Отчёт по лабораторной работе №8

Дисциплина: архитектура компьютеров

Терещенкова Маргарита Владимировна

Содержание

# 1 Цель работы

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

# 2 Задание

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

# 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.

Стек имеет вершину, адрес последнего добавленного элемента, который хранится в регистре esp (указатель стека). Противоположный конец стека называется дном. Значение, помещённое в стек последним, извлекается первым. При помещении значения в стек указатель стека уменьшается, а при извлечении — увеличивается.

Для стека существует две основные операции:

• добавление элемента в вершину стека (push);

• извлечение элемента из вершины стека (pop).

# 4 Выполнение лабораторной работы

## 4.1 Реализация циклов в NASM

Создала каталог для программ лабораторной работы № 8, перешла в него и создала файл lab8-1.asm:

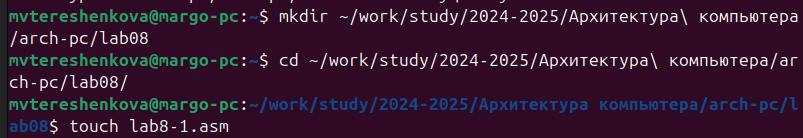


Рис. 1: Создание каталога

Копирую в текущий каталог файл in\_out.asm с помощью утилиты cp, так как он будет использоваться в других программах. И проверяю наличие файла в данной директории с помощью команды ls.

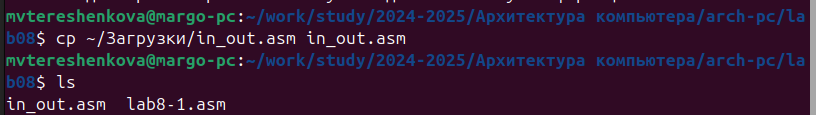


Рис. 2: Копирование файла

Ввожу в файл lab8-1.asm текст программы из листинга 8.1.

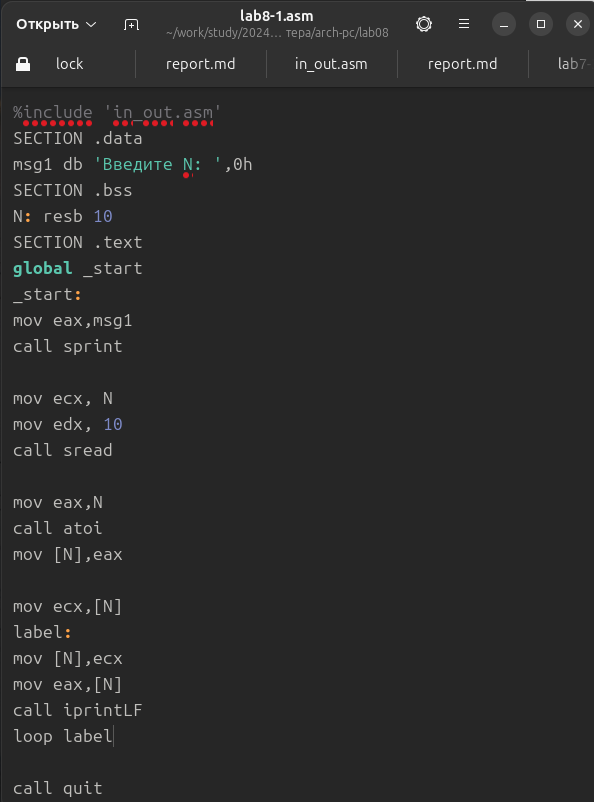


Рис. 3: Редактирование файла

Создаю исполняемый файл и запускаю его.

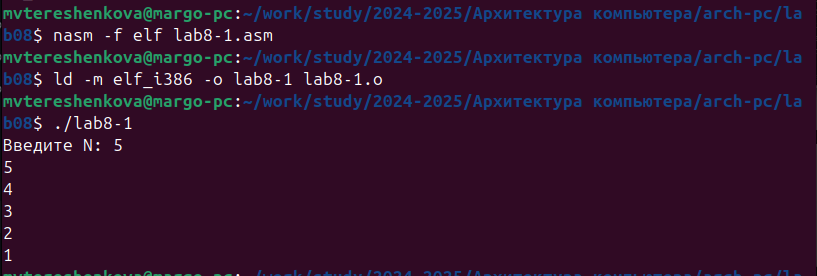


Рис. 4: Запуск файла

Исполняемый файл работает корректно.

