ОТЧЕТ по лабораторной работе №7

дисциплина: Архитектура компьютера

Студент: Идрисов Д.А.

Содержание

# 1 Цель работы

Целью работы является изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

## 1.1 Задание

1. Изучение механизма переходов в авссемблере
2. Изучение листнга программы
3. Выполнение заданий для самостоятельной работы

## 1.2 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.

Безусловный переход выполняется инструкцией jmp (от англ. jump – прыжок), которая включает в себя адрес перехода, куда следует передать управление:

jmp <адрес\_перехода>

Команда условного перехода имеет вид

j<мнемоника перехода> labe

Мнемоника перехода связана со значением анализируемых флагов или со способом формирования этих флагов.

Инструкция cmp является одной из инструкций, которая позволяет сравнить операнды и выставляет флаги в зависимости от результата сравнения. Инструкция cmp является командой сравнения двух операндов и имеет такой же формат, как и команда вычитания:

cmp <операнд\_1>, <операнд\_2>

# 2 Выполнение лабораторной работы

## 2.1 Реализация переходов в NASM

Я создал каталог для программ лабораторной работы № 7 и файл lab7-1.asm.

Инструкция jmp в NASM используется для выполнения безусловных переходов. Рассмотрим пример программы, в которой используется инструкция jmp. Написал текст программы из листинга 7.1 в файле lab7-1.asm. (рис. [[1](#fig:001)])

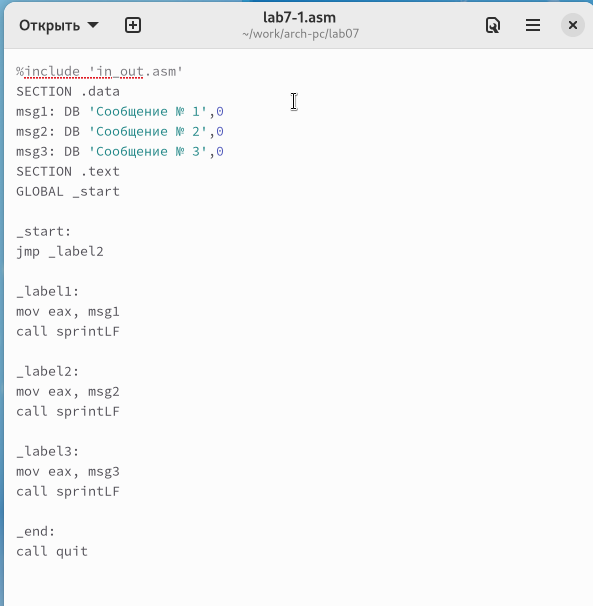


Figure 1: Изменение кода

Создал исполняемый файл и запустил его. (рис. [[2](#fig:002)])

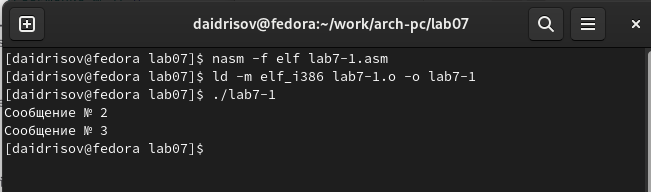


Figure 2: Запуск программы

Инструкция jmp позволяет осуществлять переходы не только вперед, но и назад. Мы изменим программу так, чтобы она сначала выводила “Сообщение № 2”, затем “Сообщение № 1” и завершала работу. Для этого мы добавим в текст программы после вывода “Сообщения № 2” инструкцию jmp с меткой \_label1 (чтобы перейти к инструкциям вывода “Сообщения № 1”) и после вывода “Сообщения № 1” добавим инструкцию jmp с меткой \_end (чтобы перейти к инструкции call quit).

