

# **Отчёт по лабораторной работе №7**

**Дисциплина: архитектура компьютера**

Анна Юрьевна Кижваткина

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>14</b>

# Список иллюстраций

3.1	Создание каталога . . . . .	7
3.2	Перемещение в каталог . . . . .	7
3.3	Создание файла . . . . .	7
3.4	Ввод программы . . . . .	7
3.5	Создание исполняемого файла . . . . .	8
3.6	Изменение программы из листинга . . . . .	8
3.7	Создание исполняемого файла и запуск . . . . .	8
3.8	Изменение программы . . . . .	9
3.9	Создание исполняемого файла и запуск . . . . .	9
3.10	Создание файла . . . . .	9
3.11	Ввод программы из листинга . . . . .	9
3.12	Создание исполняемого файла и запуск . . . . .	10
3.13	Создание листинга . . . . .	10
3.14	Открытие листинга . . . . .	10
3.15	Объяснение 1 строки . . . . .	10
3.16	Объяснение 2 строки . . . . .	10
3.17	Объяснение 3 строки . . . . .	10
3.18	Удаление операнда . . . . .	11
3.19	Трансляция файла . . . . .	11
3.20	Ошибка в листинге . . . . .	11
3.21	Создание файла . . . . .	11
3.22	Ввод программы . . . . .	12
3.23	Создание исполняемого файла и запуск . . . . .	12
3.24	Создание файла . . . . .	12
3.25	Ввод программы . . . . .	13
3.26	Создание исполняемого файла и запуск . . . . .	13

## **Список таблиц**

# 1 Цель работы

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

## 2 Задание

1. Реализация переходов в NASM.
2. Изучение структуры файлы листинга.
3. Выполнение самостоятельной работы.

### 3 Выполнение лабораторной работы

Создаем каталог для программ лабораторной работы №7. (рис. 3.1).

```
aykizhvatkina@dk8n77 ~ $ mkdir ~/work/arch-pc/lab07
```

Рис. 3.1: Создание каталога

Переходим в созданный каталог. (рис. 3.2).

```
aykizhvatkina@dk8n77 ~ $ cd ~/work/arch-pc/lab07
```

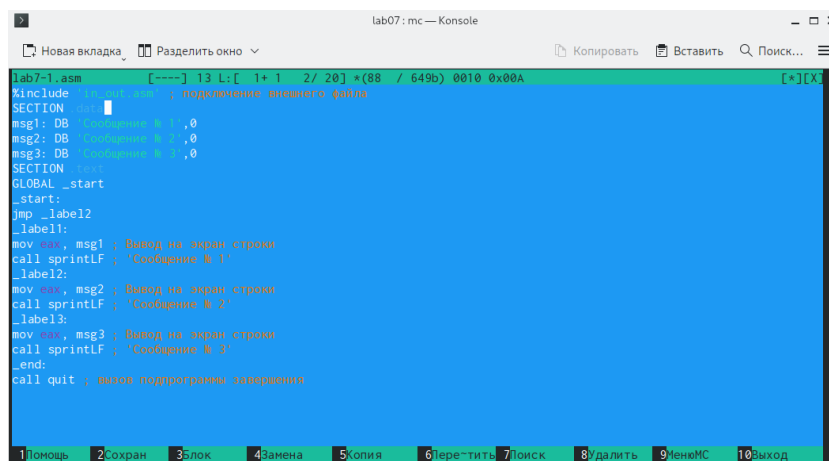
Рис. 3.2: Перемещение в каталог

Создаем файл lab7-1.asm. Проверяем наличие. (рис. 3.3).

```
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ touch lab7-1.asm
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ls
lab7-1.asm
```

Рис. 3.3: Создание файла

Вводим в файл программу из листинга 7.1. (рис. 3.4).



```
lab7-1.asm [-----] 13 L: [ 1+ 1 2/ 20] *(88 / 649b) 0010 0x00A [*][X]
%include "lab7-1.inc" ; 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 _label1
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; "Сообщение № 1"
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; "Сообщение № 2"
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; "Сообщение № 3"
_end:
call quit ; выход подпрограммы запаравления
```

