ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №8

*дисциплина: Архитектура компьютера*

Лихтенштейн Алина Алексеевна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы №8, перейдем в него и создадим файл lab8-1.asm. (рис. 1)

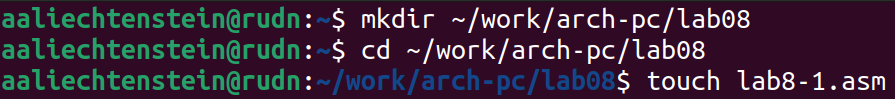


Рис. 1: Создание каталога лабораторной программы и первого текстового файла языка ассемблера

Рассмотрим пример программы с использованием инструкции jmp. Введем в файл lab8-1.asm текст программы из листинга 8.1. (рис. 2)

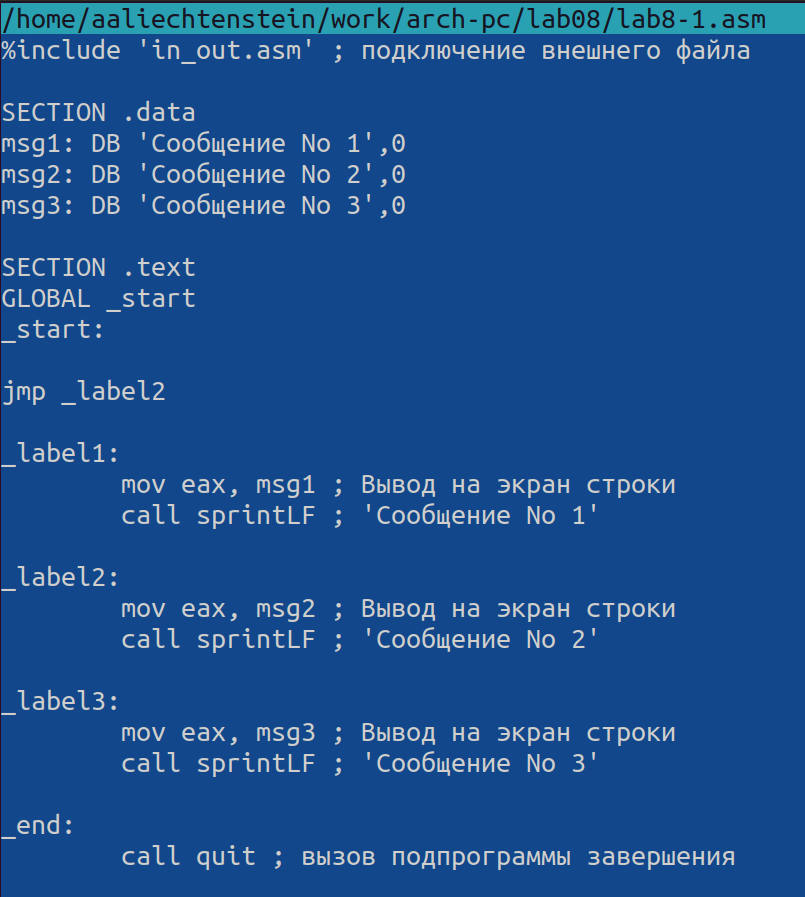


Рис. 2: Текст программы из листинга 8.1.

Создадим исполняемый файл и запустим его. Результат работы данной программы будет следующим: (рис. 3)

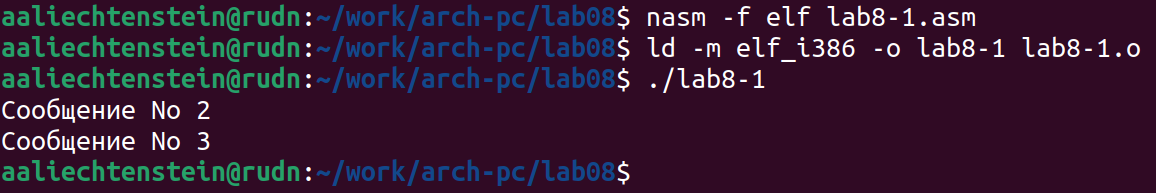


Рис. 3: Создание исполняемого файла. Запуск исполняемеого файла. Вывод программы

Изменим текст программы в соответствии с листингом 8.2. (рис. 4)