Изменил текст программы в соответствии с листингом 7.2. (рис. [[3](#fig:003)] [[4](#fig:004)])

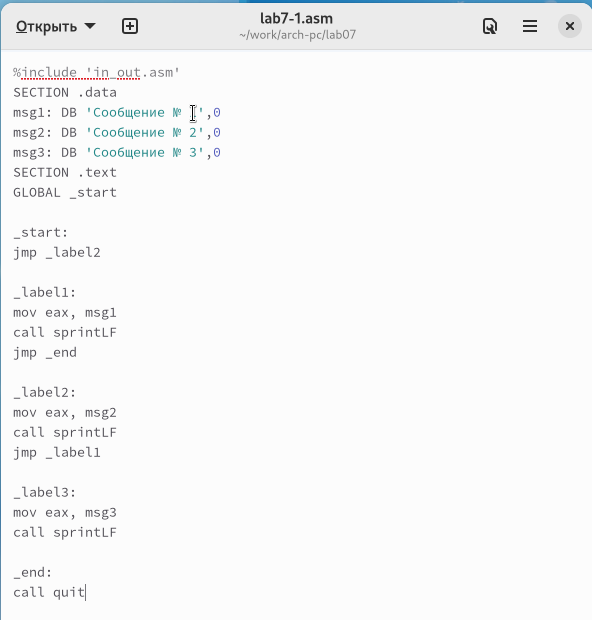


Figure 3: Изменение кода

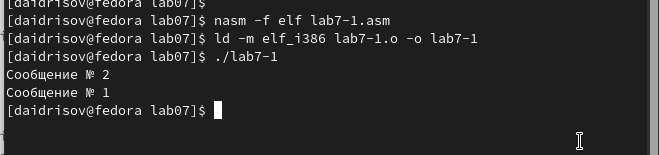


Figure 4: Запуск программы

Изменил текст программы (рис. [[5](#fig:005)] [[6](#fig:006)]), изменив инструкции jmp, чтобы вывод программы был следующим:

Сообщение № 3  
Сообщение № 2  
Сообщение № 1

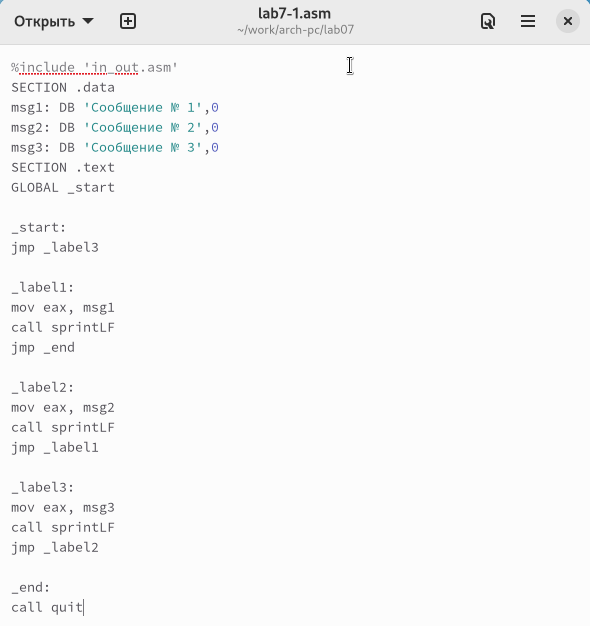


Figure 5: Изменение кода

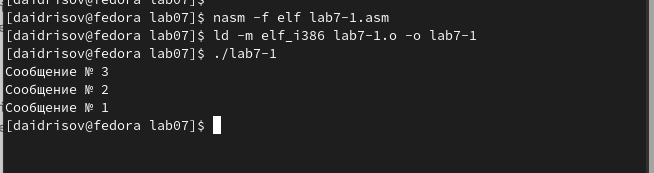


Figure 6: Запуск программы

Использование инструкции jmp приводит к переходу в любом случае. Однако, часто при написании программ необходимо использовать условные переходы, то есть переход должен происходить, если выполнено какое-либо условие.

Давайте рассмотрим программу, которая определяет и выводит на экран наибольшую из трех целочисленных переменных: A, B и C. Значения для A и C задаются в программе, а значение B вводится с клавиатуры.

