

Отчёт по лабораторной работе №7

Кучерова Виктория Васильевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Задание для самостоятельной работы	13
4	Выводы	17
	Список литературы	18

Список иллюстраций

2.1	Создание файла	6
2.2	Программа	7
2.3	Запуск	7
2.4	Программа2	8
2.5	Запуск2	8
2.6	Программа3	9
2.7	Запуск3	9
2.8	Создание файла	10
2.9	Программа4	10
2.10	Запуск4	11
2.11	Создание листинга	11
2.12	Строки 14-16	11
2.13	Удаление	11
2.14	Ошибка	12
2.15	Ошибка	12
3.1	Программа5	14
3.2	Запуск5	14
3.3	Программа6	15
3.4	Запуск6	16

Список таблиц

1 Цель работы

Изучить команды условного и безусловного переходов. Приобрести навыки написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Выполнение лабораторной работы

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

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

Рис. 2.1: Создание файла

Рассмотрим пример программы с использованием инструкции jmp(рис. 2.2), (рис. 2.3).

```

%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

```

Рис. 2.2: Программа

```

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

```

Рис. 2.3: Запуск

Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу(рис. 2.4), (рис. 2.5).

```

%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
    jmp _end

_label2:
    mov eax, msg2
    call sprintLF
    jmp _label1

_label3:
    mov eax, msg3
    call sprintLF

_end:
    call quit

```

Рис. 2.4: Программа2

```

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

```

Рис. 2.5: Запуск2

Изменим текст программы еще раз(рис. 2.6), (рис. 2.7).


```

%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 _label3

_label1:
    mov eax, msg1
    call sprintf
    jmp _end

_label2:
    mov eax, msg2
    call sprintf
    jmp _label1

_label3:
    mov eax, msg3
    call sprintf
    jmp _label2

_end:
    call quit

```

Рис. 2.6: Программа3

```

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

```

Рис. 2.7: Запуск3

Создадим файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и введем в него текст программы. Создадим исполняемый файл и проверим его работу для разных

значений B(рис. 2.8), (рис. 2.9), (рис. 2.10).

```
vvkucheroval@vbox:~/work/arch-pc/lab07$ touch lab7-2.asm
```

Рис. 2.8: Создание файла

```
%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
```

Рис. 2.9: Программа4

```

vvkucheroва@vbox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
vvkucheroва@vbox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
vvkucheroва@vbox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 12
Наибольшее число: 50
vvkucheroва@vbox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 30
Наибольшее число: 50
vvkucheroва@vbox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 51
Наибольшее число: 51

```

Рис. 2.10: Запуск4

Создадим файл листинга для программы из файла lab7-2.asm. Откроем файл листинга lab7-2.lst с помощью любого текстового редактора(рис. 2.11).

```

vvkucheroва@vbox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
vvkucheroва@vbox:~/work/arch-pc/lab07$ mcedit lab7-2.lst

```

Рис. 2.11: Создание листинга

Строка 14 имеет адрес '0000000B' и машинный код 29D8 и выполняет вычитание значения ebx из значения eax. Строка 15 имеет адрес '0000000D' и машинный код 5B и сохраняет значение в регистре ebx. Строка 16 имеет адрес '0000000E' и машинный код C3, используется для возврата из подпрограммы(рис. 2.12).

14	0000000B	29D8	<1>	sub	eax, ebx
15	0000000D	5B	<1>	mov	ebx
16	0000000E	C3	<1>	ret	

Рис. 2.12: Строки 14-16

Откроем файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалим один операнд. Программа и листинг выводят ошибку(рис. 2.13), (рис. 2.14), (рис. 2.15).

```

mov ecx|
mov edx,10
call sread

```

Рис. 2.13: Удаление

```
vykucheroval@vbox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:17: error: invalid combination of opcode and operands
```

Рис. 2.14: Ошибка

```
17                                     mov ecx
17          *****                  error: invalid combination of opcode and operands
18 000000f2 BA0A000000               mov edx,10
19 000000f7 F847FFFFFF               call sread
```

Рис. 2.15: Ошибка

3 Задание для самостоятельной работы

Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и c (рис. 3.1), (рис. 3.2).

```

%include 'in_out.asm'
section .data
    msg2 db "Наименьшее число: ",0h
    A dd '46'
    B dd '32'
    C dd '74'
section .bss
    max resb 10
section .text
    global _start
_start:

    mov ecx,[A]
    mov [max],ecx

    cmp ecx,[C]
    jl check_B
    mov ecx,[C]
    mov [max],ecx

check_B:

    mov ecx,[max]
    cmp ecx,[B]
    jl 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

```

Рис. 3.1: Программа5

```

vvkucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab0
7$ nasm -f elf lab7-2.asm
vvkucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab0
7$ ld -m elf_i386 -o lab7-2 lab7-2.o
vvkucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab0
7$ ./lab7-2
Наименьшее число: 32

```

Рис. 3.2: Запуск5

Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений(рис. 3.3), (рис. 3.4).

```

%include 'in_out.asm'

SECTION .data
msg1: DB 'Введите a: ',0
msg2: DB 'Введите x: ',0
rem: DB 'Результат: ',0

SECTION .bss
x: RESB 80
a: RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax,msg1
    call printf

    mov ecx, a
    mov edx, 80
    call read

    mov eax,a
    call atoi
    mov [a],eax

    mov eax, msg2
    call printf

    mov ecx, x
    mov edx, 80
    call read

    mov eax,x
    call atoi
    mov [x],eax

```

Рис. 3.3: Программа6

```
vykucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-3.asm
vykucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
vykucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-3
Введите а:
5
Введите х:
4
Результат: 4
vykucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-3
Введите а:
2
Введите х:
3
Результат: 5
```

Рис. 3.4: Запуск6

4 Выводы

Я изучила команды условного и безусловного переходов. Приобрела навыки написания программ с использованием переходов. Познакомилась с назначением и структурой файла листинга.

Список литературы