Отчёт по лабораторной работе №8

Команды условного и безусловного переходов в NASM. Программирование ветвлений

Аскеров Александр Эдуардович

Содержание

[1 Цель работы 1](#__RefHeading___Toc295_2808420819)

[2 Выполнение лабораторной работы 1](#__RefHeading___Toc297_2808420819)

[2.1 Реализация переходов в NASM 1](#__RefHeading___Toc299_2808420819)

[2.2 Изучение структуры файлов листинга 9](#__RefHeading___Toc301_2808420819)

[2.3 Задание для самостоятельной работы 12](#__RefHeading___Toc303_2808420819)

[3 Выводы 13](#__RefHeading___Toc305_2808420819)

# 1 Цель работы

Изучить команды условного и безусловного переходов. Приобрести навыки написания программ с использованием переходов. Познакомиться с назначением и структурой файла листинга.

# 2 Выполнение лабораторной работы

## 2.1 Реализация переходов в NASM

1. Создадим каталог для программ лабораторной работы №8, перейдём в него и создадим файл lab8-1.asm.

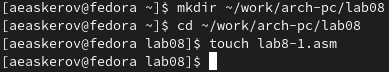


Рис. 1: Создание каталога и файла lab8-1.asm

1. Инструкция jmp в NASM используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции jmp. Введём в файл lab8-1.asm текст программы из листинга 8.1.

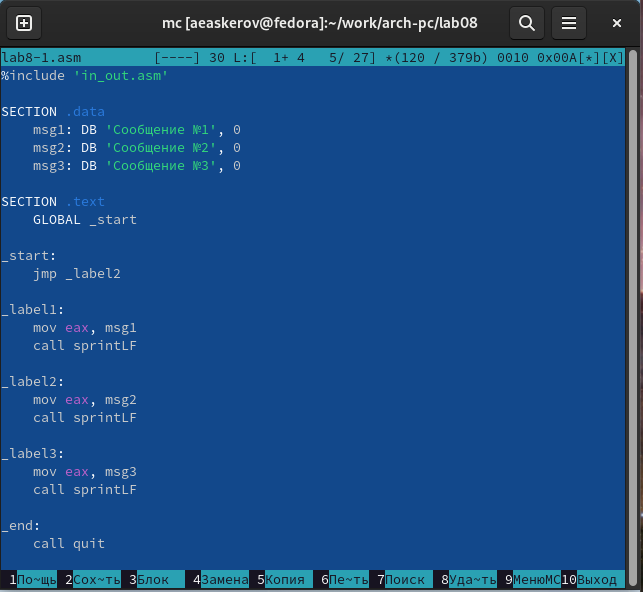


Рис. 2: Программа из листинга 8.1

Создадим исполняемый файл и запустим его. Результат работы данной программы будет следующим.

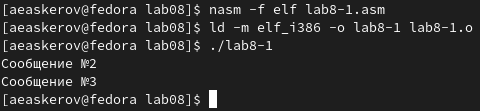


Рис. 3: Результат работы программы lab8-1

Таким образом, использование инструкции jmp \_label2 меняет порядок исполнения инструкций и позволяет выполнить инструкции начиная с метки \_label2, пропустив вывод первого сообщения.

Инструкция jmp позволяет осуществлять переходы не только вперёд, но и назад. Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение №2’, потом ‘Сообщение №1’ и завершала работу. Для этого в текст программы после вывода сообщения №2 добавим инструкцию jmp с меткой \_label1 (т.е. переход к инструкциям вывода сообщения №1) и после вывода сообщения №1 добавим инструкцию jmp с меткой \_end (т.е. переход к инструкции call quit). Изменим текст программы в соответствии с листингом 8.2.

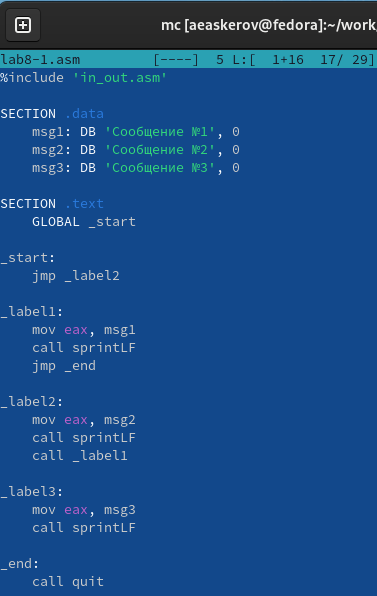


Рис. 4: Изменённый текст программы lab8-1 (листинг 8.2)

Создадим исполняемый файл и проверим его работу.

