Отчёт по лабораторной работе 7

Дисциплина: архитектура компьютера

Давлетова Мадина

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.

# 2 Задание

1. Изучение команд условного и бкзусловного перехода
2. Изучение файла листинга
3. Выполнение заданий, рассмотрение примеров
4. Выполнение заданий 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.

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

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

Команда условного перехода имеет вид

j<мнемоника перехода> label

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

# 4 Выполнение лабораторной работы

## 4.1 Реализация переходов в NASM

Я создала каталог для программ лабораторной работы No7 и файл lab7-1.asm.

Инструкция jmp в NASM используется для безусловных переходов. Давайте рассмотрим пример программы с использованием jmp. Я написала текст программы из листинга 7.1 в файле lab7-1.asm (рис. [[1](#fig:001)]).

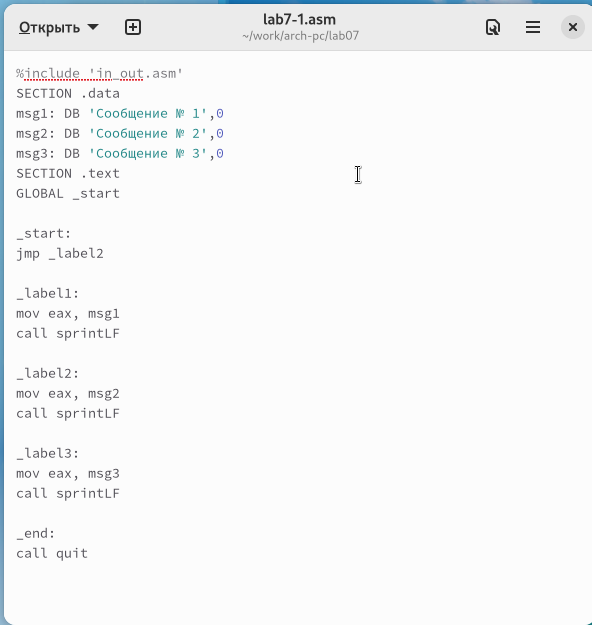


Figure 1: Изменение кода lab7-1.asm

Затем создала исполняемый файл и запустила его (рис. [[2](#fig:002)]).

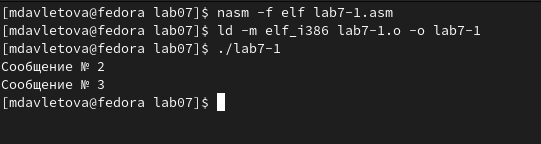


Figure 2: Компиляция текста программы lab7-1.asm

Инструкция jmp позволяет осуществлять переходы не только вперед, но и назад. Я изменила программу так, чтобы сначала выводилось “Сообщение No2”, потом “Сообщение No1”, а затем происходил выход. Для этого после вывода “Сообщения No2” добавила инструкцию jmp с меткой \_label1 (переход к выводу “Сообщения No1”). А после вывода “Сообщения No1” добавила инструкцию jmp с меткой \_end (переход к call quit).

