Отчёт по лабораторной работе №8

Дисциплина: архитектура компьютеров и операционные системы

Сахно Никита НКАбд-05-23

Содержание

[Цель работы 1](#_Toc152427525)

[Задание 1](#_Toc152427526)

[Выполнение лабораторной работы 1](#_Toc152427527)

[Реализация циклов в NASM 1](#_Toc152427528)

[Обработка аргументов командной строки 2](#_Toc152427529)

[Задание для самостоятельной работы 3](#_Toc152427530)

[Выводы 3](#_Toc152427531)

[Список литературы 3](#_Toc152427532)

# Цель работы

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

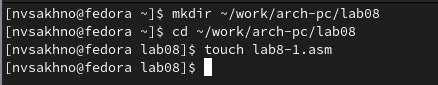
# Задание

1. Реализация циклов в NASM.
2. Обработка аргументов командной строки.
3. Задание для самостоятельной работы.

# Выполнение лабораторной работы

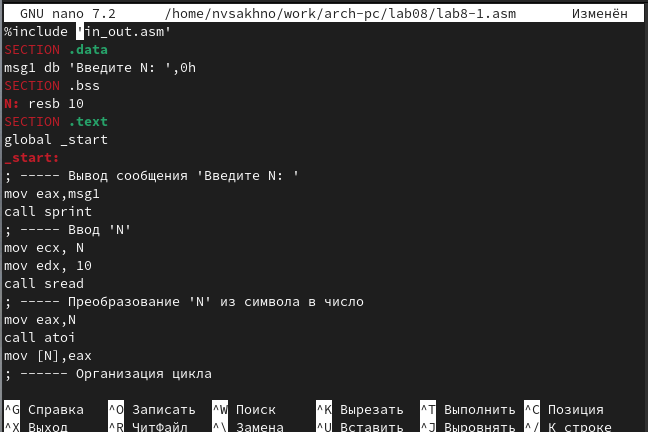
## Реализация циклов в NASM

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



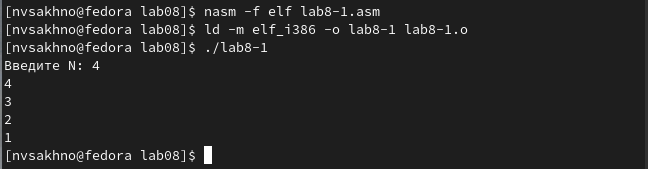
*Создание файлов для лабораторной работы*

Ввожу в файл lab8-1.asm текст программы из листинга 8.1. (рис. 2).



*Ввод текста из листинга 8.1*

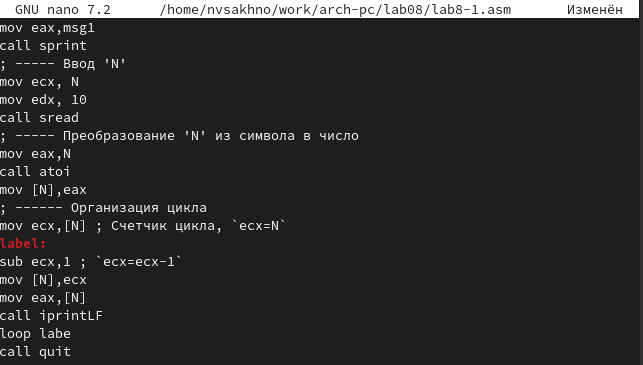
Создаю исполняемый файл и проверяю его работу. (рис. 3).



*Запуск исполняемого файла*

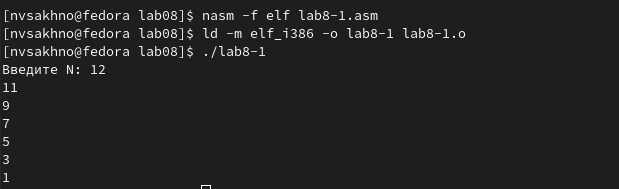
Данная программа выводит числа от N (4) до 1 включительно.

Изменяю текст программы, добавив изменение значения регистра ecx в цикле. (рис. 4).



