

# **Отчет по лабораторной работе №7**

**Дисциплина: архитектура компьютера**

Шония Ника Гигловна

# Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Выполнение лабораторной работы	8
Список литературы	12

## **Список таблиц**

## **Список иллюстраций**

## Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов и знакомство с назначением и структурой файла листинга.

# Задание

1. Реализация переходов в NASM
2. Изучение структуры файлы листинга
3. Задание для самостоятельной работы

# 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.

# Выполнение лабораторной работы

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

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

Ввожу в файл lab7-

```
GNU nano 6.2 /home/nikashoniya/work/arch-pc/lab07/lab7-1.asm *
%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 ; вызов подпрограммы завершения
```

1.asm текст программы из листинга

```
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-1.o
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-1.o
d: команда не найдена
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$
```

Создаю исполняемый файл и запускаю его

Изменяю программу таким образом, чтобы она выводила сначала 'Сообщение № 2', потом 'Сообщение № 1' и завершала работу



```
GNU nano 6.2 /home/nikashoniya/work/arch-pc/lab07/lab7-1.asm
#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'
```

Создаю исполняе-

мый файл и запускаю его

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

Создаю файл lab7-2.asm в каталоге ~/work/arch-pc/lab07

```
GNU nano 6.2 /home/nikashoniya/work/arch-pc/lab07/lab7-2.asm *
%include 'in_out.asm'
section .data
msg1 db 'Введите В: ',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 В: '
mov eax,msg1
call sprintf
; ----- Ввод 'В'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'В' из символа в число
```

Ввожу текст программы в lab7-2.asm

Создаю исполняемый файл и проверяю его работу для разных значений В

```
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 10
Наибольшее число: 50
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 25
Наибольшее число: 50
```

- Изучение структуры файлы листинга Создаю файл листинга для программы из файла lab7-2.asm

```
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
```

Открываю файл листинга lab7-2.lst с помощью текстового редак-

```

GNU nano 6.2 lab7-2.lst
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>
19 <1>
20 <1> ;----- sprint ----->

```

тора

Откры-

ваю файл с программой lab7-2.asm и в любой инструкции с двумя операндами удаляю один операнд. Выполняю трансляцию с получением файла листинга. Выдаёт ошибку после этого действия

```

nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.l
st lab7-2.asm
lab7-2.asm:33: error: symbol 'fin' not defined
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ mc

```

Также в файле ли-

стинга пишут про ошибку

```

31 00000124 8B0D[00000000] mov ecx,[max]
32 0000012A 3B0D[0A000000] cmp ecx,[B] ; Сравниваем 'max(A,C)' и '
33 *****
34 00000130 8B0D[0A000000] mov ecx,[B] ; иначе 'ecx = B'
35 00000136 890D[00000000] mov [max],ecx
36 ; ----- Вывод результата
37

```

- Задание для самостоятельной работы Написала программу нахождения наименьшей из 3 целочисленных переменных  $\boxed{x}$ ,  $\boxed{y}$  и  $\boxed{z}$ . Значения переменных

```

GNU nano 6.2 /home/nikashoniya/work/arch-pc/lab07/lab7-3.asm *
%include 'in_out.asm'

section .data
msg1 db "Наименьшее число:"
a dd 21
b dd 28
c dd 34

section .bss
min resb 10

section .text
global _start

_start:
mov eax, msg1
call sprint

mov ecx, [a]
mov [min], ecx ; 'min = A'

```

Создаю исполняе-

мый файл и проверяю его работу.

```

nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab07$ ./lab7-3
Наименьшее число:21

```

## # Выводы

Я изучила команды условного и безусловного переходов и научилась писать программы с использованием этих переходов.

# Список литературы

Архитектура компьютера Мой репозиторий: [https://github.com/NikaShoniya/study\\_2023-2024\\_arch-pc](https://github.com/NikaShoniya/study_2023-2024_arch-pc)