Отчёта по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки.

сибомана Ламек НКАбд-03-24

Содержание

# 1 Цель работы

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

# 2 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. На рис. 8.1 показана схема организации стека в процессоре. Стек имеет вершину, адрес последнего добавленного элемента, который хранится в ре- гистре esp (указатель стека). Противоположный конец стека называется дном. Значение, помещённое в стек последним, извлекается первым. При помещении значения в стек указа- тель стека уменьшается, а при извлечении — увеличивается. Для стека существует две основные операции: • добавление элемента в вершину стека (push); • извлечение элемента из вершины стека (pop). |

# 3 Выполнение лабораторной работы

1.Сначала я создал каталог для программам лабораторной работы № 8, затем перешёл в него и создал файл lab8-1.asm(рис. fig. **¿fig:001?**) ![Создание каталога и файла lab8-1] (image/1.png){#fig:001 width=70%}

Открывал файл в Midnight Commander и заполняем его в соответствии с листингом 8.1 (рис. fig. **¿fig:002?**).

![Текст программы lab8-1.asm] (image/2.png){#fig:002 width=70%}

Создал исполняемый файл и проверьте его работу.

![Создание и запуск lab8-1.asm]image/3.png){#fig:003 width=70%}

Изменил текст программы добавив изменение значение регистра ecx в циклеСнова открывал файл для редактирования и изменяем его, добавив изменение значения регистра в цикле (рис. fig. 1)

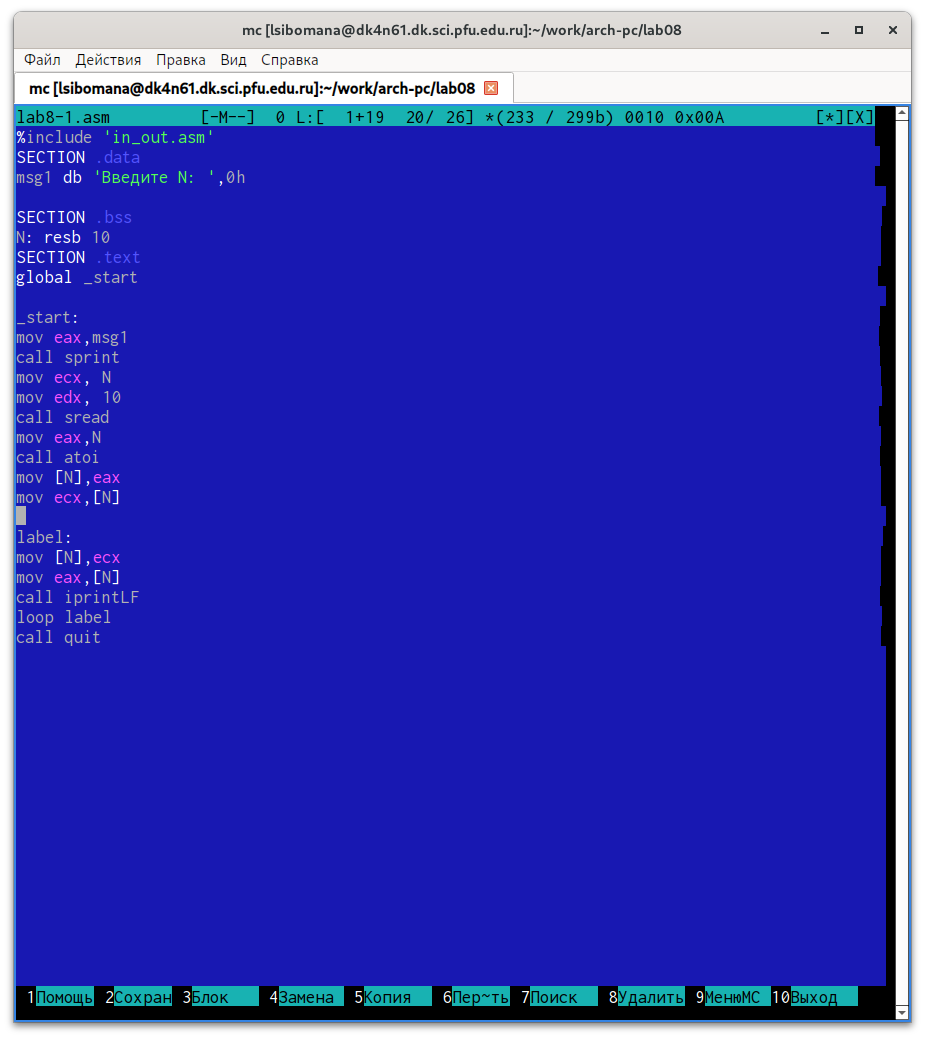


Рис. 1: Текст программы lab8-1.asm

Создаем исполняемый файл и запускаем его (рис. fig. **¿fig:005?**).

