Отчёт по лабораторной работе №8

дисциплина: Архитектура компьютера

Баранова Анна Андреевна

Содержание

# 1 Цель работы

Приобрести навыки написания программ с использованием циклов и обработкой аргументов командной строки.

# 2 Задание

В ходе выполнения данной лабораторной работы необходимо изучить:

* Организацию стека;
* реализацию циклов в NASM;
* обработку аргументов командной строки.

Выполнив эту работу, мы приобретём навыки написания программ с использованием циклов и обработкой аргументов командной строки.

# 3 Теоретическое введение

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 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.

Иструкция loop выполняется в два этапа. Сначала из регистра ecx вычитается единица и его значение сравнивается с нулём. Если регистр не равен нулю, то выполняется переход к указанной метке. Иначе переход не выполняется и управление передаётся команде, которая следует сразу после команды loop.

При разработке программ иногда встает необходимость указывать аргументы, которые будут использоваться в программе, непосредственно из командной строки при запуске программы.

При запуске программы в NASM аргументы командной строки загружаются в стек в обратном порядке, кроме того в стек записывается имя программы и общее количество аргументов. Последние два элемента стека для программы, скомпилированной NASM, – это всегда имя программы и количество переданных аргументов.

Таким образом, для того чтобы использовать аргументы в программе, их просто нужно извлечь из стека. Обработку аргументов нужно проводить в цикле. Т.е. сначала нужно извлечь из стека количество аргументов, а затем циклично для каждого аргумента выполнить логику программы.

# 4 Выполнение лабораторной работы

## 4.1 Реализация циклов в NASM

Создадим каталог для программ лабораторной работы № 8, перейдём в него и создадим файл lab8-1.asm и также создадим копию файла in\_out.asm (рис. 1), (рис. 2).

Рис. 1: Создание каталога для программ лабораторной работы № 8 и создание в нём файла lab8-1.asm

Рис. 1: Создание каталога для программ лабораторной работы № 8 и создание в нём файла lab8-1.asm

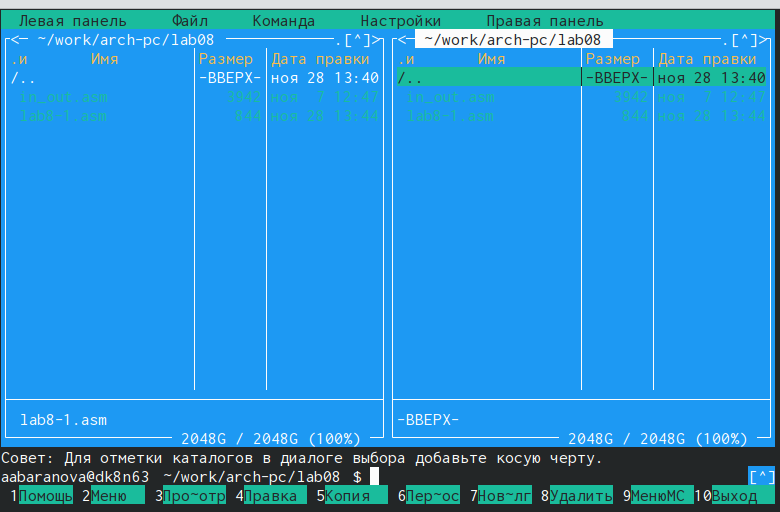


Рис. 2: Копирование файла in\_out.asm в каталог с файлом lab8-1.asm с помощью функциональной клавиши F5

Введём в файл lab8-1.asm текст программы (рис. 3).

