

# **Отчет по лабораторной работе № 8**

**Дисциплина: архитектура компьютеров**

Казазаев Даниил Михайлович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задания Лабораторной работы</b>	<b>5</b>
<b>3</b>	<b>Задания Самостоятельной работы</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Выполнение самостоятельной работы</b>	<b>15</b>
5.1	Задание 1 . . . . .	15

## Список иллюстраций

4.1	Создание файла lab8-1.asm . . . . .	7
4.2	Перенесенный листинг 8.1 . . . . .	8
4.3	Трансляция и запуск файла lab8-1.asm . . . . .	8
4.4	Редактирование файла lab8-1.asm . . . . .	9
4.5	Трансляция и запуск файла lab8-1.asm . . . . .	9
4.6	Редактирование файла lab8-1.asm . . . . .	10
4.7	Трансляция и запуск файла lab8-1.asm . . . . .	10
4.8	Создание файла lab8-2.asm . . . . .	11
4.9	Перенесенный листинг 8.2 . . . . .	11
4.10	Трансляция и запуск файла lab8-2.asm . . . . .	11
4.11	Создание файла lab8-2.asm . . . . .	12
4.12	Перенесенный листинг 8.3 . . . . .	12
4.13	Удаление одного из операндов . . . . .	13
4.14	Трансляция и запуск файла lab8-3.asm . . . . .	13
4.15	Редактирование файла lab8-1.asm . . . . .	14
4.16	Трансляция и запуск файла lab8-3.asm . . . . .	14
5.1	Созадние файла task.asm . . . . .	15
5.2	Редактирую файл . . . . .	16
5.3	Трансляция и запуск файла . . . . .	17

# 1 Цель работы

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

## 2 Задания Лабораторной работы

1. Создать файл lab8-1.asm.
2. Ввести в файл lab8-1.asm текст программы из листинга 8.1. Создать исполняемый файл и проверить его работу.
3. Изменить текст программы, добавив изменение значения регистра esx в цикле. Создать исполняемый файл и проверить его работу.
4. Еще раз изменить текст программы, добавив команды push и pop. Создать исполняемый файл и проверить его работу.
5. Создать файл lab8-2.asm.
6. Ввести в файл lab8-2.asm текст программы из листинга 8.2. Создать исполняемый файл и проверить его работу.
7. Создать файл lab8-3.asm.
8. Ввести в файл lab8-3.asm текст программы из листинга 8.3. Создать исполняемый файл и проверить его работу.
9. Отредактировать текст программы, чтобы результатом было произведение введенных аргументов.

### 3 Задания Самостоятельной работы

1. Написать программу, которая находит сумму значений функции  $f(x)$  для  $x = x_1, x_2, x_3, \dots, x_n$

## 4 Выполнение лабораторной работы

Создаю файл lab8-1.asm. (рис. [4.1])

```
dmkazaev@Ubuntu:~$ cd ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/labs/Lab08/
dmkazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/Lab08$ touch lab8-1.asm
dmkazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/Lab08$ ls
lab8-1.asm  presentation  report
```

Рис. 4.1: Создание файла lab8-1.asm

Ввожу в файл lab8-1.asm текст программы из листинга 8.1. (рис. [4.2])



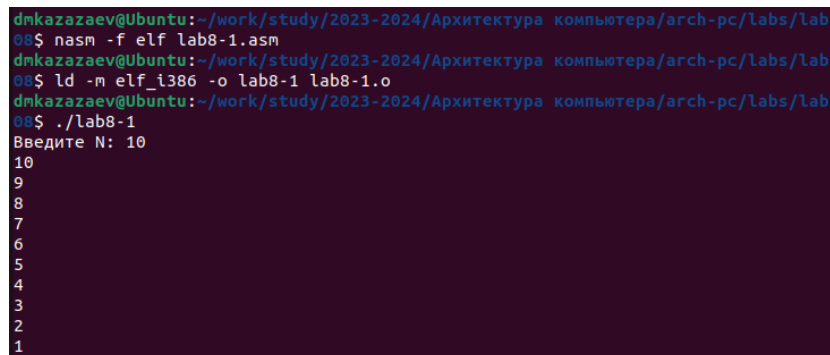
```

1 %include 'in_out.asm'
2
3 SECTION .data
4     msg1 db 'Введите N: ', 0h
5
6 SECTION .bss
7     N: resb 10
8
9 SECTION .text
10    global _start
11 _start:
12
13
14 ; ---- Вывод сообщения
15     mov eax, msg1
16     call sprint
17
18 ; ---- Ввод 'N'
19     mov ecx, N
20     mov edx, 10
21     call sread
22
23 ; ---- Преобразование из символа в число
24     mov eax, N
25     call atoi
26     mov [N],eax
27
28 ; ---- Цикл
29     mov ecx,[N] ; Счетчик цикла 'ecx = N'
30 label:
31     mov[N],ecx
32     mov eax,[N]
33     call iprintLF ; Вывод значения 'N'
34     loop label ; 'ecx = ecx - 1' и если 'ecx' не '0'
35
36     call quit

