

# **Отчет по лабораторной работе №8**

**Дисциплина: архитектура компьютера**

Рахматова Жылдыз Талантбековна

# 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>Самостотельная работа</b>	<b>14</b>
	<b>Вывод</b>	<b>16</b>

# Список иллюстраций

2.1	Создание файла и каталога . . . . .	6
2.2	Текст программы . . . . .	7
2.3	Запуск программы и проверка результата . . . . .	7
2.4	Измененный текст программы . . . . .	8
2.5	Запуск программы . . . . .	9
2.6	Редактирование текста программы . . . . .	10
2.7	Запуск измененной программы . . . . .	10
2.8	Текст программы для вывода аргументов . . . . .	11
2.9	Результаты работы программы . . . . .	11
2.10	Текст программы lab9-3 . . . . .	12
2.11	Результат работы программы . . . . .	12
2.12	Текст программы с произведением чисел . . . . .	13
2.13	Результаты работы программы с произведением . . . . .	13
3.1	Текст программы в самостоятельной работе . . . . .	15
3.2	Результаты работы программы . . . . .	15

## **Список таблиц**

# 1 Цель работы

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

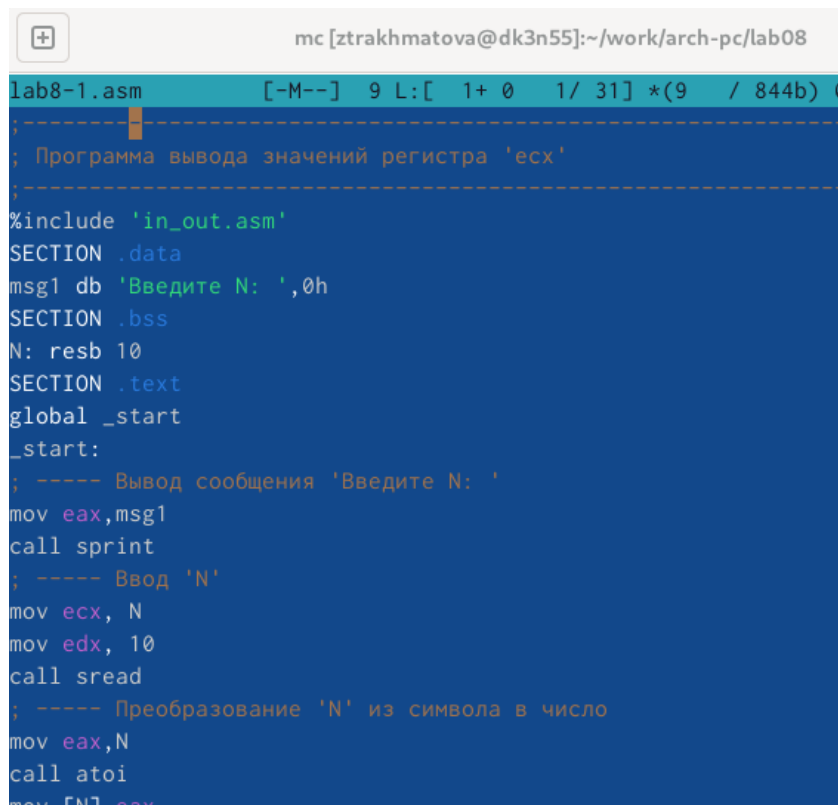
## 2 Выполнение лабораторной работы

1) Я создала каталог lab08 и файл lab8-1.asm

```
ztrakhmatova@dk3n55 ~ $ cd work/arch-pc
ztrakhmatova@dk3n55 ~/work/arch-pc $ mkdir lab08
ztrakhmatova@dk3n55 ~/work/arch-pc $ ls
lab04 lab05 lab06 lab07 lab08
ztrakhmatova@dk3n55 ~/work/arch-pc $ cd lab08
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ touch lab8-1.asm
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ pwd
/afs/.dk.sci.pfu.edu.ru/home/z/t/ztrakhmatova/work/arch-pc/lab08
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ ls
lab8-1.asm
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ █
```

