

Отчёт по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки.

Майоров Дмитрий Андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Задание для самостоятельной работы	15
5	Выводы	18
	Список литературы	19

Список иллюстраций

3.1	Создаем каталог для лабораторной работы №8 и файл в нем	7
3.2	Открываем файл и заполняем его в соответствии с листингом . . .	8
3.3	Создаем исполняемый файл и запускаем его	8
3.4	Открываем файл и изменяем его. Добавляем изменение значения регистра в цикле	9
3.5	Создаем исполняемый файл и запускаем его	9
3.6	Редактируем файл	10
3.7	Создаем исполняемый файл и запускаем его	11
3.8	Создаем новый файл	11
3.9	Открываем файл и заполняем его в соответствии с листингом . . .	12
3.10	Создаем исполняемый файл и запускаем его	12
3.11	Создаем новый файл	12
3.12	Открываем файл и заполняем его в соответствии с листингом . . .	13
3.13	Создаем исполняемый файл и запускаем его	13
3.14	Редактируем файл, чтобы вычислялось произведение вводимых значений	14
3.15	Создаем исполняемый файл и запускаем его	14
4.1	Создаем новый файл	15
4.2	Пишем программу	16
4.3	Создаем исполняемый файл и запускаем его. Смотрим на работу программы при $x_1=5$, $x_2=3$, $x_1=4$ (все правильно)	16
4.4	Создаем исполняемый файл и запускаем его. Смотрим на работу программы при $x_1=1$, $x_2=3$, $x_1=7$ (все правильно)	17

Список таблиц

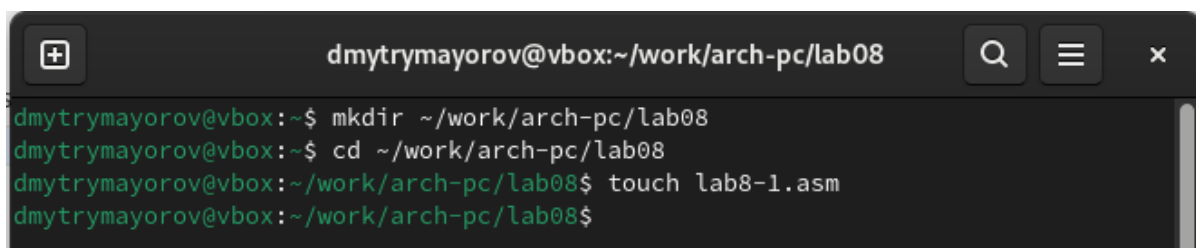
1 Цель работы

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

2 Задание

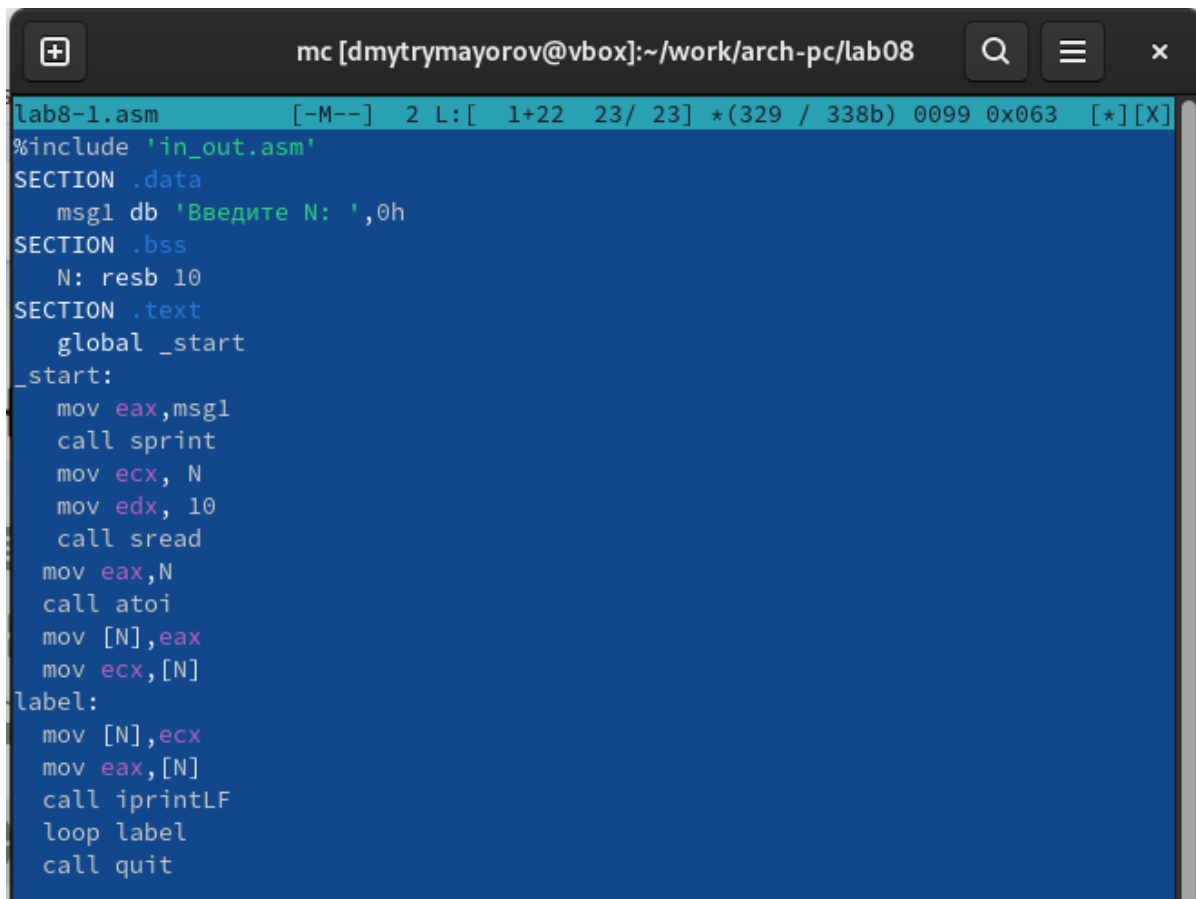
Написать программы с использованием циклов и обработкой аргументов командной строки.

