

Отчет по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки

Мальянц Виктория Кареновна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Реализация циклов в NASM	7
3.2	Обработка аргументов командной строки	11
3.3	Выполнение задания для самостоятельной работы	15
4	Выводы	18

Список иллюстраций

3.1	Создание каталога и файла для программы	7
3.2	Копирование файла и просматривание содержимого каталога . . .	7
3.3	Открытие файла lab8-1.asm в текстовом редакторе gedit	7
3.4	Редактирование файла	8
3.5	Запуск исполняемого файла	8
3.6	Открытие файла lab8-1.asm в текстовом редакторе gedit	8
3.7	Редактирование файла	9
3.8	Запуск исполняемого файла	9
3.9	Открытие файла lab8-1.asm в текстовом редакторе gedit	10
3.10	Редактирование файла	10
3.11	Запуск исполняемого файла	11
3.12	Создание файла	11
3.13	Открытие файла lab8-2.asm в текстовом редакторе gedit	11
3.14	Редактирование файла	12
3.15	Запуск исполняемого файла	12
3.16	Создание файла	12
3.17	Открытие файла lab8-3.asm в текстовом редакторе gedit	13
3.18	Редактирование файла	13
3.19	Запуск исполняемого файла	13
3.20	Открытие файла lab8-3.asm в текстовом редакторе gedit	14
3.21	Редактирование файла	14
3.22	Запуск исполняемого файла	14
3.23	Создание файла	15
3.24	Открытие файла lab8-4.asm в текстовом редакторе gedit	15
3.25	Редактирование файла	15
3.26	Запуск исполняемого файла	16
3.27	Запуск исполняемого файла	16

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 задания для самостоятельной работы

3 Выполнение лабораторной работы

3.1 Реализация циклов в NASM

Создаю каталог для программ лабораторной работы № 8, перехожу в него и создаю файл lab8-1.asm (рис. 3.1).

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

Рис. 3.1: Создание каталога и файла для программы

С помощью команды `cp` копирую файл `in_out.asm` и просматриваю содержимое каталога `lab08` с помощью команды `ls` (рис. 3.2).

```
vkmaljyanc@vbox: ~/work/arch-pc/lab08$ cp ~/Зарядки/in_out.asm in_out.asm
vkmaljyanc@vbox: ~/work/arch-pc/lab08$ ls
in_out.asm  lab8-1.asm
```

Рис. 3.2: Копирование файла и просмотр содержимого каталога

Открываю файл `lab8-1.asm` в текстовом редакторе `gedit` через терминал (рис. 3.3).

```
vkmaljyanc@vbox: ~/work/arch-pc/lab08$ gedit lab8-1.asm
```

Рис. 3.3: Открытие файла `lab8-1.asm` в текстовом редакторе `gedit`

Ввожу в файл `lab8-1.asm` программу вывода значений регистра `ecx` (рис. 3.4).

