

Отчет по Лабораторной работе №7

дисциплина: Архитектура компьютера

Комаров Владимир Артемович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Реализация переходов в NASM	8
4.2	Изучение структуры файлы листинга	11
5	Задания для самостоятельной работы	13
6	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	Переход в каталог и создание файла	8
4.2	Программа с использованием инструкции jmp	8
4.3	Исполнение программы из листинга 7.1	9
4.4	Исправленный текст программы lab7-1.asm	9
4.5	Исполнение программы lab7-1	9
4.6	Исправленный текст программы lab7-1.asm	10
4.7	Исполнение программы lab7-1	10
4.8	Программа из листинга 7.3	10
4.9	Исполнение программы из листинга 7.3	11
4.10	Содержимое файла листинга	11
4.11	Трансляция	12
5.1	Текст программы lab7-3.asm	13
5.2	Запуск программы	13
5.3	Текст программы lab7-4.asm	14
5.4	Запуск программы	14

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. Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и c . Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу.
2. Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений. Вид функции $f(x)$ выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений x и a из 7.6.

3 Теоретическое введение

4 Выполнение лабораторной работы

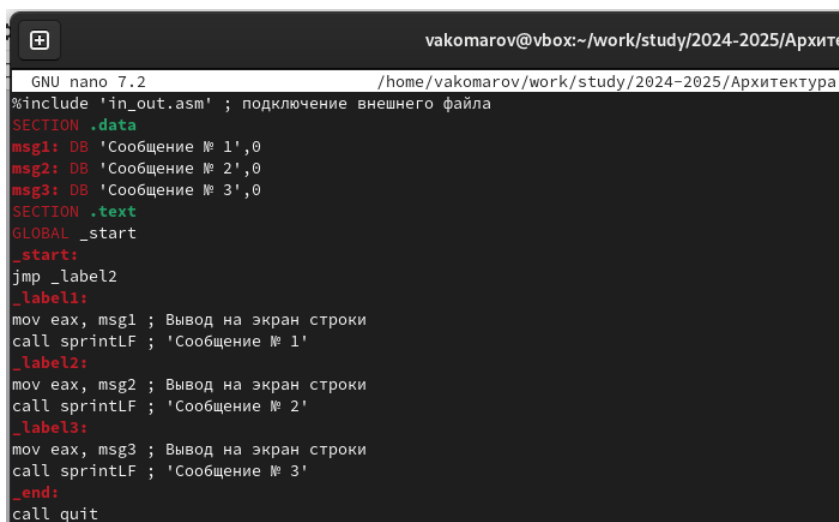
4.1 Реализация переходов в NASM

1. Создаю каталог для программ лабораторной работы №7, перехожу в него и создаю файл lab7-1.asm.

```
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ mkdir lab07
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd lab07
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ touch lab07-1.asm
```

Рис. 4.1: Переход в каталог и создание файла

2. Ввожу в файл lab7-1.asm текст программы из листинга 7.1. Запускаю исполняемый файл.



```
GNU nano 7.2 /home/vakomarov/work/study/2024-2025/Архитектура
#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'
_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
```

Рис. 4.2: Программа с использованием инструкции jmp


```

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

```

Рис. 4.3: Исполнение программы из листинга 7.1

- Изменим текст программы так, чтобы она сначала выводила “Сообщение №2”, потом “Сообщение №1” и завершала работу. Запустим исправленную программу.

```

GNU nano 7.2 /home/vakomarov/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07/1
#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

```

Рис. 4.4: Исправленный текст программы lab7-1.asm

```

vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab07-1.asm
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 -o lab07-1 lab07-1.o
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ./lab07-1
Сообщение № 2
Сообщение № 1

```

Рис. 4.5: Исполнение программы lab7-1

- Изменим текст программы так, чтобы она выводила сначала “Сообщение №1”, потом “Сообщение №2” и после этого “Сообщение №1” и завершала работу. Запустим исправленную программу

```

vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07
GNU nano 7.2 /home/vakomarov/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07/lab07
#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 _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