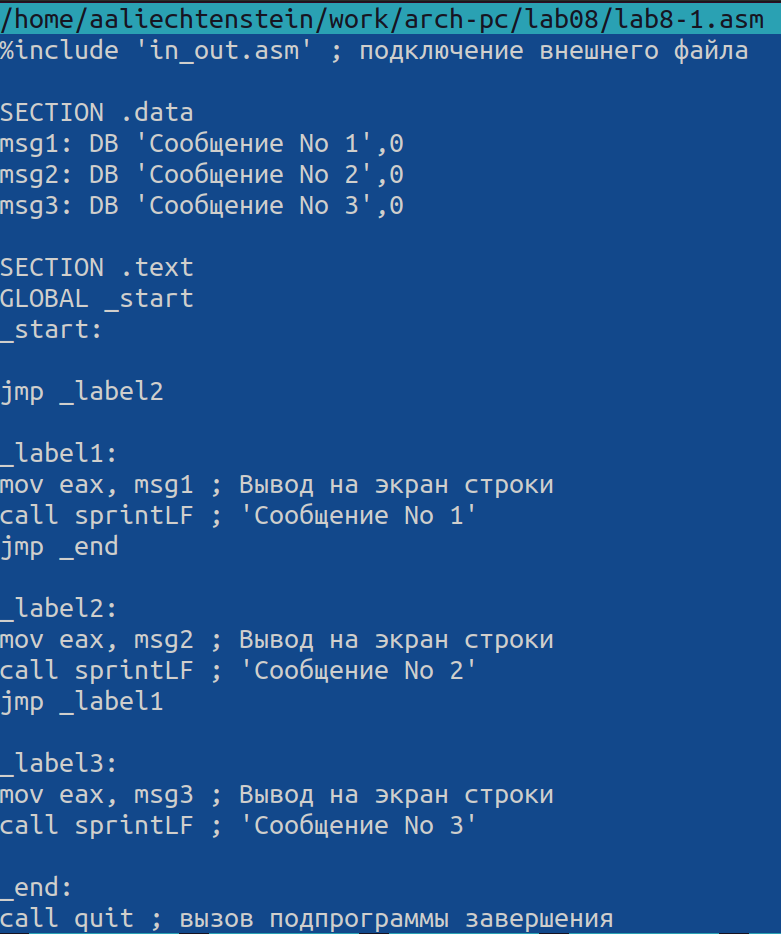


Рис. 4: Текст программы из листинга 8.2.

Создадим исполняемый файл и проверим его работу. (рис. 5)

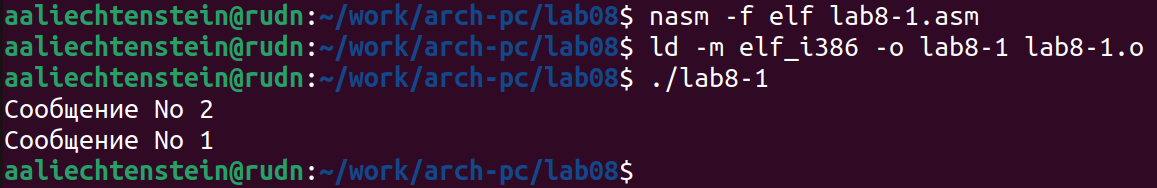


Рис. 5: Создание исполняемого файла. Запуск исполняемеого файла. Вывод программы

Изменим текст программы, добавив инструкции jmp, чтобы вывод программы был следующим: (рис. 6, 7)

