

# **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**

**дисциплина: Архитектура компьютера**

Холопов Илья Алексеевич

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12

## Список иллюстраций

2.1	Создание каталога и файла в нем lab7-1.asm . . . . .	6
2.2	Содержимое файла lab7-1.asm . . . . .	6
2.3	Создание и запуск исполняемого файла . . . . .	7
2.4	Результат работы измененной программы . . . . .	7
2.5	Результат работы программы, выводящей сообщения в обратном порядке . . . . .	7
2.6	Содержимое файла lab7-2.asm . . . . .	8
2.7	Результат программы, выводящей на экран наибольшее из трех значений . . . . .	8
2.8	Файл листинга lab7-2.asm . . . . .	9
2.9	Содержимое файла lab7-2.lst . . . . .	9
2.10	Код программы, выводящей наименьшее из трек чисел . . . . .	10
2.11	Создание и запуск исполняемого файла . . . . .	10
2.12	Содержимое файла task2.asm . . . . .	11
2.13	Создание и запуск исполняемого файла, вычисляющего выражение	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 Выполнение лабораторной работы

Создадим каталог для лабораторной работы № 7, перейдем в него и создадим файл lab7-1.asm (рис. 2.1).

```
ikholopov0323@ikholopov0323:~$ cd work/arch-pc/  
ikholopov0323@ikholopov0323:~/work/arch-pc$ mkdir lab07  
ikholopov0323@ikholopov0323:~/work/arch-pc$ cd lab07  
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ touch lab7-1.asm  
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$
```

Рис. 2.1: Создание каталога и файла в нем lab7-1.asm

Введем в файл lab7-1.asm текст программы с использованием инструкции jmp (рис. 2.2).

```
/home/ikholopov0323/work/arch-pc/lab07/lab7-1.asm 322/322 100%  
%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  
call sprintLF  
_label2:  
mov  
eax, msg2  
call sprintLF  
_label3:  
mov  
eax, msg3  
call sprintLF  
_end:  
call quit  
1Помощь 2Раз-рн 3Выход 4Нех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход
```

Рис. 2.2: Содержимое файла lab7-1.asm

Создадим исполняемый файл и запустим его (рис. 2.3).

```

ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$

```

Рис. 2.3: Создание и запуск исполняемого файла

Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу (рис. 2.4).

```

ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$

```

Рис. 2.4: Результат работы измененной программы

Снова изменим программу, чтобы она выводила сначала ‘Сообщение № 3’, потом ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу (рис. 2.5).

```

ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
ikholopov0323@ikholopov0323:~/work/arch-pc/lab07$

```

Рис. 2.5: Результат работы программы, выводящей сообщения в обратном порядке

Создадим файл lab7-2.asm и введем в него текст программы, выводящей на экран наибольшее из трех значений (рис. 2.6).

```

/home/ikhologov0323/work-arch-pc/lab07/lab7-2.asm 540/540 100%
%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:
mov eax,msg1
call sprint
mov ecx,B
mov edx,10
call sread
mov eax,B
call atoi
mov [B],eax
mov ecx,[A]
mov[max],ecx
cmp ecx,[C]
jg check_B
mov ecx,[C]
mov[max],ecx
check_B:
mov eax,max
call atoi
mov [max],eax
mov ecx,[max]
cmp ecx,[B]
jg fin
mov ecx,[B]
mov[max],ecx
fin:
mov eax, msg2
call sprint
mov eax,[max]
call iprintLF
call quit
1Помощь 2Раз-рн 3Выход 4Нех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход

```

Рис. 2.6: Содержимое файла lab7-2.asm

Создадим исполняемый файл и запустим его (2.7).

```

ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 3
Наибольшее число: 50
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 74
Наибольшее число: 74
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$

```

Рис. 2.7: Результат программы, выводящей на экран наибольшее из трех значений

Создадим файл листинга для программы из файла lab7-2.asm и откроем его. В 15 строке 0000000B - адрес, 29D8 - машинный код `sub eax,ebx`. В 16 строке 0000000D