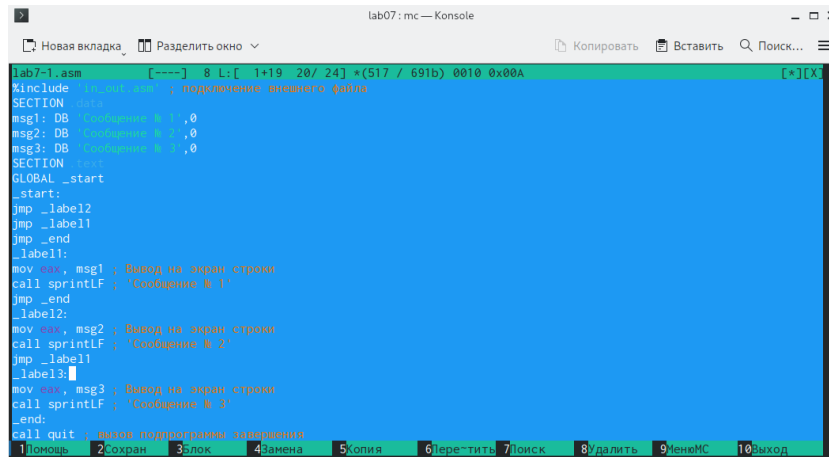
Рис. 3.4: Ввод программы

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

```
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 3
```

Рис. 3.5: Создание исполняемого файла

Вводим в файл программу из листинга 7.2 и меняем её так чтобы выводилось сначала 2 а потом 1 сообщение. (рис. 3.6).



```
lab7-1.asm [----] 8 L: [ 1+19 20/ 24] *(517 / 691b) 0010 0x00A [*][X]
#include "stdio.h" ; подключение стандартного файла.
SECTION .text
msg1: DB "Сообщение № 1",0
msg2: DB "Сообщение № 2",0
msg3: DB "Сообщение № 3",0
SECTION .text
GLOBAL _start
_start:
jmp _label2
jmp _label1
jmp _end
_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
1Помощь 2Сохран 3Блок 4Вставка 5Копия 6Переместить 7Поиск 8Удалить 9Меню 10Выход
```

Рис. 3.6: Изменение программы из листинга

Создаем исполняемый файл и запускаем его. (рис. 3.7).

```
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 1
```

Рис. 3.7: Создание исполняемого файла и запуск

Меняем программу так чтобы выводилось сначала 3, потом 2, а потом 1 сообщение. (рис. 3.8).



```

lab7-1.asm [-----] 41 L: [ 1+22 23/ 23] *(682 / 682b) <EOF> [*][X]
#include "lab7-1.inc" ; подключение внешнего файла
SECTION .text
msg1: DB "Сообщение № 1", 0
msg2: DB "Сообщение № 2", 0
msg3: DB "Сообщение № 3", 0
SECTION .text
GLOBAL _start
_start:
jmp _label1
_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"
jmp _label2
_end:
call quit ; выход подпрограммы завершили
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пере-тиг 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 3.8: Изменение программы

Создаем исполняемый файл и запускаем его. (рис. 3.9).

```

aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1

```

Рис. 3.9: Создание исполняемого файла и запуск

Создаем файл lab7-2.asm. Проверяем наличие. (рис. 3.10).

```

aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ touch lab7-2.asm
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ls
in_out.asm lab7-1 lab7-1.asm lab7-1.o lab7-2.asm

```

Рис. 3.10: Создание файла

Вводим в файл программу из листинга 7.3. (рис. 3.11).

```

lab7-2.asm [-----] 9 L: [ 1+ 8 9/ 49] *(169 /1743b) 0010 0x00A [*][X]
#include "lab7-2.inc"
section .text
msg1: db "Введите B: ", 0h
msg2: db "Наибольшее число: ", 0h
A: dd 0h
C: dd 0h
section .text
max: resb 10
B: resb 10
section .text
global _start
_start:
; ----- Вывод сообщения "Введите B: "
mov eax, msg1
call sprintf
; ----- Ввод 'B'
mov ecx, B
mov edx, 10
call sread
; ----- Преобразование 'B' из символа в число
mov eax, B
call atoi ; Выход подпрограммы перевода символа в число
mov [B], eax ; запись преобразованного числа в 'B'
; ----- Символы 'A' в переменную 'max'
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пере-тиг 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 3.11: Ввод программы из листинга

Создаем исполняемый файл и проверяем его работу для разных значений. (рис. 3.12).

```
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ./lab7-2
Введите B: 5
Наибольшее число: 50
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ./lab7-2
Введите B: 60
Наибольшее число: 60
```

Рис. 3.12: Создание исполняемого файла и запуск

Создаем файл листинга. (рис. 3.13).

```
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
```

Рис. 3.13: Создание листинга

Открываем файл листинга с помощью mcedit. 1(рис. 3.14).

```
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ mcedit lab7-2.lst
```

Рис. 3.14: Открытие листинга

Данная строка присваивает еах значение max. (рис. 3.15).

```
38 0000013F 8B0D[00000000]      mov ecx,[max]
```