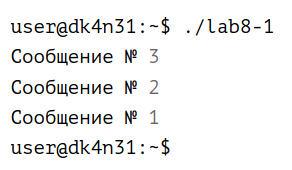


Рис. 6: Необходимый результат исполнения программы

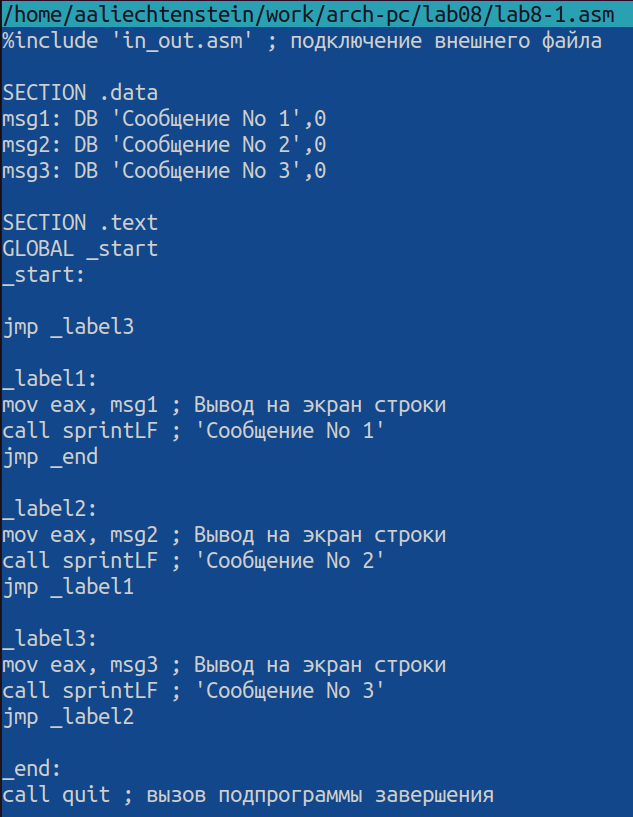


Рис. 7: Отредактированный текст программы

Создадим исполняемый файл и проверим его работу. (рис. 8)

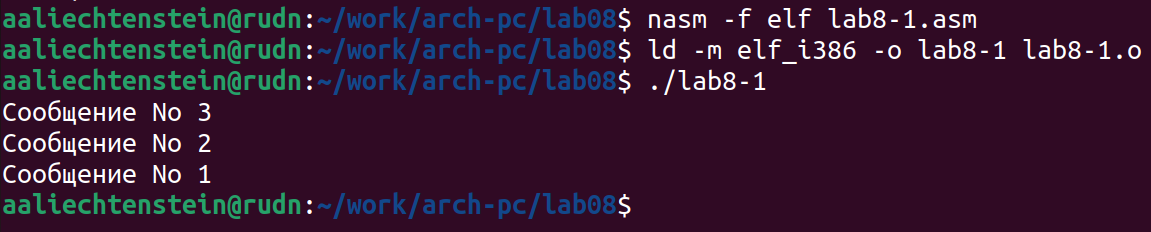


Рис. 8: Создание исполняемого файла. Запуск исполняемеого файла. Вывод программы

Создайте файл lab8-2.asm в каталоге ~/work/arch-pc/lab08. (рис. 9)

Рис. 9: Создание нового текстового файла .asm

Рис. 9: Создание нового текстового файла .asm

Внимательно изучив текст программы из листинга 8.3, введем его в lab8-2.asm. (рис. 10)

