Лабораторная работа №7

Команды безусловного и условного переходов в Nasm. Программирование ветвлений

Андрюшин Никита Сергеевич

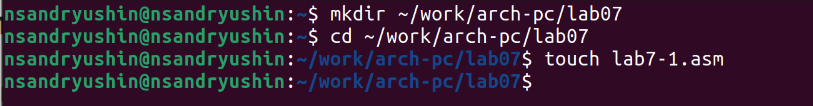
Содержание

# 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 рабочую папку lab07 и файл lab7-1.asm (рис. 2.1):



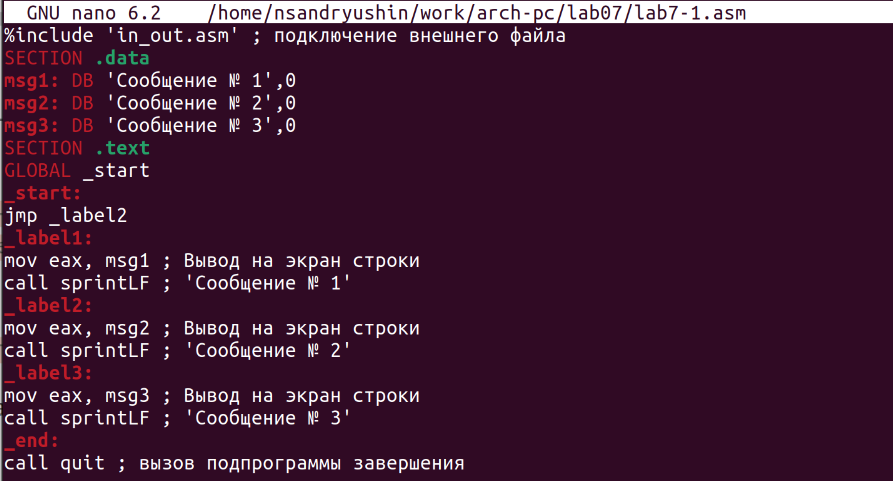
Создание рабочей директории и файла lab7-1.asm

После чего, для удобства, запустить Midnight commander (рис. 2.2):

Запуск Midnight commander

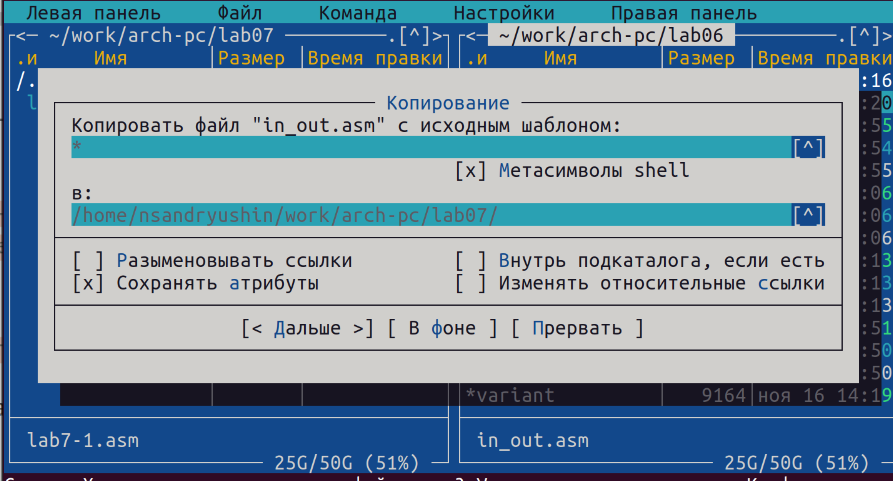
Запуск Midnight commander

Вставим код в файл lab7-1.asm из файла листинга (рис. 2.3):



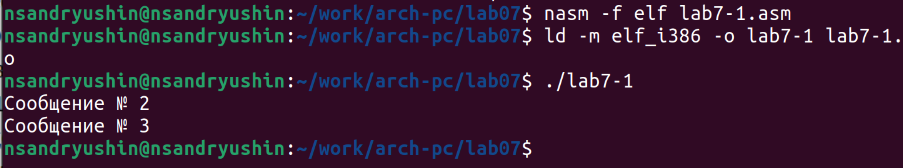
Вставка кода из файла листинга 7.1

Теперь скопируем файл in\_out.asm из рабочей директории прошлой лабораторной работы (рис. 2.4):



