

# **Отчет по лабораторной работе №7**

**Дисциплина: архитектура компьютера**

Абронина Алиса Кирилловна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>12</b>
	<b>Список литературы</b>	<b>13</b>

## Список иллюстраций

3.1	Сохранение программы . . . . .	7
3.2	Проверка файла листинга . . . . .	10
3.3	Вторая вывод задания для самостоятельной работы . . . . .	11

## **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 Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов: • условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия. • безусловный переход – выполнение передачи управления в определенную точку программы без каких-либо условий.

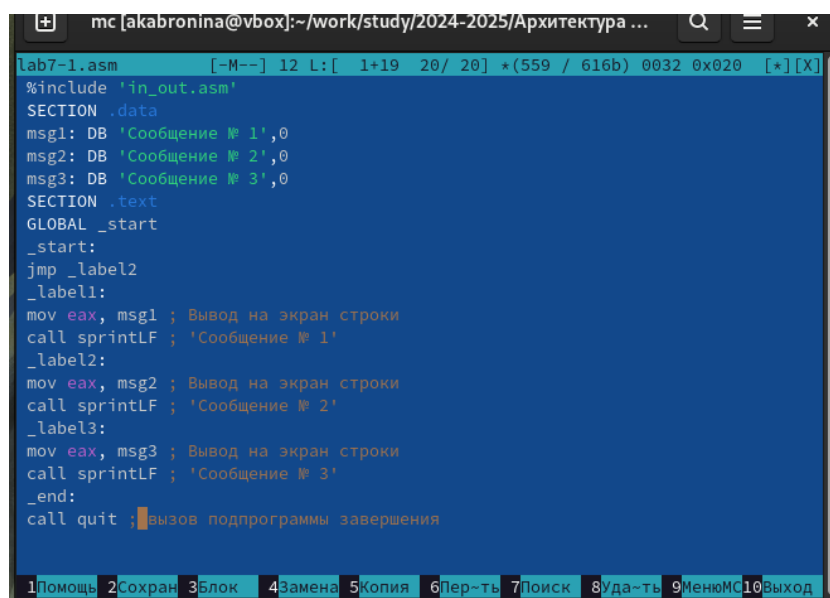
### 3 Выполнение лабораторной работы

Создаю каталог для программ лабораторной работы №7 (рис. -fig. ??).

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc$ mkdir lab07
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc$ ls
CHANGELOG.md  lab07      Makefile    README.en.md  template
config        labs       prepare     README.git-flow.md
COURSE        LICENSE    presentation README.md
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc$ cd lab07
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ touch lab7-1.asm
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ mc
```

Копирую код из листин-

га в файл будущей программы. (рис. -fig. 3.1).



```
lab7-1.asm      [-M--] 12 L: [ 1+19  20/ 20] *(559 / 616b) 0032 0x020  [*][X]
%include 'in_out.asm'
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'
_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 ; вызов подпрограммы завершения
```

Рис. 3.1: Сохранение программы

При запуске программы я убедилась в том, что безусловный переход действительно изменяет порядок выполнения инструкций (рис. -fig. ??).

```

akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ nasm -f
elf lab7-1.asm
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ld -m el
f_i386 -o lab7-1 lab7-1.o
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ mc

```

Изменяю программу

таким образом, чтобы поменялся порядок выполнения функций (рис. -fig. ??).

```

lab7-1.asm [-M--] 0 L: [ 1+ 0 1/ 27] *(0 / 795b) 0032 0x020 [*][X]
#include 'in_out.asm'
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 %include 'in_out.asm'

```

Запускаю программу

и проверяю, что примененные изменения верны (рис. -fig. ??).

```

akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ nasm -f
elf lab7-1.asm
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ld -m el
f_i386 -o lab7-1 lab7-1.o
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ mc

```

Теперь изменяю

текст программы так, чтобы все три сообщения вывелись в обратном порядке (рис. -fig. ??).

```

lab7-1.asm [-M--] 11 L: [ 3+18 21/ 25] *(574 / 675b) 0010 0x00A [*]
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label3
_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 ; вызов подпрограммы завершения %include 'in_out.asm'

```

Работа выполнена



корректно, программа в нужном мне порядке выводит сообщения (рис. -fig. ??).

```
akabronina@vbox: ~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ nasm -f elf lab7-1.asm
akabronina@vbox: ~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
akabronina@vbox: ~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
akabronina@vbox: ~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$
```

Создаю новый рабо-

чий файл и вставляю в него код из следующего листинга (рис. -fig. ??).

