Отчёт по лабораторной работе №8

Дисциплина: архитектура компьютера

Барбакова Алиса Саяновна

Содержание

# 1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

# 2 Задание

1. Реализация циклом в NASM
2. Обработка аргументов командной строки
3. Самостоятельное написание программы по материалам лабораторной работы

# 3 Теоретическое введение

## 3.1 Организация стека

Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нём выделяется память для локальных переменных и могут временно храниться значения регистров. Для стека существует две основные операции:  
• добавление элемента в вершину стека (push);  
• извлечение элемента из вершины стека (pop).

## 3.2 Инструкции организации циклов

Для организации циклов существуют специальные инструкции. Для всех инструкций максимальное количество проходов задаётся в регистре ecx. Наиболее простой является инструкция loop. Она позволяет организовать безусловный цикл, типичная структура которого имеет следующий вид:

mov ecx, 100 ; Количество проходов  
NextStep:  
...  
... ; тело цикла  
...  
loop NextStep ; Повторить `ecx` раз от метки NextStep

# 4 Выполнение лабораторной работы

## 4.1 Реализация циклов в NASM

Создаю каталог для программ лабораторной работы №8, перехожу в него и создаю файл lab8-1.asm (рис. -fig. 1).

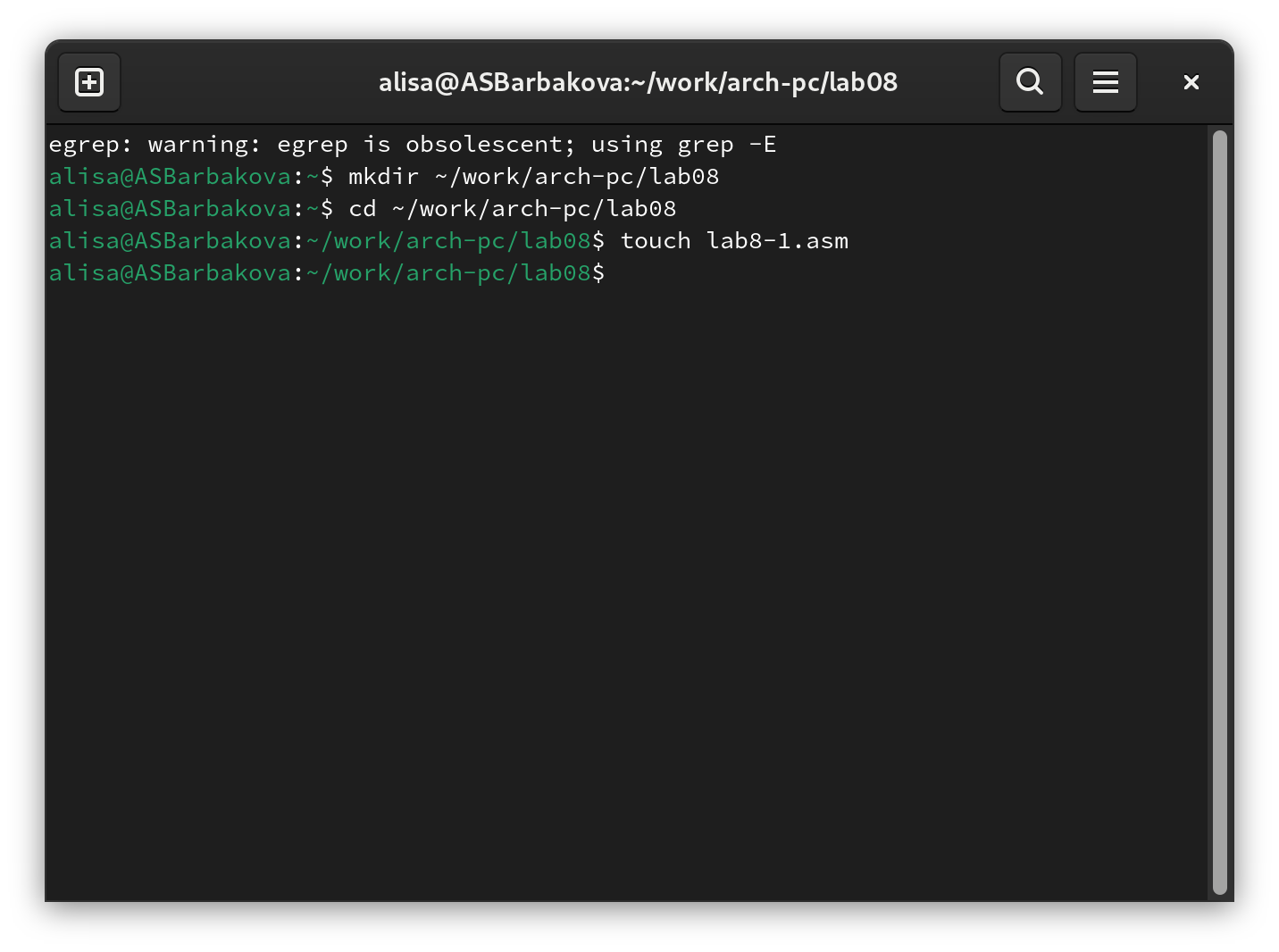


Рис. 1: Создание каталога

Копирую в созданный файл программу из листинга 8.1 (рис. -fig. 2). Копирую в текущий каталог файл in\_out.asm.