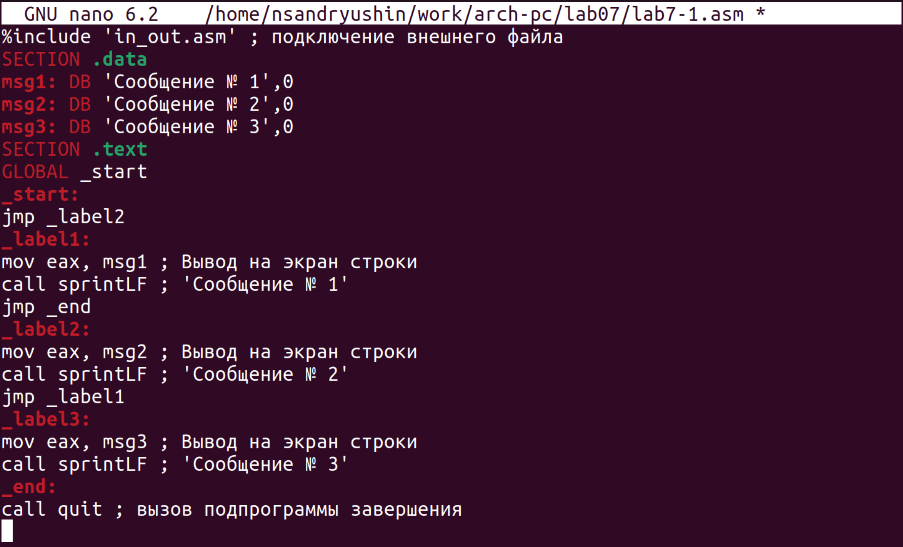
Копирование файла in\_out.asm в рабочую директорию

Теперь соберём программу из файла lab7-1.asm и запустим её (рис. 2.5):



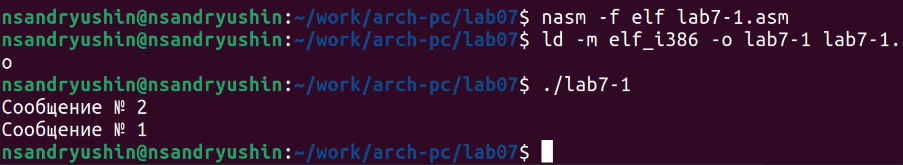
Сборка программы из файла lab7-1.asm и её запуск

Изменим файл lab7-1.asm согласно листингу 7.2 (рис. 2.6):



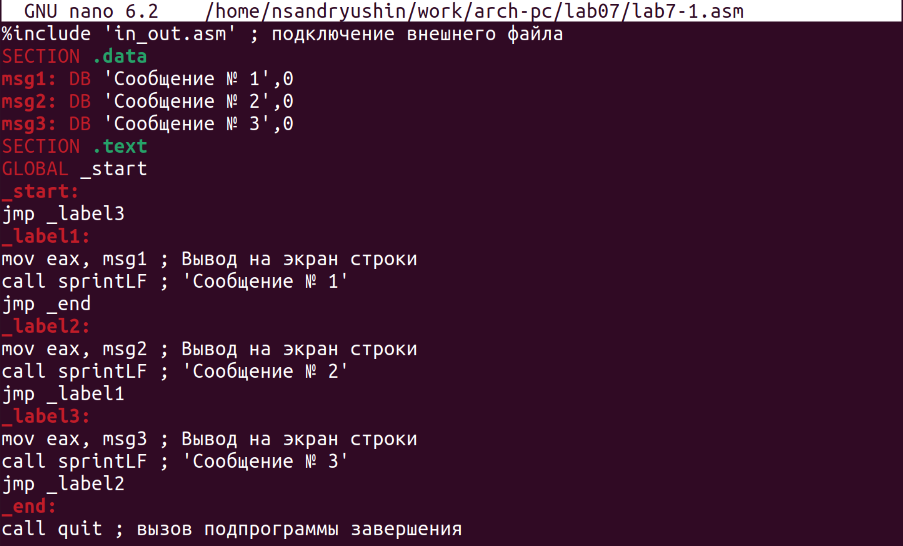
Изменение файла lab7-1.asm согласно листингу 7.2

