Отчёт по лабораторной работе 7

Архитектура компьютеров и операционные системы

Старикова Владислава Александровна НММбд-03-24

Содержание

Создала каталог для программ лабораторной работы № 7 и файл lab7-1.asm (рис. 1).

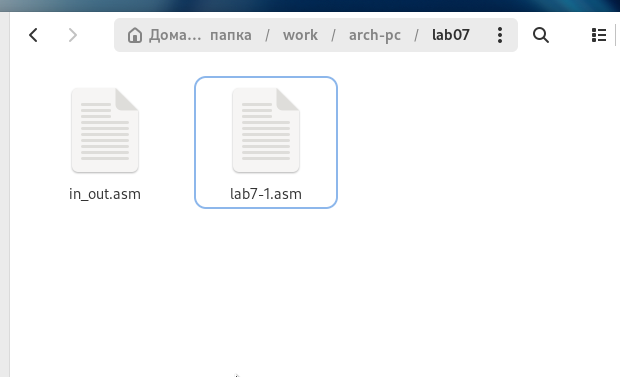


Рис. 1: Создан каталог

В NASM инструкция jmp используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции jmp. В файле lab7-1.asm разместил текст программы из листинга 7.1 (рис. 2).

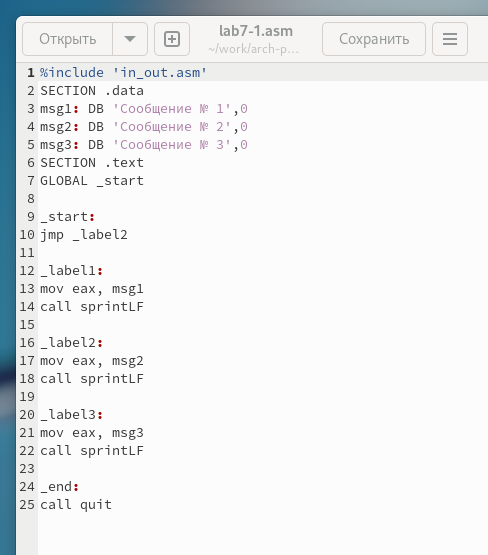


Рис. 2: Программа lab7-1.asm

Создала исполняемый файл и запустил его (рис. 3).

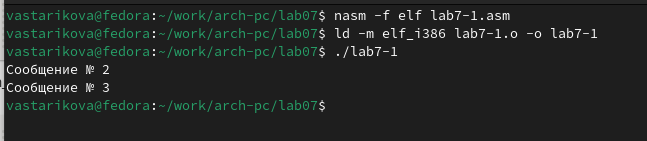


Рис. 3: Запуск программы lab7-1.asm

Инструкция jmp позволяет выполнять переходы как вперёд, так и назад. Изменил программу так, чтобы сначала выводилось сообщение № 2, затем сообщение № 1, после чего программа завершала работу. Для этого добавил в текст программы инструкцию jmp с меткой \_label1 после вывода сообщения № 2 (чтобы перейти к инструкции вывода сообщения № 1) и инструкцию jmp с меткой \_end после вывода сообщения № 1 (для перехода к инструкции call quit).

Обновила текст программы согласно листингу 7.2 (рис. 4 и 5).

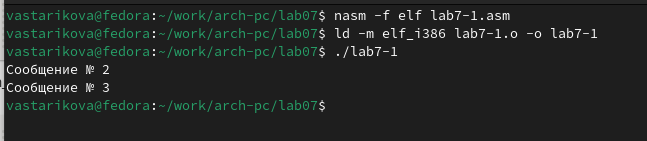


Рис. 4: Программа lab7-1.asm

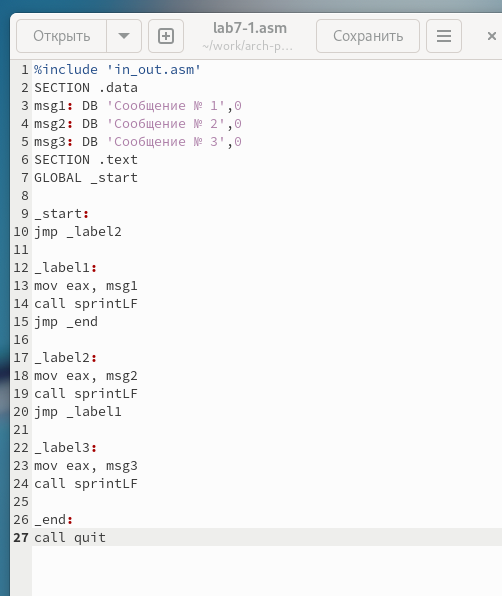


Рис. 5: Запуск программы lab7-1.asm

Изменила текст программы так, чтобы итоговый вывод программы выглядел следующим образом (рис. 6 и 7):

