Лабораторная работа №7

Команды безусловного и условного переходов в Nasm. Программирование ветвлений.

Акунаева Антонина Эрдниевна

Содержание

Список иллюстраций

Список таблиц

# Цель работы

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

# Задание

Научиться реализовывать переходы в NASM.  
Изучить структуры файлы листинга.

# Выполнение лабораторной работы

## Реализация переходов в NASM

3.1.1. Создайте каталог для программам лабораторной работы № 7, перейдите в него и создайте файл lab7-1.asm.



Рис. 1: Использование команд mkdir и touch

Создадим каталог lab07 в рабочем каталоге при помощи mkdir, перейдём в него с cd. В новом каталоге создадим NASM-файл lab7-1.asm при помощи touch.

3.1.2. Введите в файл lab7-1.asm текст программы из листинга 7.1. Создайте исполняемый файл и запустите его. Результат работы данной программы будет следующим:

user@dk4n31:~$ ./lab7-1

Сообщение № 2

Сообщение № 3

user@dk4n31:~$

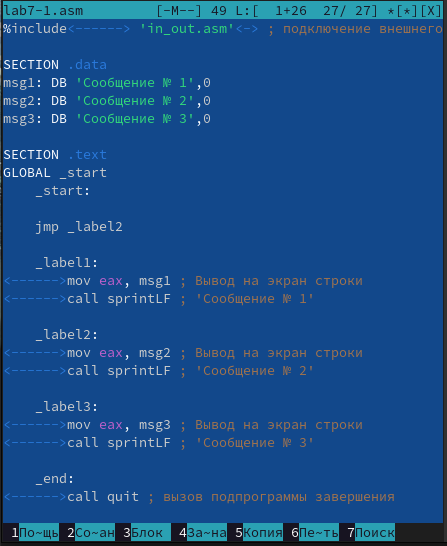


Рис. 2: mcedit: lab7-1.asm

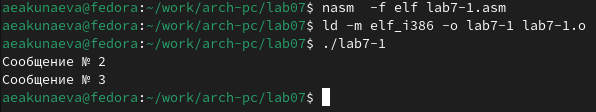


Рис. 3: Трансляция, компоновка и запуск исполняемого файла lab7-1

Откроем файл lab7-1.asm в mcedit. Скопируем текст листинга 7.1 в файл и сохраним. Затем оттранслируем, скомпонуем и запустим исполняемый файл lab7-1. Результат совпадает с предложенным в примере, потому что, несмотря на наличие в листинге сообщения 1, команда jmp \_label2 пропускает первое и переходит сразу ко второму сообщению.

Измените текст программы в соответствии с листингом 7.2. Создайте исполняемый файл и проверьте его работу.

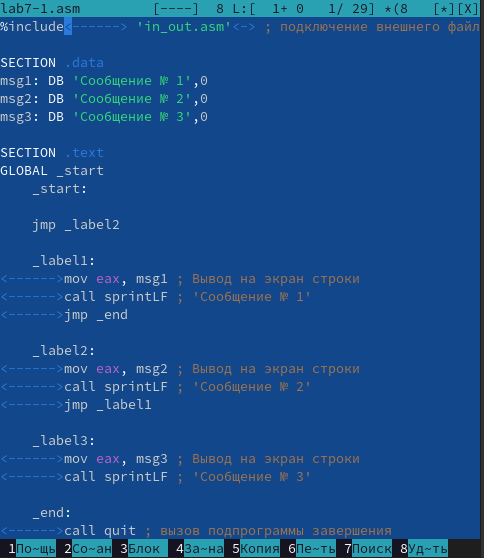


Рис. 4: mcedit: изменённый lab7-1.asm

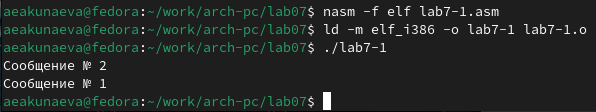


Рис. 5: lab7-1: изменение вывода сообщений

Скопируем текст листинга 7.2 в файл и сохраним. Затем оттранслируем, скомпонуем и запустим исполняемый файл lab7-1. Как и до этого, программа начинает с сообщения 2 из-за команды jmp \_label2, но в секции с сообщением 2 есть команда jmp \_label1, потому выводится первое сообщение, а затем осуществляется переход в jmp \_end, и программа завершает работу.

