

# **Отчет по лабораторной работе №8**

**Архитектура компьютеров и операционные системы**

Никита Сергеевич Кокшаров

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
2.1	Реализация циклов в NASM . . . . .	5
2.2	Обработка аргументов командной строки . . . . .	8
2.3	Задание для самостоятельной работы . . . . .	12
<b>3</b>	<b>Выводы</b>	<b>13</b>

## Список иллюстраций

2.1	Создание lab8-1.asm . . . . .	5
2.2	Код в lab8-1.asm . . . . .	6
2.3	Запуск lab8-1 . . . . .	6
2.4	Изменение кода в lab8-1.asm . . . . .	7
2.5	Запуск измененного lab8-1 . . . . .	7
2.6	Изменение кода в lab8-1.asm с использованием стека . . . . .	8
2.7	Запуск lab8-1, измененного с использованием стека . . . . .	8
2.8	Код в lab8-2.asm . . . . .	9
2.9	Запуск lab7-2 . . . . .	9
2.10	Код в lab8-3.asm . . . . .	10
2.11	Запуск lab8-3 . . . . .	10
2.12	Изменение кода в lab8-3.asm . . . . .	11
2.13	Запуск измененного lab8-1 . . . . .	11
2.14	Код в lab8-4.asm . . . . .	12
2.15	Запуск lab8-4.asm . . . . .	12

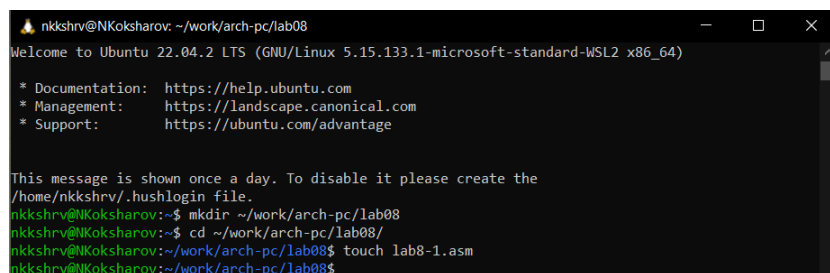
# 1 Цель работы

Целью работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

## 2 Выполнение лабораторной работы

### 2.1 Реализация циклов в NASM

Создаю файл lab8-1.asm в новой директории lab07 (рис. 2.1).



```
nkkshrv@NKKoksharov: ~/work/arch-pc/lab08
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.133.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/nkkshrv/.hushlogin file.
nkkshrv@NKKoksharov:~$ mkdir ~/work/arch-pc/lab08
nkkshrv@NKKoksharov:~$ cd ~/work/arch-pc/lab08/
nkkshrv@NKKoksharov:~/work/arch-pc/lab08$ touch lab8-1.asm
nkkshrv@NKKoksharov:~/work/arch-pc/lab08$
```

Рис. 2.1: Создание lab8-1.asm

Пишу код программы из листинга 8.1 (рис. 2.2)

```

mc [nkkshrv@Nkoksharov:~/work/arch-pc/lab08
/home/nkkshrv/work~/lab08/lab8-1.asm [-M--] 13 L: [ 1+30 31/ 31] *(379 / 379b) <EOF> [*][X] ^
#include "in_out.asm"

SECTION .data
    msg1 db "Введите N: ", 0h

SECTION .bss
    N: resb 10

SECTION .text
    global _start
_start:

    mov ecx, msg1
    call sprint

    mov ecx, N
    mov edx, 10
    call sread

    mov ecx, N
    call atoi
    mov [N], ecx

    mov ecx, [N]
label:
    mov [N], ecx
    mov ecx, [N]
    call iprintLF
    loop label

    call quit

```

Рис. 2.2: Код в lab8-1.asm

Создаю исполняемый файл и запускаю его (рис. 2.3)

```

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

