Лабораторная работа №8

Щетинин Даниил Николаевич

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

# 3 Выполнение лабораторной работы

**Шаг 1**

Создадим каталог для лабораторной работы 8, а также файл lab8-1.asm:

Введём в него текст из Листинга 8.1, для того чтобы понять принцип работы инструкции jmp

Создадим исполняемый файл и запустим его:

(рис. 1)

Как мы видим, сообщение 1 нету в терминале, хотя оно присутствует в коде файла. Это потому, что использование инструкции jmp \_label2 меняет порядок исполнения инструкций, и позволяет пропустить инструкцию.

Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение No 2’, потом ‘Сообщение No 1’ и завершала работу, введём текст из Листинга 8.2, и проверим работу файла:

(рис. 2)

**Шаг 2**

Изменим код, чтобы сообщения выводились в порядке убывания, при этом только добавляя инструкцию jmp:

(рис. 3)

(рис. 4)

**Шаг 3** Cоздим файл lab8-2.asm, и введем в него текст из Листинга 8.3, для того чтобы найти меньшее из 3 чисел, два из которых заданы программой.

Проверим его работу для различных B:

(рис. 5)

**Шаг 4**

Создадим файл листинга для программы из предыдущего файла, для этого используем ключ -l в команде nasm, и откроем его, чтобы ознакомиться с содержимым. Рассмотрим 3 строчки для примера структуры листинга:

(рис. 6)

38 0000013F 8B0D[00000000] mov ecx,[max]  
39 00000145 3B0D[0A000000] cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'  
40 0000014B 7F0C jg fin ; если 'max(A,C)>B', то переход на 'fin',

38, 39, 40 отвечают за номер строки;

00000… - адрес, он отвечает за то, чтобы инструкции выполнялись по порядку;

8B0D, 3B0D, 7F0C - машинный код, инструкция на машинном языке, они отвечают за перемещение переменной max в ecx, сравнение ecx и B, и переход к ‘fin’ если ecx больше B соответственно;

Справа находится исходный текст нашей программы.

Допустим ошибку в нашем коде, удалим один операнд в операции, требующей два, создадим файл листинга, lab8-2.lst и посмотрим как он изменился:

(рис. 7)

Как мы видим, в файле листинга около строчки кода находится предупреждение об ошибке.

# 4 Задание для самостоятельной работы

**Шаг 1**

Создадим файл lab8-3.asm для создания программы для нахождения наименьшей из 3 переменных a,b,c, за основу взяв код lab8-2.asm

Как в листинге 8.2, мы поочерёдно записываем с клавиатуры 3 переменные, переводим их в числа (необязательно)

Далее можно просто оставить код Листинга 8.3, заменив jg (переход если больше) на jl (переход если меньше) (Также См комментарии в коде)

(рис. 8)

(рис. 9)

Проверяем:

(рис. 10)

Создадим файл lab8-4.asm для создания программы нахождения ответа на систему уравнений с использованием 2 переменных a,х за основу снова взяв код lab8-2.asm

Как и указано в комментариях, мы записываем введённые значения а и х в переменные А и Х поочерёдно, преобразуем их в числа для работы с операциями сложения и умножения, и сравниваем Х и 4.

Если Х < 4 то мы переходим к Хmin, и решаем уравнение x + 4, выводим результат, в противном случае умножаем а на х и переходим в конец, где мы выводим сообщение ‘Ответ:’ на экран

Код (рис. 11)

(рис. 12)

Проверим

(рис. 13)