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

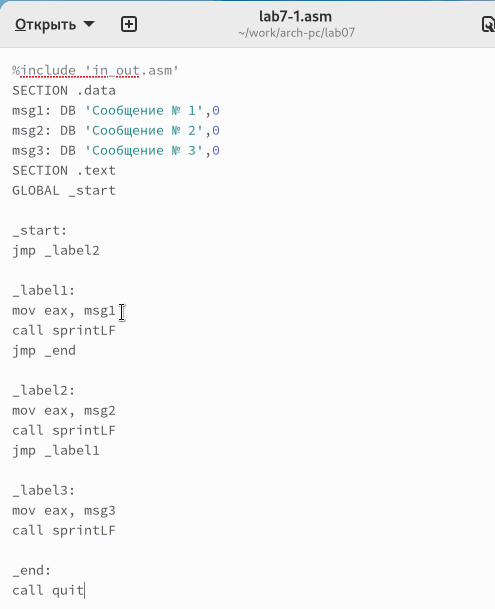


Figure 3: Изменение кода lab7-1.asm

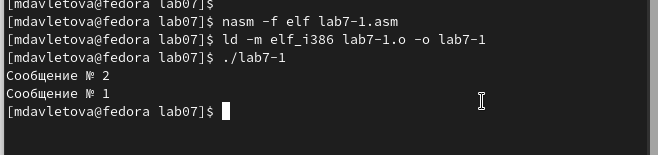


Figure 4: Компиляция текста программы lab7-1.asm

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

Сообщение № 3  
Сообщение № 2  
Сообщение № 1

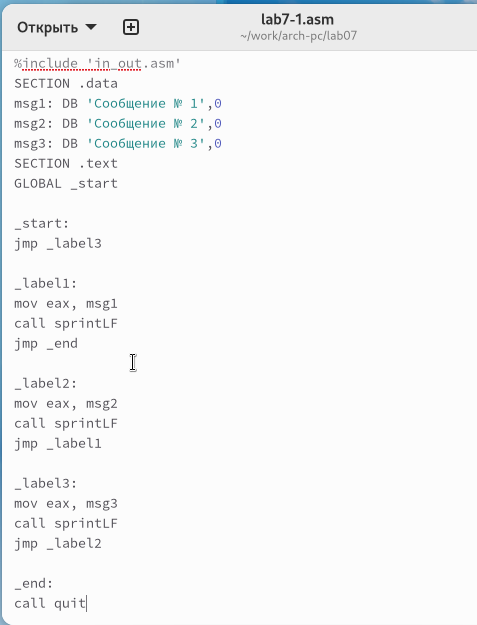


Figure 5: Изменение кода lab7-1.asm

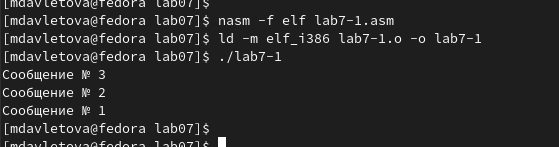


Figure 6: Компиляция текста программы lab7-1.asm

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.

Я рассмотрела программу, которая определяет и выводит наибольшее из трех чисел: A, B и C. Значения для A и C задаются в коде, а значение B вводится с клавиатуры. (рис. [[7](#fig:007)])

Создала исполняемый файл и проверила его работу для разных значений B. (рис. [[8](#fig:008)])

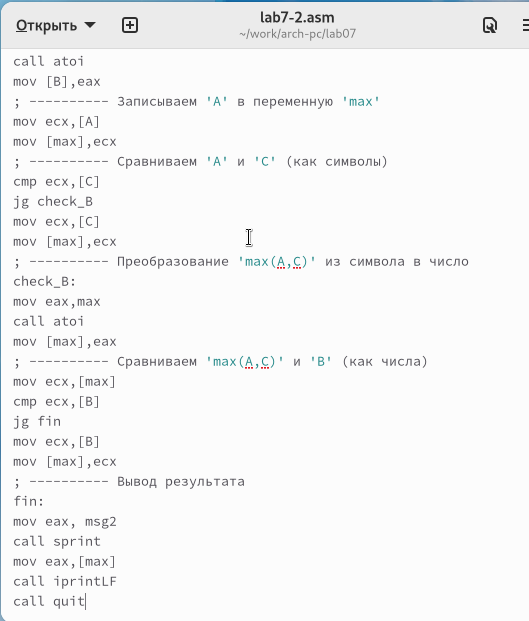


Figure 7: Изменение кода lab7-2.asm

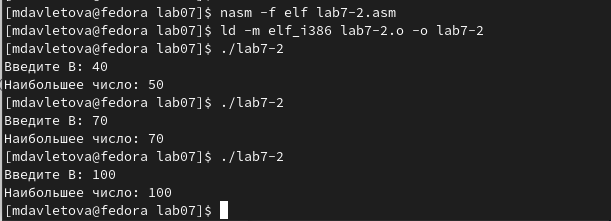


Figure 8: Компиляция текста программы lab7-2.asm

## 4.2 Изучение структуры файлы листинга

Обычно nasm создаёт только объектный файл после ассемблирования. Чтобы получить файл листинга, нужно указать ключ -l и задать имя файла листинга в командной строке.

