Отчёт по лабораторной работе №8

НКАбд-03-25

Кулаженкова Яна Сергеевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Задание

Освоить принципы организации циклов и обработки аргументов командной строки в языке ассемблера NASM.

3 Теоретическое введение

Работа с циклами и обработка аргументов командной строки являются важными аспектами программирования на низком уровне, особенно в языке ассемблера. В данной лабораторной работе основное внимание уделено двум ключевым вопросам:

1. Организация циклов в ассемблере NASM.
2. Обработка аргументов командной строки.

Часто возникает потребность передавать приложениям некоторые данные прямо при запуске программы. Такие данные называются аргументами командной строки. Процессор сохраняет их в специальной области памяти — стеке. Чтобы обработать эти аргументы, необходимо уметь обращаться к данным в стеке и выделять необходимые элементы. Кроме того, важным элементом работы с циклами и аргументами командной строки является правильное использование стека, так как большинство команд, связанных с передачей аргументов и организацией циклов, активно используют этот механизм.

4 Выполнение лабораторной работы

4.1 Реализация циклов в NASM

Перейдём в каталог для программ лабораторной работы (рис. 4.1).

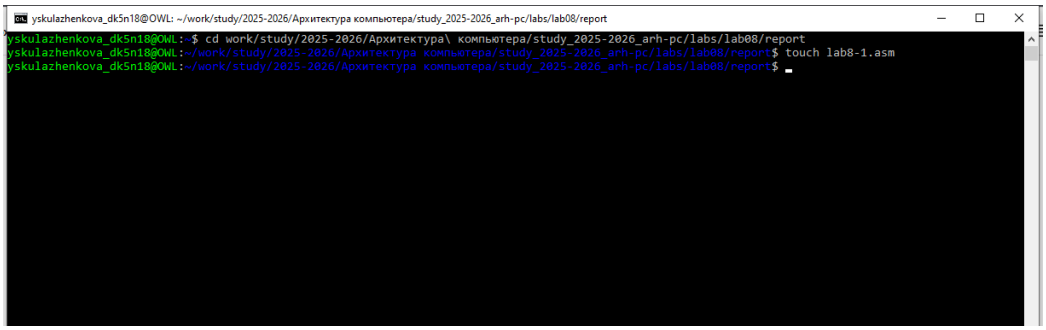

Рисунок 4.1: Переход в каталог лабораторной работы №8

Изучим реализацию циклов в NASM с использованием инструкции loop. В качестве примера рассмотрим программу, которая выводит значение регистра ecx (рис. 4.2).

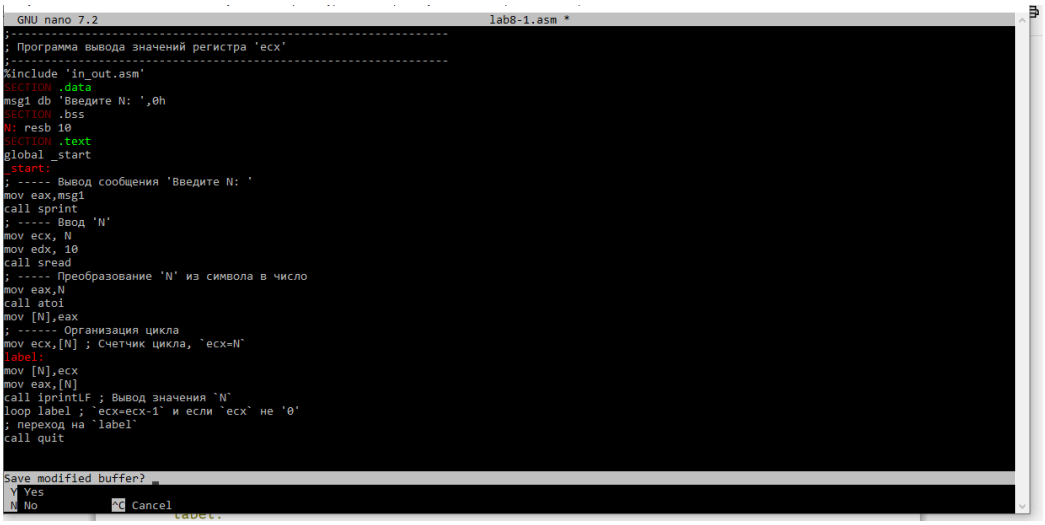

Рисунок 4.2: Содержимое файла lab8-1

Создадим исполняемый файл и проверим его работу (рис. 4.3).

