

Лабораторная работа 9

Программирование цикла. Обработка аргументов командной строки.

Зайцев Никита Кириллович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	8

Список иллюстраций

2.1	touch	5
2.2	программа	5
2.3	программа,где -2	6
2.4	программа,где -1	6
2.5	обработка аргументов	6
2.6	сложение	6
2.7	умножение	7
2.8	умножение	7

1 Цель работы

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

2 Выполнение лабораторной работы

1. Создаю файл для программирования

```
nkzayjcev@dk8n60 ~ $ mkdir ~/work/arch-pc/lab09
nkzayjcev@dk8n60 ~ $ cd ~/work/arch-pc/lab09
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ touch lab9-1.asm
```

Рис. 2.1: touch

2. Добавляя в созданный файл текст листинга получаю программу ,которая циклично выводит числа,начиная от записанного числа , уменьшая его на 1.

```
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 8
8
7
6
5
4
3
2
1
```

Рис. 2.2: программа

3. Изменив цикл программа начала выводить цифры начиная с числа-1 и последующие -2

```

nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 8
7
5
3
1
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ █

```

Рис. 2.3: программа, где -2

4. Еще раз немного изменив получаю так ,что цикл начал выводить число-1 ,вычитая из последующего -1 и до 0 включительно, кол-во совпадает

```

nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 8
7
6
5
4
3
2
1
0
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ █

```

Рис. 2.4: программа, где -1

5. Создав новую программу и вписав туда аргументы она обработала все 3, выписав их

```

nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ touch lab9-2.asm
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ nasm -f elf lab9-2.asm
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-2 lab9-2.o
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ ./lab9-2 2 5 '8'
2
5
8

```

Рис. 2.5: обработка аргументов

6. Сделав новую программу для сложения аргументов, она сложила их все и вывела результат

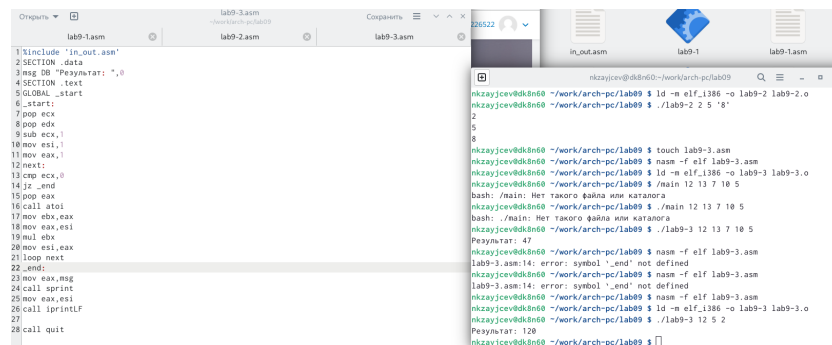
```

nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ ./lab9-3 12 13 7 10 5
Результат: 47
nkzayjcev@dk8n60 ~/work/arch-pc/lab09 $ █

```

Рис. 2.6: сложение

7. Добавив команду `mul` и убрав сложение появилась программа ,которая перемножит все аргументы.

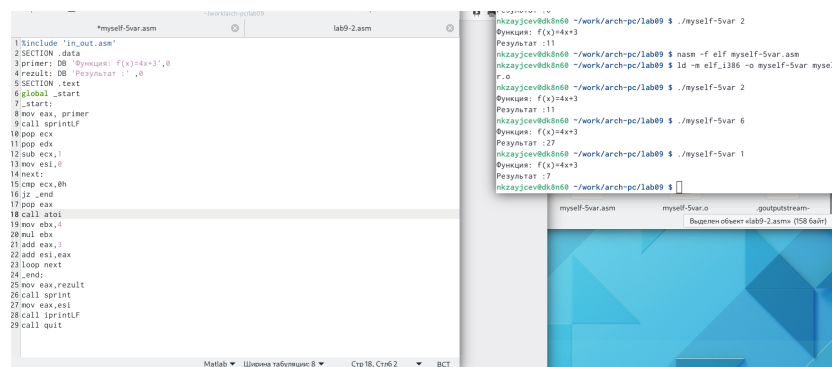


```
1#include "in_out.asm"
2SECTION .data
3neg DB "Результат: ",0
4SECTION .text
5GLOBAL _start
6_start:
7push ecx
8pop ecx
9sub ecx,1
10mov esi,1
11mov eax,1
12next:
13cmp ecx,0
14je _end
15pop eax
16call atoi
17mov ebx,eax
18mov esi,esi
19mul ebx
20mov esi,eax
21loop next
22_end:
23mov eax,neg
24call sprintf
25mov esi,esi
26call printf
27
28call quit
```

```
nkzajcev@dkn60:~/work/arch-pc/lab09 $ ./lab9-3.asm 2 2 5 8
Результат: 120
nkzajcev@dkn60:~/work/arch-pc/lab09 $
```

Рис. 2.7: умножение

Задание для самостоятельной работы варианта 5



```
1#include "in_out.asm"
2SECTION .data
3priner: DB "Функция: f(x)=4x+3",0
4result: DB "Результат: ",0
5SECTION .text
6GLOBAL _start
7_start:
8mov ecx,priner
9call sprintf
10pop ecx
11pop ecx
12sub ecx,1
13mov esi,0
14next:
15cmp ecx,0h
16je _end
17pop ecx
18call atoi
19mov ebx,1
20mul ebx
21add esi,ebx
22add esi,esi
23loop next
24_end:
25mov eax,result
26call sprintf
27mov esi,esi
28call printf
29call quit
```

```
nkzajcev@dkn60:~/work/arch-pc/lab09 $ ./myself-Svar 2
Функция: f(x)=4x+3
Результат: 11
nkzajcev@dkn60:~/work/arch-pc/lab09 $ ./myself-Svar 6
Функция: f(x)=4x+3
Результат: 27
nkzajcev@dkn60:~/work/arch-pc/lab09 $ ./myself-Svar 1
Функция: f(x)=4x+3
Результат: 7
nkzajcev@dkn60:~/work/arch-pc/lab09 $
```

Рис. 2.8: умножение

3 Выводы

За время, которое я делал лабораторную работу я научился создавать программы с циклом, а также использовать его в самых различных ситуациях.