Измените текст программы добавив или изменив инструкции jmp, чтобы вывод программы был следующим:

user@dk4n31:~$ ./lab7-1

Сообщение № 3

Сообщение № 2

Сообщение № 1

user@dk4n31:~$

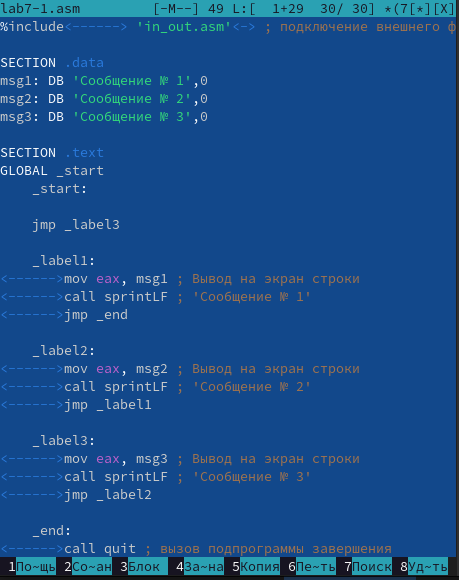


Рис. 6: mcedit: новые изменения lab7-1.asm

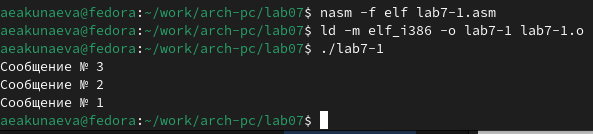


Рис. 7: lab7-1: изменение вывода сообщений: все три

Изменим первый jmp на jmp \_label3, тогда начинаем с третьего сообщения, в нём запишем команду jmp \_label2, выводится второе, оставляем остальное, как есть, тогда следующим выводится сообщение 1 и программа завершает работу.

3.1.3. Создайте файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучите текст программы из листинга 7.3 и введите в lab7-2.asm. Создайте исполняемый файл и проверьте его работу для разных значений B.

Рис. 8: Использование touch: lab7-2.asm

Рис. 8: Использование touch: lab7-2.asm

Создадим файл lab7-2.asm при помощи touch в текущей директории.

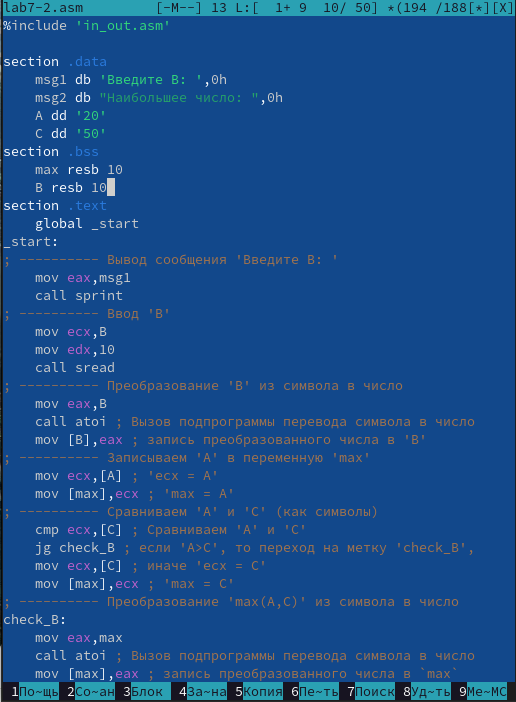


Рис. 9: mcedit: lab7-1.asm. часть 1

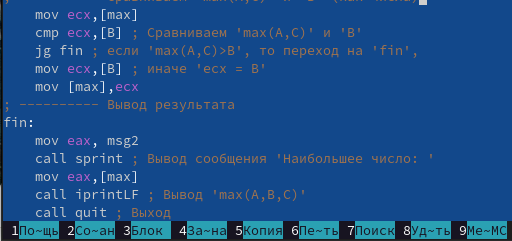


Рис. 10: mcedit: lab7-1.asm. часть 2

Скопируем текст из листинга 7.3 в файл lab7-2.asm, открыв его в mcedit.

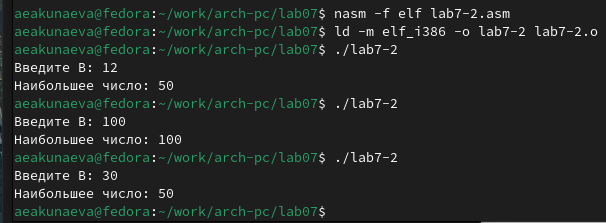


Рис. 11: Трансляция, компоновка и запуск исполняемого файла lab7-2

Запустм исполняемый файл lab7-2 несколько раз и проверим разные знаачения B. Так как по умолчанию А = 20, С = 50, то первое максимальное значение будет 50 (при В = 12), второе - 100 (В = 100), третье - 50 (В = 30).