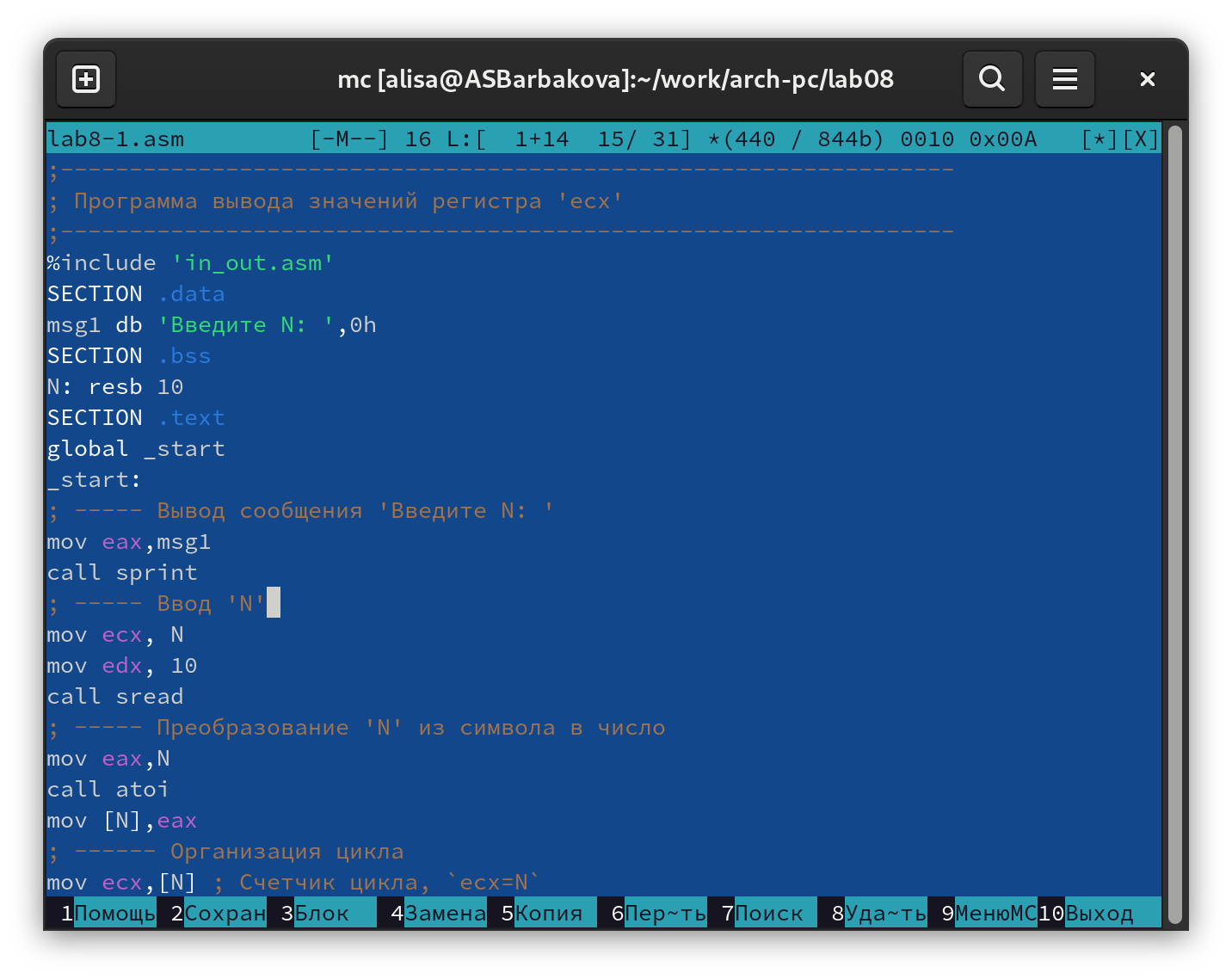


Рис. 2: Копирование программы из листинга

Компилирую и создаю исполняемый файл. Запускаю программу, она показывает работу циклов в NASM (рис. -fig. 3).

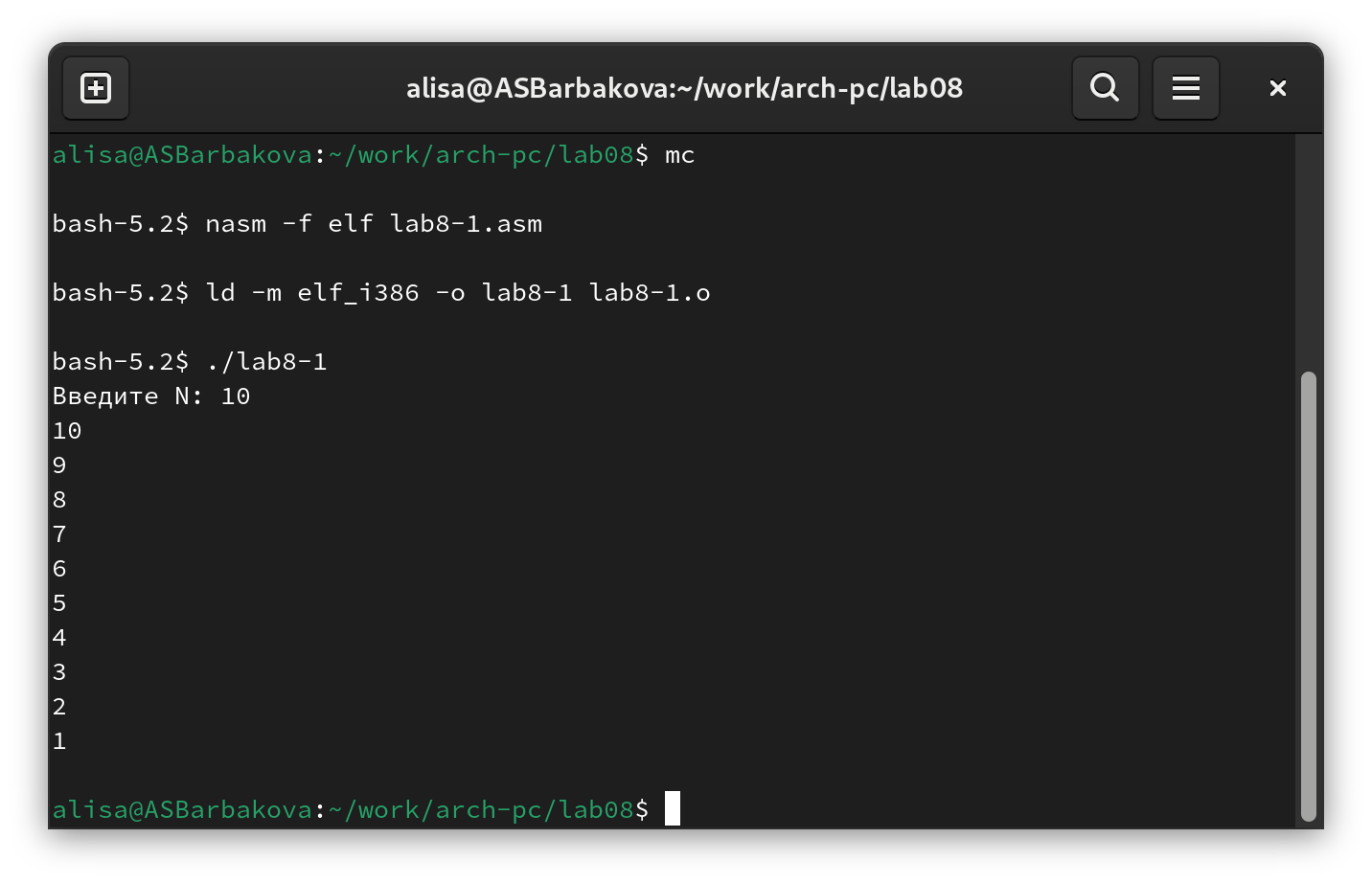


Рис. 3: Запуск программы

Заменяю изначальный текст программы так, что в теле цикла я изменяю значение регистра ecx (рис. -fig. 4).

