

lab-08

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

Владимир Андреевич Баранов

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Самостоятельная работа:	9
4	Выводы	15

Список иллюстраций

2.1	Создание файла.	6
2.2	Проверка работы программы.	6
2.3	Проверка работы программы.	7
2.4	Проверка работы программы.	7
2.5	Проверка работы программы.	7
2.6	Создание файла и открытие с помощью текстового редактора. . .	7
2.7	Ошибка при удалении операнда.	8
3.1	Программа.	10
3.2	Работа программы.	11
3.3	Программа.	12
3.4	Программа.	13
3.5	Работа программы.	14

Список таблиц

1 Цель работы

Целью работы является изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Выполнение лабораторной работы

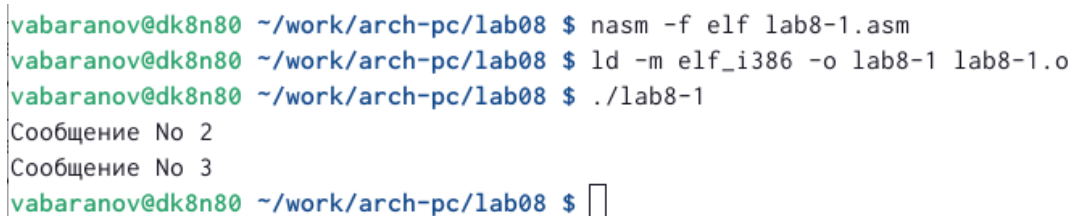
1. Создаю каталог для программ лабораторной работы No 8, перехожу в него и создаю файл lab8-1.asm (рис. 2.1).



```
vabaranov@dk8n80:~/work/arch-pc/lab08
vabaranov@dk8n80 ~ $ mkdir ~/work/arch-pc/lab08
vabaranov@dk8n80 ~ $ cd ~/work/arch-pc/lab08
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ touch lab8-1.asm
vabaranov@dk8n80 ~/work/arch-pc/lab08 $
```

Рис. 2.1: Создание файла.

2. Ввожу в файл lab8-1.asm текст программы из листинга 8.1, проверяю программу (рис. 2.2).



```
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ ./lab8-1
Сообщение No 2
Сообщение No 3
vabaranov@dk8n80 ~/work/arch-pc/lab08 $
```

Рис. 2.2: Проверка работы программы.

3. Изменяю программу таким образом, чтобы она выводила сначала 'Сообщение No 2', потом 'Сообщение No 1' и завершала работу, проверяю работу программы (рис. 2.3).

```

vabaranov@dk8n80 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ ./lab8-1
Сообщение No 2
Сообщение No 1
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ █

```

Рис. 2.3: Проверка работы программы.

4. Изменяю программу еще раз, проверяю ее работу (рис. 2.4).

```

vabaranov@dk8n80 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ ./lab8-1
Сообщение No 3
Сообщение No 2
Сообщение No 1
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ █

```

Рис. 2.4: Проверка работы программы.

5. Создаю файл lab8-2.asm в каталоге ~/work/arch-pc/lab08. Ввожу текст программы из листинга 8.3 в lab8-2.asm (рис. 2.5).

```

vabaranov@dk8n80 ~/work/arch-pc/lab08 $ nasm -f elf lab8-2.asm
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ ./lab8-2
Введите B: 51
Наибольшее число: 51
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ █

```

Рис. 2.5: Проверка работы программы.

6. Создаю файл листинга для программы из файла lab8-2.asm Открываю файл листинга lab8-2.lst с помощью mcedit (рис. 2.6).

```

vabaranov@dk8n80 ~/work/arch-pc/lab08 $ nasm -f elf -l lab8-2.lst lab8-2.asm
vabaranov@dk8n80 ~/work/arch-pc/lab08 $ mcedit lab8-2.lst

```

Рис. 2.6: Создание файла и открытие с помощью текстового редактора.

Объясняю содержимое 3-х строк:

1. 000000F2,000000F7,000000FC-это адрес строки.
2. B9[0A000000],BA0A000000,E842FFFFFF-это машинный код.
3. 'mov ecx,B', 'call atoi', mov [B], eax - это исходный текст программы.
4. Открываю файл с программой lab8-2.asm и в любой инструкции с двумя операндами удаляю один операнд. Выполняю трансляцию с получением файла листинга. В этом случае выдается ошибка (рис. 2.7).

```

215 39 ***** error: invalid combination of opcode and operands

```

Рис. 2.7: Ошибка при удалении операнда.

