

# **Отчёт по лабораторной работе №8**

Кучерова Виктория Васильевна

# 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>Задание для самостоятельной работы</b>	<b>17</b>
<b>4</b>	<b>Выводы</b>	<b>19</b>
	<b>Список литературы</b>	<b>20</b>

# Список иллюстраций

2.1	Создание файла . . . . .	6
2.2	Программа . . . . .	7
2.3	Запуск . . . . .	8
2.4	Программа2 . . . . .	9
2.5	Запуск2 . . . . .	10
2.6	Программа3 . . . . .	11
2.7	Запуск3 . . . . .	12
2.8	Создание файла . . . . .	12
2.9	Программа4 . . . . .	13
2.10	Запуск4 . . . . .	13
2.11	Создание файла . . . . .	14
2.12	Программа5 . . . . .	14
2.13	Запуск5 . . . . .	15
2.14	Программа6 . . . . .	15
2.15	Запуск6 . . . . .	16
3.1	Программа7 . . . . .	18
3.2	Запуск7 . . . . .	18

## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы № 8, перейдем в него и создадим файл lab8-1.asm(рис. 2.1).

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

Рис. 2.1: Создание файла

Введем в файл lab8-1.asm текст программы. Создадим исполняемый файл и проверим его работу(рис. 2.2), (рис. 2.3).

```

%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. 2.2: Программа

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

Рис. 2.3: Запуск

Изменим текст программы, добавив изменение значение регистра `ecx` в цикле. Регистр `ecx` принимает нечетные значения, число проходов цикла не соответствует значению введенному с клавиатуры (рис. 2.4), (рис. 2.5).



```

%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. 2.4: Программа2

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

Рис. 2.5: Запуск2

Внесем изменения в текст программы, добавив команды `push` и `pop`. Число проходов цикла соответствует значению введенному с клавиатуры(рис. 2.6), (рис. 2.7).

```

#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. 2.6: Программа3

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

Рис. 2.7: Запуск3

Создадим файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и введем в него текст программы. Программой было обработано 4 аргумента(рис. 2.8), (рис. 2.9), (рис. 2.10).

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

Рис. 2.8: Создание файла

```

%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. 2.9: Программа4

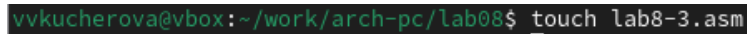
```

vvkucheroва@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
vvkucheroва@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
vvkucheroва@vbox:~/work/arch-pc/lab08$ ./lab8-2 аргумент1 аргумент 2 'аргумент 3
'
аргумент1
аргумент
2
аргумент 3

```

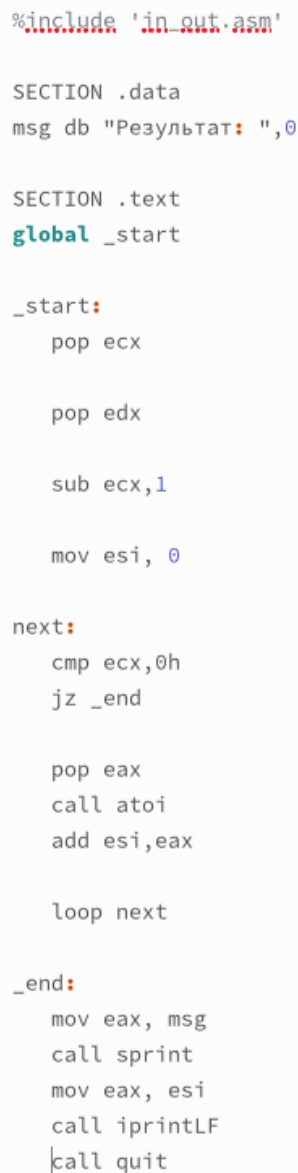
Рис. 2.10: Запуск4

Рассмотрим еще один пример программы которая выводит сумму чисел, которые передаются в программу как аргументы. Создадим файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 и введем в него текст программы(рис. 2.11), (рис. 2.12), (рис. 2.13).



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

Рис. 2.11: Создание файла



```
%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. 2.12: Программа5

```

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

```

Рис. 2.13: Запуск5

Измените текст программы для вычисления произведения аргументов командной строки(рис. 2.14), (рис. 2.15).

```

#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. 2.14: Программа6

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

Рис. 2.15: Запуск6



### **3 Задание для самостоятельной работы**

Напишем программу, которая находит сумму значений функции  $f(x)$  (рис. 3.1), (рис. 3.2).

```

%include 'in_out.asm'

SECTION .data
msg1 db "Функция: f(x)=8x - 3",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,8
    mul ebx
    sub eax,3
    add esi,eax

    loop next

_end:

```

Рис. 3.1: Программа7

```

vvkucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab0
8$ nasm -f elf lab8-3.asm
vvkucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab0
8$ ld -m elf_i386 -o lab8-3 lab8-3.o
vvkucheroва@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab0
8$ ./lab8-3 12 13 7 10 5
Функция: f(x)=8x - 3
Результат: 361

```

Рис. 3.2: Запуск7

## 4 Выводы

Я приобрела навыки написания программ с использованием циклов и обработкой аргументов командной строки.

## **Список литературы**