

Отчёт по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки.

Юсупова Ксения Равилевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Реализация циклов в NASM	7
3.2	Обработка аргументов командной строки	10
3.3	Задание для самостоятельной работы	12
4	Выводы	15

Список иллюстраций

3.1	Создали каталог с помощью команды <code>mkdir</code> и файл <code>lab8-1.asm</code> с помощью команды <code>touch</code>	7
3.2	Заполнили файл по листингу 8.1	7
3.3	Создали исполняемый файл и запустили его	8
3.4	Изменили файл	8
3.5	Создали исполняемый файл и запустили его	8
3.6	Изменили файл <code>lab8-1.asm</code>	9
3.7	Создали исполняемый файл и запустили его	9
3.8	Создали файл <code>lab8-2.asm</code>	10
3.9	Заполнили файл по листингу 8.2	10
3.10	Создали исполняемый файл и запустили его	10
3.11	Создали файл <code>lab8-3.asm</code>	10
3.12	Заполнили файл по листингу 8.3	11
3.13	Создали исполняемый файл и запустили его	11
3.14	Изменили текст программы в файле <code>lab8-2.asm</code>	12
3.15	Создали исполняемый файл и запустили его	12
3.16	Создали файл <code>lab8-4.asm</code>	13
3.17	Заполнили файл <code>lab8-4.asm</code> в соответствии с условием задачи . . .	13
3.18	Создали исполняемый файл и запустили его, проверив вывод . . .	13
3.19	Вновь создали исполняемый файл и запустили его, проверив вывод с иными значениями <code>x1</code> , <code>x2</code> , <code>x3</code>	14

Список таблиц

1 Цель работы

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

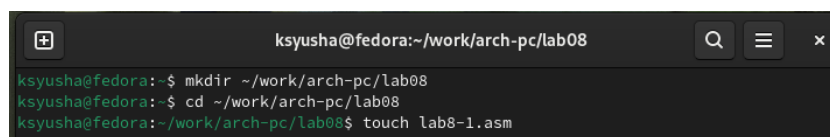
2 Задание

Написать программы с использованием циклов и обработкой аргументов командной строки.

3 Выполнение лабораторной работы

3.1 Реализация циклов в NASM

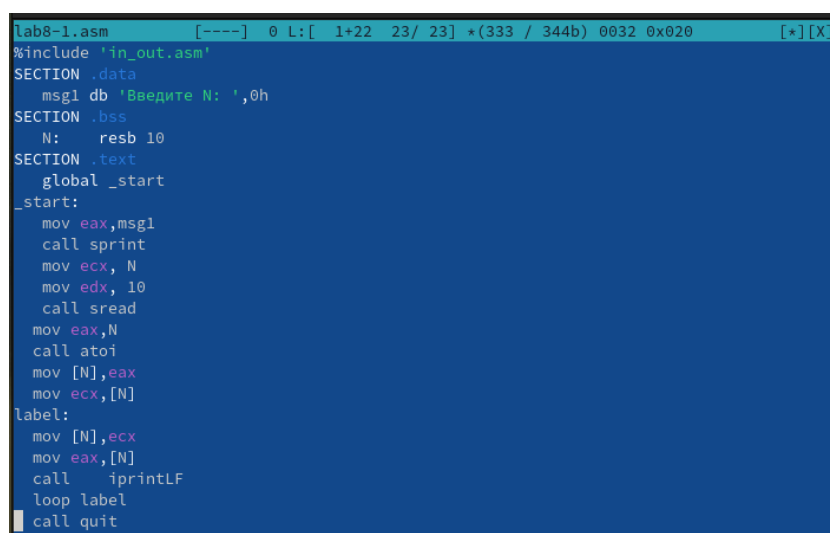
Создаём каталог для программ лабораторной работы № 8, переходим в него и создаём файл lab8-1.asm(рис. 3.1).



```
ksyusha@fedora:~/work/arch-pc/lab08
ksyusha@fedora:~$ mkdir ~/work/arch-pc/lab08
ksyusha@fedora:~$ cd ~/work/arch-pc/lab08
ksyusha@fedora:~/work/arch-pc/lab08$ touch lab8-1.asm
```

Рис. 3.1: Создали каталог с помощью команды mkdir и файл lab8-1.asm с помощью команды touch

Вводим в файл lab8-1.asm текст программы из листинга 8.1. и рассмотрим программу, которая выводит значение регистра esx. (рис. 3.2).



```
lab8-1.asm [----] 0 L: [ 1+22 23/ 23] *(333 / 344b) 0032 0x020 [*][X]
%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 iprintLF
    loop label
    call quit
```