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

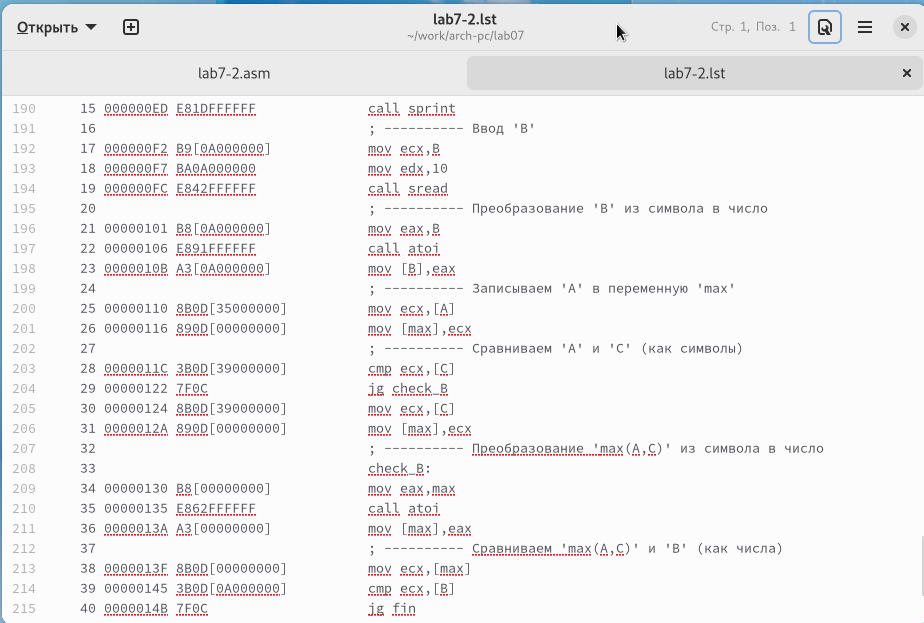


Figure 9: Файл листинга lab7-2

Внимательно ознакомилась с его форматом и содержимым. Подробно объясню содержимое трёх строк этого листинга.

**строка 203**

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

**строка 204**

* 29 - номер строки в подпрограмме
* 00000122 - адрес
* 7F0C - машинный код
* jg check\_B - код программы - если больше переходит к метке check\_B