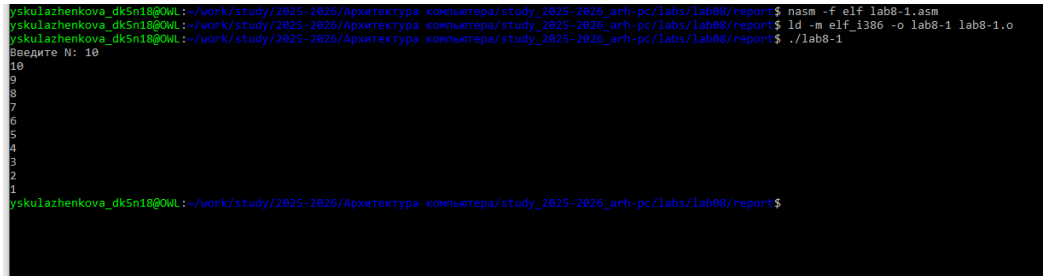

Рисунок 4.3: Продемонстрируем работу файла lab8-1

Заметим, что использование регистра ecx в теле цилка loop приводит к некорректной работе программы. Изменим текст программы (рис. 4.4).

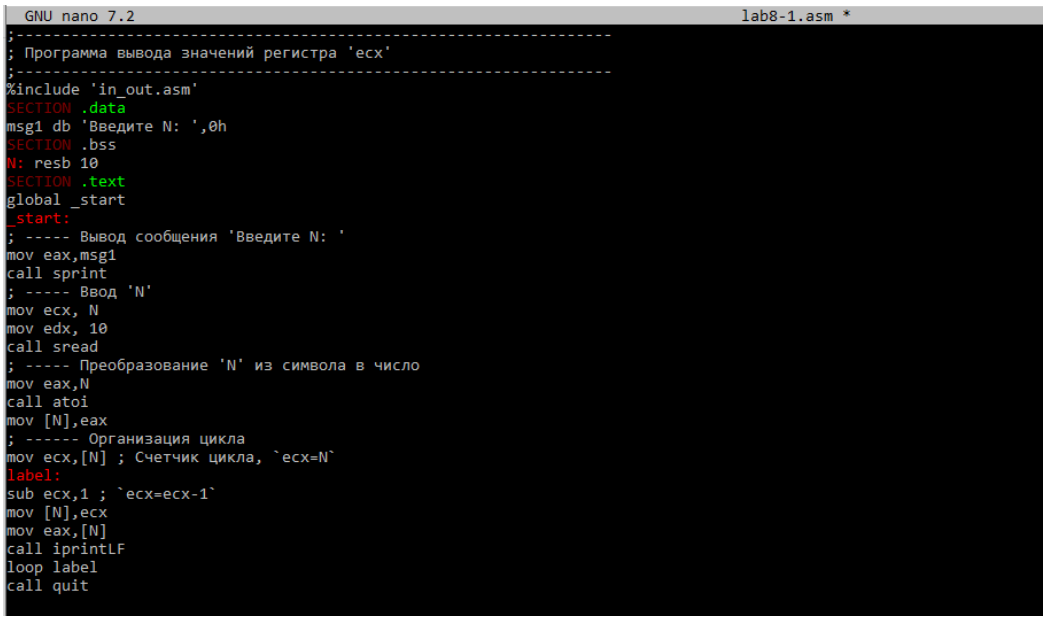

Рисунок 4.4: Изменим код

Продемонстрируем работу файла (рис. 4.5).

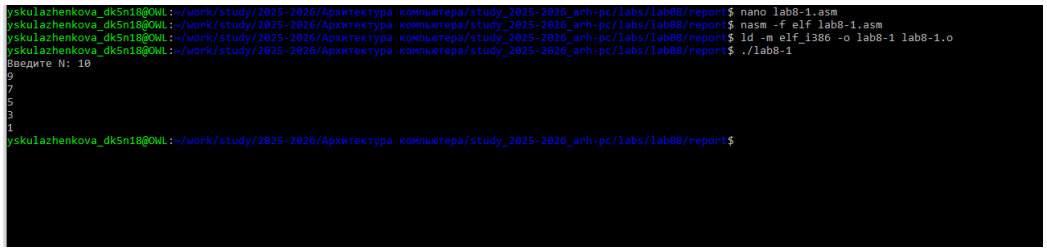

Рисунок 4.5: Работа измененного кода

Теперь напишем программу для использования стека. Для этого добавим в текст программы команды push и pop (рис. 4.6).