```

Рис. 4.2: Перенесенный листинг 8.1

Транслирую файл lab8-1.asm в объектный файл, после чего запускаю его. (рис. [4.3])



```

dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ nasm -f elf lab8-1.asm
dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ld -m elf_i386 -o lab8-1 lab8-1.o
dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1

```

Рис. 4.3: Трансляция и запуск файла lab8-1.asm

Немного редактирую файл lab8-1.asm. (рис. [4.4])

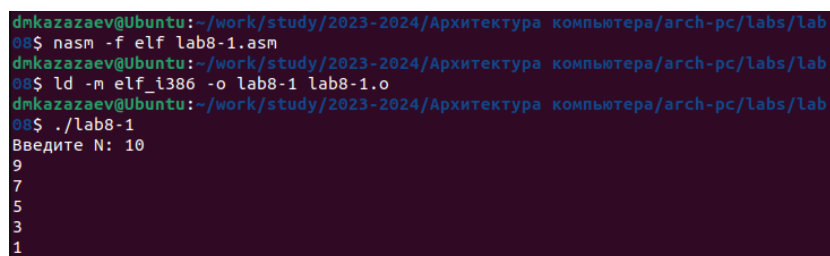




```
1 %include 'in_out.asm'
2
3 SECTION .data
4     msg1 db 'Введите N: ', 0h
5
6 SECTION .bss
7     N: resb 10
8
9 SECTION .text
10    global _start
11 _start:
12
13
14 ; ----- Вывод сообщения
15     mov eax, msg1
16     call sprint
17
18 ; ----- Ввод 'N'
19     mov ecx, N
20     mov edx, 10
21     call sread
22
23 ; ----- Преобразование из символа в число
24     mov eax, N
25     call atoi
26     mov [N],eax
27
28 ; ----- Цикл
29     mov ecx,[N] ; Счетчик цикла 'ecx = N'
30 label:
31     sub ecx,1
32     mov [N],ecx
33     mov eax,[N]
34     call iprintLF
35     loop label
36
37     call quit
```

Рис. 4.4: Редактирование файла lab8-1.asm

Транслирую файл lab8-1.asm в объектный файл, после чего запускаю его. (рис. [4.5])