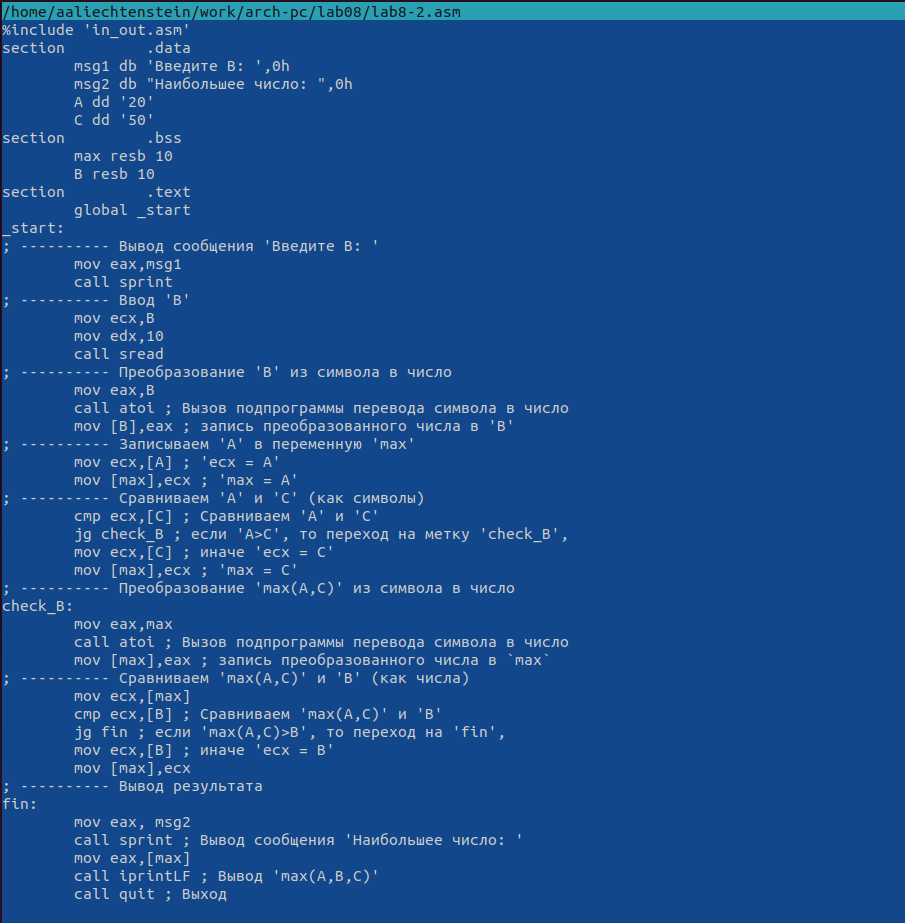


Рис. 10: Текст программы из листинга 8.3.

Создадим исполняемый файл и проверим его работу для разных значений B. (рис. 11)



Рис. 11: Создание исполняемого файла. Запуск исполняемеого файла. Вывод программы

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. 12)

Рис. 12: Создание файла листинга программы

Рис. 12: Создание файла листинга программы

Откроем файл листинга lab8-2.lst с помощью текстового редактора mcedit. (рис. 13, 14)