3 Самостоятельная работа:

1. Пишу программу нахождения наименьшей из 3 целочисленных переменных a,b,c. Создаю исполняемый файл и проверяю его работу (рис. 3.1) (рис. 3.2).

```

1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите B: ',0h
4 msg2 db "Наименьшее число: ",0h
5 A dd '26'
6 C dd '68'
7
8 section .bss
9 min resb 10
10 B resb 10
11 section .text
12
13 global _start
14 _start:
15
16 mov eax,msg1
17 call sprint
18
19 mov ecx,B
20 mov edx,10
21 call sread
22
23 mov eax,B
24 call atoi
25 mov [B],eax
26
27 mov ecx,[A]
28 mov [min],ecx
29
30 cmp ecx,[C]
31 jb check_B
32 mov ecx,[C]
33 mov [min],ecx
34
35 check_B:
36 mov eax,min
37 call atoi
38 mov [min],eax
39
40 mov ecx,[min]
41 cmp ecx,[B]
42 jb fin
43 mov ecx,[B]
44 mov [min],ecx
45

```

Рис. 3.1: Программа.

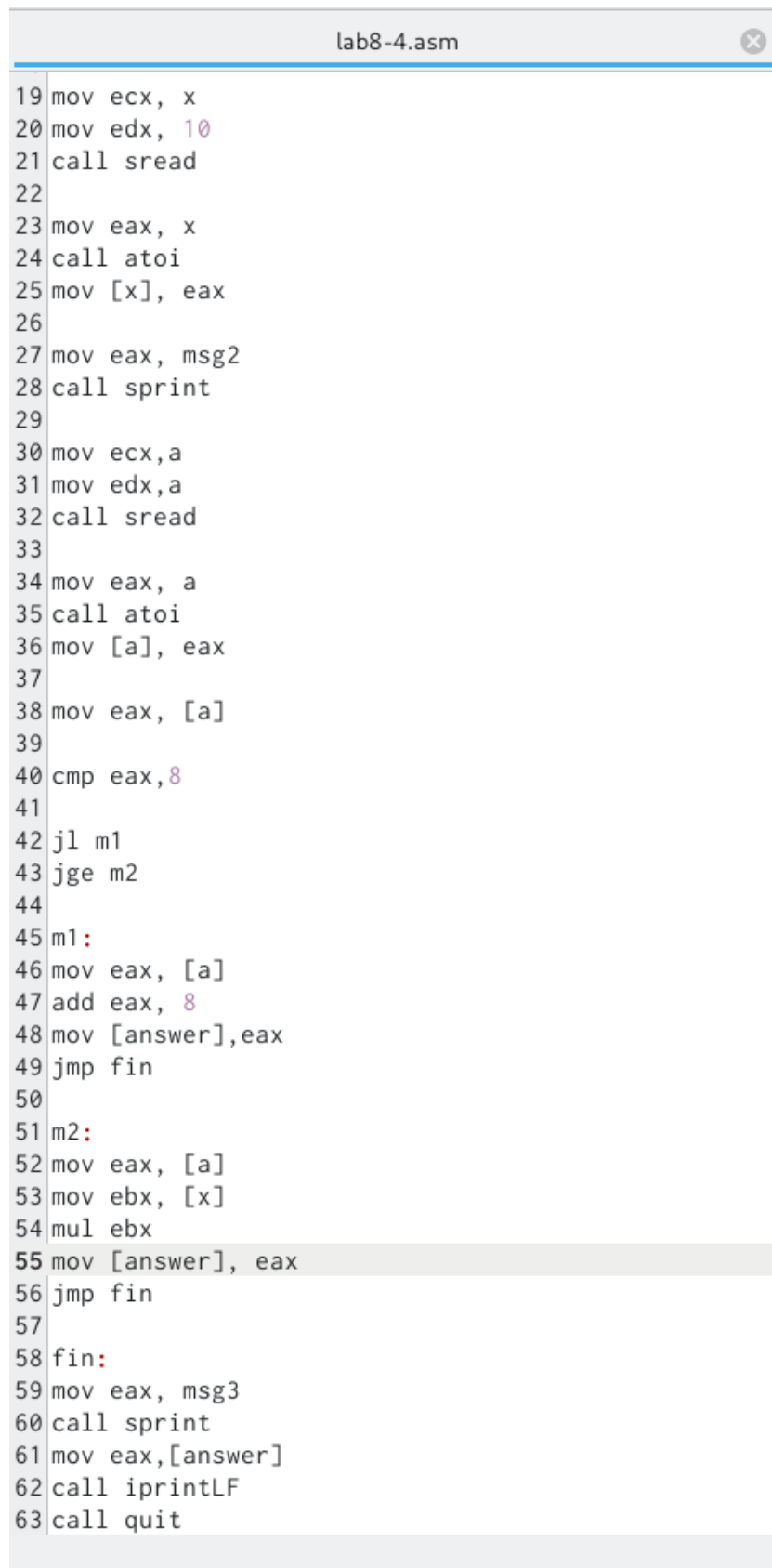
```
vabaranov@dk2n26 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
vabaranov@dk2n26 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
vabaranov@dk2n26 ~/work/arch-pc/lab08 $ ./lab8-3
Введите В: 12
Наименьшее число: 12
vabaranov@dk2n26 ~/work/arch-pc/lab08 $
```

