Лабораторная работа №8

Архитектура компьютера

Башиянц Александра Кареновна

Содержание

# 1 Цель работы

Цель работы — приобретести навыки написания программ с использованием циклов и обработкой аргументов командной строки.

# 2 Задание

В этой лабораторной работе необходимо изучить работу циклов и обратку аргументов командной строки.

Необходимо научиться:

* Изучить команды циклов;
* Приобрести навыки циклов;
* Узнать назначение циклов;
* Изучить команды использования аргументов командной строки.

Выполняя это задание, мы получим практический опыт работы циклов и получения аргументов из командной строки.

# 3 Выполнение лабораторной работы

## 3.1 Реализация циклов в NASM

Создадим директорию для 8 лабораторной работы и создадим файл lab8-1.asm (рис. 1).

Рис. 1: Создание директории

Рис. 1: Создание директории

Скопируем файл in\_out.asm из lab06 с помощью mc (рис. 2).

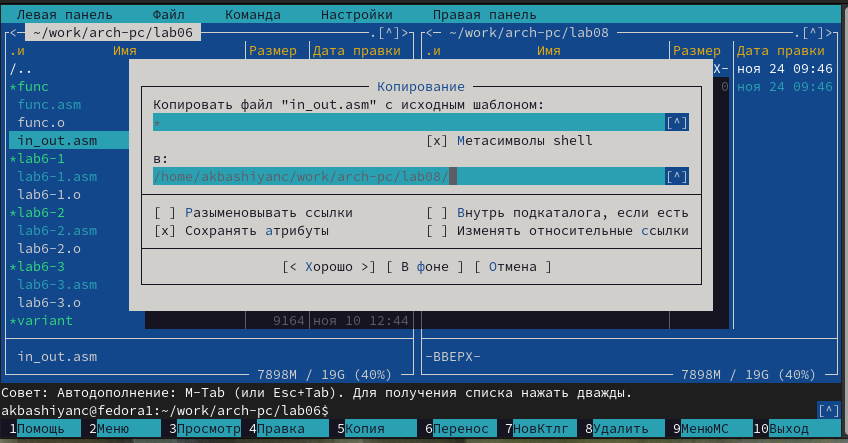


Рис. 2: Копирование in\_out.asm

Введем код в lab7-1.asm и создадим исполняемый файл и запустим его (рис. 3 и 4).

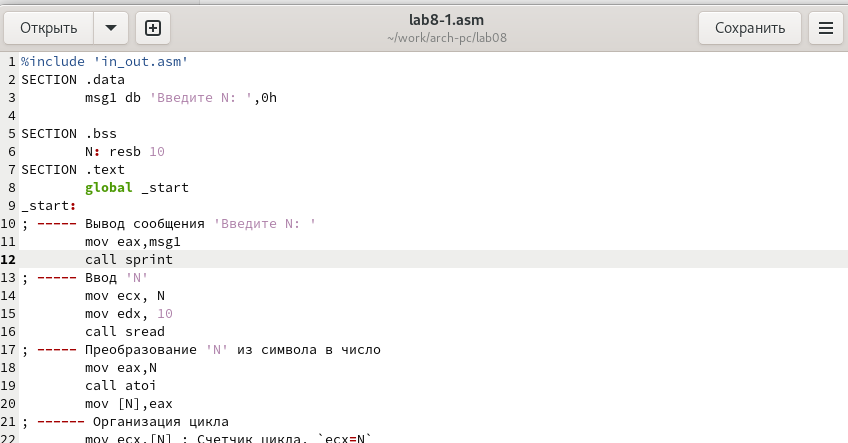


Рис. 3: Ввод кода

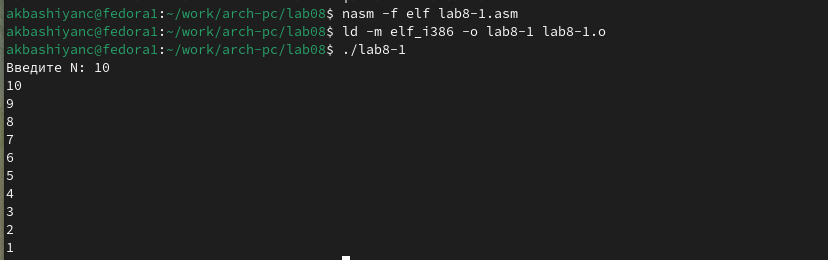


Рис. 4: Запуск файла

Изменим код так, что использование регистра ecx в теле цилка loop может привести к некорректной работе программы (рис. 5).

