

# **Лабораторная работа No8. Программирование цикла.**

**Обработка аргументов командной строки.**

Ромицына Анастасия Романовна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Задания для самостоятельной работы</b>	<b>13</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>

## Список иллюстраций

3.1	Создание 1 файла . . . . .	7
3.2	Редактирование 1 файла . . . . .	7
3.3	Вносим данные . . . . .	8
3.4	Запуск программы . . . . .	8
3.5	Редактирование файла . . . . .	9
3.6	Проверка программы . . . . .	9
3.7	Редактирование файла . . . . .	10
3.8	Проверка программы . . . . .	10
3.9	Создание нового файла . . . . .	10
3.10	Редактирование файла . . . . .	11
3.11	Запуск программы . . . . .	11
3.12	Заполнение нового файла . . . . .	12
3.13	Запуск программы . . . . .	12
4.1	Создание нового файла . . . . .	13
4.2	Написание новой программы . . . . .	14
4.3	Запуск программы(все верно) . . . . .	14

## **Список таблиц**

# 1 Цель работы

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

## **2 Задание**

### 3 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы No 8, перейдем в него и создадим файл lab8-1.asm (рис. 3.1).

```
[romitsinaar@fedora arch-pc]$ mkdir ~/work/arch-pc/lab08  
[romitsinaar@fedora arch-pc]$ cd ~/work/arch-pc/lab08  
[romitsinaar@fedora lab08]$ touch lab8-1.asm
```

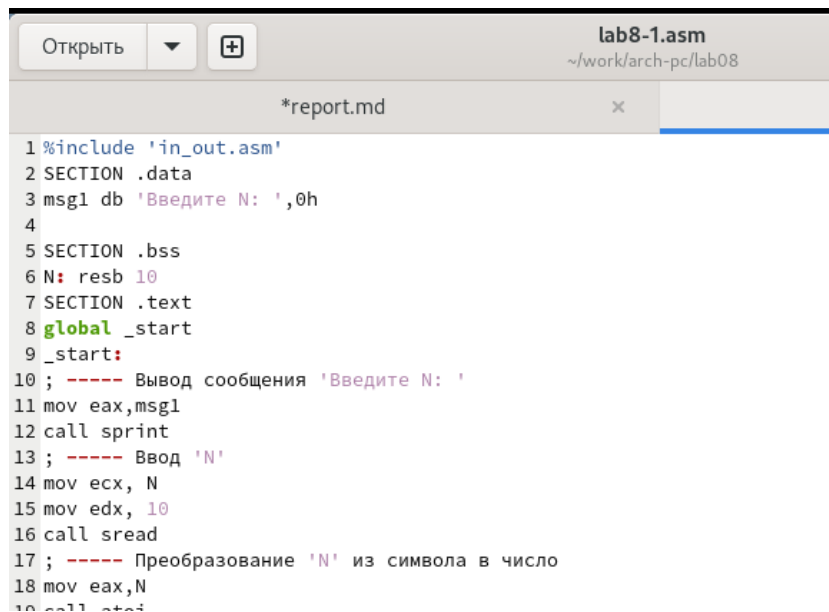
Рис. 3.1: Создание 1 файла

Отредактируем созданный файл (рис. 3.2).

```
[romitsinaar@fedora lab08]$ gedit lab8-1.asm
```

Рис. 3.2: Редактирование 1 файла

Откроем его и внесем программу из листинга (рис. 3.3).



```
lab8-1.asm
~/work/arch-pc/lab08

*report.md x

1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ',0h
4
5 SECTION .bss
6 N: resb 10
7 SECTION .text
8 global _start
9 _start:
10 ; ----- Вывод сообщения 'Введите N: '
11 mov eax,msg1
12 call sprint
13 ; ----- Ввод 'N'
14 mov ecx, N
15 mov edx, 10
16 call sread
17 ; ----- Преобразование 'N' из символа в число
18 mov eax,N
19 call atoi
```

Рис. 3.3: Вносим данные

Запустим созданный файл (рис. 3.4).



```
[romitsinaar@fedora lab08]$ nasm -f elf lab8-1.asm
[romitsinaar@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[romitsinaar@fedora lab08]$ ./lab8-1
Введите N: 13
13
12
11
10
9
8
7
6
5
4
3
2
1
[romitsinaar@fedora lab08]$
```

Рис. 3.4: Запуск программы

Отредактируем созданный файл (рис. 3.5).



