

Шаблон отчёта по лабораторной работе №8

Дисциплина: архитектура компьютера

Чашемова Гульназик

Содержание

1 Цель работы	1
2 Задание	1
3 Теоретическое введение	2
4 Выполнение лабораторной работы	2
4.1 Реализация циклов в NASM	2
4.2 Обработка аргументов командной строки	6
4.3 Самостоятельная работа	9
4.3.1 Напишите программу, которая находит сумму значений функции $f(x)$ для $x=x_1, x_2, \dots, x_n$, т.е. программа должна выводить значение $f(x_1)+f(x_2)+\dots+f(x_n)$	9
5 Выводы	10
6 Список литературы	10

1 Цель работы

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

2 Задание

1. Реализация переходов в NASM
2. Обработка аргументов командной строки

3 Теоретическое введение

Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды.

Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нем выделяется память для локальных переменных и могут временно храниться значения регистров.

Стек имеет вершину, адрес последнего добавленного элемента, который хранится в регистре esp (указатель стека). Противоположный конец стека называется дном. Значение, помещенное в стек последним, извлекается первым. При помещении значения в стек указатель стека уменьшается, а при извлечении — увеличивается.

Для стека существуют две основные операции:

- добавление элемента в вершину стека (push);
- извлечение элемента из вершины стека (pop).

4 Выполнение лабораторной работы

4.1 Реализация переходов в NASM

Для начала я создала каталог для программ Лабораторной работы, потом перешла в него и создала файл lab8-1.asm (рис. 1).

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs$ cd lab08  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab08$ touch lab8-1.asm  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab08$ ls  
lab8-1.asm presentation_report  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab08$
```

Рис. 1: Создания каталога и файла

Потом зашла на МС и через него скопировала файл `in_out.asm` в созданный каталог

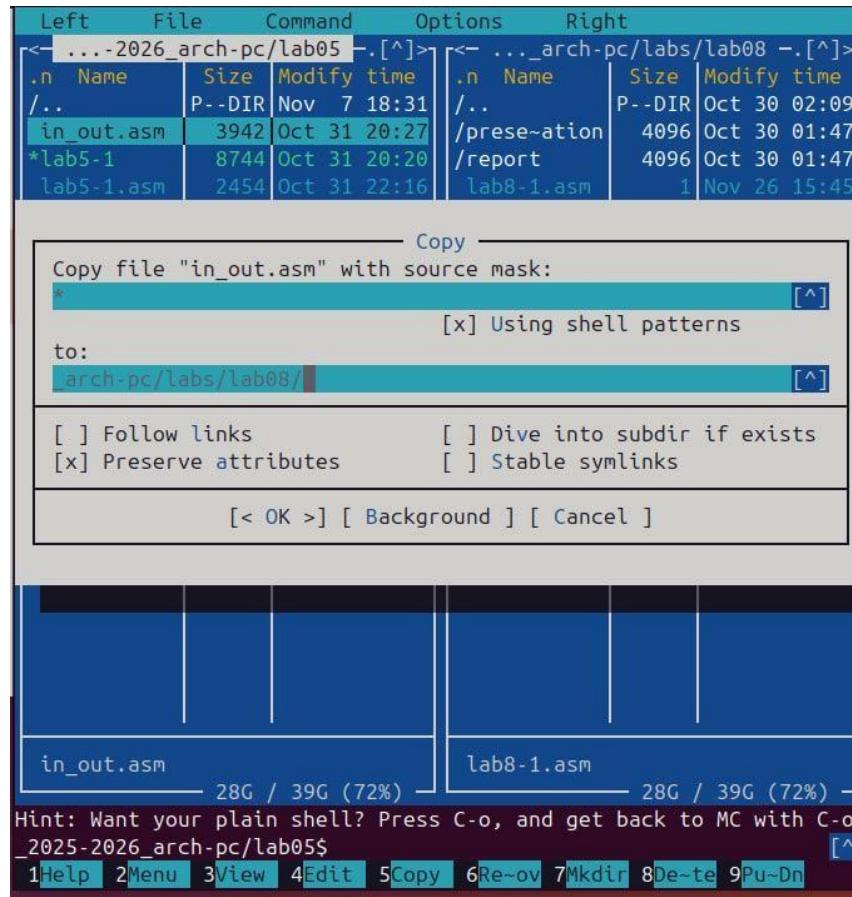


Рис. 2: Скопирования файла `in_out.asm` в нужный каталог

После этого я открыла созданной мною файл и ввела туда программу для вывода значений регистра `esx`