Рис. 3.15: Объяснение 1 строки

Данная строка выводит текст и переходит на другую строку. (рис. 3.16).

```
48 00000168 E819FFFFFF      call iprintLF
```

Рис. 3.16: Объяснение 2 строки

Данная строка выводит ответ. (рис. 3.17).

```
46 0000015E E8ACFEFFFF      call sprint
```

Рис. 3.17: Объяснение 3 строки

Открываем файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалить один операнд. (рис. 3.18).

```

lab7-2.asm [----] 8 L: [ 1+ 0 1/ 49] *(8 /1740b) 0032 0x020 [X]
#include "lab7-1.asm"
section ".data"
msg1 db "Введите B: ",0h
msg2 db "Наибольшее число: ",0h
A dd 0
C dd 0
section ".text"
max resb 10
B resb 10
section ".text"
global _start
_start:
; Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; Ввод 'B'
mov ecx,B
mov edx,10
call read
; Преобразование 'B' на символ в число
mov eax,B
call atoi ; Вывод подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в B

```

Рис. 3.18: Удаление операнда

Выполняем трансляцию с получением файла листинга. (рис. 3.19).

```

aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:34: error: invalid combination of opcode and operands

```

Рис. 3.19: Трансляция файла

Мы получаем листинг с ошибкой. (рис. 3.20).

```

aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ touch lab7-3.asm
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ls
in_out.asm lab7-1 lab7-1.asm lab7-1.o lab7-2 lab7-2.asm lab7-2.lst lab7-3.asm

```

Рис. 3.20: Ошибка в листинге

Создаем файл lab7-3.asm. (рис. 3.21).

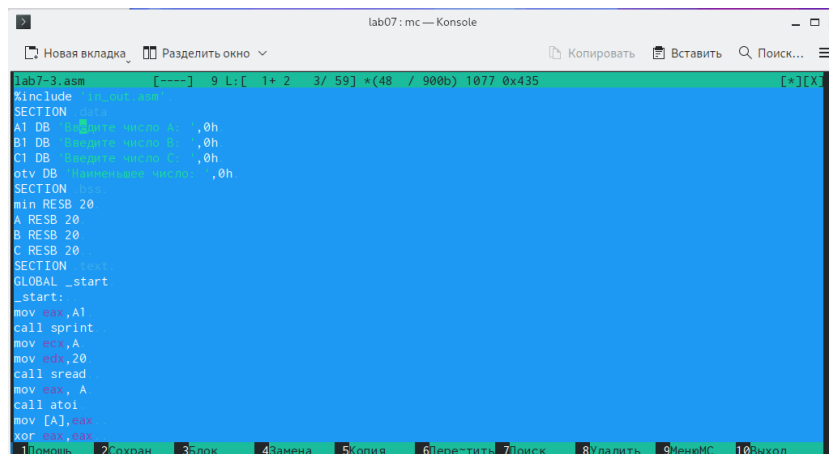
```

aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ touch lab7-3.asm
aykizhvatkina@dk8n77 ~/work/arch-pc/lab07 $ ls
in_out.asm lab7-1 lab7-1.asm lab7-1.o lab7-2 lab7-2.asm lab7-2.lst lab7-3.asm

```

Рис. 3.21: Создание файла

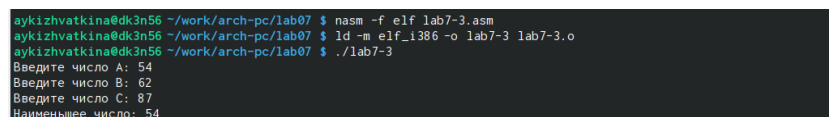
Вводим программу для вычисления наименьшей целочисленной переменной. Мой вариант 5, поэтому мои числа 54, 62 и 87. (рис. 3.22).



```
lab7-3.asm [----] 9 L: [ 1+ 2 3/ 59] *(48 / 900b) 1077 0x435 [*][X]
xinclude "lab7-1.asm"
SECTION .text
A1 DB "Введите число A: ",0h
B1 DB "Введите число B: ",0h
C1 DB "Введите число C: ",0h
otv DB "Наименьшее число: ",0h
SECTION .data
min RESB 20
A RESB 20
B RESB 20
C RESB 20
SECTION .text
GLOBAL _start
_start:
mov edi,A1
call sprint
mov edi,A
mov ebx,20
call read
mov eax, A
call atoi
mov [A],eax
xor eax,eax
```