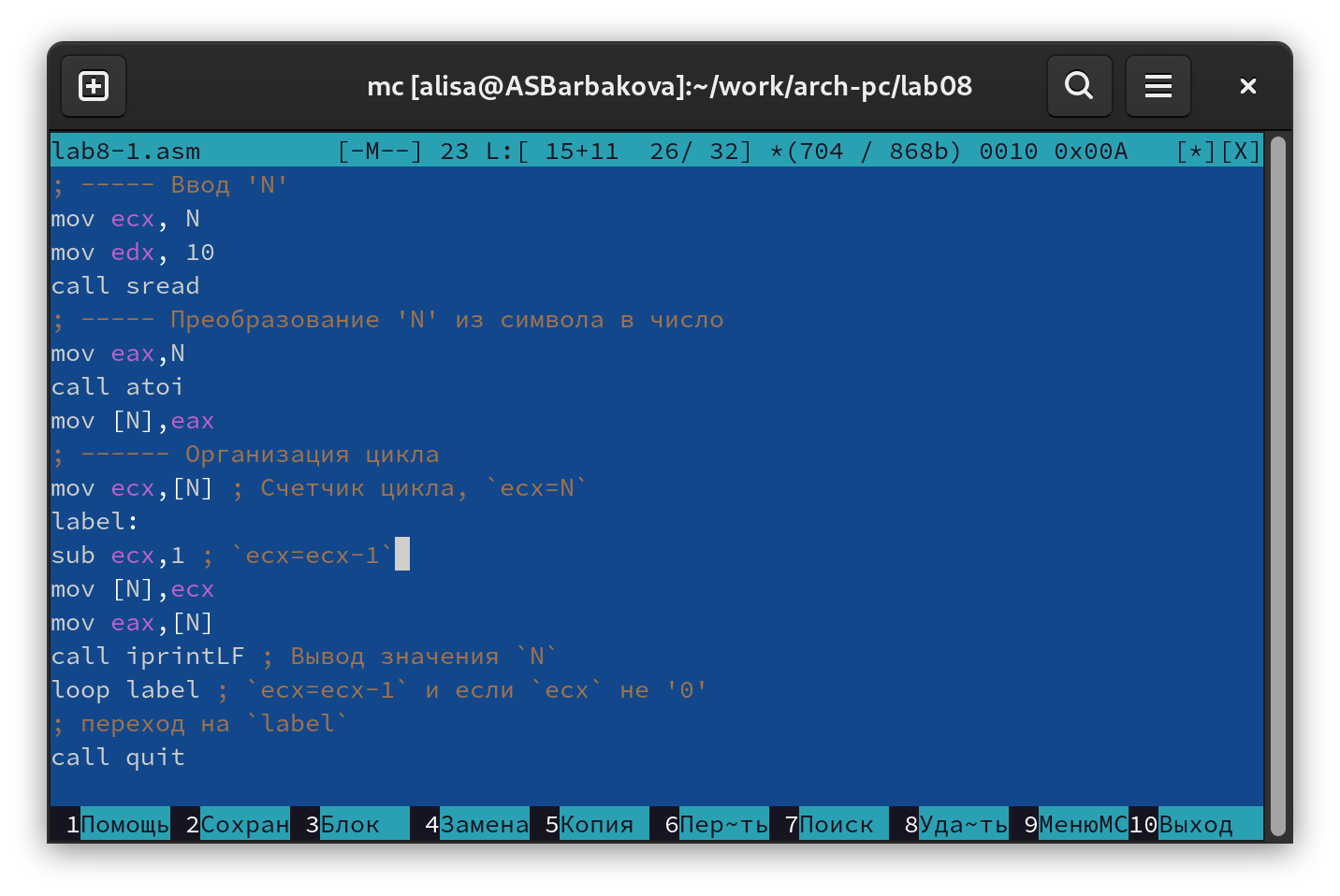


Рис. 4: Изменение программы

Создаю исполняемый файл, проверяю его работу. Из-за того, что теперь регистр ecx на каждой итерации уменьшается на 2 значения, количество итераций уменьшается вдвое (рис. -fig. 5).

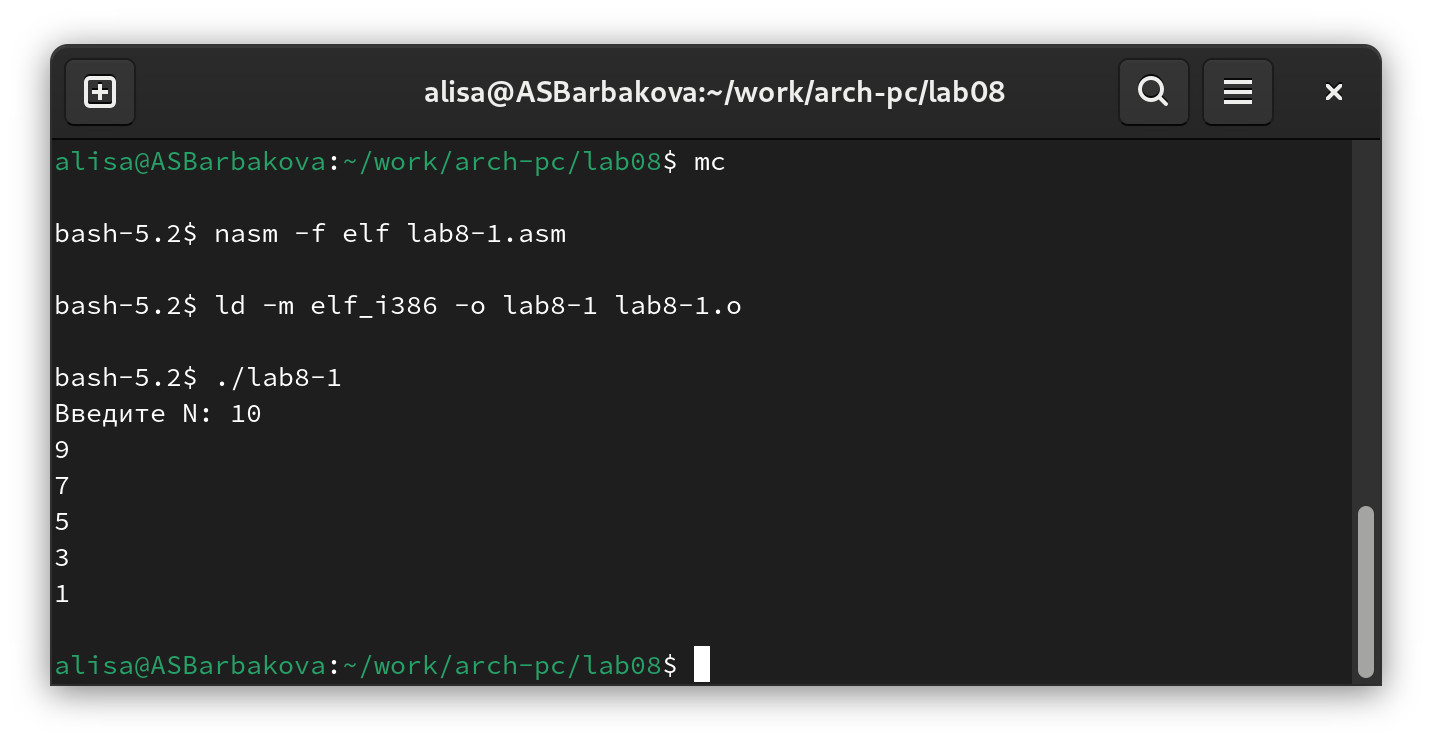


Рис. 5: Запуск измененной программы

Добавляю команды push и pop в программу (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop (рис. -fig. 6).

