Отчет лабораторной работе №7

Дисциплина: архитектура компьютера

Зайцева Ульяна Владимировна

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 Задание

1. Реализация переходов в NASM
2. Изучение структуры файлы листинга.
3. Задания для самостоятельной работы.

# 3 Теоретическое введение

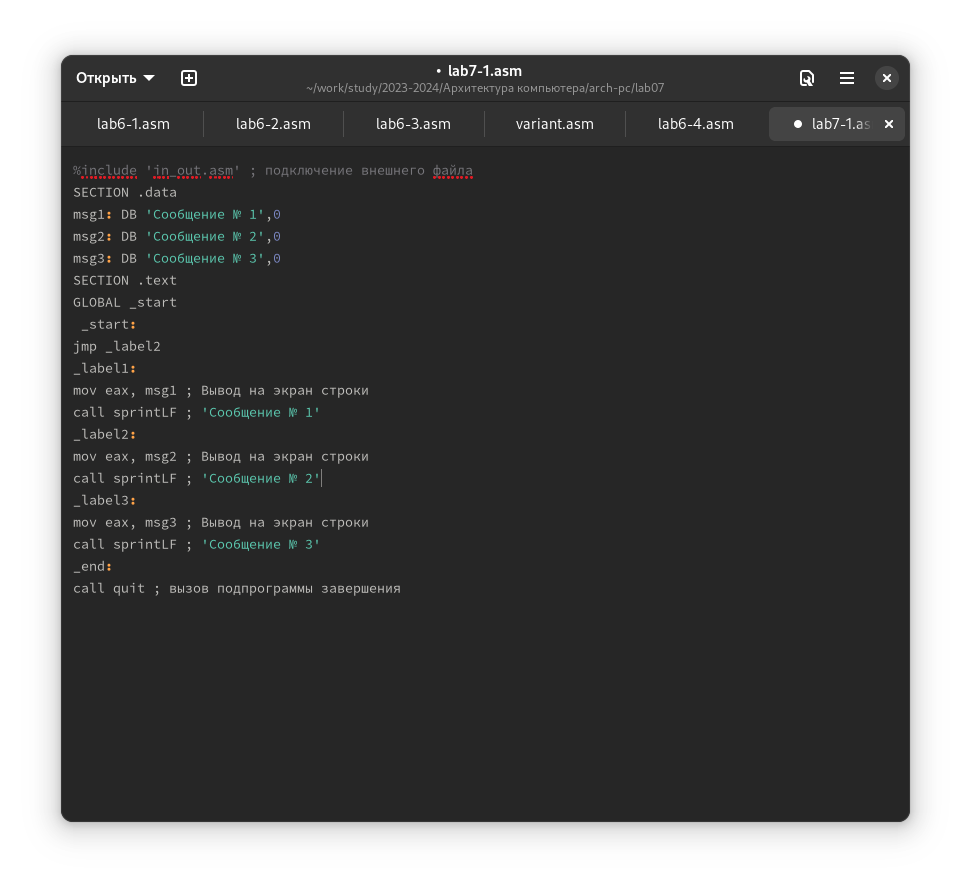
Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов: • условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия. • безусловный переход – выполнение передачи управления в определенную точку про- граммы без каких-либо условий

Безусловный переход выполняется инструкцией jmp (от англ. jump – прыжок), которая включает в себя адрес перехода, куда следует передать управление: jmp Адрес перехода может быть либо меткой, либо адресом области памяти, в которую предва- рительно помещен указатель перехода. Кроме того, в качестве операнда можно использовать имя регистра, в таком случае переход будет осуществляться по адресу, хранящемуся в этом регистре

# 4 Выполнение лабораторной работы

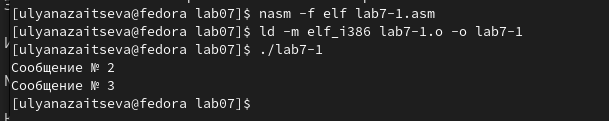
1. Реализация переходов в NASM

Создаю каталог для программ лабораторной работы №7, cоздаю файл lab7-1.asm. Ввожу в файл текст программы из листинга 7.1. (рис. ??).



