

**Лаборная работа №9 Лабораторная
работа №9. Программирование цикла.
Обработка аргументов командной
строки.**

НММ-6д-02-22

Крухмалев Артём Владиславович

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Самостоятельная работа	8
4	Выводы	9

1 Цель работы

Научиться работать с циклами и стэком. # Задание

Написать программу с использованием циклов

2 Выполнение лабораторной работы

1. С помощью терминала создадим подкаталог, создадим файл lab9-1.asm

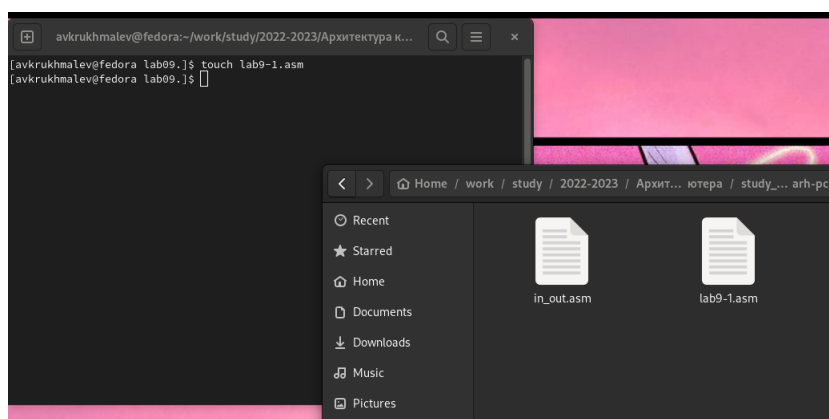
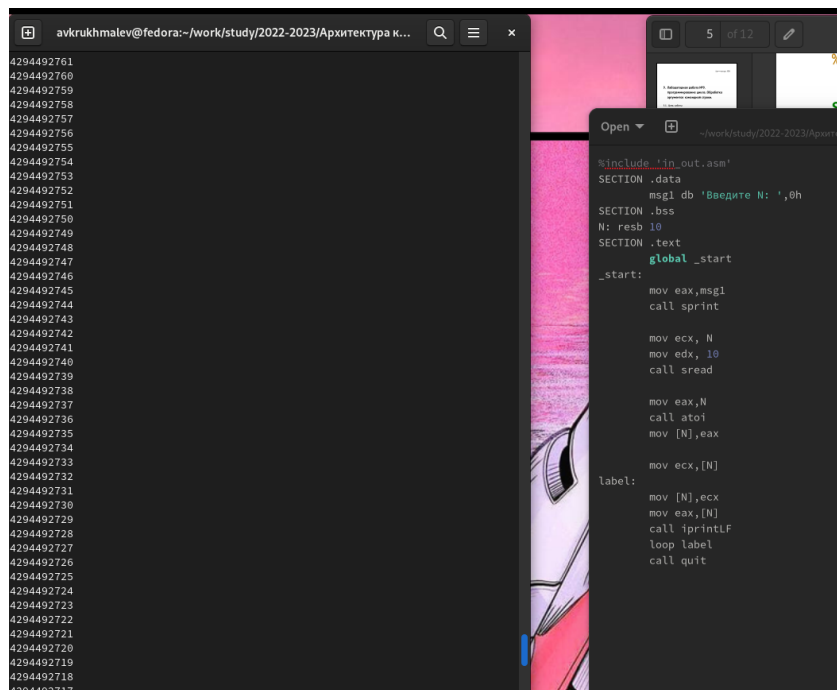


Рис. 2.1: Новый каталог

2. Изучим и запишем в него код из листинга, откомпилируем и запустим файл, заметим, что мы получили заикливание



```
4294492761
4294492760
4294492759
4294492758
4294492757
4294492756
4294492755
4294492754
4294492753
4294492752
4294492751
4294492750
4294492749
4294492748
4294492747
4294492746
4294492745
4294492744
4294492743
4294492742
4294492741
4294492740
4294492739
4294492738
4294492737
4294492736
4294492735
4294492734
4294492733
4294492732
4294492731
4294492730
4294492729
4294492728
4294492727
4294492726
4294492725
4294492724
4294492723
4294492722
4294492721
4294492720
4294492719
4294492718
4294492717
```

```
%include 'in_out.asm'
SECTION .data
    msg1 db 'Введите N: ',0h
SECTION .bss
    N: resb 10
SECTION .text
    global _start
_start:
    mov eax,msg1
    call sprint

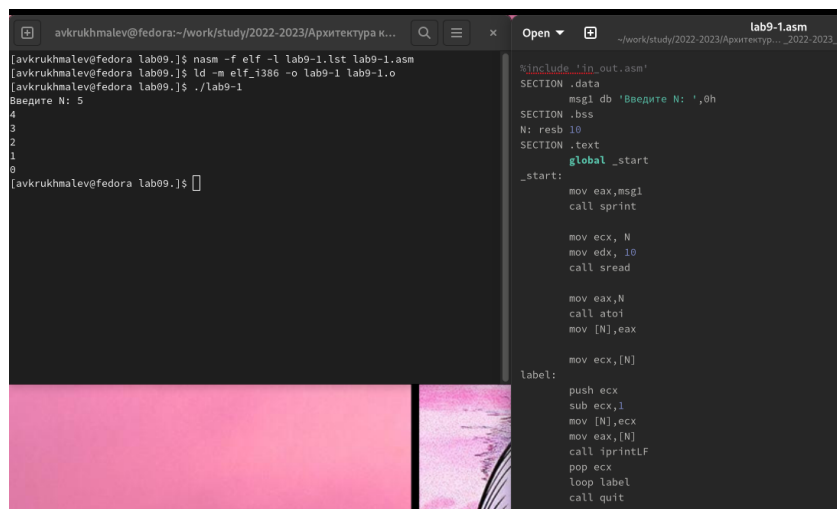
    mov ecx, N
    mov edx, 10
    call sread

    mov eax,N
    call atoi
    mov [N],eax

    mov ecx,[N]
label:
    mov [N],ecx
    mov eax,[N]
    call iprintfLF
    loop label
    call quit
```