```

Рис. 4.6: Исправленный текст программы lab7-1.asm

```

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

```

Рис. 4.7: Исполнение программы lab7-1

- Создадим файл lab7-2.asm. Введем в файл текст программы из листинга 7.3. Программа определяет и выводит на экран наибольшую из целочисленных переменных A, B, C. Значения для A, C задаются в программе, значение вводится с клавиатуры. Запускаю исполняемый файл.

```

vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07
GNU nano 7.2 /home/vakomarov/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07/lab07
#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 ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ----- Записываем 'A' в переменную 'max'
mov ecx,[A] ; 'ecx = A'

```

Рис. 4.8: Программа из листинга 7.3

```

vakomarov@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab07-2.asm
vakomarov@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 -o lab07-2 lab07-2.o
vakomarov@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ./lab07-2
Введите B: 1000000
Наибольшее число: 1000000
vakomarov@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ./lab07-2
Введите B: 47
Наибольшее число: 50
vakomarov@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ./lab07-2
Введите B: 3
Наибольшее число: 50

```

Рис. 4.9: Исполнение программы из листинга 7.3

4.2 Изучение структуры файлы листинга

- Создаю файл листинга для программы из файла lab7-2.asm. Открываю файл листинга в любом текстовом редакторе.

```

lab07-2.lst  [----] 85 L: [ 1+19 20/225] *(1311/14458b) 0010 0x00A
1          %include 'in_out.asm'
2          <1> ;----- slen -----
3          <1> ; Функция вычисления длины сообщения
4          <1> slen:.....
5          4 00000000 53          <1> push    ebx.....
6          5 00000001 89C3       <1> mov     ebx, eax.....
7          6          <1>.....
8          7          <1> nextchar:.....
9          8 00000003 803800      <1> cmp     byte [eax], 0...
10         9 00000006 7403       <1> jz      finished.....
11        10 00000008 40         <1> inc     eax.....
12        11 00000009 EBF8       <1> jmp     nextchar.....
13        12          <1>.....
14        13          <1> finished:
15        14 0000000B 29D8       <1> sub     eax, ebx
16        15 0000000D 5B         <1> pop     ebx.....
17        16 0000000E C3         <1> ret.....
18        17          <1>.....
19        18          <1> ;----- sprint -----
20        19          <1> ; Функция печати сообщения
21        20          <1> ; входные данные: mov eax, <message>
22        21          <1> sprint:
23        22 0000000F 52         <1> push    edx
24        23 00000010 51         <1> push    ecx
25        24 00000011 53         <1> push    ebx
26        25 00000012 50         <1> push    eax

```

Рис. 4.10: Содержимое файла листинга

- Объясняю три строки из файла листинга: 23 00000106 E891FFFFFF call atoi - Вызов подпрограммы перевода символа в число; 23 - номер строки, 00000106 - адрес, E891FFFFFF - машинный код; 41 0000014B 7F0C jg fin - переход на label 'fin', если 'max(A,C)>B'; 41 - номер строки, 0000014B - адрес, 7F0C - машинный код; 50 0000016D E869FFFFFF call quit - Выход из программы; 50 - номер строки; 0000016D - адрес; E869FFFFFF - машинный код.
- Открываю файл с программой lab7-2.asm и в одной из инструкций с двумя

