Отчёт по лабораторной работе №7

Дисциплина: Архитектура компьютера

Машков Илья Евгеньевич

Содержание

# 1 Цель работы

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

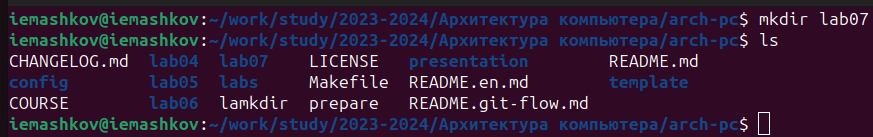
# 2 Задание

1. Реализация переходов в NASM
2. Изучение структуры файла листинга
3. Выполнение заданий для самостоятельной работы

# 3 Выполнение лабораторной работы

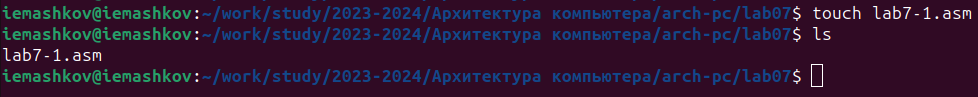
## 3.1 Реализация переходов в NASM

Для начала в папке локального репозитория я создаю директорию **lab07** для дальнейшей работы в ней (рис. [??]).



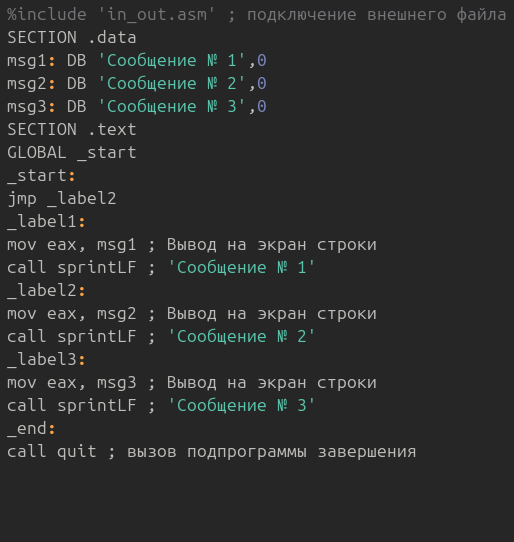
Директория lab07.

Далее я перехожу в эту директорию и создаю файл **lab7-1.asm** с помощью команды **‘touch’**, а также копирую файл **in\_out.asm** (Рис. [??]).



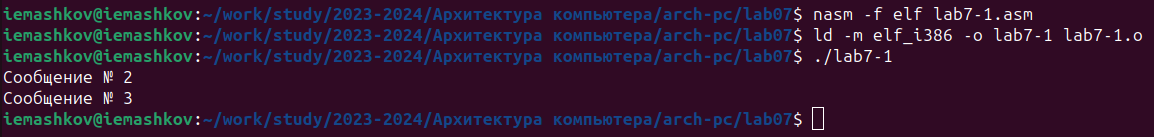
Создание рабочего файла.

Затем я ввожу код в .asm файл (Рис. [??]).



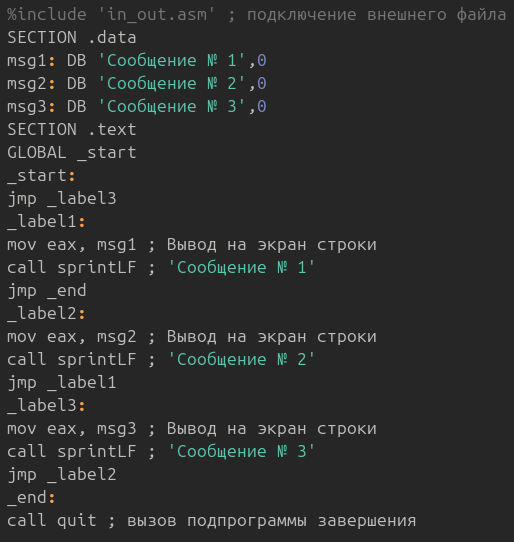
Код программы.

Создаю исполняемый файл и запускаю программу (Рис. [??]). В выводе программы я получаю символ **‘Сообщение №2’** и **‘Сообщение №3’**.