Рис. 3.2: Работа программы.

2. Пишу программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений. Создаю исполняемый файл и проверяю его работу (рис. 3.3) (рис. 3.4) (рис. 3.5).

```
lab8-4.asm
1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите x:',0h
4 msg2 db 'Введите a:',0h
5 msg3 db 'Функция равна:',0h
6
7 section .bss
8 x resb 10
9 a resb 10
10 answer resb 10
11
12 section .text
13 global _start
14 _start:
15
16 mov eax, msg1
17 call sprint
18
19 mov ecx, x
20 mov edx, 10
21 call sread
22
23 mov eax, x
24 call atoi
25 mov [x], eax
26
27 mov eax, msg2
28 call sprint
29
30 mov ecx, a
31 mov edx, a
32 call sread
33
34 mov eax, a
35 call atoi
36 mov [a], eax
37
38 mov eax, [a]
39
40 cmp eax, 8
41
42 jl m1
43 jge m2
44
45 m1:
```

Рис. 3.3: Программа.



```
lab8-4.asm
19 mov ecx, x
20 mov edx, 10
21 call sread
22
23 mov eax, x
24 call atoi
25 mov [x], eax
26
27 mov eax, msg2
28 call sprint
29
30 mov ecx, a
31 mov edx, a
32 call sread
33
34 mov eax, a
35 call atoi
36 mov [a], eax
37
38 mov eax, [a]
39
40 cmp eax, 8
41
42 jl m1
43 jge m2
44
45 m1:
46 mov eax, [a]
47 add eax, 8
48 mov [answer], eax
49 jmp fin
50
51 m2:
52 mov eax, [a]
53 mov ebx, [x]
54 mul ebx
55 mov [answer], eax
56 jmp fin
57
58 fin:
59 mov eax, msg3
60 call sprint
61 mov eax, [answer]
62 call iprintLF
63 call quit
```

Рис. 3.4: Программа.

```
vabaranov@dk2n26 ~/work/arch-pc/lab08 $ nasm -f elf lab8-4.asm
vabaranov@dk2n26 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-4 lab8-4.o
^[[Avabaranov@dk2n26 ~/work/arch-pc/lab08 $ ./lab8-4
Введите x:3
Введите a:4
Функция равна:12
vabaranov@dk2n26 ~/work/arch-pc/lab08 $ nasm -f elf lab8-4.asm
vabaranov@dk2n26 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-4 lab8-4.o
vabaranov@dk2n26 ~/work/arch-pc/lab08 $ ./lab8-4
Введите x:2
Введите a:9
Функция равна:18
vabaranov@dk2n26 ~/work/arch-pc/lab08 $ 
```

Рис. 3.5: Работа программы.

4 Выводы

В ходе данной лабораторной работы я изучил команды условного и безусловного переходов, приобрел навыки написания программ с использованием переходов, ознакомился с назначением и структурой файла листинга.