

Отчёт по лабораторной работе №8

Дисциплина: архитектура компьютера

Лобанов Владислав Олегович

Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Выполнение лабораторной работы	8
Реализация циклов в NASM	8
Обработка аргументов командной строки	13
Задание для самостоятельной работы (Вариант 20)	17
Выводы	19
Список литературы	20

Список иллюстраций

1	Создание каталога и файла в ней, редактирование файла	9
2	Исполнение программы	10
3	Изменение текста программы	11
4	Исполнение программы	11
5	Изменение программы	12
6	Исполнение программы	13
7	Создание и редактирование файла	13
8	Запуск исполняемого файла	14
9	Создание файла и его редактирование	14
10	Запуск исполняемого файла	15
11	Создание файла и его редактирование	15
12	Запуск исполняемого файла	15
13	Создание и редактирование файла	17
14	Запуск исполняемого файла	18

Список таблиц

Цель работы

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

Задание

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

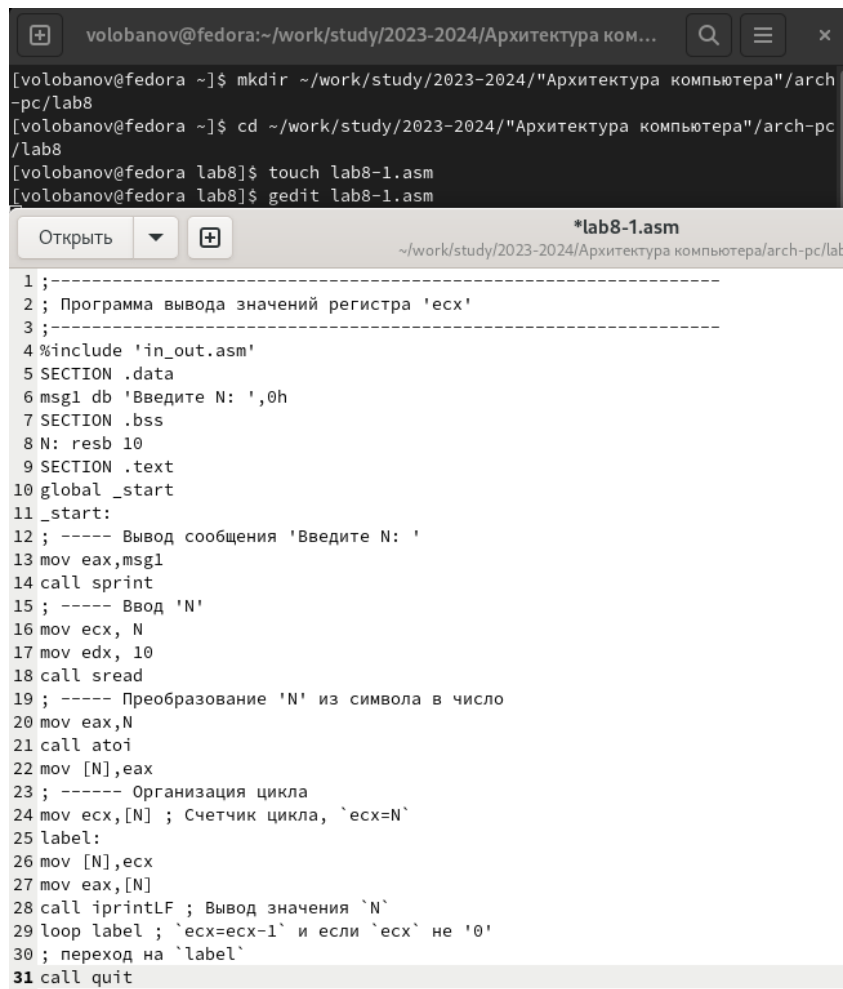
Теоретическое введение

Цикл в программировании — это управляющая конструкция, которая заставляет какой-то блок кода выполняться несколько раз.

Выполнение лабораторной работы

Реализация циклов в NASM

Создаю каталог для программ для лабораторной работы №8, перехожу в него и создаю файл lab8-1.asm, в который ввожу программу вывода значений регистра ехх (рис. @fig:001).