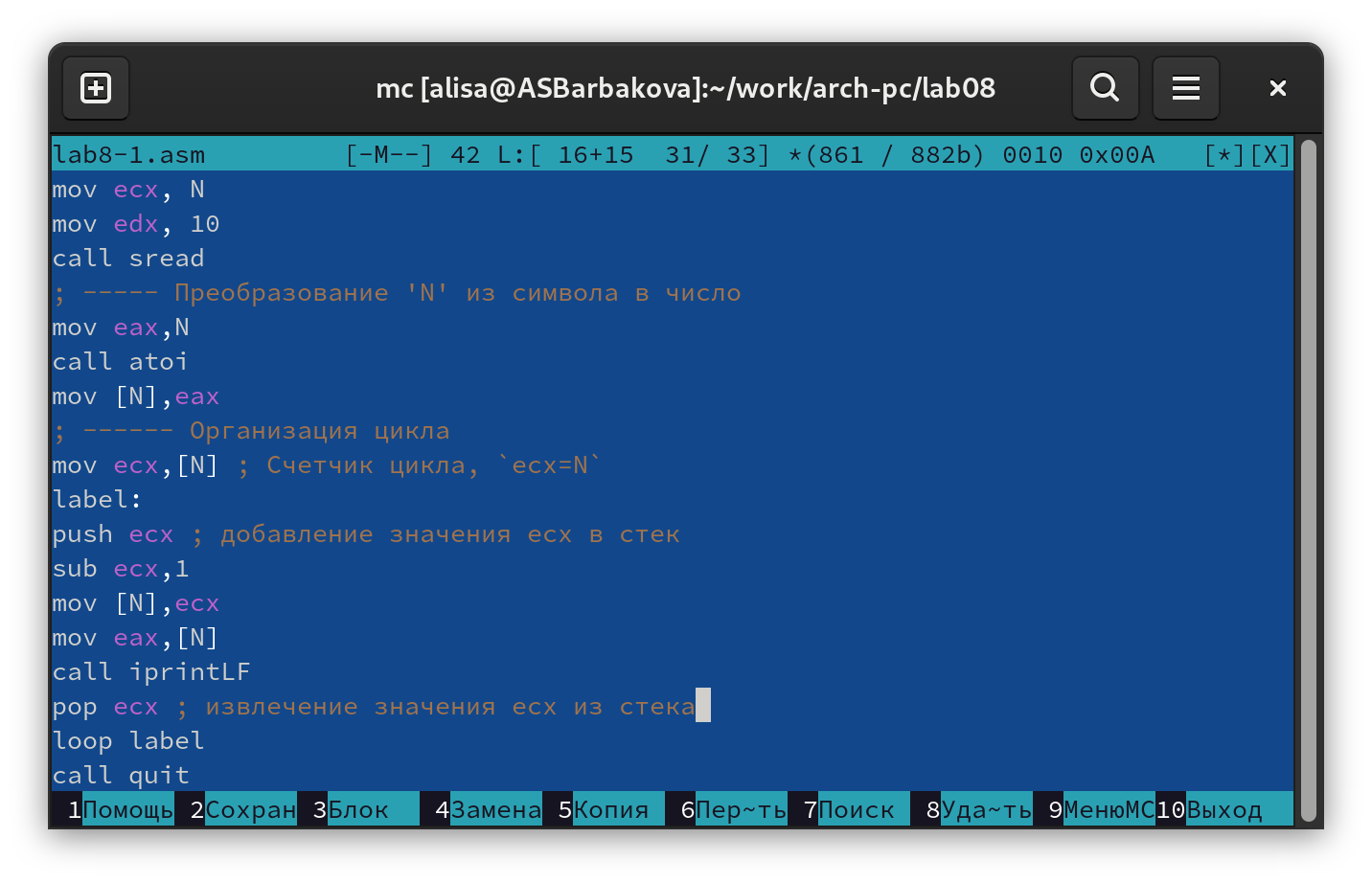


Рис. 6: Добавление push и pop в цикл программы

Запускаю программу (рис. -fig. 7). Теперь количество итераций совпадает введенному N, но произошло смещение выводимых чисел на -1.