Ввод листинга

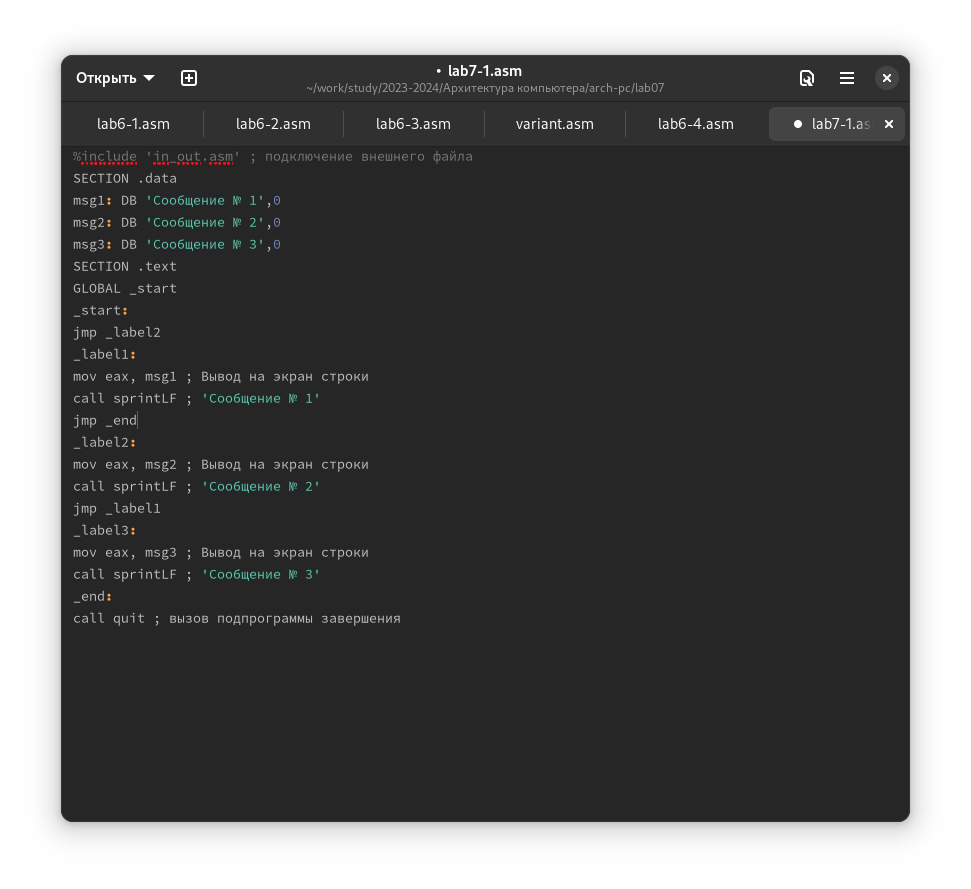
Создаю исполняемый файл и запускаю его.(рис. ??)



Выполнение

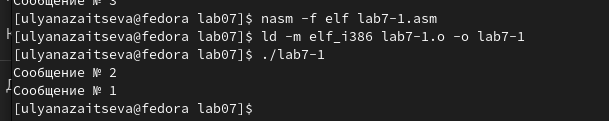
Использование инструкции jmp \_label2 изменяет порядок исполнения инструкций и позволяет выполнить инструкции начиная с \_label2, пропустив вывод первого сообщения.

Меняю программу, чтобы она выводила сначала “Сообщение № 2”, потом “Сообщение № 1’” и завершала работу. Изменяю текст программы в соответствии с листингом 7.2.(рис. ??).