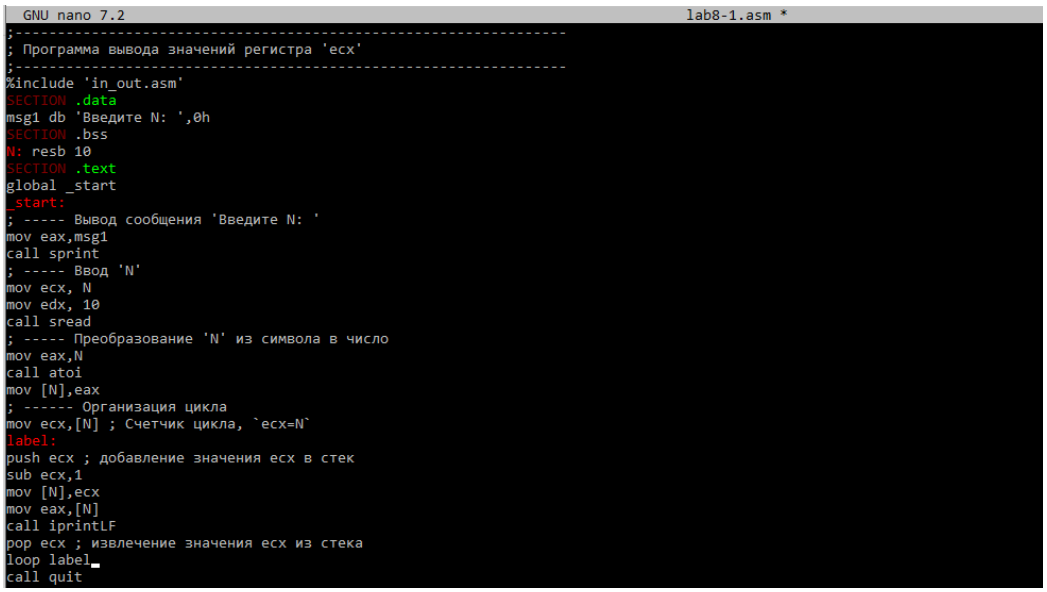

Рисунок 4.6: Добавление push и pop

Проверим работу кода (рис. 4.7).

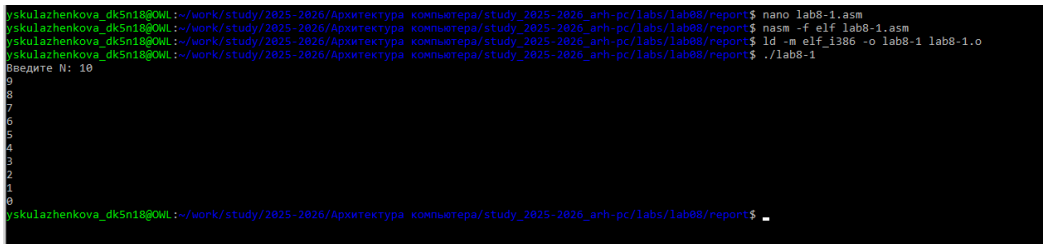

Рисунок 4.7: Выполнение нового кода

4.2 Обработка аргументов командной строки

Рассмотрим общие принципы взаимодействия со стеком. Для этого реализуем программу, которая выводит на экран аргументы командной строки. Создадим файл lab8-2.asm и введем в него текст программы из листинга 8.2 (рис. 4.8).