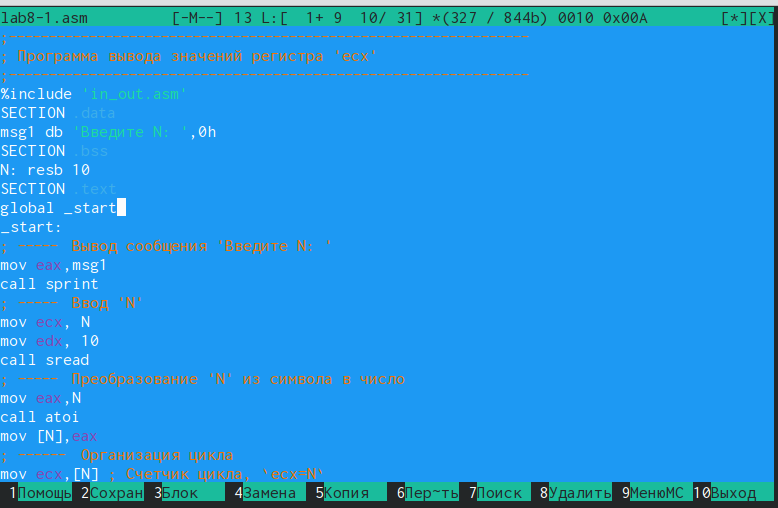


Рис. 3: Изменения в файле lab8-1.asm

Создадим исполняемый файл и запустим его (рис. 4), (рис. 5).

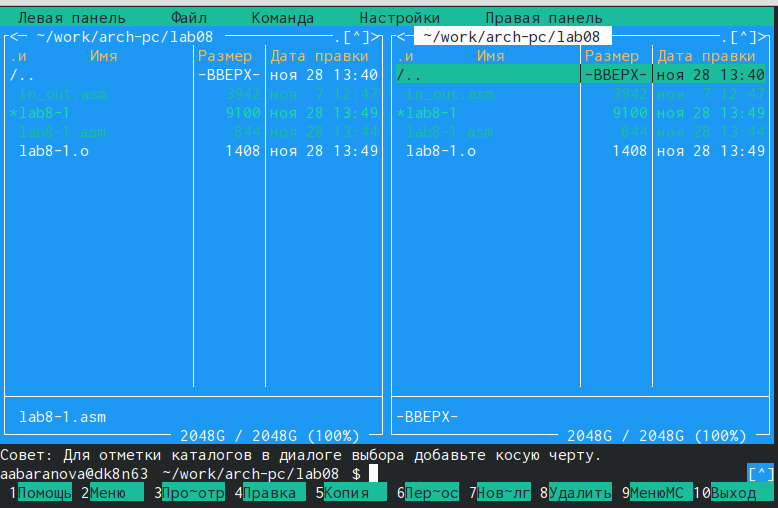


Рис. 4: Создание исполняемого файла и его запуск

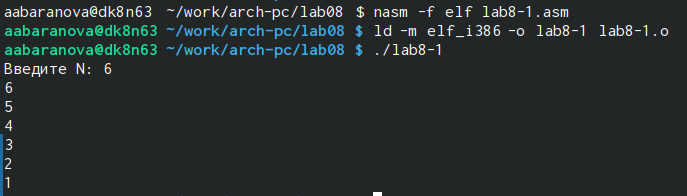


Рис. 5: Создание исполняемого файла и его запуск

Изменим текст программы файла lab8-1.asm (рис. 6).

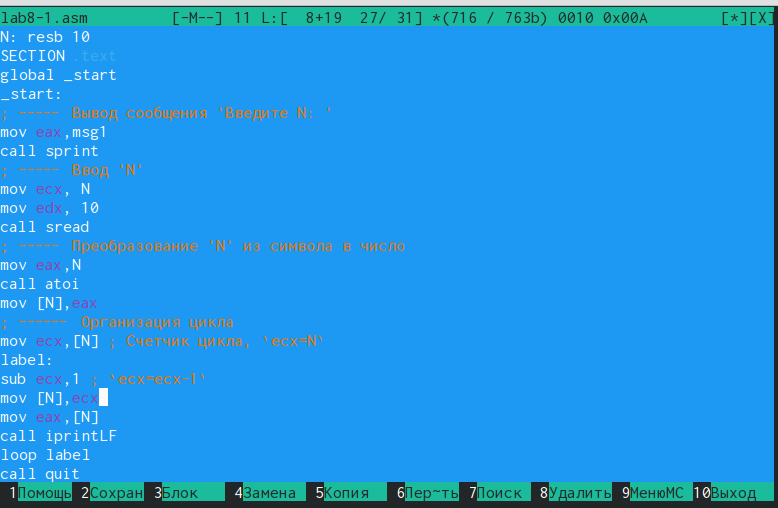


Рис. 6: Изменения в файле lab8-1.asm

Создадим исполняемый файл и запустим его (рис. 7).