**строка 205**

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

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

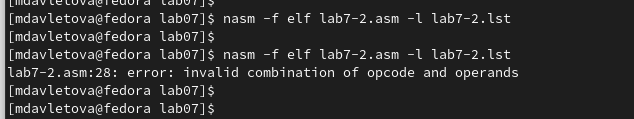


Figure 10: Ошибка трансляции lab7-2

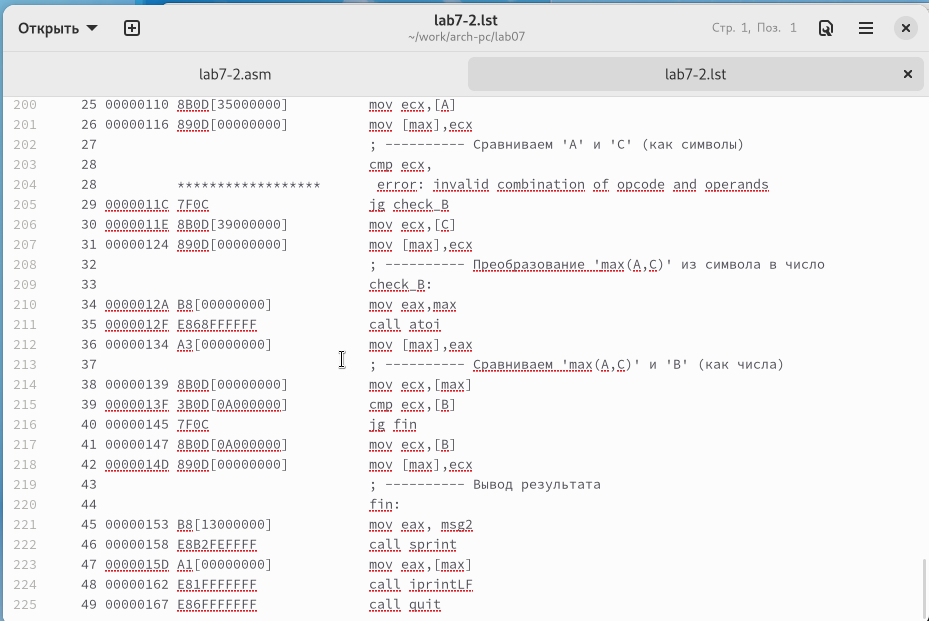


Figure 11: Файл листинга с ошибкой lab7-2

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

## 4.3 Выполнение заданий 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)])

Мой вариант 7 - числа: 45, 67, 15

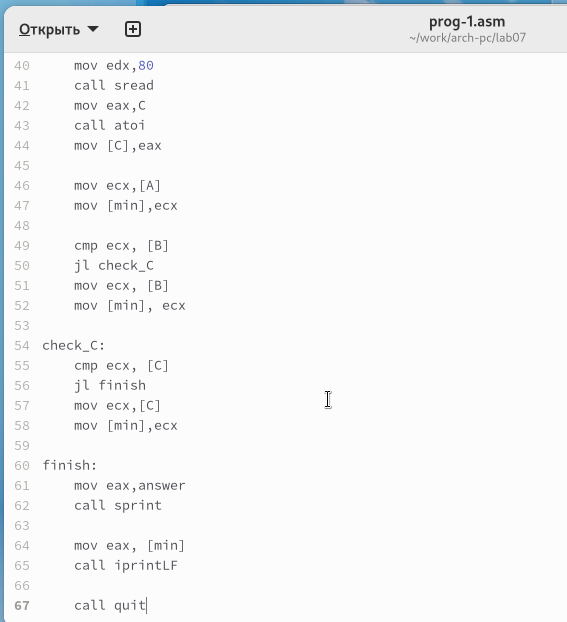


Figure 12: Изменение кода prog-1.asm

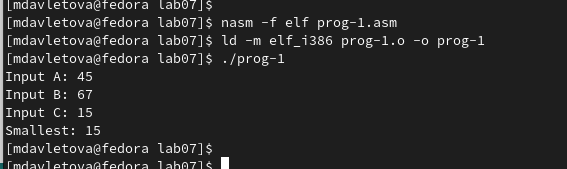


Figure 13: Компиляция текста программы prog-1.asm

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 7

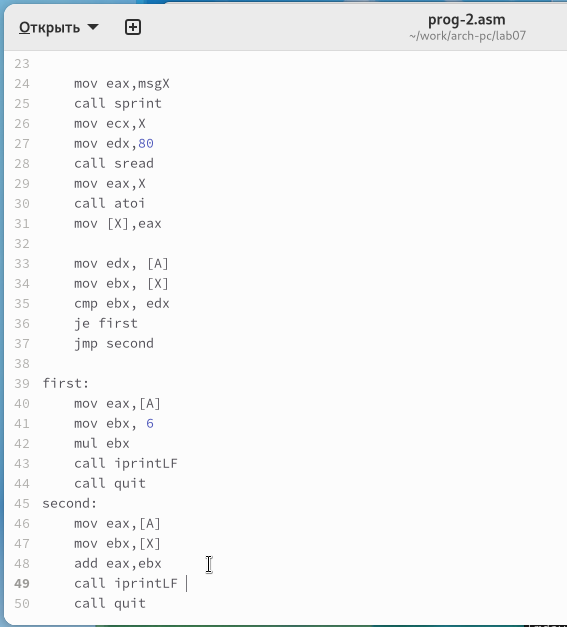


Figure 14: Изменение кода prog-2.asm

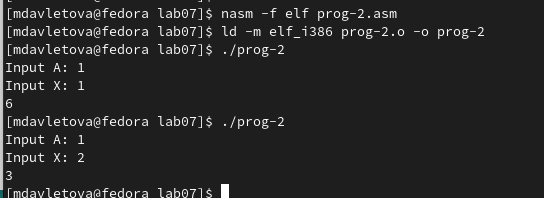


Figure 15: Компиляция текста программы prog-2.asm

# 5 Выводы

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