Отчет по лабораторной работе №7

Дисциплина: архитектура компьютера

Федорова Анжелика Игоревна

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

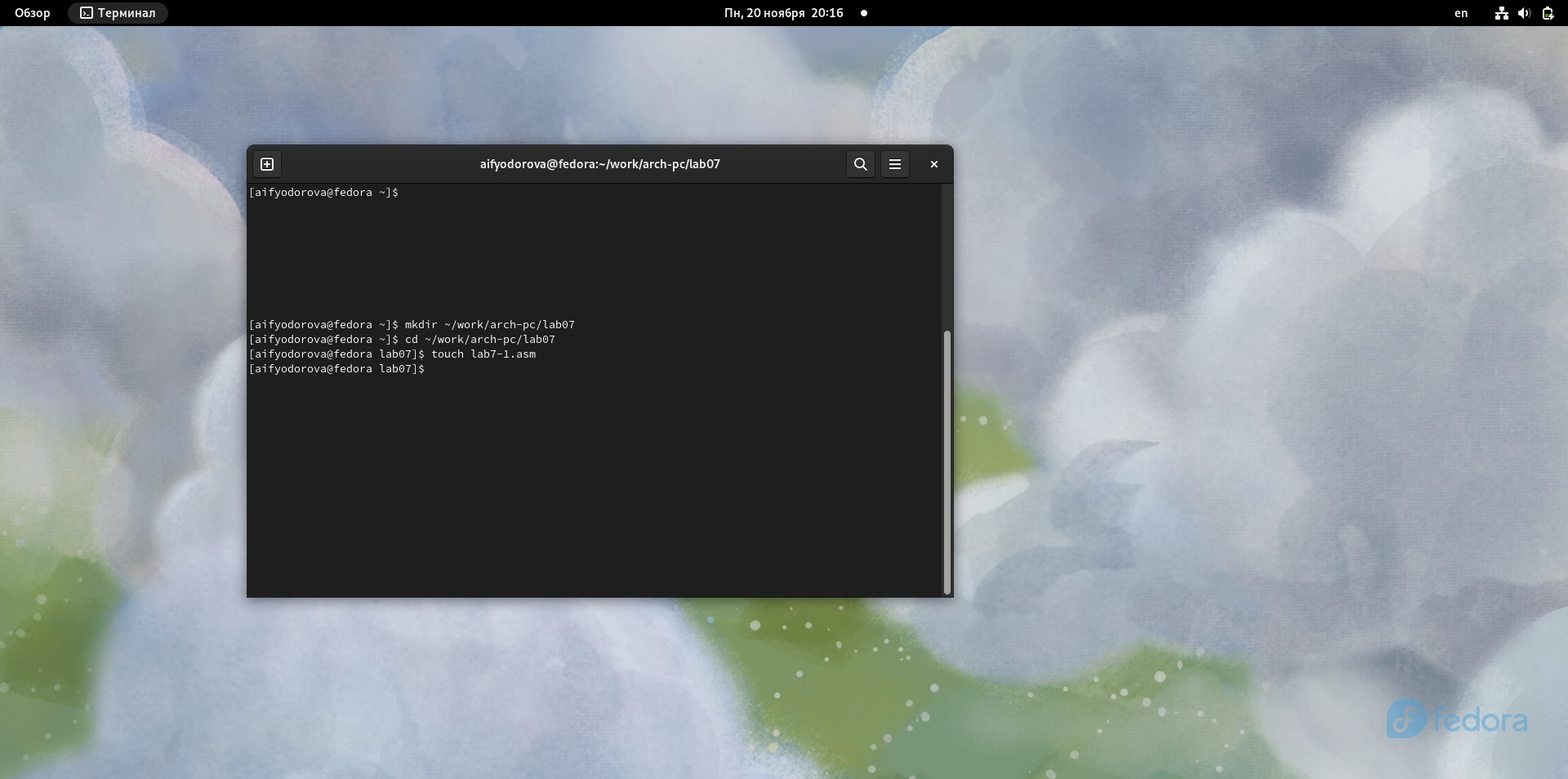
1. Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и . Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы №6. Создайте исполняемый файл и проверьте его работу.
2. Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Вид функции f(x) выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы №6. Создайте исполняемый файл и проверьте его работу для значений x и a из 7.6.

# 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.

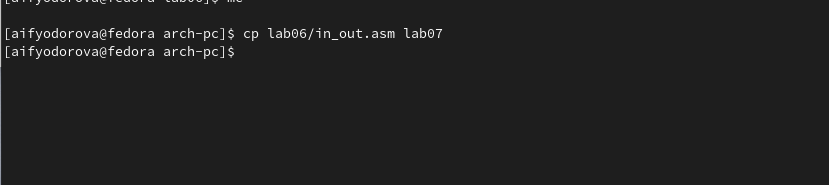
# 4 Выполнение лабораторной работы

Создаю подкаталог lab07 в каталоге arch-pc с помощью команды mkdir, захожу в него и создаю файл lab7-1.asm с помощью команды touch(рис.fig:001).