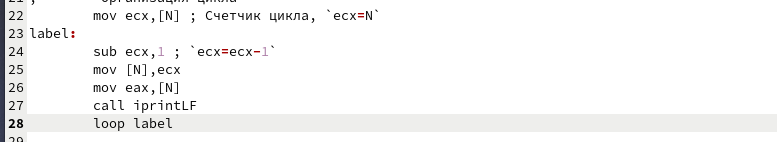


Рис. 5: Изменение файла

Создадим исполняемый файл и запустим его (рис. 6).

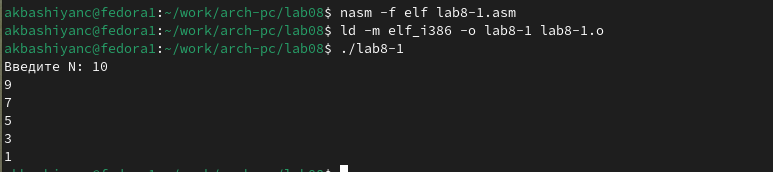


Рис. 6: Запуск файла

Заметим, что теперь выводятся только нечетные числа, так как мы, добавив ecx в тело цилка loop, стали вычитать итератор 2 раза.

Изменим код, добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop (рис. 7).

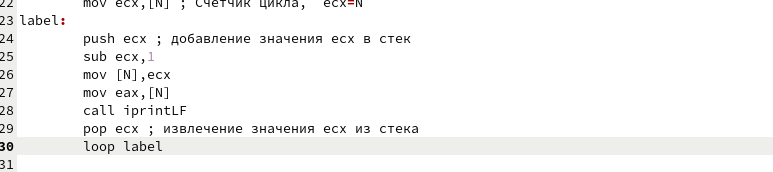


Рис. 7: Изменение файла

Создадим исполняемый файл и запустим его (рис. 8).

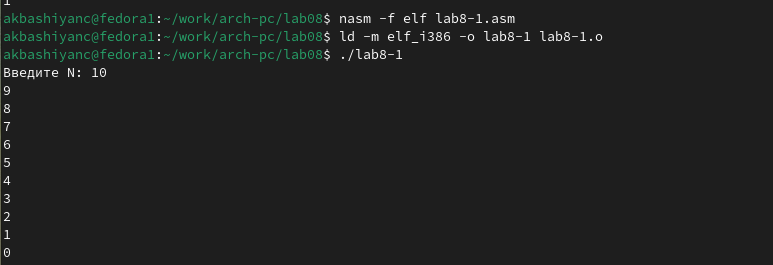


Рис. 8: Запуск файла

Число проходов цикла соответствует знаечнию N, но в отличие от изначального кода, выводится числа от N-1 до 0.

## 3.2 Обработка аргументов командной строки

Создадим файл lab8-2.asm (рис. 9).

Рис. 9: Создание файла

Рис. 9: Создание файла

Введем код в lab8-2.asm (рис. 10).

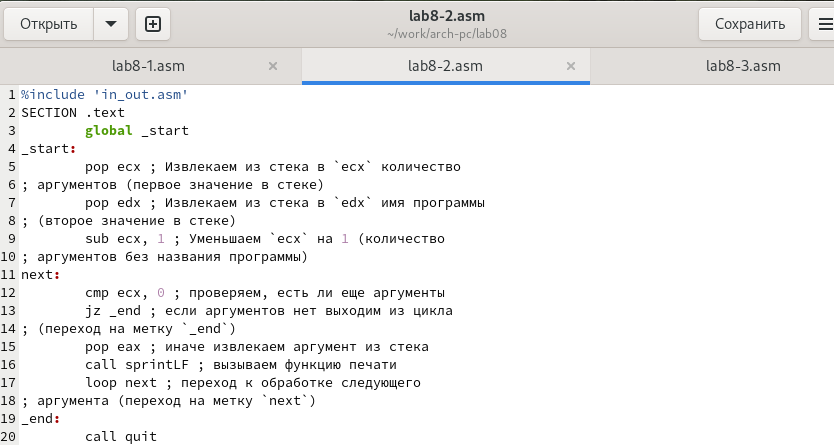


Рис. 10: Ввод кода

Создадим исполняемый файл и запустим его (рис. 11). Было обработано все 3 аргумента.