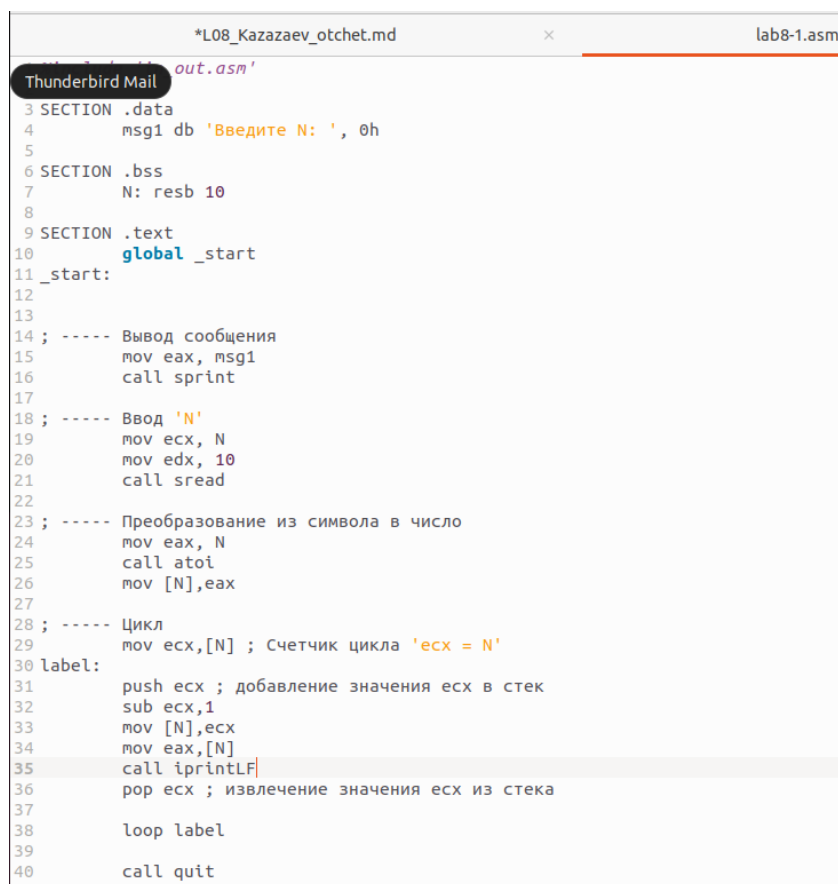


```
dmkazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ nasm -f elf lab8-1.asm
dmkazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ld -m elf_i386 -o lab8-1 lab8-1.o
dmkazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ./lab8-1
Введите N: 10
9
7
5
3
1
```

Рис. 4.5: Трансляция и запуск файла lab8-1.asm

Разница в том, что из регистра ecx с каждым проходом цикла вычитается единица (sub ecx, 1). Поэтому число проходов цикла не соответствует значению N, введенному с клавиатуры.

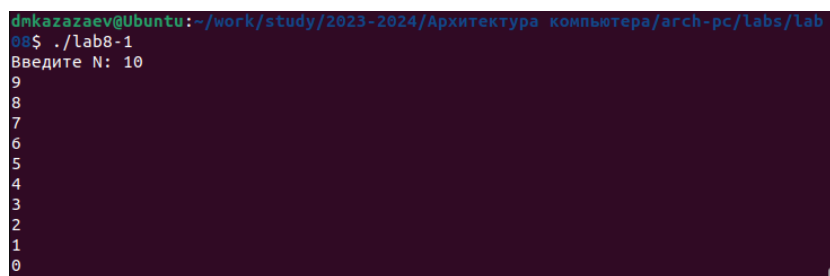
Еще раз немного редактирую файл lab8-1.asm. (рис. [4.6])



```
3 SECTION .data
4     msg1 db 'Введите N: ', 0h
5
6 SECTION .bss
7     N: resb 10
8
9 SECTION .text
10    global _start
11 _start:
12
13
14 ; ---- Вывод сообщения
15     mov eax, msg1
16     call sprint
17
18 ; ---- Ввод 'N'
19     mov ecx, N
20     mov edx, 10
21     call sread
22
23 ; ---- Преобразование из символа в число
24     mov eax, N
25     call atoi
26     mov [N],eax
27
28 ; ---- Цикл
29     mov ecx,[N] ; Счетчик цикла 'ecx = N'
30 label:
31     push ecx ; добавление значения ecx в стек
32     sub ecx,1
33     mov [N],ecx
34     mov eax,[N]
35     call iprintLF
36     pop ecx ; извлечение значения ecx из стека
37
38     loop label
39
40     call quit
```

Рис. 4.6: Редактирование файла lab8-1.asm

Транслирую файл lab8-1.asm в объектный файл, после чего запускаю его. (рис. [4.7])



```
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab08$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
```