```
...а компьютера/study_2025-2026_arch-pc/labs/lab08/lab8-1.asm *
;-----
; 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
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, `ecx=N`
label:
mov [N],ecx
mov eax,[N]
call iprintLF; Вывод значения `N`  

^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify
```

Рис. 3: Программа вывода значений регистра ecx

Потом я создала исполняемый файл и запустила его

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_20
25-2026_arch-pc/labs/lab08$ nasm -f elf lab8-1.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_20
25-2026_arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_20
25-2026_arch-pc/labs/lab08$ ./lab8-1
Введите N: 5
5
4
3
2
1
```

Рис. 4: Создания исполняемого файла

Я изменила текст файла добавив изменение значение регистра ecx в цикле. Данный пример показывает, что использование регистра ecx в теле цикла loop может привести к некорректной работе программы

```
...тупа компьютера/study_2025-2026_arch-pc/labs/lab08/lab8-1.asm
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
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, `ecx=N`
label:
sub ecx,1 ; `ecx=ecx-1`
mov [N],ecx
mov eax,[N]
call iprintf ; Вывод значения 'N'
loop label ; `ecx=ecx-1` и если `ecx` не '0'
; переход на 'label'
call quit
[ ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute
^X Exit      ^R Read File   ^\ Replace     ^U Paste      ^J Justify
```

Рис. 5: Изменения текста файла

Создала используемый файл и запустила его еще раз но уже измененный.

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2
25-2026_arch-pc/labs/lab08$ nasm -f elf lab8-1.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2
25-2026_arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2
25-2026_arch-pc/labs/lab08$ ./lab8-1
Введите N: 6
5
3
1
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2
25-2026_arch-pc/labs/lab08$ [ ]
```

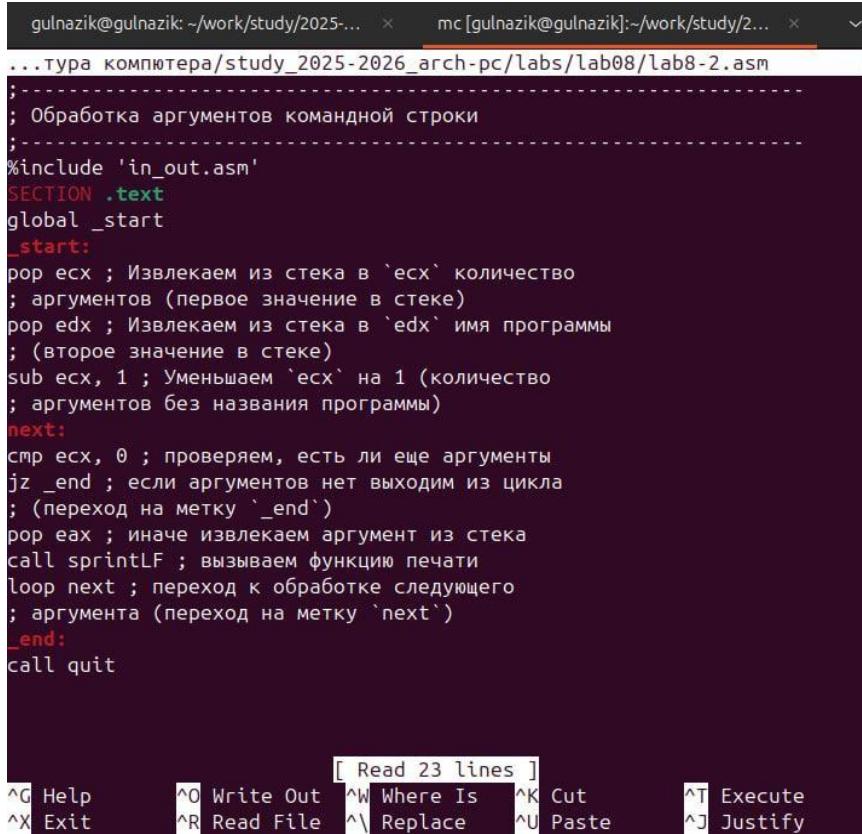
Рис. 6: Создания(изменённого) используемого файла

Потом я создала новый файл в том же каталоге lab8-2.asm

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2
25-2026_arch-pc/labs/lab08$ touch lab8-2.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2
25-2026_arch-pc/labs/lab08$ gedit lab8-2.asm
[ ]
```