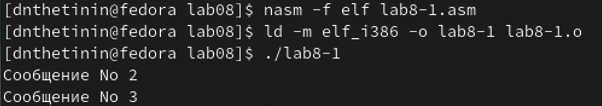


Рис. 1: работа lab8-1.asm

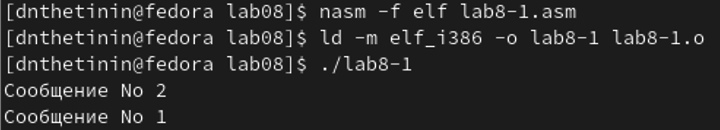


Рис. 2: работа нового lab8-1

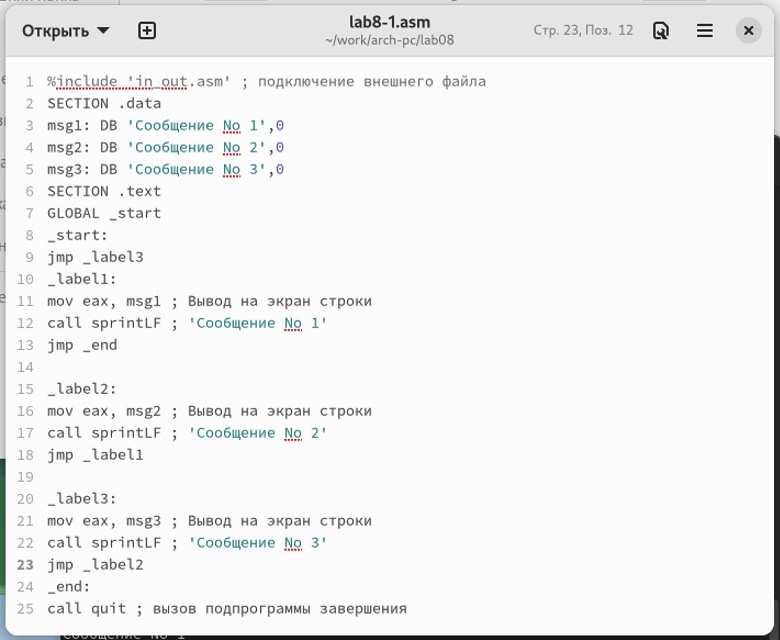


Рис. 3: код lab8-2

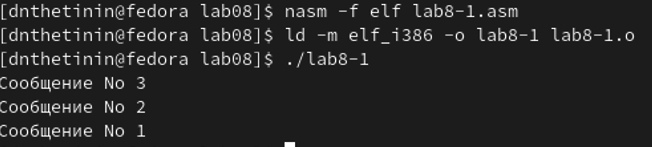


Рис. 4: проверка lab8-2

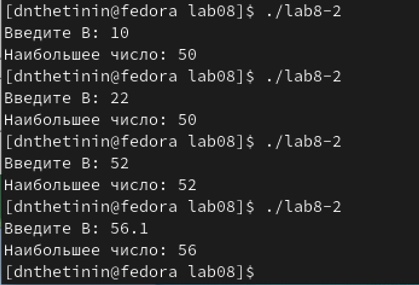