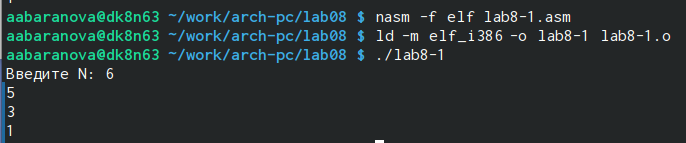


Рис. 7: Создание исполняемого файла и его запуск

Снова изменим текст программы файла lab8-1.asm, создадим исполняемый файл и запустим его (рис. 8), (рис. 9).

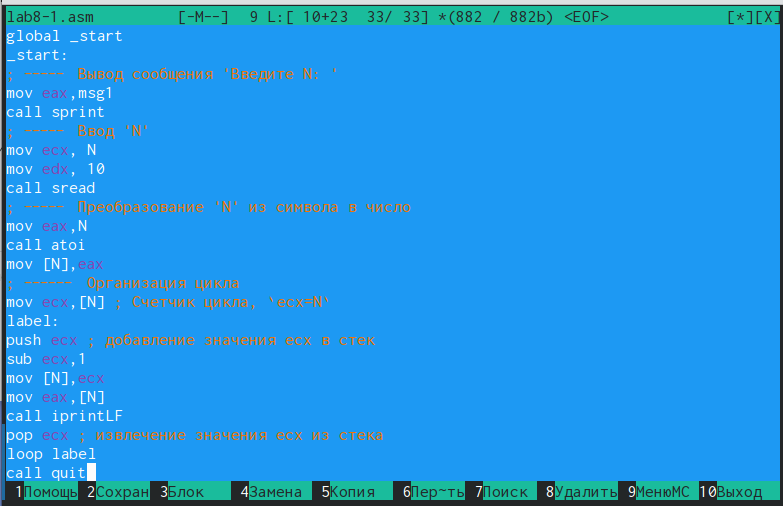


Рис. 8: Изменения в файле lab8-1.asm

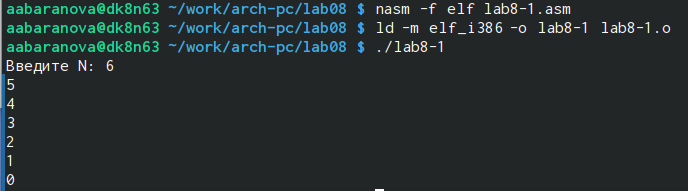


Рис. 9: Создание исполняемого файла и его запуск

## 4.2 Обработка аргументов командной строки

Создадим файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 (рис. 10), (рис. 11).

Рис. 10: Создание файла lab8-2.asm

Рис. 10: Создание файла lab8-2.asm

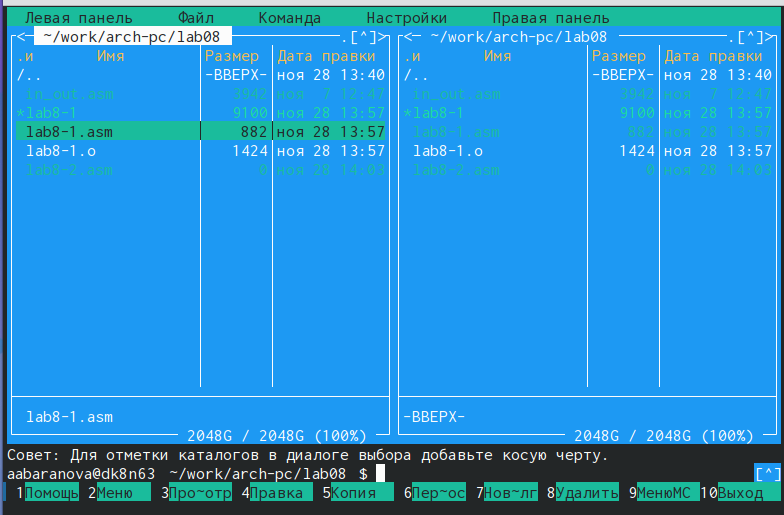


Рис. 11: Создание файла lab8-2.asm

Введём в файл lab8-2.asm текст программы (рис. 12).