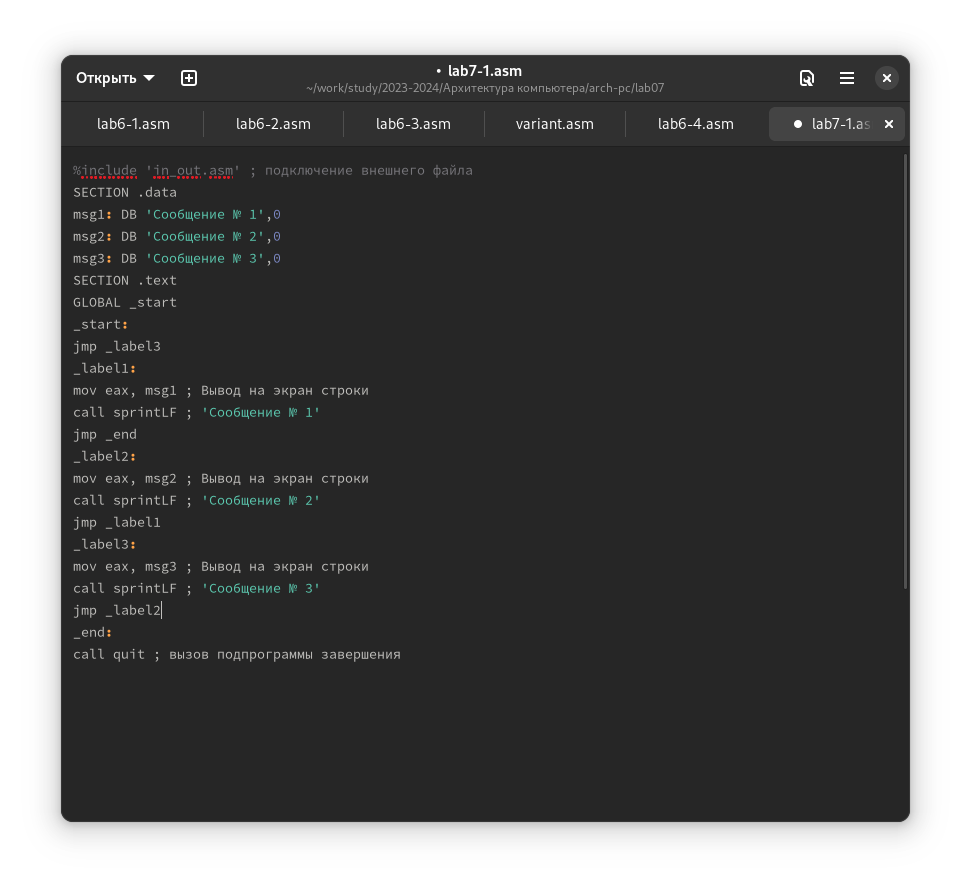
Изменяю текст программы

Создаю исполняемый файл и проверяю его работу.(рис. ??)



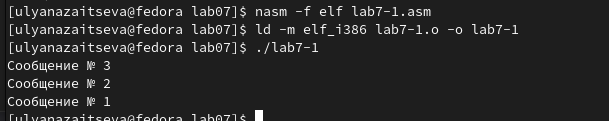
Проверка работы

После меняю текст программы, добавив в начале программы jmp \_label3, jmp \_label2 в конце метки jmp \_label3, jmp \_label1 добавляю в конце метки jmp \_label2, и добавляю jmp \_end в конце метки jmp \_label1, (рис. ??).



