Отчёта по лабораторной работе №8

Архитектура компьютера

Еремина Оксана Андреевна НКАбд-02-23

Содержание

# Цель работы

Целью данной лабораторной работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

# Задание

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

# 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.

Команда push размещает значение в стеке, т.е. помещает значение в ячейку памяти, на которую указывает регистр esp, после этого значение регистра esp увеличивается на 4. Данная команда имеет один операнд — значение, которое необходимо поместить в стек

Для организации циклов существуют специальные инструкции. Для всех инструкций максимальное количество проходов задаётся в регистре ecx. Наиболее простой является инструкция loop.

# Выполнение лабораторной работы

1. Реализация циклов в NASM

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

Figure 1: Создание папки и файла

Figure 1: Создание папки и файла

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

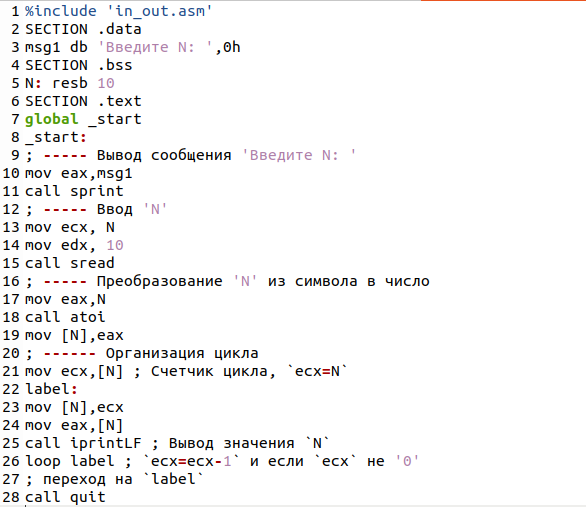


Figure 2: Содержание файла lab8-1.asm

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

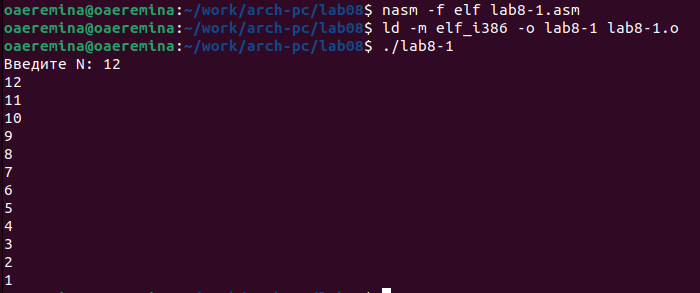


Figure 3: Выподнение программы

Изменяю код программы, добавив изменение значение регистра ecx в цикле. (рис.4)

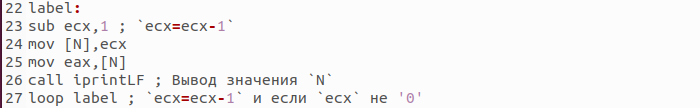


Figure 4: Изменение программы

Создаю исполняемый файл и проверяю работу программы. (рис.5) Можно заметить, что в первом случае программа уменьшала значения на 1, начиная с вводимого числа (в моем случае 12), во втором же случае на 2 (пропуская вводимое число).

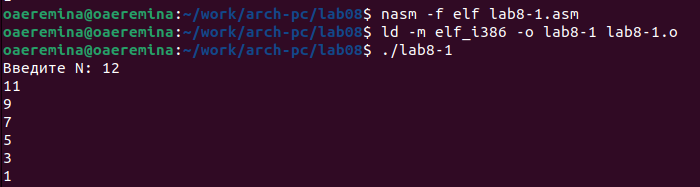


Figure 5: Выподнение программы

Изменяю код программы, добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop (рис.6)

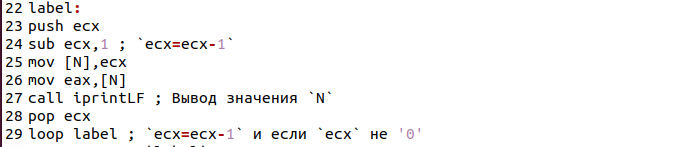


Figure 6: Изменение программы

Создаю исполняемый файл и проверяю работу программы. (рис.7) Программа уменьшает значения на 1, пропуская вводимое с клавиатуры число.

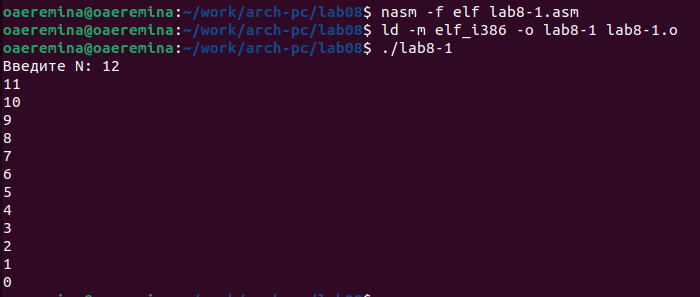


Figure 7: Выподнение программы

1. Обработка аргументов командной строки

Создаю файл lab8-2.asm в каталоге ~/work/arch-pc/lab08. Ввожу код программы в созданный файл. (рис.8)

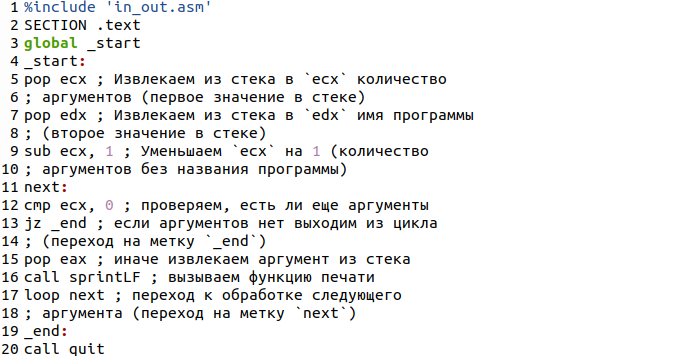


Figure 8: Содержание файла lab8-2.asm

Создаю исполняемый файл и провяю его работу для разных аргументов. (рис.9). Программа обработала все аргументы.

