

Отчет по лабораторной работе №7

Архитектура компьютера

Садыков Ильдар Ильфатович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Программа с безусловными переходами:	7
3.2	Программа с условными переходами:	9
4	Задание для самостоятельной работы	10
4.1	Программа нахождения наименьшего числа:	10
4.2	Программа вычисления функции с системой:	12
5	Выводы	13

Список иллюстраций

3.1	Вывод программы lab7-1.asm	7
3.2	Запуск изменной программы lab7-1	8
3.3	Запуск программы lab7-2 с различными значениями	9
3.4	Ошибка в файле листинга при удалении обязательной операнды . .	9
4.1	Код и вывод программы для нахождения наименьшего числа	11
4.2	Код программы вычисления функции и тестирование с различными значениями	12

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.

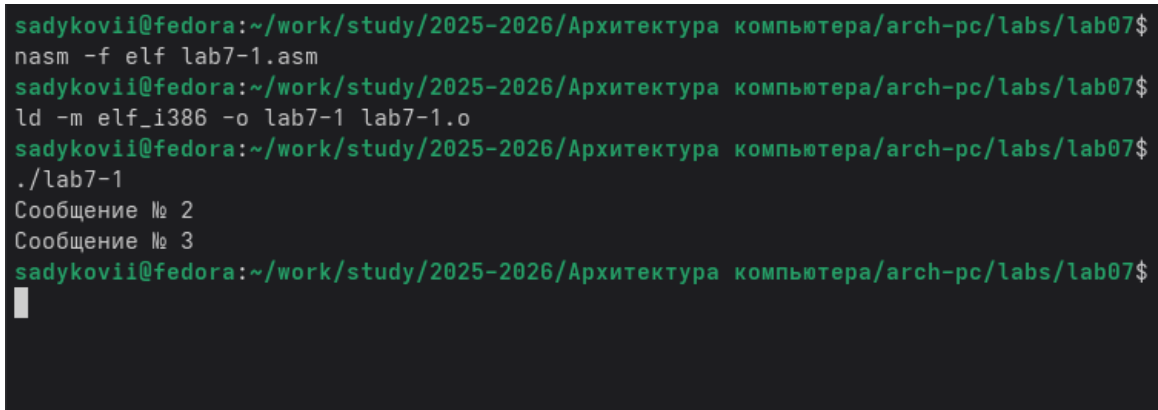
2 Задание

1. Изучить команды безусловного и условного переходов в NASM.
2. Освоить написание программ с использованием инструкций `jmp` и условных переходов.
3. Изучить структуру файла листинга и его назначение.
4. Написать программы для нахождения наименьшего числа и вычисления функции с ветвлениями.

3 Выполнение лабораторной работы

3.1 Программа с безусловными переходами:

Создаем каталог для программ лабораторной работы №7 и создаем файл lab7-1.asm. Вводим текст программы с использованием инструкции jmp для реализации безусловных переходов (рис. 3.1).