Сообщение № 3 Сообщение № 2 Сообщение № 1

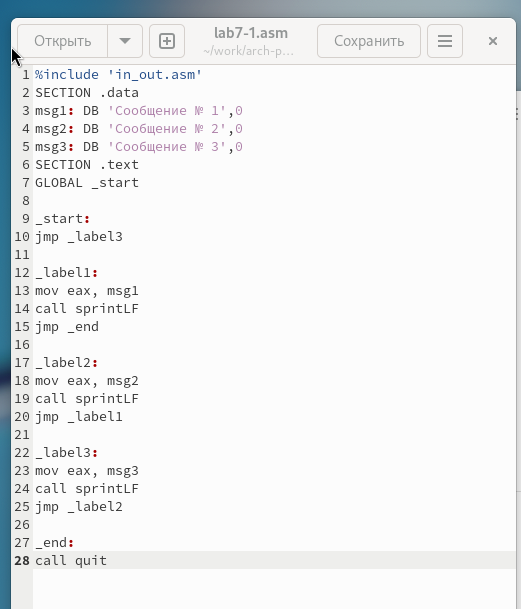


Рис. 6: Программа в файле lab7-1.asm

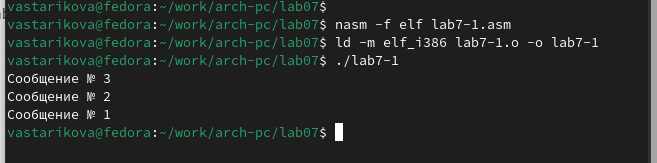


Рис. 7: Запуск программы lab7-1.asm

Использование инструкции jmp приводит к переходу в любом случае. Однако часто при написании программ требуется использовать условные переходы, когда переход должен происходить при выполнении какого-либо условия. В качестве примера рассмотрим программу, которая находит и выводит на экран наибольшую из 3 целочисленных переменных: A, B и C. Значения для A и C заданы в программе, значение B вводится с клавиатуры.

Я создала исполняемый файл и проверила его работу для разных значений B (рис. 8) (рис. 9).