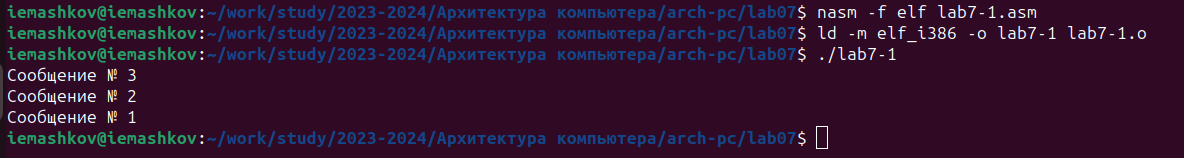
Результат выполнения программы.

Теперь я меняю в программе пару строк, чтобы получился такой вывод: **‘Сообщение №3’**, **‘Сообщение №2’**, **‘Сообщение №1’**. (Рис. [??]).



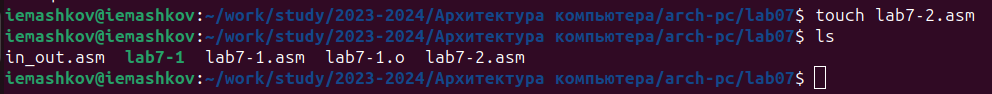
Изменённый код программы.

Создаю исполняемый файл и запускаю его (Рис. [??]). В выводе получаю именно то, что мне и надо было.



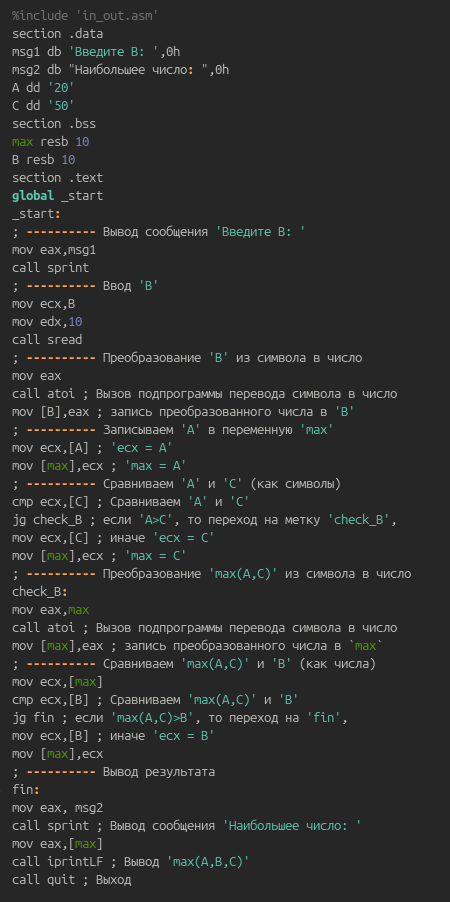
Результат второго выполнения программы.

Создаю файл **lab7-2.asm** (Рис. [??]).



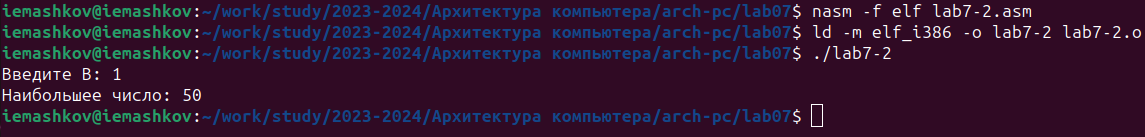
Создание файла lab7-2.asm.

Ввожу код программы (Рис. [??]).



Код программы.

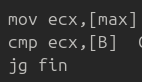
Создаю исполняемый файл и запускаю программу (Рис. [??]). В выводе получаю число **50**, т.к. это и есть максимальное из всех значений, но если в **‘B’** ввести число, которое будет больше 50-ти, то программа выведет введённое с клавиатуры число.



Результат выполнения программы.

## 3.2 Изучение структуры файла листинга

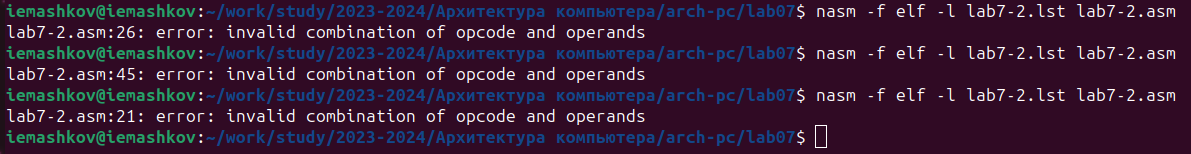
С помощью команды **‘nasm -f elf -l lab7-2.lst lab7-2.asm’** создаю файл листинга, внимательно изучаю его структуру. После чего я выбрал три строчки к которым буду писать пояснение (Рис. [??]).