Рис. 4.7: Трансляция и запуск файла lab8-1.asm

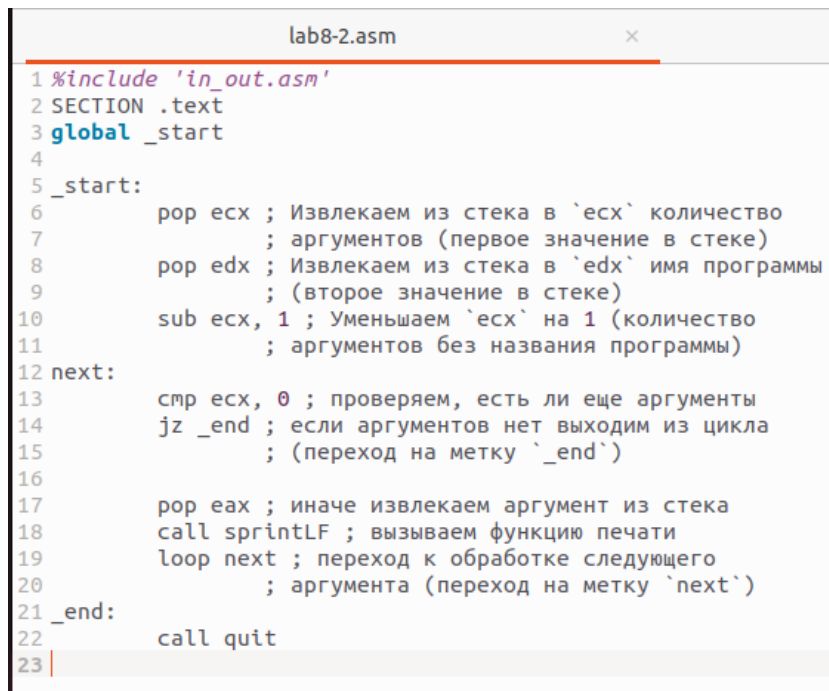
Количество циклов равно введенному числу N.

Создаю файл lab8-2.asm. (рис. [4.8])

```
dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ touch lab8-2.asm
```

Рис. 4.8: Создание файла lab8-2.asm

Переношу в файл lab8-2.asm текст программы из листинга 8.2. (рис. [4.9])



```
lab8-2.asm
1 %include 'in_out.asm'
2 SECTION .text
3 global _start
4
5 _start:
6     pop ecx ; Извлекаем из стека в `ecx` количество
7             ; аргументов (первое значение в стеке)
8     pop edx ; Извлекаем из стека в `edx` имя программы
9             ; (второе значение в стеке)
10    sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество
11              ; аргументов без названия программы)
12 next:
13    cmp ecx, 0 ; проверяем, есть ли еще аргументы
14    jz _end ; если аргументов нет выходим из цикла
15             ; (переход на метку `_end`)
16
17    pop eax ; иначе извлекаем аргумент из стека
18    call printf ; вызываем функцию печати
19    loop next ; переход к обработке следующего
20              ; аргумента (переход на метку `next`)
21 _end:
22    call quit
23
```

Рис. 4.9: Перенесенный листинг 8.2

Транслирую файл lab8-2.asm в объектный файл, после чего запускаю его. (рис. [4.10])

```
dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ nasm -f elf lab8-2.asm
dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ld -m elf_i386 -o lab8-2 lab8-2.o
dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ./lab8-2 3 2 '4'
3
2
4
```

Рис. 4.10: Трансляция и запуск файла lab8-2.asm

Создаю файл lab8-3.asm. (рис. [4.11])

```
dmkazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ touch lab8-3.asm
```

Рис. 4.11: Создание файла lab8-2.asm

Переношу в файл lab8-3.asm текст программы из листинга 8.3. (рис. [4.13])



```
lab8-2.asm  x  lab8-3.asm
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4
5 SECTION .text
6 global _start
7
8 _start:
9
10     pop ecx        ; Извлекаем из стека в `ecx` количество
11                   ; аргументов (первое значение в стеке)
12     pop edx        ; Извлекаем из стека в `edx` имя программы
13                   ; (второе значение в стеке)
14     sub ecx,1      ; Уменьшаем `ecx` на 1 (количество
15                   ; аргументов без названия программы)
16     mov esi, 0     ; Используем `esi` для хранения
17                   ; промежуточных сумм
18 next:
19     cmp ecx,0h     ; проверяем, есть ли еще аргументы
20     jz _end        ; если аргументов нет выходим из цикла
21                   ; (переход на метку `_end`)
22
23     pop eax        ; иначе извлекаем следующий аргумент из стека
24     call atoi      ; преобразуем символ в число
25     add esi,eax    ; добавляем к промежуточной сумме
26                   ; след. аргумент `esi=esi+eax`
27
28     loop next      ; переход к обработке следующего аргумента
29 _end:
30     mov eax, msg    ; вывод сообщения "Результат: "
31     call sprint
32     mov eax, esi    ; записываем сумму в регистр `eax`
33     call iprintfLF ; печать результата
34     call quit      ; завершение программы
```