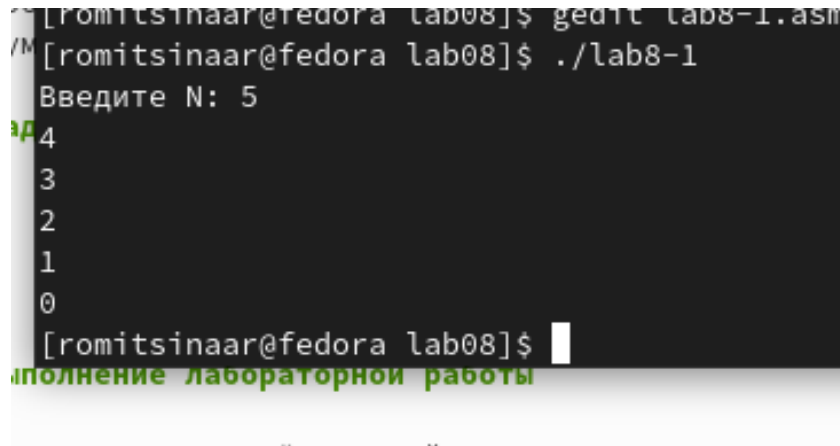
```

23 label:
24 sub ecx,1 ; `ecx=ecx-1`
25 mov [N],ecx
26 mov eax,[N]
27 call iprintLF
28 loop label; `ecx=ecx-1` и если `ecx` не '0'
29 ; переход на `label`
30 call quit

```

Рис. 3.5: Редактирование файла

Запустим отредактированный файл, числа начинаются не с заданного, а с числа на 1 меньше и заканчиваются 0, а не 1(рис. 3.6).



```

[romitsinaar@fedora lab08]$ gedit lab8-1.asm
[romitsinaar@fedora lab08]$ ./lab8-1
Введите N: 5
4
3
2
1
0
[romitsinaar@fedora lab08]$

```

Рис. 3.6: Проверка программы

Отредактируем файл по примеру (рис. 3.7).

```

20 mov [N],eax
21 ; ----- Организация цикла
22 mov ecx,[N] ; Счетчик цикла, `ecx=N`
23 label:
24 push ecx ; добавление значения ecx в стек
25 sub ecx,1
26 mov [N],ecx
27 mov eax,[N]
28 call iprintLF
29 pop ecx ; извлечение значения ecx из стека
30 loop label `ecx=ecx-1` и если `ecx` не `0`
31 ; переход на `label`
32 call quit

```

Рис. 3.7: Редактирование файла

Запустим отредактированный файл, количество значений соответствует заданному N(рис. 3.8).

```

[romitsinaar@fedora lab08]$ nasm -f elf lab8-1.asm
[romitsinaar@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[romitsinaar@fedora lab08]$ ./lab8-1
Введите N: 13
12
11
10
9
8
7
6
5
4
3
2
1
0
[romitsinaar@fedora lab08]$

```

Рис. 3.8: Проверка программы

Создадим новый файл(рис. 3.9).

