

# **Отчет по лабораторной работе № 8**

**Дисциплина: архитектура компьютера**

Выслоух Алиса Александровна

# 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>
3.1	Реализация циклов в NASM . . . . .	7
3.2	Самостоятельная работа . . . . .	13
<b>4</b>	<b>Выводы</b>	<b>14</b>

## Список иллюстраций

3.1	Создание папки и файла. . . . .	7
3.2	Ввод текста. . . . .	8
3.3	Создание исполняемого файла. . . . .	8
3.4	Изменение файла при четных значениях. . . . .	9
3.5	Изменение файла при нечетных значениях. . . . .	9
3.6	Создание исполняемого файла после изменения. . . . .	9
3.7	Создание исполняемого файла после изменения. . . . .	10
3.8	Создание исполняемого файла после изменения. . . . .	10
3.9	Ввод текста программы. . . . .	11
3.10	Запуск программы. . . . .	11
3.11	Ввод измененной программы. . . . .	12
3.12	Запуск изменной программы. . . . .	12

## **Список таблиц**

# 1 Цель работы

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

## **2 Задание**

1. Реализация циклов в NASM
2. Обработка аргументов командной строки
3. Выполнение заданий для самостоятельной работы

## 3 Выполнение лабораторной работы

### 3.1 Реализация циклов в NASM

Создаю папку lab6 и перехожу в нее и создаю и создаю файл lab8-1.asm (рис. 3.1).

A terminal window titled 'aavihsloukh@dk8n60 - lab08' showing three commands being executed: 'mkdir ~/work/arch-pc/lab08', 'cd ~/work/arch-pc/lab08', and 'touch lab8-1.asm'. The prompt changes from '~' to '~/work/arch-pc/lab08' after the 'cd' command.

```
aavihsloukh@dk8n60 ~ $ mkdir ~/work/arch-pc/lab08
aavihsloukh@dk8n60 ~ $ cd ~/work/arch-pc/lab08
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ touch lab8-1.asm
```

Рис. 3.1: Создание папки и файла.

В файл я ввела текст первой программы. (рис. 3.2).

```

lab8-1.asm  [-M--] 24 L: [ 1+ 0 1/ 31] *(24 / 843b) 0045 0x02D [*][X]
;-----
; Программа вывода значений регистра 'ecx'
;-----
%include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
; ----- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint
; ----- Ввод 'N'
mov ecx, N
mov edx, 10
call sread
; ----- Преобразование 'N' из символа в число
mov eax,N
call atoi
mov [N],eax
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход

```

Рис. 3.2: Ввод текста.

Создала исполняемый файл. (рис. 3.3).

```

aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 1
1
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 5
5
4
3
2
1

```

Рис. 3.3: Создание исполняемого файла.

Я изменила текст программы, в теле цикла label добавила строку `sub eax,1`. Цикл закольцевался и стал бесконечным при вводе нечетных цифр и при вводе четных чисел он выводил нечетные на уменьшение (рис. 3.4).(рис. 3.5).

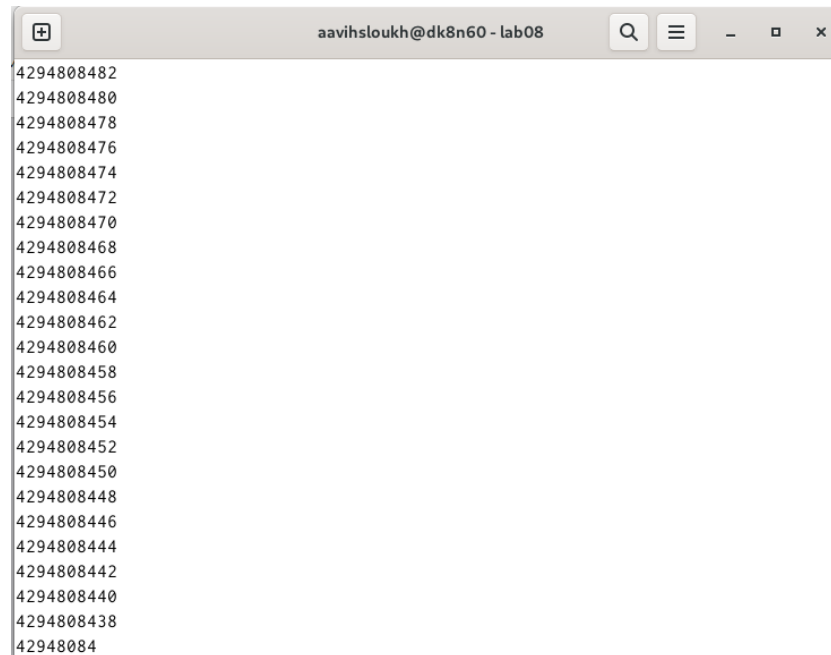


```

aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 6
5
3
1