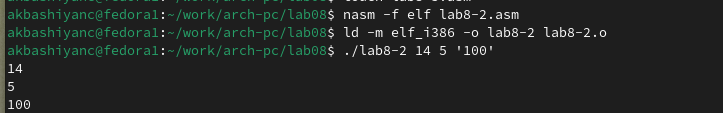


Рис. 11: Запуск файла

Создадим файл lab8-3.asm (рис. 12).

Рис. 12: Создание файла

Рис. 12: Создание файла

Введем код в lab8-2.asm (рис. 13).

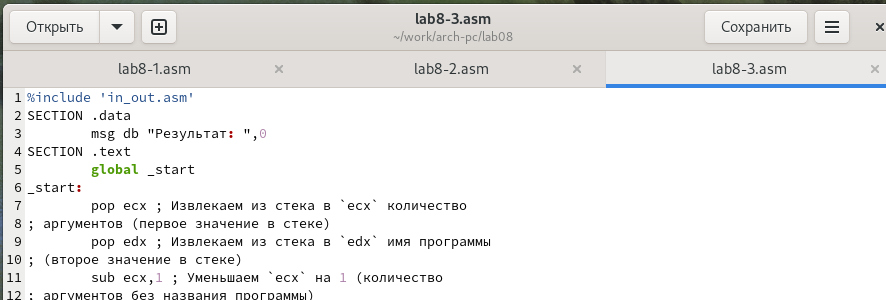


Рис. 13: Ввод кода

Создадим исполняемый файл и запустим его (рис. 14). Было обработано все 3 аргумента.

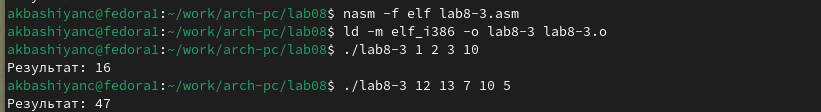


Рис. 14: Запуск файла

Изменим код так, чтобы значения не складывались, а переменожались (рис. 15).



Рис. 15: Изменение файла

Создадим исполняемый файл и запустим его (рис. 16).

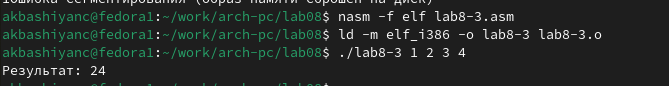


Рис. 16: Запуск файла

## 3.3 Задание для самостоятельной работы

Создадим файл ex1.asm (рис. 17).

Рис. 17: Создание файл

Рис. 17: Создание файл

Введем код для нахождения суммы функции f(x)=5\*x+17 (вариант 18) в ex1.asm (рис. 18).

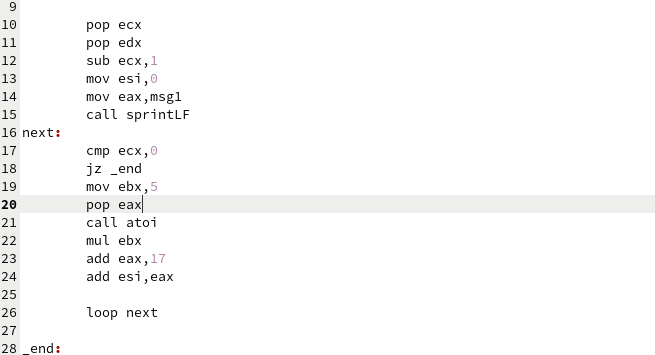


Рис. 18: Ввод кода

Создадим исполняемый файл и запустим его (рис. 19).

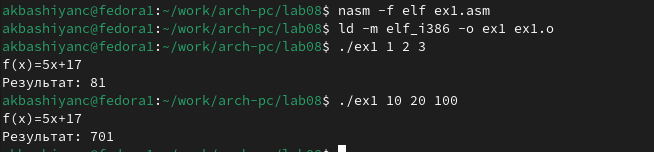


Рис. 19: Запуск файла

# 4 Выводы

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