

# **ОТЧЕТ по лабораторной работе №7**

Полякова Юлия Александровна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Результаты выполнения лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Результаты выполнения заданий для самостоятельной работы</b>	<b>13</b>
<b>4</b>	<b>Вывод</b>	<b>17</b>

# Список иллюстраций

2.1	Создание каталога и lab7-1.asm . . . . .	6
2.2	lab6-1.asm из листинга 7.1. . . . .	7
2.3	Запуск lab7-1.asm из листинга 7.1. . . . .	7
2.4	Измененный lab7-1.asm по листингу 7.2 . . . . .	8
2.5	Запуск измененного lab7-1.asm по листингу 7.2 . . . . .	8
2.6	Измененный текст для нужного вывода . . . . .	9
2.7	Запуск измененной программы . . . . .	9
2.8	lab7-2.asm из листинга 7.3. . . . .	10
2.9	Запуск lab7-2.asm из листинга 7.3. . . . .	10
2.10	Создаем файл листинга . . . . .	11
2.11	Строки для объяснения . . . . .	11
2.12	Удаляем операнд . . . . .	12
2.13	Трансляция с созданием листинга . . . . .	12
2.14	Листинг, строка с ошибкой . . . . .	12
3.1	Листинг lab7-1-1.asm . . . . .	14
3.2	Запуск lab7-1-1.asm . . . . .	14
3.3	Листинг lab7-1-2.asm . . . . .	15
3.4	Запуск lab7-1-2.asm . . . . .	16

## **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 Результаты выполнения лабораторной работы

1. Создаем каталог для программ лабораторной №7, переходим в него и создаем файл lab7-1.asm (Рис. 1).

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

Рис. 2.1: Создание каталога и lab7-1.asm

2. Записываем в файл программу из листинга 7.1. (Рис. 2).

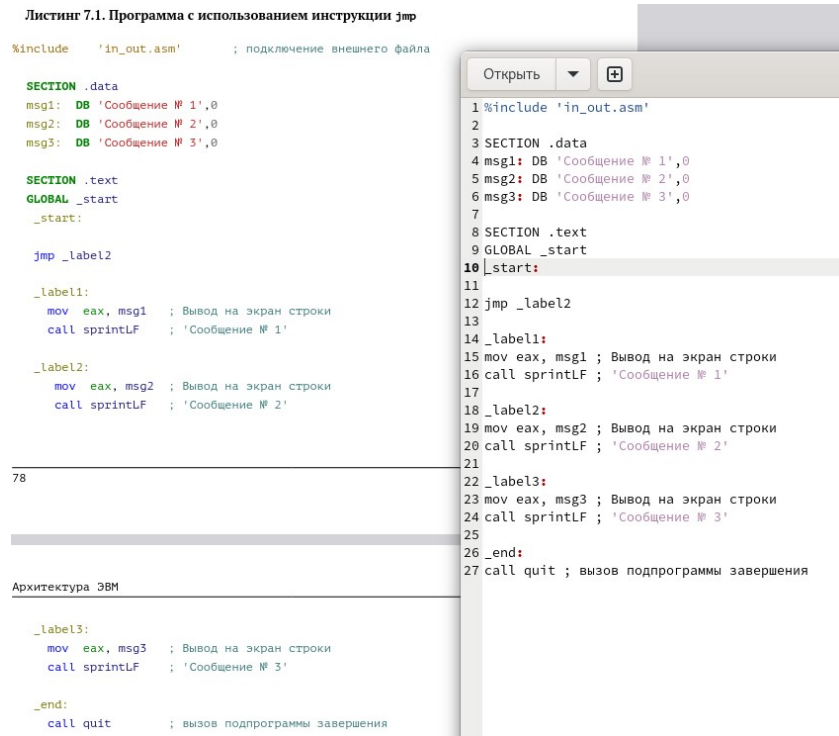


Рис. 2.2: lab6-1.asm из листинга 7.1.