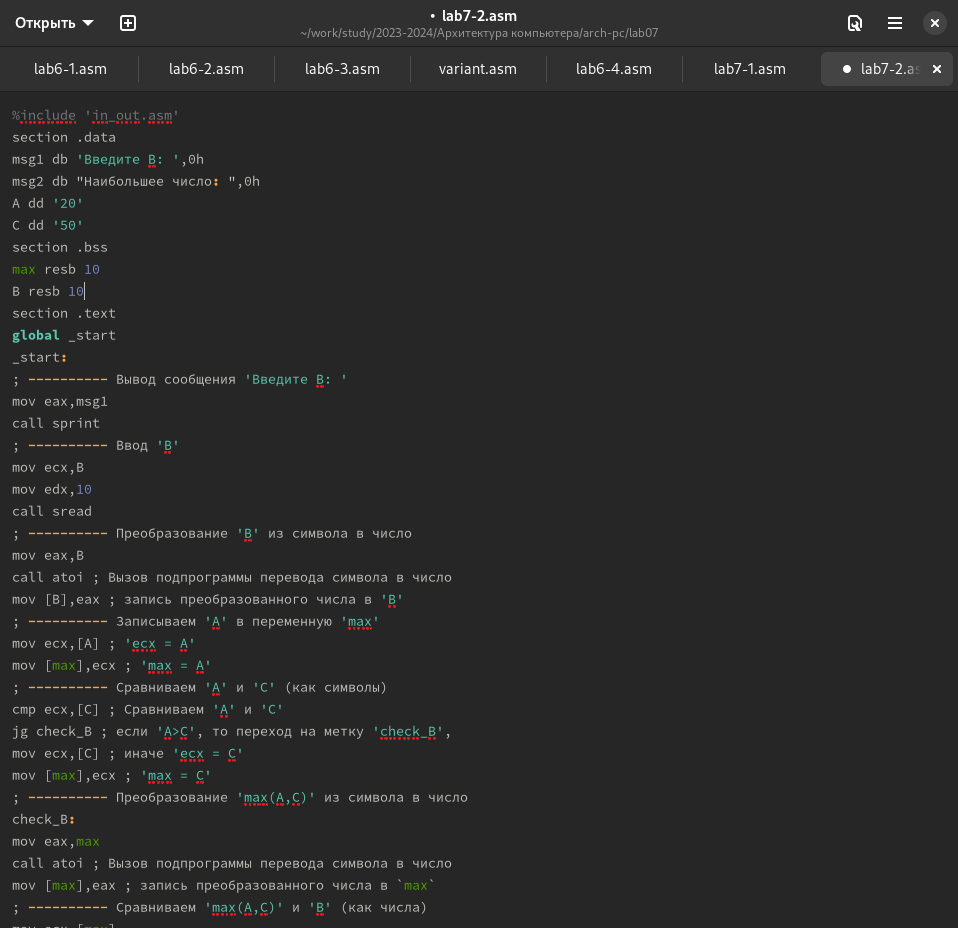
Изменяю текст

Создаю исполняемый файл и проверяю его работу.(рис. ??)



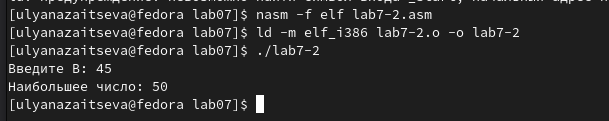
Вывод

Создаю файл lab7-2.asm. Ввожу текст программы из листинга 7.3 (рис. ??)