Рис. 4.12: Перенесенный листинг 8.3

```

lab8-2.asm
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4
5 SECTION .text
6 global _start
7
8 _start:
9
10     pop ecx        ; Извлекаем из стека в `ecx` количество
11                    ; аргументов (первое значение в стеке)
12     pop edx        ; Извлекаем из стека в `edx` имя программы
13                    ; (второе значение в стеке)
14     sub ecx,1      ; Уменьшаем `ecx` на 1 (количество
15                    ; аргументов без названия программы)
16     mov esi, 0     ; Используем `esi` для хранения
17                    ; промежуточных сумм
18 next:
19     cmp ecx,0h     ; проверяем, есть ли еще аргументы
20     jz _end        ; если аргументов нет выходим из цикла
21                    ; (переход на метку `_end`)
22
23     pop eax        ; иначе извлекаем следующий аргумент из стека
24     call atoi      ; преобразуем символ в число
25     add esi,eax    ; добавляем к промежуточной сумме
26                    ; след. аргумент `esi=esi+eax`
27
28     loop next      ; переход к обработке следующего аргумента
29 _end:
30     mov eax, msg   ; вывод сообщения "Результат: "
31     call sprintf   ; записываем сумму в регистр `eax`
32     mov eax, esi   ; печатаем результат
33     call iprintfLF
34     call quit      ; завершение программы

```

Рис. 4.13: Удаление одного из операндов

Транслирую файл lab8-3.asm в объектный файл, после чего запускаю его. (рис. [4.14])

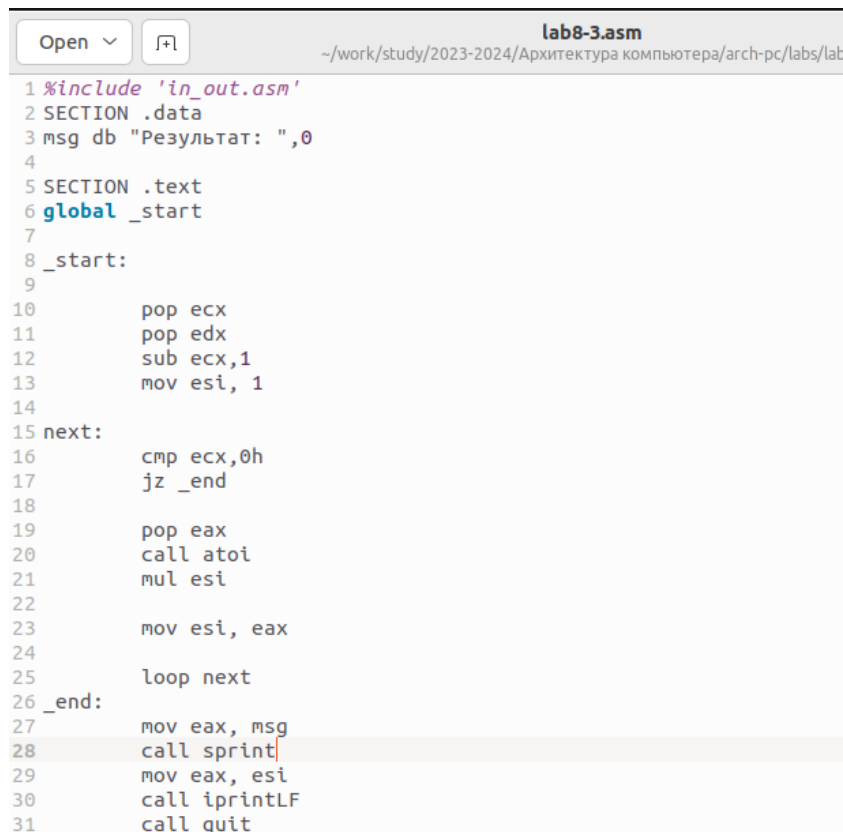
```

dnkazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ nasm -f elf lab8-3.asm
dnkazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ld -m elf_i386 -o lab8-3 lab8-3.o
dnkazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ./lab8-3 12 13 7 10 5
Результат: 47

```

Рис. 4.14: Трансляция и запуск файла lab8-3.asm

Немного редактирую файл lab8-3.asm, чтобы результатом было произведение введенных чисел. (рис. [??])

