**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7**

*дисциплина: Архитектура компьютера*

Студент: Армихос Гонзалез Карла

Группа: НКАбд-02-24

**МОСКВА**

2024 г.

Содержание

[1. Цель работы 4](#_rk73iti44iis)

[2. Задание 5](#_excfirwaqflf)

[3. Технические введение 6](#_16umgbxblflt)

[4. Порядок выполнения лабораторной работы 9](#_427nfev9i8fe)

[5. Задание для самостоятельной работы 14](#_s48tflb8idx8)

[6. Выводы 17](#_pv6euzc6a234)

[7. Список используемой литературы 18](#_g3c17rgwaije)

Список иллюстраций

### 

[пример использования команды сравнения и команд условного перехода. 7](#_eb51ocj67i1w)

[рис. 4.1.1 Создание директории 9](#_3tjj6v1auxhz)

[рис. 4.2 Программа jmp 9](#_95osf6go4kim)

[рис. 4.3 Запустить программу (2-3) 10](#_2hrru8izbpxl)

[рис. 4.4 Добавление новых инструкций в программу 11](#_ghvmhwpffbn8)

[рис. 4.5 Запустить программу(2,1) 11](#_x83t72u6ya2y)

[рис. 4.1.7 Изменяем программу 11](#_vgvt2hdf5mqp)

[рис 4.7 Запустить программу 12](#_387ar9ycscr9)

[рис 4.8 Создайте файл lab7-2.asm 12](#_cb8he3vri8t5)

[рис 4.9 12](#_fuuehllnxa5m)

[рис 4.10 программа 13](#_zaepimph5z80)

[рис 5.2 Программа 16](#_qczd7ex9hyk3)

### 

## 

# **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. Работа с числовыми символами.
2. Запускать разные программы
3. Выполнение арифметических задач.
4. Выполнять работу самостоятельно

# **3. Технические введение**

Файл листинга в NASM используется для анализа и отладки программы. Он представляет собой текстовый файл, в котором содержатся адреса инструкций, машинный код, исходный текст программы и дополнительные комментарии. Основное отличие от текста программы заключается в том, что текст программы — это чисто исходный код на языке ассемблера, а листинг дополнительно содержит машинные коды и помогает понять, как исходный код транслируется в команды для процессора.

Файл листинга NASM состоит из следующих частей:

* **Адресация**: Показаны адреса каждой команды в памяти.
* **Машинный код**: Байт-код инструкций, который генерируется компилятором.
* **Исходный текст**: Код на языке ассемблера.
* **Комментарии**: Добавлены для наглядности, чтобы облегчить понимание.

Пример структуры:

00000000 B8 01 00 00 00 mov eax, 1

Ветвление в ассемблере выполняется с помощью переходов. Оно может быть:

* **Безусловным** (выполняется всегда).
* **Условным** (зависит от результата предыдущих операций, таких как сравнение или арифметика).
* **Безусловные переходы**:  
  *jmp* — переход на указанную метку или адрес.
* **Условные переходы**:  
  je / jz — переход при равенстве / нулевом флаге.
* jne / jnz — переход при неравенстве / отсутствии нулевого флага.
* jg / jnle — переход, если больше (без знака).
* jl / jnge — переход, если меньше (без знака).
* jge / jnl — переход, если больше или равно.
* jle / jng — переход, если меньше или равно.

Команда **cmp** используется для сравнения двух операндов. Она выполняет вычитание второго операнда из первого, не сохраняя результат, а только устанавливая флаги процессора:

* **ZF (Zero Flag)**: Устанавливается, если операнды равны.
* **CF (Carry Flag)**: Устанавливается, если произошло заимствование (в случае беззнаковых чисел).
* **SF (Sign Flag)**: Указывает знак результата.
* **OF (Overflow Flag)**: Устанавливается при переполнении.