Рис. 3.2: Заполнили файл по листингу 8.1

Создаем исполняемый файл и запускаем его(рис. 3.3).

```
ksyusha@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
ksyusha@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
ksyusha@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
```

Рис. 3.3: Создали исполняемый файл и запустили его

Далее изменяем текст программы, добавив изменение значения регистра есх в цикле.(рис. 3.4).

```
lab8-1.asm [-M--] 12 L:[ 1+22 23/ 24] *(341 / 353b) 0010 0x00A [*]
%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:
    sub ecx,1
    mov [N],ecx
    mov eax,[N]
    call iprintLF
    loop label
    call quit
```

Рис. 3.4: Изменили файл

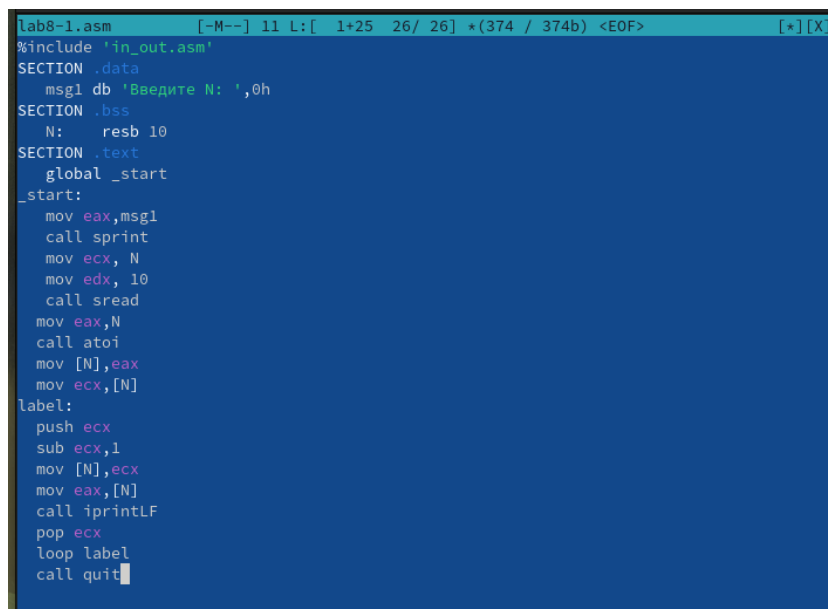
Создаем исполняемый файл и запускаем его((рис. 3.5).

```
ksyusha@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
ksyusha@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
ksyusha@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
7
5
3
1
ksyusha@fedora:~/work/arch-pc/lab08$
```

Рис. 3.5: Создали исполняемый файл и запустили его

Регистр `ecx` принимает значения 9,7,5,3,1, регистр при каждой итерации уменьшается на 2. Число проходов цикла не соответствует числу `N` введенному с клавиатуры.

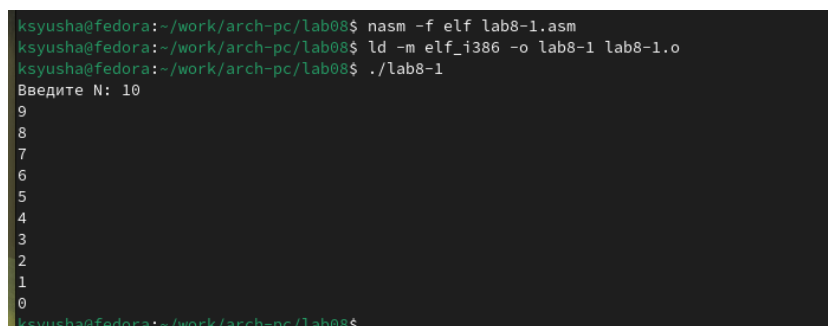
Далее также как и ранее изменяем текст программы, добавив команды `push` и `pop`(рис. 3.6).



```
lab8-1.asm [-M--] 11 L: [ 1+25 26/ 26] *(374 / 374b) <EOF> [*][X]
#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 iprintLF
    pop ecx
    loop label
    call quit
```

Рис. 3.6: Изменили файл `lab8-1.asm`

Создаем исполняемый файл и запускаем его(рис. 3.7).



```
ksyusha@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
ksyusha@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
ksyusha@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
ksyusha@fedora:~/work/arch-pc/lab08$
```

Рис. 3.7: Создали исполняемый файл и запустили его

Число проходов цикла соответствует значению `N` введенному с клавиатуры в данном случае.