![Создание и запуск lab8-1.asm]image/5.png){#fig:005 width=70%}

Регистр ecx принимает значения 9,7,5,3,1(на вход подается число 10, в цикле label данный регистр уменьшается на 2 командой sub и loop).

Число проходов цикла не соответсвует числу N, так как уменьшается на 2.

Снова открываем файл для редактирования и изменяем его, чтобы все корректно работало (рис.fig. 2).

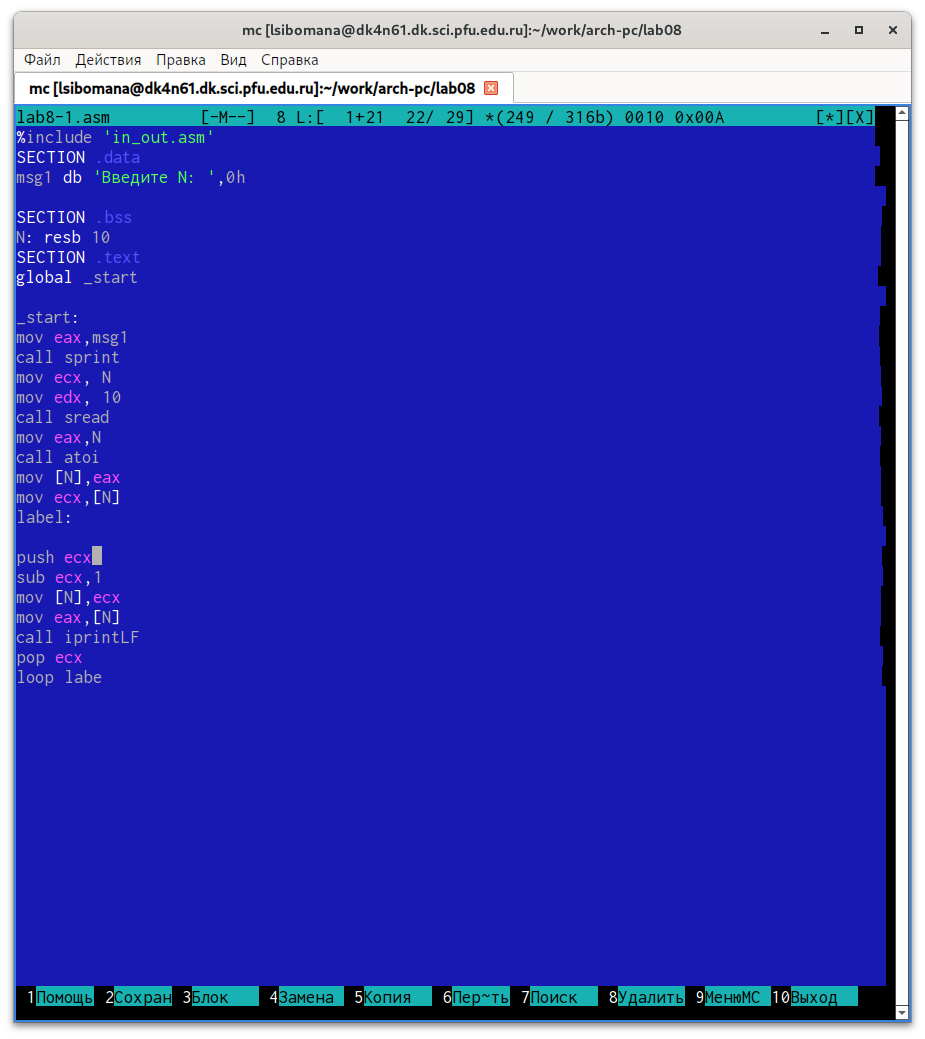


Рис. 2: Текст программы lab8-1.asm

Создаем исполняемый файл и запускаем его (рис. fig. 3).