Рис. 7: Создания файла lab8-2.asm

4.2 После создания я открыла файл и ввела туда программу, которая выводит на экран аргументы командной строки

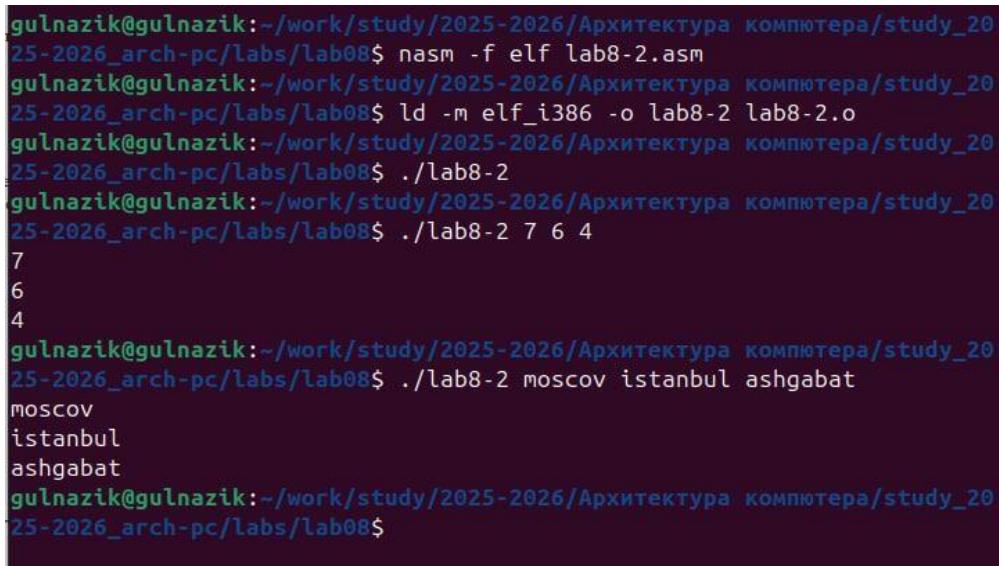


```
gulnazik@gulnazik:~/work/study/2025-2026_арх-pc/labs/lab08/lab8-2.asm
...тура компьютера/study_2025-2026_арх-pc/labs/lab08/lab8-2.asm
;-----
; Обработка аргументов командной строки
;-----
%include 'in_out.asm'
SECTION .text
global _start
_start:
    pop ecx ; Извлекаем из стека в `ecx` количество
    ; 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 sprintLF ; вызываем функцию печати
    loop next ; переход к обработке следующего
    ; аргумента (переход на метку `next`)
_end:
    call quit

[ Read 23 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\| Replace   ^U Paste      ^J Justify
```