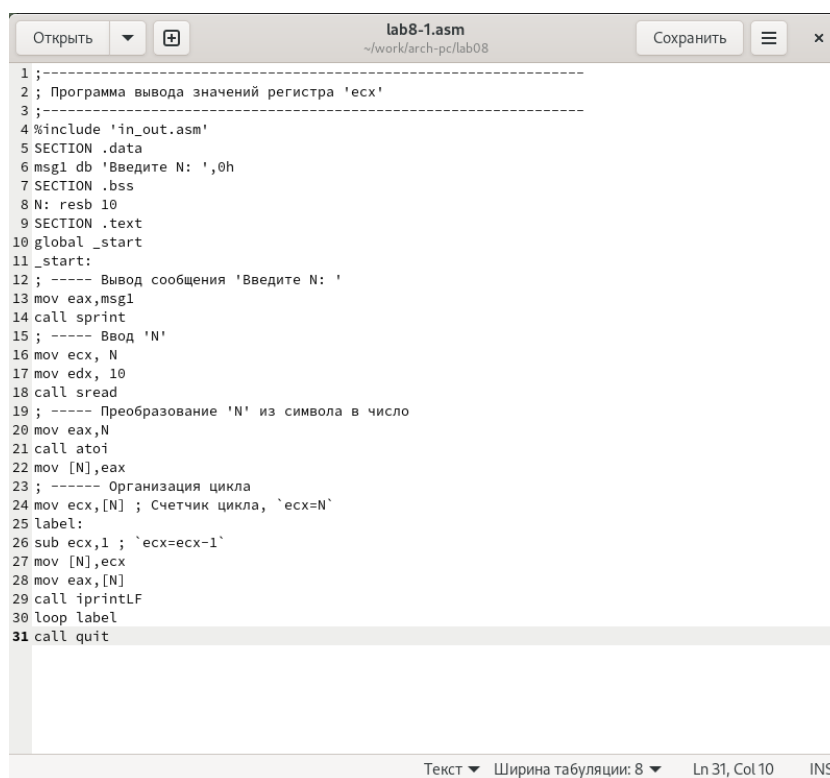


Рис. 3.4: Редактирование файла

Создаю исполняемый файл и запускаю его. Проверяю работу исполняемого файла для значения N равного 6. Убеждаюсь в том, что программа работает корректно (рис. 3.5).

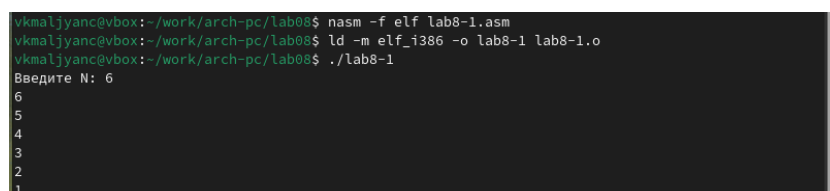


Рис. 3.5: Запуск исполняемого файла

Открываю файл lab8-1.asm в текстовом редакторе gedit через терминал (рис. 3.6).

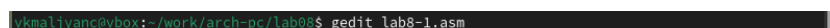
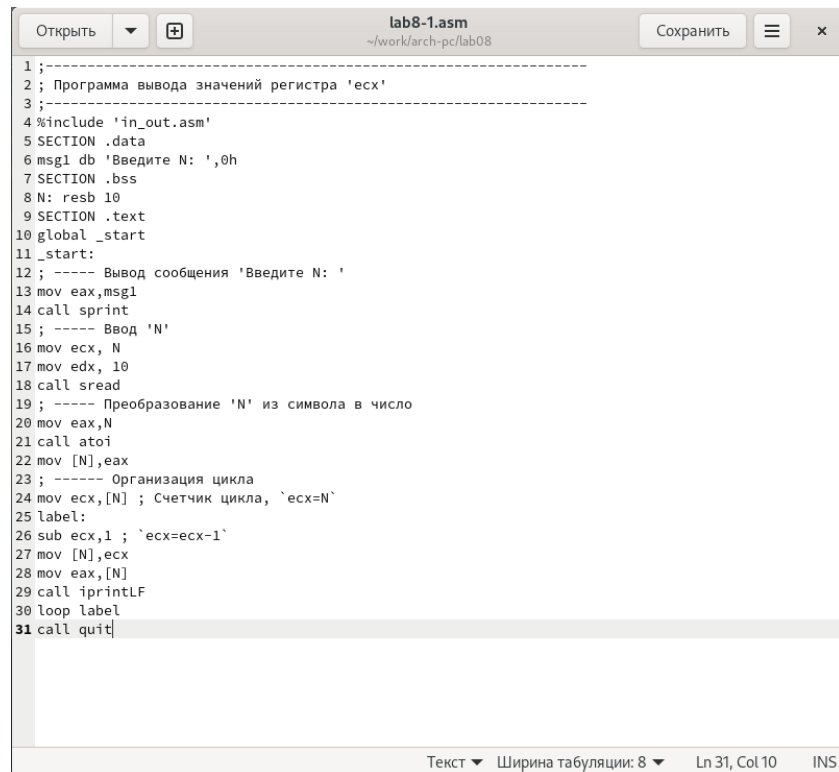


Рис. 3.6: Открытие файла lab8-1.asm в текстовом редакторе gedit

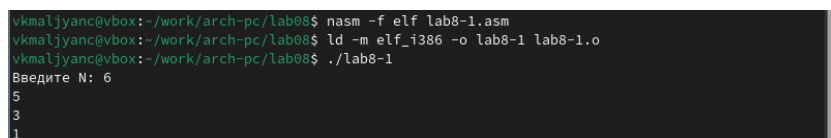
В тексте программы файла lab8-1.asm добавляю изменение значения регистра ecx в цикле (рис. 3.7).