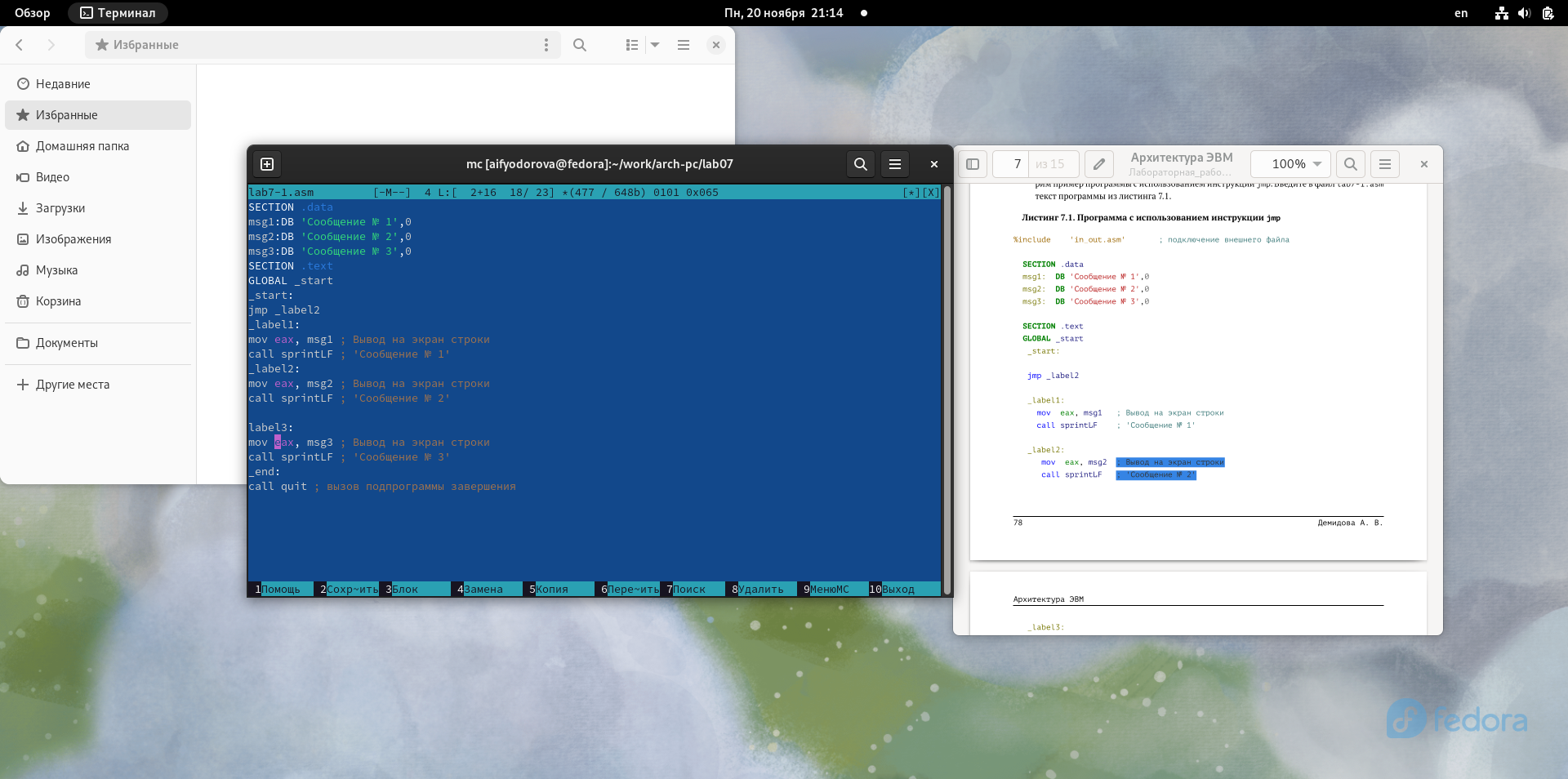
Создание каталога lab07

Также я должна скопировать файл in\_out.asm в данную директорию, чтобы в дальнейшем подключить ее. (рис.fig:002).



Копирование файла in\_out.asm в нужный каталог

Теперь я открываю файл lab7-1.asm и копирую в него содержание листинга из лабораторной работы (рис.fig:003).



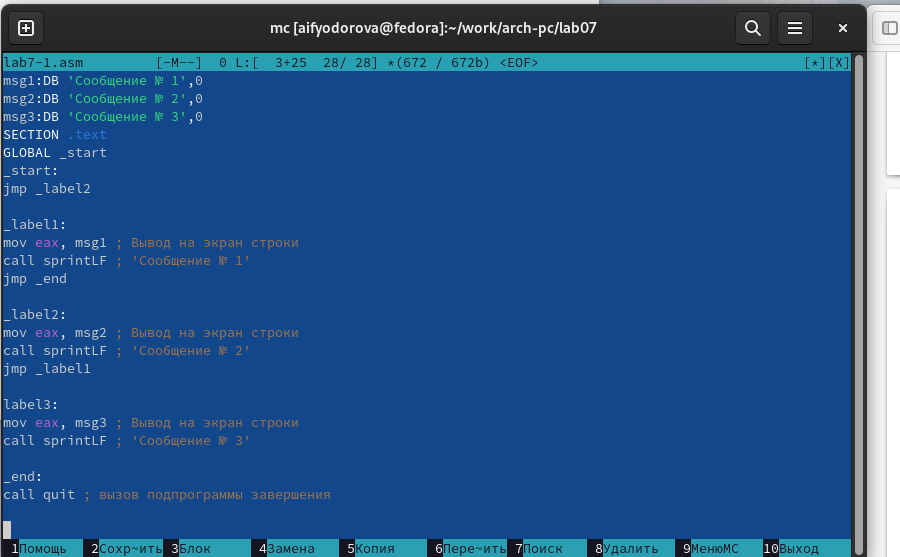
Запись содержания листинга в файл lab7-1.asm