Снова соберём программу и запустим её (рис. 2.7):



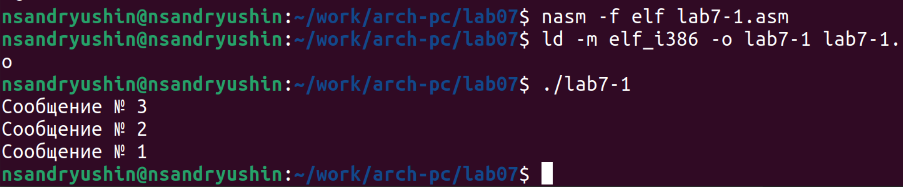
Повторная сборка программы из файла lab7-1.asm и её запуск

Теперь сделаем так, чтобы код выводил сообщения в обратном порядке (от 3 сообщения к первому). Для этого внесём в код следующие изменения (рис. 2.8):



Редактирование файла lab7-1.asm

И запустим её, предварительно собрав (рис. 2.9):



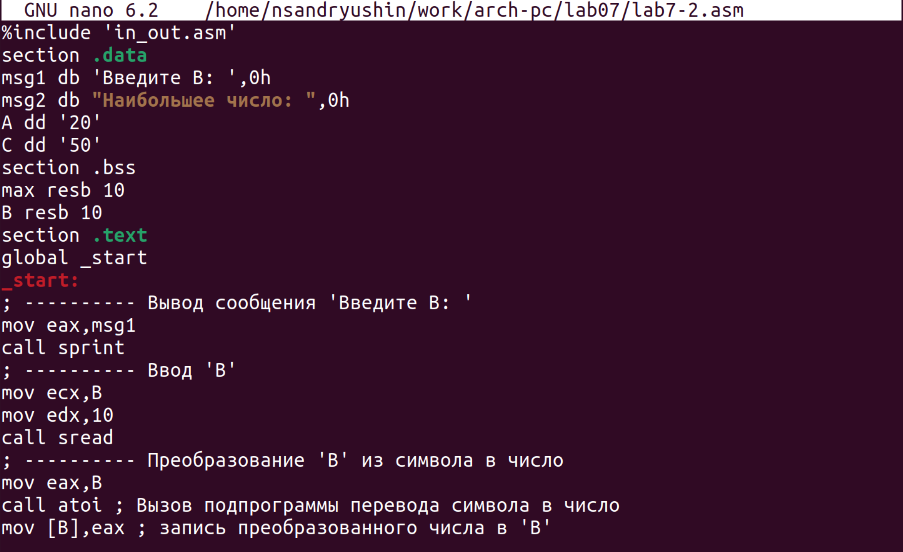
Повторная сборка программы из файла lab7-1.asm и её запуск

Теперь создадим файл lab7-2.asm (рис. 2.10):

Создание второго файла: lab7-2.asm

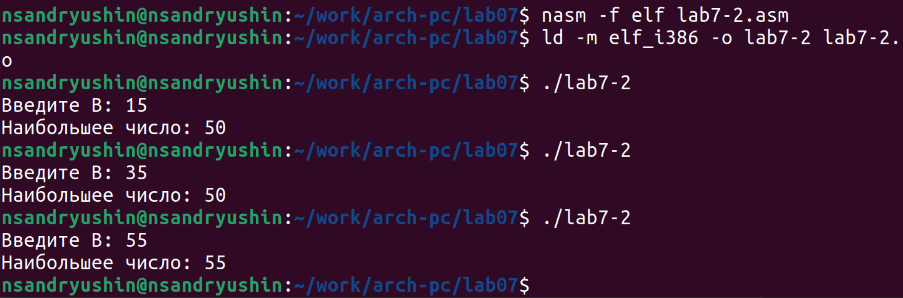
Создание второго файла: lab7-2.asm

Запишем код из листинга 7.3 в файл lab7-2.asm (рис. 2.11):