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

Рис. 3.7: Редактирование файла

Создаю исполняемый файл и запускаю его. Проверяю работу исполняемого файла для значения N равного 6. Регистр ecx на каждой итерации уменьшается на 2 значения, количество итераций уменьшается в 2 раза, поэтому число проходов цикла не соответствует значению N, введенному с клавиатуры (рис. 3.8).



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

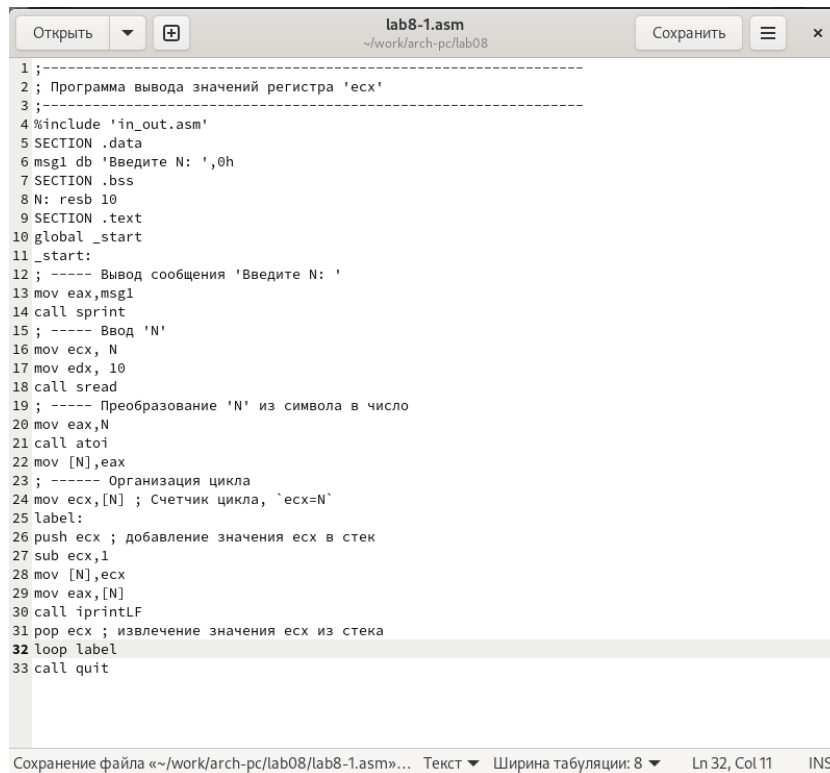
Рис. 3.8: Запуск исполняемого файла

Открываю файл lab8-1.asm в текстовом редакторе gedit через терминал (рис. 3.9).

```
vkmaljyanc@vbox: ~/work/arch-pc/lab08$ gedit lab8-1.asm
```

Рис. 3.9: Открытие файла lab8-1.asm в текстовом редакторе gedit

В тексте программы файла lab8-1.asm добавляю команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop(рис. 3.10).



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

Рис. 3.10: Редактирование файла

Создаю исполняемый файл и запускаю его. Проверяю работу исполняемого файла для значения N равного 6. В данном случае число проходов цикла соответствует значению N введенному с клавиатуры, но со смещением выводимых чисел на -1 (рис. 3.11).

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

Рис. 3.11: Запуск исполняемого файла

3.2 Обработка аргументов командной строки

С помощью команды `touch` создаю файл `lab8-2.asm` (рис. 3.12).

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

Рис. 3.12: Создание файла

Открываю файл `lab8-2.asm` в текстовом редакторе `gedit` через терминал (рис. 3.13).

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

Рис. 3.13: Открытие файла `lab8-2.asm` в текстовом редакторе `gedit`

Ввожу в файл `lab8-2.asm` программу, выводящую на экран аргументы командной строки (рис. 3.14).