3 Выполнение лабораторной работы



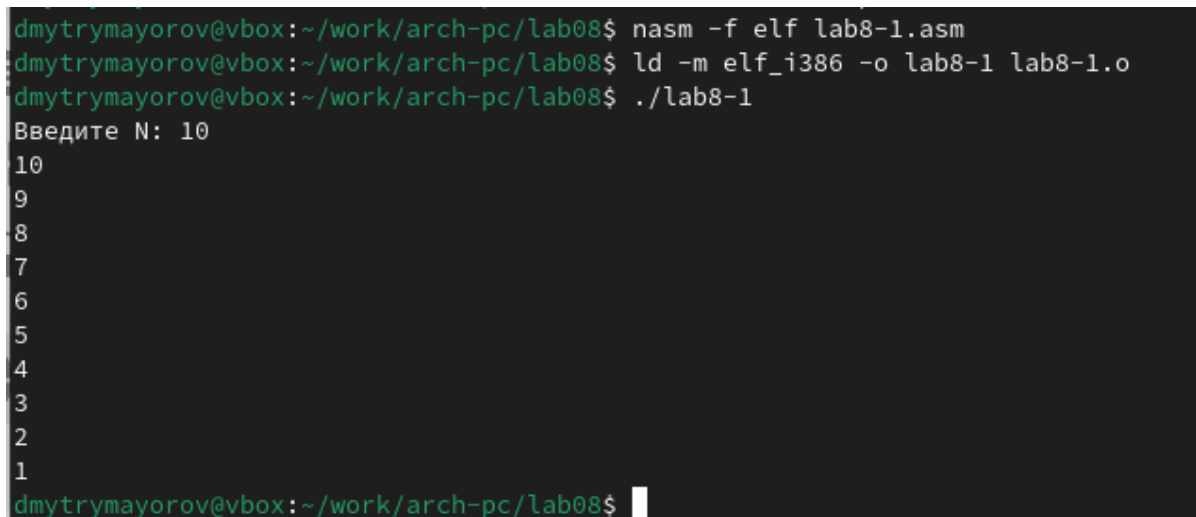
```
dmytrymayorov@vbox:~/work/arch-pc/lab08
dmytrymayorov@vbox:~$ mkdir ~/work/arch-pc/lab08
dmytrymayorov@vbox:~$ cd ~/work/arch-pc/lab08
dmytrymayorov@vbox:~/work/arch-pc/lab08$ touch lab8-1.asm
dmytrymayorov@vbox:~/work/arch-pc/lab08$
```