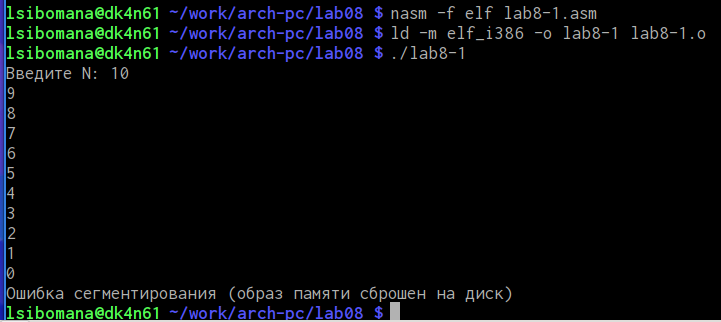


Рис. 3: Создание и запуск lab8-1.asm

В данном случае число проходов цикла равна числу N.

ОБРАБОТКА АРГУМЕНТОВ КОМАНДНОЙ СТРОКИ

Создаем новый файл (рис.fig. 4).

Рис. 4: Создание lab8-2.asm

Рис. 4: Создание lab8-2.asm

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 8.2 (рис.fig. 5).

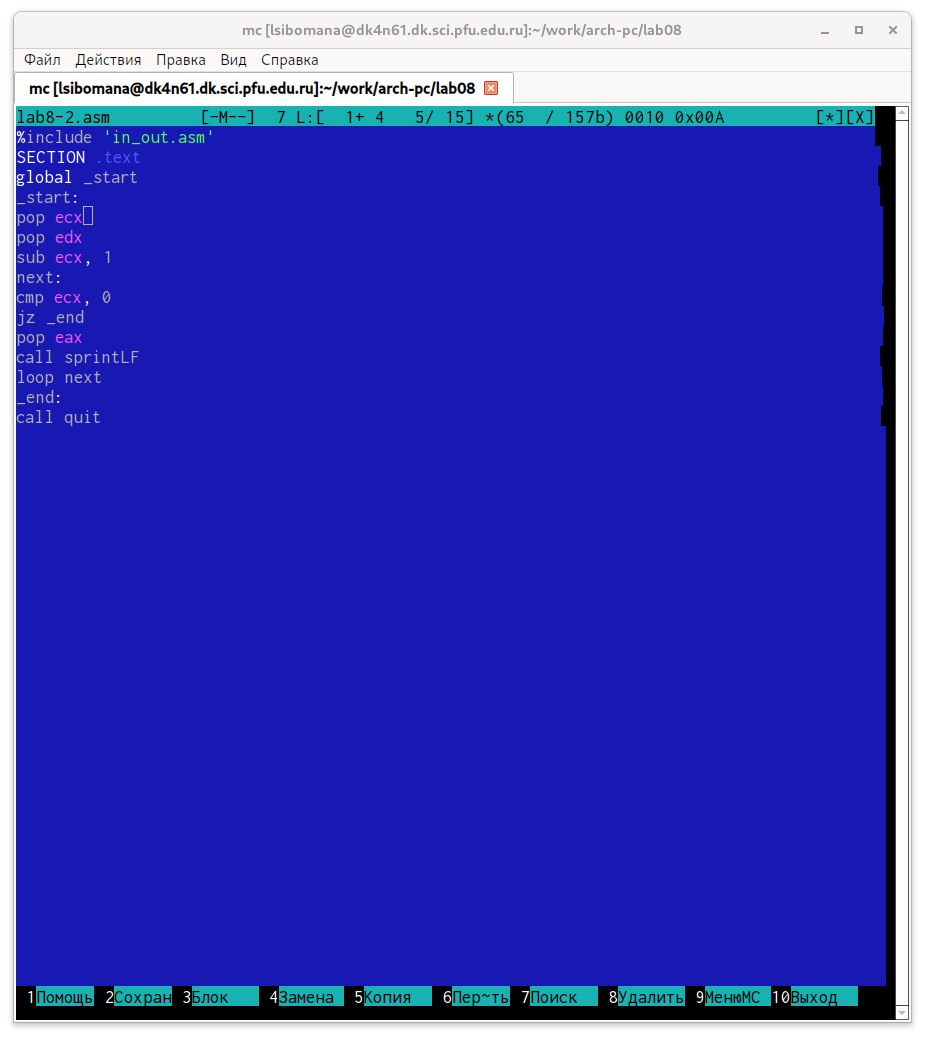


Рис. 5: Текст программы lab8-2.asm

Создаем исполняемый файл и проверяем его работу, указав аргументы (рис.fig. **¿fig:010?**).

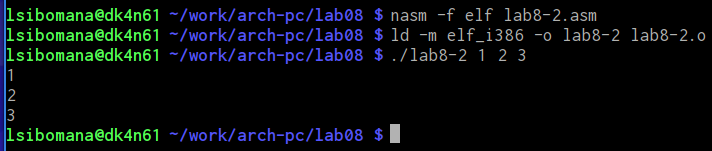


Рис. 6: Создание и запуск lab8-2.asm

Програмой было обработано 3 аргумента.