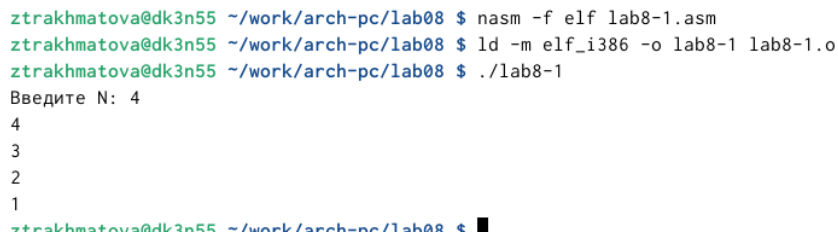
Рис. 2.1: Создание файла и каталога

2) В файл я ввела текст первой программы и создала исполняемый файл.



```
mc [ztrakhmatova@dk3n55]:~/work/arch-pc/lab08
lab8-1.asm [-M--] 9 L:[ 1+ 0 1/ 31] *(9 / 844b) 0
; -----
; 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
```

Рис. 2.2: Текст программы



```
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 4
4
3
2
1
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $
```

Рис. 2.3: Запуск программы и проверка результата

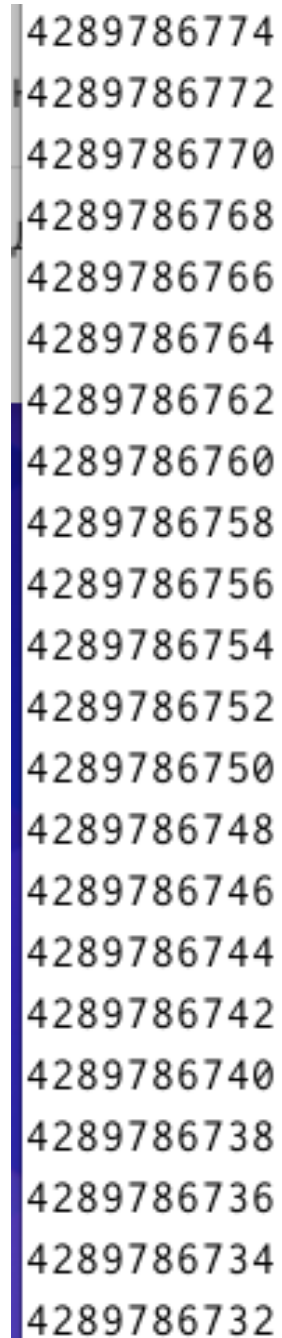
3)Я изменила текст программы, в теле цикла label добавила строку `sub eax,1`. Циклы закольцевался и стал бесконечным.

```
mov ecx,[N] ; Счетчик цикла, 'ecx=N'
label:
sub eax,1
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения 'N'
loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
; переход на 'label'
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6

Рис. 2.4: Измененный текст программы





4289786774  
4289786772  
4289786770  
4289786768  
4289786766  
4289786764  
4289786762  
4289786760  
4289786758  
4289786756  
4289786754  
4289786752  
4289786750  
4289786748  
4289786746  
4289786744  
4289786742  
4289786740  
4289786738  
4289786736  
4289786734  
4289786732

Рис. 2.5: Запуск программы

- 4) 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.

```
label:
push ecx ; добавление значения ecx в стек
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
pop ecx ; извлечение значения ecx из стека
loop label
call quit
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Печать

Рис. 2.6: Редактирование текста программы

```
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-1 5
Введите N: 5
4
3
2
1
0
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $
```

Рис. 2.7: Запуск измененной программы

5)Я создала файл lab8-2.asm и ввела туда программу, которая выводит все аргументы,которые ввели. Программа выводит все 3 аргумента которые ввели, но в разной вариации.

```

lab8-2.asm      [-M--]  0 L: [ 1+ 1  2/ 24] *(67 /1152b
;
; -----
;  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 printf ; вызываем функцию печати

```

Рис. 2.8: Текст программы для вывода аргументов

```

ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-2.asm
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
ztrakhmatova@dk3n55 ~/work/arch-pc/lab08 $ █

```

Рис. 2.9: Результаты работы программы

- 6) Я создала файл lab8-3.asm. Ввела текст программы и запустила ее. Программа вывела сумму чисел, которые я ввела.

```

lab8-3.asm      [-M--] 21 L:[ 1+ 0  1/ 29] *(21 /1428
%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

