

# Отчет по выполнению лабораторной работы №7

## Дисциплина: архитектура компьютеров

Новиков Никита Владимирович

### Содержание

#### 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. Реализация переходов в NASM
2. Изучение структуры файла листинга

#### 3 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. Листинг (в рамках понятийного аппарата NASM) — это один из выходных файлов, создаваемых транслятором. Он имеет текстовый вид и нужен при отладке программы, так как кроме строк самой программы он содержит дополнительную информацию.

#### 4 Выполнение лабораторной работы

##Реализация переходов в NASM

Открываю терминал и создаю каталог для программ лабораторной работы, перехожу в него и создаю файл lab7-1.asm (рис. [??]).

```

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ touch lab7-1.asm
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ls
Makefile  bib  image  lab7-1.asm  pandoc  report.md

```

## Создание файла

Ввожу в файл lab7-1.asm текст программы из листинга 7.1.(рис. [??])

```

~\work\study\2023-2024\Архитектура компьютера\arch-pc\labs\lab07\report\lab7-1.asm - Mousepad
File Edit Search View Document Help
1 %include "in_out.asm" ; подключение внешнего файла
2 SECTION .data
3 msg1 DB 'Сообщение № 1',0
4 msg2 DB 'Сообщение № 2',0
5 msg3 DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label2
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 _label2:
14 mov eax, msg2 ; Вывод на экран строки
15 call sprintf ; 'Сообщение № 2'
16 _label3:
17 mov eax, msg3 ; Вывод на экран строки
18 call sprintf ; 'Сообщение № 3'
19 _end:
20 call quit ; вызов подпрограммы завершения

```

## Ввод текста из листинга 7.1

Создаю исполняемый файл и запускаю его.(рис. [??])

```

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ nasm -f elf lab7-1.asm
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ld -m elf_i386 -o lab7-1 lab7-1.o
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ./lab7-1
Сообщение № 2
Сообщение № 3

```

## Запуск файла

Изменяю текст программы в соответствии с листингом 7.2.(рис. [??])

```
File Edit Search View Document Help
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label2
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 jmp _end
14 _label2:
15 mov eax, msg2 ; Вывод на экран строки
16 call sprintf ; 'Сообщение № 2'
17 jmp _label1
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение № 3'
21 _end:
22 call quit ; вызов подпрограммы завершения
```

## Изменение файла

Создаю исполняемый файл и запускаю его.(рис. [??])

```
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ nasm -f elf lab7-1.asm

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ld -m elf_i386 -o lab7-1 lab7-1.o

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ./lab7-1
Сообщение № 2
Сообщение № 1
```

## Запуск измененного файла

Далее изменяю файл так, чтобы он выводил “Сообщения” в обратном порядке.(рис. [??])

```
File Edit Search View Document Help
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label3
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 jmp _end
14 _label2:
15 mov eax, msg2 ; Вывод на экран строки
16 call sprintf ; 'Сообщение № 2'
17 jmp _label1
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение № 3'
21 jmp _label2
22 _end:
23 call quit ; вызов подпрограммы завершения
```

## Редактирование файла

Создаю исполняемый файл и запускаю его.(рис. [??])

```

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ nasm -f elf lab7-1.asm

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ld -m elf_i386 -o lab7-1 lab7-1.o

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1

```

## Запуск программы

Убеждаюсь, что всё работает верно.

Создаю файл lab7-2.asm и вставляю в него текст программы из листинга 7.3.  
(рис. [??])

```

~\work\study\2023-2024\Архитектура компьютера\arch-pc\labs\lab07\report\lab7-2.asm - Mousepad
File Edit Search View Document Help
1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите B: ',0h
4 msg2 db 'Наибольшее число: ',0h
5 A dd '20'
6 C dd '50'
7 section .bss
8 max resb 10
9 B resb 10
10 section .text
11 global _start
12 _start:
13 ; ----- Вывод сообщения 'Введите B: '
14 mov eax,msg1
15 call sprint
16 ; ----- Ввод 'B'
17 mov ecx,B
18 mov edx,10
19 call sread
20 ; ----- Преобразование 'B' из символа в число
21 mov eax,B
22 call atoi ; Вызов подпрограммы перевода символа в число
23 mov [B],eax ; запись преобразованного числа в 'B'
24 ; ----- Записываем 'A' в переменную 'max'
25 mov ecx,[A] ; 'ecx = A'
26 mov [max],ecx ; 'max = A'
27 ; ----- Сравниваем 'A' и 'C' (как символы)
28 cmp ecx,[C] ; Сравниваем 'A' и 'C'
29 jg check_B ; если 'A>C', то переход на метку 'check_B',
30 mov ecx,[C] ; иначе 'ecx = C'
31 mov [max],ecx ; 'max = C'
32 ; ----- Преобразование 'max(A,C)' из символа в число
33 check_B:
34 mov eax,max
35 call atoi ; Вызов подпрограммы перевода символа в число
36 mov [max],eax ; запись преобразованного числа в 'max'
37 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)