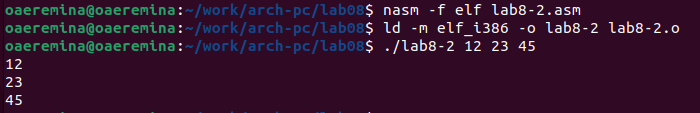


Figure 9: Выподнение программы

Создаю файл lab8-3.asm в каталоге ~/work/arch-pc/lab08. Ввожу код программы в созданный файл. (рис.10)

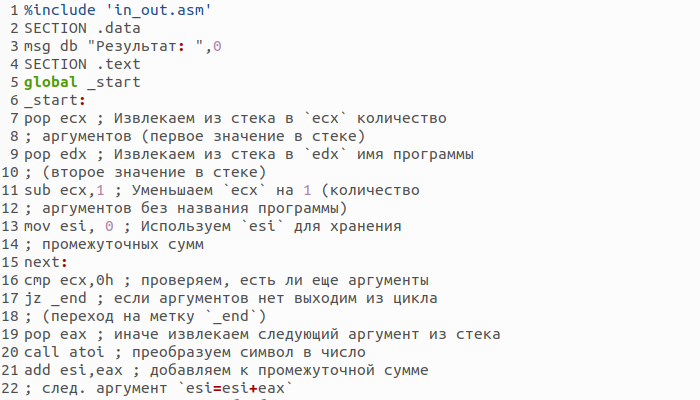


Figure 10: Содержание файла lab8-3.asm

Создаю исполняемый файл и провяю его работу. (рис.11). Программа обработала и сложила все аргументы.

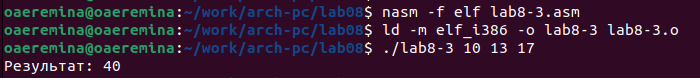


Figure 11: Выподнение программы

Изменяю программу так, чтобы она умножала, а не складывала аргументы. (рис.12)

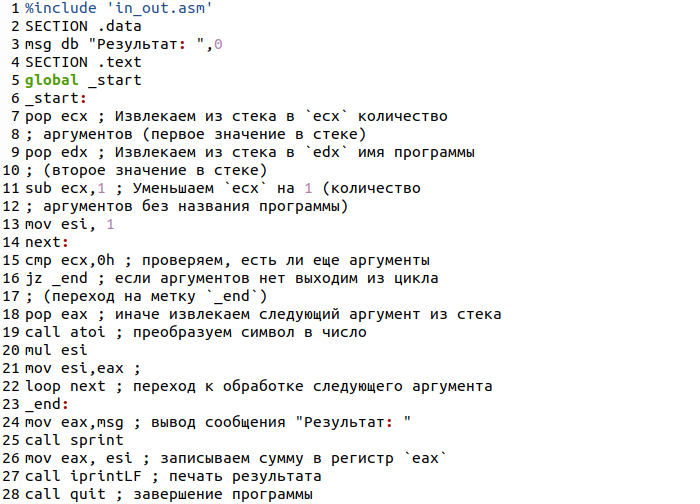


Figure 12: Изменение программы

Создаю исполняемый файл и провяю его работу. (рис.13). Программа обработала и умножила все аргументы.

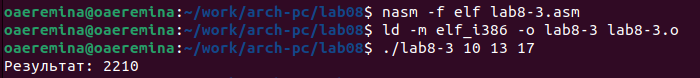


Figure 13: Выподнение программы

1. Выполнение заданий для самостоятельной работы

Пишу 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 переменных выбираю из табл. 8.1. Мой вариант под номером 17, поэтому мой вариант: 10(𝑥 − 1).(рис.14)

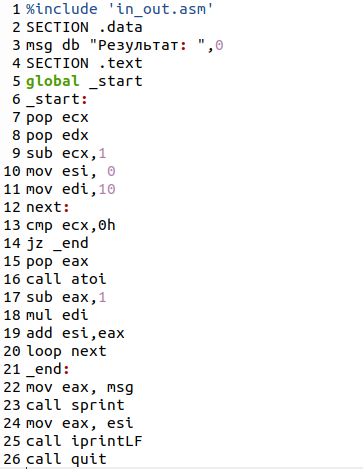


Figure 14: Код прогрраммы

Создаю исполняемый файл и проверяю работу программы, вводя разные значения (рис.15)

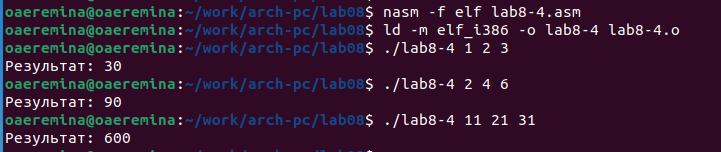


Figure 15: Выподнение программы

Код программы:

%include ‘in\_out.asm’

SECTION .data

msg db “Результат:”,0

SECTION .text

global \_start

\_start:

pop ecx

pop edx

sub ecx,1

mov esi, 0

mov edi,10

next:

cmp ecx,0h

jz \_end

pop eax

call atoi

sub eax,1

mul edi

add esi,eax

loop next

\_end:

mov eax, msg

call sprint

mov eax, esi

call iprintLF

call quit

# Вывод

При выполнении данной лабораторной работы я приобрела практические навыки написания программ использованием циклов и обработкой аргументов командной строки.

# Список литературы

Архитектура ЭВМ