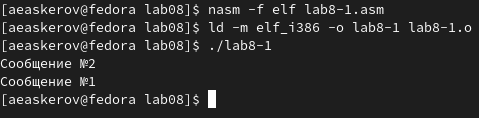


Рис. 5: Результат работы программы lab8-1

Изменим текст программы, добавив или изменив инструкции jmp так, чтобы вывод программы был следующим: сначала Сообщение №3, потом Сообщение №2 и потом Сообщение №1.

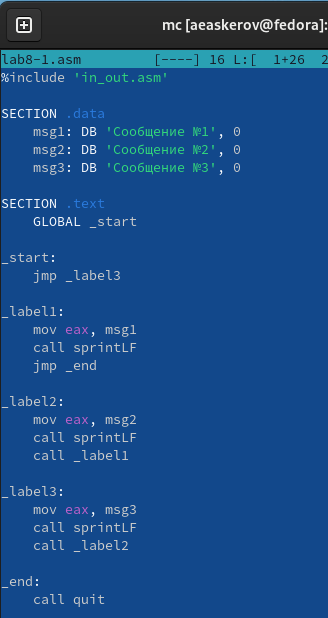


Рис. 6: Изменённый текст программы lab8-1

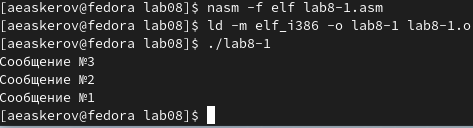


Рис. 7: Результат работы программы lab8-1

1. Использование инструкции jmp приводит к переходу в любом случае. Однако 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 вводится с клавиатуры.

Создадим файл lab8-2.asm в каталоге ~/work/arch-pc/lab08. Внимательно изучим текст программы из листинга 8.3 и введём в lab8-2.asm.

Рис. 8: Создание файла lab8-2.asm

Рис. 8: Создание файла lab8-2.asm

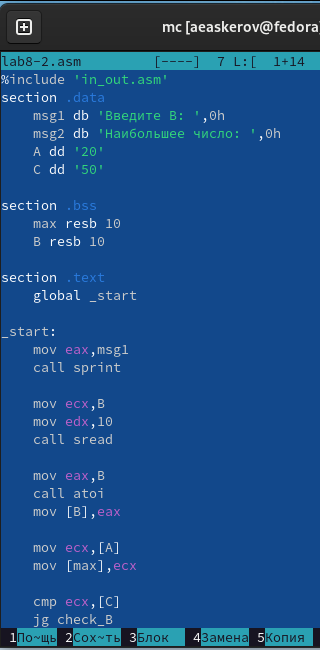


Рис. 9: Программа из листинга 8.3 (фрагмент)

Создадим исполняемый файл и проверим его работу для разных значений B.

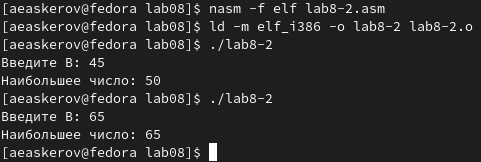


Рис. 10: Результат работы программы lab8-2

Обратим 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 в числа, над ними можно корректно проводить арифметические операции.

## 2.2 Изучение структуры файлов листинга

1. Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ -l и задав имя файла листинга в командной строке. Создадим файл листинга для программы из файла lab8-2.asm.