3. Создаем исполняемый файл и запускаем его (Рис. 3).

```

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

```

Рис. 2.3: Запуск lab7-1.asm из листинга 7.1.

4. Изменяем текст программы в соответствии с листингом 7.2 (Рис. 4)

Листинг 7.2. Программа с использованием инструкции `jmp`

```

%include 'in_out.asm' ; подключение внешнего файла

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

SECTION .text
GLOBAL _start
_start:

jmp _label2

_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение № 1'
jmp _end

_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение № 2'
jmp _label1

_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение № 3'

_end:
call quit ; вызов подпрограммы завершения

```

Демидова А. В.

Рис. 2.4: Измененный lab7-1.asm по листингу 7.2

5. Создаем исполняемый файл и запускаем его (Рис. 5).

```

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

```

Рис. 2.5: Запуск измененного lab7-1.asm по листингу 7.2

6. Изменяем программу для требуемого вывода (Рис. 6).



```

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1: DB 'Сообщение № 1',0
5 msg2: DB 'Сообщение № 2',0
6 msg3: DB 'Сообщение № 3',0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 jmp _label3
13
14 _label1:
15 mov eax, msg1 ; Вывод на экран строки
16 call sprintLF ; 'Сообщение № 1'
17 jmp _end
18
19 _label2:
20 mov eax, msg2 ; Вывод на экран строки
21 call sprintLF ; 'Сообщение № 2'
22 jmp _label1
23
24 _label3:
25 mov eax, msg3 ; Вывод на экран строки
26 call sprintLF ; 'Сообщение № 3'
27 jmp _label2
28
29 _end:
30 call quit ; вызов подпрограммы завершения

```

Рис. 2.6: Измененный текст для нужного вывода

7. Создаем исполняемый файл и запускаем его (Рис. 7).

```

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

```

Рис. 2.7: Запуск измененной программы

8. Создаем lab7-2.asm по листингу 7.3. (Рис. 8).

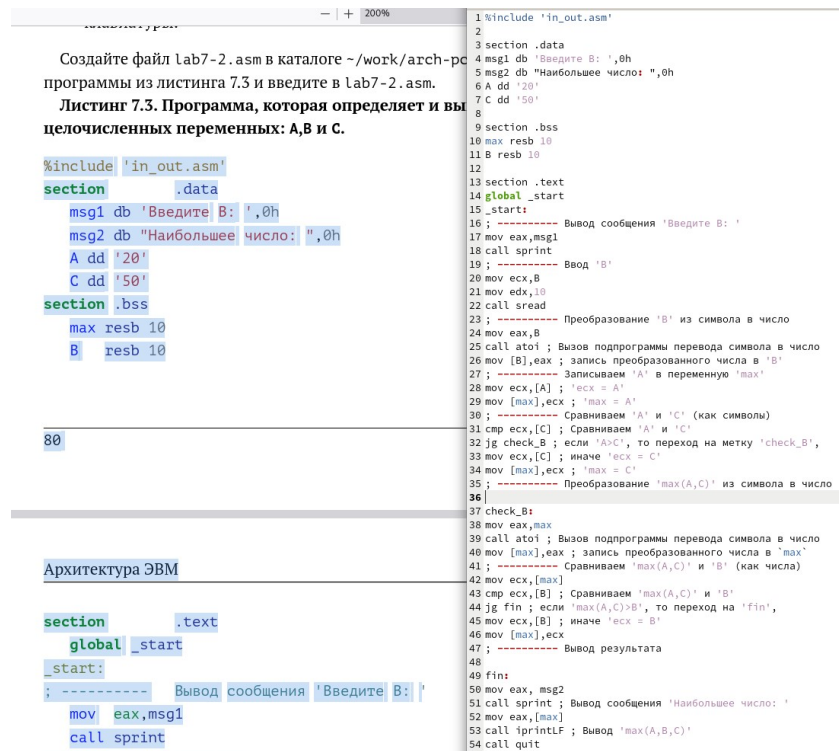


Рис. 2.8: lab7-2.asm из листинга 7.3.