Рис. 3.1: Создаем каталог для лабораторной работы №8 и файл в нем



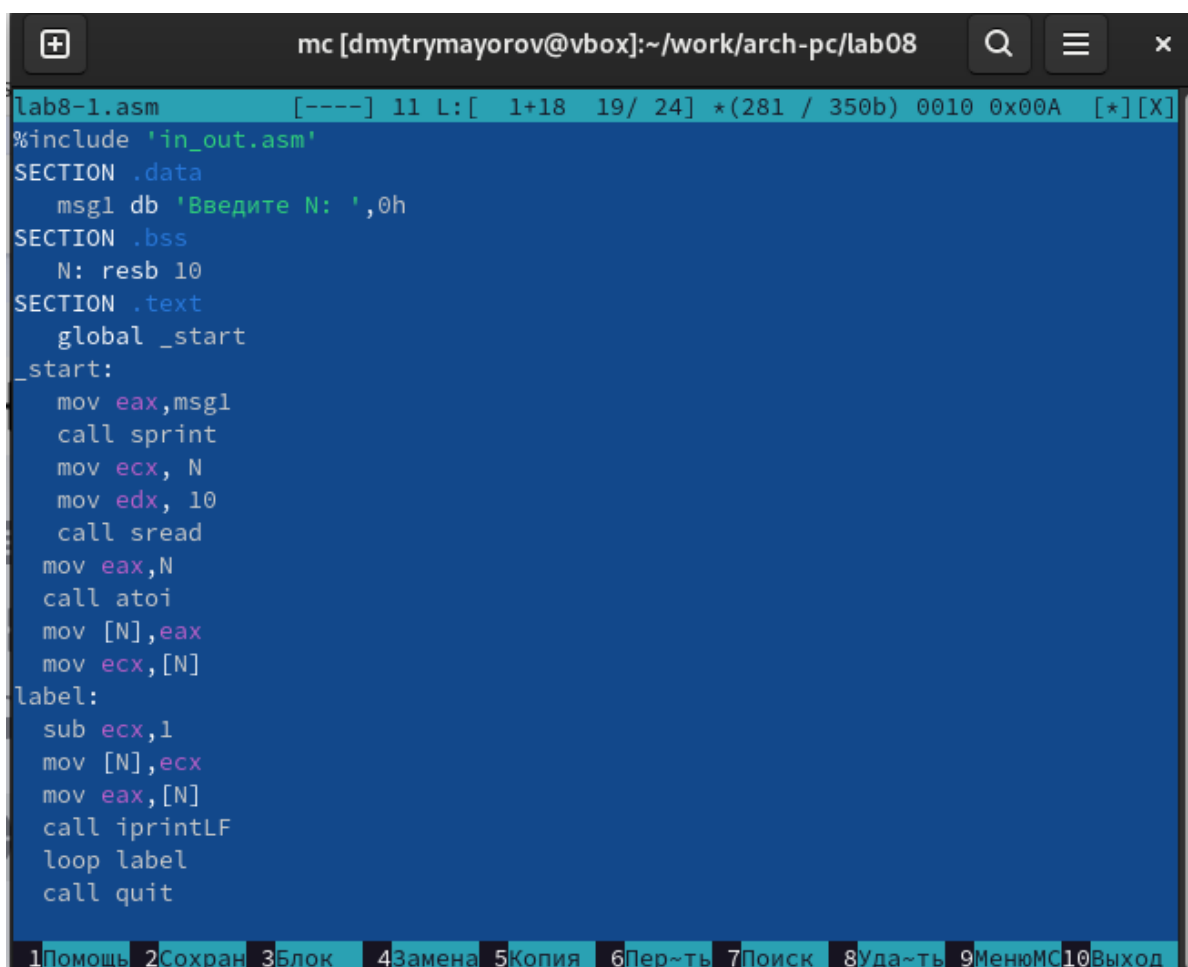
```
lab8-1.asm [-M--] 2 L: [ 1+22 23/ 23] *(329 / 338b) 0099 0x063 [*] [X]
#include 'in_out.asm'
SECTION .data
    msg1 db 'Введите N: ',0h
SECTION .bss
    N: resb 10
SECTION .text
    global _start
_start:
    mov eax,msg1
    call sprint
    mov ecx, N
    mov edx, 10
    call sread
    mov eax,N
    call atoi
    mov [N],eax
    mov ecx,[N]
label:
    mov [N],ecx
    mov eax,[N]
    call iprintLF
    loop label
    call quit
```

Рис. 3.2: Открываем файл и заполняем его в соответствии с листингом



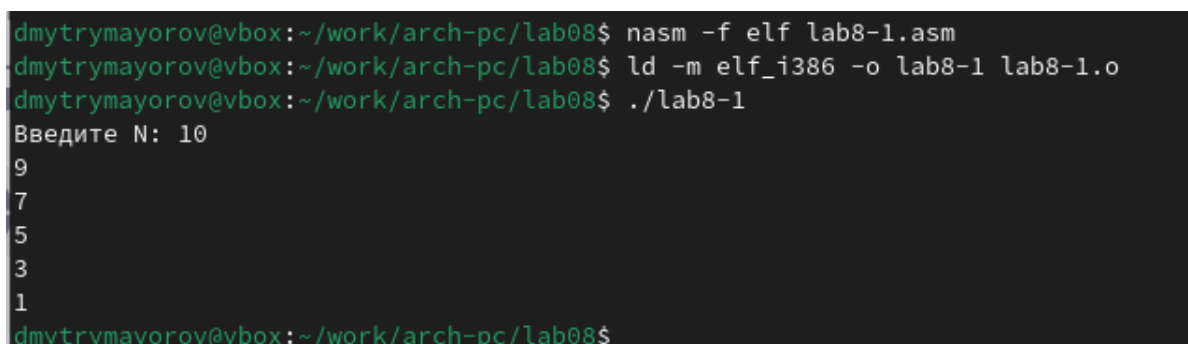
```
dmytrymayorov@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
dmytrymayorov@vbox:~/work/arch-pc/lab08$
```