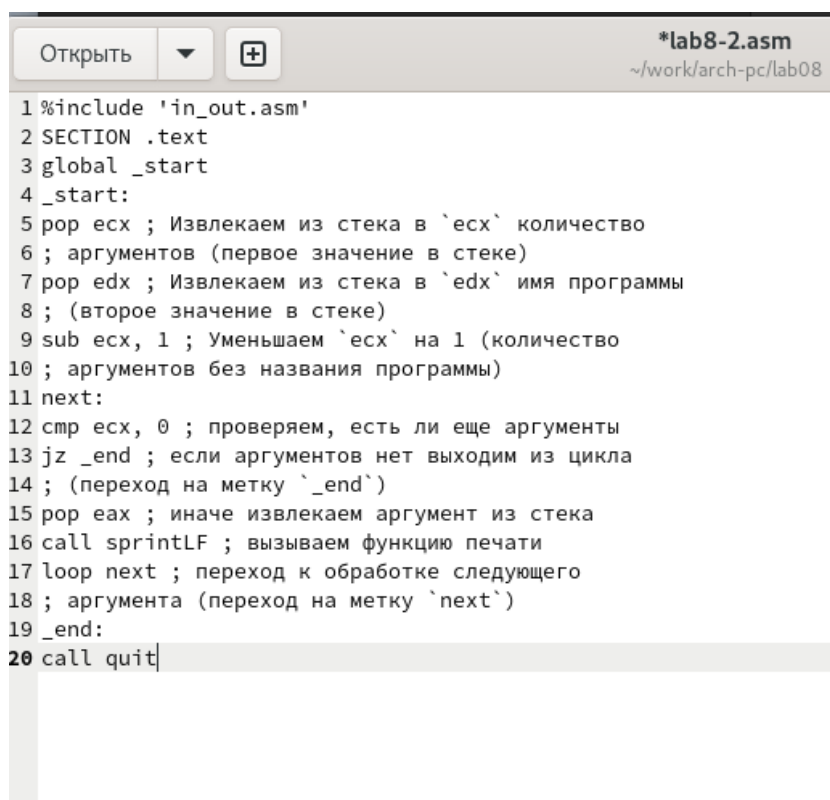
```

[romitsinaar@fedora lab08]$ touch lab8-2.asm
[romitsinaar@fedora lab08]$ gedit lab8-2.asm

```

Рис. 3.9: Создание нового файла

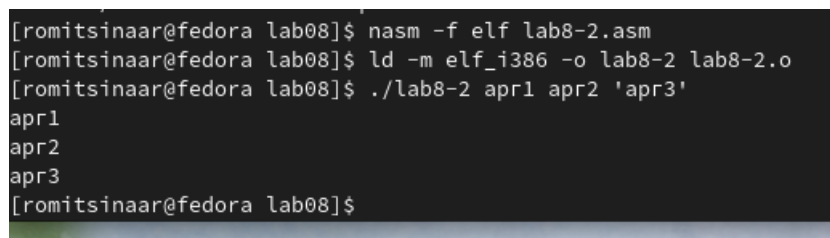
Внесем в файл программу из примера (рис. 3.10).



```
Открыть ▼ + *lab8-2.asm
~/work/arch-pc/lab08
1 %include 'in_out.asm'
2 SECTION .text
3 global _start
4 _start:
5 pop ecx ; Извлекаем из стека в `ecx` количество
6 ; аргументов (первое значение в стеке)
7 pop edx ; Извлекаем из стека в `edx` имя программы
8 ; (второе значение в стеке)
9 sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество
10 ; аргументов без названия программы)
11 next:
12 cmp ecx, 0 ; проверяем, есть ли еще аргументы
13 jz _end ; если аргументов нет выходим из цикла
14 ; (переход на метку `_end`)
15 pop eax ; иначе извлекаем аргумент из стека
16 call printf ; вызываем функцию печати
17 loop next ; переход к обработке следующего
18 ; аргумента (переход на метку `next`)
19 _end:
20 call quit
```