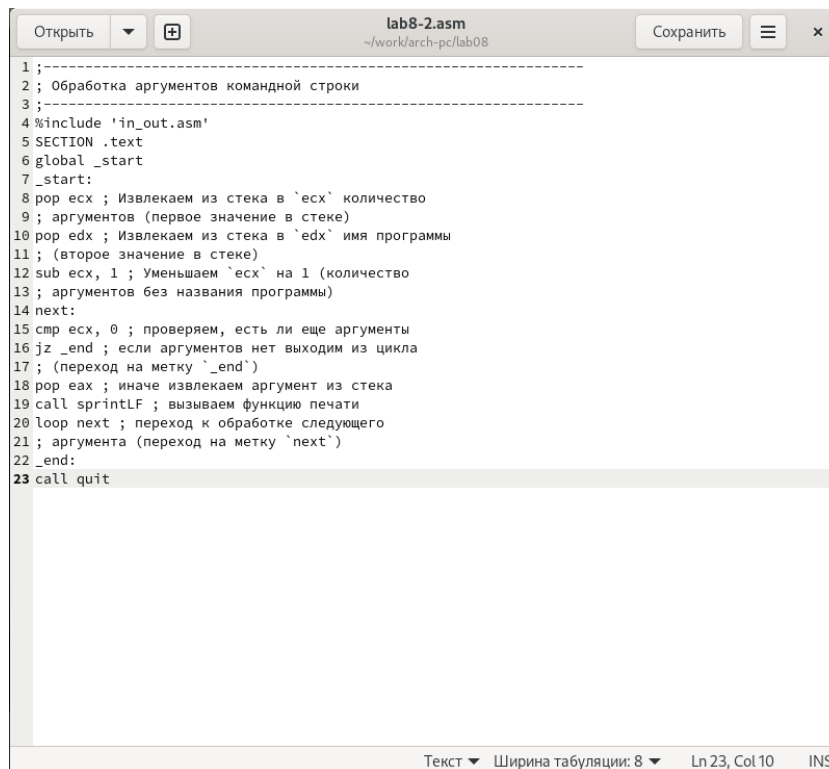


Рис. 3.14: Редактирование файла

Создаю исполняемый файл и запускаю его. Проверяю работу исполняемого файла, указав аргументы: аргумент1 аргумент 2 'аргумент 3'. Программа обработала 4 аргумента (Аргумент 2 - 'Аргумент' и '2' - это два разных аргумента, так как они записаны через пробел) (рис. 3.15).

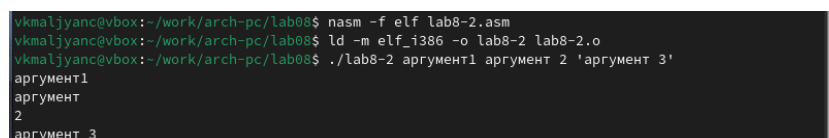


Рис. 3.15: Запуск исполняемого файла

С помощью команды `touch` создаю файл `lab8-3.asm` (рис. 3.16).

