Команды безусловного и условного переходов в Nasm. Программирование ветвлений.

Лабораторная работа №7

Приходько Иван Иванович

Содержание

# 1 Цель работы

Понять принцип работы условных и безусловных переходов в Ассемблере и научиться писать программы с командами, отвечающими за переходы. Научиться работать с файлами листинга и уметь их читать.

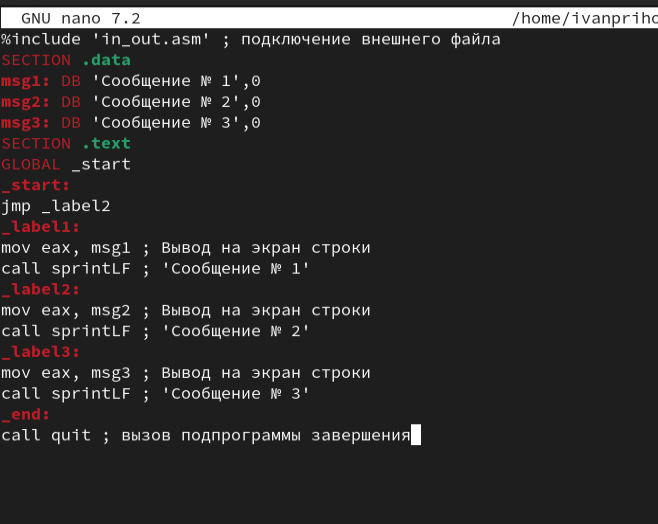
# 2 Выполнение лабораторной работы

Для начала выполнения лабораторной работы необходимо создать файл lab7-1.asm (рис. 2.1).

Создание lab7-1.asm

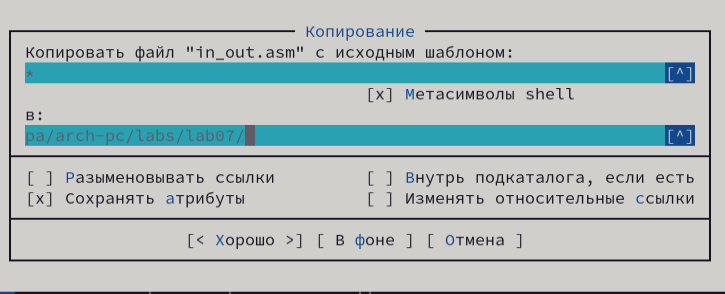
Создание lab7-1.asm

Вставим в него код из листинга 7.1 (рис. 2.2).



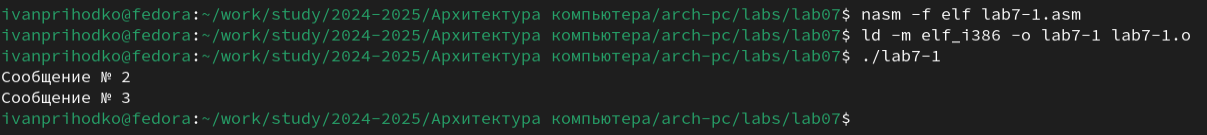
Вставка кода из файла листинга 7.1

Скопируем файл in\_out.asm из рабочей директории прошлой лабораторной работы (рис. 2.3).



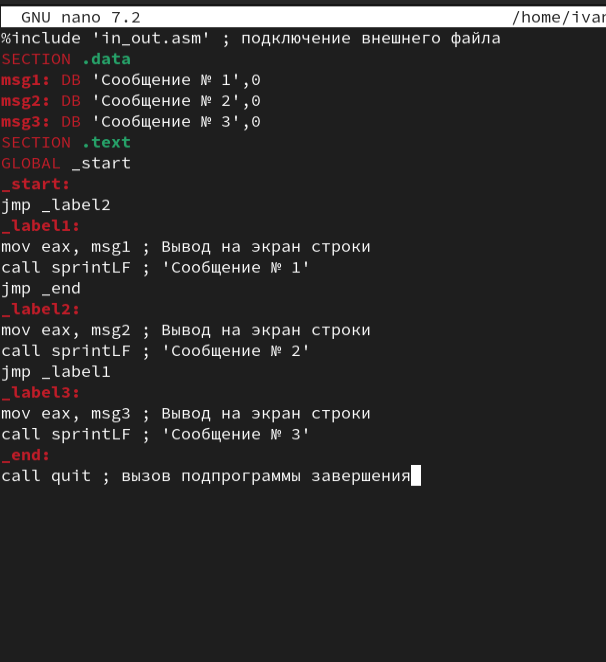
Копирование in\_out.asm

Соберем и запустим (рис. 2.4).



