Лабораторная работа №8

Команды безусловного и условного переходов в Nasm. Программирование ветвлений.

Татьяна Алексеевна Коннова, НПИбд-01-22

Содержание

[1 Цель работы 1](#__RefHeading___Toc366_3860812536)

[1.1 Задание 1](#__RefHeading___Toc368_3860812536)

[1.2 Выполнение лабораторной работы 1](#__RefHeading___Toc370_3860812536)

[1.3 Изучение структуры файлы листинга 8](#__RefHeading___Toc372_3860812536)

[1.4 Самостоятельная работа 9](#__RefHeading___Toc374_3860812536)

[2 Выводы 13](#__RefHeading___Toc376_3860812536)

# 1 Цель работы

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

## 1.1 Задание

Программирование ветвлений

## 1.2 Выполнение лабораторной работы

1. Создаем каталог для программам лабораторной работы N 8, переходим в него и создайте файл lab8-1.asm:

mkdir ~/work/arch-pc/lab08

cd ~/work/arch-pc/lab08

touch lab8-1.asm

1. Инструкция jmp в NASM используется для реализации безусловных пере- ходов. Рассмотрим пример программы с использованием инструкции jmp. Введем в файл lab8-1.asm текст программы из листинга 8.1. (рис. 1)



Рис. 1: lab8\_1

Создаем исполняемый файл и запускаем его. Результат работы данной программы будет следующим:

./lab8-1

Сообщение No 2

Сообщение No 3

Таким образом, использование инструкции jmp \_label2 меняет порядок ис- полнения инструкций и позволяет выполнить инструкции начиная с метки \_label2, пропустив вывод первого сообщения. Инструкция jmp позволяет осуществлять переходы не только вперед но и назад. Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение No 2’, потом ‘Сообщение No 1’ и завершала работу. Для этого в текст программы после вывода сообщения No 2 добавим инструкцию jmp с меткой \_label1 (т.е. переход к инструкциям вывода сообщения No 1) и после вывода сообщения No 1 добавим инструкцию jmp с меткой \_end (т.е. переход к инструкции call quit). Изменим текст программы (рис. 2) (рис. 3) (рис. 4)

Рис. 2: lab8_1

Рис. 2: lab8\_1

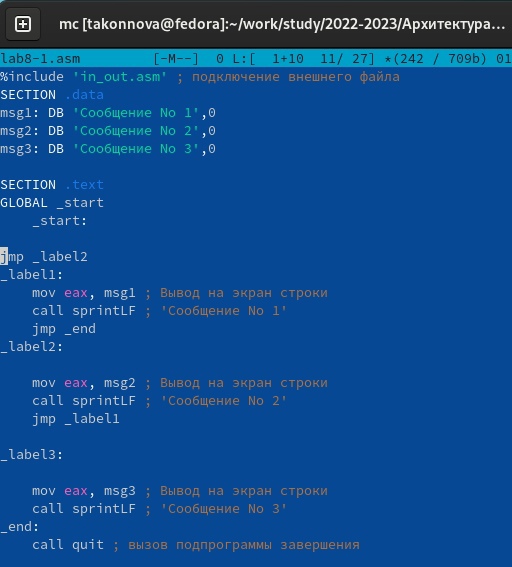


Рис. 3: Change

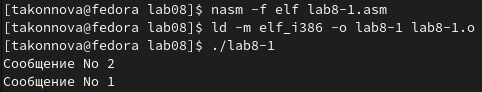


Рис. 4: First change lab8\_1

Создаем исполняемый файл и проверяем его работу.

Изменяем текст программы добавив или изменив инструкции jmp, чтобы вывод программы был следующим: (рис. 5) (рис. 6) ./lab8-1