Рис. 13: Команда для открытия файла в текстовом редакторе mcedit

Рис. 13: Команда для открытия файла в текстовом редакторе mcedit

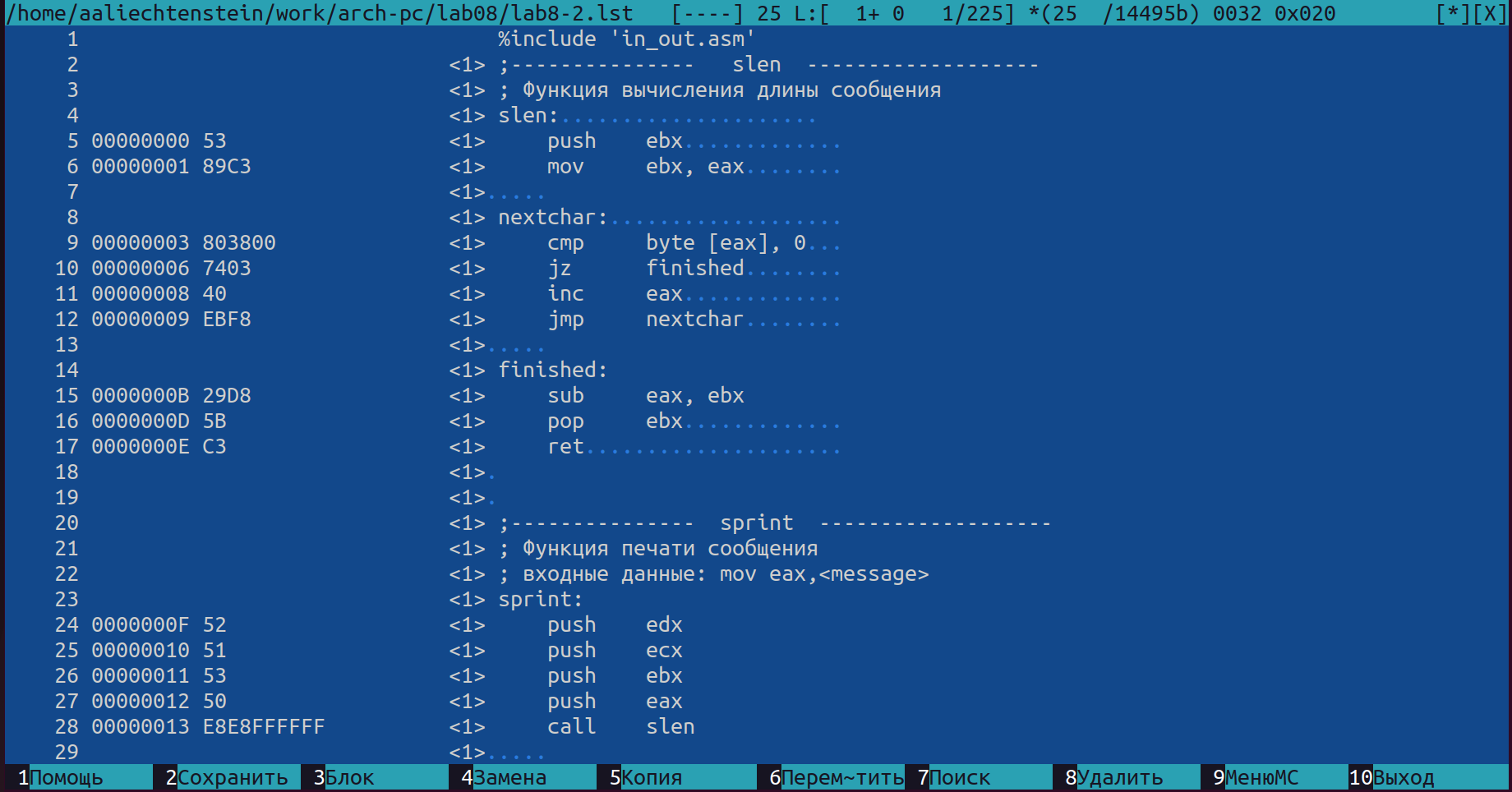


Рис. 14: Файл листинга в текстовом редакторе mcedit

Внимательно ознакомимся с его форматом и содержимым. Подробно объясним содержимое трёх строк файла листинга по выбору. (рис. 15)



Рис. 15: Три строки из файла листинга

Инструкция mov ebx, 0 начинается по смещению 00000045 в сегменте кода; далее идёт машинный код, в который ассемблируется инструкция, то есть инструкция mov ebx, 0 ассемблируется в BB00000000 (в шестнадцатеричном представлении);

Инструкция mov eax, 3 начинается по смещению 0000004A в сегменте кода; далее идёт машинный код, в который ассемблируется инструкция, то есть инструкция int 80h ассемблируется в B803000000;

Инструкция int 80h начинается по смещению 0000004F в сегменте кода; далее идёт машинный код, в который ассемблируется инструкция, то есть инструкция int 80h ассемблируется в CD80; CD80 — это инструкция на машинном языке, вызывающая прерывание ядра);

Откроем файл с программой lab8-2.asm и в любой инструкции с двумя операндами удалим один операнд. (рис. 16, 17)