Меняю текст программы добавив изменение значение регистра ecx в цикле.

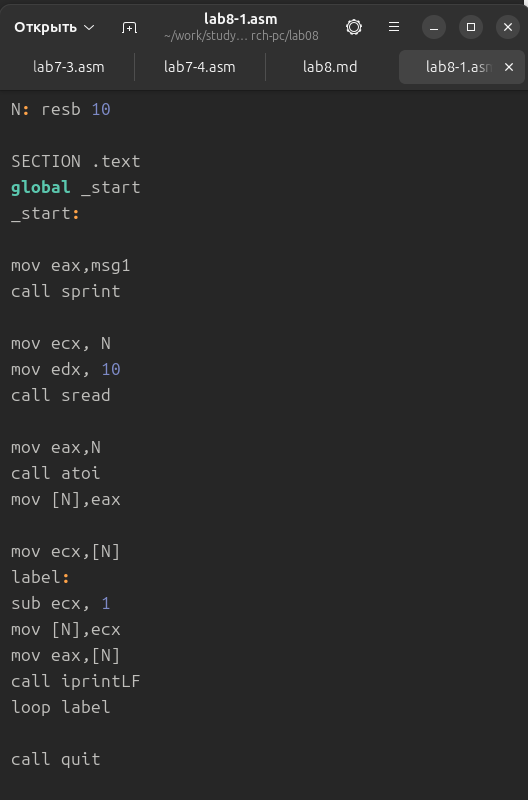


Рис. 5: Редактирование файла

Создаю исполняемый файл и запускаю его.

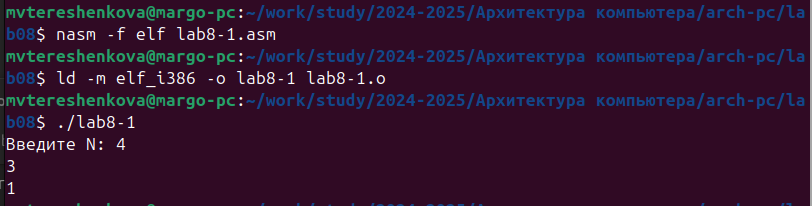


Рис. 6: Запуск файла

Данный пример показывает, что использование регистра ecx в теле цикла loop может привести к некорректной работе программы. Число проходов не соответствует значению N.

Вношу изменения в текст программы, добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop.

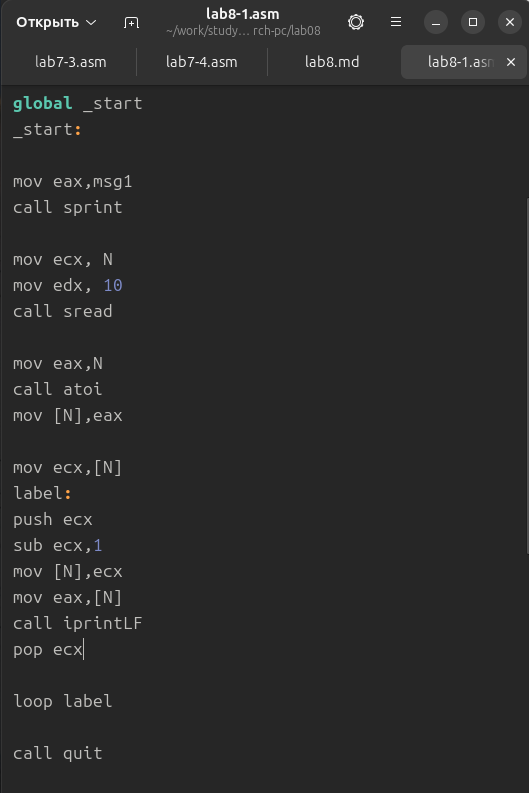


Рис. 7: Редактирование файла

Создаю исполняемый файл и запускаю его.