Сообщение No 3

Сообщение No 2

Сообщение No 1

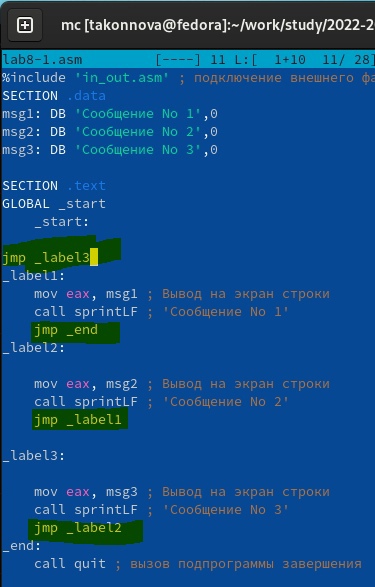


Рис. 5: End change

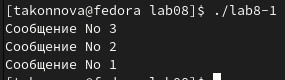


Рис. 6: Output 3 2 1

Использование инструкции jmp приводит к переходу в любом случае. Од- нако, часто при написании программ необходимо использовать условные переходы, т.е. переход должен происходить если выполнено какое-либо условие. В качестве примера рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C. Значения для A и C задаются в программе, значение B вводиться с клавиатуры. Создаем файл lab8-2.asm в каталоге

~/work/arch-pc/lab08

Внимательно изучаем текст программы из листинга 8.3 и введем в lab8-2.asm. (рис. 7) (рис. 8)

Создаем исполняемый файл и проверяем его работу для разных значений B.

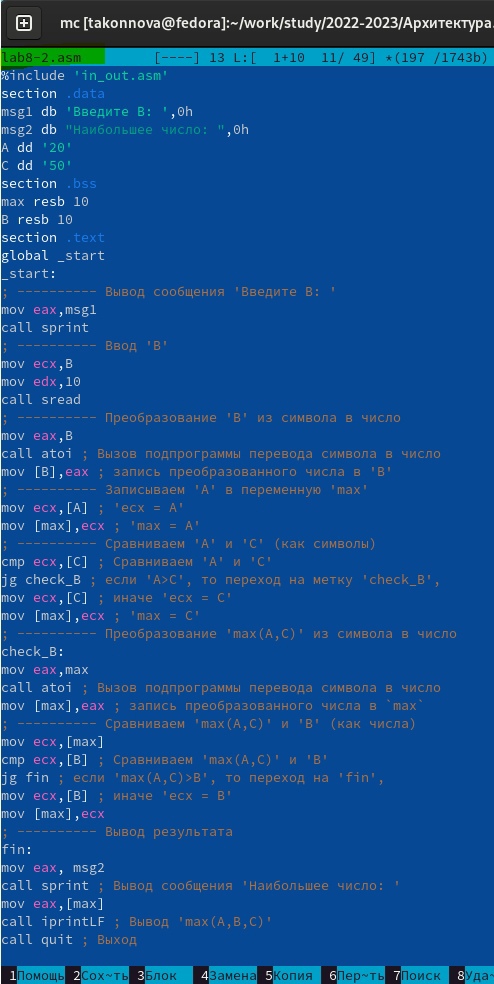


Рис. 7: lab8\_2

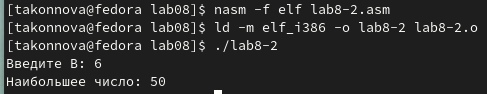


Рис. 8: output lab8\_2

Обращаем внимание, в данном примере переменные A и С сравниваются как символы, а переменная B и максимум из A и С как числа (для этого используется функция atoi преобразования символа в число). Это сделано для демонстрации того, как сравниваются данные. Данную программу можно упростить и сравнивать все 3 переменные как символы (т.е. не использовать функцию atoi). Однако если переменные преобразовать из символов числа, над ними можно корректно проводить арифметические операции.

## 1.3 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. Создаем файл листинга для программы из файла lab8-2.asm

nasm -f elf -l lab8-2.lst lab8-2.asm

Откроем файл листинга lab8-2.lst с помощью любого текстового редактора, например mcedit:

mcedit lab8-2.lst

Внимательно ознакомимся с его форматом и содержимым. Подробно объясним содержимое трёх строк файла листинга по выбору.

строка 51

51 - номер строки

00000033 - адрес

B80A000000 - машинный код

mov eax, 0AH - код программы

строка 37

37 - номер строки

00000135 - адрес

E862FFFFFF - машинный код

call atoi- код программы

строка 53

53 - номер строки

00000039 - адрес

89E0 - машинный код

mov eax, esp - код программы

Откроем файл с программой lab8-2.asm и в любой инструкции с двумя операндами удалить один операнд. Выполним трансляцию с получением файла листинга:

nasm -f elf -l lab8-2.lst lab8-2.asm

Какие выходные файлы создаются в этом случае? Что добавляется в листинге?