9. Создаем исполняемый файл и запускаем его для разных значений (Рис. 9).

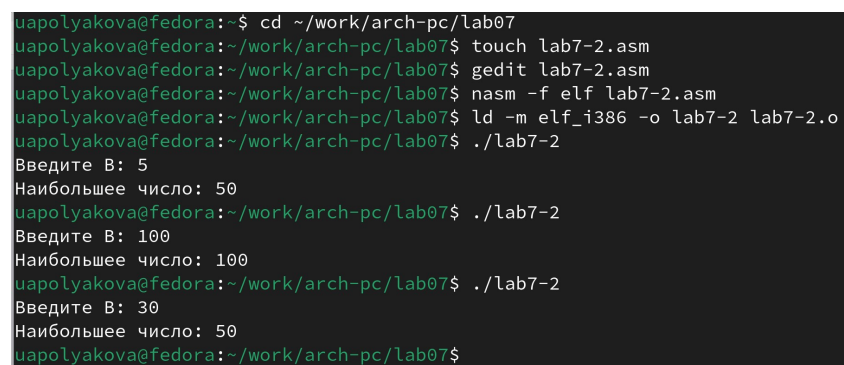


Рис. 2.9: Запуск lab7-2.asm из листинга 7.3.

10. Создаем файл листинга для lab7-2.asm и открываем его любым редактором. Был выбран gedit (Рис. 10).

```
uapolyakova@fedora:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
uapolyakova@fedora:~/work/arch-pc/lab07$ gedit lab7-2.lst
```

Рис. 2.10: Создаем файл листинга

11. Для объяснения выбраны последние строки файла in\_out.asm (Рис. 11).

164	163	<1>	
165	164	<1>	;----- quit -----
166	165	<1>	; Функция завершения программы
167	166	<1>	quit:
168	167 000000DB BB00000000	<1>	mov ebx, 0
169	168 000000E0 B801000000	<1>	mov eax, 1
170	169 000000E5 CD80	<1>	int 80h
171	170 000000E7 C3	<1>	ret
172	2		
173	3		section .data
174	4 00000000 D092D0B2D0B5D0B4D0-		msg1 db 'Введите B: ',0h
175	4 00000009 B8D182D0B520423A20-		
176	4 00000012 00		
177	5 00000013 D09DD0B0D0B8D0B1D0-		msg2 db "Наибольшее число: ",0h
178	5 0000001C BED0BBD18CD188D0B5-		
179	5 00000025 D0B520D187D0B8D181-		
180	5 0000002E D0BBD0BE3A2000		
181	6 00000035 32300000	A dd	'20'
182	7 00000039 35300000	C dd	'50'

Рис. 2.11: Строки для объяснения

Каждая строка состоит из дополнительной информации и строк кода из основной программы. В данном случае дополнительная информация состоит из трех столбцов. Первый - это номер строки в основном коде (167, 168, 169). Второй - адрес (то есть смещение машинного кода от начала текущего сегмента). Третий - это инструкция на машинном коде, то есть CD80 - это инструкция прерывания ядра.

12. Удаляем один операнд из lab7-2.asm (Рис. 12).

```

23 ; ----- Преобразование 'B' из символа в число
24 mov eax,B
25 call atoi ; Вызов подпрограммы перевода символа в число
26 mov [B],eax ; запись преобразованного числа в 'B'
27 ; ----- Записываем 'A' в переменную 'max'
28 mov ecx,[A] ; 'ecx = A'
29 mov [max],ecx ; 'max = A'
30 ; ----- Сравниваем 'A' и 'C' (как символы)
31 cmp ecx ; Сравниваем 'A' и 'C' Removed ,[C] !!!!!!!!!!!
32 jg check_B ; если 'A>C', то переход на метку 'check_B',
33 mov ecx,[C] ; иначе 'ecx = C'
34 mov [max],ecx ; 'max = C'
35 ; ----- Преобразование 'max(A,C)' из символа в число
36
37 check_B:

```

Рис. 2.12: Удаляем операнд

13. Выполняем трансляцию с созданием листинга. Создался только листинг (исполняемый файл остался с прошлых запусков) (Рис. 13).

```
upolyakova@fedora:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:31: error: invalid combination of opcode and operands
upolyakova@fedora:~/work/arch-pc/lab07$ ls
in_out.asm  lab7-1  lab7-1.asm  lab7-1.o  lab7-2  lab7-2.asm  lab7-2.lst
upolyakova@fedora:~/work/arch-pc/lab07$ gedit lab7-2.lst
```

Рис. 2.13: Трансляция с созданием листинга

В листинге добавляется строчка с ошибкой (Рис. 14).

```

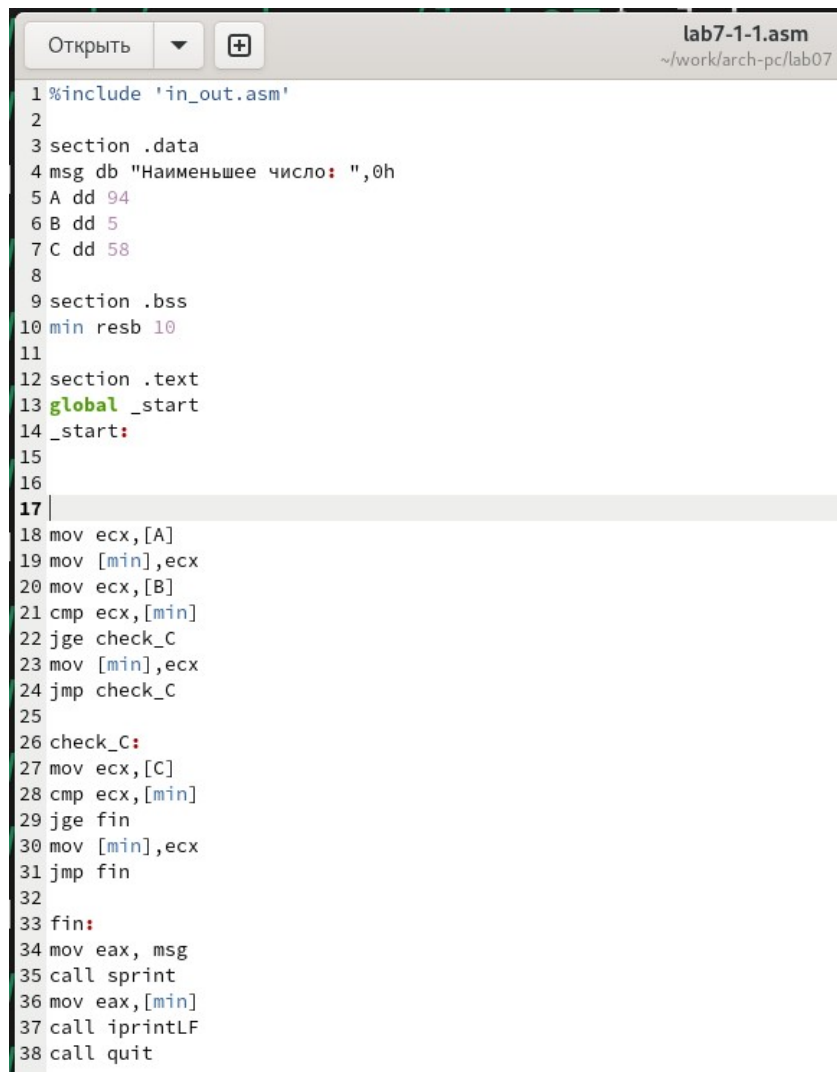
204 29 00000110 8900[00000000]      mov [max],ecx ; 'max' = A'
205 30 ; ----- Сравниваем 'A' и 'C' (как символы)
206 31      cmp ecx,'A'-'C' ; Removed ,C] !!!!!!!!!!!!!
207 31      error: invalid combination of opcode and operands
208 32 0000001C 7F0C      jg check_B ; если 'A'>'C', то переход на метку 'check_B',
209 32 0000001C 8800[30000000]      mov ecx,[C] ; иначе 'ecx' = 'C'