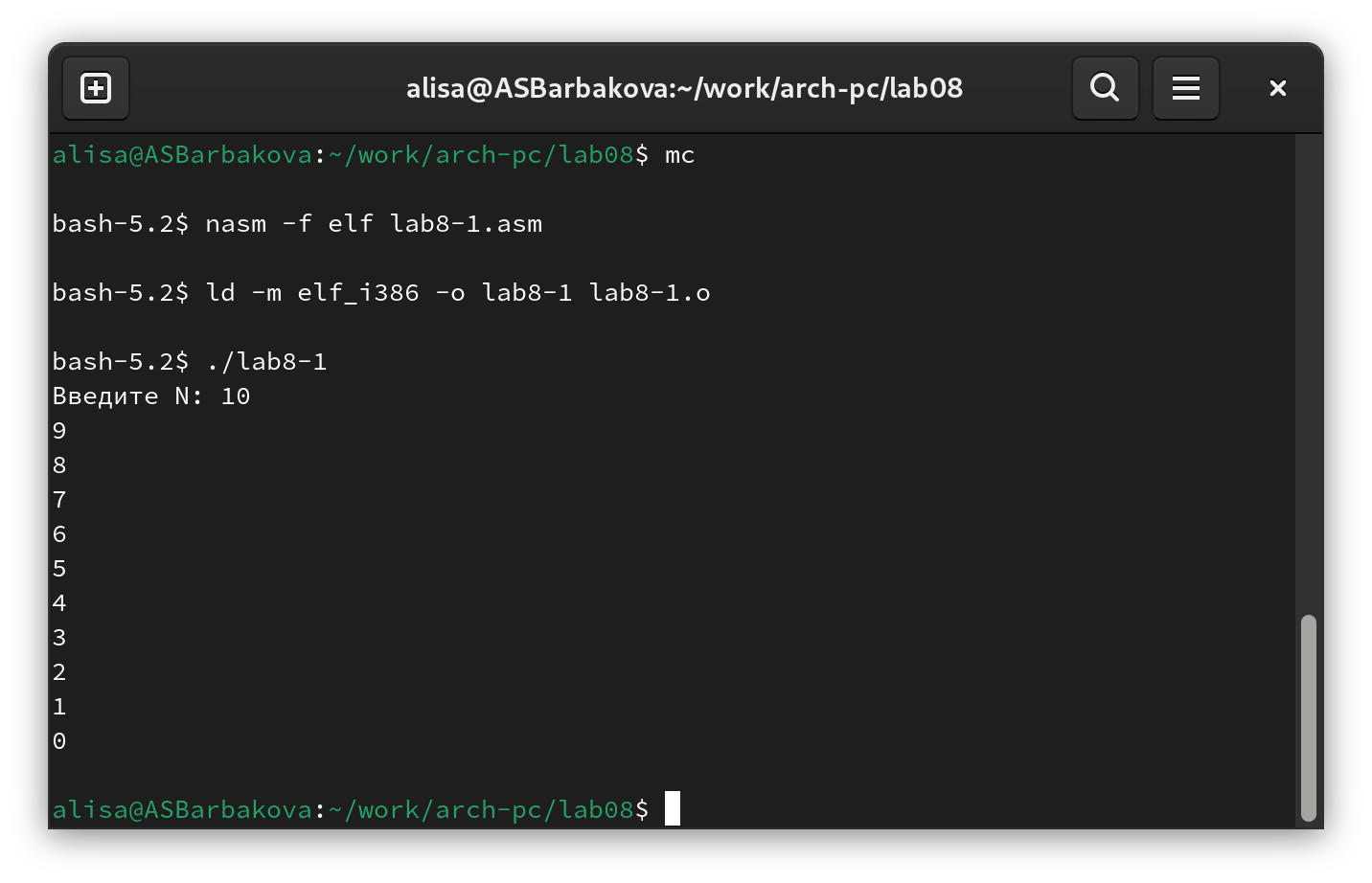


Рис. 7: Запуск измененной программы

## 4.2 Обработка аргументов командной строки

Создаю новый файл lab8-2.asm и копирую в него код из листинга 2 (рис. -fig. 8).

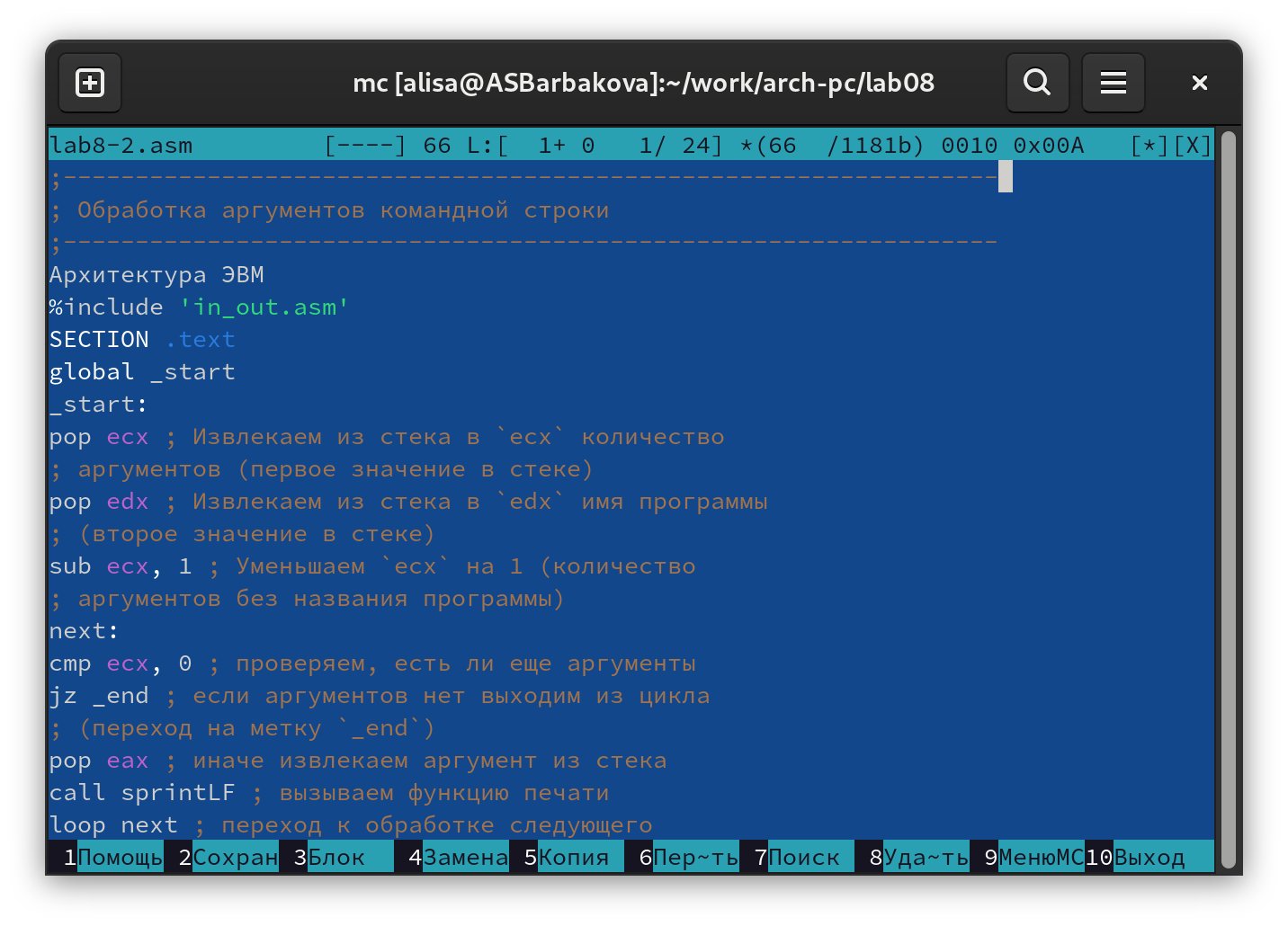


Рис. 8: Копирование программы из листинга

Компилирую файл и запускаю, указав аргументы. Программой было обработано то же количество аргументов, что и было введено (рис. -fig. 9).