Создаем новый файл lab8-3.asm (рис. fig. **¿fig:011?**).

Рис. 7: Создание lab8-3.asm

Рис. 7: Создание lab8-3.asm

Открываем файл и заполняем его в соответствии с листингом 8.3 (рис. fig. **¿fig:012?**)

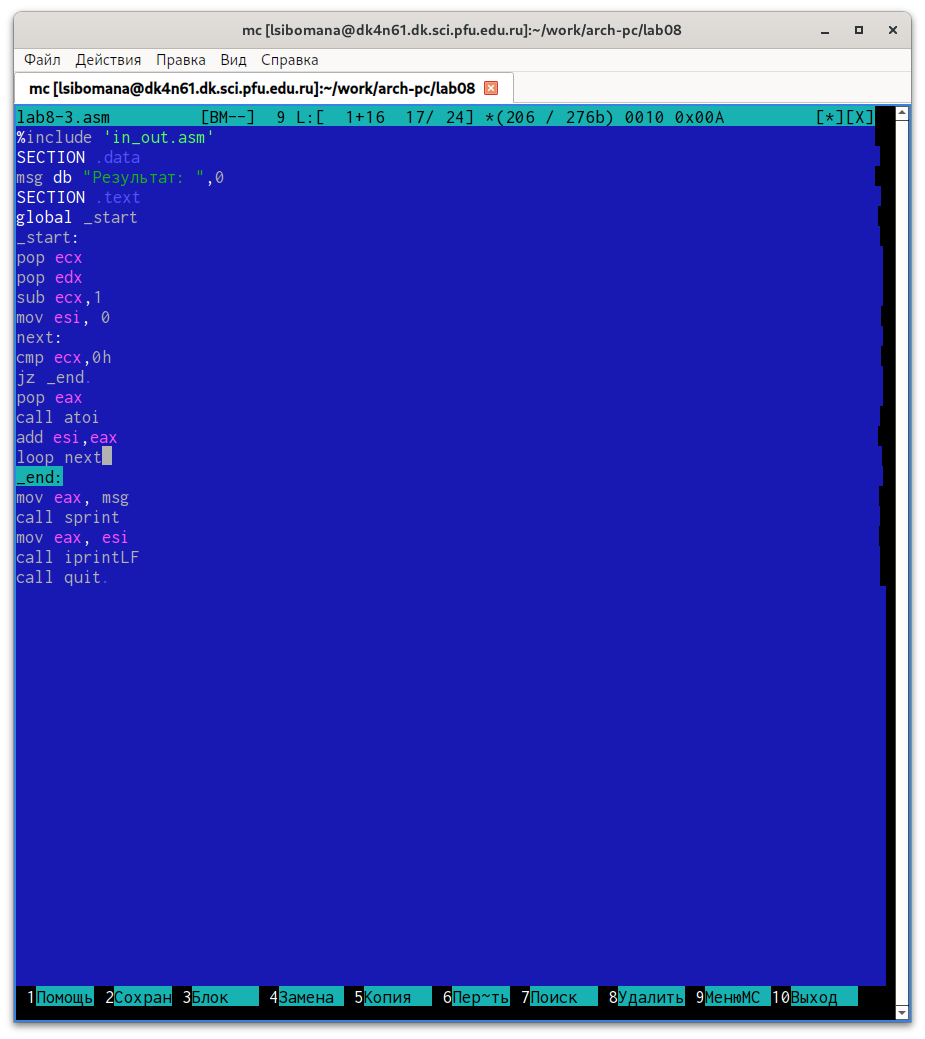


Рис. 8: Текст программы lab8-3.asm

Создаём исполняемый файл и запускаем его, указав аргументы (рис. fig. **¿fig:013?**).