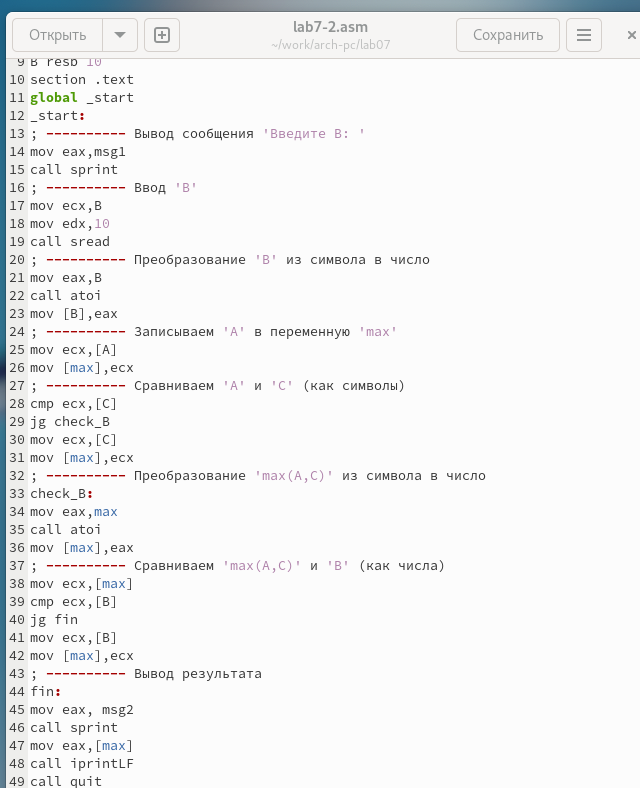


Рис. 8: Программа в файле lab7-2.asm

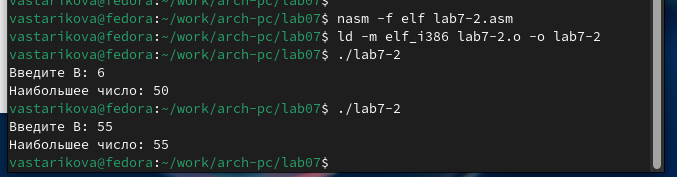


Рис. 9: Запуск программы lab7-2.asm

Обычно nasm создает в результате ассемблирования только объектный файл. Чтобы получить файл листинга, нужно указать ключ -l и задать имя файла листинга в командной строке.

Я создала файл листинга для программы из файла lab7-2.asm (рис. 10)

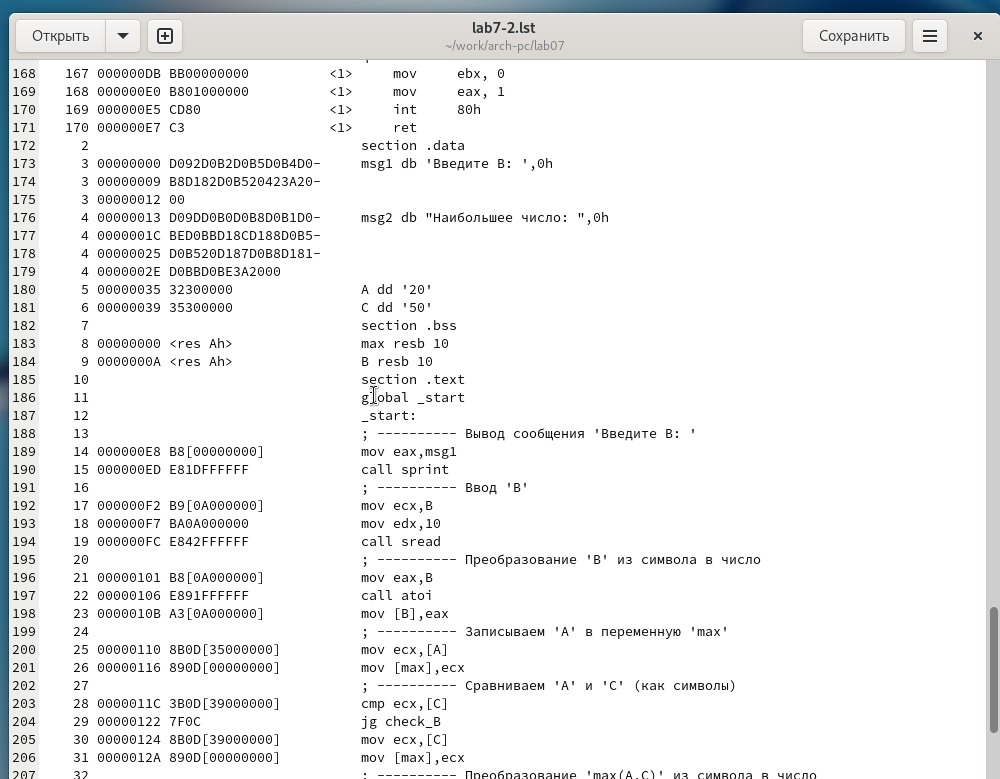


Рис. 10: Файл листинга lab7-2

Внимательно ознакомилась с его форматом и содержимым. Подробно объяснила содержимое трёх строк файла листинга.