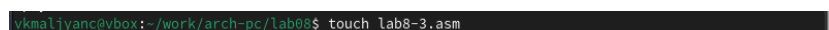


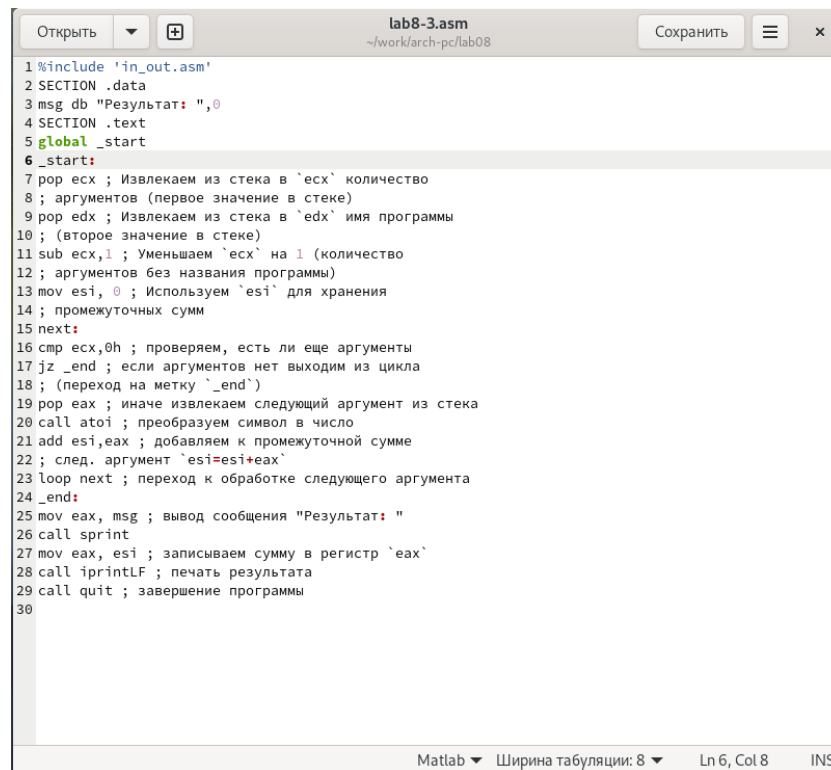
Рис. 3.16: Создание файла

Открываю файл `lab8-3.asm` в текстовом редакторе `gedit` через терминал (рис. 3.17).

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

Рис. 3.17: Открытие файла lab8-3.asm в текстовом редакторе gedit

Ввожу в файл lab8-3.asm программу вычисления суммы аргументов командной строки (рис. 3.18).



```
lab8-3.asm
~/work/arch-pc/lab08

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

Рис. 3.18: Редактирование файла

Создаю исполняемый файл и запускаю его. Проверяю работу исполняемого файла, указав аргументы: 12 13 7 10 5. Убеждаюсь в том, что программа работает корректно (рис. 3.19).

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

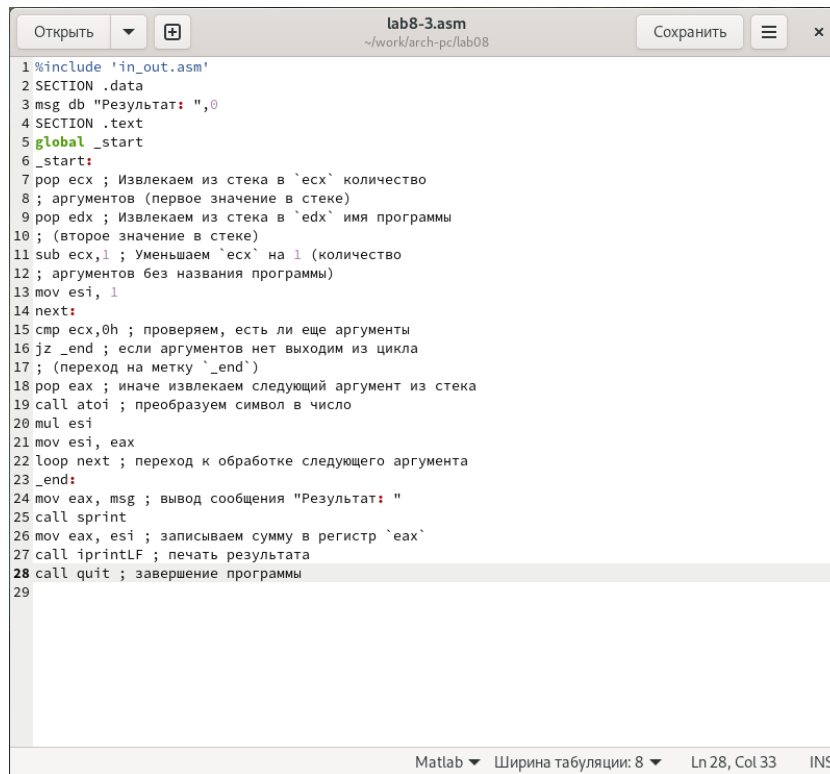
Рис. 3.19: Запуск исполняемого файла

Открываю файл lab8-3.asm в текстовом редакторе gedit через терминал (рис. 3.20).

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

Рис. 3.20: Открытие файла lab8-3.asm в текстовом редакторе gedit

Изменяю текст программы файла lab8-3.asm для вычисления произведения аргументов командной строки (рис. 3.21).



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

Рис. 3.21: Редактирование файла

Создаю исполняемый файл и запускаю его. Проверяю работу исполняемого файла, указав аргументы: 12 13 7 10 5. Убеждаюсь в том, что программа работает корректно (рис. 3.22).