Ввод текста листинга

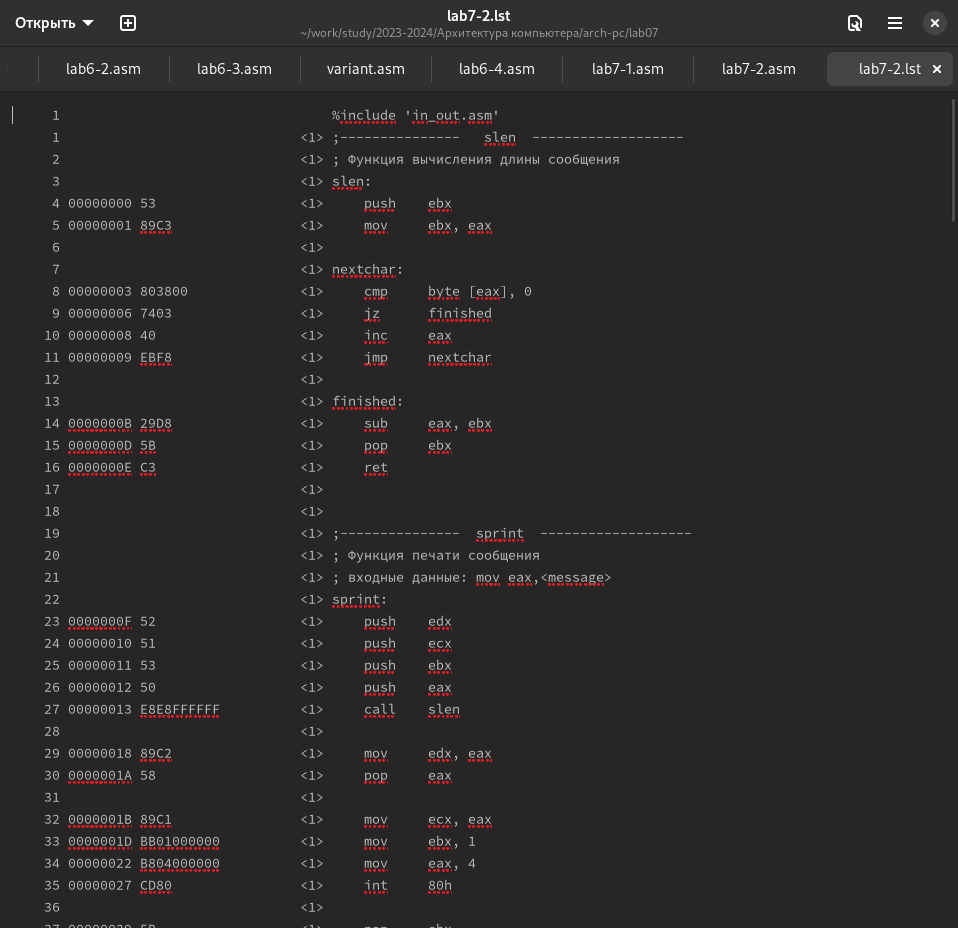
Создаю исполняемый файл и проверьте его работу. (рис. ??)



Проверка работы

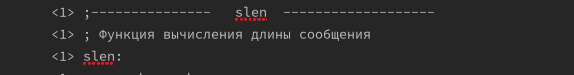
1. Изучение структуры файлы листинга.

Создаю файл листинга для программы из файла lab7-2.asm. Открываю файл листинга lab7-2.lst с помощью текстового редактора.(рис. ??)



Изучение содержимого

Рассмотрим 3 строки(рис. ??)



Рассматриваемые строки

“; —slen—–” - комментарий к коду с названием последующей функции

“; Функция вычисления длины сообщения” - комментарий к коду, не имеющий адреса и машинного кода.

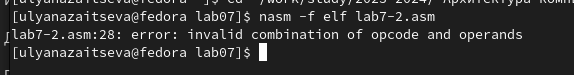
“slen” - название функции, не имеет адреса и машинного кода.

Открываю файл с программой lab7-2.asm и в выбранной мной инструкции с двумя операндами удаляю выделенный операнд.(рис. ??)



