Очёт по лабораторной работе № 8

Архитектура Компьютера

Петросян Эмиль Манукович

Содержание

# 1 Цель работы

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

# 2 Задание

1. Реализовать переходы в NASM
2. Изучить структуру файлов листинга
3. Выполнить задание для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Реализация переходов в NASM

1. Создали каталог для программам лабораторной работы № 8, перешли в него и создали файл lab8-1.asm: (рис. 1)

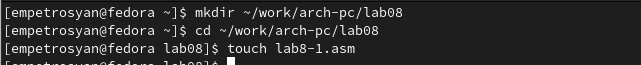


Рис. 1: lab8-1.asm

1. Инструкция jmp в NASM используется для реализации безусловных переходов. Рассмотрели пример программы с использованием инструкции jmp. Ввели в файл lab8-1.asm текст программы из листинга 8.1. (рис. 2)

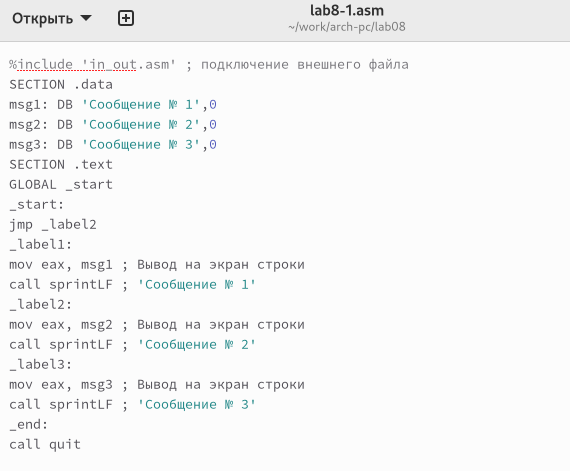


Рис. 2: Текст программы

Создали исполняемый файл и запустили его. Результат работы данной программы следующий: (рис. 3)

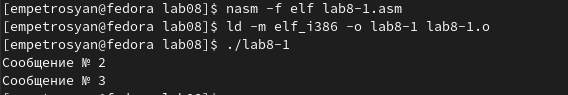


Рис. 3: Результат работы

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 вперед но и назад. Изменили программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого в текст программы после вывода сообщения № 2 добавили инструкцию jmp с меткой \_label1 (т.е. переход к инструкциям вывода сообщения № 1) и после вывода сообщения № 1 добавили инструкцию jmp с меткой \_end (т.е. переход к инструкции call quit). Изменили текст программы в соответствии с листингом 8.2 (рис. 4), (рис. 5)

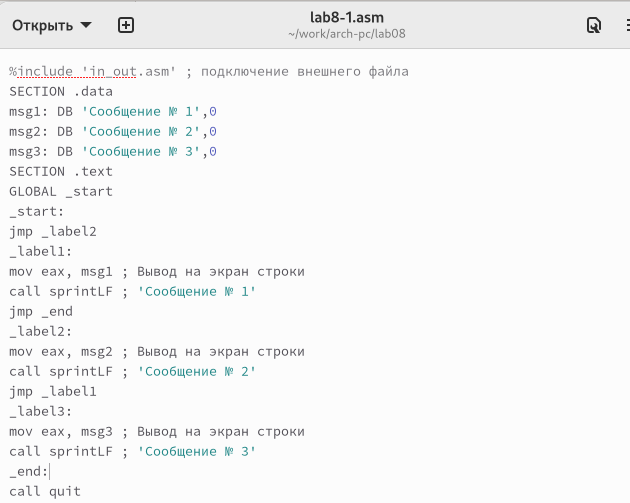


Рис. 4: Использование инструкций

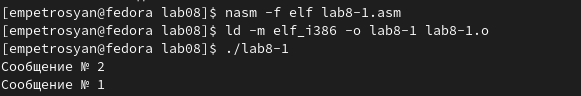


Рис. 5: Текст программы

Измените текст программы добавив или изменив инструкции jmp. (рис. 6), (рис. 7)