Рис. 8: Программа, которая выводит на экран аргументы командной строки

Потом создала исполняемый файл и запустила его. И еще я проверила его работу



```
gulnazik@gulnazik:~/work/study/2025-2026_Архитектура компьютера/study_2025-2026_arch-pc/labs/lab08$ nasm -f elf lab8-2.asm
gulnazik@gulnazik:~/work/study/2025-2026_Архитектура компьютера/study_2025-2026_arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
gulnazik@gulnazik:~/work/study/2025-2026_Архитектура компьютера/study_2025-2026_arch-pc/labs/lab08$ ./lab8-2
gulnazik@gulnazik:~/work/study/2025-2026_Архитектура компьютера/study_2025-2026_arch-pc/labs/lab08$ ./lab8-2 7 6 4
7
6
4
gulnazik@gulnazik:~/work/study/2025-2026_Архитектура компьютера/study_2025-2026_arch-pc/labs/lab08$ ./lab8-2 moscov istanbul ashgabat
moscov
istanbul
ashgabat
gulnazik@gulnazik:~/work/study/2025-2026_Архитектура компьютера/study_2025-2026_arch-pc/labs/lab08$
```

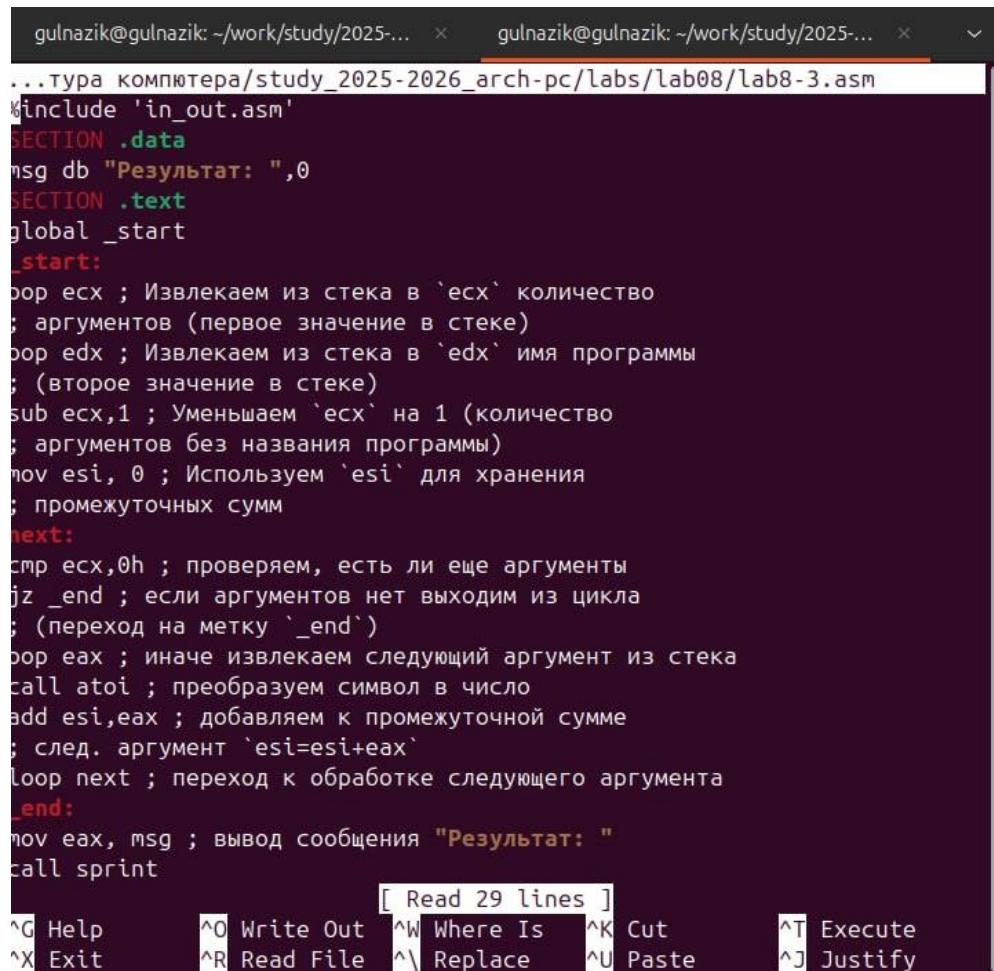
Рис. 9: Создания исполняемого файла lab8-2.asm