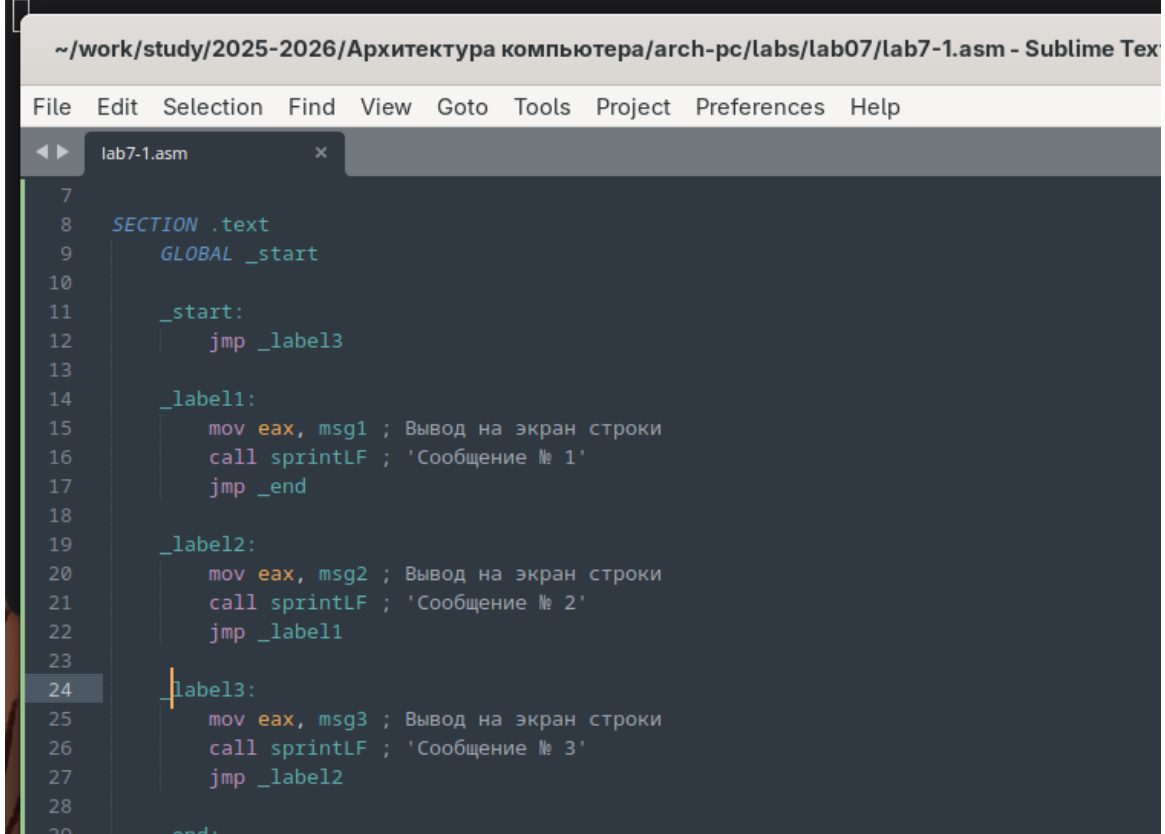


```
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$  
nasm -f elf lab7-1.asm  
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$  
ld -m elf_i386 -o lab7-1 lab7-1.o  
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$  
./lab7-1  
Сообщение № 2  
Сообщение № 3  
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
```

Рисунок 3.1: Вывод программы lab7-1.asm

Изменяем программу для нового условия и проверяем порядок вывода сообщений (рис. 3.2).

```
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
nasm -f elf lab7-1.asm
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
ld -m elf_i386 -o lab7-1 lab7-1.o
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
./lab7-1
Сообщение № 2
Сообщение № 1
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
nasm -f elf lab7-1.asm
lab7-1.asm:12: error: symbol `_label3' not defined
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
nasm -f elf lab7-1.asm
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
ld -m elf_i386 -o lab7-1 lab7-1.o
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
```



```
~ /work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07/lab7-1.asm - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
lab7-1.asm x
7
8 SECTION .text
9 GLOBAL _start
10
11 _start:
12     jmp _label3
13
14 _label1:
15     mov eax, msg1 ; Вывод на экран строки
16     call sprintf ; 'Сообщение № 1'
17     jmp _end
18
19 _label2:
20     mov eax, msg2 ; Вывод на экран строки
21     call sprintf ; 'Сообщение № 2'
22     jmp _label1
23
24 _label3:
25     mov eax, msg3 ; Вывод на экран строки
26     call sprintf ; 'Сообщение № 3'
27     jmp _label2
28
29 end;
```

Рисунок 3.2: Запуск изменной программы lab7-1

3.2 Программа с условными переходами:

Создаем файл lab7-2.asm для нахождения наибольшего из трех чисел. Запускаем программу и проверяем ее работу для разных значений переменной В (рис. 3.3).

```
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$ touch lab7-2.asm
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-2.asm
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите В: 15
Наибольшее число: 50
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите В: 75
Наибольшее число: 75
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
```

Рисунок 3.3: Запуск программы lab7-2 с различными значениями

Убираем одну обязательную операнду из функции и ищем ошибку в файле листинга.(рис. 3.4).

```
33 00000116 8B00[33000000]      mov ecx,[A] ; ecx = A
34 00000116 890D[00000000]      mov [max],ecx ; 'max = A'
35
36                      ; ----- Сравниваем 'A' и 'C' (как символы)
37                      cmp ecx ; Сравниваем 'A' и 'C'
37 *****
error: invalid combination of opcode and operands
```