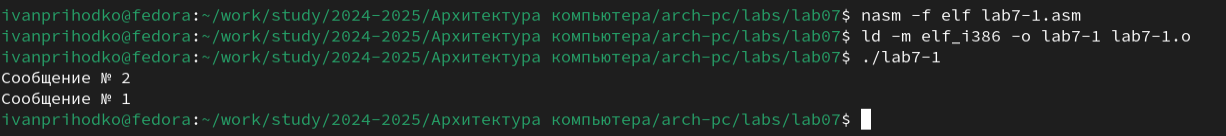
Сборка и запуск lab7-1.asm

Изменим файл lab7-1.asm согласно листингу 7.2 (рис. 2.5).



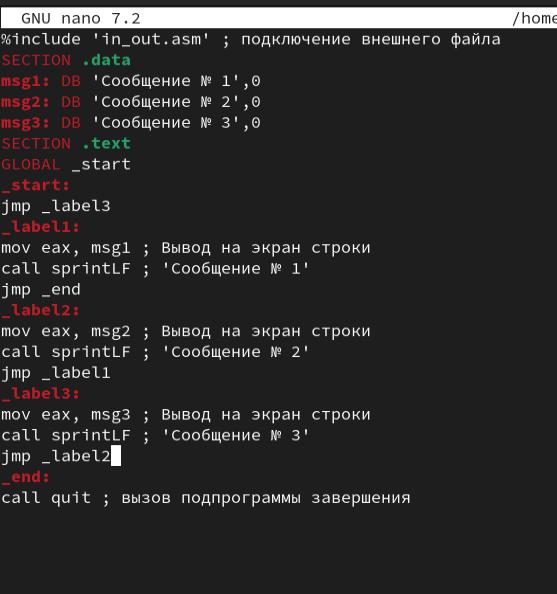
Изменение файла lab7-1.asm согласно листингу 7.2

Повторно соберем и запустим (рис. 2.6).



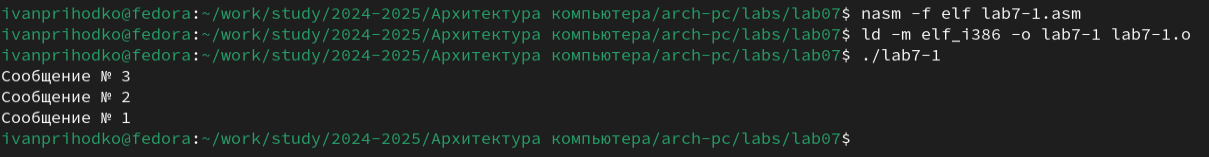
Повторная сборка и запуск lab7-1.asm

Теперь сделаем так, чтобы код выводил сообщения в обратном порядке, для этого внесём в код следующие изменения (рис. 2.7).



Редактирование lab7-1.asm

Запустим (рис. 2.8).



