

# **Лабораторная работа №8**

**Программирование цикла. Обработка аргументов командной строки.**

Казначеев Сергей Ильич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выполнение задания для самостоятельной работы</b>	<b>9</b>
<b>4</b>	<b>Выводы</b>	<b>10</b>

# Список иллюстраций

2.1	01	.....	6
2.2	02	.....	6
2.3	03	.....	6
2.4	04	.....	6
2.5	05	.....	6
2.6	06	.....	7
2.7	07	.....	7
2.8	08	.....	7
2.9	09	.....	7
2.10	10	.....	7
2.11	11	.....	7
2.12	12	.....	8
2.13	13	.....	8
2.14	14	.....	8
2.15	15	.....	8
2.16	16	.....	8
2.17	17	.....	8
2.18	18	.....	8
3.1	19	.....	9
3.2	20	.....	9
3.3	21	.....	9

## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Для начала создадим файл lab8-1.asm.

01

Рис. 2.1: 01

Далее запускаем Midnight commander через команду `mc` теперь вставляем в ранее созданный файл код из листинга 8.1. Он должен запускать цикл и выводить каждую итерацию числа , на единицу меньше предыдущего.

02

Рис. 2.2: 02

Копируем файл `in_out.asm`, чтобы собирать файл.

03

Рис. 2.3: 03

Теперь собираем программу и запускаем.

04

Рис. 2.4: 04

Как видим, она выводит числа от N до единицы включительно. Теперь пробуем изменить код, чтобы в цикле также отнималась единица.

05

Рис. 2.5: 05

Собираем файл и запускаем ее.

06

Рис. 2.6: 06

Введём в качестве N число 5 и посмотрим на результат выполнения.

07

Рис. 2.7: 07

Цикл выполняется бесконечно, если входное число нечетное, потому что условие остановки цикла  $esx=0$  никогда не будет достигнуто. Это происходит из-за того, что регистр  $esx$  уменьшается на 2 за каждую интеграцию. Если же входное число четное, то цикл выполняется  $N/2$  раз, выводя числа в порядке убавления от  $N-1$  до 1 с шагом 2.

08

Рис. 2.8: 08

Теперь пробуем изменить программу так, чтобы она сохраняла значение регистра  $esx$  в стек.

09

Рис. 2.9: 09

Пробуем собрать и запустить программу.

10

Рис. 2.10: 10

Теперь программа выводит все числа от  $N-1$  до нуля, далее создаем второй файл.

11

Рис. 2.11: 11

Затем вставляем код из файла листинга 8.2

12

Рис. 2.12: 12

Соберем и запустим его указав некоторые аргументы.

13

Рис. 2.13: 13

Создадим третий файл.

14

Рис. 2.14: 14

И вставляем в него код из листинга 8.3. Данная программа находит сумму всех аргументов.

15

Рис. 2.15: 15

Теперь собираем файл и запускаем его.

16

Рис. 2.16: 16

Как видим программа выводит сумму всех аргументов.Изменим её так, чтобы она находила не сумму, а произведение всех аргументов

17

Рис. 2.17: 17

Собираем программу и запускаем ее.

18

Рис. 2.18: 18

Как выдим программа выведет правильнойей ответ.



### 3 Выполнение задания для самостоятельной работы

Для выполнения самостоятельной работы создадим файл в формате .asm

19

Рис. 3.1: 19

В рамках самостоятельной работы необходимо сделать задание под вариантом 10. Там, необходимо сложить результаты выполнения функции  $f(x)=5(2+x)$  для всех введённых аргументов.

20

Рис. 3.2: 20

Собираем и запускаем программу, вводя различные аргументы.

21

Рис. 3.3: 21

Пересчитав результат вручную, убеждаемся что программа работает верно.

## **4 Выводы**

В результате выполнения лабораторной работы я приобрел навыки написания программ с использованием циклов и обработкой аргументов командной строки.