The image shows a terminal window and a text editor window. The terminal window displays the following commands and their outputs:

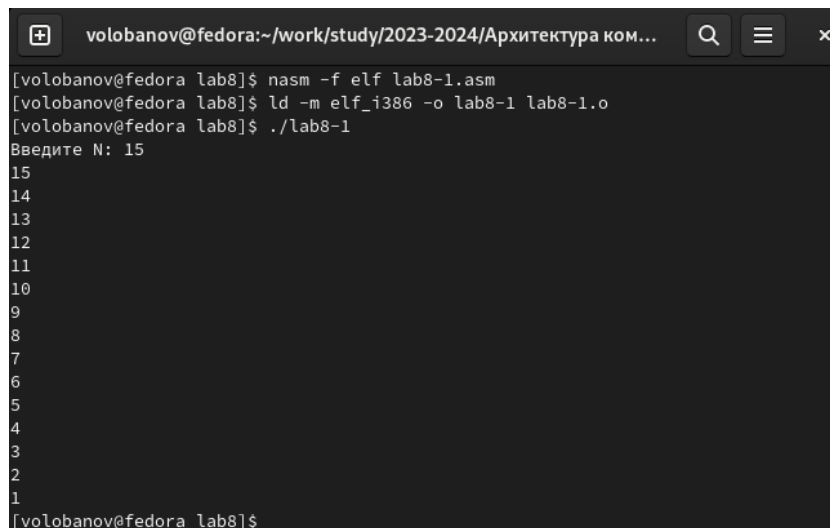
```
[volobanov@fedora ~]$ mkdir ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab8
[volobanov@fedora ~]$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab8
[volobanov@fedora lab8]$ touch lab8-1.asm
[volobanov@fedora lab8]$ gedit lab8-1.asm
```

The text editor window shows the content of the file `lab8-1.asm`:

```
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 mov [N],ecx
27 mov eax,[N]
28 call iprintLF ; Вывод значения `N`
29 loop label ; `ecx=ecx-1` и если `ecx` не `0`
30 ; переход на `label`
31 call quit
```

Рис. 1: Создание каталога и файла в ней, редактирование файла

Создаю исполняемый файл и запускаю его (рис. @fig:002). Программа отработала корректно.

A terminal window with a dark background and light text. The title bar shows the user 'volobanov@fedora' and the path '~/work/study/2023-2024/Архитектура ком...'. The terminal contains the following text:

```
[volobanov@fedora lab8]$ nasm -f elf lab8-1.asm
[volobanov@fedora lab8]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[volobanov@fedora lab8]$ ./lab8-1
Введите N: 15
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
[volobanov@fedora lab8]$
```

Рис. 2: Исполнение программы

Изменяю текст программы добавив изменение значения регистра есх в цикле (рис. @fig:003).

```
[volobanov@fedora lab8]$ gedit lab8-1.asm

lab8-1.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab8

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 ; 'ecx=ecx-1' и если 'ecx' не '0'
31 ; переход на 'label'
32 call quit
```

Рис. 3: Изменение текста программы

Создаю исполняемый файл и запускаю его (рис. @fig:004). Число проходов цикла равно $N/2$.

```
volobanov@fedora:~/work/study/2023-2024/Архитектура ком...
[volobanov@fedora lab8]$ gedit lab8-1.asm
[volobanov@fedora lab8]$ nasm -f elf lab8-1.asm
[volobanov@fedora lab8]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[volobanov@fedora lab8]$ ./lab8-1
Введите N: 10
9
7
5
3
1
[volobanov@fedora lab8]$
```

Рис. 4: Исполнение программы

Вношу изменения в текст программы добавив команды `push` и `pop` для сохранения значения счётчика `loop`(рис. @fig:005).

```
[volobanov@fedora lab8]$ gedit lab8-1.asm

Открыть ▼ + *lab8-1.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab8-1.asm

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
27 sub ecx,1 ; 'ecx=ecx-1'
28 mov [N],ecx
29 mov eax,[N]
30 call iprintLF
31 pop ecx
32 loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
33 ; переход на 'label'
34 call quit
```

Рис. 5: Изменение программы

Создаю исполняемый файл и проверяю его работу (рис. @fig:006). Число проходов соответствует значению N введённому с клавиатуры.

```
[volobanov@fedora lab8]$ nasm -f elf lab8-1.asm
[volobanov@fedora lab8]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[volobanov@fedora lab8]$ ./lab8-1
Введите N: 15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
[volobanov@fedora lab8]$
```