Создал исполняемый файл и проверил его работу для разных значений B. (рис. [[7](#fig:007)] [[8](#fig:008)])

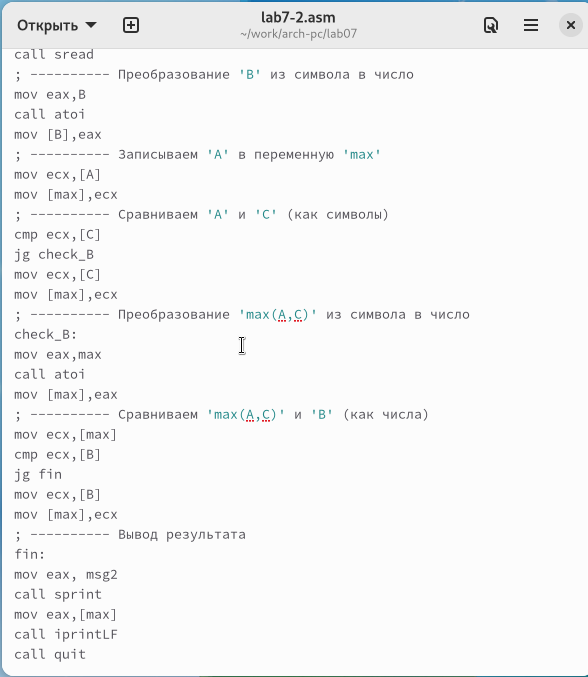


Figure 7: Изменение кода

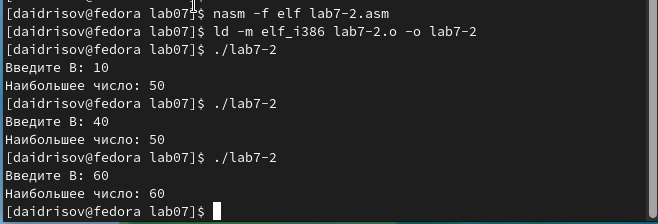


Figure 8: Запуск программы

## 2.2 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.

Создал файл листинга для программы из файла lab7-2.asm (рис. [[9](#fig:009)])

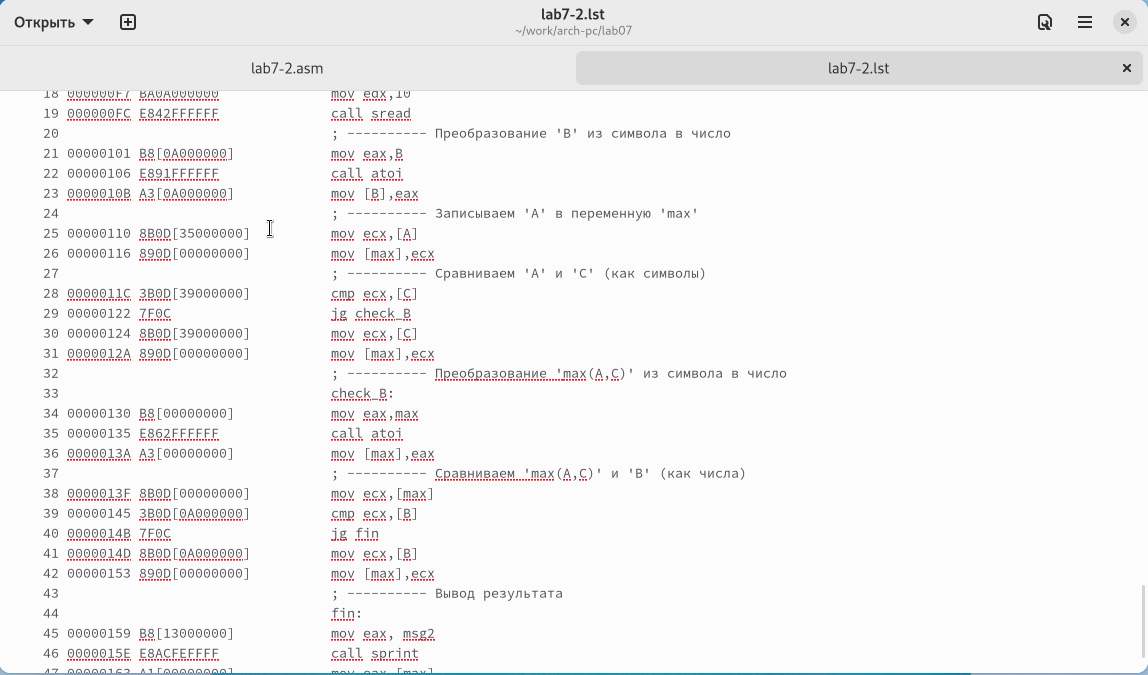


Figure 9: Файл листинга

Внимательно ознакомился с его форматом и содержимым. Подробно объясню содержимое трёх строк файла листинга.