Рис. 3.3: Создаем исполняемый файл и запускаем его



```
lab8-1.asm [----] 11 L: [ 1+18 19/ 24] *(281 / 350b) 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 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: Открываем файл и изменяем его. Добавляем изменение значения регистра в цикле

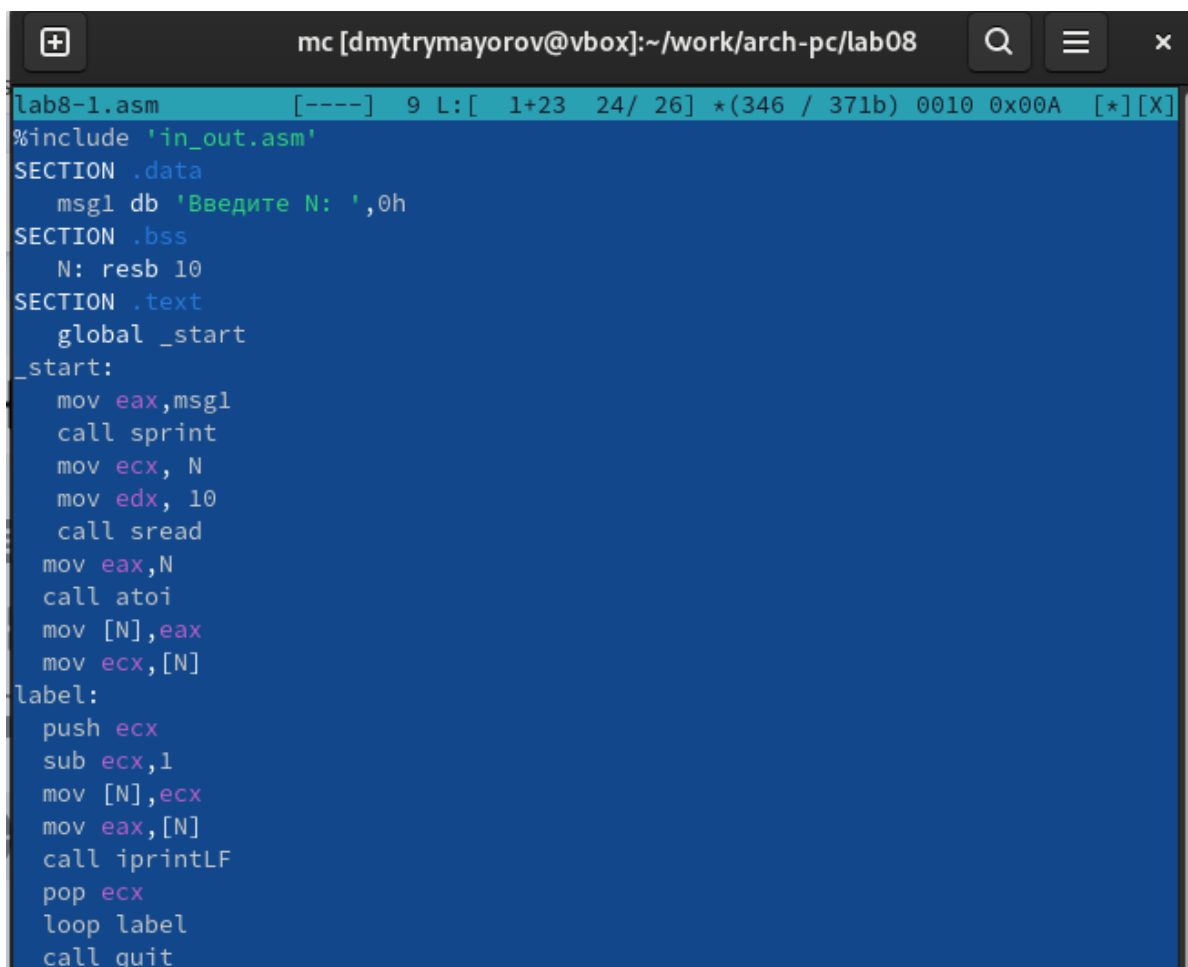


```
dmytrymayorov@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
7
5
3
1
dmytrymayorov@vbox:~/work/arch-pc/lab08$
```

Рис. 3.5: Создаем исполняемый файл и запускаем его

Регистр `ecx` принимает значения 9,7,5,3,1(на вход подается число 10, в цикле `label` данный регистр уменьшается на 2 командой `sub` и `loop`). Число проходов

цикла не соответствует числу N, так как уменьшается на 2. Снова открываем файл для редактирования и изменяем его, чтобы все корректно работало.



```
lab8-1.asm [----] 9 L: [ 1+23 24/ 26] *(346 / 371b) 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 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: Редактируем файл