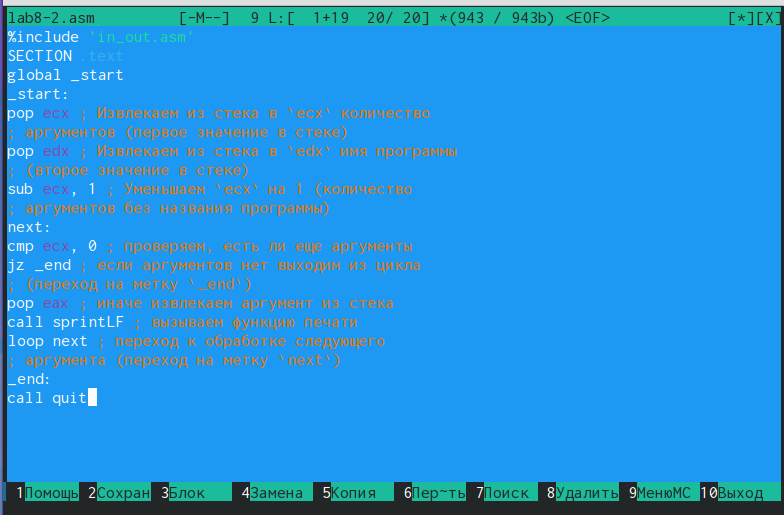


Рис. 12: Изменения в файле lab8-2.asm

Создадим исполняемый файл и запустим его, указав аргументы (рис. 13), (рис. 14).

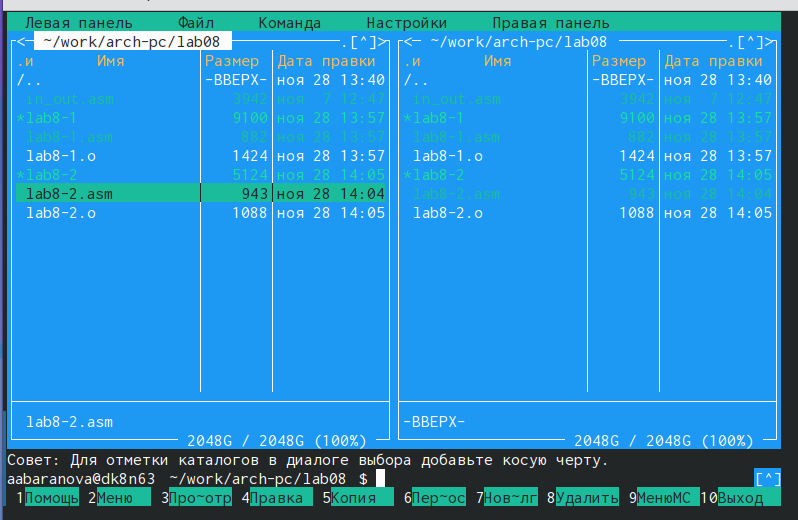


Рис. 13: Создание исполняемого файла и его запуск

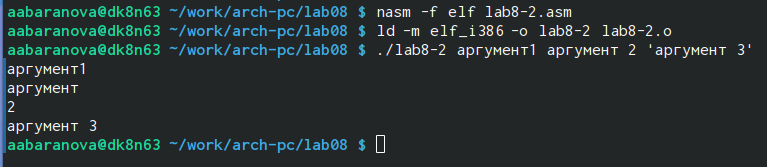


Рис. 14: Создание исполняемого файла и его запуск

Создадим файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 (рис. 15), (рис. 16).

Рис. 15: Создание файла lab8-3.asm

Рис. 15: Создание файла lab8-3.asm

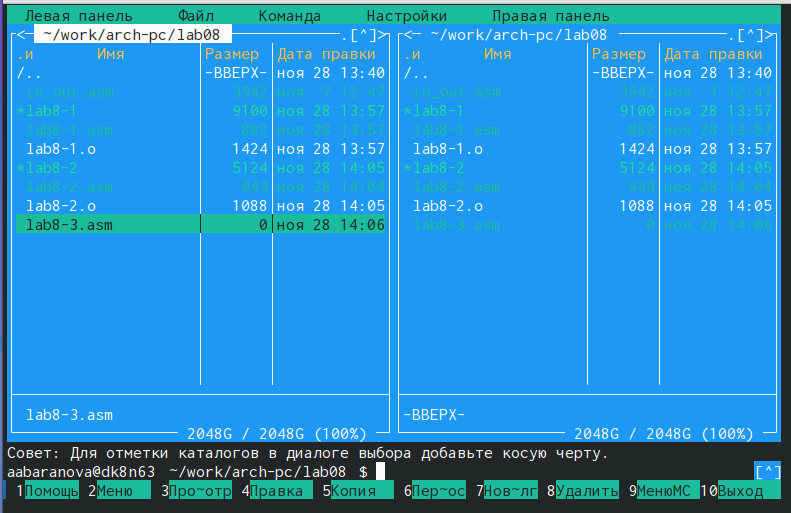


Рис. 16: Создание файла lab8-3.asm

Введём в файл lab8-3.asm текст программы (рис. 17).