Теперь я создаю исполняемый файл и запускаю его. (рис.fig:004)



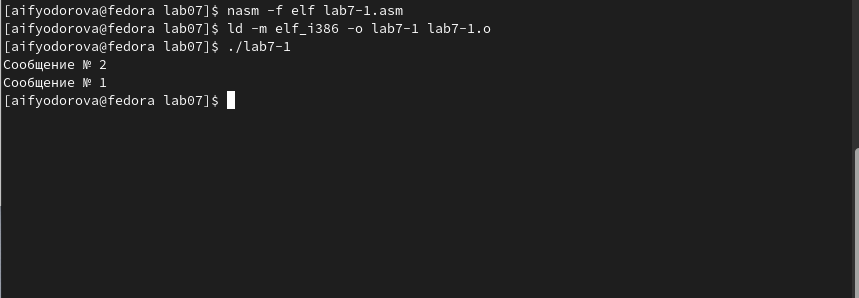
Запуск файла lab7-1

Вижу, что выводятся только тексты “Сообщение 2” и “Сообщение 3” . Теперь я переделываю код в листинге, согласно второму данному мне листингу 7.2 в лабораторной работе. Нужно, чтобы сначала выводилось “Сообщение 1”, а потом “Сообщение 2”. (рис.fig:005)



Переделываю код файла lab7-1.asm в соответствии с лист.7.2

Снова создаю объектный файл и запускаю его. (рис.fig:006)



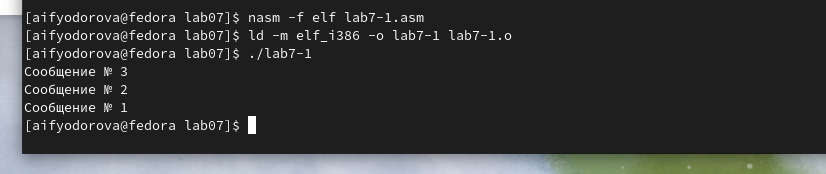
Запуск переделанного файла lab7-1

Вижу, что теперь текст выводится иначе. Теперь нужно, чтобы сначала выводилось “Сообщение №3”, а затем последовательно по убыванию номера остальные. Для этого надо расположить функции jmp в опредленной последовательности (рис.fig:007)



Переделанный код файла lab7-1.asm для вывода нужной текста

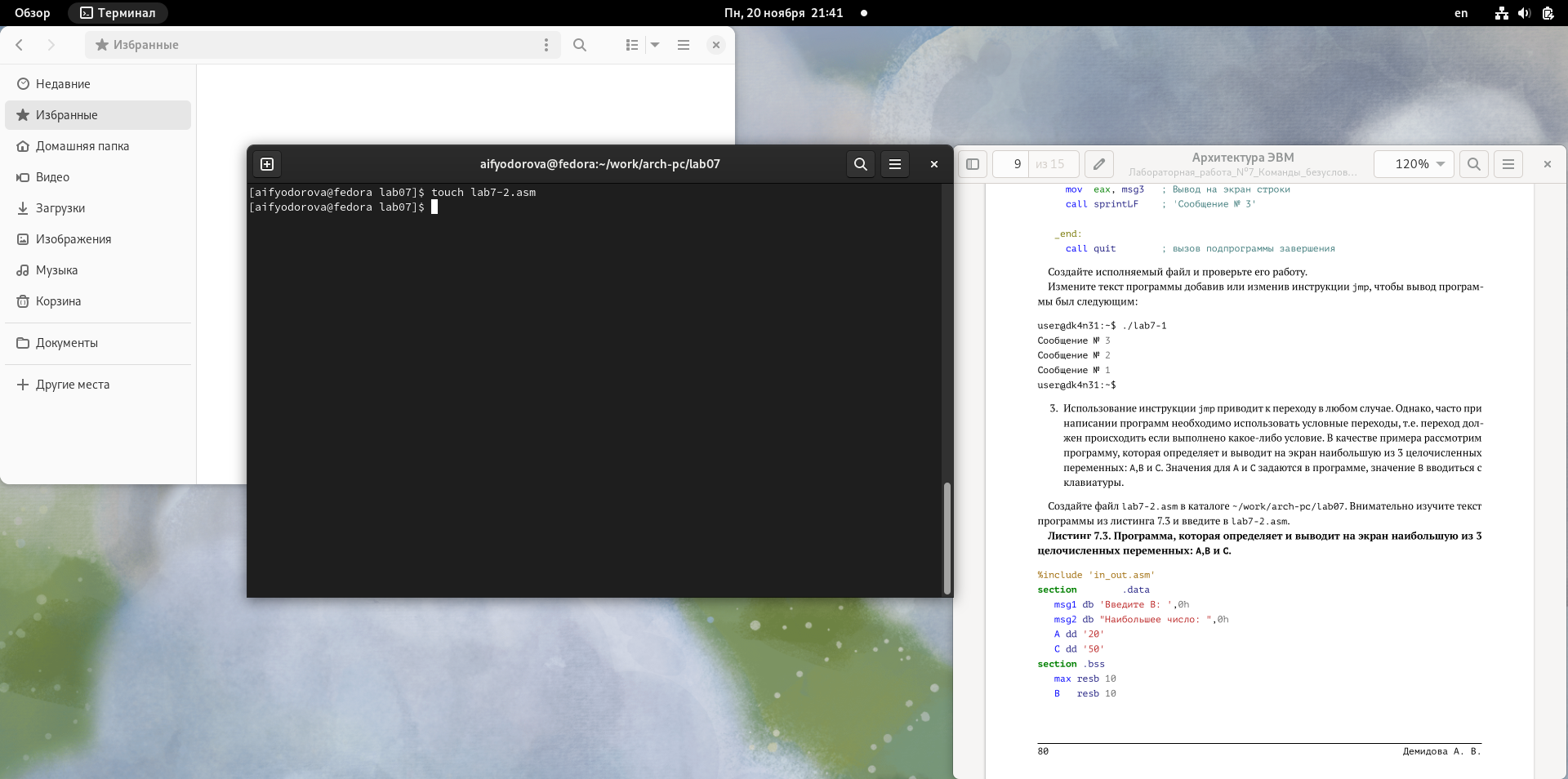
Cоздаю исполняемый файл и запускаю его (рис.fig:008)