Рисунок 3.4: Ошибка в файле листинга при удалении обязательной операнды

4 Задание для самостоятельной работы

4.1 Программа нахождения наименьшего числа:

Написана программа для нахождения наименьшей из трех целочисленных переменных a , b и c . Для варианта 7 использованы значения: $a=67$, $b=45$, $c=14$. (рис. 4.1).

```
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
nasm -f elf lab7-3.asm
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
ld -m elf_i386 -o lab7-3 lab7-3.o
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
./lab7-3
Наименьшее число: 15
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
```

~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07/lab7-3.asm - Sublin

File Edit Selection Find View Goto Tools Project Preferences Help

lab7-1.asm x lab7-2.asm x lab7-3.asm x

```
1 %include '/home/sadykovii/Test/in_out.asm'
2
3 section .data
4     msg db 'Наименьшее число: ', 0h
5     B dd '67'
6     A dd '45'
7     C dd '15'
8
9 section .bss
10    min resb 10
11
12 section .text
13    global _start
14
15    _start:
16
17    mov eax,B
18    call atoi ; Вызов подпрограммы перевода символа в число
19    mov [B],eax ; запись преобразованного числа в 'B'
20
21    ; ----- Записываем 'A' в переменную 'min'
22    mov ecx,[A] ; 'ecx = A'
23    mov [min],ecx ; 'min = A'
24
25    ; ----- Сравниваем 'A' и 'C' (как символы)
26    cmp ecx,[C] ; Сравниваем 'A' и 'C'
27    jl check_B ; если 'A<C', то переход на метку 'check_B',
28    mov ecx,[C] ; иначе 'ecx = C'
29    mov [min],ecx ; 'min = C'
30
31    ; ----- Преобразование 'min(A,C)' из символа в число
32    check_B:
33    mov eax,min
34    call atoi
35    mov [min],eax ; запись преобразованного числа в 'min'
```

Рисунок 4.1: Код и вывод программы для нахождения наименьшего числа

4.2 Программа вычисления функции с системой:

Реализована программа вычисления функции с условными переходами согласно варианту 7. Функция имеет различное поведение в зависимости от соотношения значений x и a . (рис. 4.2).

```
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-f.asm
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-f lab7-f.o
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-f
Function:
6*a | x == a
a*x | x != a
Введите a: 1
Введите x: 1
Вывод: 6
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-f
Function:
6*a | x == a
a*x | x != a
Введите a: 2
Введите x: 1
Вывод: 3
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab07$
```

```
11  x resb 10
12
13  section .text
14  global _start
15
16  _start:
17
18  mov eax,msg_output_function
19  call sprintf
20
21  mov eax,msg_input_a
22  call sprint
23  mov ecx,a
24  mov edx,10
25  call tread
26  mov eax,a
27  call atoi
28  mov [a],eax
29
30  mov eax,msg_input_x
31  call sprint
32  mov ecx,x
33  mov edx,10
34  call tread
35  mov eax,x
36  call atoi
37  mov [x],eax
38
39  mov ecx,[a]
40  cmp ecx,[x]
41  jne check
42
43  mov ebx,6
44  mov eax,[a]
45  mul ebx
46  mov edx,eax
47  mov eax,msg_output_answer
48  call sprint
49  mov eax,edx
50  call sprintf
51
52  call quit
53
54  check:
55  mov ebx,[x]
56  mov eax,[a]
57  add eax,ebx
```

Рисунок 4.2: Код программы вычисления функции и тестирование с различными значениями

5 Выводы

В ходе выполнения лабораторной работы были изучены команды условного и безусловного переходов в языке ассемблера NASM. Освоено написание программ с использованием инструкций `jmp` и условных переходов. Приобретены навыки работы с файлами листинга и анализа их структуры. Написаны программы для решения практических задач с ветвлениями, что позволило закрепить полученные знания о механизмах передачи управления в программах на ассемблере.