Рис. 3.10: Редактирование файла

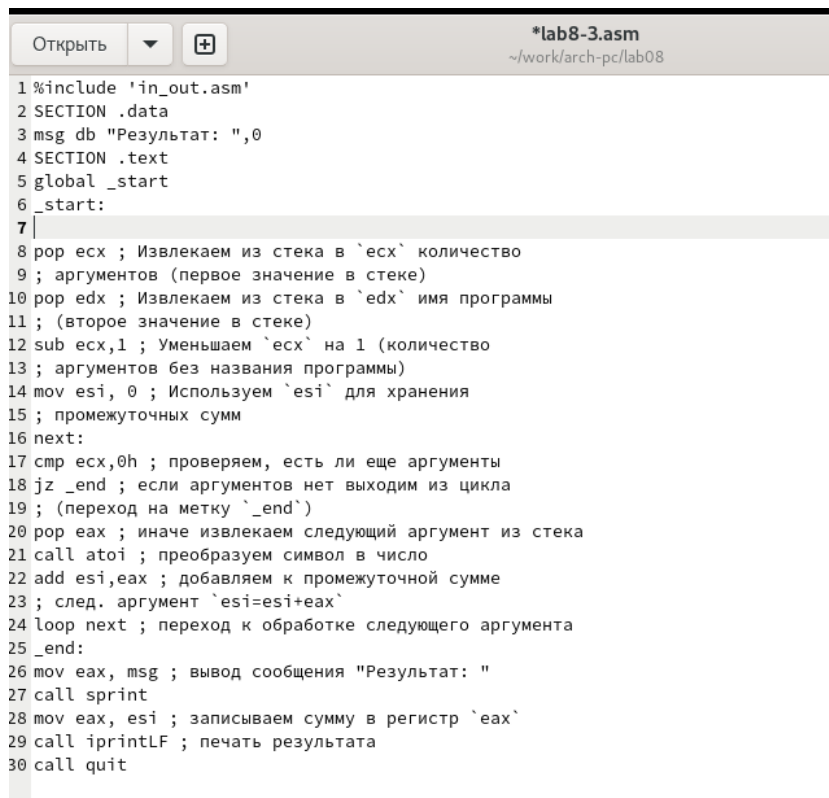
Запустим отредактированный файл, все значения выводятся (рис. 3.11).



```
[romitsinaar@fedora lab08]$ nasm -f elf lab8-2.asm
[romitsinaar@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[romitsinaar@fedora lab08]$ ./lab8-2 arg1 arg2 'arg3'
arg1
arg2
arg3
[romitsinaar@fedora lab08]$
```

Рис. 3.11: Запуск программы

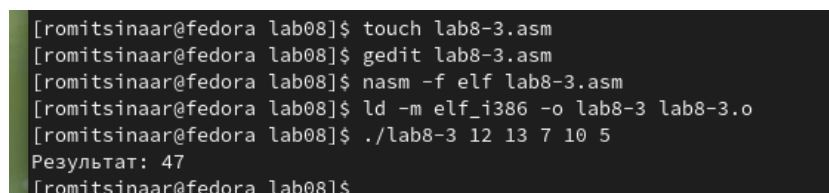
Создадим новый файл и заполним его программой и листинга (рис. 3.12).



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 SECTION .text
5 global _start
6 _start:
7
8 pop ecx ; Извлекаем из стека в `ecx` количество
9 ; аргументов (первое значение в стеке)
10 pop edx ; Извлекаем из стека в `edx` имя программы
11 ; (второе значение в стеке)
12 sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
13 ; аргументов без названия программы)
14 mov esi, 0 ; Используем `esi` для хранения
15 ; промежуточных сумм
16 next:
17 cmp ecx,0h ; проверяем, есть ли еще аргументы
18 jz _end ; если аргументов нет выходим из цикла
19 ; (переход на метку `_end`)
20 pop eax ; иначе извлекаем следующий аргумент из стека
21 call atoi ; преобразуем символ в число
22 add esi,eax ; добавляем к промежуточной сумме
23 ; след. аргумент `esi=esi+eax`
24 loop next ; переход к обработке следующего аргумента
25 _end:
26 mov eax, msg ; вывод сообщения "Результат: "
27 call sprint
28 mov eax, esi ; записываем сумму в регистр `eax`
29 call iprintfLF ; печать результата
30 call quit
```

Рис. 3.12: Заполнение нового файла

Запустим отредактированный файл, все значения выводятся как в примере(рис. 3.13).

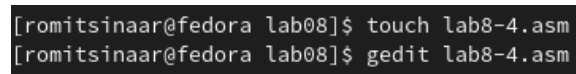


```
[romitsinaar@fedora lab08]$ touch lab8-3.asm
[romitsinaar@fedora lab08]$ gedit lab8-3.asm
[romitsinaar@fedora lab08]$ nasm -f elf lab8-3.asm
[romitsinaar@fedora lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[romitsinaar@fedora lab08]$ ./lab8-3 12 13 7 10 5
Результат: 47
[romitsinaar@fedora lab08]$
```

Рис. 3.13: Запуск программы

## 4 Задания для самостоятельной работы

Создадим новый файл и отредактируем его(рис. 4.1).