Запуск файла lab7-1.asm после

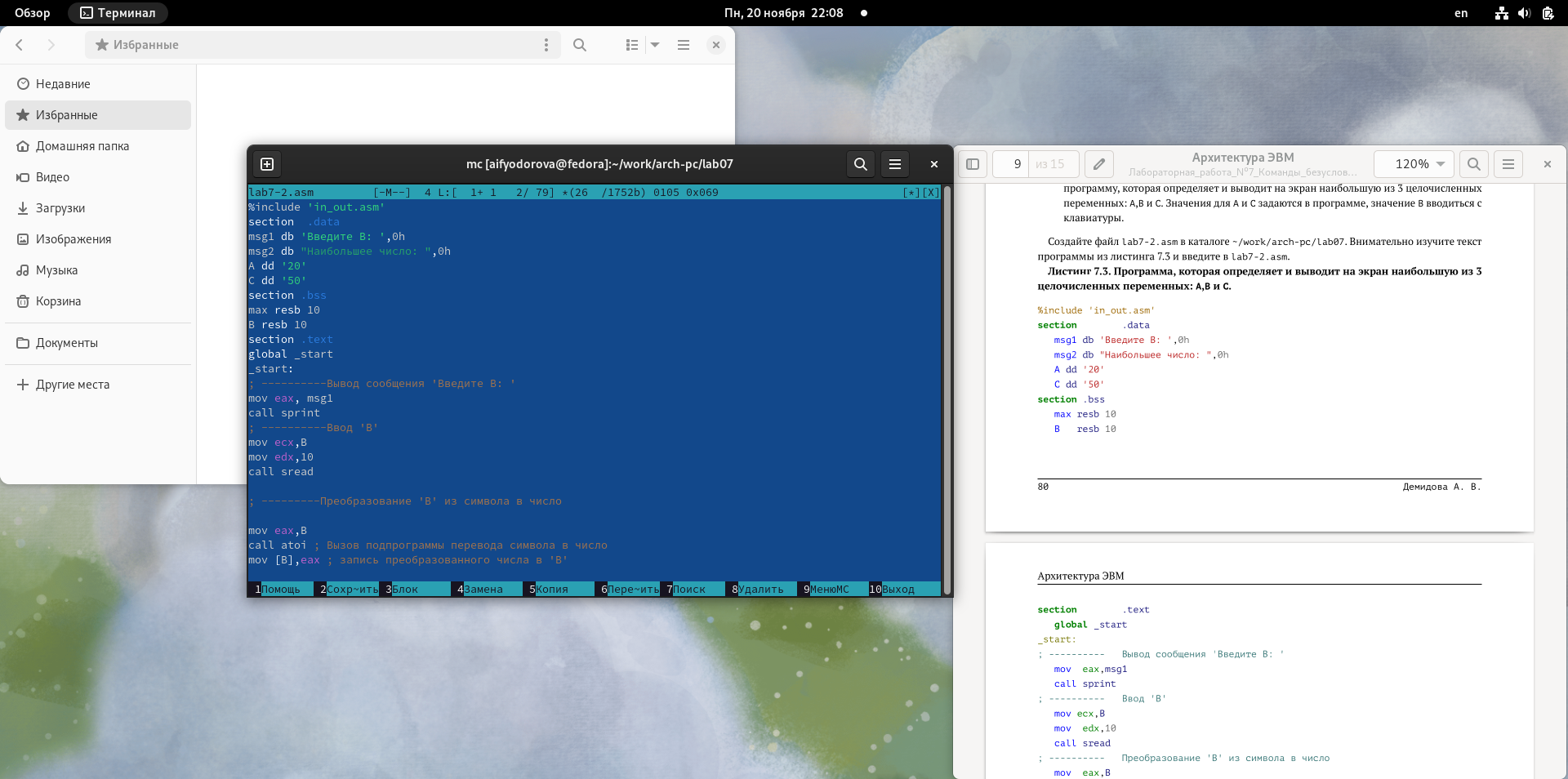
Вижу, что текст вывелся в соответсвии с требованиями.