```

Рис. 2.10: Текст программы lab9-3

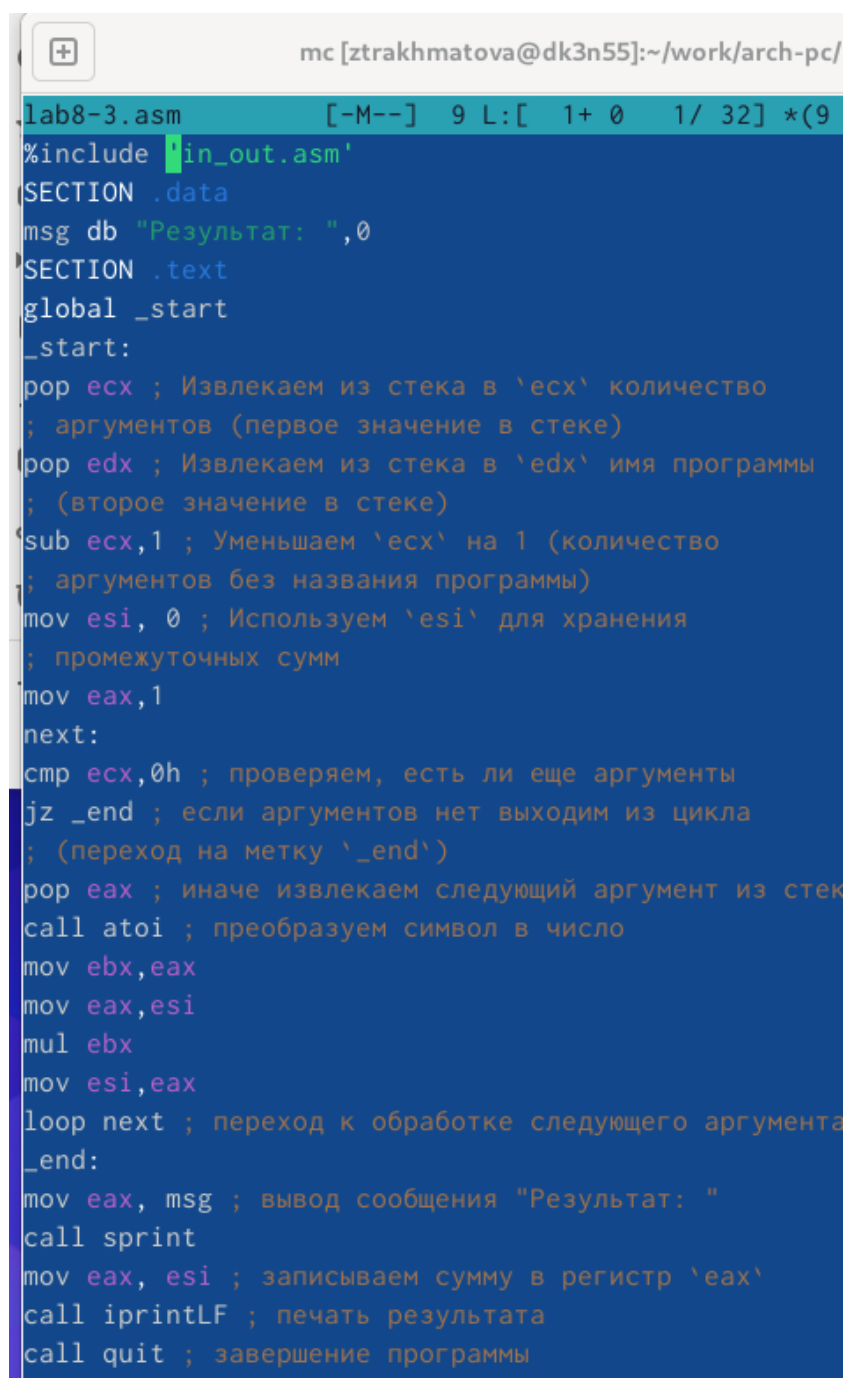
```

matova@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
matova@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
matova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-3 1 2 3 4
.тат: 10
matova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-3 10 20 30
.тат: 60
matova@dk3n55 ~/work/arch-pc/lab08 $

```

Рис. 2.11: Результат работы программы

- 7) Я изменила программу, чтобы она выводила произведение введенных чисел.



```
mc [ztrakhmatova@dk3n55]:~/work/arch-pc/
lab8-3.asm [-M--] 9 L:[ 1+ 0 1/ 32] *(9
#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
mov eax,1
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
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
```