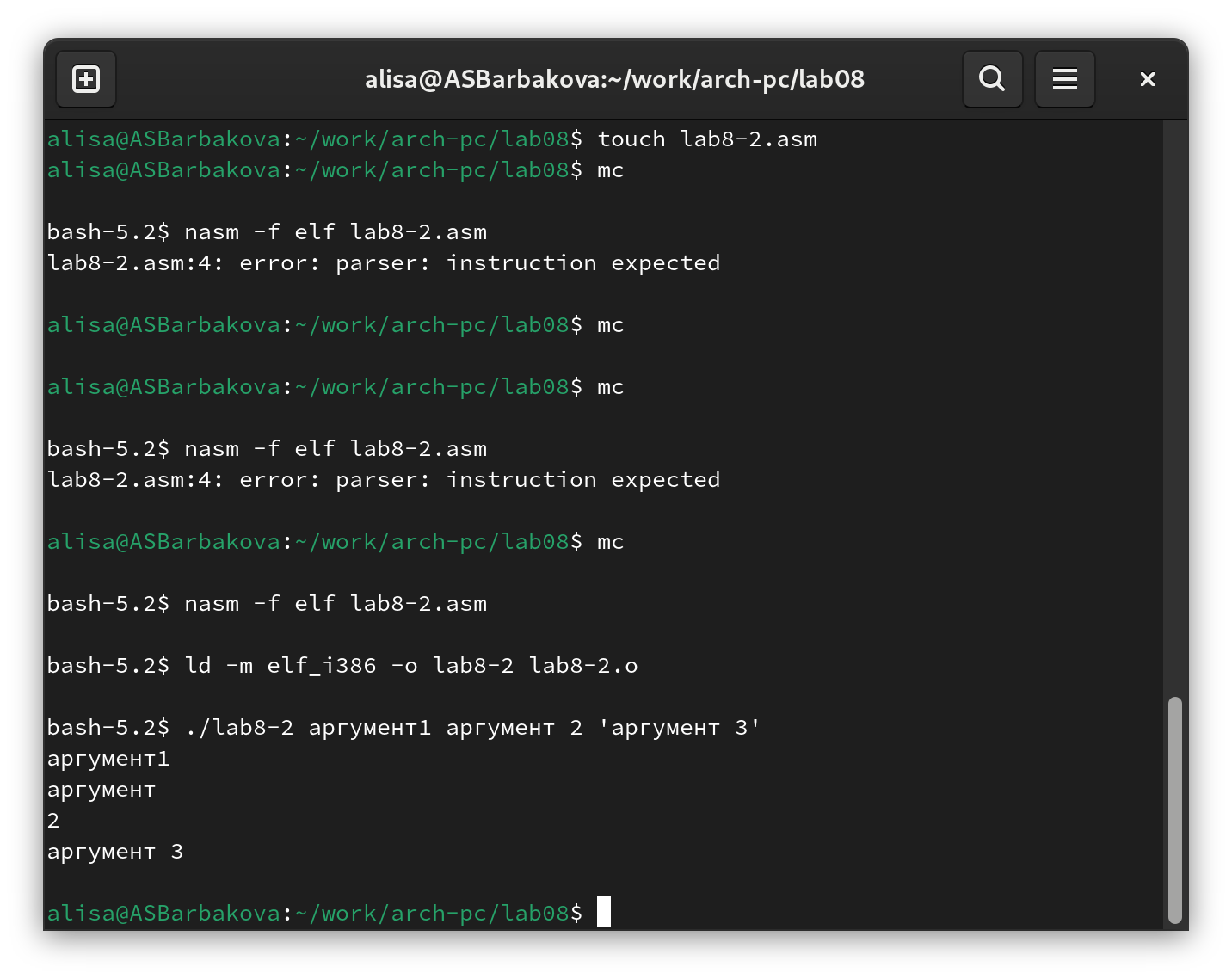


Рис. 9: Запуск второй программы

Создаю новый файл lab8-3.asm для программы и копирую в него код из листинга 3 (рис. -fig. 10).

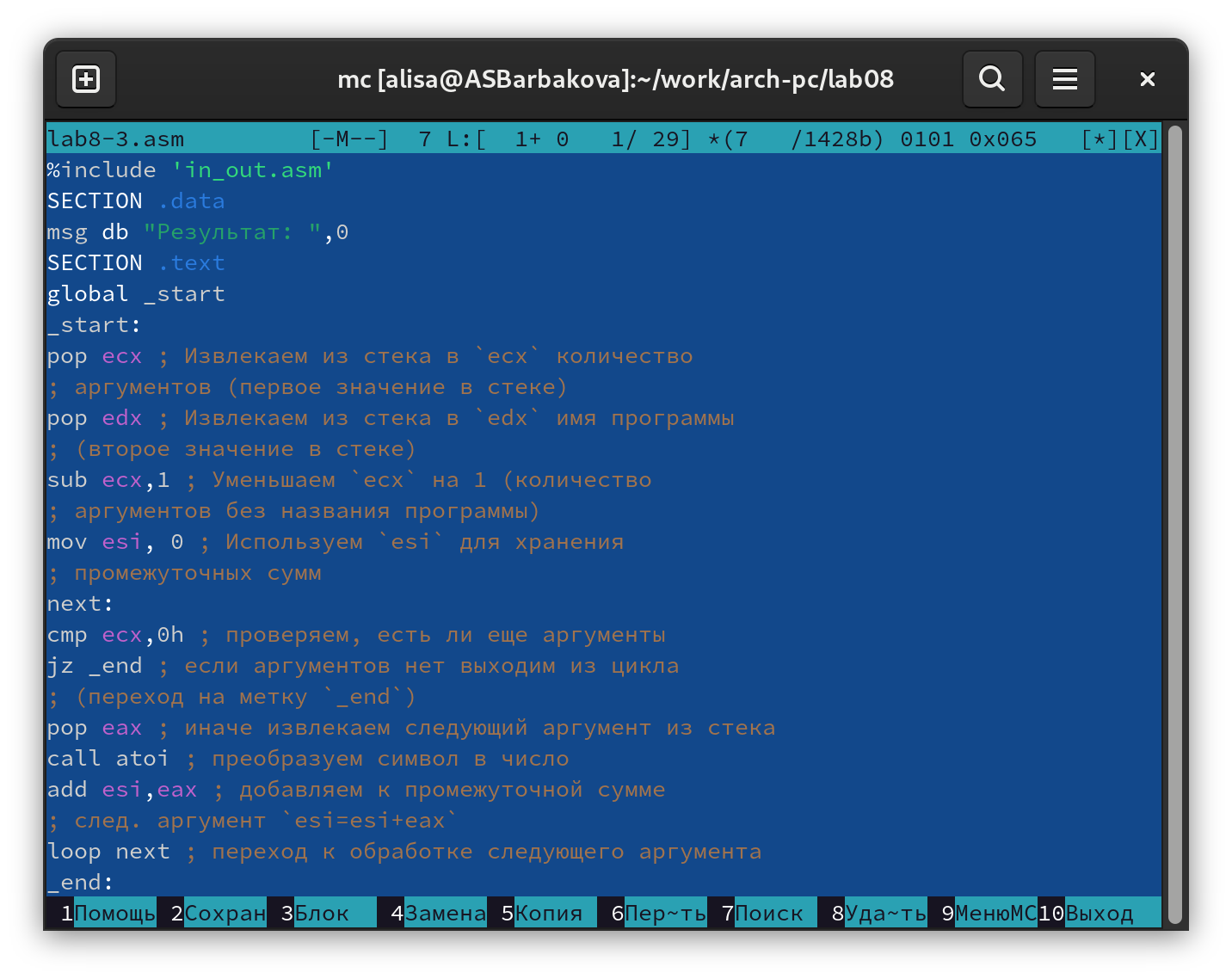


Рис. 10: Копирование программы из третьего листинга

Создаю исполняемый файл и запускаю его. Программа выводит сумму чисел, которые передаются в программу как аргументы (рис. -fig. 11).