Изменение текста листинга

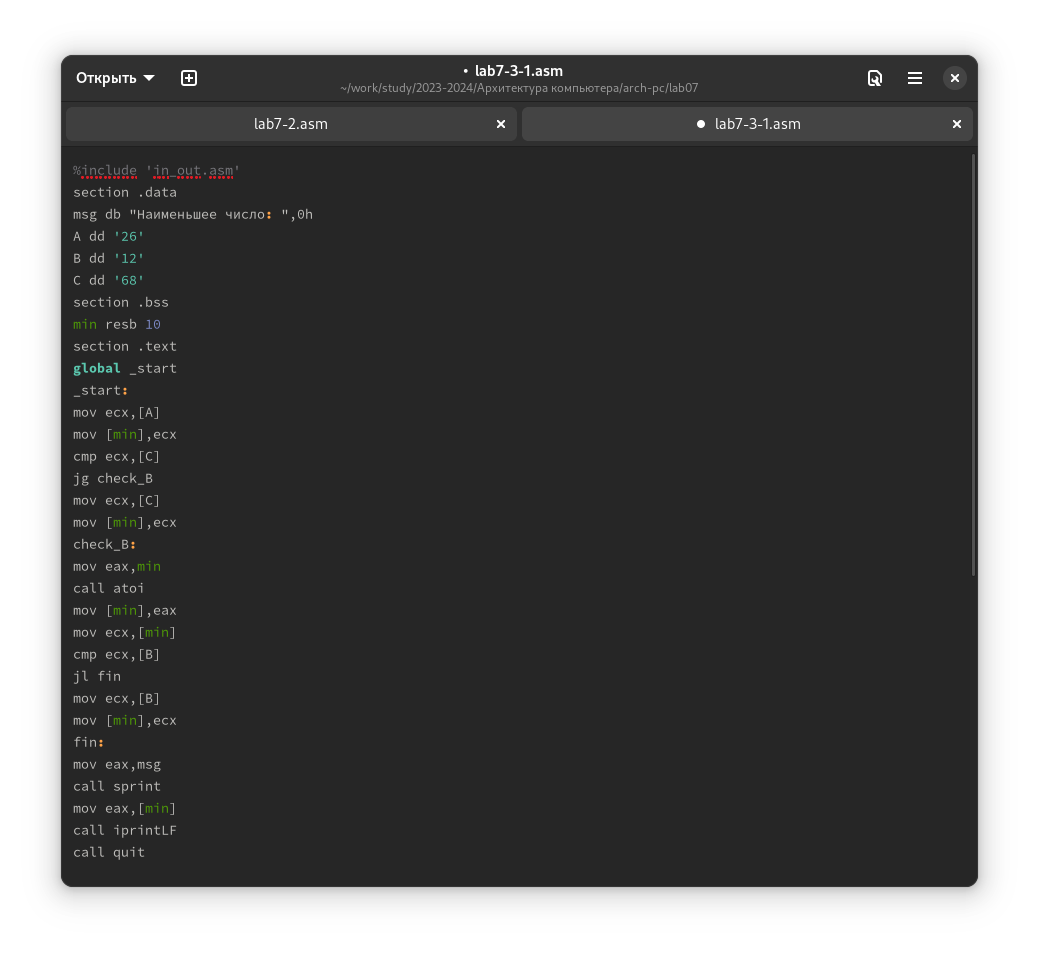
На выходе получаю ошибку.Инструкция mov не может работать, имея только один операнд.(рис. ??)



Ошибка

1. Задания для самостоятельной работы.

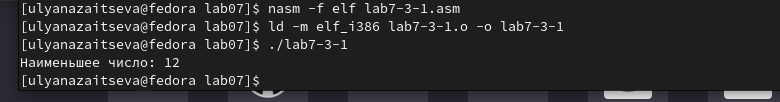
Пишу программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Мой вариант - 17, поэтому мои значения - 26, 12 и 68 (рис. ??)



Текст программы

%include ‘in\_out.asm’ section .data msg db “Наименьшее число:”,0h A dd 26 B dd 12 C dd 68 section .bss min resb 10 section .text global \_start \_start: mov ecx,[A] mov [min],ecx cmp ecx,[C] jl check\_B mov ecx,[C] mov [min],ecx check\_B: mov ecx,[min] cmp ecx,[B] jl fin mov ecx,[B] mov [min],ecx fin: mov eax,msg call sprint mov eax,[min] call iprintLF call quit

Проверяю его работу(рис. ??)