Рис. 6: Исполнение программы

Обработка аргументов командной строки

Создаю файл lab8-2.asm и ввожу в него программу обработки аргументов командной строки (рис. @fig:007).

```
[volobanov@fedora lab8]$ gedit lab8-2.asm
lab8-2.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab8-2.asm
1 %include 'in_out.asm'
2 SECTION .text
3 global _start
4 _start:
5 pop ecx ; Извлекаем из стека в `ecx` количество
6 ; аргументов (первое значение в стеке)
7 pop edx ; Извлекаем из стека в `edx` имя программы
8 ; (второе значение в стеке)
9 sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество
10 ; аргументов без названия программы)
11 next:
12 cmp ecx, 0 ; проверяем, есть ли еще аргументы
13 jz _end ; если аргументов нет выходим из цикла
14 ; (переход на метку `_end`)
15 pop eax ; иначе извлекаем аргумент из стека
16 call sprintf ; вызываем функцию печати
17 loop next ; переход к обработке следующего
18 ; аргумента (переход на метку `next`)
19 _end:
20 call quit
```

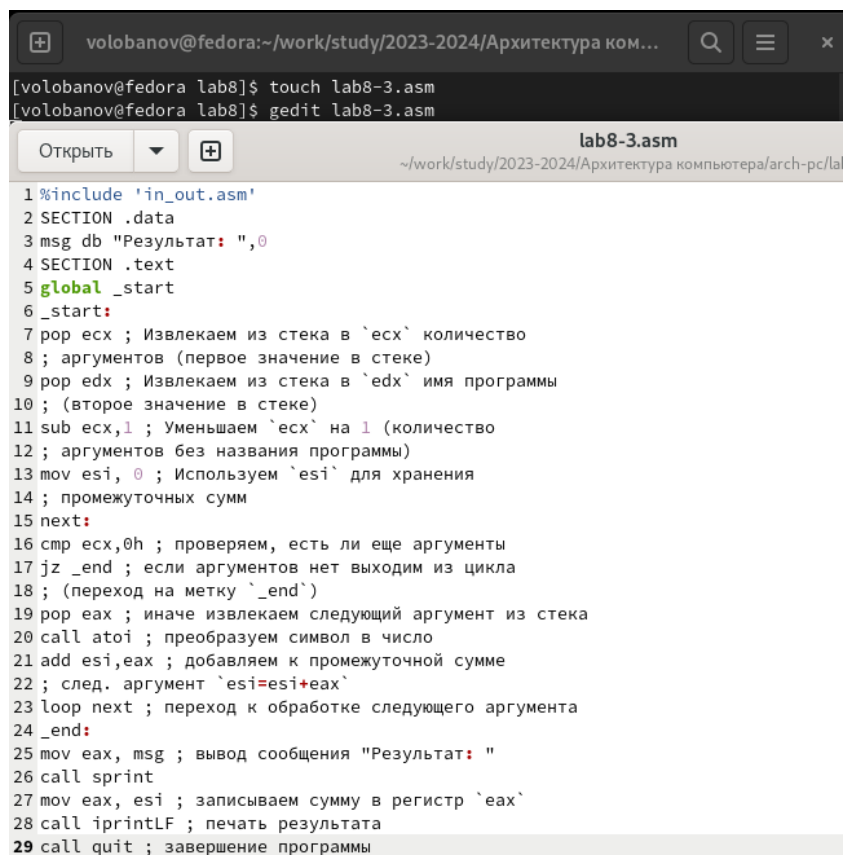
Рис. 7: Создание и редактирование файла

Создаю исполняемый файл. Запускаю исполняемый файл (рис. @fig:008). Программой были обработаны все аргументы.

```
[volobanov@fedora lab8]$ nasm -f elf lab8-2.asm
[volobanov@fedora lab8]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[volobanov@fedora lab8]$ ./lab8-2 arg1 arg2 arg3
arg1
arg2
arg3
```

Рис. 8: Запуск исполняемого файла

Создаю файл lab8-3.asm и ввожу в него программу вычисления суммы аргументов командной строки (рис. @fig:009).



```
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 iprintLF ; печать результата
29 call quit ; завершение программы
```