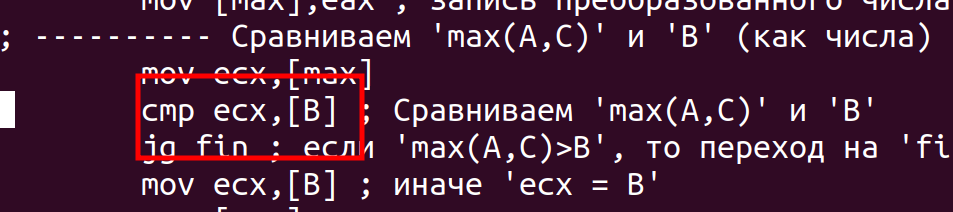


Рис. 16: Операнд [B] не удален

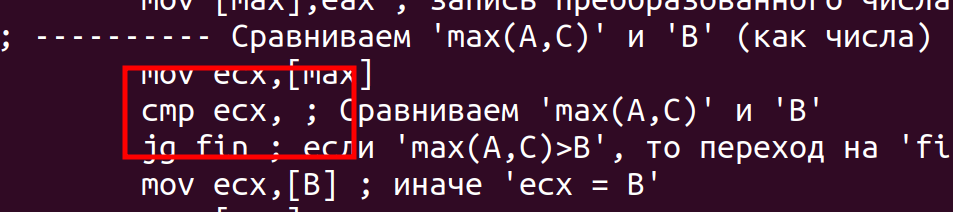


Рис. 17: Операнд [B] удален

Выполним трансляцию с получением файла листинга. (рис. 18)

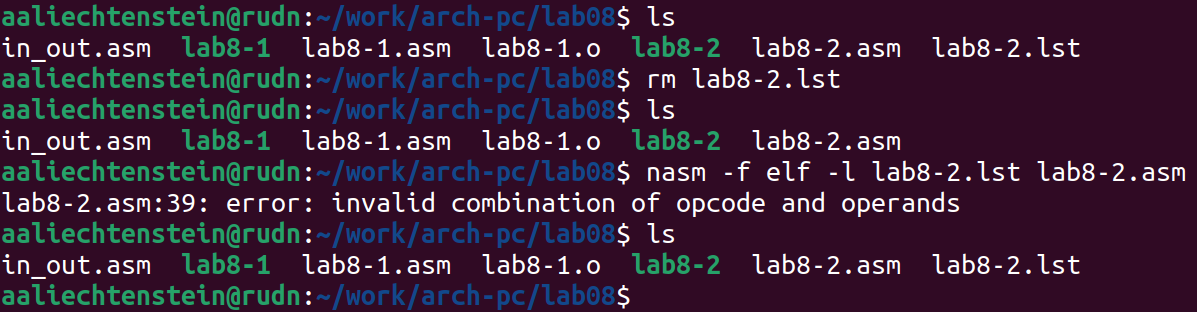


Рис. 18: Трансляция файла для получения файла листинга

Компилятор выводит ошибку: error: invalid combination of opcode and operands. Создается такой же файл листинга, но только с удаленным операндом. (рис. 19)