```

Рис. 3.4: Изменение файла при четных значениях.



```

4294808482
4294808480
4294808478
4294808476
4294808474
4294808472
4294808470
4294808468
4294808466
4294808464
4294808462
4294808460
4294808458
4294808456
4294808454
4294808452
4294808450
4294808448
4294808446
4294808444
4294808442
4294808440
4294808438
42948084

```

Рис. 3.5: Изменение файла при нечетных значениях.

Я изменила текст программы так, чтобы цикл и счетчик работал правильно. По итогу после изменения программы, яисло проходки циклов стал соответствовать числу введенному с клавиатуры. (рис. 3.6).

```

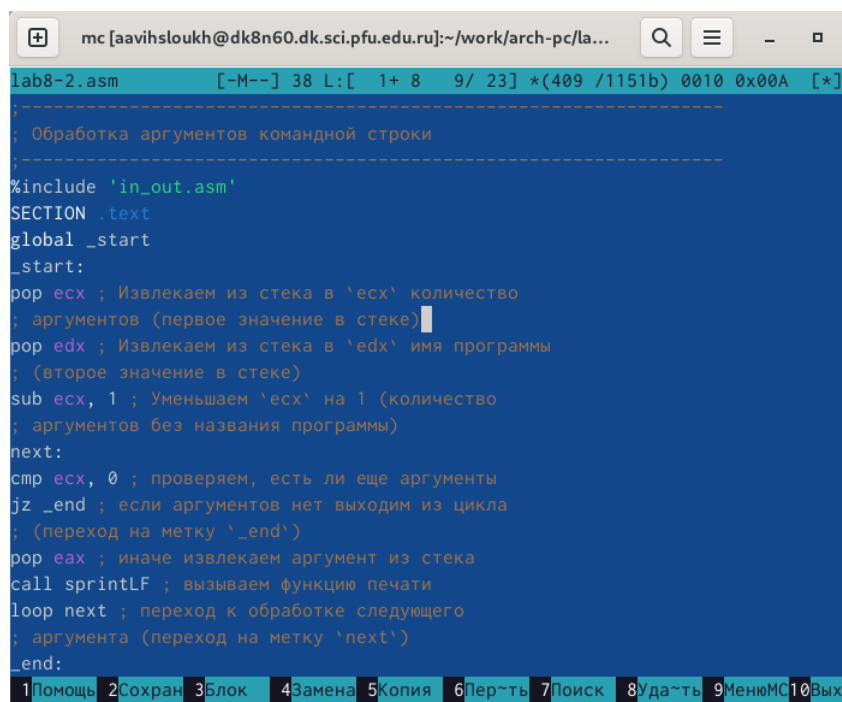
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 4
3
2
1
0

```

Рис. 3.6: Создание исполняемого файла после изменения.

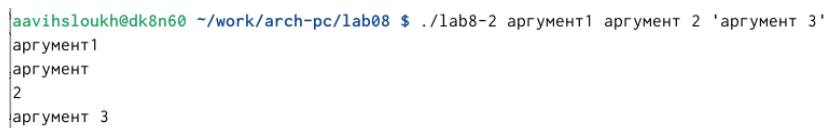
Я создал файл lab8-2.asm и ввел туда программу, которая выводит все аргу-

мент, которые ввели. Программа выводит все 3 аргумента которые ввели, но в разной вариации (рис. 3.7) (рис. 3.7).



```
lab8-2.asm [-M--] 38 L: [ 1+ 8 9/ 23] *(409 /1151b) 0010 0x00A [*]
;-----
; Обработка аргументов командной строки
;-----
#include 'in_out.asm'
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в 'ecx' количество
; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в 'edx' имя программы
; (второе значение в стеке)
sub ecx, 1 ; Уменьшаем 'ecx' на 1 (количество
; аргументов без названия программы)
next:
cmp ecx, 0 ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку '_end')
pop eax ; иначе извлекаем аргумент из стека
call sprintf ; вызываем функцию печати
loop next ; переход к обработке следующего
; аргумента (переход на метку 'next')
_end:
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9Меню 10Выход
```