```
vkmaljyanc@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
vkmaljyanc@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
vkmaljyanc@vbox:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 54600
```

Рис. 3.22: Запуск исполняемого файла

3.3 Выполнение задания для самостоятельной работы

С помощью команды `touch` создаю файл `lab8-4.asm` (рис. 3.23).

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

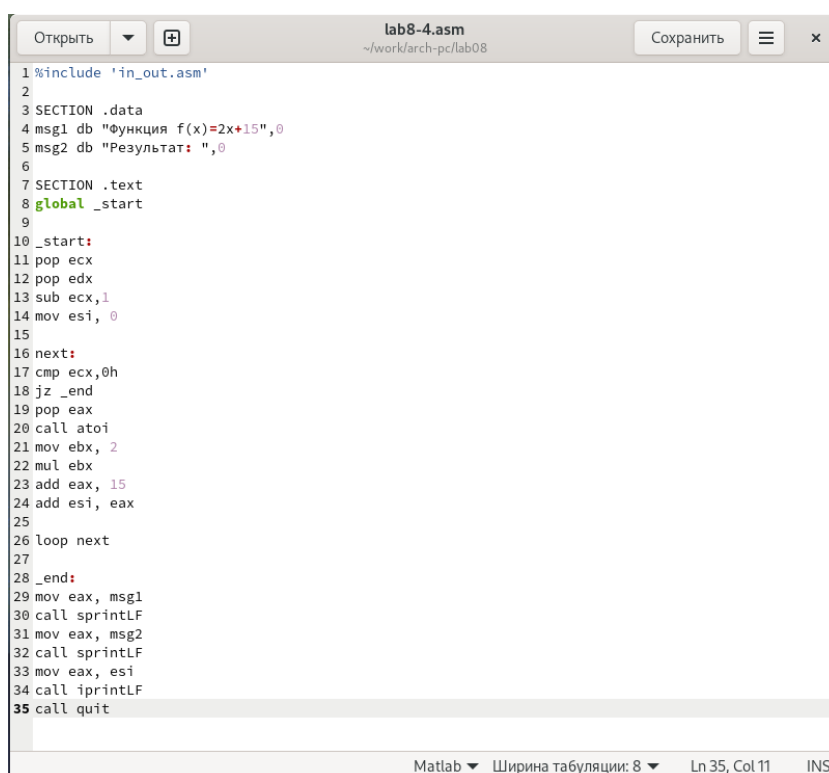
Рис. 3.23: Создание файла

Открываю файл `lab8-4.asm` в текстовом редакторе `gedit` через терминал (рис. 3.24).

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

Рис. 3.24: Открытие файла `lab8-4.asm` в текстовом редакторе `gedit`

Ввожу в файл `lab8-4.asm` функцию из варианта №1 (рис. 3.25).



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1 db "Функция f(x)=2x+15",0
5 msg2 db "Результат: ",0
6
7 SECTION .text
8 global _start
9
10 _start:
11 pop ecx
12 pop edx
13 sub ecx,1
14 mov esi, 0
15
16 next:
17 cmp ecx,0h
18 jz _end
19 pop eax
20 call atoi
21 mov ebx, 2
22 mul ebx
23 add eax, 15
24 add esi, eax
25
26 loop next
27
28 _end:
29 mov eax, msg1
30 call sprintLF
31 mov eax, msg2
32 call sprintLF
33 mov eax, esi
34 call iprintLF
35 call quit
```

Рис. 3.25: Редактирование файла

Создаю исполняемый файл и запускаю его. Проверяю работу исполняемого файла, указав аргументы: 1 2 3 4 5 (рис. 3.26) и 1 3 5 (рис. 3.27). Убеждаюсь в том, что программа работает корректно.

```

vkmal@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
vkmal@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
vkmal@vbox:~/work/arch-pc/lab08$ ./lab8-4 1 2 3 4 5
Функция f(x)=2x+15
Результат:
105

```

Рис. 3.26: Запуск исполняемого файла

```

vkmal@vbox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
vkmal@vbox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
vkmal@vbox:~/work/arch-pc/lab08$ ./lab8-4 1 3 5
Функция f(x)=2x+15
Результат:
63

```

Рис. 3.27: Запуск исполняемого файла

Листинг программы:

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
msg1 db "Функция f(x)=2x+15",0
```

```
msg2 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
mul ebx
add eax, 15
add esi, eax
```

```
loop next
```

```
_end:
mov eax, msg1
call sprintf
mov eax, msg2
call sprintf
mov eax, esi
call iprintf
call quit
```

4 Выводы

Я приобрела навыки написания программ с использованием циклов и обработки аргументов командной строки.