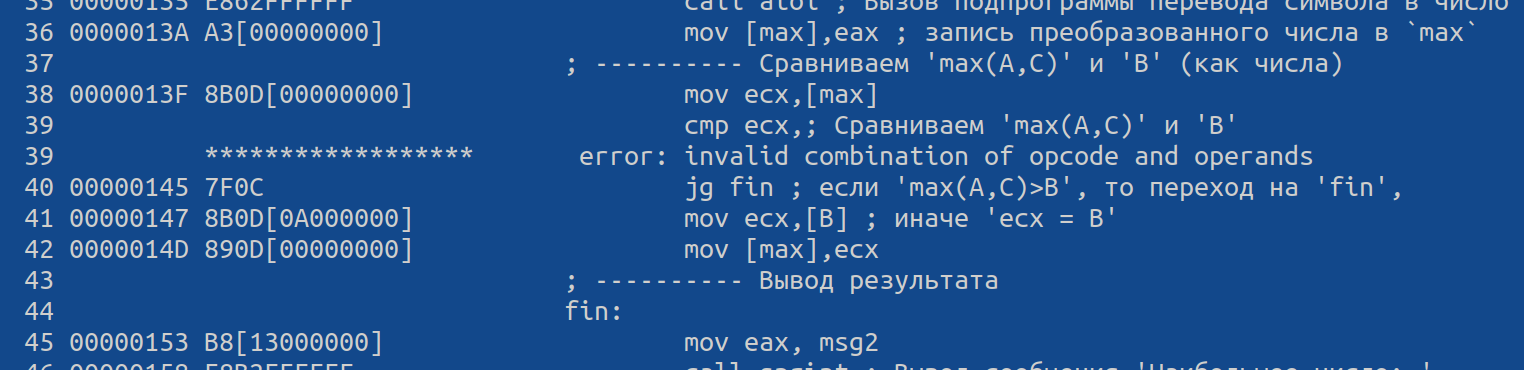


Рис. 19: Текст файла листинга с ошибкой (удален операнд)

# 3 Выполнение заданий для самостоятельной работы

1. Напишем программу нахождения наименьшей из 3 целочисленных переменных a, b и c (Вариант 14: a = 81, b = 22, c = 72). (рис. 20, 21, 22)

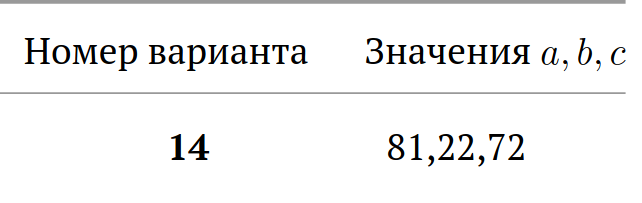


Рис. 20: Вариант 14

Рис. 21: Создание текстового файла для выполнения первого задания

Рис. 21: Создание текстового файла для выполнения первого задания

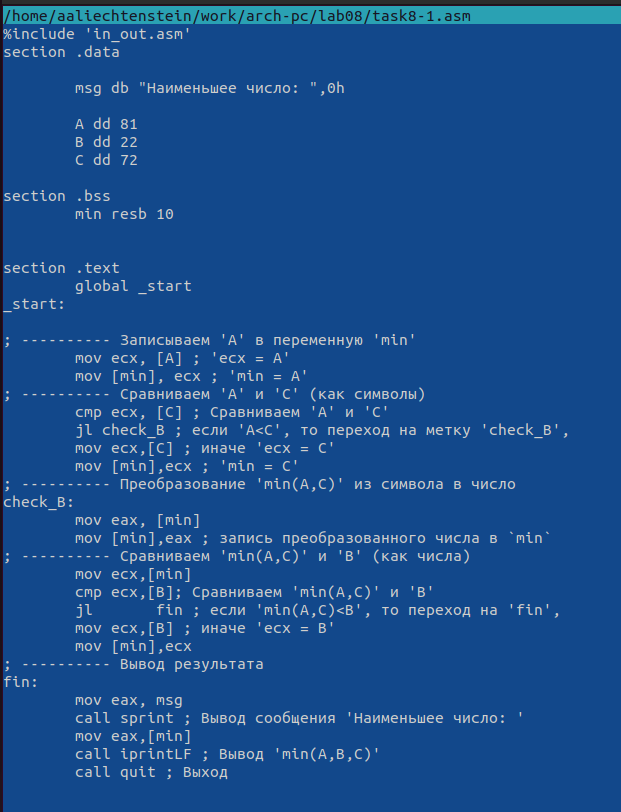


Рис. 22: Текст программы первого задания

Создадим исполняемый файл и проверим его работу. (рис. 23)

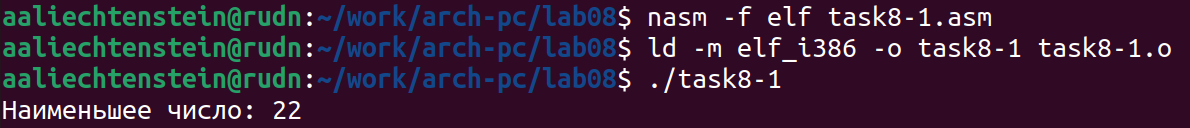


Рис. 23: Результат выполнения первой программы

Вывод программы правильный.