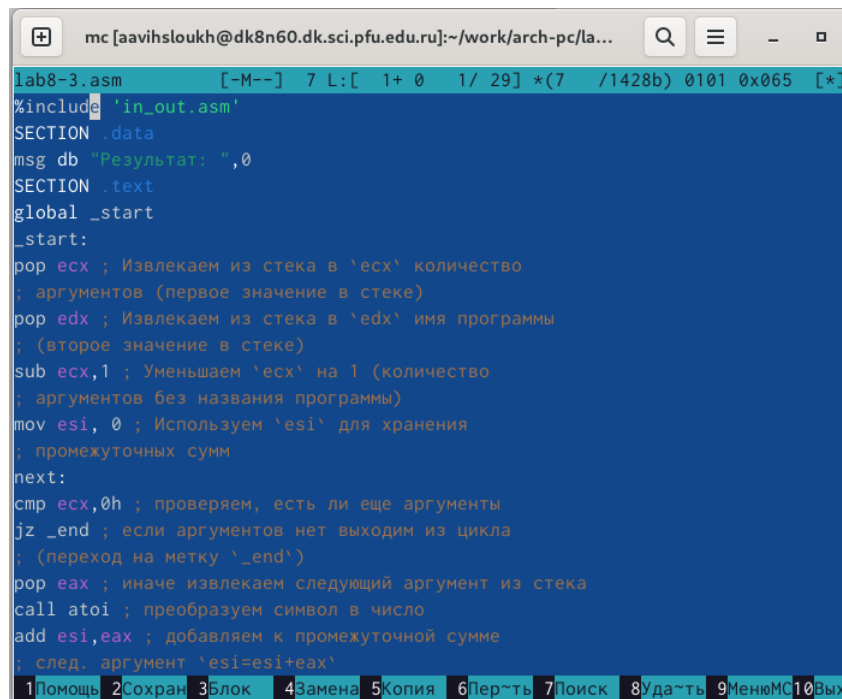
Рис. 3.7: Создание исполняемого файла после изменения.



```
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
```

Рис. 3.8: Создание исполняемого файла после изменения.

Я создала файл lab8-3.asm. Ввел текст программы и запустил ее. Программа вывела сумму чисел, которые я ввела. (рис. 3.9) (рис. 3.10).



```
lab8-3.asm [-M--] 7 L: [ 1+ 0 1/ 29] *(7 /1428b) 0101 0x065 [*]
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в 'ecx' количество
; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в 'edx' имя программы
; (второе значение в стеке)
sub ecx,1 ; Уменьшаем 'ecx' на 1 (количество
; аргументов без названия программы)
mov esi, 0 ; Используем 'esi' для хранения
; промежуточных сумм
next:
cmp ecx,0h ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку '_end')
pop eax ; иначе извлекаем следующий аргумент из стека
call atoi ; преобразуем символ в число
add esi,eax ; добавляем к промежуточной сумме
; след. аргумент 'esi=esi+eax'
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выз
```

Рис. 3.9: Ввод текста программы.

```
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ./lab8-3 12 13 17 10 5
Результат: 57
```

Рис. 3.10: Запуск программы.

Я изменила программу, чтобы она выводила произведение введенных чисел.  
(рис. 3.11).

```
mc [aavihsloukh@dk8n60.dk.sci.pfu.e
lab8-3.asm [----] 7 L:[ 2+20 22/ 32] *(652 /1
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx

pop edx ; Извлекаем из стека в 'edx' имя программы

sub ecx,1 ; Уменьшаем 'ecx' на 1 (количество
; аргументов без названия программы)
mov esi, 1 ; Используем 'esi' для хранения
mov eax, 1
; проверяем, есть ли еще аргументы
next:
cmp ecx, 0
jz _end

pop eax ; иначе извлекаем следующий аргумент из стека
call atoi ; преобразуем символ в число
mov ebx, eax
mov eax, esi
mul ebx
mov esi, eax
loop next ; переход к обработке следующего аргумента
_end:
mov eax, msg ; вывод сообщения "Результат: "
call sprint
mov eax, esi ; записываем сумму в регистр 'eax'
call iprintLF ; печать результата
call quit ; завершение программы
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер
```

Рис. 3.11: Ввод измененной программы.