операндами удаляю один операнд. Транслирую файл с текстом программы с получением файла листинга. Я не получаю выходных файлов, программа выдает ошибку, так как в данной операции должны присутствовать два операнда, а не один.

```

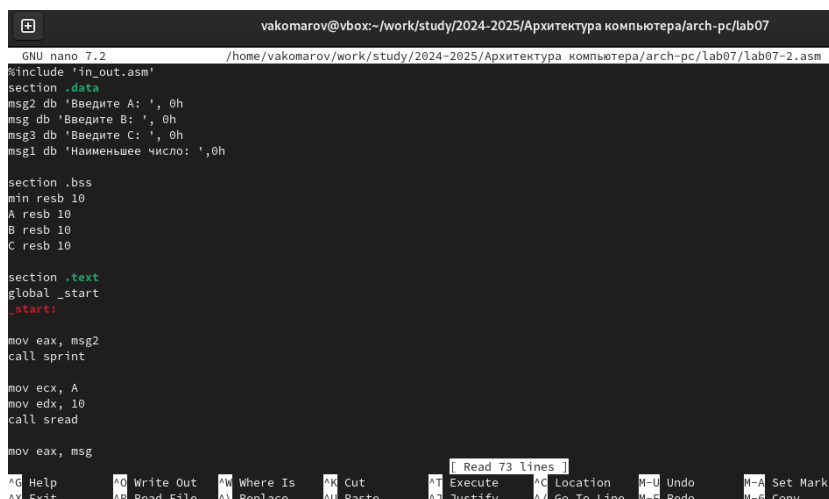
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07
lab07-2.lst [----] 0 L: [189+12 201/226] *(12461/14546b) 0032 0x020
14 000000E8 B8[00000000] mov eax,msg1
15 000000ED E81DFFFFFF call sprint
16 ; ----- Ввод 'B'
17 000000F2 B9[0A000000] mov ecx,B
18 000000F7 BA0A000000 mov edx,10
19 000000FC E842FFFFFF call sread
20 ; ----- Преобразование 'B' из символа в число
21 00000101 B8[0A000000] mov eax,B
22 00000106 E891FFFFFF call atoi ; Вызов подпрограммы перевода символа в число
23 0000010B A3[0A000000] mov [B],eax ; запись преобразованного числа в 'B'
24 ; ----- Записываем 'A' в переменную 'max'
25 00000110 8B0D[35000000] mov ecx,[A] ; 'ecx = A'
26 00000116 890D[00000000] mov [max],ecx ; 'max = A'
27 ; ----- Сравниваем 'A' и 'C' (как символы)
28 0000011C 3B0D[39000000] cmp ecx,[C] ; Сравниваем 'A' и 'C'
29 00000122 7F0C jg check_B ; если 'A>C', то переход на метку 'check_B',
30 00000124 8B0D[39000000] mov ecx,[C] ; иначе 'ecx = C'
31 0000012A 890D[00000000] mov [max],ecx ; 'max = C'
32 ; ----- Преобразование 'max(A,C)' из символа в число
33 check_B:
34 mov eax,
35 error: invalid combination of opcode and operands
36 call atoi ; Вызов подпрограммы перевода символа в число
37 mov [max],eax ; запись преобразованного числа в 'max'
38 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
39 0000013A 8B0D[0A000000] mov ecx,[max]
40 00000140 3B0D[0A000000] cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'

```

Рис. 4.11: Трансляция

5 Задания для самостоятельной работы

1. Создаю файл lab7-3.asm и ввожу в него текст программы для нахождения наименьшей из трех целочисленных переменных a, b, c. Мой вариант 10. Программа работает корректно.



```
GNU nano 7.2 /home/vakomarov/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07/lab07-2.asm
#include 'in_out.asm'
section .data
msg2 db 'Введите A: ', 0h
msg db 'Введите B: ', 0h
msg3 db 'Введите C: ', 0h
msg1 db 'Наименьшее число: ', 0h

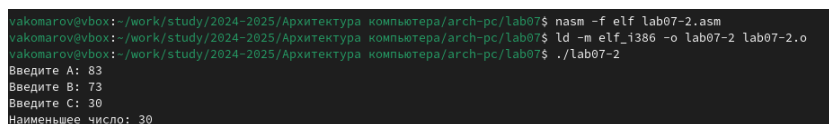
section .bss
min resb 10
A resb 10
B resb 10
C resb 10

section .text
global _start
_start:
mov eax, msg2
call sprint

mov ecx, A
mov edx, 10
call sread

mov eax, msg
```