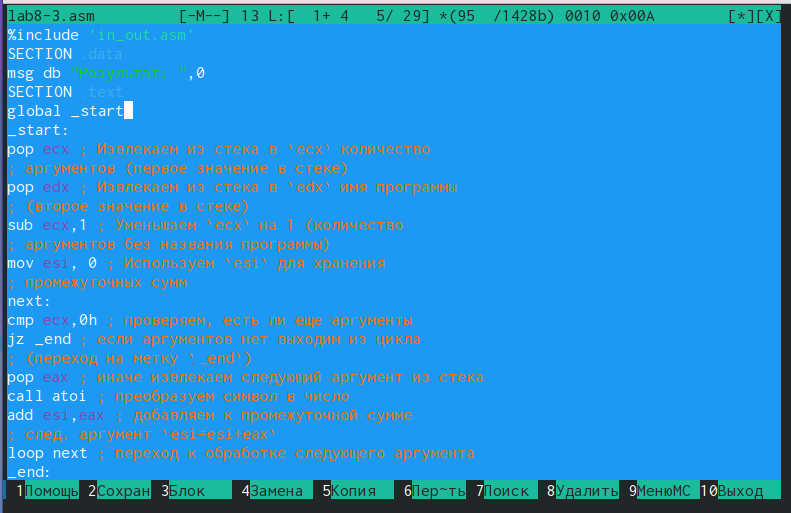


Рис. 17: Изменения в файле lab8-3.asm

Создадим исполняемый файл и запустим его, указав аргументы (рис. 18), (рис. 19).

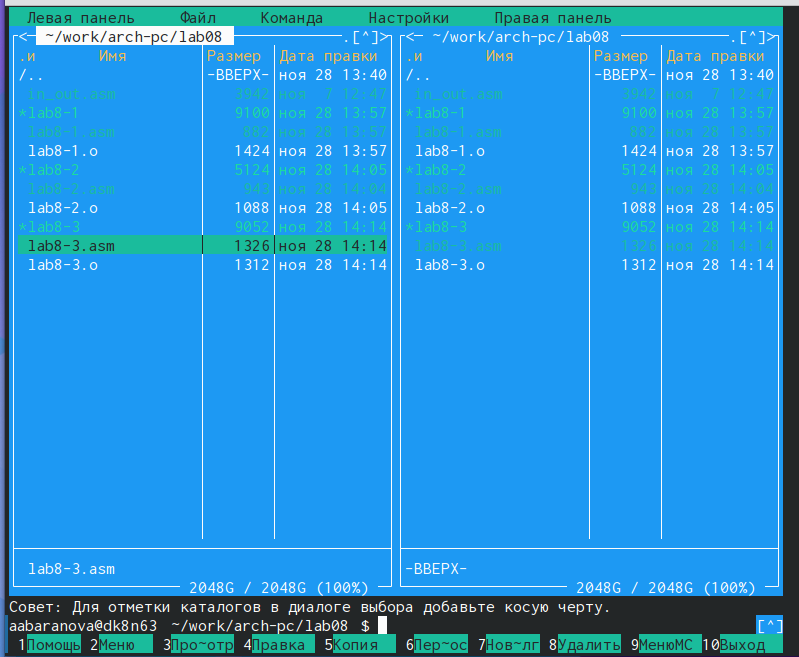


Рис. 18: Создание исполняемого файла и его запуск

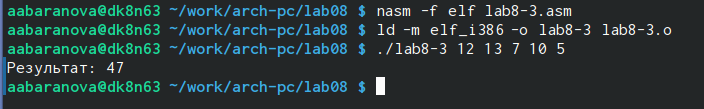


Рис. 19: Создание исполняемого файла и его запуск

Изменим в файле lab8-3.asm текст программы для вычисления произведения аргументов командной строки (рис. 20).