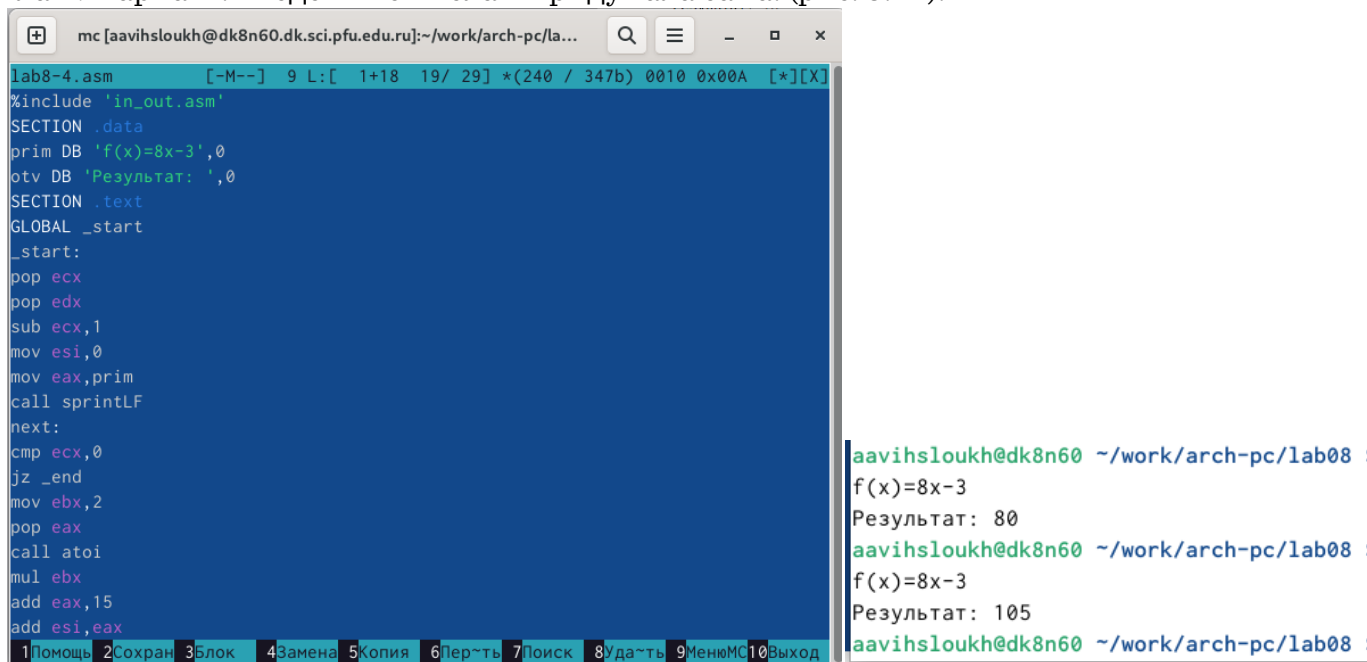
Запуск измененной программы. (рис. 3.12).

```
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ./lab8-3 1 2 3
Результат: 6
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 $ ./lab8-3 12 13 17
Результат: 2652
```

Рис. 3.12: Запуск измененной программы.

## 3.2 Самостоятельная работа

Я написал программу, которая выводит сумму всех решений примера. Я получила 19 вариантов. Введенные числа я придумала сама. (рис. 3.11).



The screenshot shows a terminal window with the following content:

```
lab8-4.asm [-M--] 9 L:[ 1+18 19/ 29] *(240 / 347b) 0010 0x00A [*][X]
#include 'in_out.asm'
SECTION .data
prim DB 'f(x)=8x-3',0
otv DB 'Результат: ',0
SECTION .text
GLOBAL _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi,0
mov eax,prim
call sprintfLF
next:
cmp ecx,0
jz _end
mov ebx,2
pop eax
call atoi
mul ebx
add eax,15
add esi,eax
1 Помощь 2 Сохран 3 Блок 4 Замена 5 Копия 6 Пер-ть 7 Поиск 8 Уда-ть 9 МенюМС 10 Выход
```

On the right side of the terminal, the output of the program is shown:

```
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 :
f(x)=8x-3
Результат: 80
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 :
f(x)=8x-3
Результат: 105
aavihsloukh@dk8n60 ~/work/arch-pc/lab08 :
```

## **4 Выводы**

Здесь кратко описываются итоги проделанной работы.