Рис. 9: Создание файла и его редактирование

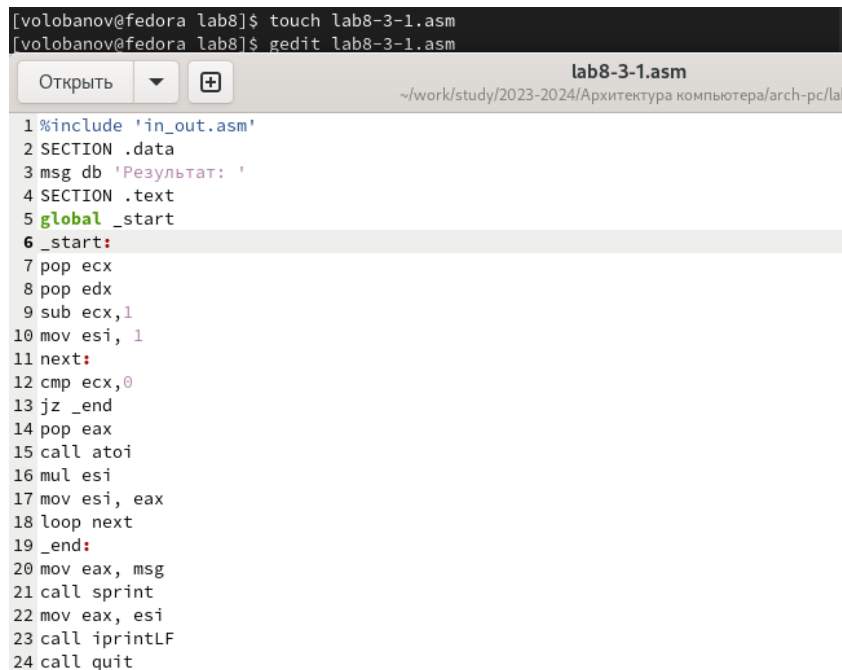
Создаю исполняемый файл и запускаю его, указав аргументы 12, 13, 7, 10, 5 (рис. @fig:010). Программа отработала корректно.

```
[volobanov@fedora lab8]$ nasm -f elf lab8-3.asm
[volobanov@fedora lab8]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[volobanov@fedora lab8]$ ./lab8-3 12 13 7 10 5
Результат: 47
[volobanov@fedora lab8]$
```

Рис. 10: Запуск исполняемого файла

Создаю файл lab8-3-1.asm и ввожу в него программу вычисления произведения аргументов командной строки (рис. @fig:011).

```
[volobanov@fedora lab8]$ touch lab8-3-1.asm
[volobanov@fedora lab8]$ gedit lab8-3-1.asm
```



```
lab8-3-1.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/la

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

Рис. 11: Создание файла и его редактирование

Создаю исполняемый файл и запускаю его, указав аргументы 4 5 6 7 8 9 (рис. @fig:012). Программа отработала корректно.

```
[volobanov@fedora lab8]$ nasm -f elf lab8-3-1.asm
[volobanov@fedora lab8]$ ld -m elf_i386 -o lab8-3-1 lab8-3-1.o
[volobanov@fedora lab8]$ ./lab8-3-1 4 5 6 7 8 9
Результат: 60480
[volobanov@fedora lab8]$
```

Рис. 12: Запуск исполняемого файла

Текст программы из файла lab8-3-1.asm:

```

#include "in_out.asm"

SECTION .data
msg db 'результат: '

