3.2 Изучение структуры файлы листинга

Открываем файл листинга с помощью команды `mcedit` и изучаем его (рис. 3.9).

```

/home/va~7-2.lst  [----]  0 L: 1+ 0 1/225] *(0 /14458b) 0032 0x020 [*][X]
1      %include 'in_out.asm'
2      <1> ;----- slen -----
3      <1> ; Функция вычисления длины сообщения
4      <1> slen:-----
5      00000000 53      <1> push    ebx-----
6      00000001 89C3    <1> mov     ebx, eax-----
7      <1> .....
8      <1> nextchar:-----
9      00000003 803800  <1> cmp     byte [eax], 0...
10     00000006 7403    <1> jz      finished-----
11     00000008 40      <1> inc     eax-----
12     00000009 EBF8    <1> jmp     nextchar-----
13     <1> .....
14     <1> finished:-----
15     0000000B 29D8    <1> sub     eax, ebx-----
16     0000000D 5B      <1> pop     ebx-----
17     0000000E C3      <1> ret-----
18     <1> .
19     <1> .
20     <1> ;----- sprint -----
21     <1> ; Функция печати сообщения
22     <1> ; входные данные: mov eax, <message>
1Помощь 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход

```

Рис. 3.9: Изучаем файл

Строка 33: 0000001D-адрес в сегменте кода, BB01000000-машинный код, mov ebx,1-присвоение переменной ebx значения 1.

Строка 34: 00000022-адрес в сегменте кода, B804000000-машинный код, mov eax,4-присвоение переменной eax значения 4.

Строка 35 00000027-адрес в сегменте кода, CD80-машинный код, int 80h-вызов ядра.

Открываем файл и удаляем один операндум (рис. 3.10).

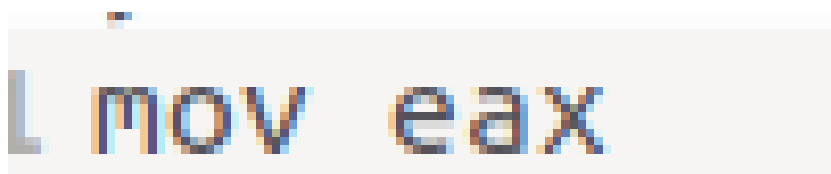


Рис. 3.10: Удаляем операндум из файла

Транслируем с получением файла листинга (рис. 3.11).

```

vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2
.asm
lab7-2.asm:21: error: invalid combination of opcode and operands

```

Рис. 3.11: Транслируем файл

При трансляции файла, выдается ошибка, но создаются исполнительный файл lab7-2 и lab7-2.lst

### 3.3 Задание для самостоятельной работы

#### ВАРИАНТ-2

1. 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 № 7. Создайте исполняемый файл и проверьте его работу.

Создаем новый файл (рис. 3.12).



```
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ touch lab7-3.asm
```

Рис. 3.12: Создаем файл командой touch

Открываем его и пишем программу, которая выберет наименьшее число из трех (2 числа уже в программе, 3е вводится из консоли) (рис. 3.13).

```

1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите B: ',0h
4 msg2 db "Наименьшее число: ",0h
5 A dd '82'
6 C dd '61'
7 section .bss
8 min resb 10
9 B resb 10
0 section .text
1 global _start
2 _start:
3 mov eax,msg1
4 call sprint
5 mov ecx,B
6 mov edx,10
7 call sread
8 mov eax,B
9 call atoi
0 mov [B],eax
1 mov ecx,[A]
2 mov [min],ecx
3 cmp ecx,[C]
4 jl check_B
5 mov ecx,[C]
6 mov [min],ecx
7 check_B:
8 mov eax,min
9 call atoi
0 mov [min],eax
1 mov ecx,[min]
2 cmp ecx,[B]
3 jl fin
4 mov ecx,[B]
5 mov [min],ecx
6 fin:

```

Рис. 3.13: Пишем программу

Транслируем файл и смотрим на работу программы (рис. 3.14).

```
vagazizyanov@vagazizyanov:~/work/arch-pc/lab06$ cd ~/work/arch-pc/lab07/  
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm  
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o  
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ ./lab7-3  
Введите B: 59  
Наименьшее число: 59
```

Рис. 3.14: Смотрим на работу программы(всё верно)

2. 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.

Создаем новый файл (рис. 3.15).

```
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ touch lab7-4.asm  
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$
```

Рис. 3.15: Создаем файл командой touch

Открываем его и пишем программу, которая решит систему уравнений, при данных, введенных в консоль (рис. 3.16).

```

1 %include 'in_out.asm'
2 section .data
3 msg1 DB 'Введите X: ',0h
4 msg2 DB "Введите A: ",0h
5 ans: DB 'F(x)=',0h
6 section .bss
7 x: RESB 80
8 a: RESB 80
9 res: RESB 80
10 section .text
11 global _start
12 _start:
13 mov eax,msg1
14 call sprint
15 mov ecx,x
16 mov edx,80
17 call sread
18 mov eax,x
19 call atoi
20 mov [x],eax
21 mov eax,msg2
22 call sprint
23 mov ecx,a
24 mov edx,80
25 call sread
26 mov eax,a
27 call atoi
28 mov [a],eax
29 mov eax, [x]
30 cmp eax, [a]
31
32 jl x_smaller_a
33 mov eax, [x]
34 dec eax
35 jmp ansv
36

```

Рис. 3.16: Пишем программу

Транслируем файл и проверяем его работу при  $x=1$  и  $a=1$  (рис. 3.17).

```

vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ nasm -f elf lab7-4.asm
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-4 lab7-4.o
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ ./lab7-4
Введите X: 5
Введите A: 7
F(x)=6

```

Рис. 3.17: Проверяем работу программы

Транслируем файл и проверяем его работу при  $x=2$  и  $a=2$  (рис. 3.18).

```
vagazizyanov@vagazizyanov:~/work/arch-pc/lab07$ ./lab7-4  
Введите X: 6  
Введите A: 4  
F(x)=5
```

Рис. 3.18: Проверяем работу программы



## 4 Выводы

Мы познакомились с структурой файла листинга, изучили команды условного и безусловного перехода.