## Изучение структуры файлы листинга

3.2.1. Создайте файл листинга для программы из файла lab7-2.asm. Откройте файл листинга lab7-2.lst с помощью любого текстового редактора, например mcedit. Внимательно ознакомиться с его форматом и содержимым. Подробно объяснить содержимое трёх строк файла листинга по выбору.

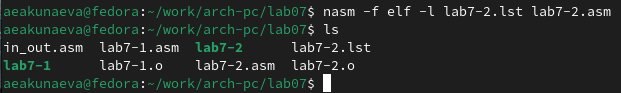


Рис. 12: Создание листинга lab7-2.asm

Создадим листинг для lab7-2.asm, указав ключ -l и задав название файла листинга lab7-2.lst.

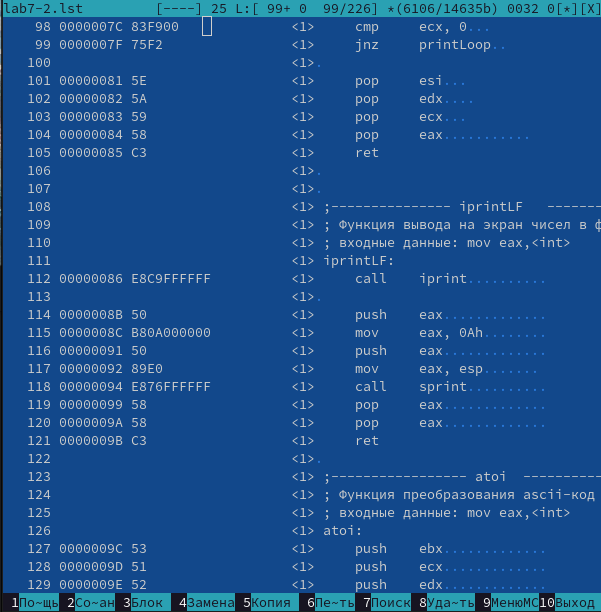


Рис. 13: mcedit: листинг lab7-2.asm

Откроем листинг в текстовом редакторе при помощи mcedit. Изучим содержимое: в листинге указан наш код программы и пояснение слева в машинном коде.

Строка 115: адрес в сегменте кода 0000008С, машинный код В80А000000, команда из текста файла .asm mov eax, 0Ah - присвоение eax значения 0Ah по таблице ASCII.

Строка 116: адрес в сегменте кода 00000091, машинный код 50, команда из текста файла .asm push eax - позволяет сохранить начальные данные регистров при старте программы с последующим восстановлением.

Строка 117: адрес в сегменте кода 00000092, машинный код 89Е0, команда из текста файла .asm mov eax,esp - присвоение переменной eax значения из esp.

Откройте файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалить один операнд. Выполните трансляцию с получением файла листинга. Какие выходные файлы создаются в этом случае? Что добавляется в листинге?