```
[romitsinaar@fedora lab08]$ touch lab8-4.asm  
[romitsinaar@fedora lab08]$ gedit lab8-4.asm
```

Рис. 4.1: Создание нового файла

Напишем программу (рис. 4.2).

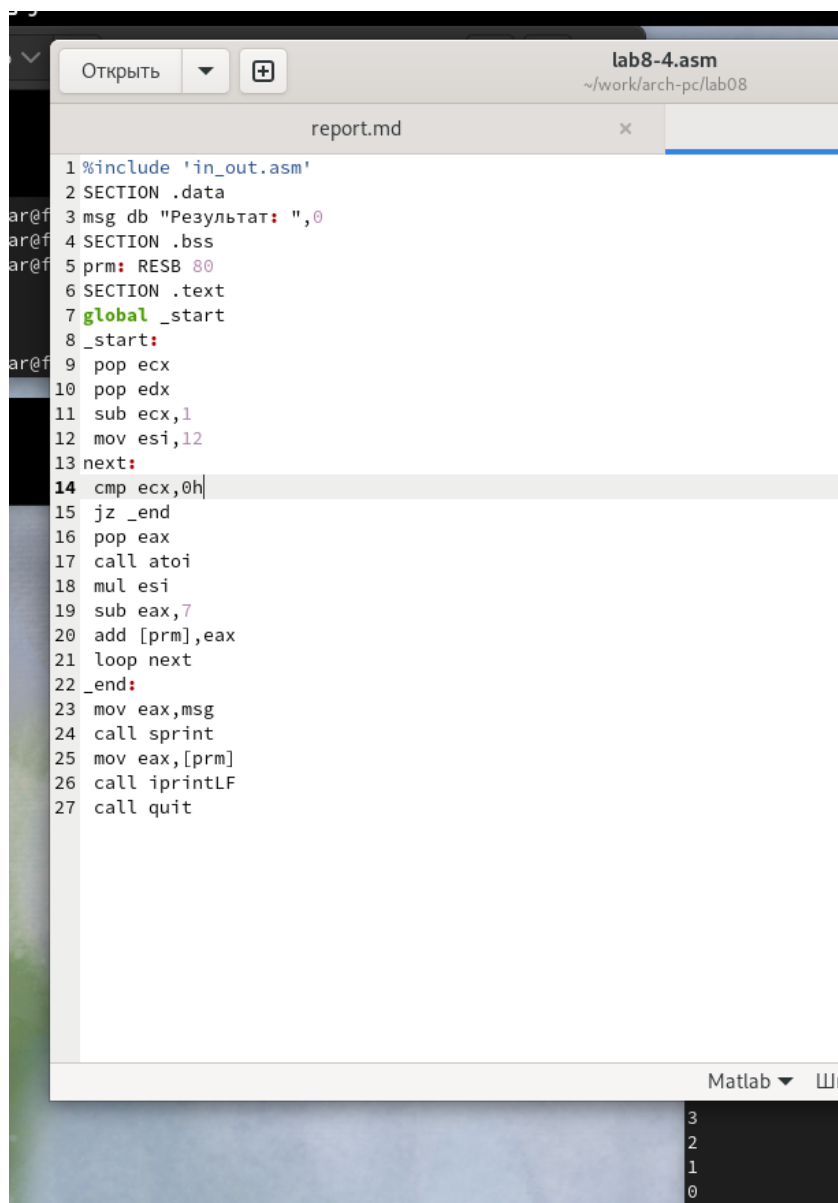


Рис. 4.2: Написание новой программы

Запустим программу и проверим ее работу (рис. 4.3).

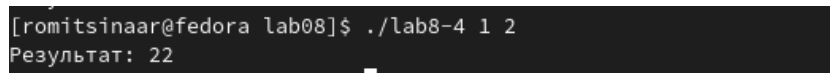


Рис. 4.3: Запуск программы(все верно)

## 5 Выводы

Мы приобрели навыки написания программ с использованием циклов и обработкой аргументов командной строки.