строка 203

* 28 - номер строки в подпрограмме \*
* 0000011C - адрес \*
* 3B0D[39000000] - машинный код \*
* cmp ecx,[C] - код программы - сравнивает ecx и переменную \*

строка 204

* 29 - номер строки в подпрограмме \*
* 00000122 - адрес \*
* 7F0C - машинный код \*
* jg check\_B - код программы - если сравнение покажет что одно число больше то пеход к метке check\_B\*

строка 205

* 30 - номер строки в подпрограмме \*
* 00000124 - адрес \*
* 8B0D[39000000] - машинный код \*
* mov ecx,[C] - код программы - копирует переменную С в ecx \*

Открыл файл с программой lab7-2.asm и в инструкции с двумя операндами удалил один операнд. Выполнил трансляцию с получением файла листинга. (рис. [[10](#fig:010)]) (рис. [[11](#fig:011)])

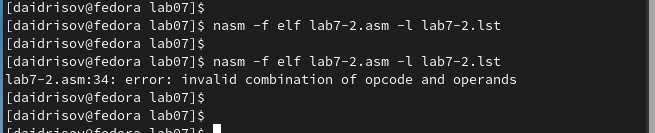


Figure 10: Ошибка трансляции

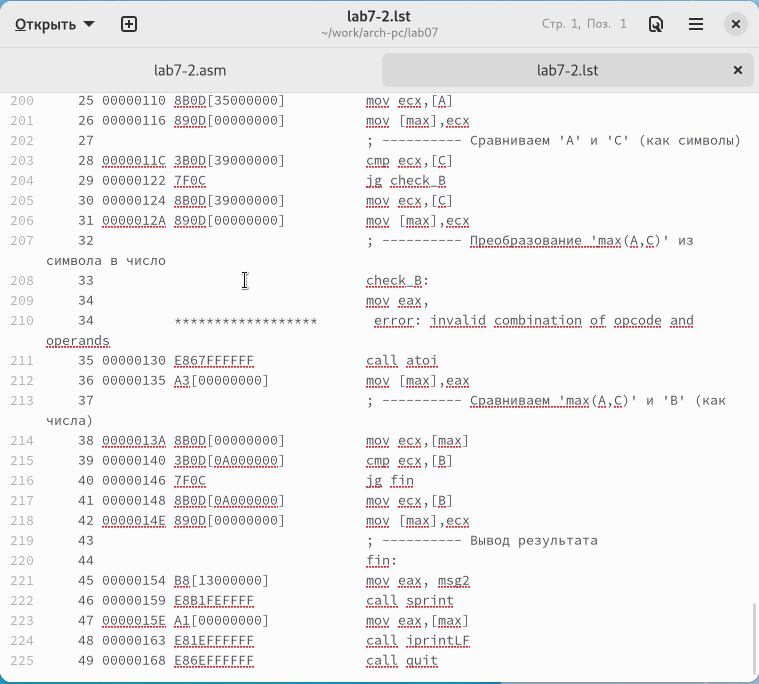


Figure 11: Файл листинга

Объектный файл не смог создаться из-за ошибки. Но получился листинг, где выделено место ошибки.