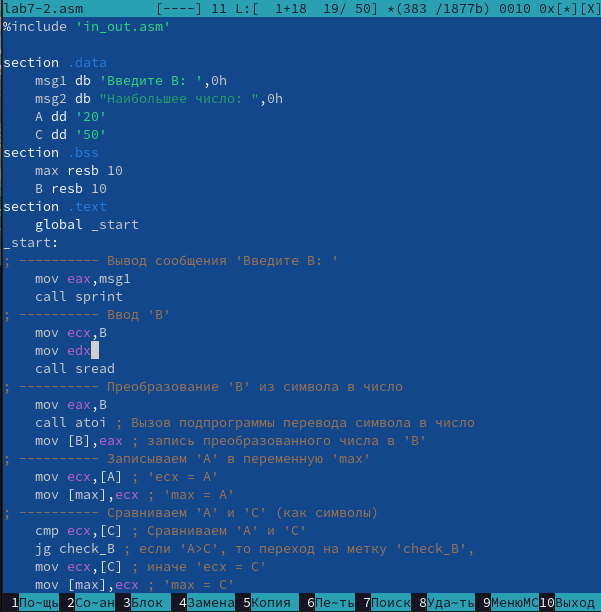


Рис. 14: mcedit: файл lab7-2.asm. Удаление операнда

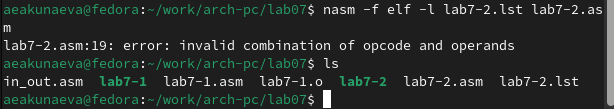


Рис. 15: Неудачная трансляция файла lab7-2.asm

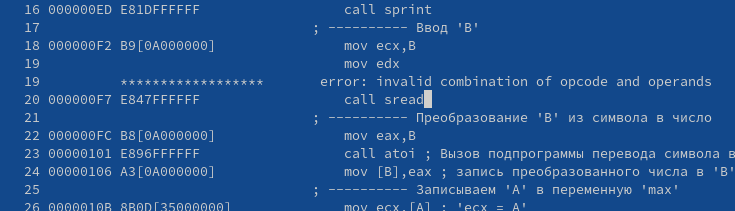


Рис. 16: mcedit: листинг lab7-2.lst с ошибкой

Удалим операнд 10 в mov edx, 10. Попробуем оттранслировать файл, но отобразится ошибка, т.к. не хватает операнда. Тем не менее листинг создан, откроем его и просмотрим. В строке 19 листинга появляется текст ошибки, которая выводится на экран.

# Описание результатов выполнения заданий для самостоятельной работы

4.1. 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.

Вариант 13 из предыдущей лабораторной. Значения 84, 32, 77.

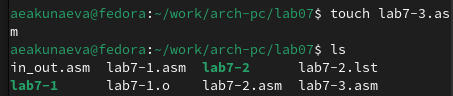


Рис. 17: touch lab7-3.asm

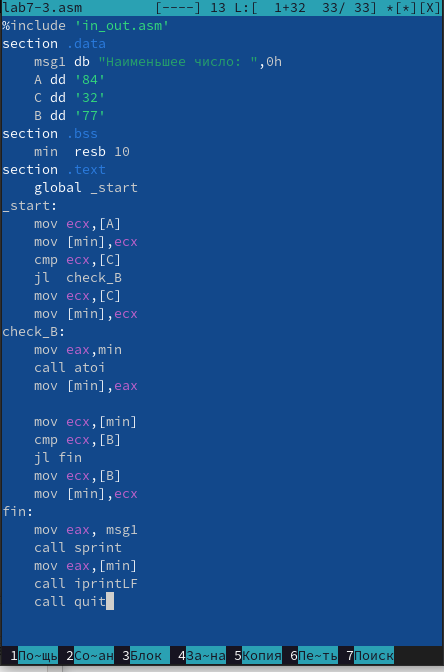


Рис. 18: mcedit: lab7-3.asm

Напишем программу нахождения минимального значения из 3-ёх переменных.

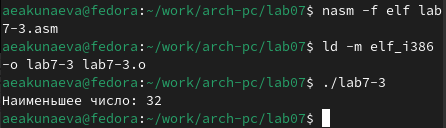


Рис. 19: Трансляция, компоновка и запуск исполняемого файла lab7-3

Создадим и запустим исполняемый файл. Программа успешно выводит наименьшее значение 32.

4.2. 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 №6. Создайте исполняемый файл и проверьте его работу для значений x и a из 7.6.

Вариант 13 из предыдущей лабораторной.



Рис. 20: Вариант 13

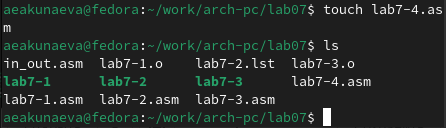


Рис. 21: touch lab7-4.asm



Рис. 22: mcedit: lab7-4.asm. part 1

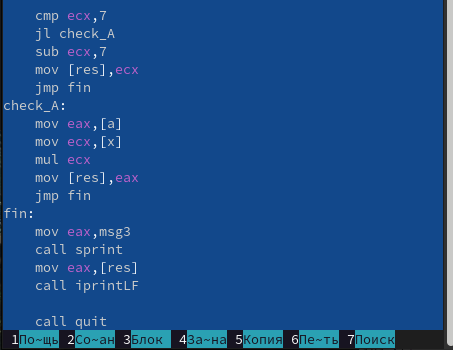


Рис. 23: mcedit: lab7-4.asm. part 2

Напишем программу нахождения значения функции для вводимых с клавиатуры переменных.

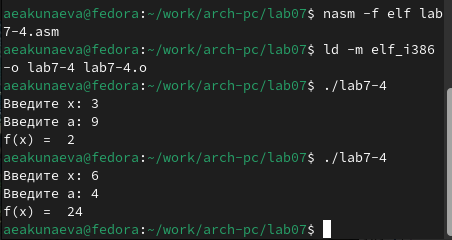


Рис. 24: Трансляция, компоновка и запуск исполняемого файла lab7-4

Создадим и запустим исполняемый файл. Введём значения переменных с клавиатуры. Программа успешно выводит значение функции для заданных x и a.

# Выводы

Я изучила команды условного и безусловного переходов, приобрела навыки написания программ с использованием переходов, познакомилась с назначением и структурой файла листинга.