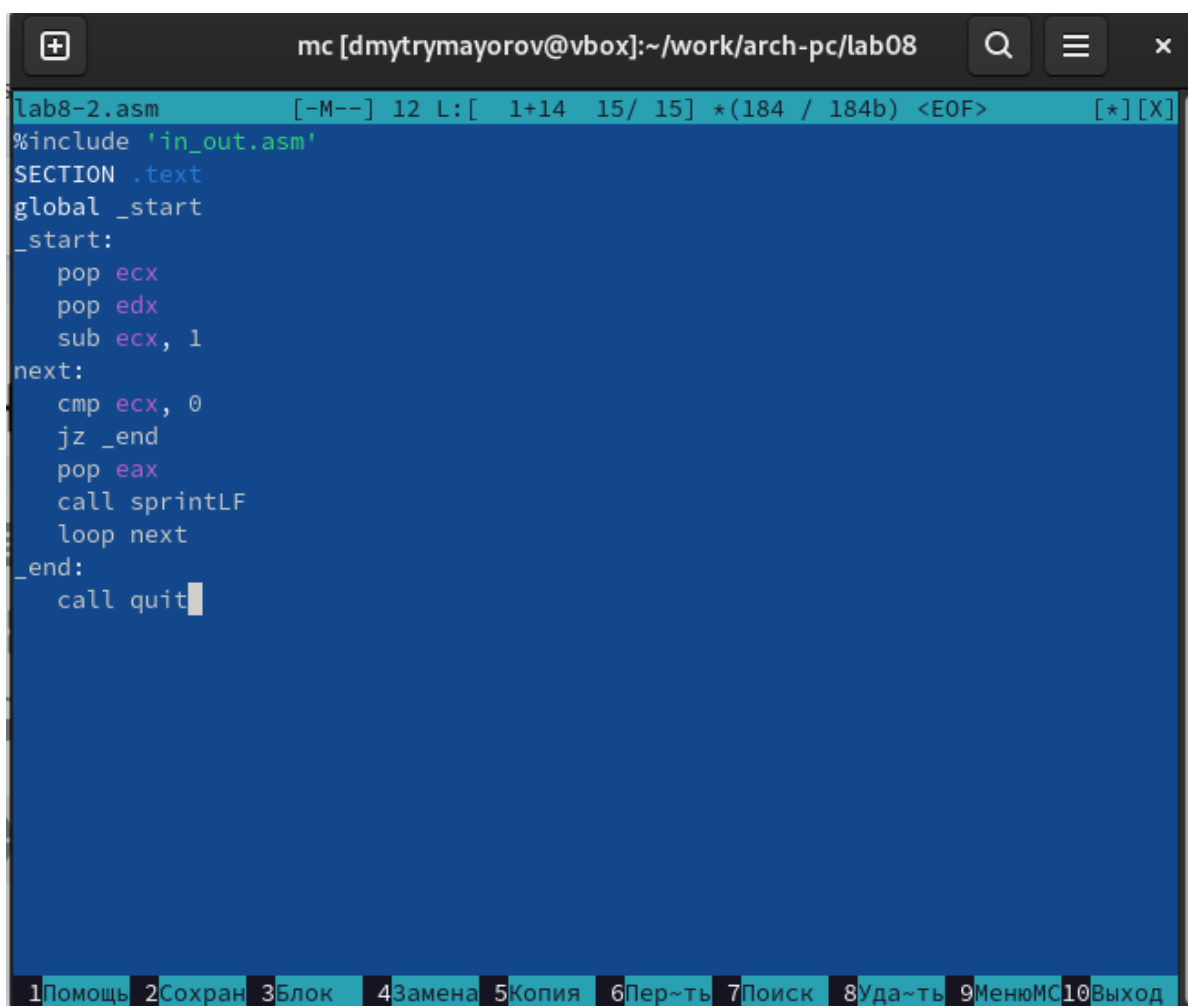
```
dmytrymayorov@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
dmytrymayorov@vbox:~/work/arch-pc/lab08$
```

Рис. 3.7: Создаем исполняемый файл и запускаем его

В данном случае число проходов цикла равна числу N.

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

Рис. 3.8: Создаем новый файл

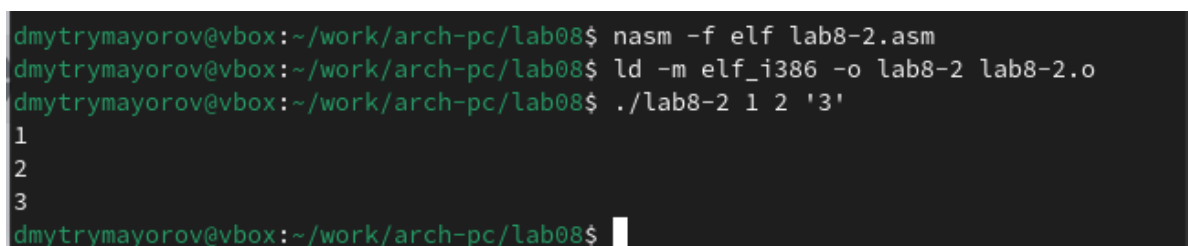


The screenshot shows a text editor window titled "mc [dmytrymayorov@vbox]:~/work/arch-pc/lab08". The editor is displaying the contents of "lab8-2.asm". The code is as follows:

```
lab8-2.asm      [-M--] 12 L:[ 1+14 15/ 15] *(184 / 184b) <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
```

At the bottom of the window, there is a menu bar with the following items: 1Помощь, 2Сохран, 3Блок, 4Замена, 5Копия, 6Пер~ть, 7Поиск, 8Уда~ть, 9МенюМС, 10Выход.