## 2.3 Задание для самостоятельной работы

Напишите программу нахождения наименьшей из 3 целочисленных переменных a,b и c.  Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 6. Создайте исполняемый файл и проверьте его работу (рис. [[12](#fig:012)]) (рис. [[13](#fig:013)])

Для варианта 17 - числа: 26,12,68

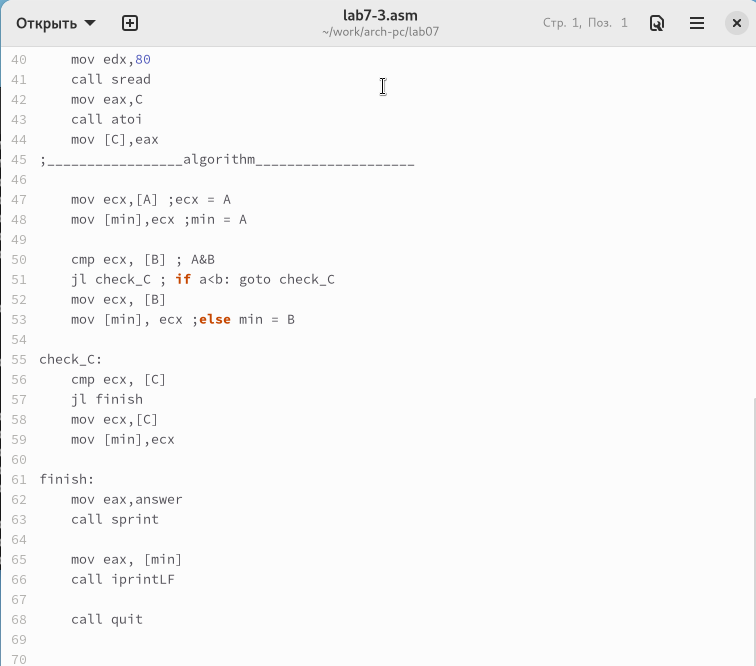


Figure 12: Изменение кода

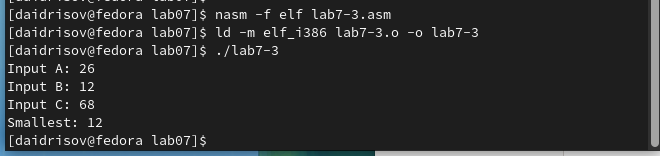


Figure 13: Запуск программы

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. (рис. [[14](#fig:014)]) (рис. [[15](#fig:015)])

Мой вариант 17

Если подставить , тогда

Если подставить , тогда

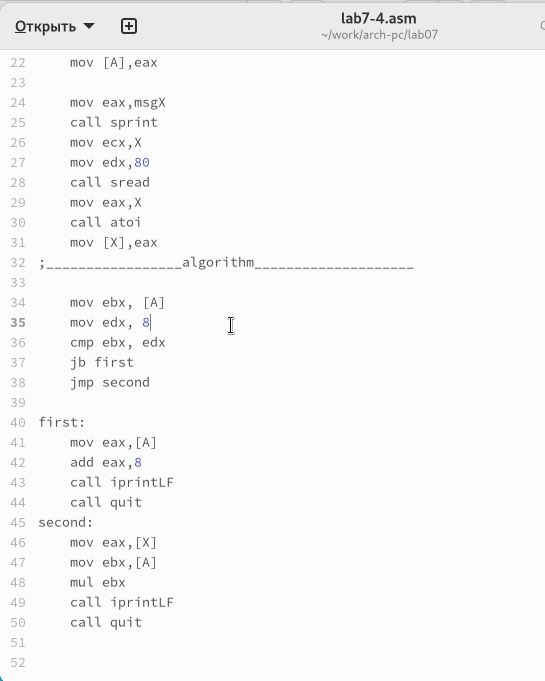


Figure 14: Изменение кода

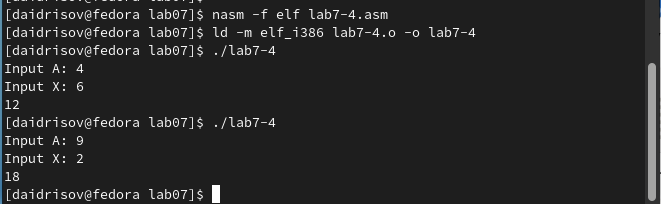


Figure 15: Запуск программы

# 3 Выводы

Изучили команды условного и безусловного переходов, познакомились с фалом листинга.