Теперь я создаю файл lab7-2.asm. (рис.fig:009)



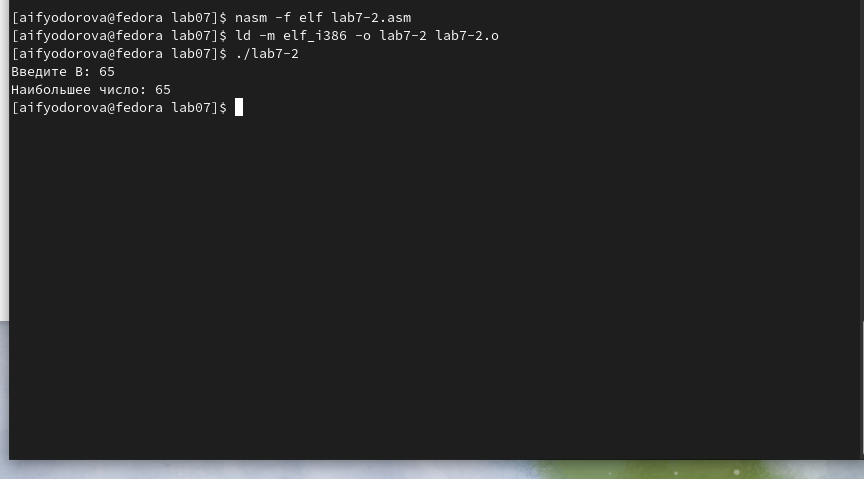
Создание файла lab7-2.asm

Я должна заполнить его кодом из листинга 7.3. (рис.fig:010)



Заполнение файла lab7-2.asm кодом из листинга 7.3

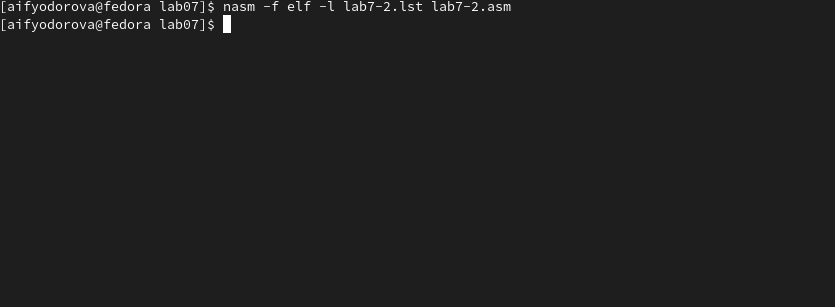
Теперь я снова провожу трансляцию исходного файла в объектный и запускаю, а затем ввожу рандомное число. (рис.fig:011)



Запуск файла lab7-2

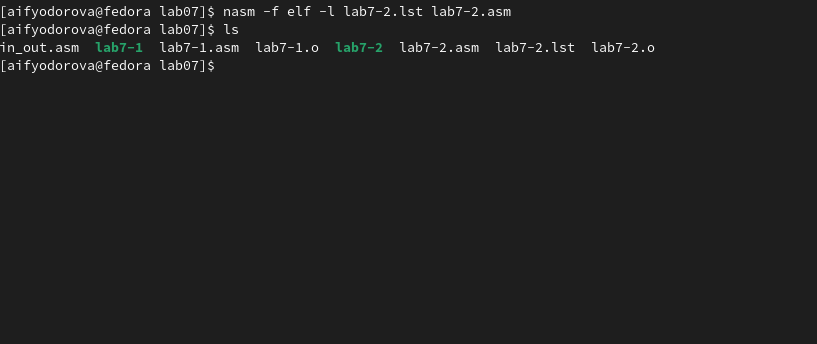
Программа сравнивает введенное пользователем число с “вшитыми” числами 20(в качестве А) и 50(в качестве С).

Теперь я с помощью команды nasm должна получить файл с расширением .lst от файла lab7-2.asm. (рис.fig:012)