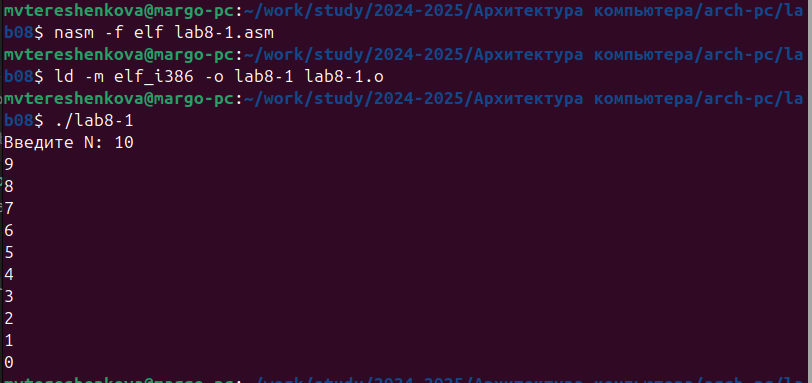


Рис. 8: Запуск файла

В данном случае число проходов цикла соответствует значению N.

## 4.2 Обработка аргументов командной строки

Создаю файл с названием lab8-2.asm.

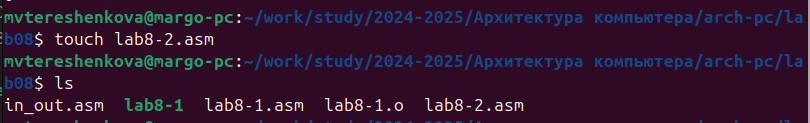


Рис. 9: Создание файла

Ввожу в него текст программы из листинга 8.2

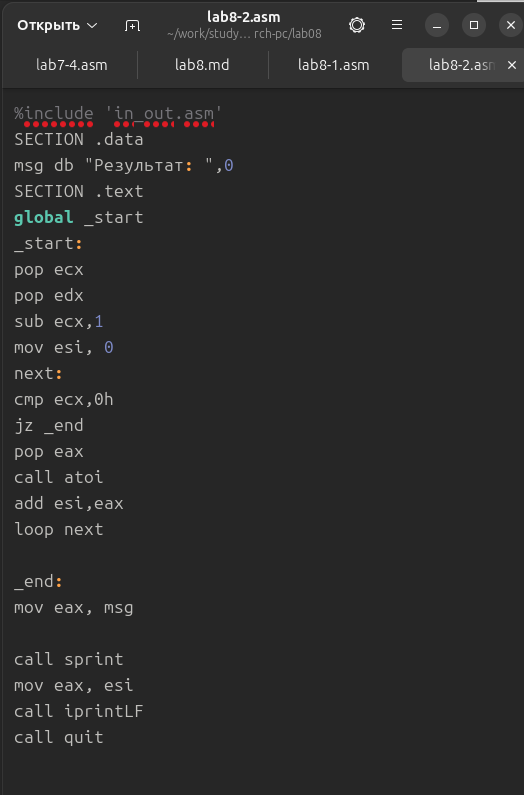


Рис. 10: Редактирование файла

Создаю исполняемый файл и запускаю его, указав аргументы.

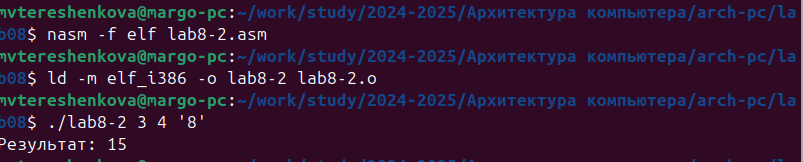


Рис. 11: Запуск исполняемого файла

Программа обработала 3 аргумента.

Создаю файл с названием lab8-3.asm.