Рис. 11: Запуск третьей программы

Изменяю текс программы так, чтобы она вычисляла произведение вводимых аргументов (рис. -fig. 12).

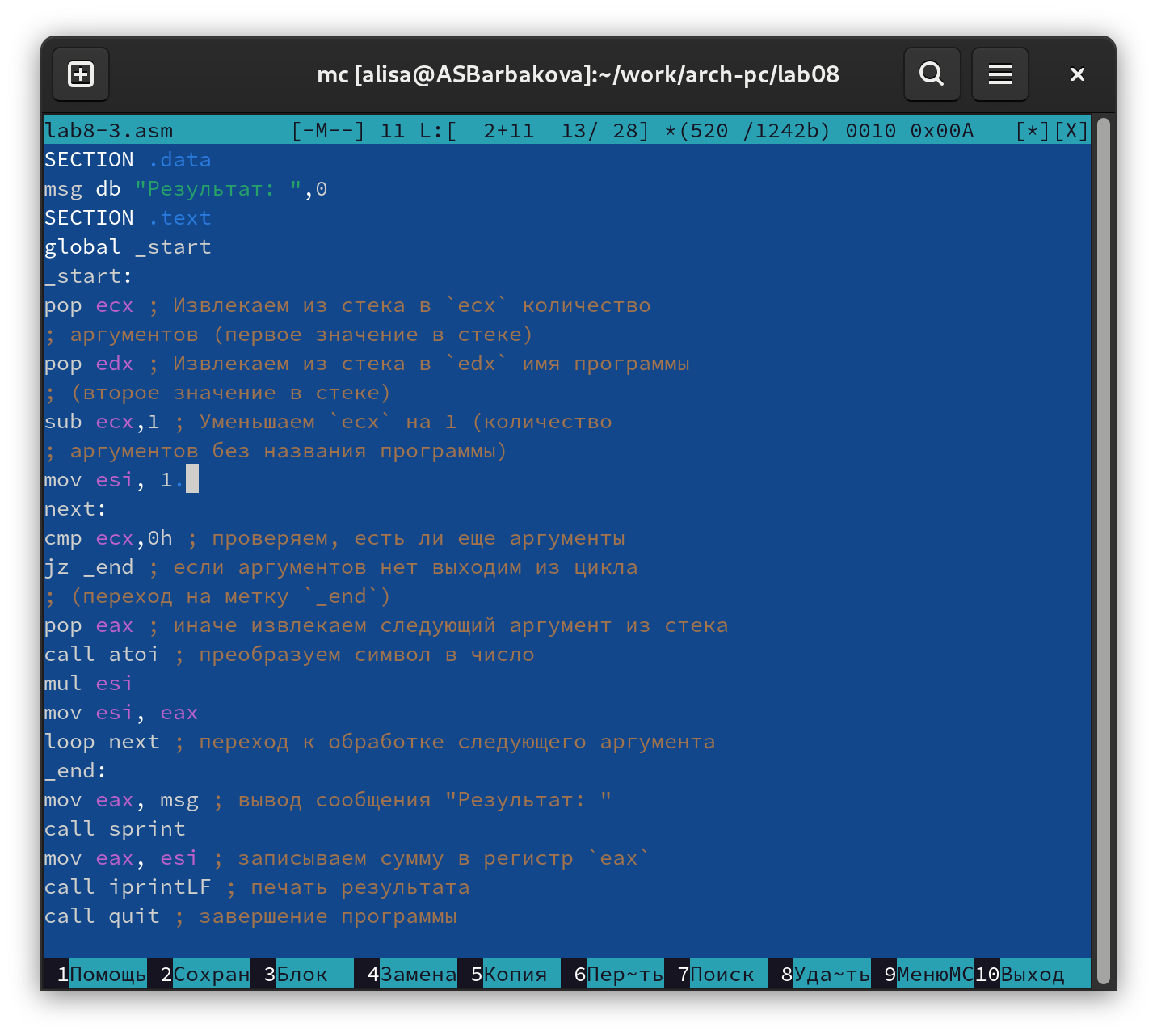


Рис. 12: Изменение третьей программы

Теперь программа умножает данные на вход числа (рис. -fig. 13).