Рис. 3.9: Открываем файл и заполняем его в соответствии с листингом

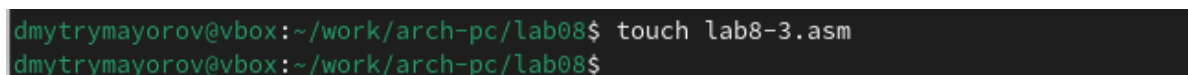


The screenshot shows a terminal window with the following commands and output:

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

Рис. 3.10: Создаем исполняемый файл и запускаем его

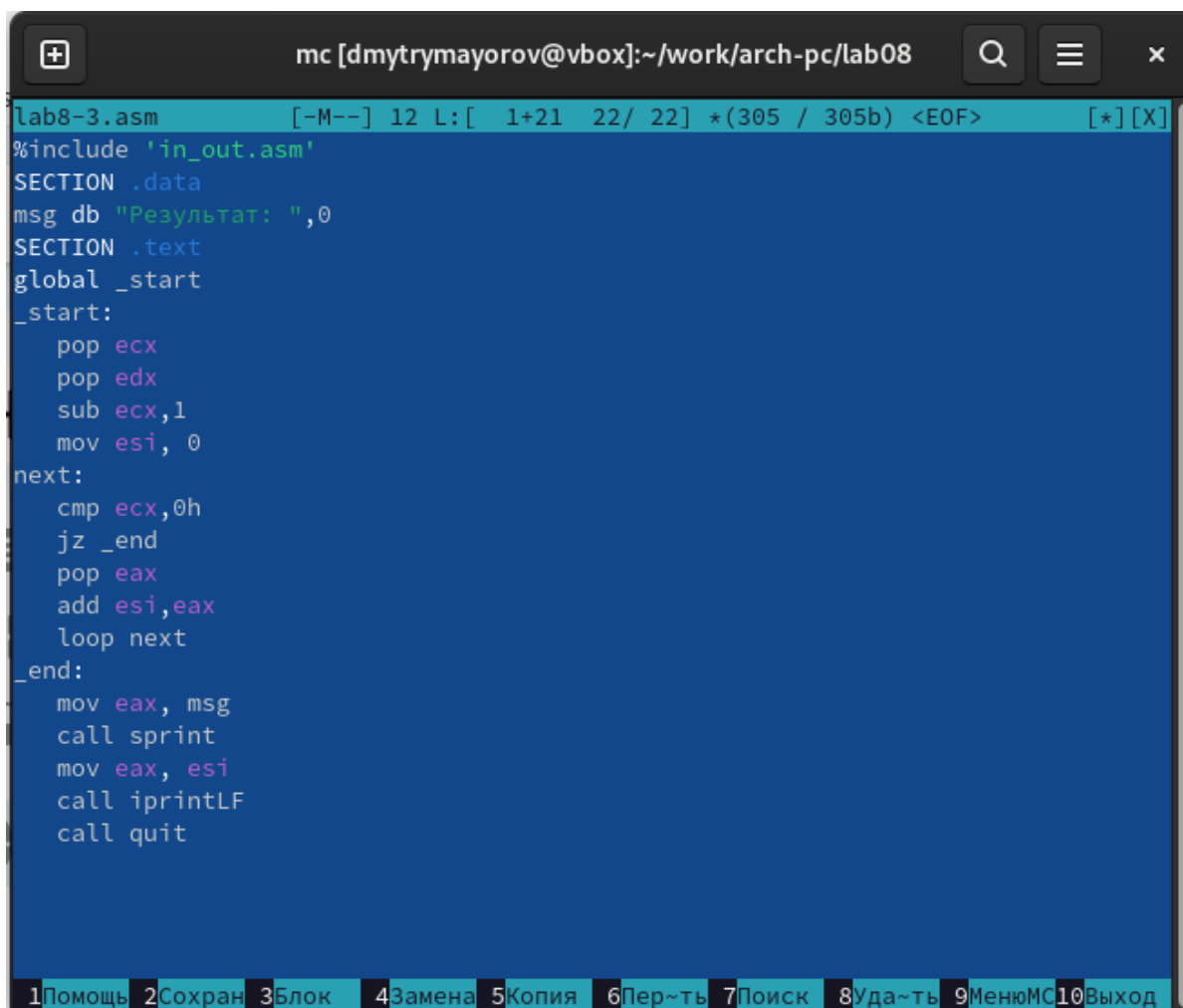
Программой было обработано 3 аргумента.



The screenshot shows a terminal window with the following commands:

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

Рис. 3.11: Создаем новый файл



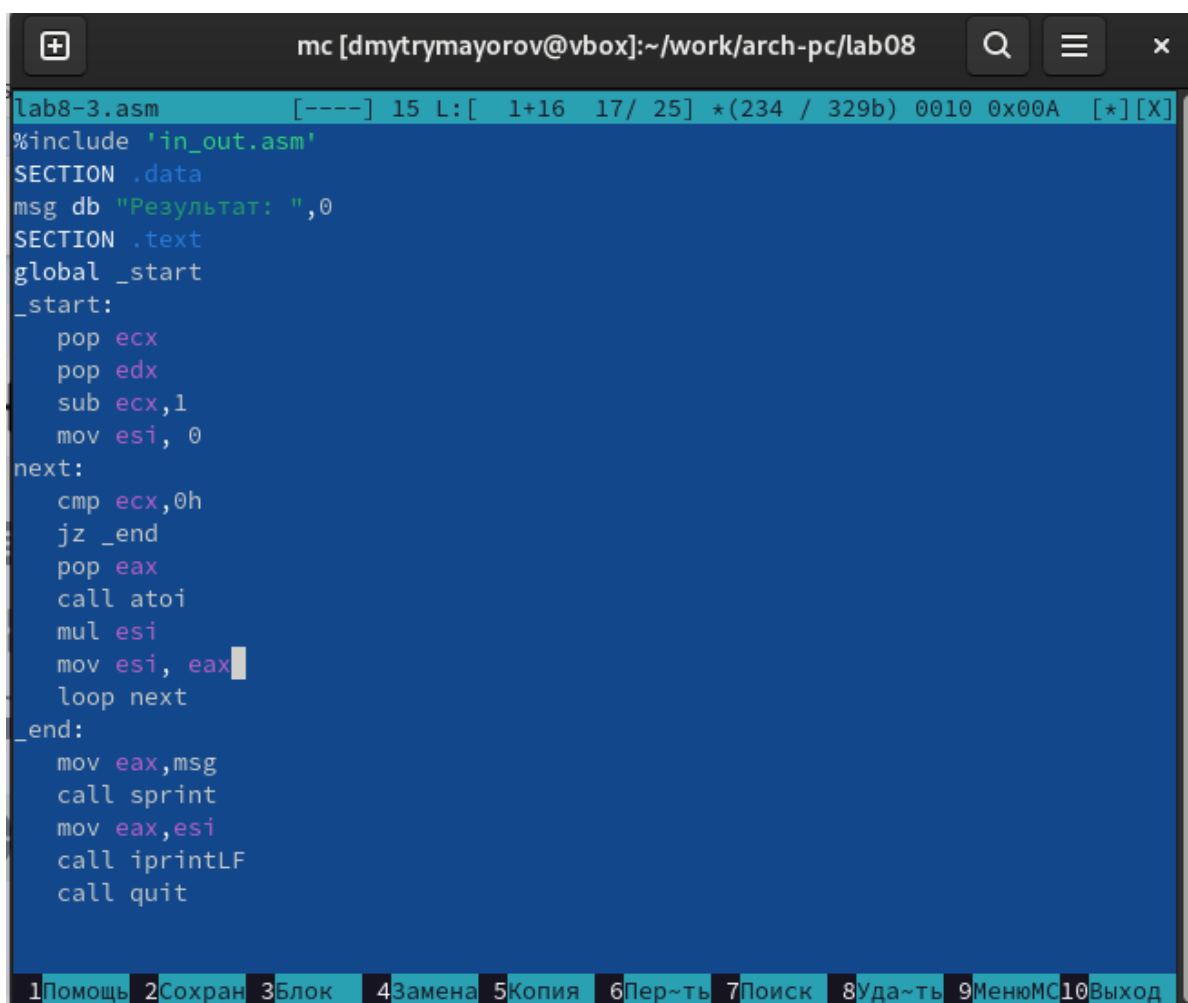
```
lab8-3.asm [-M--] 12 L: [ 1+21 22/ 22] *(305 / 305b) <EOF> [*] [X]
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx
    pop edx
    sub ecx,1
    mov esi, 0
next:
    cmp ecx,0h
    jz _end
    pop eax
    add esi,eax
    loop next
_end:
    mov eax, msg
    call sprint
    mov eax, esi
    call iprintLF
    call quit
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюМС10Выход

Рис. 3.12: Открываем файл и заполняем его в соответствии с листингом

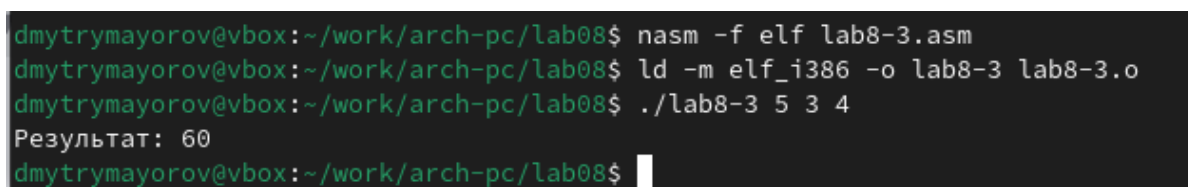
```
dmytrymayorov@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
dmytrymayorov@vbox:~/work/arch-pc/lab08$
```

Рис. 3.13: Создаем исполняемый файл и запускаем его



```
lab8-3.asm [----] 15 L: [ 1+16 17/ 25] *(234 / 329b) 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 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: Редактируем файл, чтобы вычислялось произведение вводимых значений



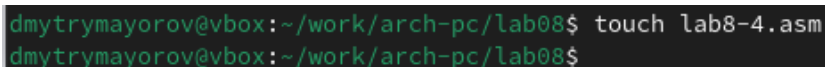
```
dmytrymayorov@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ./lab8-3 5 3 4
Результат: 60
dmytrymayorov@vbox:~/work/arch-pc/lab08$
```