Строчки программы в листинге.

1. Командой **‘mov ecx,[max]’** мы перемещаем в регистр **ecx** значения из **max**, где лежат результаты сравнения **А** и **С**.
2. Комманда **‘cmp ecx,[B]’** похволяет нам провести сравнение **В** со значениями **А** и **С**.
3. Если **‘max(A, C) > B’** то команда **‘jg’** позволяет нам перейти на финальный этап программы, обозначенный меткой **‘fin’**. Команда **‘jg’** позволяет перейти к следующей метке только если мы получаем такое сравнение: **‘A > B’** - иначе будет использоваться **‘jl’**, **‘jle’**, **‘jge’** и т.д..

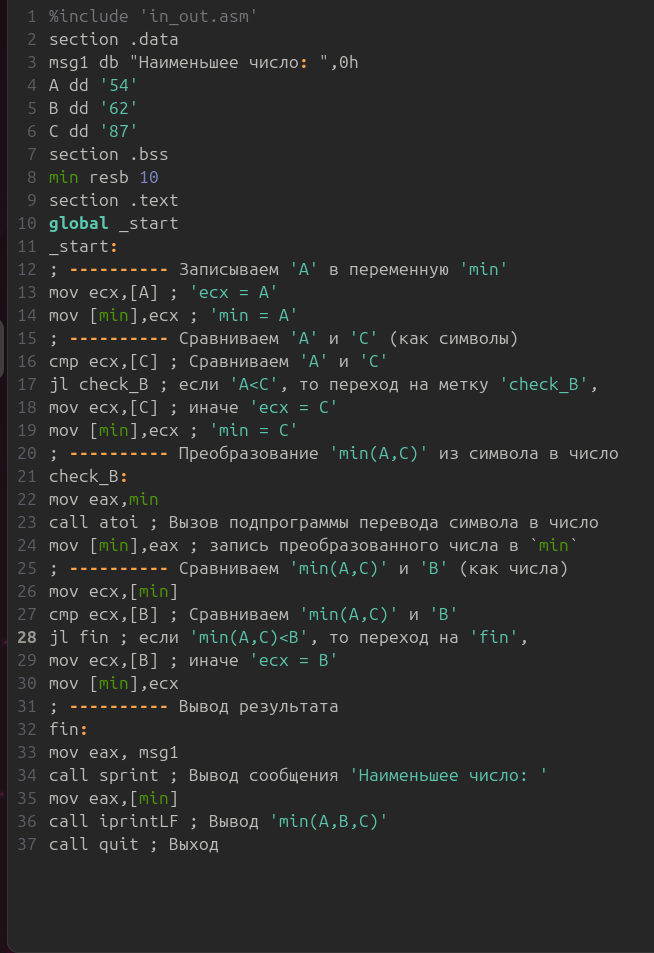
Далее по заданию мне нужно было убрать из любой инструкции с двумя операндами один операнд и выполнить трансляцию с получением файла листинга. Я сделал это в разных местах, но при трансляции вылетала ошибка, а изменённый файл листинга не создавался, т.к. программа не может работать при отсутствии одного операнда. (Рис. [??]).



Ошибки при трансляции.

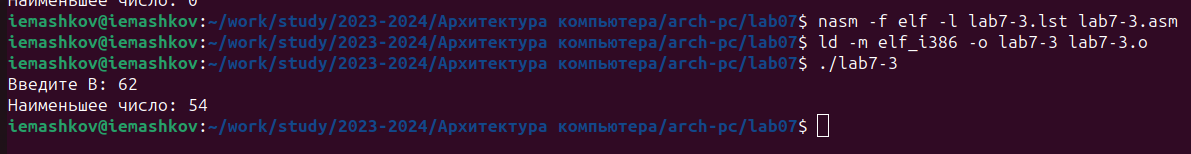
## 3.3 Выполнение заданий для самостоятельной работы

1. Создаю файл **lab7-3.asm** и ввожу код программы, которая будет находить минимальное из значений **А**, **В** и **С**. Для этого я немного поменял код программы **‘lab7-2.asm’** (Рис. [??]).