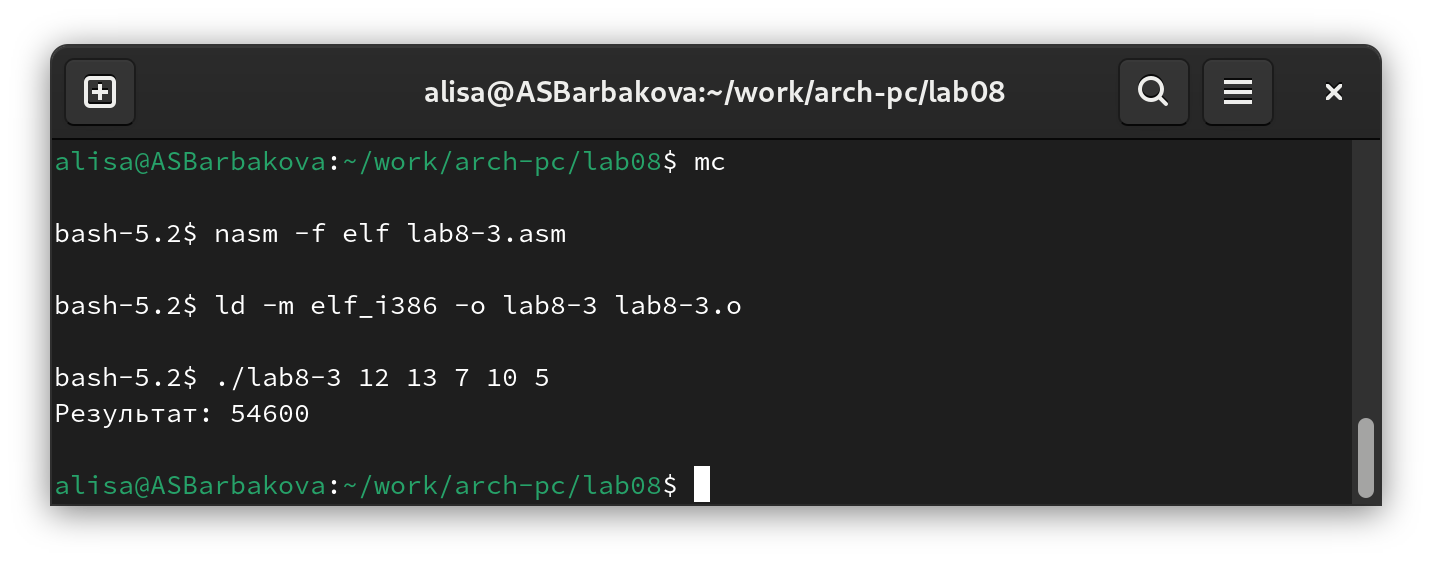


Рис. 13: Запуск измененной третьей программы

## 4.3 Задание для самостоятельной работы

Пишу программму, которая будет находить сумму значений для функции f(x) = 7 + 2x из 8-го варианта, полученного в 6-ой лабораторной работе (рис. -fig. 14).

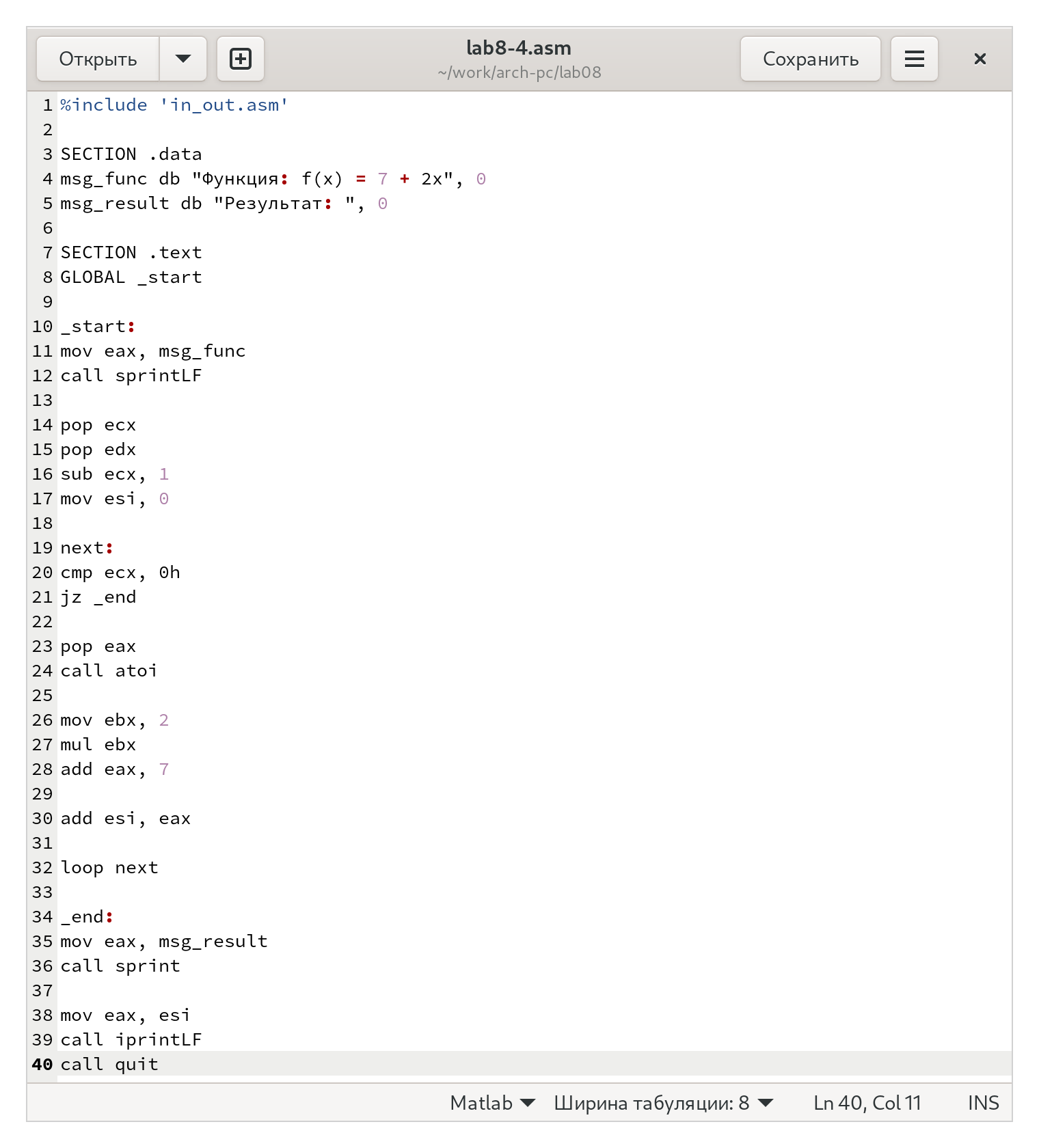


Рис. 14: Написание программы для самостоятельной работы

Код программы:

%include 'in\_out.asm'  
.  
SECTION .data.  
msg\_func db "Функция: f(x) = 7 + 2x", 0.  
msg\_result db "Результат: ", 0.  
  
SECTION .text.  
GLOBAL \_start.  
  
\_start:.  
mov eax, msg\_func.  
call sprintLF  
  
pop ecx  
pop edx.  
sub ecx, 1  
mov esi, 0  
  
next:.  
cmp ecx, 0h  
jz \_end  
pop eax  
call atoi  
  
mov ebx, 2  
mul ebx  
add eax, 7  
  
add esi, eax  
  
loop next  
  
\_end:  
mov eax, msg\_result.  
call sprint  
mov eax, esi.  
call iprintLF.  
call quit.

Проверяю работу программы, указав в качестве аргумента несколько чисел (рис. -fig. 15). Программа работает верно.



Рис. 15: Запуск программы для самостоятельной работы

# 5 Выводы

В результате выполнения данной лабораторной работы я приобрела навыки написания программ с использованием циклов а также научилась обрабатывать аргументы командной строки.

# Список литературы

1. [Курс на ТУИС](https://esystem.rudn.ru/course/view.php?id=112)
2. [Лабораторная работа №8](https://esystem.rudn.ru/pluginfile.php/2089095/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%968.%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%D1%86%D0%B8%D0%BA%D0%BB%D0%B0.%20%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0%20%D0%B0%D1%80%D0%B3%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%20%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9%20%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8..pdf)