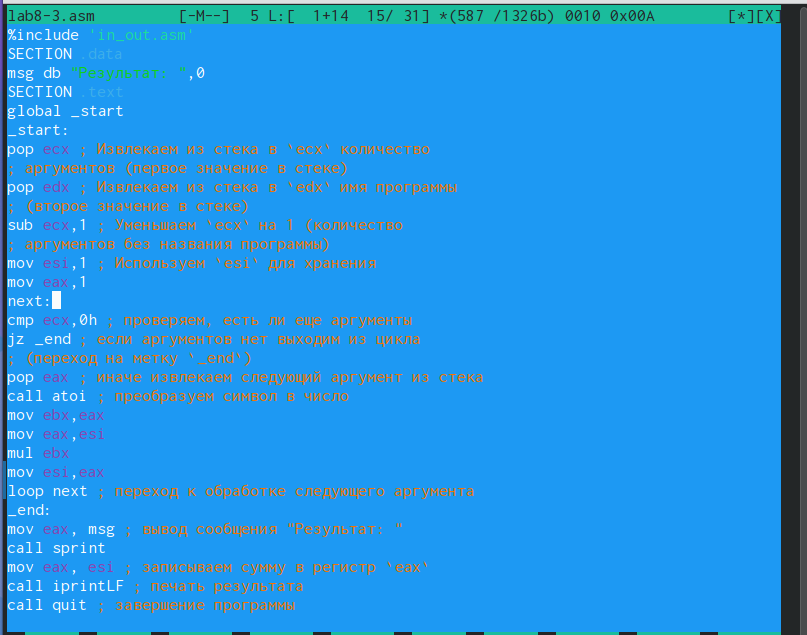


Рис. 20: Изменения в файле lab8-3.asm

Создадим исполняемый файл и запустим его, указав аргументы (рис. 21).

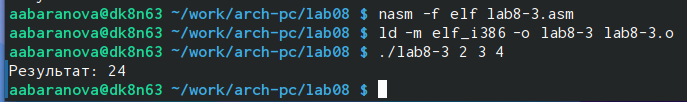


Рис. 21: Создание исполняемого файла и его запуск

## 4.3 Задание для самостоятельной работы

Напишем программу, которая находит сумму значений функции f(x) для x = x\_1, x\_2,…, x\_n, т.е. программа должна выводить f(x\_1) + f(x\_2) + … + f(x\_n). Значения x\_i передаются как аргументы. Создадим исполняемый файл и проверим его работу на нескольких наборах x = x\_1, x\_2,…, x\_n. (рис. 22), (рис. 23), (рис. 24), (рис. 25).