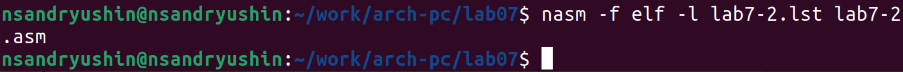
Запись кода из листинга 7.3 в файл lab7-2.asm

И запустим его, предварительно собрав (рис. 2.12):



сборка программы из файла lab7-2.asm и её запуск

Теперь попробуем создать файл листинга при сборке файла lab7-2.asm (рис. 2.13):



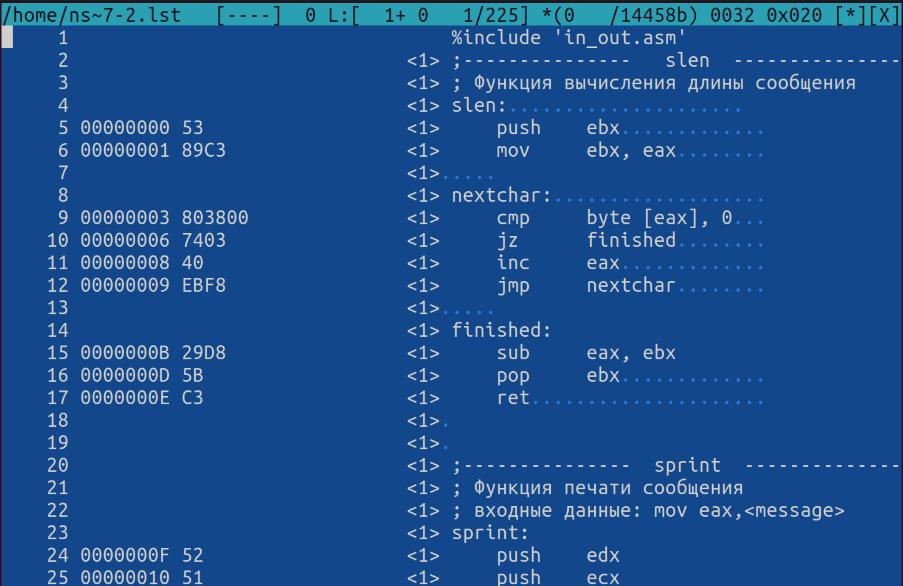
Создание файла листинга из файла lab6-2.asm

Теперь посмотрим, как выглядит файл листинга изнутри. Для этого откроем его в mcedit (рис. 2.14):

Открытие файла листинга в текстовом редакторе

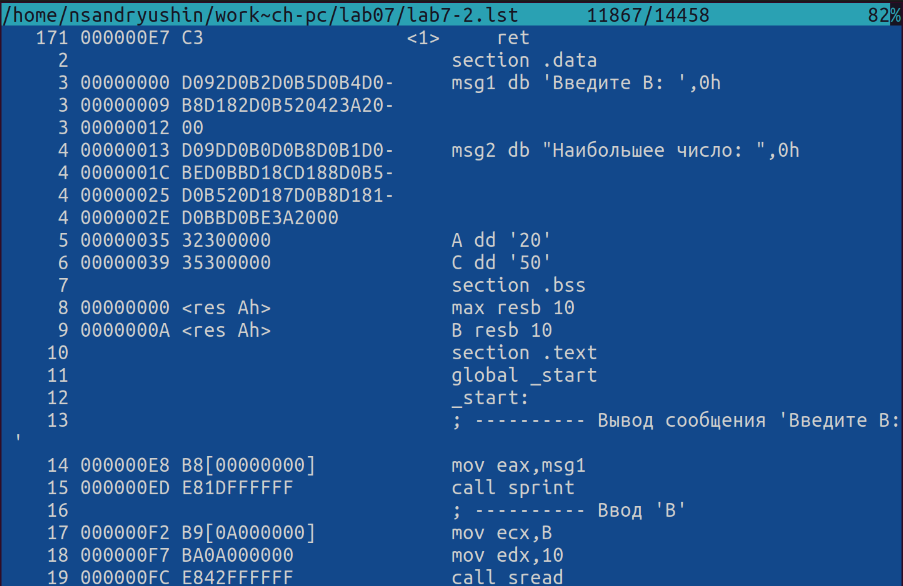
Открытие файла листинга в текстовом редакторе