Рис. 5.1: Текст программы lab7-3.asm

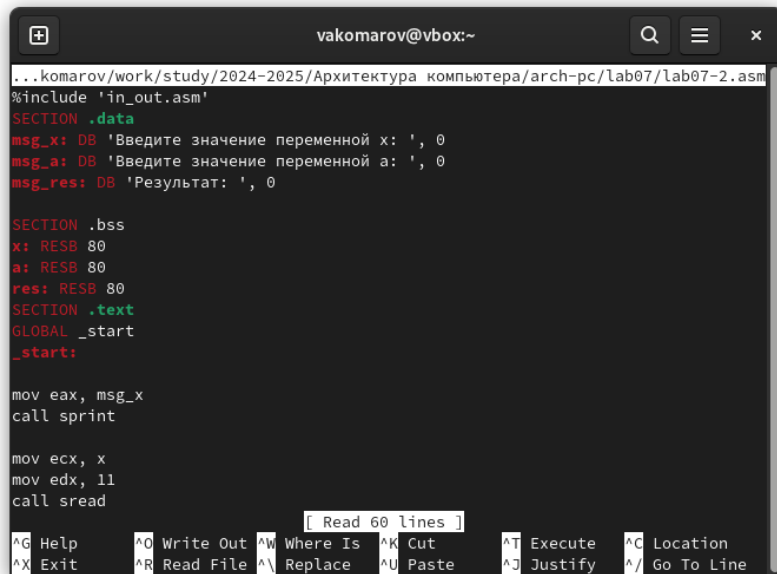


```
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab07-2.asm
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 -o lab07-2 lab07-2.o
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ./lab07-2
Введите A: 83
Введите B: 73
Введите C: 30
Наименьшее число: 30
```

Рис. 5.2: Запуск программы

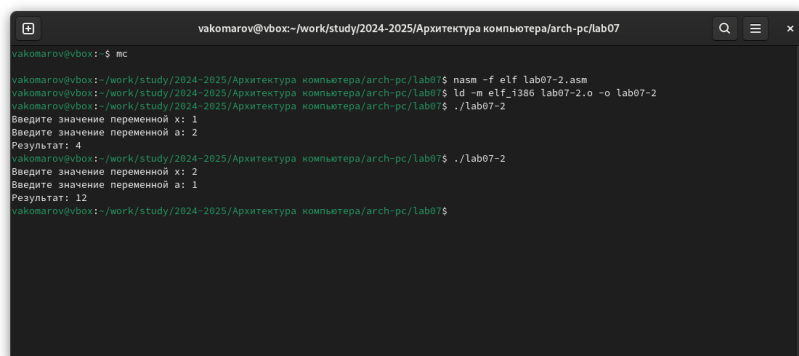
2. Создаю файл lab7-4.asm и ввожу в него текст программы, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции и

выводит результат вычислений. Мой вариант - 10.



```
vakomarov@vbox:~  
...komarov/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07/lab07-2.asm  
%include 'in_out.asm'  
SECTION .data  
msg_x: DB 'Введите значение переменной x: ', 0  
msg_a: DB 'Введите значение переменной a: ', 0  
msg_res: DB 'Результат: ', 0  
  
SECTION .bss  
x: RESB 80  
a: RESB 80  
res: RESB 80  
SECTION .text  
GLOBAL _start  
_start:  
  
mov eax, msg_x  
call sprint  
  
mov ecx, x  
mov edx, 11  
call sread  
  
[ Read 60 lines ]  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Рис. 5.3: Текст программы lab7-4.asm



```
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07  
vakomarov@vbox:~$ mc  
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab07-2.asm  
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 lab07-2.o -o lab07-2  
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ./lab07-2  
Введите значение переменной x: 1  
Введите значение переменной a: 2  
Результат: 4  
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$ ./lab07-2  
Введите значение переменной x: 2  
Введите значение переменной a: 1  
Результат: 12  
vakomarov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
```

Рис. 5.4: Запуск программы

6 Выводы

В результате выполнения лабораторной работы я изучил команды условного и безусловного переходов, а так же приобрёл навыки написания программ с использованием переходов. Познакомился с назначением и структурой файла листинга.

Список литературы