Рис. 22: Создание файла lab8-4.asm

Рис. 22: Создание файла lab8-4.asm

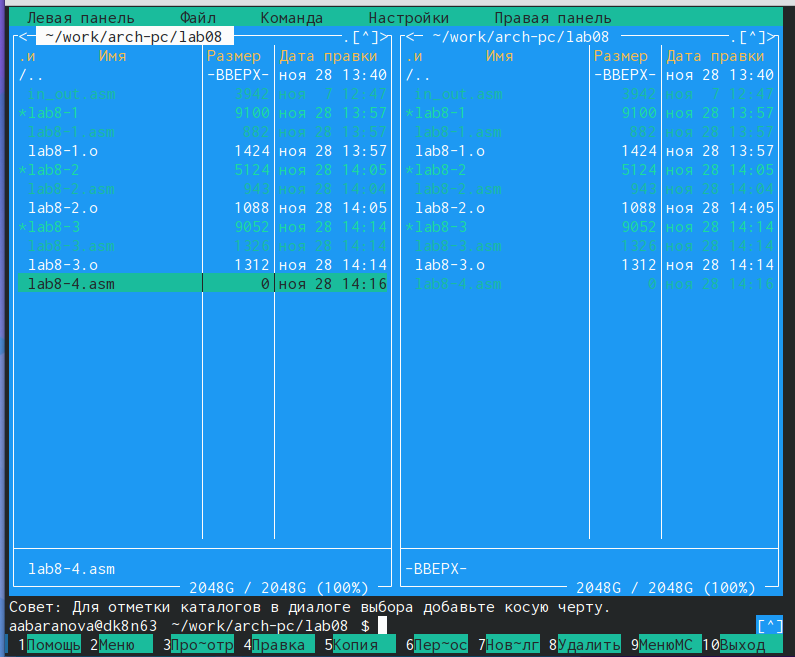


Рис. 23: Создание файла lab8-4.asm

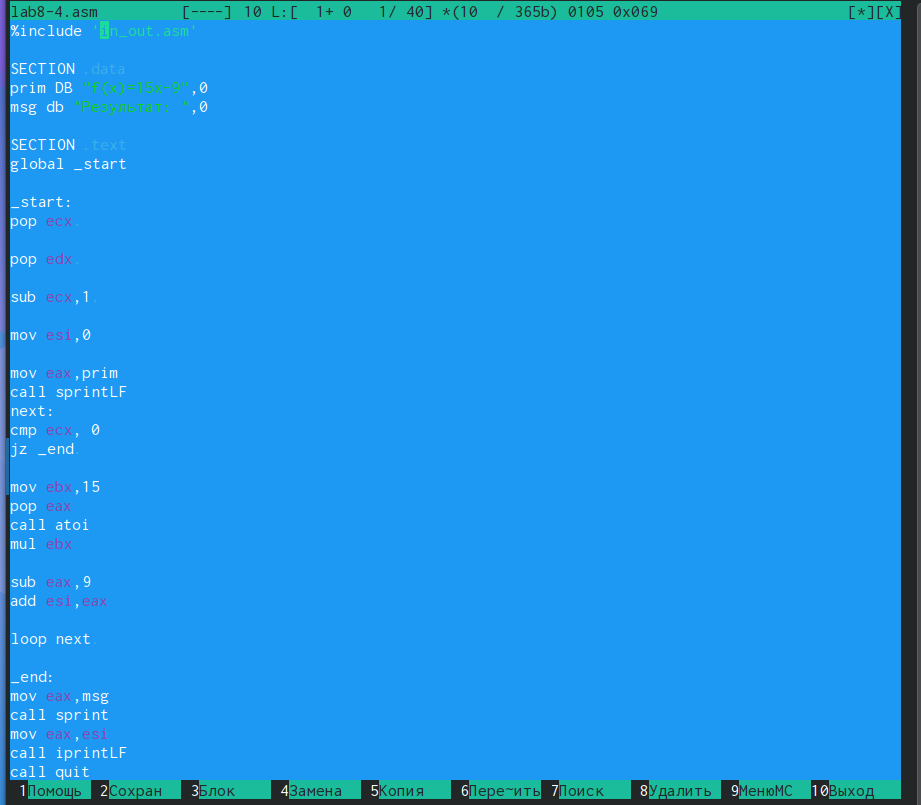


Рис. 24: Написание программы в lab8-4.asm

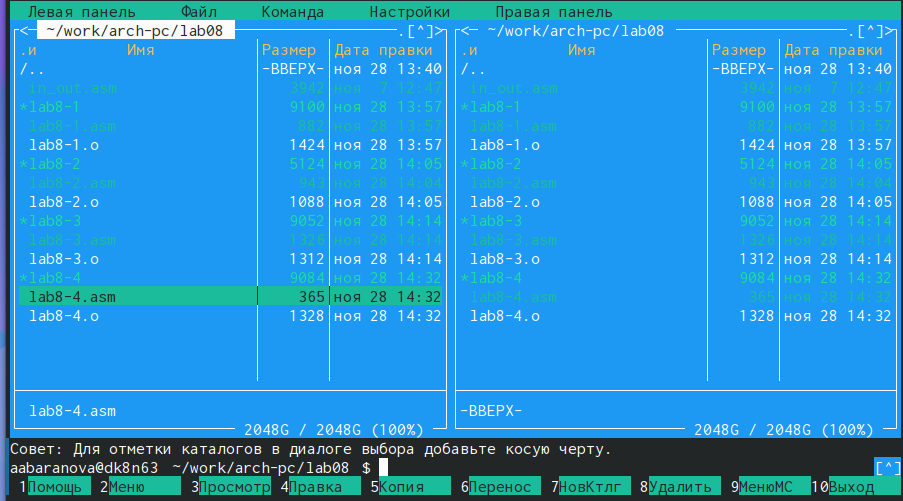


Рис. 25: Создание исполняемого файла и его запуск с указанными аргументами

# 5 Выводы

В ходе выполнения данной лабораторной работы были приобретены навыки написания программ с использованием циклов и обработкой аргументов командной строки.