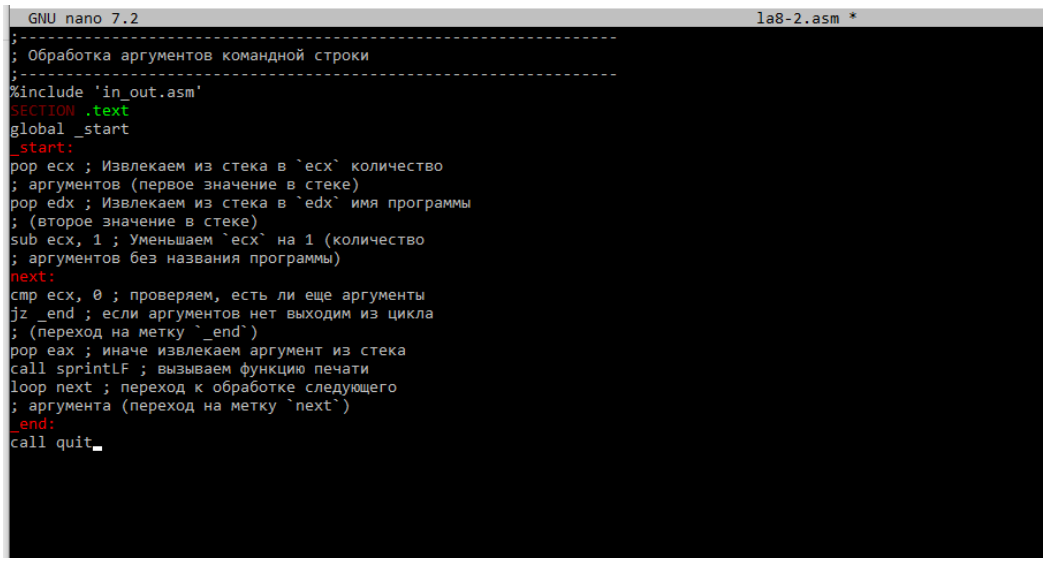

Рисунок 4.8: Содержимое файла lab8-2

Запустим программу (рис. 4.9).

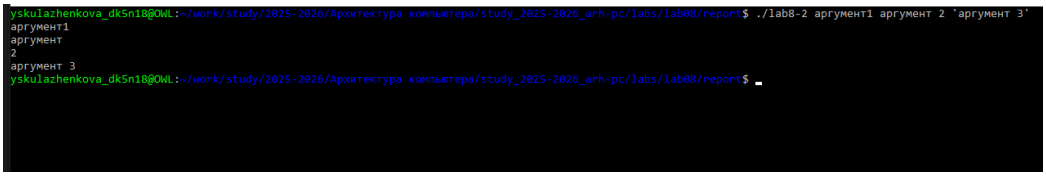

Рисунок 4.9: Работа кода файла lab8-2

В качестве ещё одного примера напишем код программы, которая выводит сумму аргументов. Для этого в файле lab8-3.asm запишем следующее (рис. 4.10).

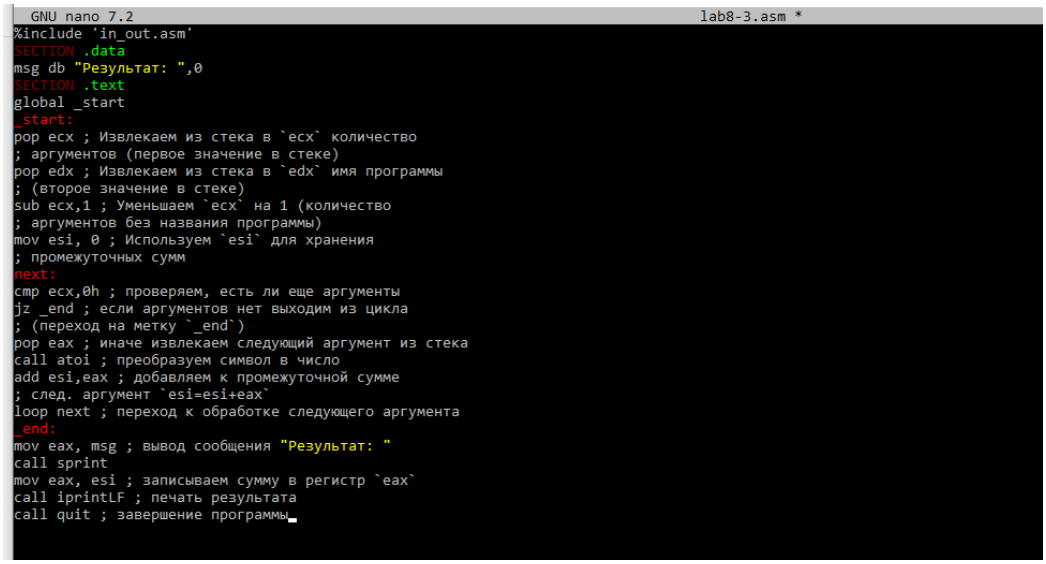

Рисунок 4.10: Программа в файле lab8-3

Создадим исполняемый файл и запустим его, указав аргументы (рис. 4.11).