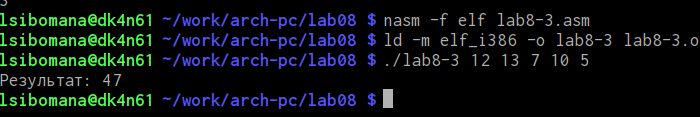


Рис. 9: Создание и запуск lab8-3.asm

Снова открываем файл для редактирования и изменяем его, чтобы вычислялось произведение вводимых значений (рис. fig. **¿fig:014?**)

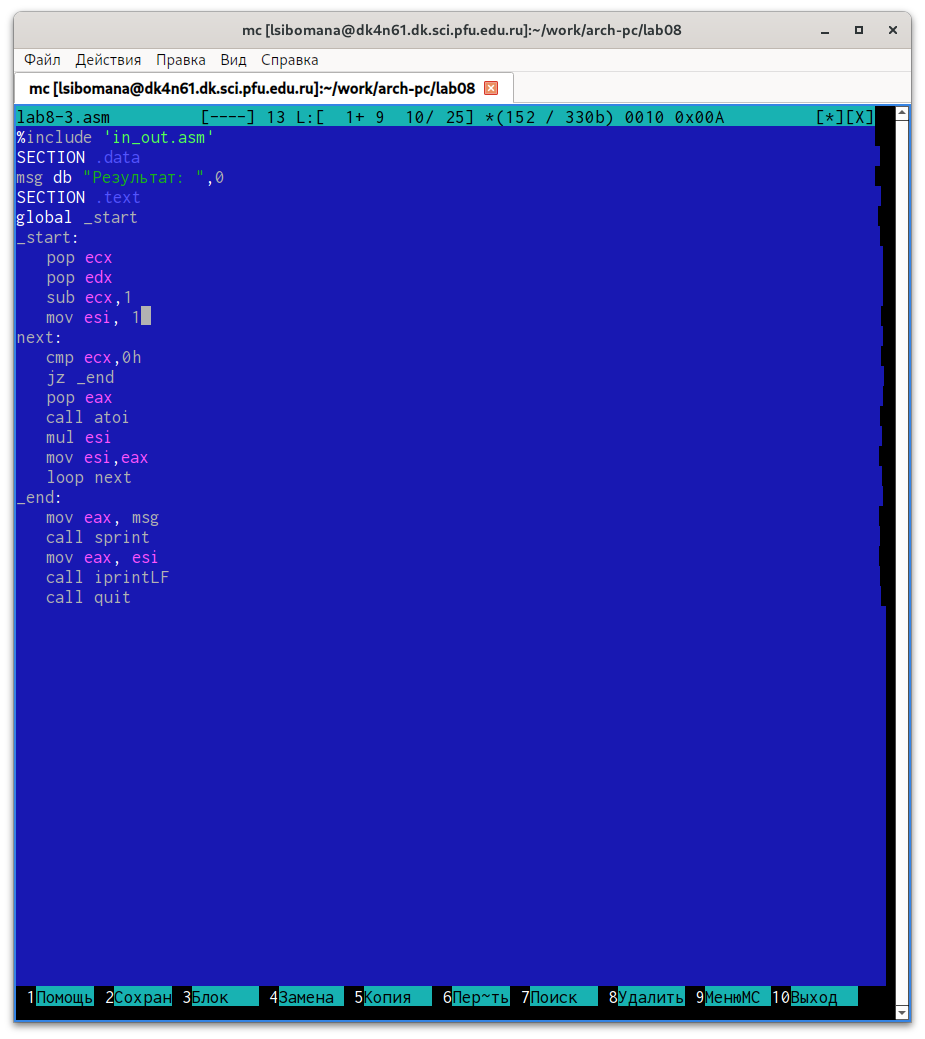


Рис. 10: Текст программы lab8-3.asm

Создаём исполняемый файл и запускаем его, указав аргументы (рис. fig. **¿fig:015?**).