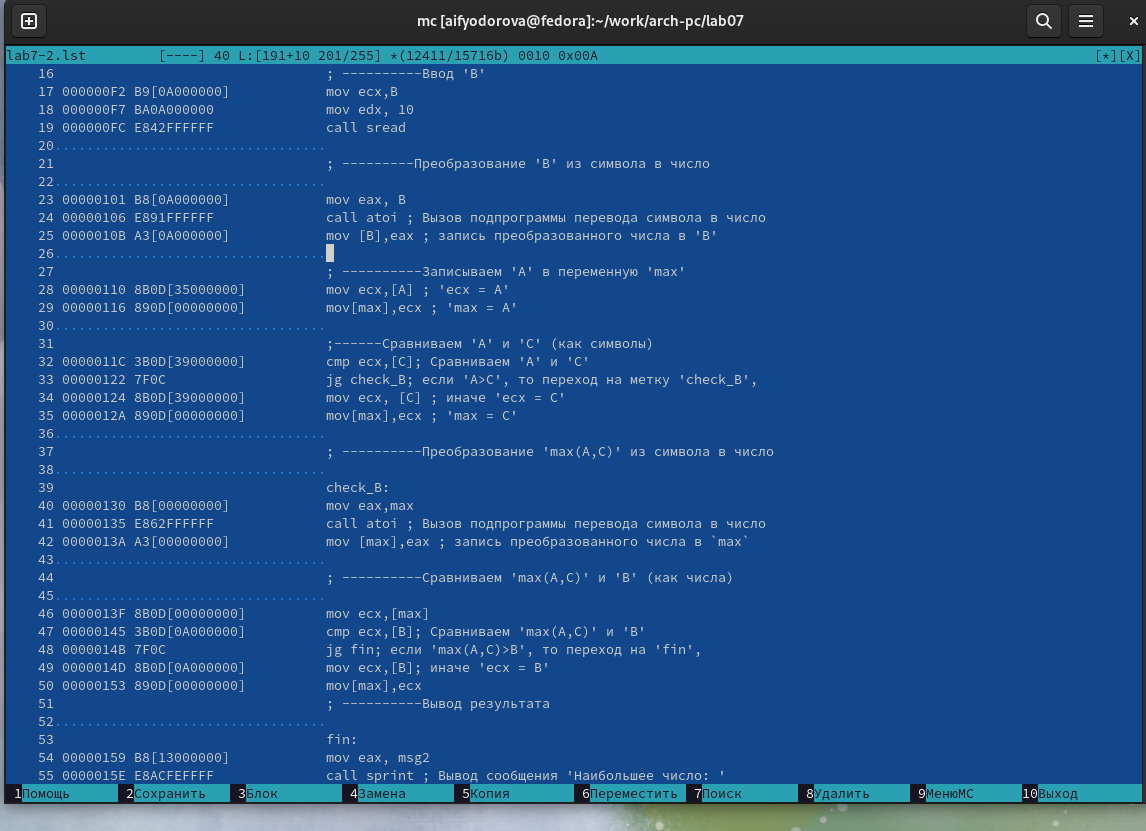
Создание файла lab7-2.lst

Проверяю наличие файла lab7-2.lst с помощью команды ls (рис.fig:013)



Провека наличия файла lab7-2.lst

Открываю данный файл в mc и смотрю на код. (рис.fig:014)



Содержание файла lab7-2.lst

Строка 25 записывает значение по адресу В в регистр eax

В строке 32 с помощью операнда cmp сравнивается значение А, записанное в регистр ecx со значением, записанным по адресу С.

В строке 33 выполняется переход на секцию “check\_B” с помощью jg при условии A>C.

Теперь я удалю любую операнду в файле lab7-2.asm и проверю, что произойдет с файлом lab7-2.lst в реузльтате данного действия(рис.fig:015)

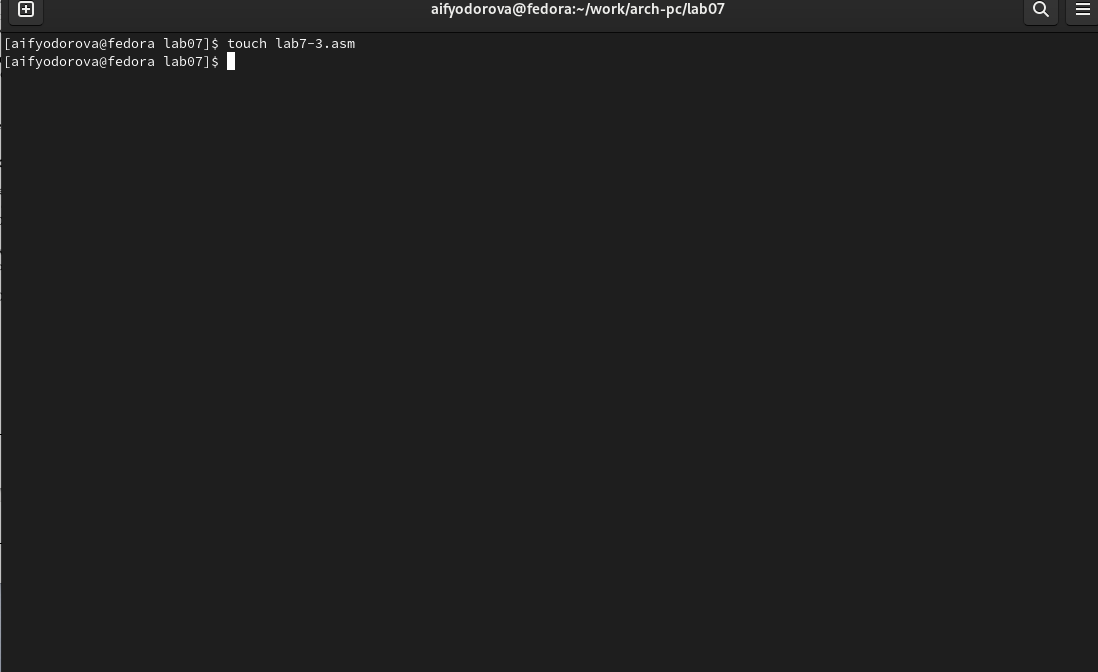


Результат удаления одного операнда в lab7-2.asm

Как вижу, что программа выдает ошибку.