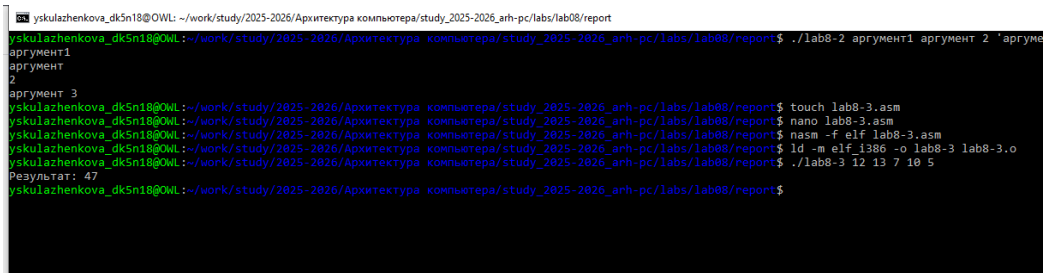

Рисунок 4.11: Результат работы программы lab8-3

5 Задание для самостоятельной работы

По условию задания нам необходимо написать программу, которая находит сумму значений функции (рис. 5.1).

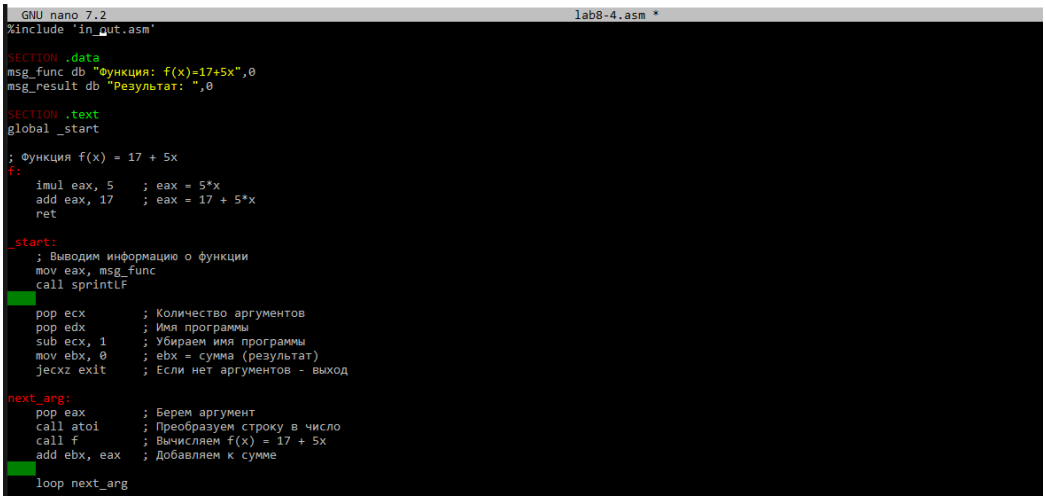

Рисунок 5.1: Самостоятельная работа

Проверим правильность работы программы (рис. 5.2).

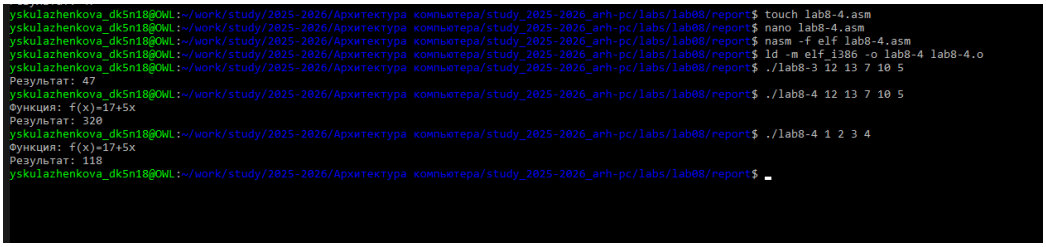

Рисунок 5.2: Выполнение программы для поиска суммы

6 Выводы

В ходе выполнения лабораторной работы были получены и закреплены важные навыки программирования на языке ассемблера NASM. Был освоен механизм организации циклов, также были изучены ключевые инструкции, такие как `loop`, `jmp`, `push` и `pop`.

Полученные знания позволят проектировать и разрабатывать эффективные и надежные программы на ассемблере, обеспечивающие быструю и гибкую обработку больших объемов данных.