Потом я создала новый файл в том же каталоге lab8-3.asm

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arch-pc/labs/lab08$ touch lab8-3.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arch-pc/labs/lab08$ gedit lab8-3.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arch-pc/labs/lab08$
```

Рис. 10: Создания файла lab8-3.asm

После создания я открыла файл и ввела туда программу, которая выводит на экран вычисления суммы аргументов командной строки



```
gulnazik@gulnazik: ~/work/study/2025-2026_arch-pc/labs/lab08/lab8-3.asm
...тута компьютера/study_2025-2026_arch-pc/labs/lab08/lab8-3.asm
;include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    ;ори ecx ; Извлекаем из стека в `ecx` количество
    ; аргументов (первое значение в стеке)
    ;ори 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:
    cmpr ecx,0h ; проверяем, есть ли еще аргументы
    jz _end ; если аргументов нет выходим из цикла
    ; (переход на метку `_end`)
    ;ори eax ; иначе извлекаем следующий аргумент из стека
    call atoi ; преобразуем символ в число
    add esi,eax ; добавляем к промежуточной сумме
    ; след. аргумент `esi=esi+eax`
    loop next ; переход к обработке следующего аргумента
_end:
    mov eax, msg ; вывод сообщения "Результат: "
    call sprint
[ Read 29 lines ]
^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify
```

Рис. 11: Программа, которая выводит на экран вычисления суммы аргументов командной строки

Потом создала исполняемый файл и запустила его и еще я проверила его работу

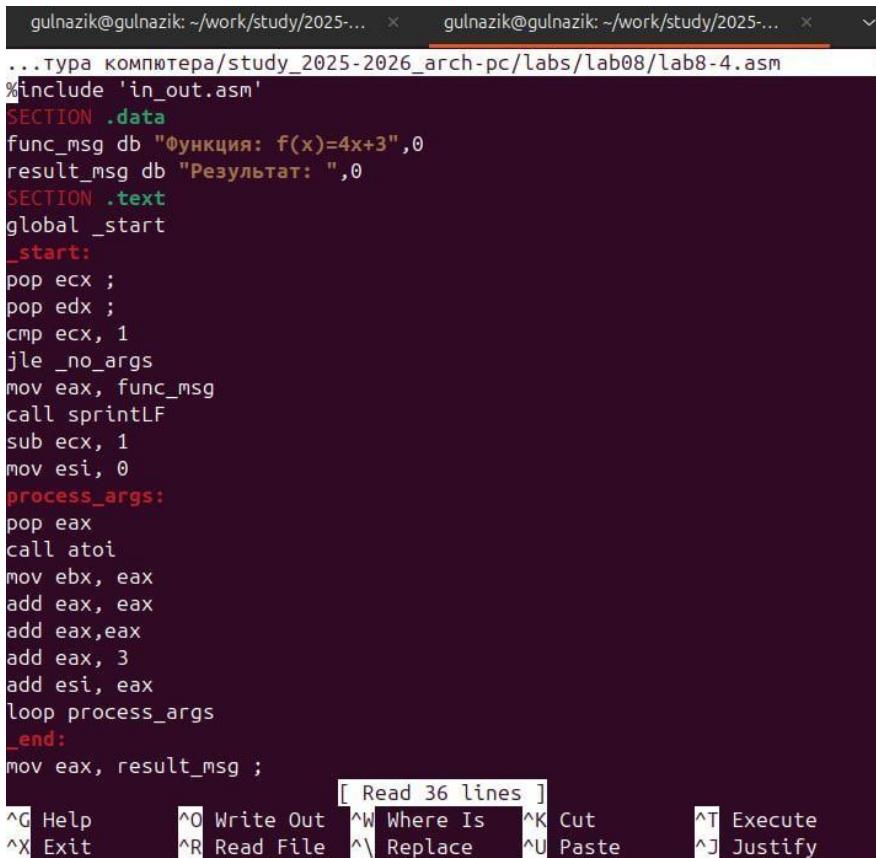
```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_20  
25-2026_arch-pc/labs/lab08$ nasm -f elf lab8-3.asm  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_20  
25-2026_arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_20  
25-2026_arch-pc/labs/lab08$ ./lab8-3  
Результат: 0  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_20  
25-2026_arch-pc/labs/lab08$ ./lab8-3 10 8 2 23  
Результат: 43  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_20  
25-2026_arch-pc/labs/lab08$ ./lab8-3 5 45 71 68  
Результат: 189  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_20  
25-2026_arch-pc/labs/lab08$
```

Рис. 12: Создания исполняемого файла lab8-3.asm

4.3 Самостоятельная работа

4.3.1. Напишите программу, которая находит сумму значений функции $f(x)$ для $x = x_1, x_2, \dots, x_n$, т.е. программа должна выводить значение $f(x_1) + f(x_2) + \dots + f(x_n)$.

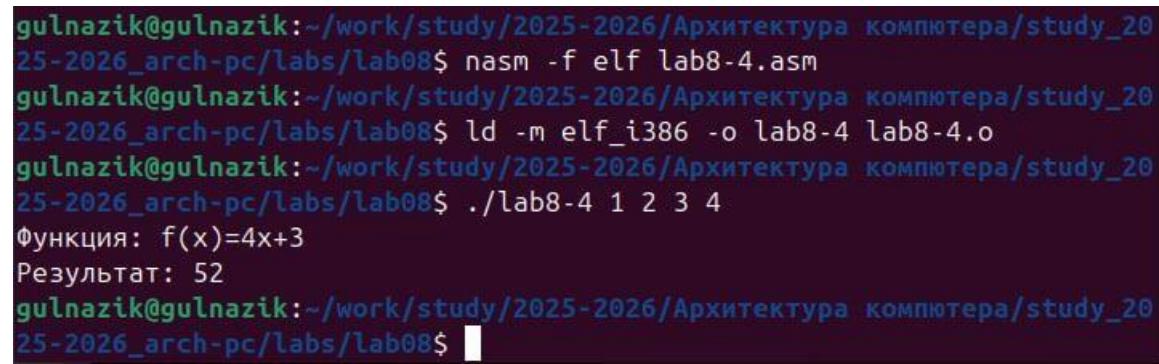
Для начала я создала файл и в него я написала программу.