Рис. 3.22: Ввод программы

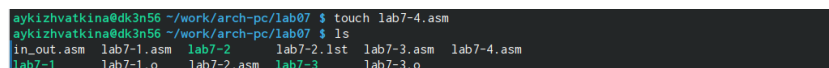
Создаем исполняемый файл и запускаем его. (рис. 3.23).



```
aykizhvatkina@dk3n56 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
aykizhvatkina@dk3n56 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
aykizhvatkina@dk3n56 ~/work/arch-pc/lab07 $ ./lab7-3
Введите число A: 54
Введите число B: 62
Введите число C: 87
Наименьшее число: 54
```

Рис. 3.23: Создание исполняемого файла и запуск

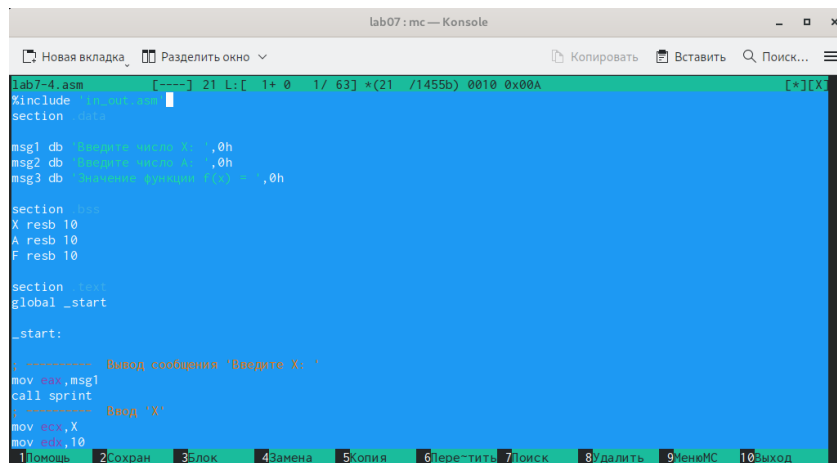
Создаем файл lab7-4.asm. (рис. 3.24).



```
aykizhvatkina@dk3n56 ~/work/arch-pc/lab07 $ touch lab7-4.asm
aykizhvatkina@dk3n56 ~/work/arch-pc/lab07 $ ls
in_out.asm  lab7-1.asm  lab7-2      lab7-2.lst  lab7-3.asm  lab7-4.asm
lab7-1.o    lab7-1.o      lab7-2.asm  lab7-3     lab7-3.o
```

Рис. 3.24: Создание файла

Вводим программу для вычисления значения заданной функции. Мой вариант 5. (рис. 3.25).



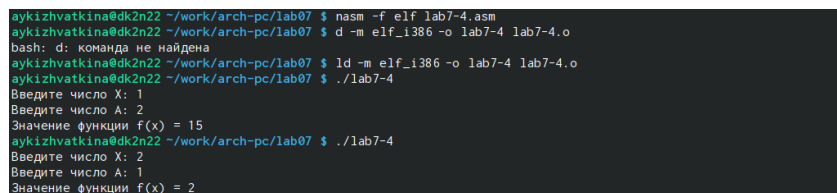
```
lab7-4.asm [----] 21 L: [ 1+ 0 1/ 63] *(21 /1455b) 0010 0x00A [*][X]
%include "lab7-4.inc"
section .data
msg1 db "Введите число X: ",0h
msg2 db "Введите число A: ",0h
msg3 db "Значение функции f(x) = ",0h

section .bss
X resb 10
A resb 10
F resb 10

section .text
global _start
_start:
;----- Вывод сообщения 'Введите X: '
mov rax,msg1
call sprint
;----- Ввод 'X'
mov rdx,X
mov rdi,10
```

Рис. 3.25: Ввод программы

Создаем исполняемый файл и запускаем его. (рис. 3.26).



```
aykizhvatkina@dk2n22 ~/work/arch-pc/lab07 $ nasm -f elf lab7-4.asm
aykizhvatkina@dk2n22 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
bash: d: команда не найдена
aykizhvatkina@dk2n22 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
aykizhvatkina@dk2n22 ~/work/arch-pc/lab07 $ ./lab7-4
Введите число X: 1
Введите число A: 2
Значение функции f(x) = 15
aykizhvatkina@dk2n22 ~/work/arch-pc/lab07 $ ./lab7-4
Введите число X: 2
Введите число A: 1
Значение функции f(x) = 2
```

Рис. 3.26: Создание исполняемого файла и запуск

## **4 Выводы**

С помощью данной лабораторной работы мы изучили команды условного и безусловного переходов. Приобрели навыки написания программ с использованием переходов. Ознакомились с назначением и структурой файла листинга.