Команд условного перехода

Синтаксис условного перехода в NASM:

<команда перехода> <метка>

Пример:

je equal\_label

### пример использования команды сравнения и команд условного перехода.

Пример:

mov eax, 5

cmp eax, 10 ; Сравнение 5 и 10

jl less\_label ; Переход, если eax < 10

; Продолжение программы

less\_label:

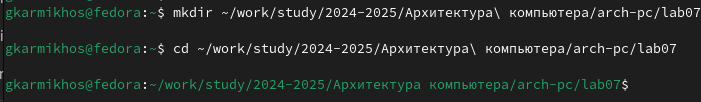
; Код выполняется только если eax < 10

### 

Безусловные переходы (jmp) не анализируют флаги. Они всегда выполняются, вне зависимости от состояния процессора.

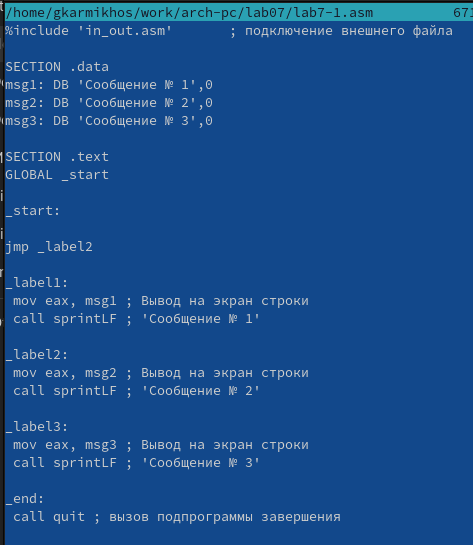
# **4. Порядок выполнения лабораторной работы**

Создайте каталог для программ лабораторной работы № 7, перейдите в него и создайте файл lab7-1.asm: (рис. 4.1)



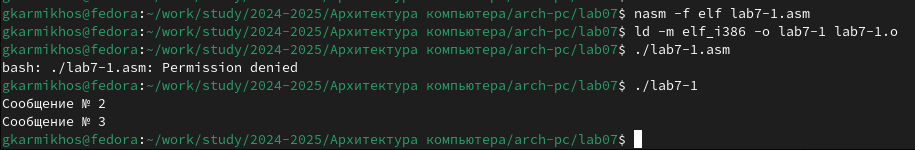
### рис. 4.1.1 Создание директории

В файле 7-1. asm напишите программу вывода значения регистра eax (рис. 4.2)



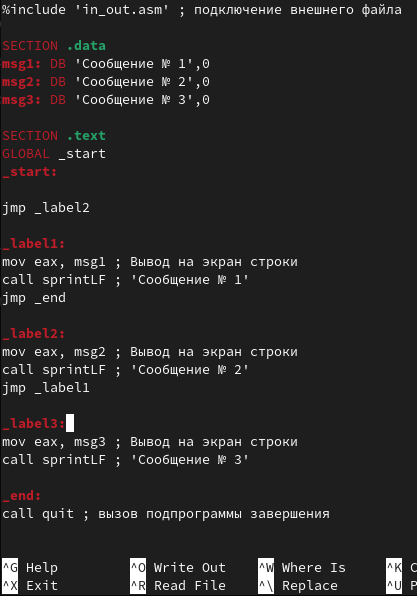
### рис. 4.2 Программа jmp

Создайте исполняемый файл и запустите его. В этом случае, когда отображается значение регистра jmp, результатом будет: сообщение 2, сообщение 3 (рис.4.3)

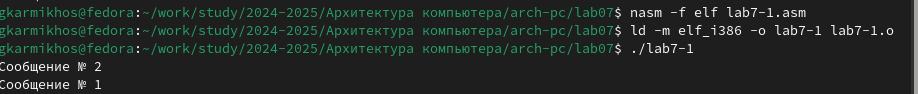


### рис. 4.3 Запустить программу (2-3)

Изменить текст программы: добавляем jmp \_end в \_label1: и jmp\_label1 в \_label2 (рис. 4.4).