3.2 Обработка аргументов командной строки

Создаем новый файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 с помощью команды touch (рис. 3.8).

```
ksyusha@fedora:~/work/arch-pc/lab08$ touch lab8-2.asm
ksyusha@fedora:~/work/arch-pc/lab08$
```

Рис. 3.8: Создали файл lab8-2.asm

Вводим в файл lab8-2.asm текст программы из листинга 8.2 (рис. 3.9).

```
lab8-2.asm [-M--] 12 L: [ 1+14 15/ 15] *(185 / 185b) <EOF> [
#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 sprintLF
    loop next
_end:
    call quit
```

Рис. 3.9: Заполнили файл по листингу 8.2

Создаем исполняемый файл и запускаем его, указав аргументы (рис. 3.10).

```
ksyusha@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
ksyusha@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
ksyusha@fedora:~/work/arch-pc/lab08$ ./lab8-2 1 2 '3'
1
2
3
```

Рис. 3.10: Создали исполняемый файл и запустили его

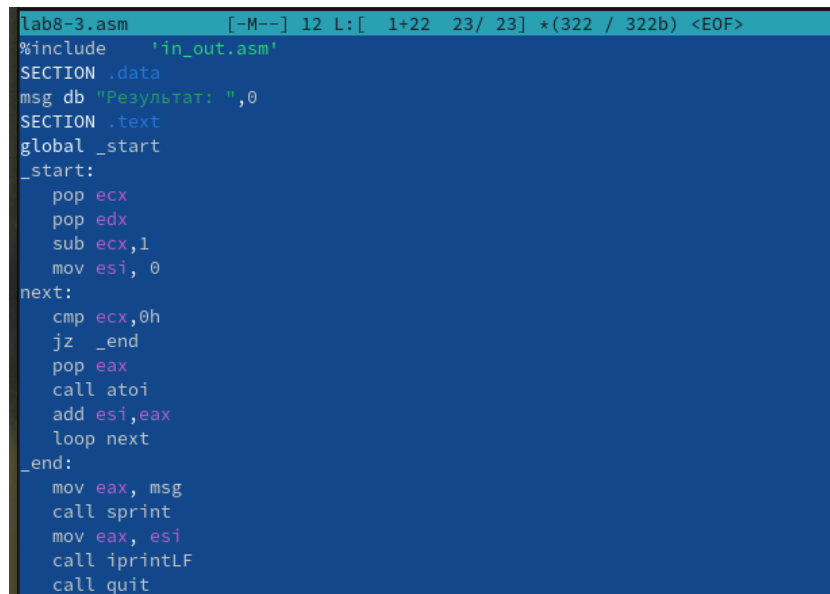
Программой было обработано 3 аргумента.

Создаем новый файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 с помощью команды touch (рис. 3.11).

```
ksyusha@fedora:~/work/arch-pc/lab08$ touch lab8-3.asm
ksyusha@fedora:~/work/arch-pc/lab08$
```

Рис. 3.11: Создали файл lab8-3.asm

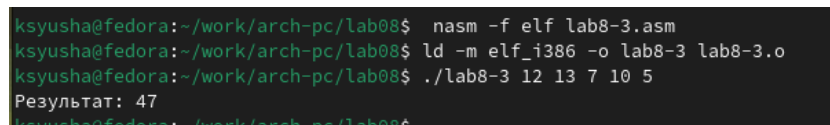
Вводим в файл lab8-3.asm текст программы из листинга 8.3(рис. 3.12).



```
lab8-3.asm [-M--] 12 L:[ 1+22 23/ 23] *(322 / 322b) <EOF>
#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 iprintLF
    call quit
```

Рис. 3.12: Заполнили файл по листингу 8.3

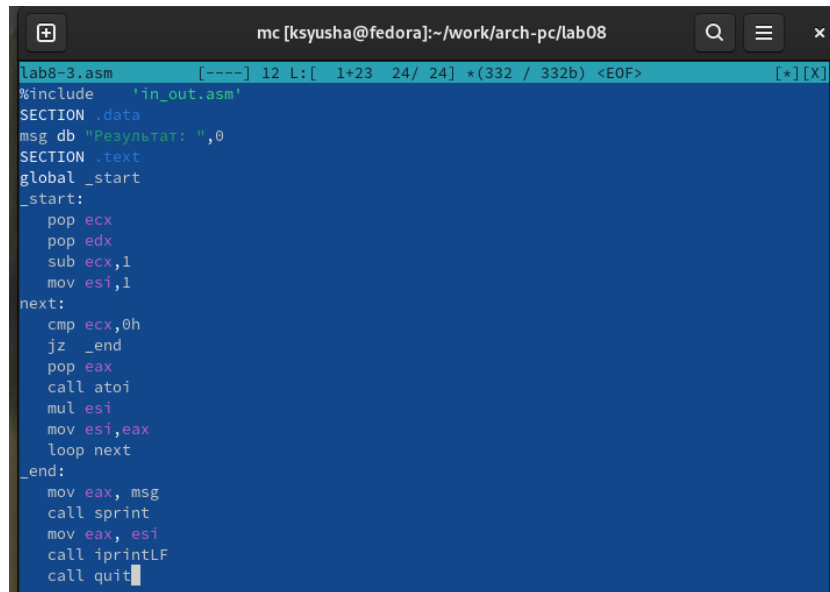
Создаем исполняемый файл и запускаем его(рис. 3.13).



```
ksyusha@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
ksyusha@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
ksyusha@fedora:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
ksyusha@fedora:~/work/arch-pc/lab08$
```