```

Рис. 2.14: Листинг, строка с ошибкой

### **3 Результаты выполнения заданий для самостоятельной работы**

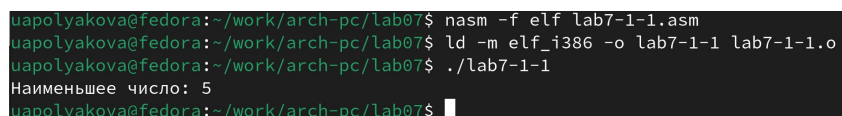
1. Создаем программу для нахождения наименьшего из трех чисел. Вариант 3, значит значения (94, 5, 58). Файл lab7-1-1.asm (Рис. 15).



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

Рис. 3.1: Листинг lab7-1-1.asm

2. Создаем исполняемый файл и запускаем его (Рис. 16).



```
uapolyakova@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-1-1.asm
uapolyakova@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1-1 lab7-1-1.o
uapolyakova@fedora:~/work/arch-pc/lab07$ ./lab7-1-1
Наименьшее число: 5
uapolyakova@fedora:~/work/arch-pc/lab07$
```

Рис. 3.2: Запуск lab7-1-1.asm

3. Создаем программу, которая вычисляет значение функции в зависимости от введенных x и a. Вариант 3, файл lab7-1-2.asm (Рис. 17).

Открыть

lab7-1-2.asm

~/work/arch-pc/lab07

```

1 %include 'in_out.asm'
2 SECTION .data
3 msg1: DB 'Введите x: ',0
4 msg2: DB 'Введите a: ',0
5 rem: DB 'Результат функции: ',0
6
7 SECTION .bss
8 x: RESB 80
9 a: RESB 80
10
11 SECTION .text
12 GLOBAL _start
13 _start:
14 ; ----- x
15 mov eax,msg1
16 call sprintLF
17
18 mov ecx,x
19 mov edx,80
20 call sread
21
22 mov eax,x
23 call atoi
24 mov [x],eax
25
26 ; ----- a
27 mov eax,msg2
28 call sprintLF
29
30 mov ecx,a
31 mov edx,80
32 call sread
33
34 mov eax,a
35 call atoi
36 mov [a],eax
37
38 mov eax,[x]
39 cmp eax,3
40 je fin1
41 jmp fin2
42
43 fin1:
44 mov ecx,3
45 mul ecx
46 mov edi,eax
47 mov eax,rem
48 call sprint
49 mov eax,edi
50 call iprintLF
51 call quit
52
53 fin2:
54 mov eax,[a]
55 add eax,1
56 mov edi,eax
57 mov eax,rem
58 call sprint
59 mov eax,edi
60 call iprintLF
61 call quit

```

Рис. 3.3: Листинг lab7-1-2.asm

4. Создаем исполняемый файл и запускаем его (рис. 18).

```
uapolyakova@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-1-2.asm
uapolyakova@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1-2 lab7-1-2.o
uapolyakova@fedora:~/work/arch-pc/lab07$ ./lab7-1-2
Введите x:
3
Введите a:
4
Результат функции: 9
uapolyakova@fedora:~/work/arch-pc/lab07$ ./lab7-1-2
Введите x:
1
Введите a:
4
Результат функции: 5
uapolyakova@fedora:~/work/arch-pc/lab07$
```

Рис. 3.4: Запуск lab7-1-2.asm



## 4 Вывод

Были изучены команды условного и безусловного переходов и приобретены навыки написания программ с использованием переходов. Были изучены назначение и структура файла листинга.