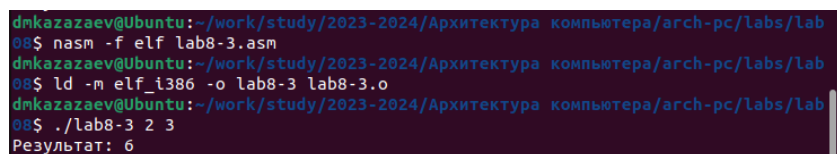


```
lab8-3.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab

1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4
5 SECTION .text
6 global _start
7
8 _start:
9
10     pop ecx
11     pop edx
12     sub ecx,1
13     mov esi, 1
14
15 next:
16     cmp ecx,0h
17     jz _end
18
19     pop eax
20     call atoi
21     mul esi
22
23     mov esi, eax
24
25     loop next
26 _end:
27     mov eax, msg
28     call sprint
29     mov eax, esi
30     call iprintLF
31     call quit
```

Рис. 4.15: Редактирование файла lab8-1.asm

Транслирую файл lab8-3.asm в объектный файл, после чего запускаю его. (рис. [4.16])



```
dmkazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ nasm -f elf lab8-3.asm
dmkazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ld -m elf_i386 -o lab8-3 lab8-3.o
dmkazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ ./lab8-3 2 3
Результат: 6
```

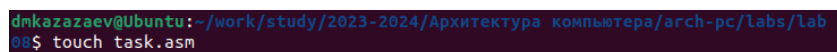
Рис. 4.16: Трансляция и запуск файла lab8-3.asm

## 5 Выполнение самостоятельной работы

После выполнения прошлой лабораторной работы я получил вариант 8.

### 5.1 Задание 1

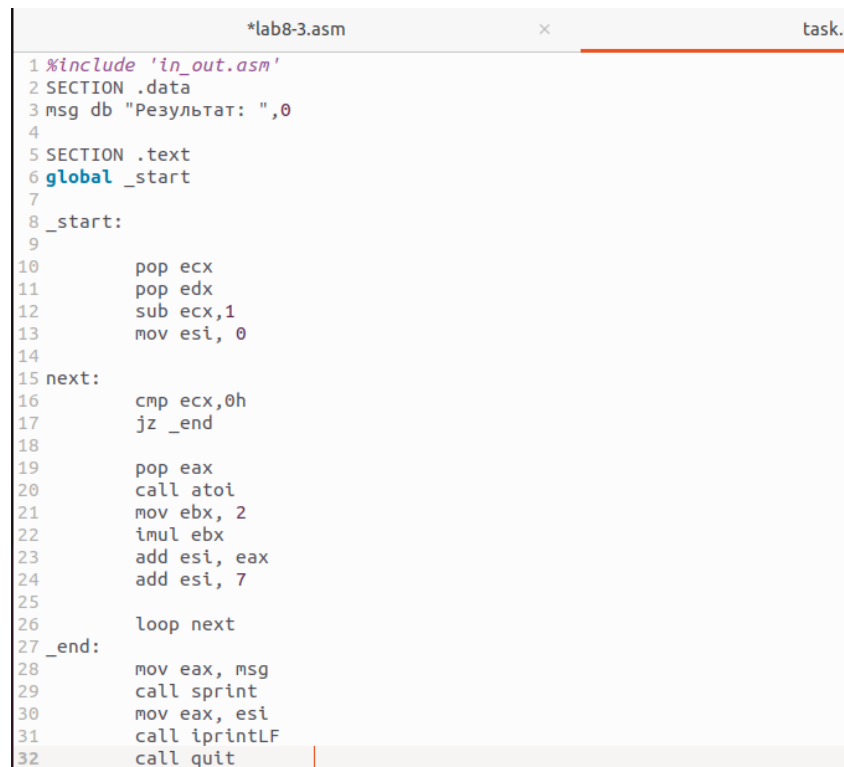
Создаю файл task.asm, в котором буду выполнять задание. (рис. [5.1])



```
dmkazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
08$ touch task.asm
```