SECTION .text
GLOBAL _start

_start:
pop ecx
pop edx
sub ecx,1
mov esi,1

next:
cmp ecx,0
jz _end

pop eax
call atoi
mul esi
mov esi, eax

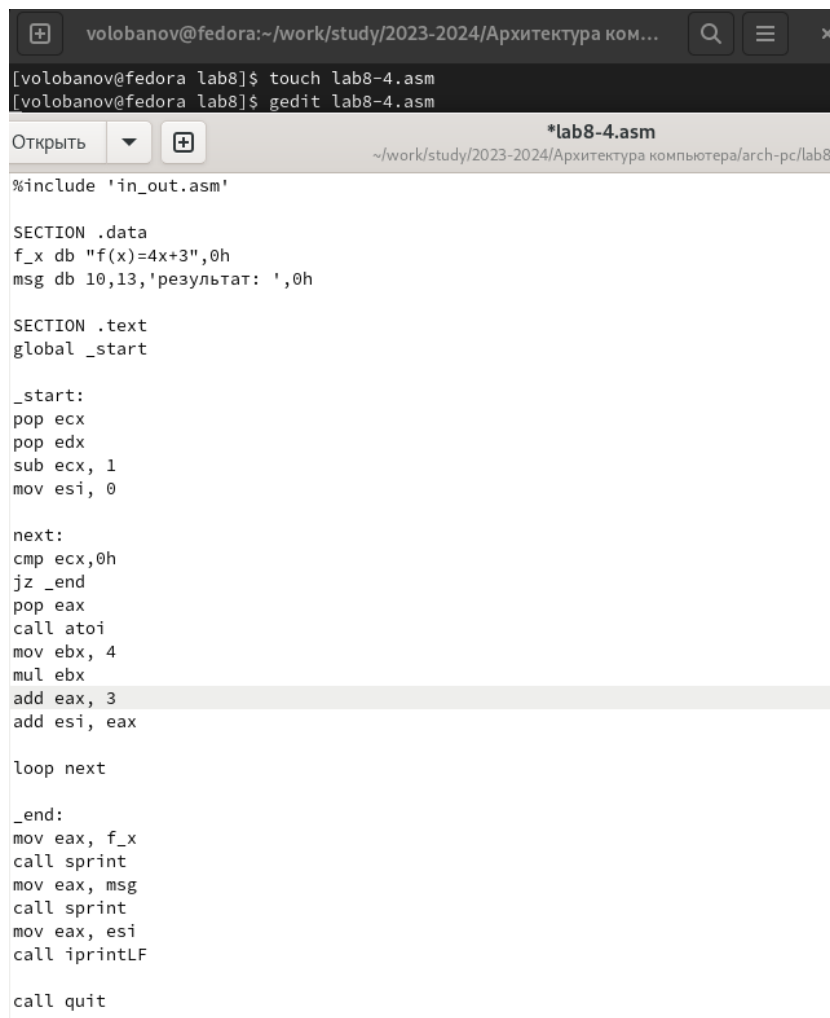
loop next

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

```


Задание для самостоятельной работы (Вариант 20)

Создаю файл lab8-4.asm и ввожу текст программы для вычисления суммы значений функции $f(x)=3(10+x)$ (рис. @fig:013).



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

```
volobanov@fedora:~/work/study/2023-2024/Архитектура ком...  
[volobanov@fedora lab8]$ touch lab8-4.asm  
[volobanov@fedora lab8]$ gedit lab8-4.asm
```

Below the terminal is a text editor window titled '*lab8-4.asm' showing the following assembly code:

```
%include 'in_out.asm'  
  
SECTION .data  
f_x db "f(x)=4x+3",0h  
msg db 10,13,'результат: ',0h  
  
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, 4  
mul ebx  
add eax, 3  
add esi, eax  
  
loop next  
  
_end:  
mov eax, f_x  
call sprint  
mov eax, msg  
call sprint  
mov eax, esi  
call iprintLF  
  
call quit
```

Рис. 13: Создание и редактирование файла

Текст программы из файла lab8-3-1.asm:

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

```
SECTION .data f_x db "f(x)=4x+3",0h msg db 10,13,'результат:',0h
```

```
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, 4
mul ebx add eax, 3 add esi, eax
loop next
_end: mov eax, f_x call sprint mov eax, msg call sprint mov eax, esi call iprintLF
call quit
```

Создаю исполняемый файл и запускаю исполняемый файл с разными аргументами (рис. @fig:014). Программа отработала корректно.

```
[volobanov@fedora lab8]$ nasm -f elf lab8-4.asm
[volobanov@fedora lab8]$ ld -m elf_i386 -o lab8-4 lab8-4.o
[volobanov@fedora lab8]$ ./lab8-4
f(x)=4x+3
результат: 0
[volobanov@fedora lab8]$ ./lab8-4 100 200 300
f(x)=4x+3
результат: 2409
[volobanov@fedora lab8]$
```

Рис. 14: Запуск исполняемого файла

Выводы

Были получены навыки по организации циклов и работе со стеком на языке NASM.

Список литературы

1. Лабораторная работа №8