```
gulnazik@gulnazik: ~/work/study/2025-2026_арх-pc/labs/lab08/lab8-4.asm
...тута компьютера/study_2025-2026_арх-pc/labs/lab08/lab8-4.asm
%include 'in_out.asm'
SECTION .data
func_msg db "Функция: f(x)=4x+3",0
result_msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx ;
    pop edx ;
    cmp ecx, 1
    jle _no_args
    mov eax, func_msg
    call sprintLF
    sub ecx, 1
    mov esi, 0
process_args:
    pop eax
    call atoi
    mov ebx, eax
    add eax, eax
    add eax, eax
    add eax, 3
    add esi, eax
    loop process_args
_end:
    mov eax, result_msg ;
[ Read 36 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify
```

Рис. 13: Внесения программы в файл lab8-4.asm

После этого я создала исполняемый файл и запустила его. Потом я написала цифры которые были таблице



```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_арх-pc/labs/lab08$ nasm -f elf lab8-4.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_арх-pc/labs/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_арх-pc/labs/lab08$ ./lab8-4 1 2 3 4
Функция: f(x)=4x+3
Результат: 52
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_арх-pc/labs/lab08$
```

Рис. 14: Создания исполняемого файла lab8-4.asm.

Все готово!

5 Выводы

В ходе лабораторной работы были приобретены навыки работы с циклами в ассемблере NASM, освоены методы обработки аргументов командной строки и работы со стеком для сохранения значений регистров.

Список литературы

(<https://esystem.rudn.ru>) Архитектура компьютеров, Лабораторная работа №8