строка 189: - 14 — номер строки в подпрограмме - 000000E8 — адрес - B8[00000000] — машинный код - mov eax,msg1 — код программы — перекладывает msg1 в eax

строка 190: - 15 — номер строки в подпрограмме - 000000ED — адрес - E81DFFFFFF — машинный код - call sprint — код программы — вызов подпрограммы печати

строка 192: - 17 — номер строки в подпрограмме - 000000F2 — адрес - B9[0A000000] — машинный код - mov ecx,B — код программы — перекладывает B в eax

Я открыла файл с программой lab7-2.asm и в инструкции с двумя операндами удалила один операнд. Выполнила трансляцию с получением файла листинга. (рис. 11) (рис. 12)

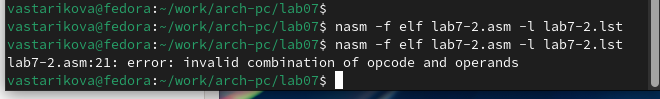


Рис. 11: Ошибка трансляции lab7-2

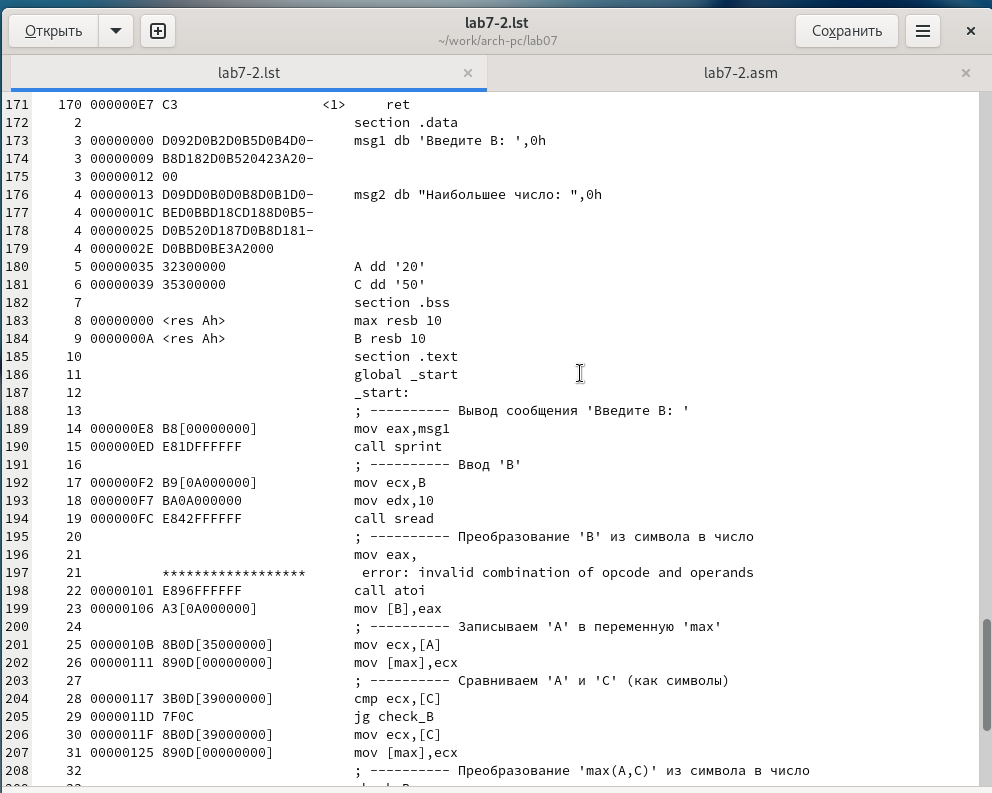


Рис. 12: Файл листинга с ошибкой lab7-2

Объектный файл не смог создаться из-за ошибки. Однако я получила листинг, в котором выделено место ошибки.

## 0.1 Самостоятельное задание

Напиши 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. Создаю исполняемый файл и проверяю его работу (рис. 13) (рис. 14).