```

## Ввод текста из листинга 7.3

Создаю исполняемый файл и запускаю его.(рис. [??])

```

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ nasm -f elf lab7-2.asm
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ld -m elf_i386 -o lab7-2 lab7-2.asm
lab7-2.asm: file not recognized: file format not recognized

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ld -m elf_i386 -o lab7-2 lab7-2.o

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ ./lab7-2
Введите B: 3
Наибольшее число: 50

```

Запуск программы

## 4.1 Изуение структуры файла листинга

Создаю файл листинга для программы из файла lab7-2.asm.(рис. [??])

```

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report]
$ nasm -f elf -l lab7-2.lst lab7-2.asm

```

Создание файла листинга

Далее открываю файл листинга lab7-2.lst.(рис. [??])

```

File Edit Search View Document Help
~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab07/report/lab7-2.lst - Mousepad

1 | 1 | %include 'in_out.asm'
2 | 1 | <1> ; _____ slen _____
3 | 2 | <1> ; Функция вычисления длины сообщения
4 | 3 | <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>
20 | 19 | | <1>
21 | 20 | | <1> ; _____ sprint _____
22 | 21 | | <1> ; Функция печати сообщения
23 | 22 | | <1> ; входные данные: mov eax, <message>
24 | 23 | 0000000F 52 | <1> push edx
25 | 24 | 00000010 51 | <1> push ecx
26 | 25 | 00000011 53 | <1> push ebx
27 | 26 | 00000012 50 | <1> push eax
28 | 27 | 00000013 EBE8FFFFFF | <1> call slen
29 | 28 | | <1>
30 | 29 | 00000018 89C2 | <1> mov edx, eax
31 | 30 | 0000001A 58 | <1> pop eax
32 | 31 | | <1>
33 | 32 | 0000001B 89C1 | <1> mov ecx, eax
34 | 33 | 0000001D BB01000000 | <1> mov ebx, 1
35 | 34 | 00000022 B804000000 | <1> mov eax, 4
36 | 35 | 00000027 CD80 | <1> int 80h
37 | 36 | | <1>

```

Открывание файла листинга

Опишу строчку номер 16:

Здесь “15”-это номер строчки в коде программы “0000000D”- это адрес “5B”- это машинный код “ret” - исходный кол программы

Опишу строчку номер 36:

Здесь “35”-это номер строчки в коде программы “00000027”- это адрес “CD80”- это машинный код “int” - исходный кол программы

Опишу строчку номер 24:

Здесь “23”-это номер строчки в коде программы “0000000F”- это адрес “52”- это машинный код “push” - исходный кол программы

Теперь открываю файл с программой lab7-2.asm и удаляю один операнд в случайном месте.(рис. [??])

A screenshot of a text editor showing a line of assembly code. The line is numbered 30 and contains the instruction 'mov ecx' followed by a semicolon and a comment in Russian: '; иначе 'ecx = C''. The text is highlighted in a dark background.

*Удаление операнда*

Пытаюсь создать файл листинга, но он не создается из-за ошибки.(рис. [??])

A screenshot of a terminal window. The prompt shows the user is in a directory ~/.../arch-pc/labs/lab07/report. The command executed is 'nasm -f elf -l lab7-2.lst lab7-2.asm'. The output shows an error: 'lab7-2.asm:30: error: invalid combination of opcode and operands'.

*Попытка создание файла*

## 5 Задание для самостоятельной работы

1. Создаю файл test.asm.(рис. [??])