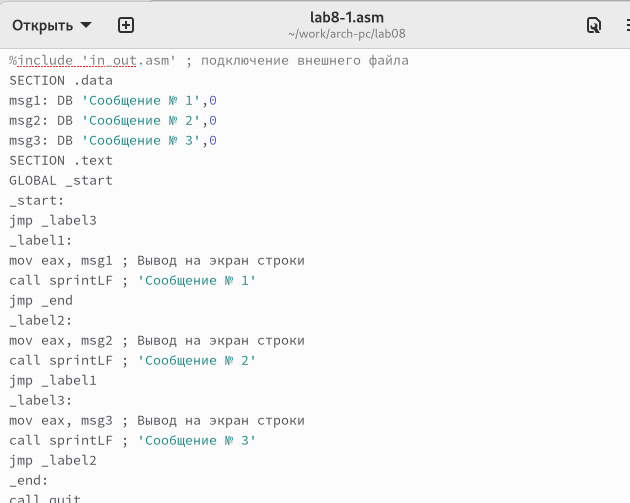


Рис. 6: Инструкции jmp

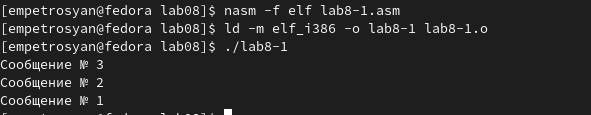


Рис. 7: Исполняемый файл

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 рассмотрели 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. Создали файл lab8-2.asm в каталоге ~/work/arch-pc/lab08. (рис. 8) Внимательно изучили текст программы из листинга 8.3 и введите в lab8-2.asm. (рис. 9)

Рис. 8: lab8-2.asm

Рис. 8: lab8-2.asm

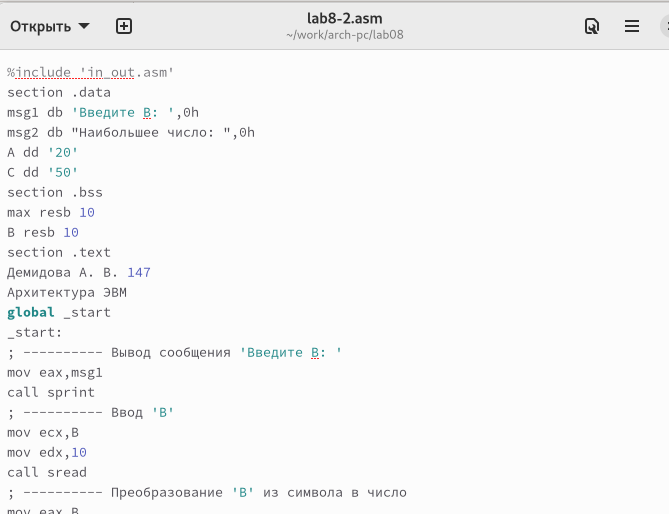


Рис. 9: Текст программы

Создали исполняемый файл и проверили его работу для разных значений B. (рис. 10)

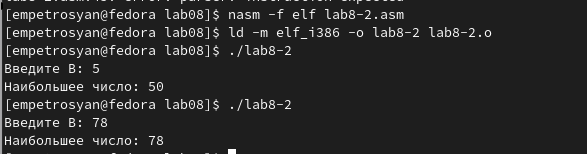


Рис. 10: Исполняемый файл

Обратили 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.

## 3.2 Изучение структуры файлы 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. Создали файл листинга для программы из файла lab8-2.asm. (рис. 11)

Рис. 11: Ключ -l

Рис. 11: Ключ -l

Открыли файл листинга lab8-2.lst с помощью текстового редактора mcedit: (рис. 12), (рис. 13)

Рис. 12: mcedit

Рис. 12: mcedit

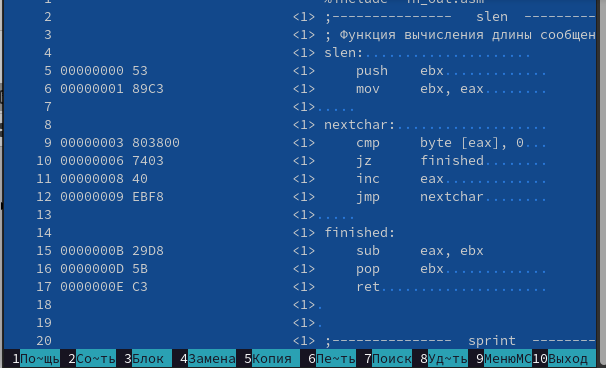


Рис. 13: lab8-2.lst

Внимательно ознакомились с его форматом и содержимым. Содержимое трёх строк файла листинга: 1)45 00000154 B8[13000000] mov eax, msg2 - строка 45, адрес 00000154, B8[13000000] - машинный код, mov eax, msg2 - исходный текст программы 2)46 00000159 E8B1FEFFFF call sprint - строка 46, адрес 00000159, E8B1FEFFFF - машинный код, call sprint - исходный текст программы 3)47 0000015E A1[00000000] mov eax,[max] - строка 47, адрес 0000015E, A1[00000000] - машинный код, mov eax,[max] - исходный текст программы

Открыли файл с программой lab8-2.asm и в инструкции mov с двумя операндами удалить один операнд. (рис. 14) Выполните трансляцию с получением файла листинга: (рис. 15), (рис. 16)