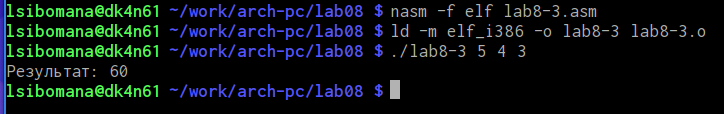


Рис. 11: Создание и запуск lab8-3.asm

ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

ВАРИАНТ-5

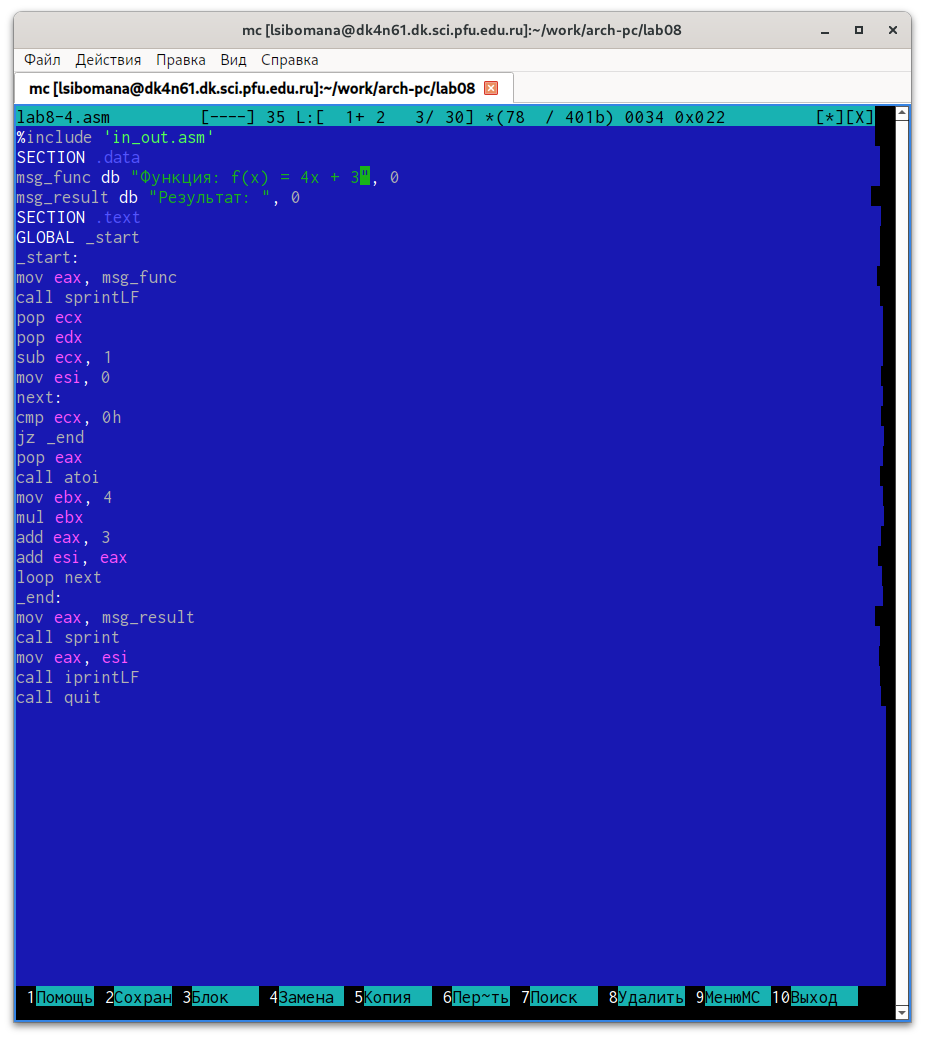
Напишите программу, которая находит сумму значений функции 𝑓(𝑥) для 𝑥 = 𝑥1, 𝑥2, ..., 𝑥𝑛, т.е. программа должна выводить значение 𝑓(𝑥1) + 𝑓(𝑥2) + ... + 𝑓(𝑥𝑛). Значения 𝑥𝑖 передаются как аргументы. Вид функции 𝑓(𝑥) выбрать из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах 𝑥 = 𝑥1, 𝑥2, ..., 𝑥𝑛.

Создаем новый файл (рис.fig. **¿fig:016?**).

Рис. 12: Создание lab8-4.asm

Рис. 12: Создание lab8-4.asm

Открываем его и пишем программу, которая выведет сумму значений, получившихся после решения выражения 4𝑥 + 3 (рис. fig. **¿fig:017?**).

{#fig:0017width=70%}

Транслируем файл и смотрим на работу программы (рис. fig. **¿fig:018?**).

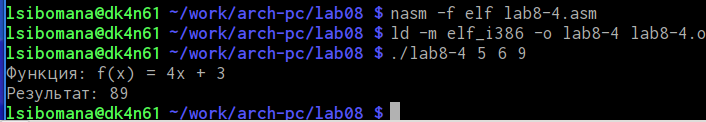


Рис. 13: Создание и запуск lab8-4.asm

Транслируем файл и смотрим на работу программы (рис. fig. **¿fig:019?**).

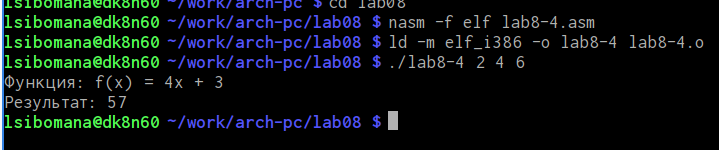


Рис. 14: Создание и запуск lab8-4.asm

# 4 Выводы

Мы научились решать программы с использованием циклов и обработкой аргументов командной строки.