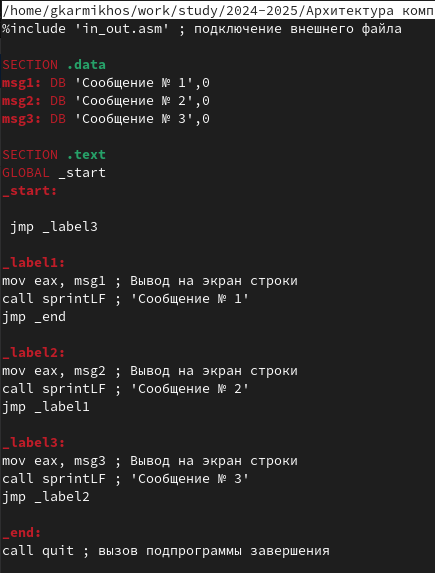


### рис. 4.4 Добавление новых инструкций в программу

При сохранении и запуске program мы получаем в результате сообщение 2 и сообщение3(рис.4.5)

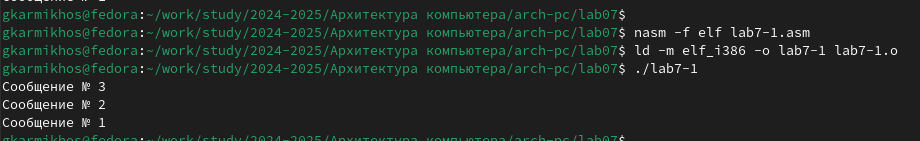
### рис. 4.5 Запустить программу(2,1)

Мы модифицируем программу так, чтобы при отображении она отображалась в порядке 3,2,1 (рис. 4.6)

****

### рис. 4.1.7 Изменяем программу

Запускаем программу, чтобы увидеть ее результат (рис 4.7)



### рис 4.7 Запустить программу

Создайте файл lab7-2.asm рис 4.8



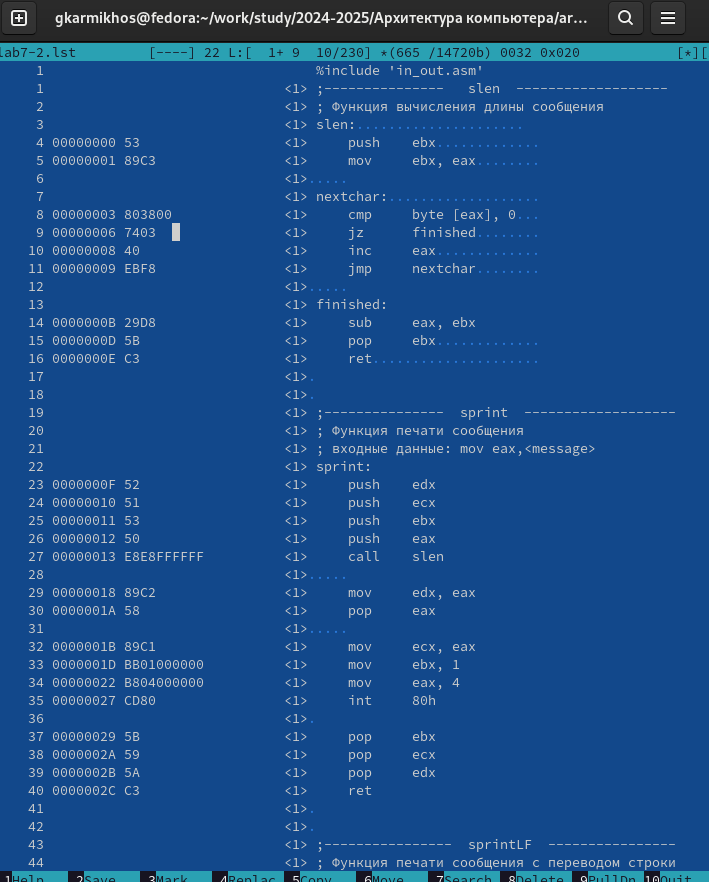
### рис 4.8 Создайте файл lab7-2.asm

рис 4.9

### рис 4.9

Копируем программу (рис 4.10)