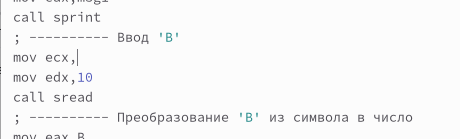


Рис. 14: lab8-2.asm

Рис. 15: mcedit

Рис. 15: mcedit

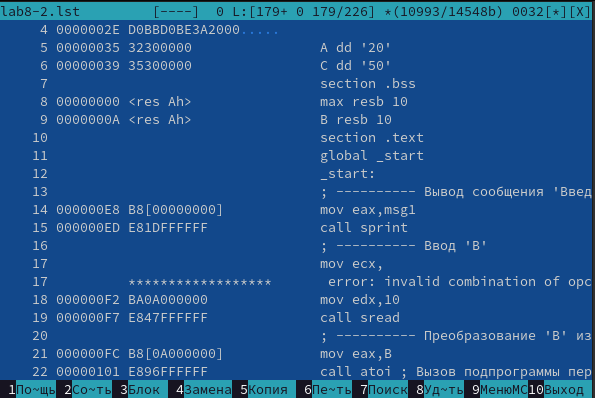


Рис. 16: lab8-2.lst

Создаётся выходной файл lst. В листинге добавляется сообщение об ошибке.

## 3.3 Задание для самостоятельной работы

1. Написали программу нахождения наименьшей из 3 целочисленных переменных a, b и c. (рис. 17) Значения переменных выбрали из таблицы в соответствии с 9 вариантом, полученным при выполнении лабораторной работы № 7. Создали исполняемый файл и проверили его работу. (рис. 18)

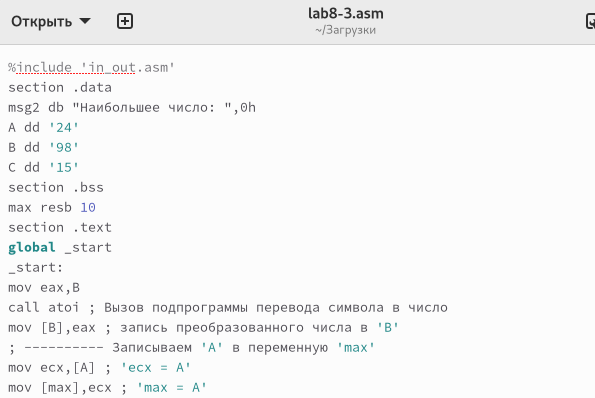


Рис. 17: lab8-3.asm

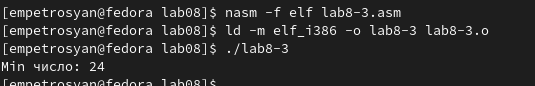


Рис. 18: Исполняемый файл

1. Написали программу, которая для введенных с клавиатуры значений 𝑥 и 𝑎 вычисляет значение заданной функции f(х) и выводит результат вычислений. (рис. 19) Вид функции f(x) выбрали из таблицы вариантов заданий в соответствии с вариантом 9 , полученным при выполнении лабораторной работы № 7. Создали исполняемый файл и проверили его работу для значений х и а. (рис. 20)

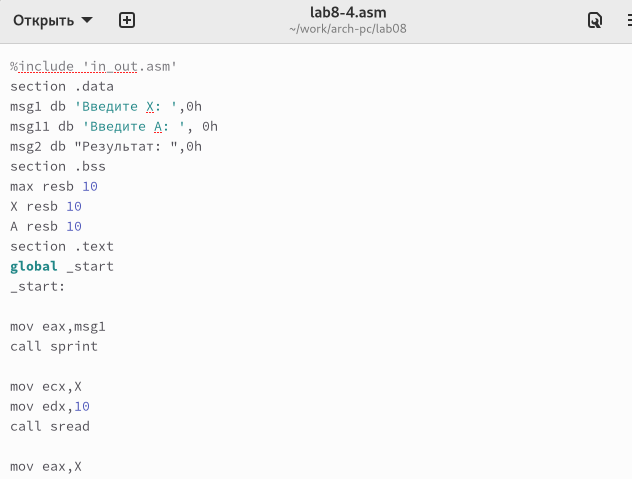


Рис. 19: lab8-4.asm

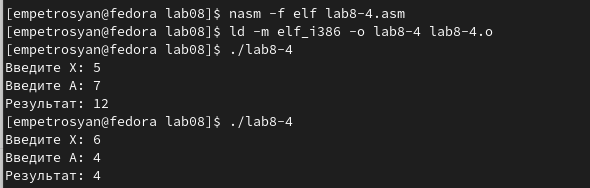


Рис. 20: Исполняемый файл

# 4 Выводы

В ходе выполнения лабораторной работы были изучены команды условного и безусловного переходов. Были приобретены навыки написания программ с использованием переходов. Ознакомились с назначением и структурой файла листинга.