Рис. 5.1: Созадние файла task.asm

Редактирую файл task.asm. (рис. [5.3])

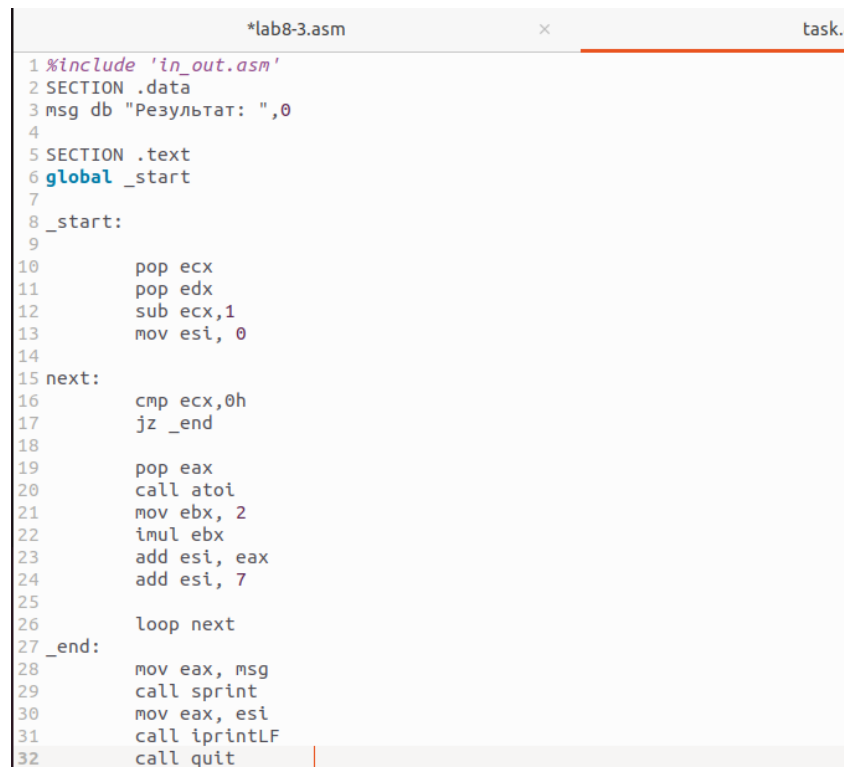


```
*lab8-3.asm
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4
5 SECTION .text
6 global _start
7
8 _start:
9
10     pop ecx
11     pop edx
12     sub ecx,1
13     mov esi, 0
14
15 next:
16     cmp ecx,0h
17     jz _end
18
19     pop eax
20     call atoi
21     mov ebx, 2
22     imul ebx
23     add esi, eax
24     add esi, 7
25
26     loop next
27 _end:
28     mov eax, msg
29     call sprint
30     mov eax, esi
31     call iprintLF
32     call quit
```

Рис. 5.2: Редактирую файл

Транслирую файл task.asm в объектный файл, после чего запускаю его. (рис. [5.3])





```
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4
5 SECTION .text
6 global _start
7
8 _start:
9
10     pop ecx
11     pop edx
12     sub ecx,1
13     mov esi, 0
14
15 next:
16     cmp ecx,0h
17     jz _end
18
19     pop eax
20     call atoi
21     mov ebx, 2
22     imul ebx
23     add esi, eax
24     add esi, 7
25
26     loop next
27 _end:
28     mov eax, msg
29     call sprint
30     mov eax, esi
31     call iprintLF
32     call quit
```

Рис. 5.3: Трансляция и запуск файла

Программа работает корректно.

#### Листинг task.asm

```
%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
    mov ebx, 2
    imul ebx
    add esi, eax
    add esi, 7

    loop next
_end:
    mov eax, msg
    call sprint
    mov eax, esi
    call iprintLF
    call quit

```

#### #Вывод

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