*Изменение текста программы*

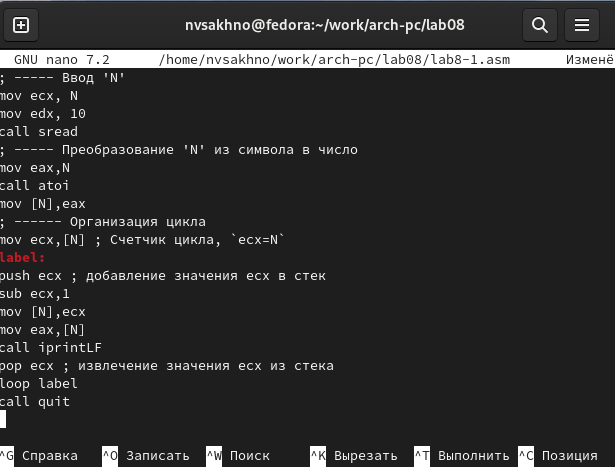
Создаю исполняемый файл и проверяю его работу. (рис. 5).



*Запуск обновленной программы*

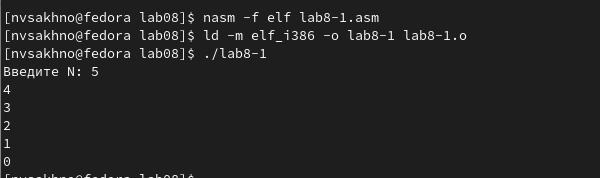
В данном случае число проходов цикла не соответствует введенному с клавиатуры значению.

Вношу изменения в текст программы, добавив команды push и pop для сохранения значения счетчика цикла loop. (рис. 6).



*Изменение текста программы*

Создаю исполняемый файл и проверяю его работу. (рис. 7).

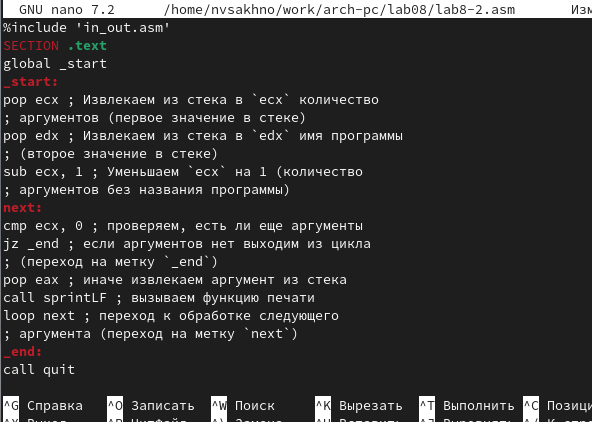


*Запуск исполняемого файла*

В данном случае число проходов цикла соответствует введенному с клавиатуры значению и выводит числа от N-1 до 0 включительно.

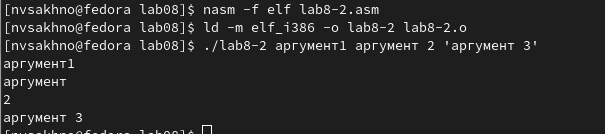
## Обработка аргументов командной строки

Создаю файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и ввожу в него текст программы из листинга 8.2. (рис. 8).



*Ввод текста программы из листинга 8.2*

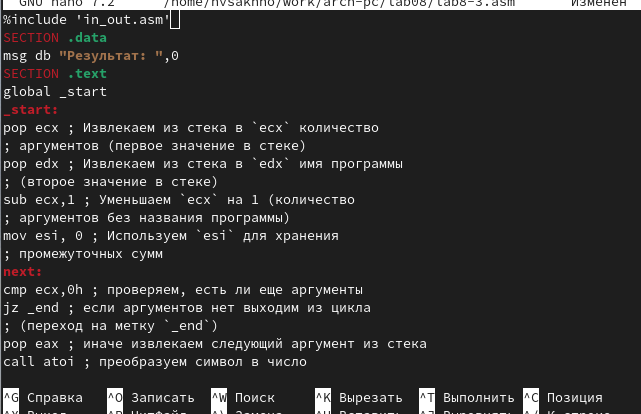
Создаю исполняемый файл и запускаю его, указав нужные аргументы. (рис. 9).



*Запуск исполняемого файла*

Программа вывела 4 аргумента, так как аргумент 2 не взят в кавычки, в отличии от аргумента 3, поэтому из-за пробела программа считывает “2” как отдельный аргумент.

Рассмотрим пример программы, которая выводит сумму чисел, которые передаются в программу как аргументы. Создаю файл lab8-3.asm в каталоге ~/work/archpc/lab08 и ввожу в него текст программы из листинга 8.3. (рис. 10).