A screenshot of a terminal window. The prompt shows the user is in a directory ~/.../arch-pc/labs/lab07/report. The command executed is 'touch test.asm'. The output shows the file test.asm has been created. Below the terminal output, there is a list of files in the directory: Makefile, bib, image, in\_out.asm, lab7-1, lab7-1.asm, lab7-1.o, lab7-2, lab7-2.asm, lab7-2.lst, pandoc, report.md, test.asm.

*Создание файла для самостоятельной работы*

Далее открываю его и пишу программу для нахождения наименьшей из переменных a,b,c в соответствии с моим вариантом (вариант 20).(рис. [??])

```
File Edit Search View Document Help
~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab07/report/test.asm - Mousepad

1 include 'in_out.asm'
2 section .data
3 msg2 db "Наименьшее число: ",0h
4 a dd 95
5 b dd 2
6 c dd 61
7 section .bss
8 min resb 10
9 section .text
10 global _start
11 _start:
12 global main
13 main:
14 mov eax, [a]
15 mov ebx, [b]
16 cmp eax, ebx
17 jl min1
18 min2:
19 mov eax, [c]
20 cmp eax, ebx
21 jl min3
22 jmp .end
23 min3:
24 mov ebx, eax
25 min1:
26 mov eax, [c]
27 cmp eax, ebx
28 jl min4
29 jmp .end
30 min4:
31 mov ebx, eax
32 .end:
33 mov [min], ebx
34 mov eax, msg2
35 call sprint
36 mov eax, [min]
37 call iprintf
38 call quit
```

## Написание программы

Создаю исполняемый файл и запускаю его.(рис. [??])

```
(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab07/report]
$ nasm -f elf test.asm

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab07/report]
$ ld -m elf_i386 -o test test.o

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab07/report]
$ ./test
Наименьшее число: 2
```

## Проверка работы программы

Программа работает верно

### 2. Создаю файл test2.asm.(рис. [??])

```
(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab07/report]
$ touch test2.asm

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab07/report]
$ ls
Makefile  bib  image  in_out.asm  lab7-1  lab7-1.asm  lab7-1.o  lab7-2  lab7-2.asm  lab7-2.lst  pandoc  report.md  test  test.asm  test.o  test2.asm
```

## Создание файла

Начинаю написание программы, которая для введенных с клавиатуры значение x и a вычисляет значение заданной функции f(x) и выводит результат вычислений.(рис. [??])

```
File Edit Search View Document Help
1 %include 'in out.asm'
2 section .data
3 msg1 db 'Введите X: ',0h
4 msg2 db 'Введите A: ',0h
5 msg3 db 'Результат: ',0h
6 section .bss
7 x resb 10
8 a resb 10
9 f resb 10
10 section .text
11 global _start
12 _start:
13 mov eax,msg1
14 call sprint
15 mov ecx,x
16 mov edx,10
17 call sread
18 mov eax,msg2
19 call sprint
20 mov ecx,a
21 mov edx,10
22 call sread
23 mov eax,x
24 call atoi
25 mov [x],eax
26 mov eax,a
27 call atoi
28 mov [a],eax
29 mov edi,[x]
30 mov ecx,edi
31 mov ecx,5
32 mov [f],ecx
33 cmp edi,[a]
34 jl fin
35 mov ebx,[x]
36 mov edi,[a]
37 sub ebx,edi
38 mov [f],eax
39 fin:
40 mov eax,msg3
41 call sprint
42 mov eax,[f]
43 call sprintf
44 call quit
```

## Написание программы

Создаю исполняемый файл и запускаю его.(рис. [??])

```
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report ]
$ nasm -f elf test2.asm

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report ]
$ ld -m elf_i386 -o test2 test2.o

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report ]
$ ./test2
Введите X:1
Введите A:2
Результат:5

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab07/report ]
$ ./test2
Введите X:2
Введите A:1
Результат:1
```

## Проверка программы

Программы работает верно

## 6 Выводы

После выполнения данной лабораторной работы я изучил команды условного и безусловного переходов, приобрел навыки написания программ с использованием переходов и познакомился с назначением и структурой файла листинга.

## Список литературы