для варианта 17 — 26,12,68

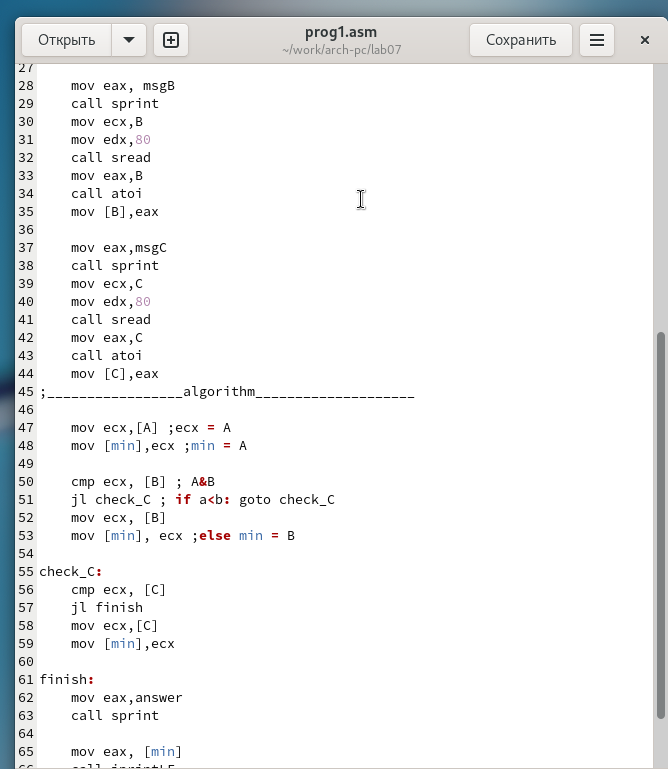


Рис. 13: Программа в файле prog1.asm

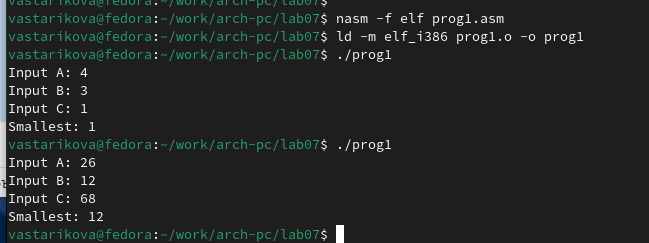


Рис. 14: Запуск программы prog1.asm

Теперь пишу программу, которая для введённых с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Вид функции f(x) выбираю из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создаю исполняемый файл и проверяю его работу для значений X и a из 7.6. (рис. 15) (рис. 16).

для варианта 17:

Если подставить получается .

Если подставить получается .

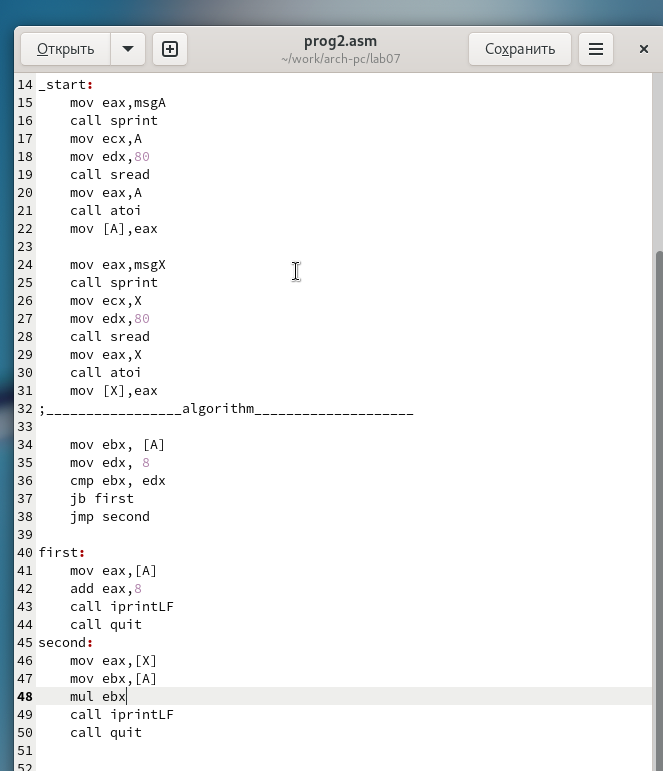


Рис. 15: Программа в файле prog2.asm

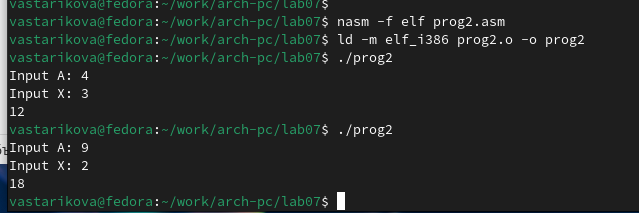


Рис. 16: Запуск программы prog2.asm

# 1 Выводы

Изучили команды условного и безусловного переходов, познакомились с фалом листинга.