Рис. 11: Создание файла листинга

Рис. 11: Создание файла листинга

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

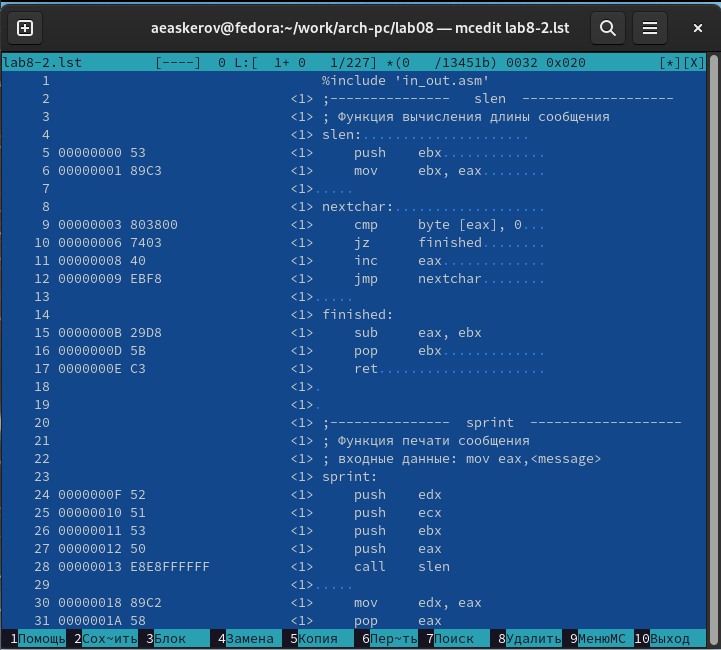


Рис. 12: Содержимое листинга lab8-2.lst

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



Рис. 13: Фрагмент листинга lab8-2.lst

36, 37, 38 – номера строк.

00000130, 00000135, 0000013A – адрес строки.

B8[00000000], E862FFFFFF, A3[00000000] – машинный код.

“mov eax,max”, “call atoi”, “mov [max],eax” – исходный текст программы.

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

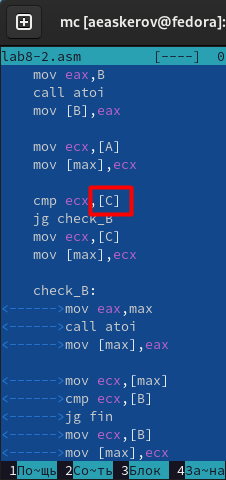


Рис. 14: Удалим операнд, выделенный красным

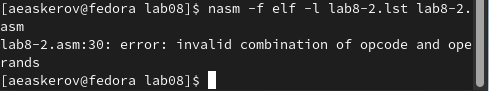


Рис. 15: Выполнение трансляции с получением файла листинга

Мы видим, что в результате удаления операнда вышла ошибка и никакие файлы не сформировались, а в листинге появилось сообщение об ошибке.

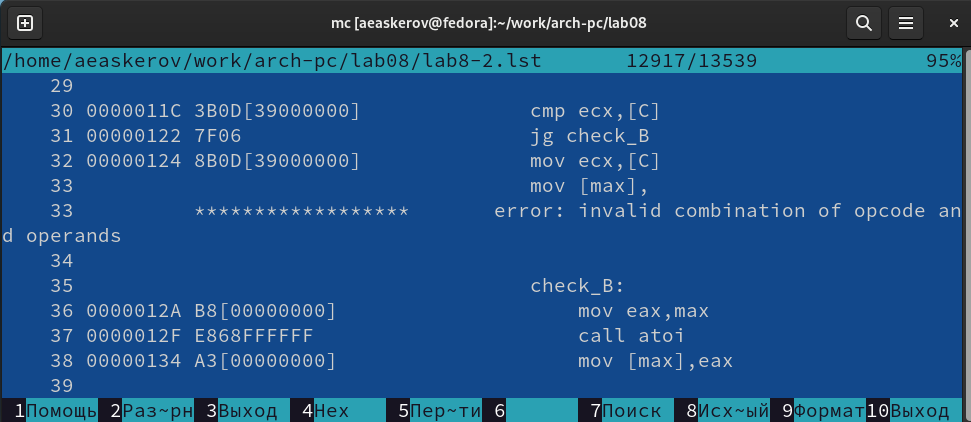


Рис. 16: Сообщение об ошибке в листинге

## 2.3 Задание для самостоятельной работы

1. Напишем программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения переменных выберем из табл. 8.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создадим исполняемый файл и проверим его работу.

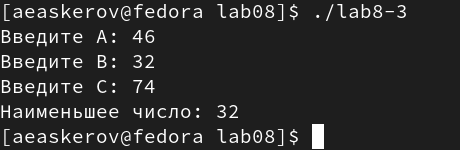


Рис. 17: Результат работы программы lab8-3.asm (вариант 19)

1. Напишем программу, которая для введённых с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Вид функции f(x) выберем из таблицы 8.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создадим исполняемый файл и проверим его работу для значений x и a из 8.6.

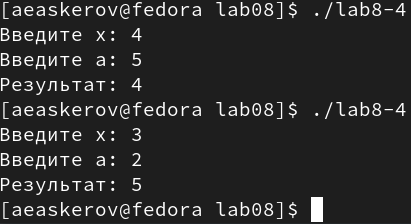


Рис. 18: Результат работы программы lab8-4.asm (вариант 19)

# 3 Выводы

Изучены команды условного и безусловного переходов. Приобретены навыки написания программ с использованием переходов. Ознакомлен с назначением и структурой файла листинга.