Код программы.

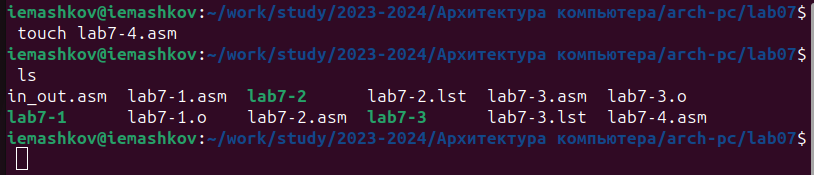
Создаю исполняемый файл и запускаю его со значениями из 5-го варианта: **‘54’**, **‘62’**, **‘87’** (Рис. [??]).



Результаты выполненй программы.

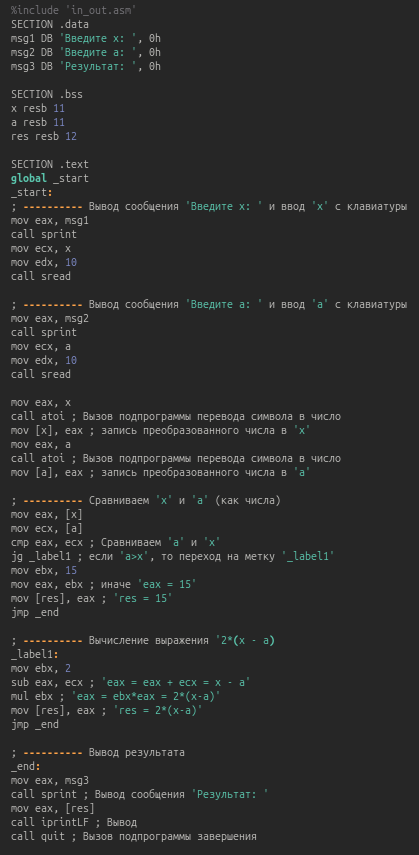
**P.s.**: Изначально я запумкал вариант программы, где значение **В** запрашивалось с клавиатуры, но, как видно из (Рис. [??]), я поменял программу так, чтобы ничего с клавиатуры не запрашивалось и в выводе получал число **‘54’**, что и является правильным ответом.

1. Создаю файл **lab7-4.asm** (Рис. [??]).



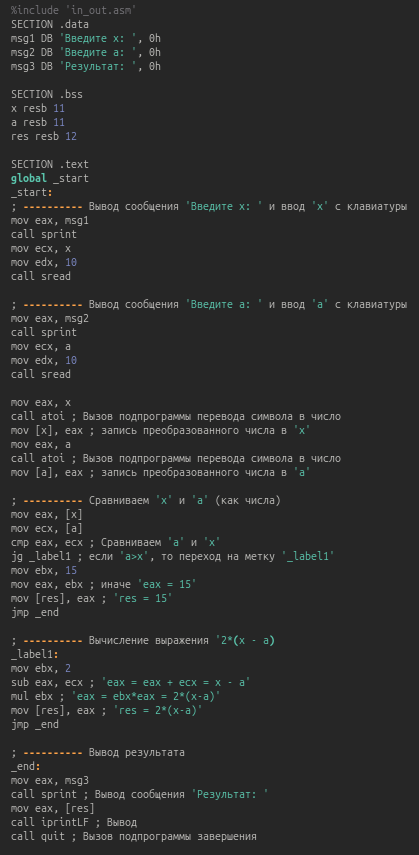
Создание .asm файла.

Ввожу код программы, которая будет запрашивать значения **x** и **a** с клавиатуры и после сравнения их значений вычислять значения функции, которые указаны напротив результатов сравнения в варианте **5** (Рис. [??]).



Код программы.

Далее создаю исполняемый файл и запускаю программу в первый раз со значениями **(1;2)**, а потом и с **(2;1)** (Рис. [??])



Результаты выполнений программы.

Результаты я проверил, пожтому могу сказать, что программа отработала верно.

# 4 Выводы

При выполнении данной лабораторной работы я освоил переходы в NASM, а также научился создавать и разбираться в файле листинга.

# 5 Список литературы

[Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089087/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%967.%20%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B%20%D0%B1%D0%B5%D0%B7%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B8%20%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B4%D0%BE%D0%B2%20%D0%B2%20Nasm.%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%82%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9..pdf)