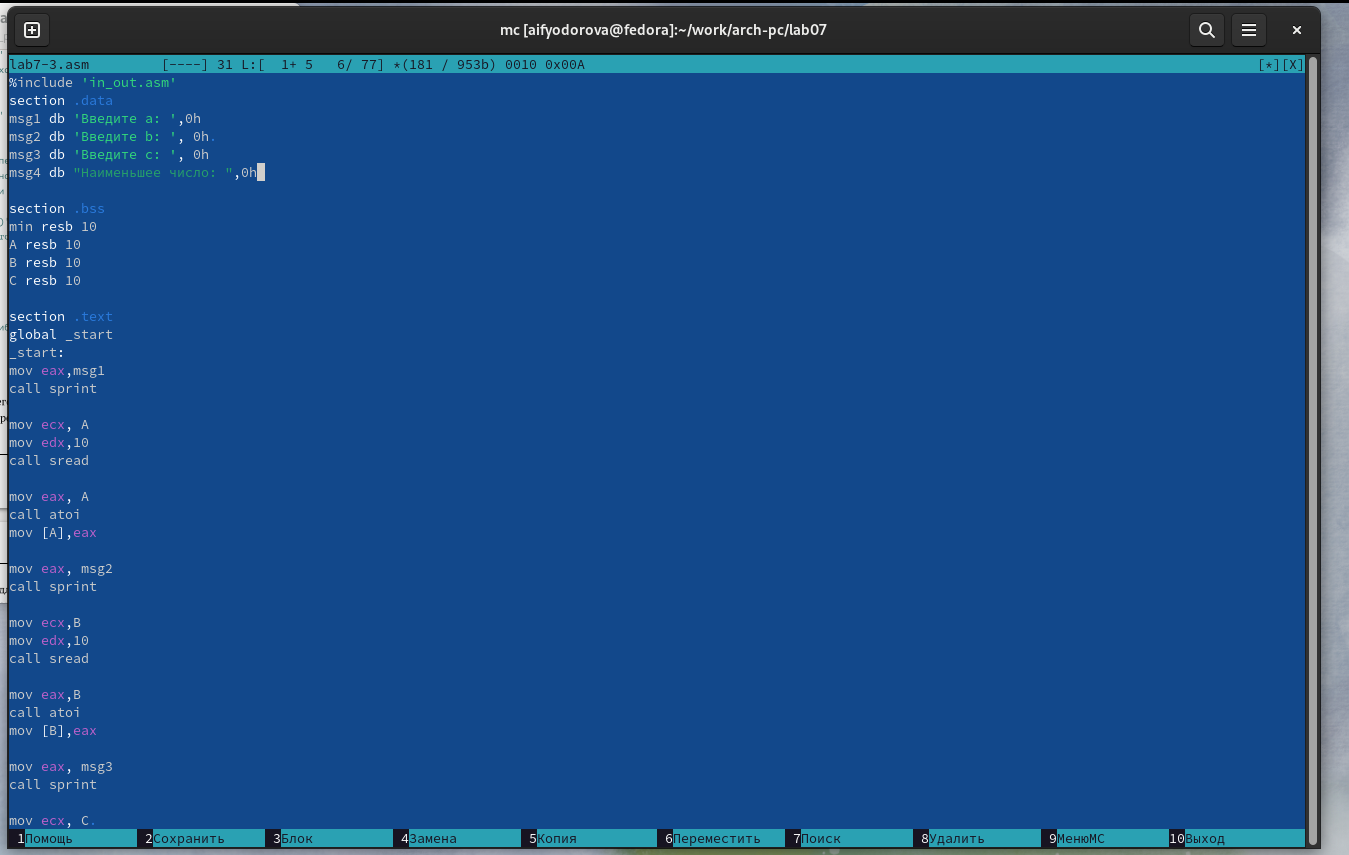
# 5 Выполнение самостоятельной работы

1. Мой вариант в лабораторной работе №6 был 12. Значит, я создаю файл lab7-3.asm для вычисления наименьшего из чисел а=99, b=29, c=26. (рис.fig:016)



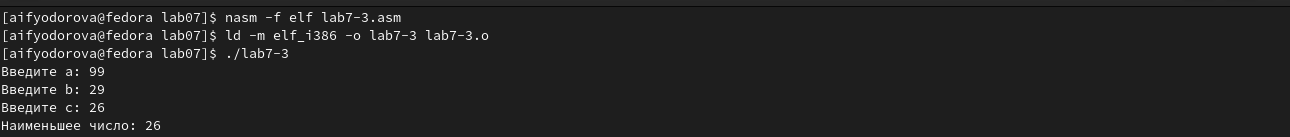
Создание файла lab7-3.asm

Пишу код программы в соответсвии с заданием в самостоятельной работе. (рис.fig:017)



Заполнение файла lab7-3.asm

Проверяю работы написанной программы, создав объектный файл и вводя данные мне числа в 12-ом варианте. (рис.fig:018)



Запуск файла lab7-3

Вижу, что программа выводит правильный ответ.

Код программы:

%include 'in\_out.asm'  
section .data  
msg1 db 'Введите a: ',0h  
msg2 db 'Введите b: ', 0h   
msg3 db 'Введите c: ', 0h  
msg4 db "Наименьшее число: ",0h  
  
section .bss  
min resb 10  
A resb 10  
B resb 10  
C resb 10  
  
section .text  
global \_start  
\_start:  
mov eax,msg1  
call sprint  
  
mov ecx, A  
mov edx,10  
call sread  
  
mov eax, A  
call atoi  
mov [A],eax  
  
mov eax, msg2  
call sprint  
  
mov ecx,B  
mov edx,10  
call sread  
  
mov eax,B  
call atoi  
mov [B],eax  
  
mov eax, msg3  
call sprint  
  
mov ecx, C   
mov edx, 10  
call sread  
  
mov eax, C  
call atoi   
mov [C], eax  
  
mov ecx,[A]  
mov [min],ecx  
  
cmp ecx,[C]; Сравниваем 'A' и 'С'  
jl check\_B; если 'A<C', то переход на метку 'check\_B',  
mov ecx,[C]; иначе 'ecx = C'  
mov [min], ecx  
  
  
check\_B:  
mov ecx,[min]  
cmp ecx,[B]  
jl fin  
mov ecx, [B]; иначе 'ecx = B'  
mov [min], ecx  
  
fin:  
mov eax, msg4  
call sprint  
mov eax,[min]  
call iprintLF ; Вывод 'max(A,B,C)'  
call quit