Открыв его, мы видим следующую картину (рис. 2.15):



Вид файла листинга

Наша программа находится чуть ниже (рис. 2.16):



Нахождение нашей программы в файле листинга

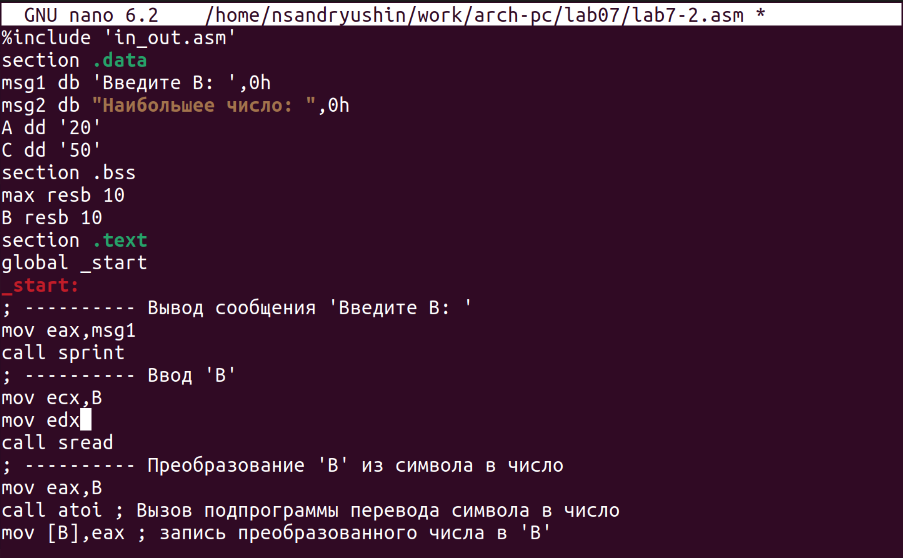
Разберём несколько строк файла листинга:

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

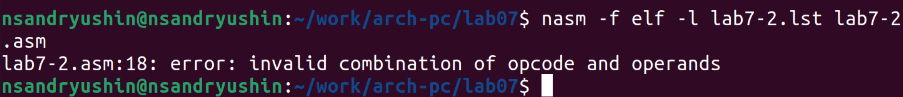
3. Строка 17 отвечает за запись переменной B в регистр ecx. Как видно, все строки имеют номер, адрес, машинный код и исходный код.

Теперь попробуем намеренно допустить ошибку в нашем коде, убрав у команды move 1 операнд (рис. 2.17):



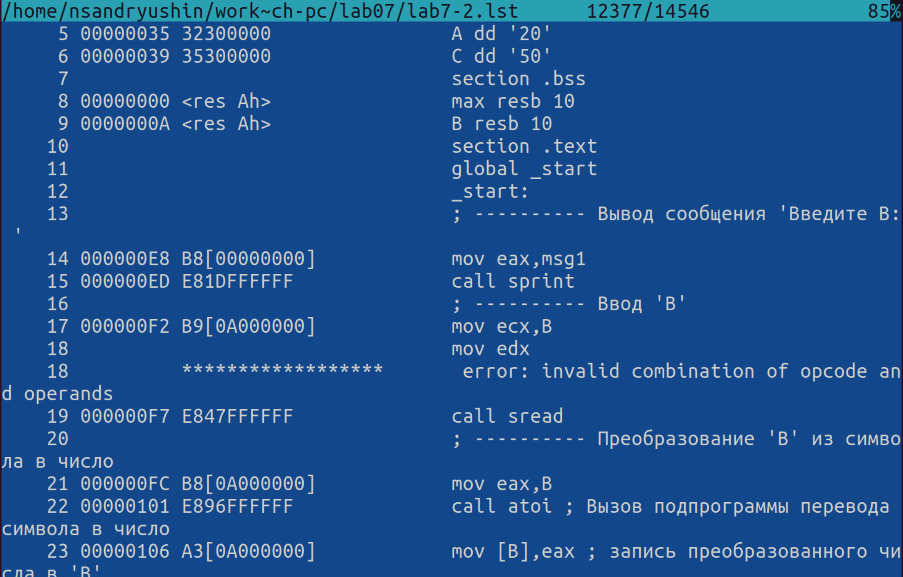
Изменение исходного файла

И попробуем собрать файл с ошибкой, генерируя файл листинга (рис. 2.18):