Рис. 3.15: Создаем исполняемый файл и запускаем его

4 Задание для самостоятельной работы

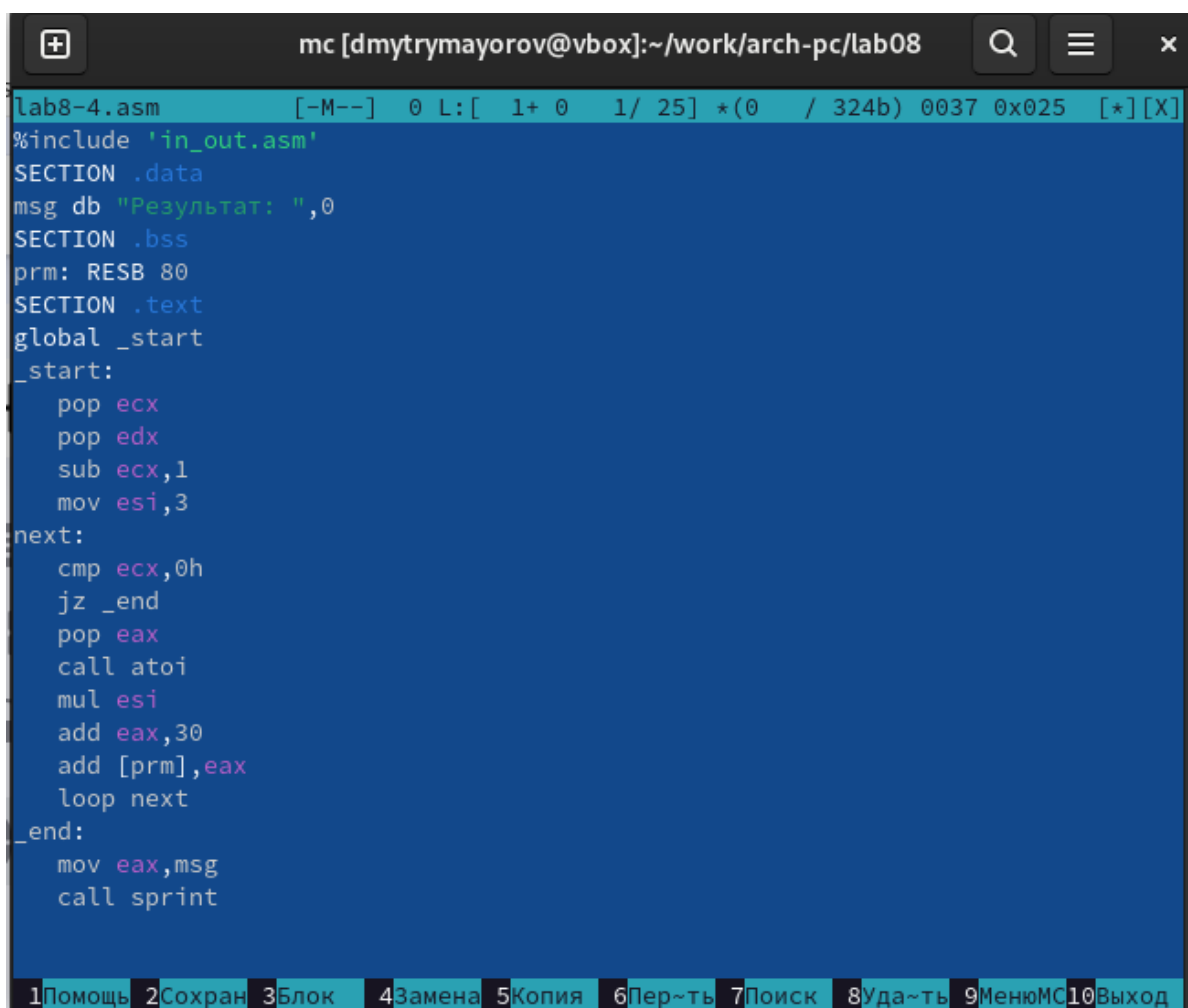
Вариант 20

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 $x = x_1, x_2, \dots, x_n$.



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

Рис. 4.1: Создаем новый файл



```
lab8-4.asm [-M--] 0 L:[ 1+ 0 1/ 25] *(0 / 324b) 0037 0x025 [*][X]
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .bss
prm: RESB 80
SECTION .text
global _start
_start:
    pop ecx
    pop edx
    sub ecx,1
    mov esi,3
next:
    cmp ecx,0h
    jz _end
    pop eax
    call atoi
    mul esi
    add eax,30
    add [prm],eax
    loop next
_end:
    mov eax,msg
    call sprint
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход

Рис. 4.2: Пишем программу

```
dmytrymayorov@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ./lab8-4 5 3 4
Результат: 126
dmytrymayorov@vbox:~/work/arch-pc/lab08$
```

Рис. 4.3: Создаем исполняемый файл и запускаем его. Смотрим на работу программы при $x_1=5$, $x_2=3$, $x_1=4$ (все правильно)


```
dmytrymayorov@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
dmytrymayorov@vbox:~/work/arch-pc/lab08$ ./lab8-4 5 3 4
Результат: 126
dmytrymayorov@vbox:~/work/arch-pc/lab08$
```

Рис. 4.4: Создаем исполняемый файл и запускаем его. Смотрим на работу программы при $x1=1$, $x2=3$, $x1=7$ (все правильно)

5 Выводы

Мы научились решать программы с использованием циклов и обработкой аргументов командной строки.

Список литературы