

Отчёт по лабораторной работе №7

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

Исупов Олег Денисович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Задания для самостоятельной работы	11
4	Выводы	16

Список иллюстраций

2.1	Создание файла	6
2.2	Заполнение файла	7
2.3	Результат	7
2.4	Изменение	8
2.5	Изменение	8
2.6	Создание файла и проверка	8
2.7	Открытие файла	9
2.8	Создание и проверка	9
2.9	Открытие файла	10
3.1	Создание файла	11
3.2	Заполнение файла	12
3.3	Проверка	13
3.4	Изменение файла	14
3.5	Проверка	15
3.6	Изменение	15

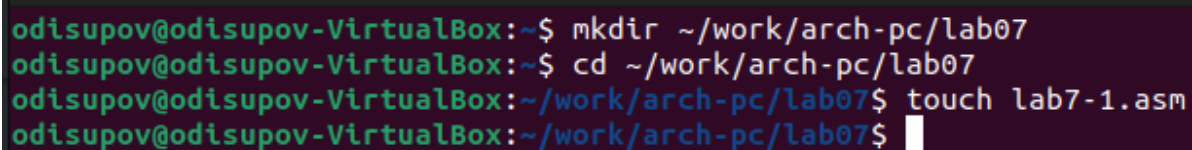
Список таблиц

1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 7, перейдите в него и создайте файл lab7-1.asm

A screenshot of a terminal window with a dark background and light-colored text. It shows four lines of commands being executed in a shell. The first line creates a directory named ~/work/arch-pc/lab07. The second line changes the current directory to ~/work/arch-pc/lab07. The third line uses the 'touch' command to create a file named lab7-1.asm. The fourth line shows the prompt after the command has been executed.

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

Рис. 2.1: Создание файла

2. Инструкция jmp в NASM используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции jmp. Введите в файл lab7-1.asm текст программы из листинга 7.1.

```

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 78 Демидова А. В.
17 Архитектура ЭВМ
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение № 3'
21 _end:
22 call quit ; вызов подпрограммы завершения

```

Рис. 2.2: Заполнение файла

```

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

```

Рис. 2.3: Результат

```

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

```

Рис. 2.4: Изменение

```

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

```

Рис. 2.5: Изменение

3. Создайте файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучите текст программы из листинга 7.3 и введите в lab7-2.asm

```

odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ touch lab7-2.asm
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 10
Наибольшее число: 50
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$

```

Рис. 2.6: Создание файла и проверка


```
odisupov@odisupov-VirtualBox: ~/work/arch-pc/lab07
/home/odisupov/work-pc/lab07/lab7-2.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>
23     <1> sprint:
24     0000000F 52      <1>      push     edx
25     00000010 51      <1>      push     ecx
26     00000011 53      <1>      push     ebx
27     00000012 50      <1>      push     eax
28     00000013 E8E8FFFFFF <1>      call    slen
29     <1>.....
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перенести 7Поиск 8Удалить 9МенюМС 10Выход

Рис. 2.7: Открытие файла

4. Создайте файл листинга для программы из файла lab7-2.asm

```
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:18: error: invalid combination of opcode and operands
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ ls
in_out.asm lab7-1 lab7-1.asm lab7-1.o lab7-2 lab7-2.asm lab7-2.lst
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 2.8: Создание и проверка

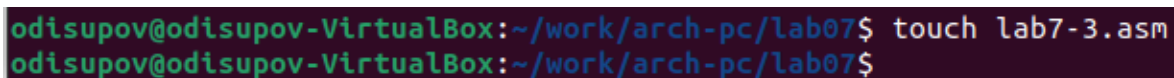
```
odisupov@odisupov-VirtualBox: ~/work/arch-pc/lab07
/home/odisupov/work-pc/lab07/lab7-2.lst [----] 0 L:[ 1+ 0 1/226] *(0 /14546b) 0032 0x020 [*][X]
1      %include 'in_out.asm'
2      <1> ;----- slen -----
3      <1> ; Функция вычисления длины сообщения
4      <1> slen:-----
5      00000000 53      <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>
23     <1> sprint:
24     0000000F 52      <1> push    edx
25     00000010 51      <1> push    ecx
26     00000011 53      <1> push    ebx
27     00000012 50      <1> push    eax
28     00000013 E8E8FFFF <1> call    slen
29     <1> .....
```

1 Помощь 2 Сохранить 3 Блок 4 Замена 5 Копия 6 Переименовать 7 Поиск 8 Удалить 9 Меню 10 Выход

Рис. 2.9: Открытие файла

3 Задания для самостоятельной работы

1. Напишите программу нахождения наименьшей из 3 целочисленных переменных x, y и z . Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу.



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

Рис. 3.1: Создание файла

```

1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите B: ',0h
4 msg2 db "Наименьшее число: ",0h
5 A dd '94'
6 C dd '58'
7 section .bss
8 min resb 10
9 B resb 10
10 section .text
11 global _start
12 _start:
13 mov eax,msg1
14 call sprint
15 mov ecx,B
16 mov edx,10
17 call sread
18 mov eax,B
19 call atoi
20 mov [B],eax
21 mov ecx,[A]
22 mov [min],ecx
23 cmp ecx,[C]
24 jl check_B
25 mov ecx,[C]
26 mov [min],ecx
27 check_B:
28 mov eax,min
29 call atoi
30 mov [min],eax
31 mov ecx,[min]
32 cmp ecx,[B]
33 jl fin
34 mov ecx,[B]
35 mov [min],ecx
36 fin:
37 mov eax, msg2
38 call sprint
39 mov eax,[min]
40 call iprintLF
41 call quit

```

Рис. 3.2: Заполнение файла

```
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ ./lab7-3
Введите В: 5
Наименьшее число: 5
```

Рис. 3.3: Проверка

2. Напишите программу, которая для введенных с клавиатуры значений x и y вычисляет значение заданной функции $f(x, y)$ и выводит результат вычислений. Вид функции $f(x, y)$ 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 y из 7.6.

```

1 %include 'in_out.asm'
2 section .data
3 msg1 DB 'Введите x: ',0h
4 msg2 DB "Введите a: ",0h
5 otv: 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, 3
31 je x_is_3
32 mov eax, [a]
33 add eax, 1
34 jmp calc_res
35 x_is_3:
36 mov eax, [x]
37 imul eax,3
38 calc_res:
39 mov [res],eax
40 fin:
41 mov eax,otv
42 call sprint
43 mov eax,[res]
44 call iprintLF
45 call quit

```

Рис. 3.4: Изменение файла

```
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-4.asm
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-4 lab7-4.o
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ ./lab7-4
Введите x: 1
Введите a: 4
F(x)=5
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 3.5: Проверка

```
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$ ./lab7-4
Введите x: 3
Введите a: 4
F(x)=9
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 3.6: Изменение

4 Выводы

Я изучил команды условного и безусловного переходов. Приобрёл навыки написания программ с использованием переходов. Познакомился с назначением и структурой файла листинга.