Ответ: после удаления операнда появилась ошибка и файлы не формировались, а в листинге ошибка.(рис. 9)

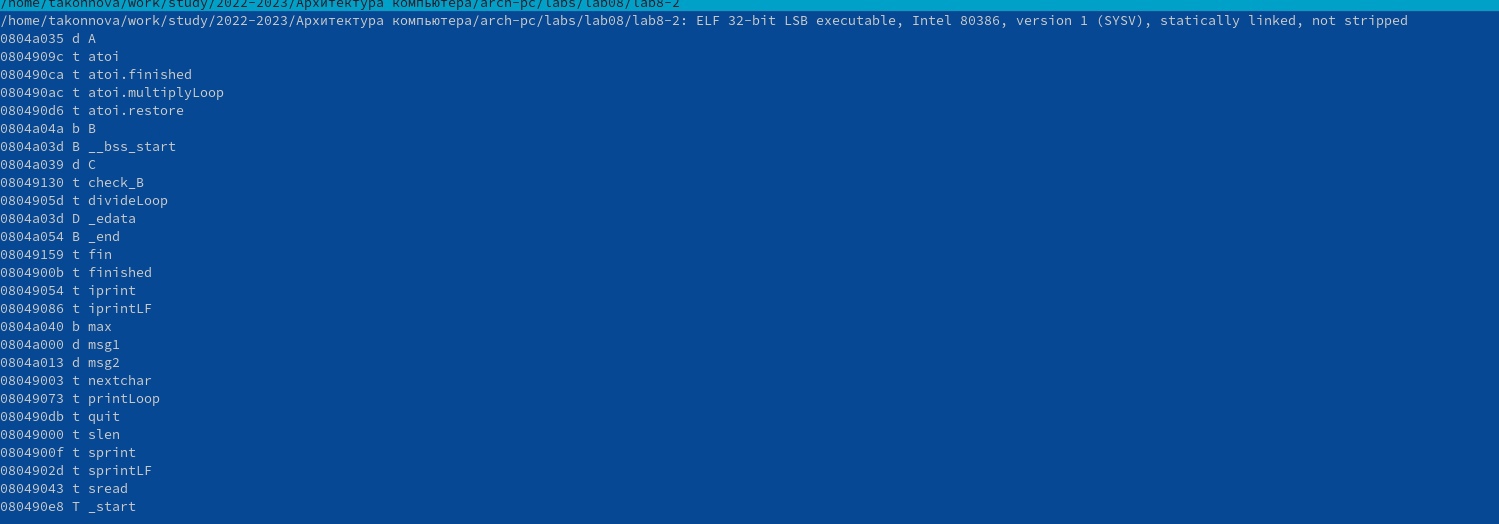


Рис. 9: Error in listing

## 1.4 Самостоятельная работа

1. Напишите программу нахождения наименьшей из 3 целочисленных пере- менных a b и c . Значения переменных выбрать из табл. 8.5 в соответствии с вариантом, полученным при выполнении лабораторной работы No 7. Создайте исполняемый файл и проверьте его работу.
2. Напишите программу, которая для введенных с клавиатуры значений x and a вычисляет значение заданной функции и выводит результат вычислений. Вид функции выбрать из таблицы 8.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы No 7. Создайте исполняемый файл и проверьте его работу для значений x and a

Мой вариант = 15. Напишем программу нахождения наибольшего из переменных a, b, с то есть 32, 6 и 54.

Создадим исполняемый файл и проверим его работу. (рис. 10) (рис. 11)