```

Рис. 2.3: Запуск lab8-1

Изменяю код программы (рис. 2.4)

```
nkshrv@NKoksharov: ~/work/arch-pc/lab08
/home/nkshrv/work~/lab08/lab8-1.asm [-M--] 14 L: [ 1+25 26/ 32] *(312 / 394b) 0010 0x00A [*][X] ^
%include 'in_out.asm'

SECTION .data
    msg1 db 'Введите N: ', 0h

SECTION .bss
    N: resb 10

SECTION .text
    global _start
    _start:

    mov ecx, msg1
    call sprint

    mov ecx, N
    mov edx, 10
    call sread

    mov ecx, N
    call atoi
    mov [N], ecx

    mov ecx, [N]
label:
    sub ecx, 1
    mov [N], ecx
    mov ecx, [N]
    call iprintf
    loop label

    call quit
```

Рис. 2.4: Изменение кода в lab8-1.asm

Создаю исполняемый файл и запускаю его (рис. 2.5)

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

Рис. 2.5: Запуск измененного lab8-1

Разница вывода наглядна на изображении: поскольку из регистра `ecx` с каждым проходом цикла вычитается единица (`sub ecx, 1`) при вводе четного числа (например, 10) будут выведены все меньшие нечетные числа, при вводе нечетного числа - все меньшие четные числа. Очевидно, число проходов цикла не соответствует значению `N`, введенному с клавиатуры.

Изменяю код программы, используя стек (рис. 2.6)

```

nkkshrv@NKKoksharov: ~/work/arch-pc/lab08
/home/nkkshrv/work~/lab08/lab8-1.asm [-M--] 13 L: [ 1+34 35/ 35] *(420 / 420b) <EOF> [*][X] ^
#include "in_out.asm"

SECTION .data
    msg1 db "Введите N: ", 0h

SECTION .bss
    N: resb 10

SECTION .text
    global _start
    _start:

    mov ecx, msg1
    call sprint

    mov ecx, N
    mov edx, 10
    call sread

    mov ecx, 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: Изменение кода в lab8-1.asm с использованием стека

Создаю исполняемый файл и запускаю его (рис. 2.7)

```

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

```

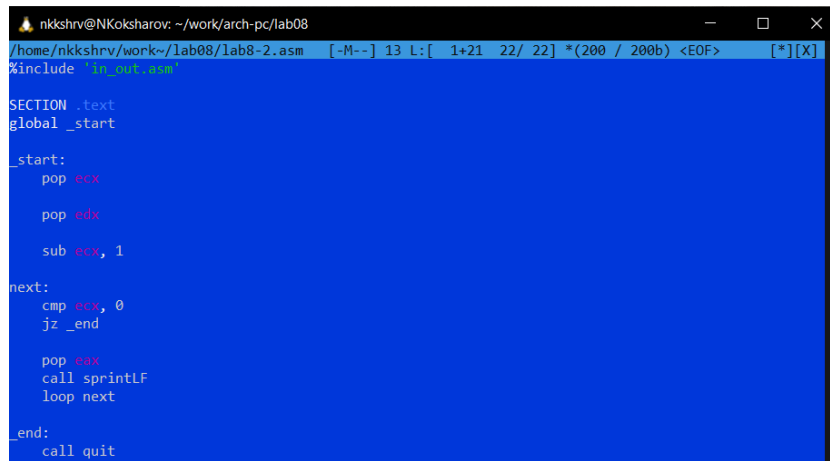
Рис. 2.7: Запуск lab8-1, измененного с использованием стека

В данном случае число проходов цикла соответствует значению N, введенному с клавиатуры.

## 2.2 Обработка аргументов командной строки

Создаю lab8-2.asm, пишу туда код из листинга 8.2 (рис. 2.8)





```

nkkshrv@NKKoksharov: ~/work/arch-pc/lab08
/home/nkkshrv/work~/lab08/lab8-2.asm [-M--] 13 L: [ 1+21 22/ 22] *(200 / 200b) <EOF> [*][X]
#include "in_out.asm"