Рис. 5: Наибольшая переменная с разными В

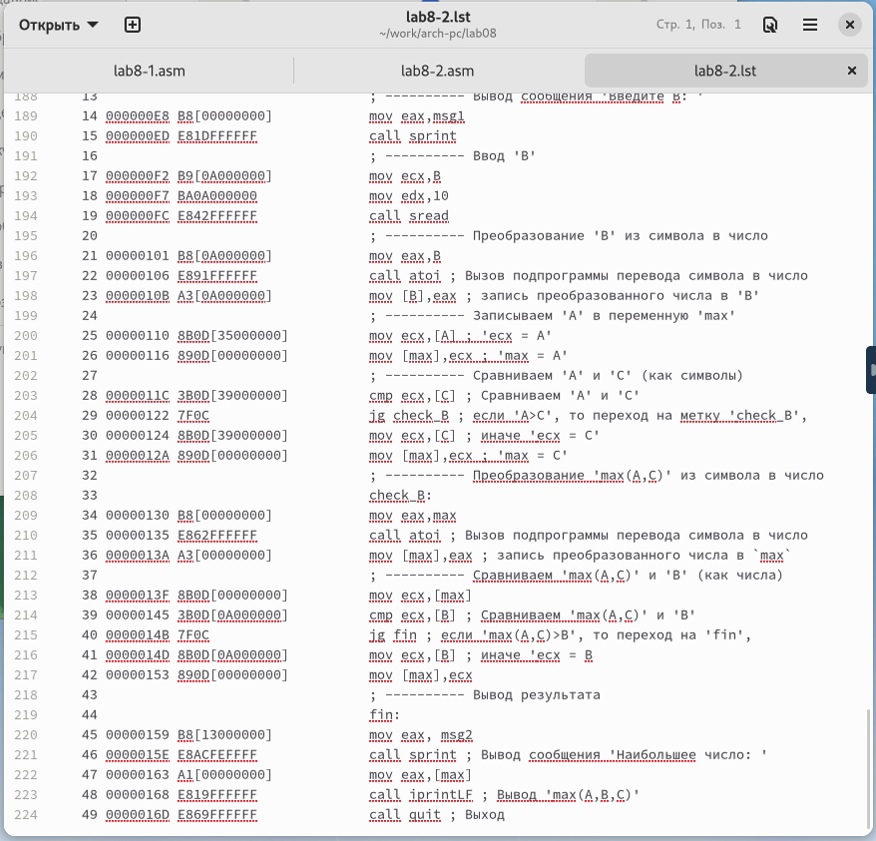


Рис. 6: Файл листинга

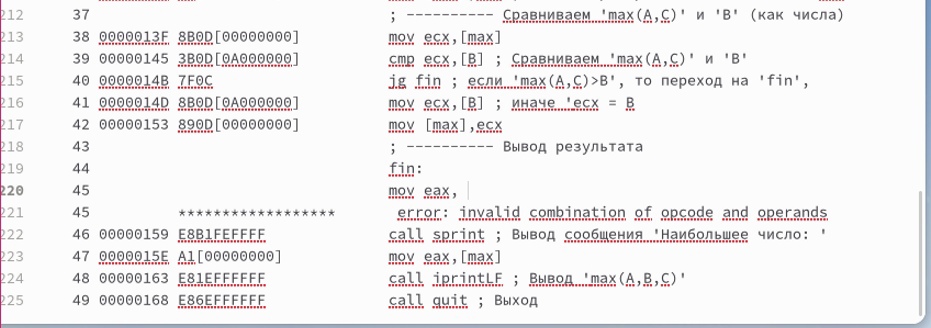


Рис. 7: Ошибка в файле листинга

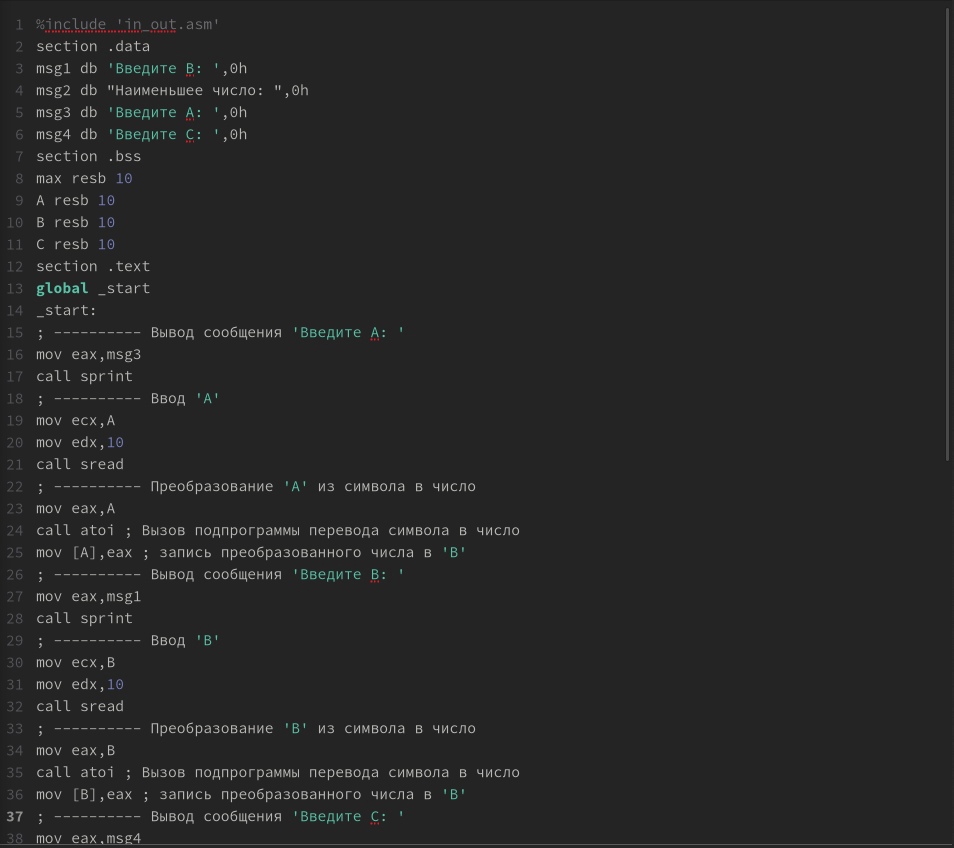


Рис. 8: код lab8-3.asm

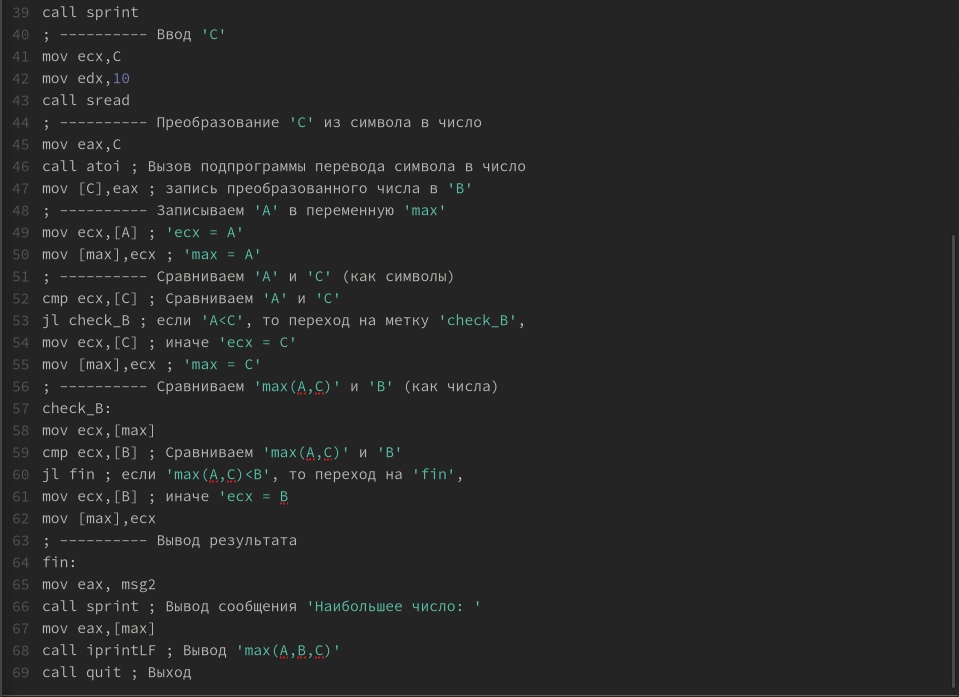


Рис. 9: код lab8-3.asm