Проверка работы программы

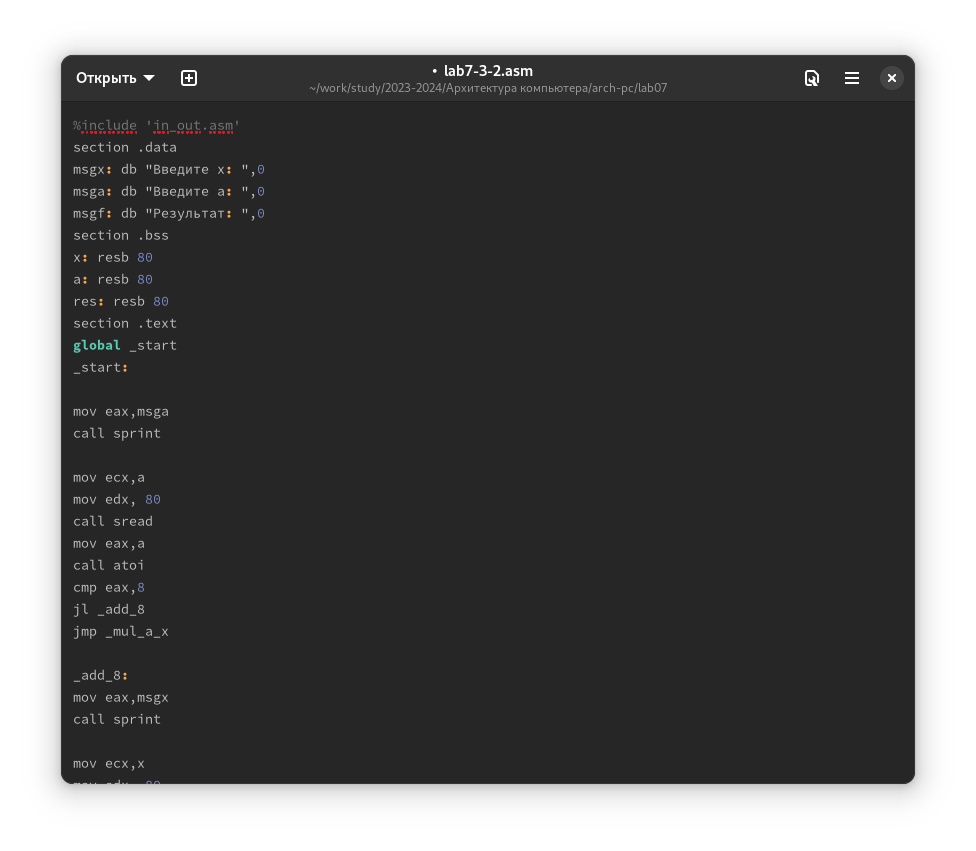
Программа работает исправно!

Пишу программу, которая для введенных с клавиатуры значений х и а вычисляет значение и выводит результат вычислений заданной для моего варианта функции f(x):

а + 8, если а < 8

а\*х, если а >= 8

(рис. ??).



Текст программы

%include ‘in\_out.asm’ section .data msgx: db “Введите x:”,0 msga: db “Введите a:”,0 msgf: db “Результат:”,0 section .bss x: resb 80 a: resb 80 res: resb 80 section .text global \_start \_start:

mov eax,msga call sprint

mov ecx,a mov edx, 80 call sread mov eax,a call atoi cmp eax,8 jl \_add\_8 jmp \_mul\_a\_x

\_add\_8: mov eax,msgx call sprint

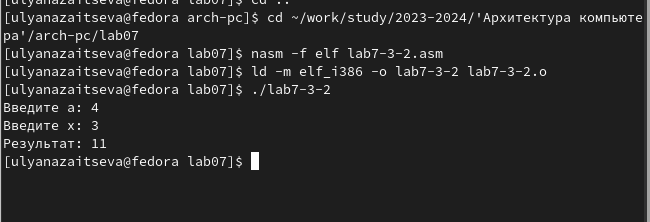
mov ecx,x mov edx, 80 call sread mov eax,x call atoi

add eax, 8 jmp \_end

\_mul\_a\_x: mov edx, a mov eax, x mul edx

\_end: mov ecx, eax mov eax, msgf call sprint mov eax, ecx call iprintLF call quit

Проверяю его работу(рис. ??)



Проверка работы

Ответ верный.

# 5 Выводы

Во время выполнения лабораторной работы я изучила команды условного и безусловного переходов, приобрела навыки написания программ с использованием переходов, познакомилась с назначением и структурой файла листинга.