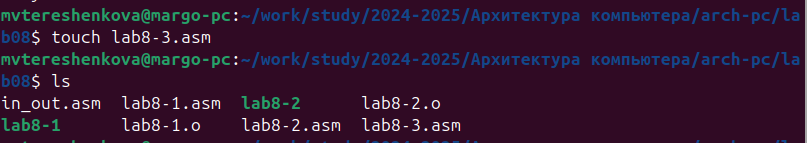


Рис. 12: Создание файла

Ввожу в него текст программы из листинга 8.3

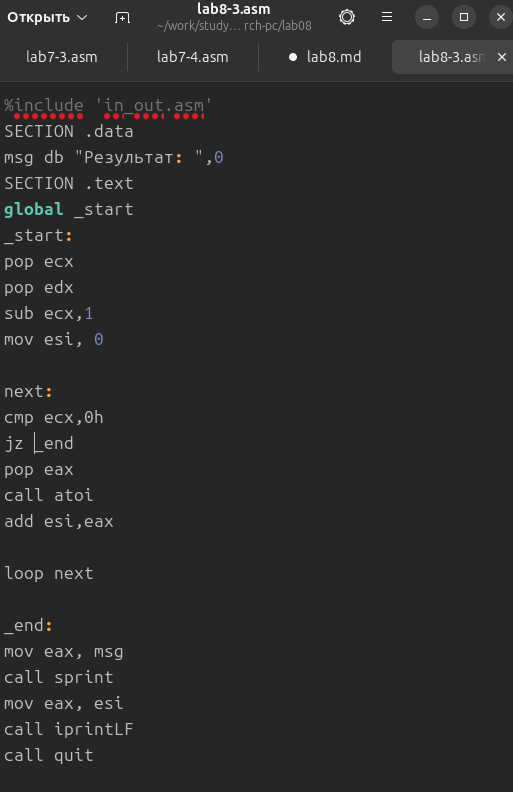


Рис. 13: Редактирование файла

Создаю исполняемый файл и запускаю его, указав аргументы.

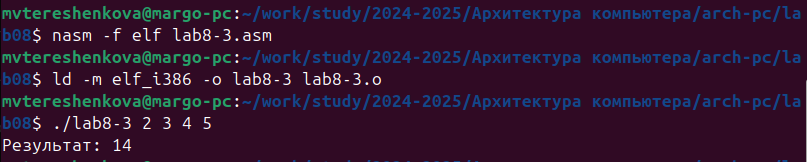


Рис. 14: Запуск исполняемого файла

Редактирую текст программы для вычисления произведения аргументов командной строки.

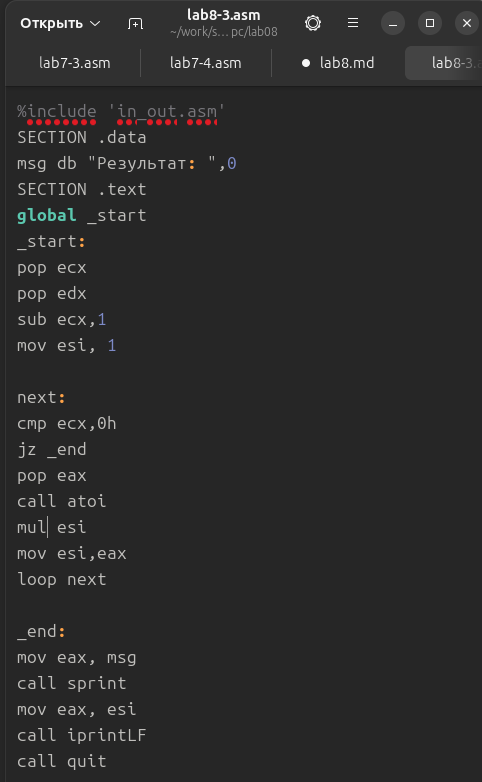


Рис. 15: Редактирование файла

Создаю исполняемый файл и запускаю его, указав аргументы.