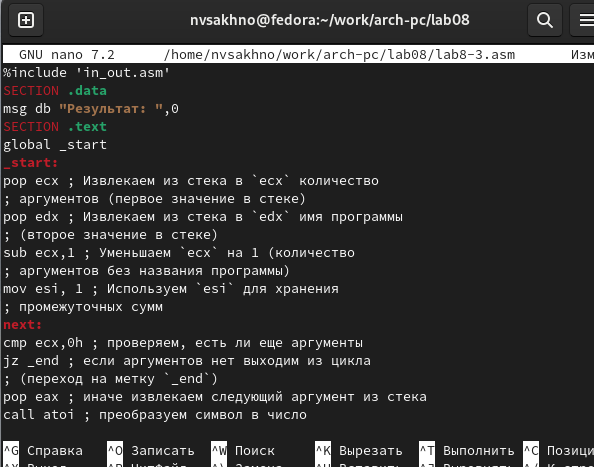
*Ввод текста программы из листинга 8.3*

Создаю исполняемый файл и запускаю его, указав аргументы. (рис. 11).



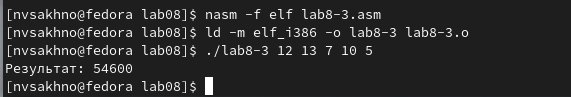
*Запуск исполняемого файла*

Изменяю текст программы из листинга 8.3 для вычисления произведения аргументов командной строки. (рис. 12).



*Изменение текста программы*

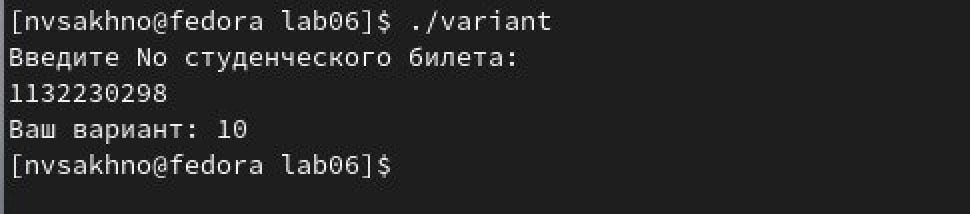
Создаю исполняемый файл и запускаю его, указав аргументы. (рис. 13).



*Запуск исполняемого файла*

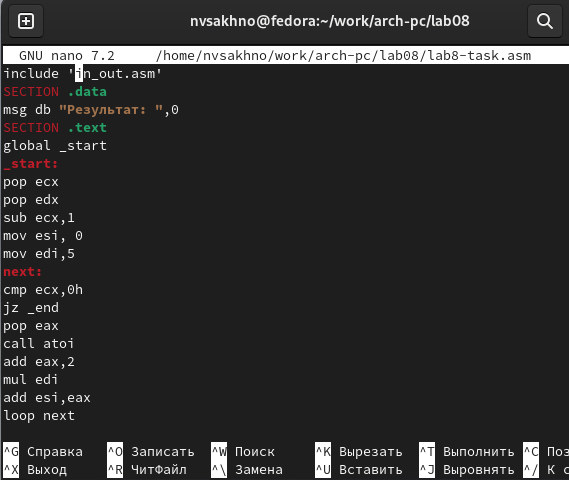
## Задание для самостоятельной работы

У меня вариант 10 (рис. 0)



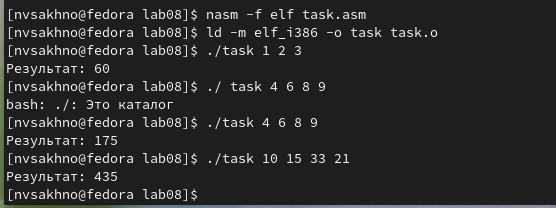
*Вариант из лабораторной работы 6*

Пишу текст программы, которая находит сумму значений функции f(x) = 5\*(2 + x) в соответствии для x = x1, x2, …, xn. Значения xi передаются как аргументы. (рис. 14).



*Текст программы*

Создаю исполняемый файл и проверьте его работу на нескольких наборах x = x1, x2, …, xn. (рис. 15).



*Запуск исполняемого файла и проверка его работы*

Программа работает корректно.

# Выводы

Благодаря данной лабораторной работе я приобрела навыки написания программ использованием циклов и обработкой аргументов командной строки, что поможет мне при выполнении последующих лабораторных работ.

# Список литературы

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.
2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.
3. Midnight Commander Development Center. — 2021. — URL: https://midnight-commander. org/.
4. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.