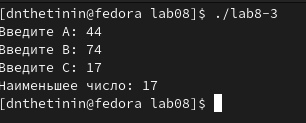


Рис. 10: проверка lab8-3.asm

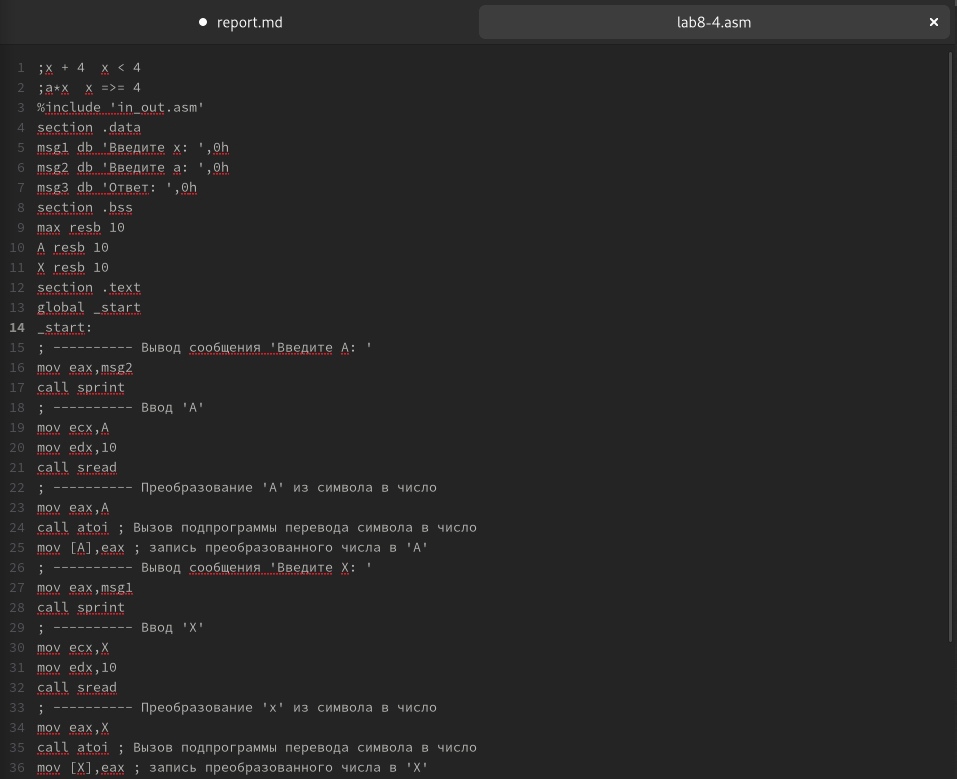


Рис. 11: код lab8-4.asm



Рис. 12: код lab8-4.asm

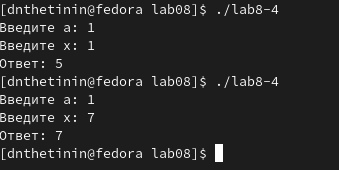


Рис. 13: проверка lab8-4.asm

# 5 Выводы

Я смог написать базовую программу вычисления функции на языке ассемблера NASM