SECTION .text
global _start

_start:
    pop ecx

    pop edx

    sub ecx, 1

next:
    cmp ecx, 0
    jz _end

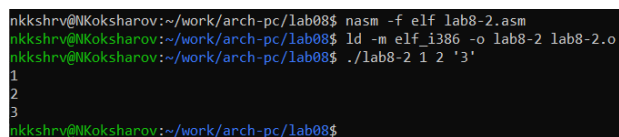
    pop eax
    call sprintf
    loop next

_end:
    call quit

```

Рис. 2.8: Код в lab8-2.asm

Создаю исполняемый файл и запускаю его, проверяю для 3 аргументов (рис. 2.9)



```

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

```

Рис. 2.9: Запуск lab7-2

Все 3 аргумента были обработаны.

Создаю lab8-3.asm, пишу туда код из листинга 8.3 (рис. 2.10)

```

nkkshrv@NKKoksharov: ~/work/arch-pc/lab08
/home/nkkshrv/work~/lab08/lab8-3.asm [-M--] 13 L: [ 1+29 30/ 30] *(344 / 344b) <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, 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. 2.10: Код в lab8-3.asm

Создаю исполняемый файл и запускаю его, указав аргументы (рис. 2.11)

```

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

```

Рис. 2.11: Запуск lab8-3

Меняю текст программы для вычисления произведения аргументов командной строки (рис. 2.12)

```
nkshrv@NKKoksharov: ~/work/arch-pc/lab08
/home/nkshrv/work~/lab08/lab8-3.asm [-M--] 13 L: [ 1+29 30/ 30] *(339 / 339b) <EOF> [*][X] ^
#include "in_out.asm"

SECTION .data
msg db "Pezymnar: ", 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

    loop next

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

Рис. 2.12: Изменение кода в lab8-3.asm

Создаю исполняемый файл и запускаю его, указав аргументы (рис. 2.13)

```
nkshrv@NKKoksharov: ~/work/arch-pc/lab08
/home/nkshrv/work~/lab08/lab8-3.asm [-M--] 16 L: [ 1+22 23/ 32] *(253 / 357b) 0010 0x00A [*][X] ^
#include "in_out.asm"

SECTION .data
msg db "Pezymnar: ", 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 ecx, esi
    call iprintf
    call quit
```

Рис. 2.13: Запуск измененного lab8-1

Результат верный.

## 2.3 Задание для самостоятельной работы

Мой вариант: №18.

Пишу код программы для вычисления суммы значений функции  $f(x) = 17 + 5x$  для  $x = x_1, x_2, x_3, \dots, x_n$  (рис. 2.14)

```
nkshrv@NKoksharov:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
nkshrv@NKoksharov:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
nkshrv@NKoksharov:~/work/arch-pc/lab08$ ./lab8-3 1 2 3 4 5
Результат: 120
nkshrv@NKoksharov:~/work/arch-pc/lab08$
```

Рис. 2.14: Код в lab8-4.asm

Результат верный.

Создаю исполняемый файл и запускаю его, указав аргументы (рис. 2.15)

```
nkshrv@NKoksharov: ~/work/arch-pc/lab08
/home/nkshrv/work/~lab08/lab8-4.asm [----] 12 L: [ 1+22 23/ 35] *(252 / 390b) 0010 0x00A [*][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, 0

next:
    cmp ecx, 0h
    jz _end

    pop ecx
    call atoi
    mov ebx, 5
    imul ebx, ecx
    add esi, ebx
    add esi, 17

    loop next

_end:
    mov ecx, msg
    call sprint
    mov eax, esi
    call iprintLF
    call quit
```

Рис. 2.15: Запуск lab8-4.asm

Результаты для разных наборов аргументов верны.

## **3 Выводы**

При выполнении лабораторной работы я приобрел навыки написания программ с использованием циклов и обработкой аргументов командной строки.