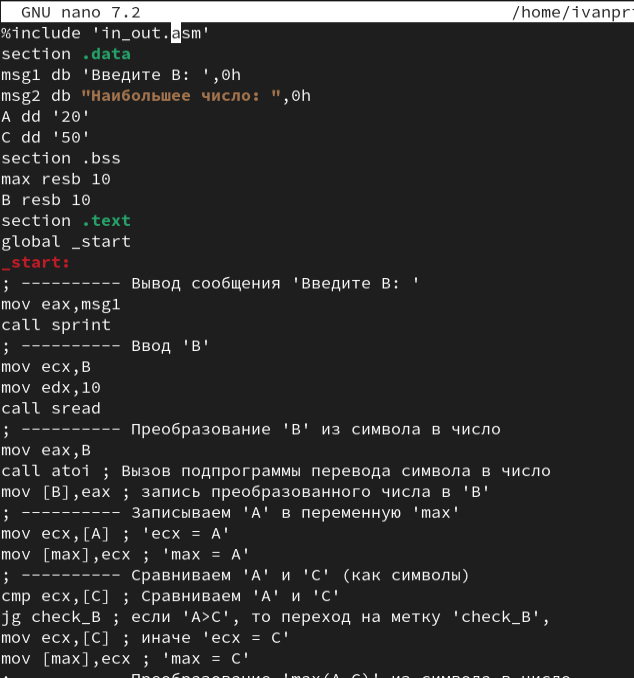
Повторная сборка и запуск lab7-1.asm

Теперь создадим файл lab7-2.asm (рис. 2.9).

Создание lab7-2.asm

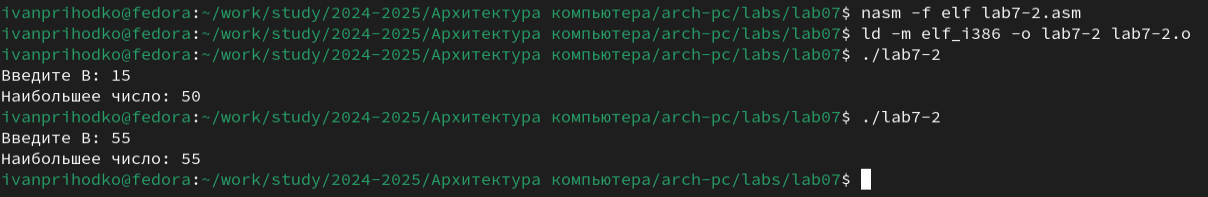
Создание lab7-2.asm

Вставим код из листинга 7.3 в файл lab7-2.asm (рис. 2.10).



Запись кода из листинга 7.3 в файл lab7-2.asm

Соберем и запустим (рис. 2.11).



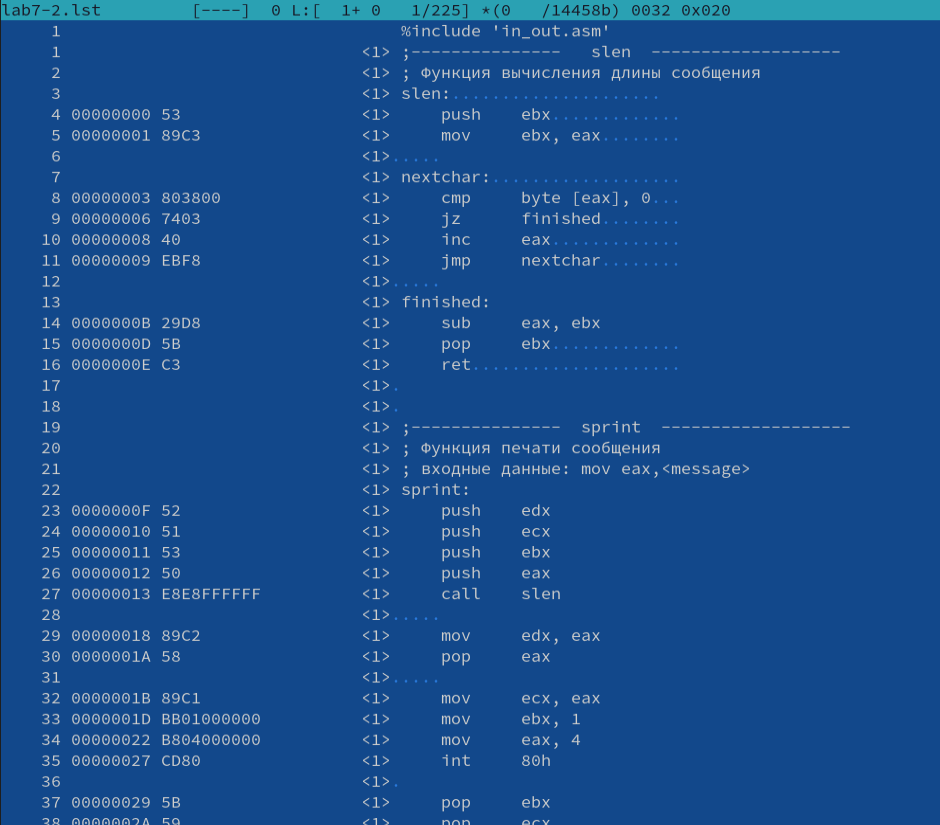
Сборка и запуск lab7-2.asm

Теперь попробуем создать файл листинга при сборке файла lab7-2.asm (рис. 2.12).