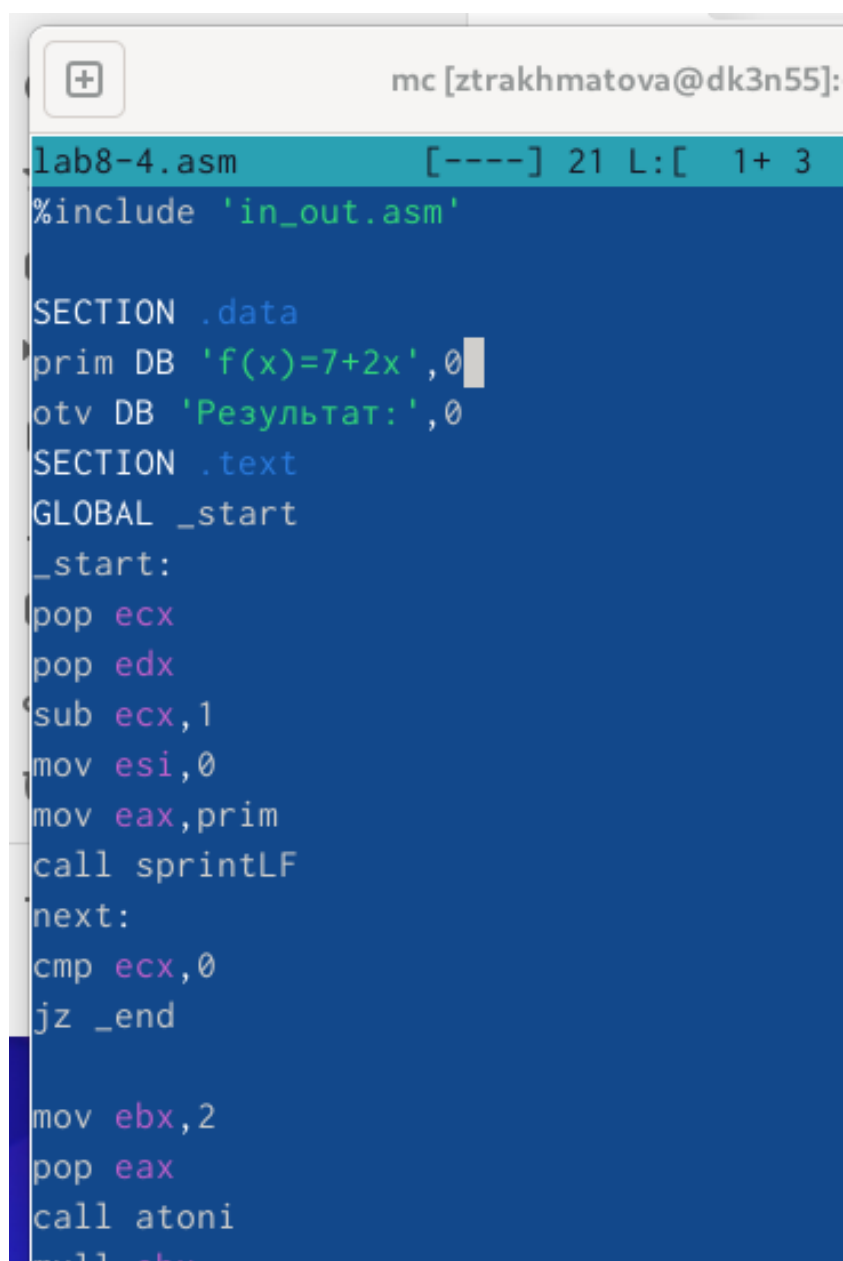
Рис. 2.12: Текст программы с произведением чисел

Результаты работы программы с произведением

Рис. 2.13: Результаты работы программы с произведением

### **3 Самостоятельная работа**

Я написала программу, которая выводит сумму всех решений примера. В лабораторной работе №6, я получила 8 вариантов, поэтому я писала программу для восьмого варианта. Введенные числа я придумала сама, и посчитала их, чтобы проверить работу программы.



```
lab8-4.asm [----] 21 L:[ 1+ 3
#include 'in_out.asm'

SECTION .data
prim DB 'f(x)=7+2x',0
otv DB 'Результат:',0
SECTION .text
GLOBAL _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi,0
mov eax,prim
call sprintLF
next:
cmp ecx,0
jz _end

mov ebx,2
pop eax
call atoni
push ebx
```

Рис. 3.1: Текст программы в самостоятельной работе

Результаты работы программы

Рис. 3.2: Результаты работы программы

## **Вывод**

Я приобрела навыки написания программы с использованием цикла.