

# **Отчет по лабораторной работе №7**

**Дисциплина: Архитектура компьютера**

Челухаев Кирилл Александрович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задания</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Изучение структуры файлы листинга . . . . .	9
4.2	Задание для самостоятельной работы . . . . .	11
<b>5</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

# Список иллюстраций

4.1	5	.....	9
4.2	6	.....	11
4.3	7	.....	12

## **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 Задания

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

### **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.

## 4 Выполнение лабораторной работы

Я создал каталог lab07 для лабораторной работы №7 и создал в нем файл lab7-1.asm.(рис. ??).

```
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $  
mkdir lab07  
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $  
cd lab07  
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la  
b07 $ touch lab7-1.asm  
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la  
b07 $
```

Я ввел в файл код программы из ТУИС, скомпилировал исполняемый файл и запустил его.(рис. ??).

```
b07 $ nasm -f elf lab7-1.asm  
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la  
b07 $ ld -m elf_i386 -o lab7-1 lab7-1.o  
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la  
b07 $ ./lab7-1  
Сообщение № 2  
Сообщение № 3  
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la  
b07 $
```

Далее я изменил текст программы добавив или изменив инструкции jmp, чтобы вывод программы был следующим:

*Сообщение № 3* *Сообщение № 2* \**Сообщение № 1*

Я скомпилировал исполняемый файл и проверил его работу (рис. ??).

```
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la  
b07 $ nasm -f elf lab7-1.asm  
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la  
b07 $ ld -m elf_i386 -o lab7-1 lab7-1.o  
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la  
b07 $ ./lab7-1  
Сообщение № 3  
Сообщение № 2  
Сообщение № 1  
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la  
b07 $
```

Далее я создал файл lab7-2.asm и ввел в него код программы, которая опреде-



ляет и выводит на экран наибольшую из 3 целочисленных переменных: А,В и С. Создал исполняемый файл и проверил его работу для разных значений В. (рис. ??).

```
b07 $ touch lab7-2.asm
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la
b07 $ nasm -f elf lab7-2.asm
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la
b07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la
b07 $ ./lab7-2
Введите В: 3
Наибольшее число: 50
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la
b07 $ ./lab7-2
Введите В: 51
Наибольшее число: 51
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la
b07 $
```

## 4.1 Изучение структуры файлы листинга

Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ -l и задав имя файла листинга в командной строке. Я создал файл листинга для программы из файла lab7-2.asm и открыл его. (рис. 4.1).

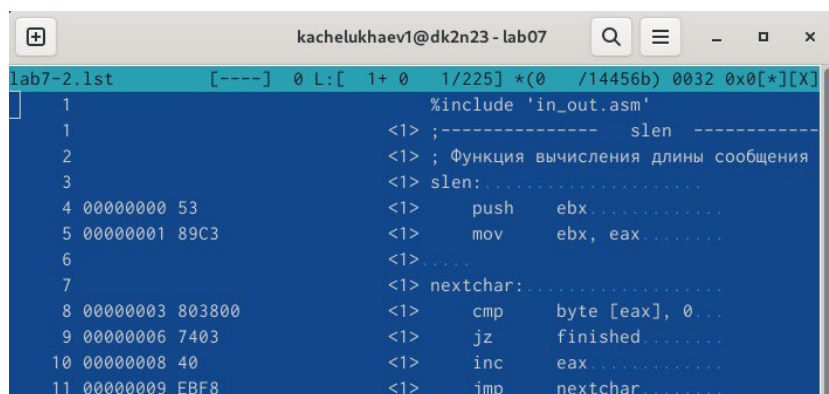


Рис. 4.1: 5

1. Строка 4: 4 00000000 53 <1> push ebx

- Адрес: 00000000 — это первый байт подпрограммы. Код будет помещён в память, начиная с этого адреса

- Машинный код: 53 — это шестнадцатеричное представление машинного кода для инструкции `push ebx`. Инструкция `push` помещает содержимое регистра `ebx` в стек.
- Комментарии: `<1>` указывают на тип элемента в листинге. В данном случае это просто инструкция на ассемблере
- Действие: сохраняет текущее значение регистра `ebx` в стеке. Это необходимо для сохранения значения регистра `ebx`, которое будет использоваться в этой подпрограмме.

## 2. Строка 5: `5 00000001 89C3 <1> mov ebx, eax`

- Адрес: `00000001` — инструкция будет помещена по этому адресу, который на 1 байт больше, чем у предыдущей инструкции.
- Машинный код: `89C3` — это машинный код для инструкции `mov ebx, eax`
- Комментарии: `<1>` — Указывает, что это ассемблерная инструкция.
- Действие: перемещает содержимое регистра `eax` в регистр `ebx`.

## 3. Строка 8: `8 00000003 803800 <1> cmp byte [eax], 0`

- Адрес: `00000003` - Адрес памяти для этой инструкции.
- Машинный код: `803800` - Машинный код для инструкции `cmp byte [eax], 0`.
- Комментарии: `<1>` - Указывает, что это ассемблерная инструкция.
- Действие: Сравнивает байт по адресу, указанному в `eax`, с нулем. Это нужно для определения конца строки.

## 4.2 Задание для самостоятельной работы

1. Напишите программу нахождения наименьшей из 3 целочисленных переменных a b 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.

Я создал файл lab7-3.asm и написал в нем код для программы нахождения наименьшей из 3 целочисленных переменных a b c и проверил его работу (рис. 4.2).

```
b07 $ touch lab7-3.asm
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la
b07 $ nasm -f elf lab7-3.asm
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la
b07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/la
b07 $ ./lab7-3
Введите A: 95
Введите B: 2
Введите C: 61
Наименьшее число: 2kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура ко
мпьютера/arch-pc/lab07 $
```

Рис. 4.2: 6

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. Я создал файл lab7-4.asm и написал в нем код для вычисления функции в соответствии с моим вариантом (рис. ??).

$$20 \quad \begin{cases} x - a, & x \geq a \\ 5, & x < a \end{cases} \quad (1;2) \quad (2;1)$$

и проверил его работу (рис. 4.3).

```
мьютера/arch-pc/lab07 $ touch lab7-4.asm
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf lab7-4.asm
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07 $ ./lab7-4
Введите значение x: 1
Введите значение a: 2
Результат: 5kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07 $ ./lab7-4
Введите значение x: 2
Введите значение a: 1
Результат: 1kachelukhaev1@dk2n23 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07 $
```

Рис. 4.3: 7

## 5 Выводы

В итоге я изучил команды условного и безусловного переходов и приобрел навыки написания программ с использованием переходов. познакомился с значением и структурой файла листинга.

## **Список литературы**