Рис. 2.2: Зацикливание

3. Изменим код программы и посмотрим, что он нам выведет. Действительно количество значений удовлетворяет введенному числу, но т.к. вывод начинается с 0, он дойдет только до 4.



```
avkrukhmalev@fedora:~/work/study/2022-2023/Архитектура к...
avkrukhmalev@fedora lab09:~$ nasm -f elf -l lab9-1.lst lab9-1.asm
avkrukhmalev@fedora lab09:~$ ld -m elf_i386 -o lab9-1 lab9-1.o
avkrukhmalev@fedora lab09:~$ ./lab9-1
Введите N: 5
4
3
2
1
0
avkrukhmalev@fedora lab09:~$
```

```
%include 'in_out.asm'
SECTION .data
    msg1 db 'Введите N: ',0h
SECTION .bss
    N: resb 10
SECTION .text
    global _start
_start:
    mov eax,msg1
    call sprint

    mov ecx, N
    mov edx, 10
    call sread

    mov eax,N
    call atoi
    mov [N],eax

    mov ecx,[N]
label:
    push ecx
    sub ecx,1
    mov [N],ecx
    mov eax,[N]
    call iprintfLF
    pop ecx
    loop label
    call quit
```

Рис. 2.3: Обработка аргументов

4. Создадим новый файл, запишем в него предложенный код, предварительно

изучив его. Было обработано 4 аргумента,



The image shows a terminal window on the left and an assembly editor on the right. The terminal displays the execution of a program with arguments: `[avkrukhmalev@fedora lab09.]$./lab9-2 аргумент1 аргумент 2 'аргумент 3'`. The assembly editor shows the following code:

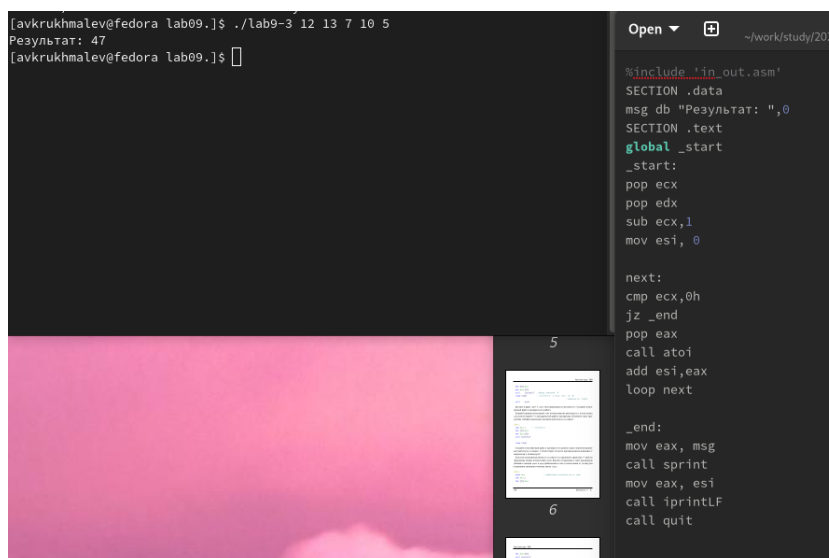
```
%include 'in_out.asm'
SECTION .text
global _start
_start:
    pop ecx
    pop edx
    sub ecx, 1

next:
    cmp ecx, 0
    jz _end
    pop eax
    call sprintf
    loop next

_end:
    call quit
```

Рис. 2.4: Работа с аргументами

5. Напишем программу для вычисления суммы.



The image shows a terminal window on the left and an assembly editor on the right. The terminal displays the execution of a program with arguments: `[avkrukhmalev@fedora lab09.]$./lab9-3 12 13 7 10 5`. The output is `Результат: 47`. The assembly editor shows the following code:

```
%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

next:
    cmp ecx, 0h
    jz _end
    pop eax
    call atoi
    add esi, eax
    loop next

_end:
    mov eax, msg
    call sprint
    mov eax, esi
    call iprintf
    call quit
```

Рис. 2.5: Простейшая программа

6. Изменим код, чтобы найти произведение введенных значений. Не забудем изменить начальное значение `esi` с 0 на 1.