Создание файла листинга из lab7-2.asm

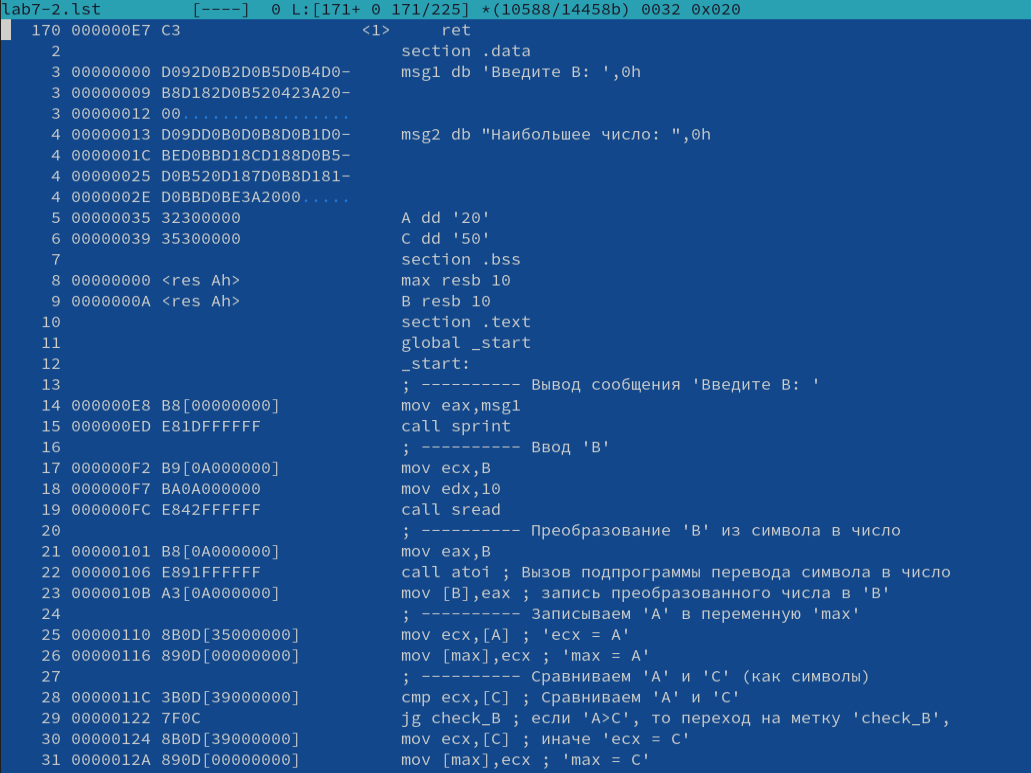
Создание файла листинга из lab7-2.asm

Открыв его, мы видим следующую картину (рис. 2.13).



Вид файла lab7-2.lst

Наша программа находится ниже (рис. 2.14).

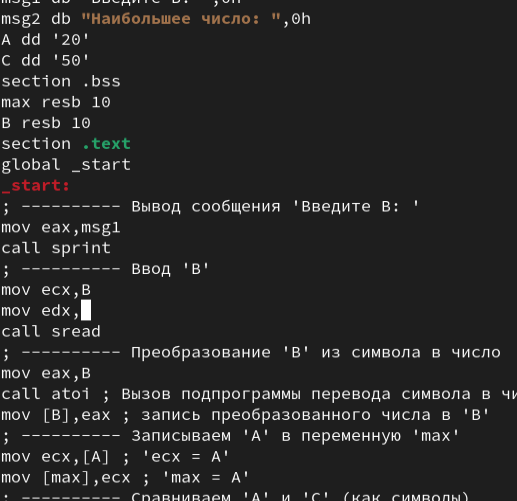


Нахождение программы в файле листинга

Разберём несколько строк файла листинга:

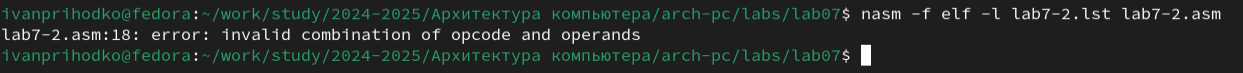
1. Строка под номером 14 перемещает содержимое msg1 в регистр eax. Адрес указывается сразу после номера. Следом идёт машинный код, который представляет собой исходную ассемблированную строку в виде шестнадцатиричной системы. Далее идёт исходный код
2. 15-ая строка отвечает за вызов функции sprint. Она также имеет адрес и машинный код
3. Строка 17 отвечает за запись переменной B в регистр ecx. Как видно, все строки имеют номер, адрес, машинный код и исходный код.

Теперь попробуем намеренно допустить ошибку в нашем коде, убрав у команды mov 1 операнд (рис. 2.15).



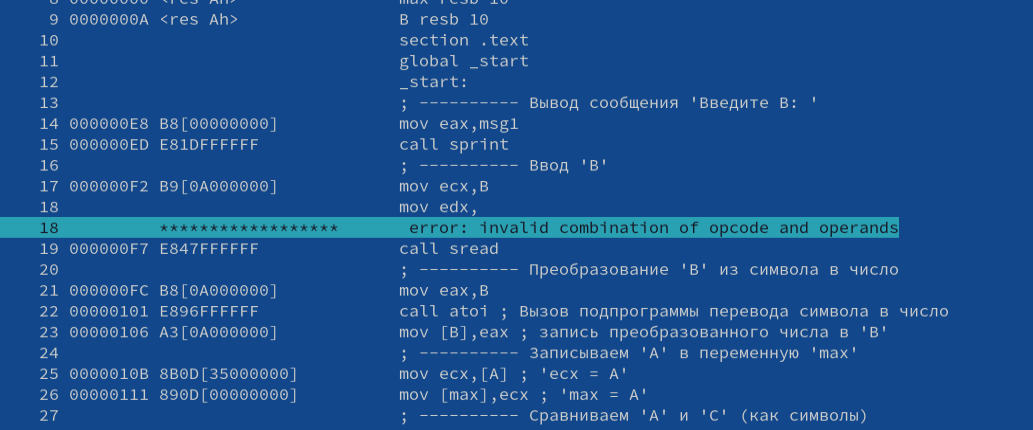
Допуск ошибки в lab7-2.asm

И попробуем собрать файл с ошибкой, генерируя файл листинга (рис. 2.16).



Сборка файла с ошибкой

Посмотрим как выглядит ошибка в файле листинга (рис. 2.17).