### 

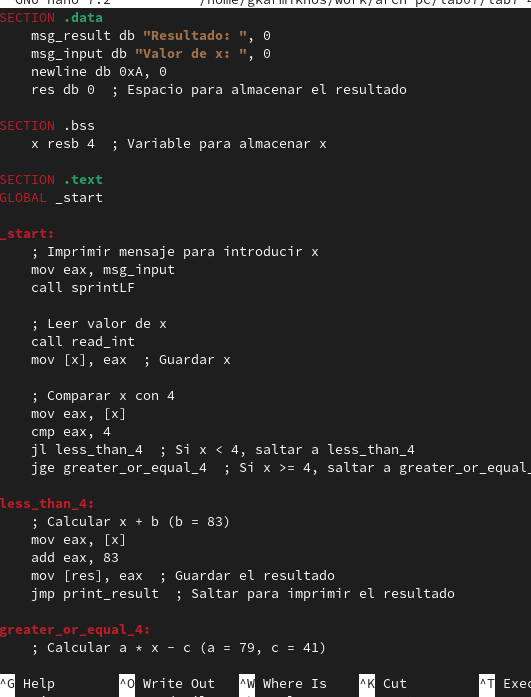


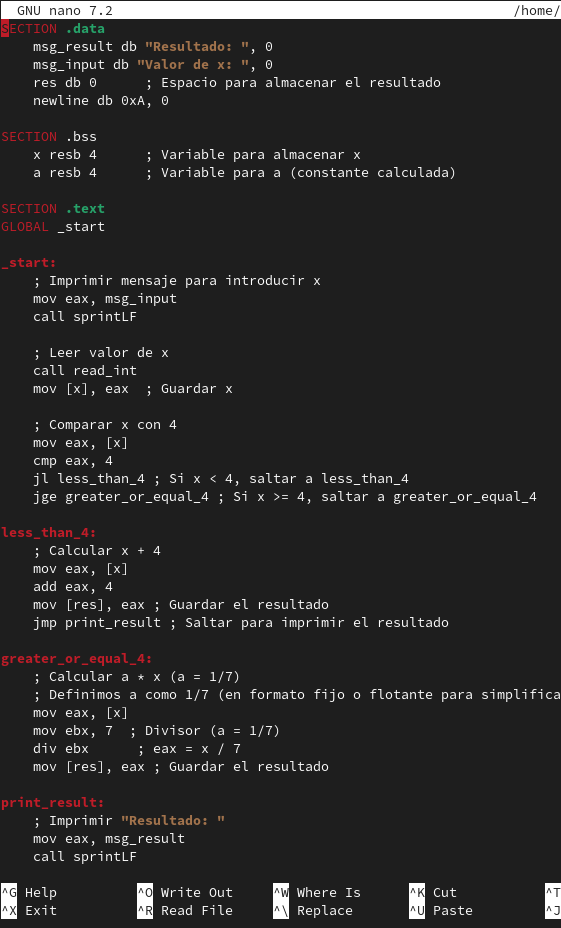
### рис 4.10 программа

### 

# **5. Задание для самостоятельной работы**

Написать программу вычисления выражения (рис 5.1) (5.2).

рис 5.1



### рис 5.2 Программа

### 

### 

# **6. Выводы**

Использовали метки (jmp, je, jl, jge) для управления потоком программы и реализации условий.

Узнали, как считывать и выводить данные с помощью процедур sprintLF, read\_int и print\_int.

Управляли секциями данных (.data и .bss) для хранения констант, переменных и промежуточных результатов.

Мы модифицировали программу с метками для изменения порядок вывода заранее определённых сообщений.

# **7. Список используемой литературы**

1. [Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089086/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%966.%20%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8%20%D0%B2%20NASM..pdf)