Вывод ошибки при сборке объектного файла

Мы видим, что объектный файл не создался, однако появился файл листинга. Теперь зайдём в файл листинга, и посмотрим, отображается ли в нём ошибка (рис. 2.19):



Отображение ошибки в листинге

Как видим, в листинге прописана ошибка

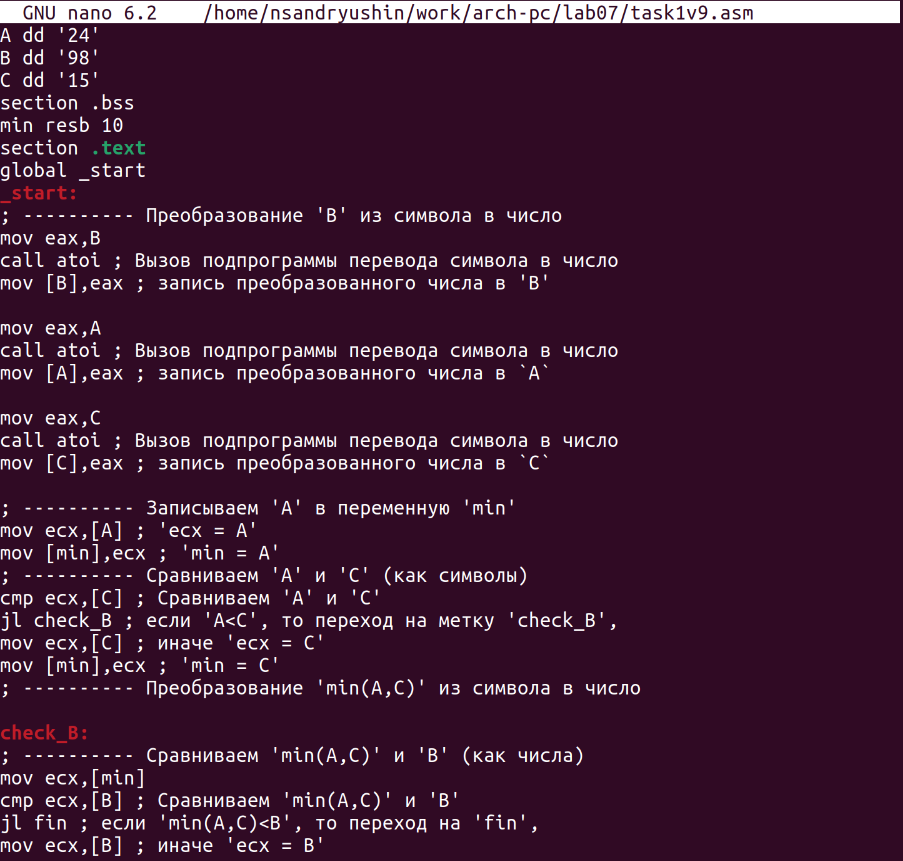
# 3 Выполнение задания для самостоятельной работы

Создадим файл для выполнения самостоятельной работы. Мой вариант - 9 (рис. 3.1):

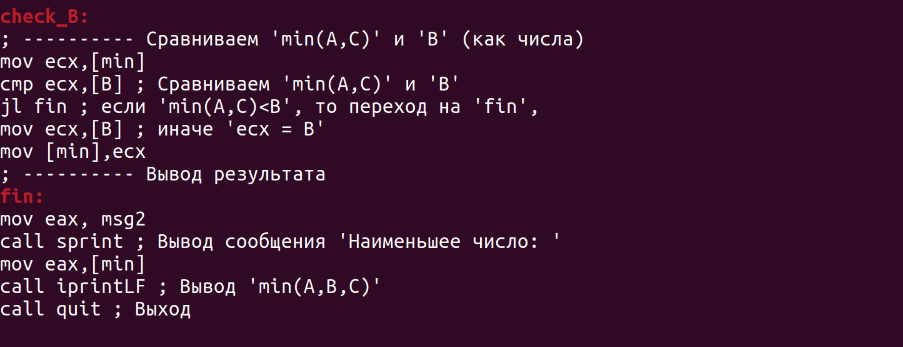
Создание первого файла самостоятельной работы