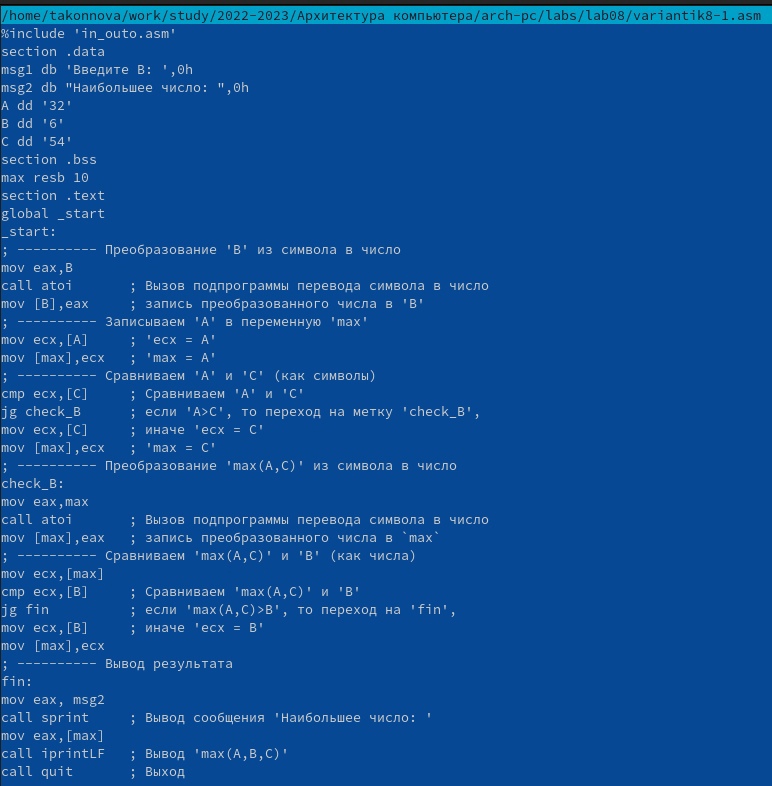


Рис. 10: Номер 1 Самостоятельной работы

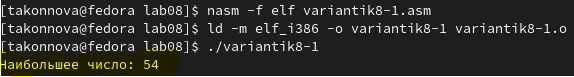


Рис. 11: Вывод программы номера 1

Напишем вторую программу для введенных с клавиатуры значений x и a она вычисляет значение заданной функции 15 и выводит результат вычислений. Создадим исполняемый файл и проверим его работу для значений 2, 3 для первого примера и 4, 2 - для вт. примера (рис. 12) (рис. 13)

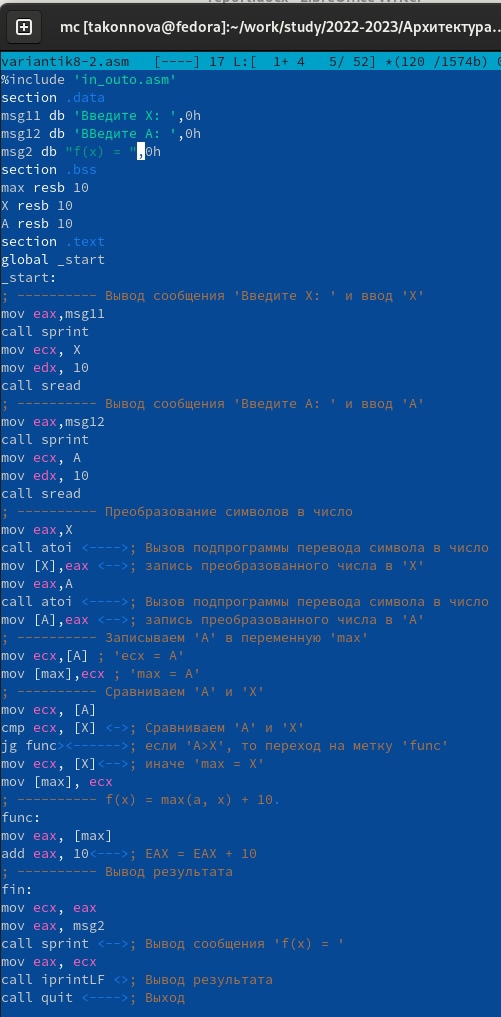


Рис. 12: Программа 2 Самостоятельной работы

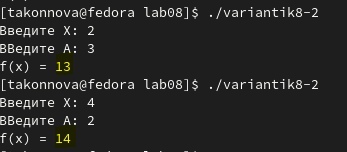


Рис. 13: Вывод программы номер 2 Самостоятельной работы

# 2 Выводы

Мы научились командам условного и безусловного переходов, приобрели навыков написания программ с использованием переходов, познакомились с назначением и структурой файла листинга.