1. Напишем программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. (рис. 24, 25, 26)

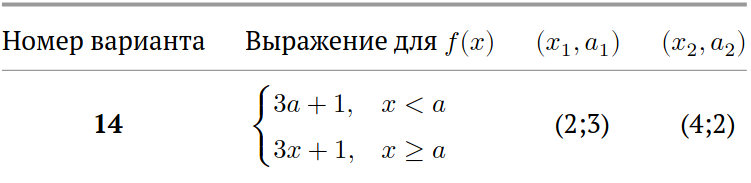


Рис. 24: Вариант 14

Рис. 25: Создание текстового файла для выполнения второго задания

Рис. 25: Создание текстового файла для выполнения второго задания

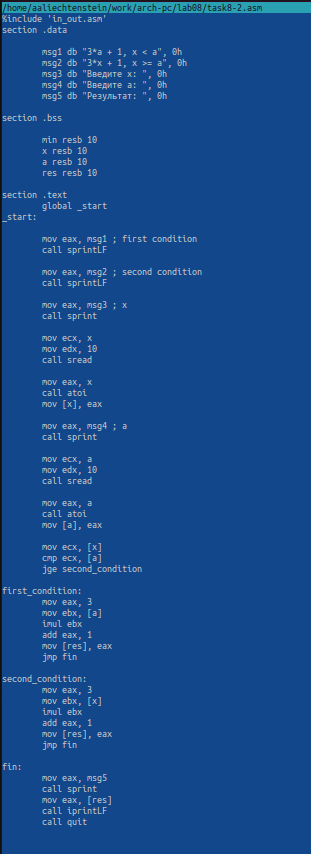


Рис. 26: Текст программы второго задания

Создайте исполняемый файл и проверьте его работу для значений x и a (Вариант 14: {x = 2, a = 3}, {x = 4, a = 2}). (рис. 27)

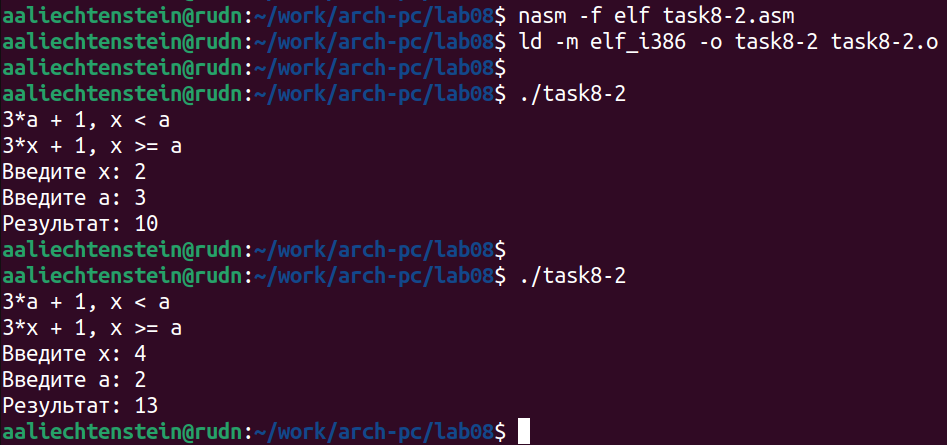


Рис. 27: Результат выполнения второй программы

# 4 Выводы

В процессе выполнения лабораторной работы были изучены команды условного и безусловного переходов, назначение и структура файла листинга. Приобретены навыки написания программ с использованием переходов.