```
lab7-2.asm [-M--] 12 L:[ 1+ 6 7/ 55] *(153 /1790b) 0115 0x073 [*][X]
#include 'in_out.asm'
section .data
msg1 db 'Введите B: ',0h
msg2 db "Наибольшее число: ",0h
A dd '20'
C dd '50'
section .bss
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 sread
;----- Преобразование 'B' из символа в число
mov eax,B
call atoi
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC 10Выход
```

Программа выводит

значение переменной с максимальным значением (рис. -fig. ??).

```
akabronina@vbox: ~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ nasm -f
elf lab7-2.asm
akabronina@vbox: ~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ld -m el
f_i386 -o lab7-2 lab7-2.o
akabronina@vbox: ~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ./lab7-2
Введите B: 3
Наибольшее число: 50
akabronina@vbox: ~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ nasm -f
elf -l lab7-2 -lst lab7-2.asm
```

Создаю файл листин-

га с помощью флага -l команды nasm (рис. -fig. 3.2).

```

lab7-2.lst  [----]  0 L: [ 1+ 2 3/225] *(148 /14503b) 0032 0x020 [*][X]
1                                     %include 'in_out.asm'
2                                     <1> ;----- slen -----
3                                     <1> ; Функция вычисления длины сообщения
4                                     <1> slen:.....
5                                     <1> push ebx.....
6                                     <1> mov ebx, eax.....
7                                     <1>.....
8                                     <1> nextchar:.....
9                                     <1> cmp byte [eax], 0...
10                                    <1> jz finished.....
11                                    <1> inc eax.....
12                                    <1> jmp nextchar.....
13                                    <1>.....
14                                    <1> finished:
15                                    <1> sub eax, ebx
16                                    <1> pop ebx.....
17                                    <1> ret.....
18                                    <1>.....
19                                    <1> ;----- sprint -----
20                                    <1> ; Функция печати сообщения
21                                    <1> ; входные данные: mov eax, <message>

```

Рис. 3.2: Проверка файла листинга

Удаляю один операнд из случайной инструкции, чтобы проверить поведение файла листинга в дальнейшем .

В новом файле листинга показывает ошибку, которая возникла при попытке трансляции файла. Никакие выходные файлы при этом помимо файла листинга не создаются. (рис. -fig. ??).

```

lab7-2.lst  [B---]  78 L: [205+16 221/231] *(14139/14731b) 0097 0x061[*][X]
30 00000122 7F0C          jg check_B ; если 'A>C', то переход на м
31 00000124 8B0D[39000000] mov ecx,[C] ; иначе 'ecx = C'
32 0000012A 890D[00000000] mov [max],ecx ; 'max = C'
33                                     ;----- Преобразование 'max(A,C)' из
34                                     check_B:
35 00000130 B8[00000000]      mov eax,max
36 00000135 E862FFFFFF      call atoi
37                                     ; Вызов подпрограммы перевода символа в
38 0000013A 890D[00000000]      mov [max],ecx ; запись преобразованного
39                                     ;-----Сравниваем 'max(A,C)' и 'B' (
40 00000140 8B0D[00000000]      mov ecx,[max]
41 00000146 3B0D[0A000000]      cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
42 0000014C 7F06          jg fin
43                                     ; если 'max(A,C)>B', то переход на 'fin'
44 0000014E 8B0D[0A000000]      mov ecx,[B] ; иначе 'ecx = B'
45                                     mov [max]
46                                     ***** error: invalid combination of opcode and operand
47                                     ;-----Вывод результата
48                                     fin:
49 00000154 B8[13000000]      mov eax, msg2
50 00000159 E8B1FFFFFF      call sprint
51 0000015E A1[00000000]      mov eax,[max]

```

## Задания для само-

стоятельной работы Буду использовать свой вариант - 18 - из предыдущей лабораторной работы. Проверяю корректность написания первой программы (рис.

-fig. ??).

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ nasm -f  
elf zadaniel.asm  
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ld -m el  
f_i386 -o zadaniel zadaniel.o  
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/lab07$ ./zadani  
e1  
Наименьшее число: 30
```

Пишу программу, ко-

торая будет вычислять значение заданной функции согласно моему варианту для введенных с клавиатуры переменных а и х (рис. -fig. 3.3).

```
akabronina@vbox:~/work/study/2024-2025/Арх  
akabronina@vbox:~/work/study/2024-2025/Арх  
ue2.o  
akabronina@vbox:~/work/study/2024-2025/Арх  
Введите значение х: 1  
Введите значение а: 2  
Результат вычислений: 4  
akabronina@vbox:~/work/study/2024-2025/Арх  
Введите значение х: 2  
Введите значение а: 1  
Результат вычислений: 12
```

Рис. 3.3: Вторая вывод задания для самостоятельной работы

## **4 Выводы**

При выполнении лабораторной работы я изучила команды условных и безусловных переходов, а также приобрела навыки написания программ с использованием переходов, познакомилась с назначением и структурой файлов листинга.

## **Список литературы**