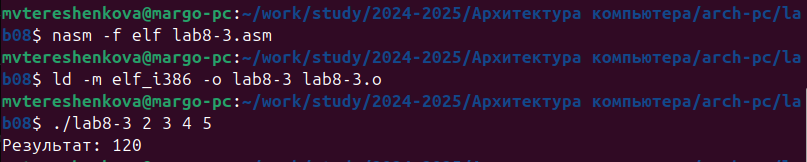


Рис. 16: Запуск исполняемого файла

## 4.3 Самостоятельная работа

Создаю файл lab8-4.asm.

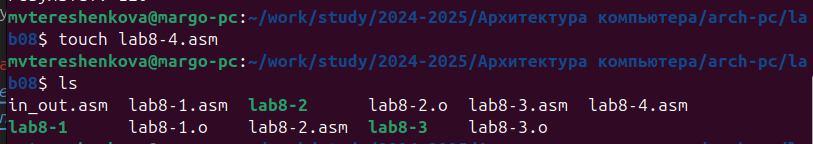


Рис. 17: Создание файла

Начинаю написание программы, которая будет вычислять сумму значений f(x)=2(x-1).(вариант 4)

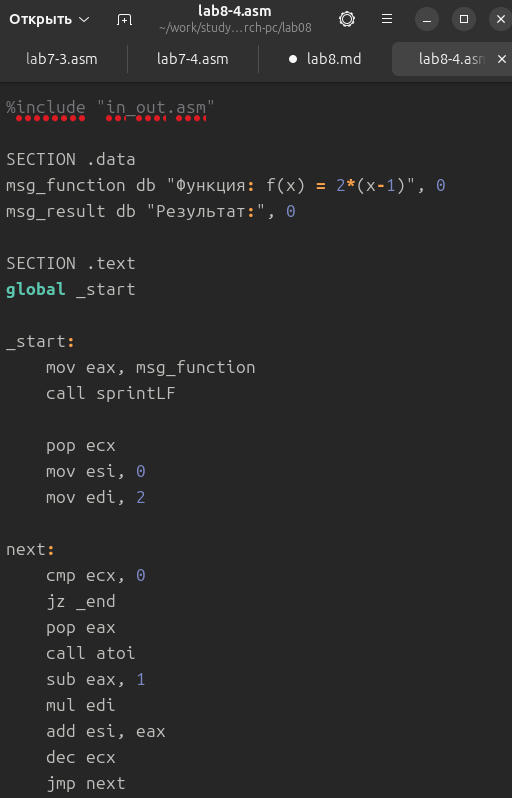


Рис. 18: Редактирование файла

Создаю исполняемый файл и запускаю его.

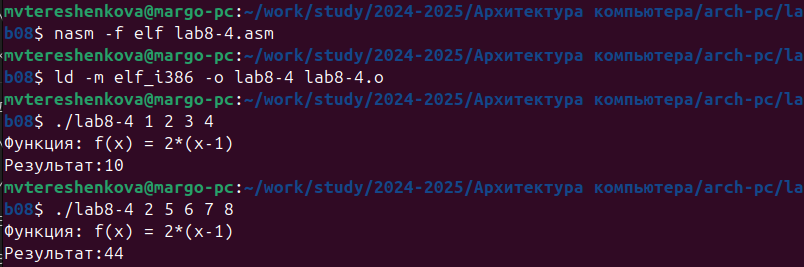


Рис. 19: Запуск исполняемого файла

Произведя несложные математические вычисления, делаю вывод, что программа работает верно.

**Текст программы:**

%include “in\_out.asm”

SECTION .data

msg\_function db “Функция: f(x) = 2\*(x-1)“, 0

msg\_result db “Результат:”, 0

SECTION .text

global \_start

\_start:

mov eax, msg\_function

call sprintLF

pop ecx

mov esi, 0

mov edi, 2

next:

cmp ecx, 0

jz \_end

pop eax

call atoi

sub eax, 1

mul edi

add esi, eax

dec ecx

jmp next

\_end:

mov eax, msg\_result

call sprint

mov eax, esi

call iprintLF

call quit

# 5 Выводы

Благодаря данной лабораторной работе, приобрела навыки написания программ с использованием циклов и обработкой аргументов командной строки.

# Список литературы

1. Архитектура компьютеров