1. Теперь я создаю файл lab7-4.asm для работы со вторым заданием.(рис.fig:019)



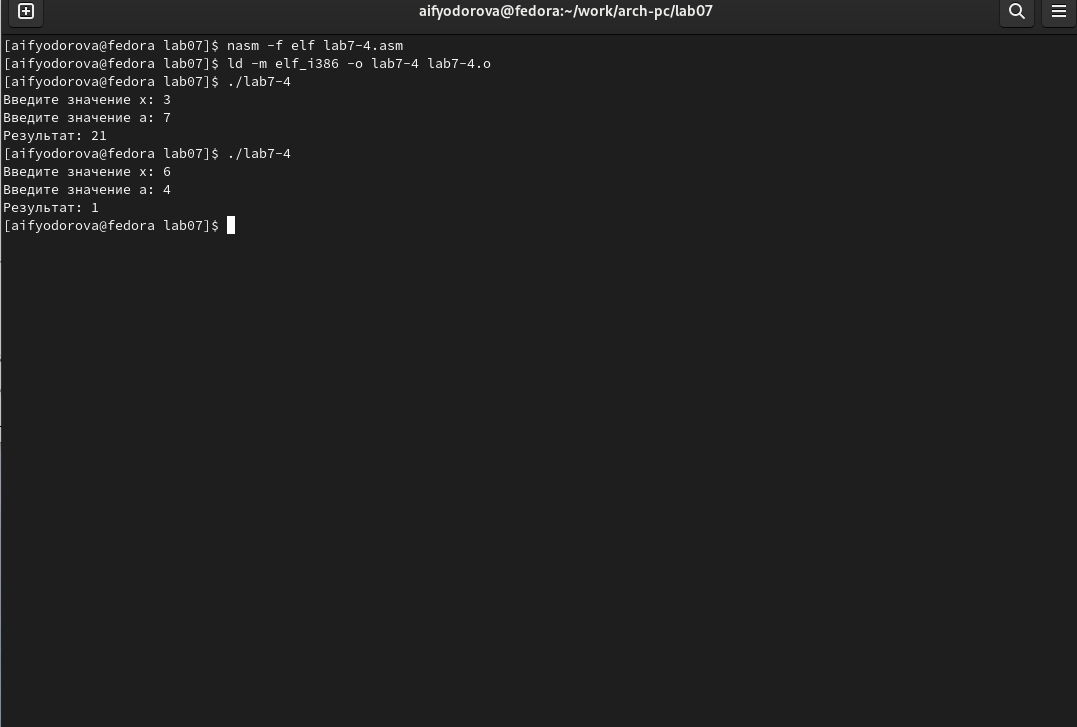
Создание файла lab7-4.asm

Пишу код для вычисления функции f= a\*x при x<5 и f=х-5 при x=>5. (рис.fig:020)



Код в файле для вычисления выражения

Теперь я проверяю работу написанной программы, вводя данные мне значения x и a в материале лабораторной работы.(рис.fig:021)



Проверка работы программы

Вижу, что программа выводит правильный ответ в обоих случаях.

Код программы:

%include 'in\_out.asm'  
SECTION .data  
msg: DB 'Введите значение х: ',0  
msg2: DB 'Введите значение а: ',0  
msg3: DB 'Результат: ', 0  
C dd '5'  
  
SECTION .bss  
r: RESB 10  
x: RESB 10  
A: RESB 10  
  
SECTION .text  
GLOBAL \_start  
\_start:  
 mov eax, msg ; Вызываем программу печати   
 call sprint ; сообщения "Введите значение х:"  
   
 mov ecx, x ; вводим переменную, значение которой вводит пользователь  
 mov edx, 10  
 call sread ; Считываем переменную x  
   
 mov eax, x ; вызов подпрограммы преобразования  
 call atoi ; ASCII кода в число, eax=x  
 mov [x], eax ; запись числа по адресу [x]  
   
 mov eax,msg2  
 call sprint  
   
 mov ecx, A  
 mov edx,10  
 call sread  
   
 mov eax,A  
 call atoi  
 mov [A],eax  
   
 mov eax, C  
 call atoi  
 mov [C], eax  
   
 mov ecx, [C]  
 cmp ecx, [x]  
 jg function\_1 ; переход на другую операцию в случае, что х<5  
 mov eax, [x] ; выполнение операции х-5 в случае, если х=>5  
 sub eax, [C]  
 mov [r], eax ; запись результата по адресу "r"  
 jmp fin ; переход на вывод результата  
   
function\_1:  
 mov eax, [x] ; выполнение операции а\*х   
 mov ebx, [A];   
 mul ebx ;   
 mov [r],eax ; запись результата по адресу "r"  
  
   
   
fin:  
 mov eax,msg3 ; Вызов программы печати  
 call sprint; сообщения 'Результат: '  
 mov eax, [r] ; вызов подпрограммы печати значения  
 call iprintLF ; из 'r' (результат вычисления) в виде символов  
 call quit; вызов подпрограммы завершения

# 6 Выводы

Я изучила команды условного и безусловного переходов и приобрела навыки написания программ с использованием переходов, а также познакомилась с назначением и структурой файла листинга.

# 7 Список литературы

[Лабораторная работа №7](https://esystem.rudn.ru/pluginfile.php/2089087/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%967.%20%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B%20%D0%B1%D0%B5%D0%B7%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B8%20%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B4%D0%BE%D0%B2%20%D0%B2%20Nasm.%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%82%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9..pdf)