Рис. 3.13: Создали исполняемый файл и запустили его

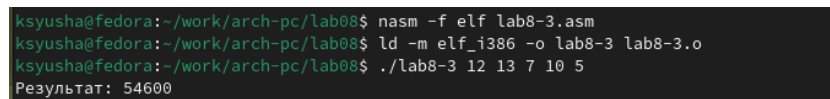
Изменяем текст программы из листинга 8.3 для вычисления произведения аргументов командной строки.(рис. 3.14).



```
lab8-3.asm [----] 12 L: [ 1+23 24/ 24] *(332 / 332b) <EOF> [*] [X]
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx
    pop edx
    sub ecx,1
    mov esi,1
next:
    cmp ecx,0h
    jz _end
    pop eax
    call atoi
    mul esi
    mov esi,eax
    loop next
_end:
    mov eax,msg
    call sprint
    mov eax,esi
    call iprintLF
    call quit
```

Рис. 3.14: Изменили текст программы в файле lab8-2.asm

Создаем исполняемый файл и запускаем его(рис. 3.15).



```
ksyusha@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
ksyusha@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
ksyusha@fedora:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 54600
```

Рис. 3.15: Создали исполняемый файл и запустили его

3.3 Задание для самостоятельной работы

1. Напишите программу, которая находит сумму значений функции $f(x)$ для $x=x_1, x_2, \dots, x_n$ т.е. программа должна выводить значение $f(x_1)+f(x_2)+\dots+f(x_n)$. Значения x_i передаются как аргументы. Вид функции $f(x)$ выбрать из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах $x= x_1, x_2, \dots, x_n$.

Вид функции $f(x)$ выбран в соответствии с 12 вариантом из таблицы 8.1

Создаем новый файл lab8-4.asm для выполнения 1 задания(рис. 3.16).

```
ksyusha@fedora:~/work/arch-pc/lab08$ touch lab8-4.asm
```

Рис. 3.16: Создали файл lab8-4.asm

Открываем файл и пишем программу, которая находит сумму значений функции $f(x)$ для $x=x_1, x_2, \dots, x_n$ (функция $f(x)$ для 12 варианта)(рис. 3.17).

```
lab8-4.asm [----] 12 L:[ 1+30 31/ 31] *(441 / 441b) <EOF> [*] [X]
%include 'in_out.asm'
SECTION .data
msg1 db "Функция f(x)=15x-9",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
next:
    cmp ecx,0h
    jz _end
    pop eax
    call atoi
    ...
    mov ebx, 15
    mul ebx
    sub eax, 9
    add esi,eax
    loop next
_end:
    mov eax, msg2
    call sprintf
    mov eax, esi
    call iprintf
    call quit
```

Рис. 3.17: Заполнили файл lab8-4.asm в соответствии с условием задачи

Создаем исполняемый файл и запускаем его, убедившись в правильности вывода при $x_1=1, x_2=2, x_3=3$ (рис. 3.18).

```
ksyusha@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
ksyusha@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
ksyusha@fedora:~/work/arch-pc/lab08$ ./lab8-4 1 2 3
Функция f(x)=15x-9
Результат: 63
```

Рис. 3.18: Создали исполняемый файл и запустили его, проверив вывод

Вновь создаем исполняемый файл и запускаем его, убедившись в правильности вывода при $x_1=11, x_2=5, x_3=12$ (рис. 3.19).

```
ksyusha@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
ksyusha@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
ksyusha@fedora:~/work/arch-pc/lab08$ ./lab8-4 11 5 12
Функция  $f(x)=15x-9$ 
Результат: 393
```

Рис. 3.19: Вновь создали исполняемый файл и запустили его, проверив вывод с иными значениями x_1 , x_2 , x_3

4 Выводы

В ходе лабораторной работы мы приобрели навыки написания программ с использованием циклов и обработкой аргументов командной строки.