Создание первого файла самостоятельной работы

Напишем код для выполнения задания. Код выглядит так (рис. 3.2 и рис. 3.3):

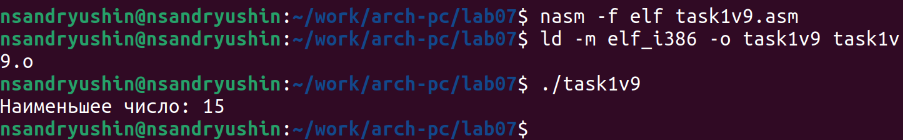


Код первого файла самостоятельной работы



Код первого файла самостоятельной работы (продолжение)

Соберём, запустим его и посмотрим на результат (рис. 3.4):



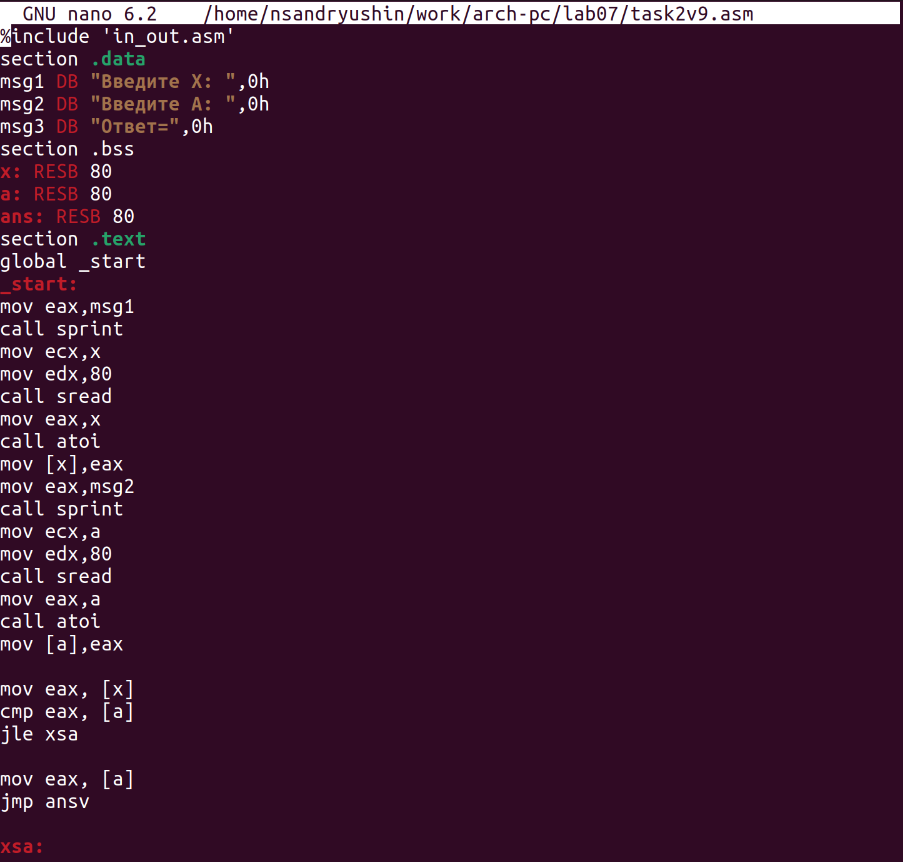
Сборка и запуск программы первого задания самостоятельной работы, а также результат выполнения

Теперь создадим второй файл самостоятельной работы для второго задания (рис. 3.5):

Создание второго файла самостоятельной работы

Создание второго файла самостоятельной работы

Код будет выглядеть так (рис. 3.6 и рис. 3.7):



Код второго файла самостоятельной работы



Код второго файла самостоятельной работы (продолжение)

Соберём исполняемый файл и запустим его (рис. 3.8):



Сборка и тестирование второго файла самостоятельной работы

Как видим, программа всё посчитала правильно

# 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.