Ошибка в lab7-2.lst

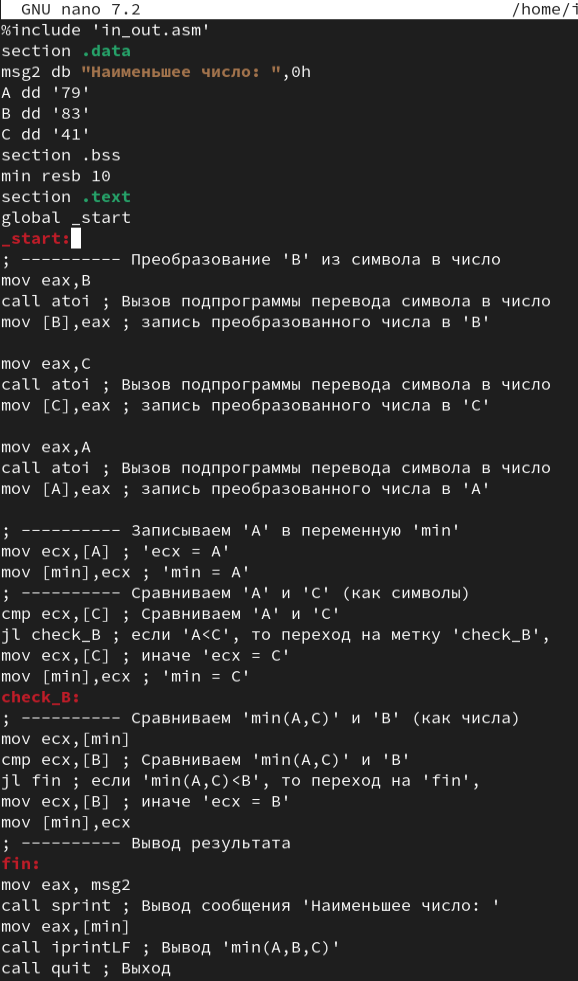
# 3 Задания для самостоятельной работу

Создадим файл для выполнения самостоятельной работы (рис. 3.1).

Создание файла task1v6.asm

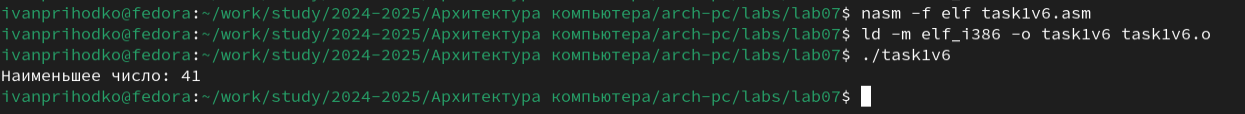
Создание файла task1v6.asm

Напишем код для первого задания (рис. 3.2).



Код task1v6.asm

Соберем и запустим его (рис. 3.3).



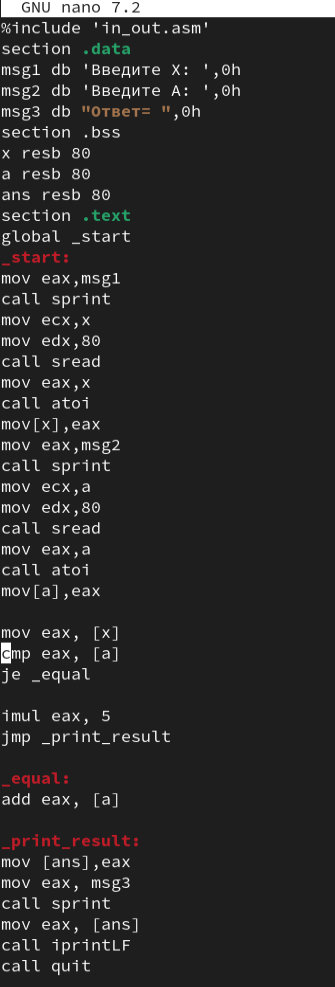
Сборка и запуск task1v6.asm

Теперь создадим второй файл (рис. 3.4).

Создание файла task2v6.asm

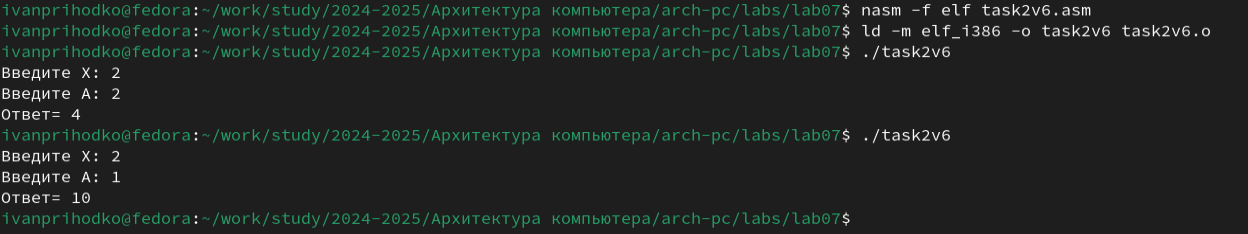
Создание файла task2v6.asm

Мой код получился таким (рис. 3.5).



Код task2v6.asm

Соберем и проведем тесты (рис. 3.6).



Сборка и запуск task2v6.asm

# 4 Выводы

В результате работы над лабораторной работой были написаны программы, которые используют команды условных и безусловных переходов, были получены навыки работы с этими командами, а также были созданы и успешно прочитаны листинги для некоторых из программ.