- адрес, 5B - машинный код pop ebx. В 17 строке 0000000E - адрес, C3 - машинный код ret (2.8).

```

/home/ikhopolov0323/work-ch-pc/lab07/lab7-2.lst 2414/12934 18%
1                                     %include 'in_out.asm'
2                                     <1> ;----- slen -----
-----
3                                     <1> ; Функция вычисления длины сообщения
4                                     <1> slen:
5 00000000 53                         <1>      push    ebx
6 00000001 89C3                       <1>      mov     ebx, eax
7                                     <1>
8                                     <1> nextchar:
9 00000003 803800                     <1>      cmp     byte [eax], 0
10 00000006 7403                      <1>      jz      finished
11 00000008 40                        <1>      inc     eax
12 00000009 EBF8                      <1>      jmp     nextchar
13                                     <1>
14                                     <1> finished:
15 0000000B 29D8                      <1>      sub     eax, ebx
16 0000000D 5B                        <1>      pop     ebx
17 0000000E C3                       <1>      ret
18                                     <1>

```

Рис. 2.8: Файл листинга lab7-2.asm

Откроем файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалим один операнд. Выполним трансляцию с получением файла листинга на рис. 2.9 видно в 13 строке информацию об ошибке.

```

10                                     section .text
11                                     global _start
12                                     _start:
13                                     mov eax,
13                                     ***** error: invalid combination of opcode and operands
14                                     ; mov eax,msg1
15 000000E8 E822FFFFFF                 call sprint
16 000000ED B9[0A000000]               mov ecx,B
17 000000F2 BA0A000000                mov edx,10
18 000000F7 E847FFFFFF                 call sread
19 000000FC B8[0A000000]               mov eax,B

```

Рис. 2.9: Содержимое файла lab7-2.lst

Напишем программу нахождения наименьшей из 3 целочисленных переменных 54, 62 и 87 (рис. 2.10).

```

/home/ikhologov0323/wo-ch-pc/lab07/task1.asm 421/421 100%
%include 'in_out.asm'
section .data
msg2 db "Наибольшее число: ",0h
A dd '54'
B dd '62'
C dd '87'
section .bss
max resb 10
section .text
global _start
_start:
mov ecx,[A]
mov[max],ecx
cmp ecx,[C]
jg check_B
mov ecx,[C]
mov[max],ecx

check_B:
mov ecx,[max]
cmp ecx,[B]
jg fin
mov ecx,[B]
mov[max],ecx

fin:
mov eax,max
call atoi
mov [max],eax
mov eax,msg2
call sprint
mov eax,[max]
call iprintLF
call quit
1По-щ 2Раз-рн 3Выход 4Дек 5Пер-ти 6 7Поиск 8Иск-ый 9Формат 10Выход

```

Рис. 2.10: Код программы, выводящей наименьшее из трех чисел

Создадим и запустим исполняемый файл (рис. 2.11).

```

ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ nasm -f elf task1.asm
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ ld -m elf_i386 -o task1 task
1.o
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ ./task1
Наибольшее число: 87

```

Рис. 2.11: Создание и запуск исполняемого файла

Напишем программу, которая для введенных с клавиатуры значений  $x$  и  $a$  вычисляет значение заданной функции  $f(x) = 2x - a, x > a$  или  $f(x) = 15, x \leq a$ . и выводит результат вычислений (рис. 2.12).

```

/home/ikhologov0323/wo~ch-pc/lab07/task2.asm 558/558 100%
#include 'in_out.asm'
SECTION .data
msgx: DB 'Введите x: ',0
msga: DB 'Введите a: ',0
res: DB 'f(x) = ',0
SECTION .bss
x: RESB 80
a: RESB 80
y: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msga
call sprint
mov ecx, a
mov edx, 80
call sread
mov eax, a
call atoi
mov [a], eax
mov eax, msgx
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
mov [x], eax
cmp eax, [a]
jg greater
mov ebx, 15
mov [y], ebx
jmp fin
greater:
sub eax, [a]
mov ebx, 2
mul ebx
mov [y], eax
fin:
mov eax, res
call sprint
mov eax, [y]
call iprintLF
call quit
1По-щ 2Раз-рн 3Выход 4Тек 5Пер-ти 6 7Поиск 8Исх-ый 9Формат 10Выход

```

Рис. 2.12: Содержимое файла task2.asm

Создадим исполняемый файл и запустим его (рис. 2.13).

```

ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ nasm -f elf task2.asm
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ ld -m elf_i386 -o task2 task2.o
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ ./task2
Введите a: 1
Введите x: 2
f(x) = 2
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ 2
2: command not found
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$ ./task2
Введите a: 2
Введите x: 1
f(x) = 15
ikhologov0323@ikhologov0323:~/work/arch-pc/lab07$

```

Рис. 2.13: Создание и запуск исполняемого файла, вычисляющего выражение

## **3 Выводы**

В результате выполнения работы были изучены команды условного и безусловного переходов, приобретены навыки написания программ с использованием переходов.