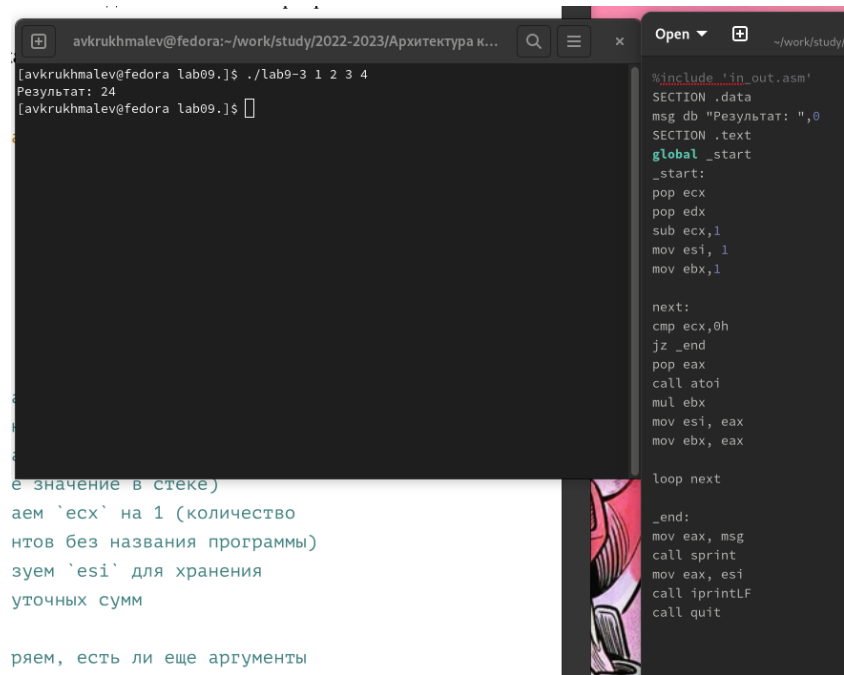
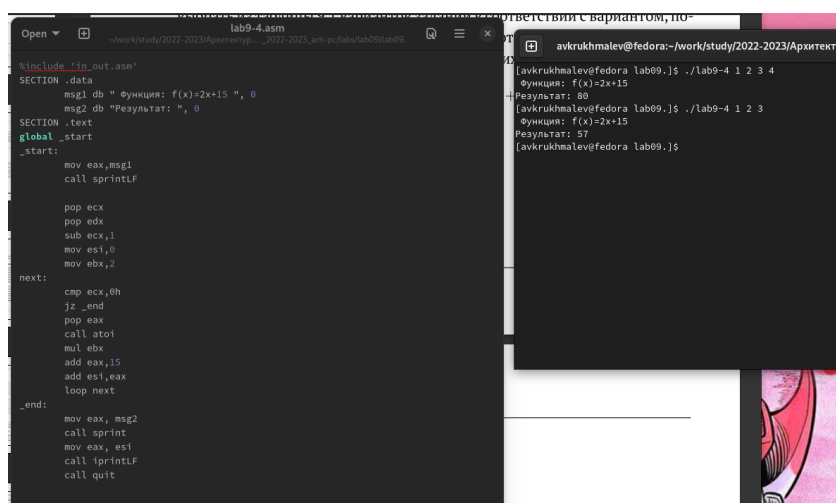


Рис. 2.6: Умножение

3 Самостоятельная работа

1. Мне попался 1 вариант, напишем код и выведем результат



The screenshot shows a code editor with assembly code for a program named 'lab9-4.asm'. The code includes a data section with two messages, a text section, and a loop that calculates the sum of the first four integers (1+2+3+4) and prints the result. The output window shows the program's execution, displaying the function definition and the calculated result of 10.

```
%include "in_out.asm"
SECTION .data
    msg1 db " Функция: f(x)=2x+15 ", 0
    msg2 db "Результат: ", 0
SECTION .text
    global _start
_start:
    mov eax, msg1
    call sprintf
    pop ecx
    pop edx
    sub ecx, 1
    mov esi, 0
    mov ebx, 2
next:
    cmp ecx, 0h
    jz _end
    pop eax
    call atoi
    mul ebx
    add eax, 15
    add esi, eax
    loop next
_end:
    mov eax, msg2
    call sprintf
    mov eax, esi
    call iprintf
    call quit
```

```
[avkrukhmalev@fedora lab09]$ ./lab9-4 1 2 3 4
Функция: f(x)=2x+15
Результат: 10
[avkrukhmalev@fedora lab09]$ ./lab9-4 1 2 3
Функция: f(x)=2x+15
Результат: 57
[avkrukhmalev@fedora lab09]$
```

Рис. 3.1: Самостоятельная работа

4 Выводы

В данной работе мы познакомились с циклами и научились вводить аргументы напрямую