

# ***Отчёт по лабораторной работе***

***Лабораторная №8***

Полина Витальевна Барабаш

# ***Содержание***

<b><i>1</i></b>	<b><i>Цель работы</i></b>	<b><i>4</i></b>
<b><i>2</i></b>	<b><i>Выполнение работы</i></b>	<b><i>5</i></b>
<b><i>3</i></b>	<b><i>Выполнение самостоятельной работы</i></b>	<b><i>10</i></b>
<b><i>4</i></b>	<b><i>Выводы</i></b>	<b><i>12</i></b>

## ***Список иллюстраций***

2.1	Создание каталога, файла, работа программы . . . . .	5
2.2	Запуск программы с изменением значения регистра esx в цикле .	6
2.3	Изменение программы с использованием push и pop и результат её работы . . . . .	7
2.4	Создание файла и запуск программы, обрабатывающей введенные аргументы . . . . .	8
2.5	Создание файла, работа программы, выполняющей сложение введенных чисел . . . . .	8
2.6	Работа изменённой программы для вычисления произведения чисел	9
3.1	Работа программы по вычислению значения функции . . . . .	10

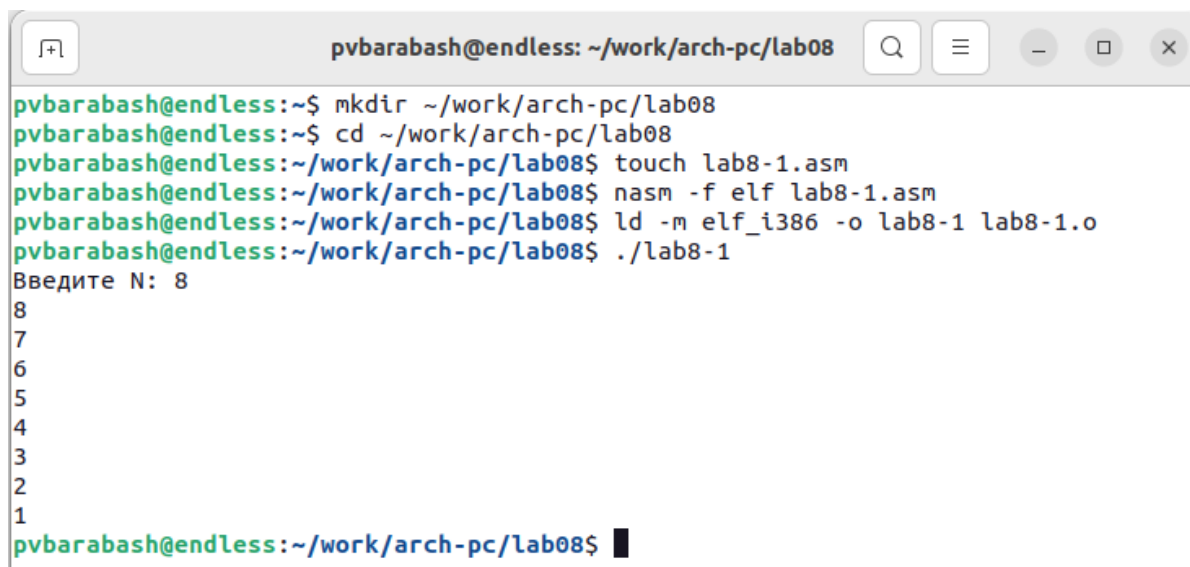
# ***1 Цель работы***

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

## 2 Выполнение работы

**Задание №1.** Создайте каталог для программ лабораторной работы № 8, перейдите в него и создайте файл lab8-1.asm. Внимательно изучите текст программы (Листинг 8.1). Введите в файл lab8-1.asm текст программы из листинга 8.1. Создайте исполняемый файл и проверьте его работу.

Я создала каталог для программ лабораторной работы № 8 с помощью команды `mkdir ~/work/arch-pc/lab08`. Затем я перешла в него с помощью команды `cd` и создала файл `lab8-1.asm` с помощью `touch`. Я внимательно изучила текст программы и ввела его в созданный файл. После этого создала исполняемый файл и проверила его работу (рис. 2.1).



```
pvbarabash@endless: ~/work/arch-pc/lab08
pvbarabash@endless:~$ mkdir ~/work/arch-pc/lab08
pvbarabash@endless:~$ cd ~/work/arch-pc/lab08
pvbarabash@endless:~/work/arch-pc/lab08$ touch lab8-1.asm
pvbarabash@endless:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
pvbarabash@endless:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
pvbarabash@endless:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 8
8
7
6
5
4
3
2
1
pvbarabash@endless:~/work/arch-pc/lab08$
```

Рис. 2.1: Создание каталога, файла, работа программы

В теле цикла при каждом заходе печатается значение `ecx`, то есть изначальное

N и результаты последовательного вычитания единицы, которое происходит в loop. Программа работает корректно.

**Задание №2.** Измените текст программы, добавив изменение значение регистра esx в цикле. Создайте исполняемый файл и проверьте его работу. Какие значения принимает регистр esx в цикле? Соответствует ли число проходов цикла значению N введенному с клавиатуры?

Я изменила текст программы, добавив вычитание из esx в цикле единицы. Создала исполняемый файл и проверила его работу (рис. 2.2).



```
pvbarabash@endless: ~/work/arch-pc/lab08
pvbarabash@endless:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
pvbarabash@endless:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
pvbarabash@endless:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 8
7
5
3
1
pvbarabash@endless:~/work/arch-pc/lab08$
```

Рис. 2.2: Запуск программы с изменением значения регистра esx в цикле

Регистр esx в цикле принимает значения 7, 5, 3, 1. Число проходов не соответствует значению N, введенному с клавиатуры (8).

**Задание №3.** Внесите изменения в текст программы, добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop. Создайте исполняемый файл и проверьте его работу. Соответствует ли в данном случае число проходов цикла значению N, введенному с клавиатуры?

Я изменила текст программы, добавив команды push и pop для сохранения значения счетчика цикла loop. Я создала исполняемый файл и проверила его работу (рис. 2.3).



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

Рис. 2.3: Изменение программы с использованием push и pop и результат её работы

В данном случае число проходов цикла соответствует введенному значению N.

**Задание №4.** Внимательно изучите текст программы (Листинг 8.2). Создайте файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и введите в него текст программы из листинга 8.2. Создайте исполняемый файл и запустите его, указав аргументы:

user@dk4n31:~\$ ./lab8-2 аргумент1 аргумент 2 ‘аргумент 3’

Сколько аргументов было обработано программой?

Я внимательно изучила текст программы. Создала файл lab8-2.asm и ввела в него текст программы. Затем создала исполняемый файл и запустила его (рис. 2.4).



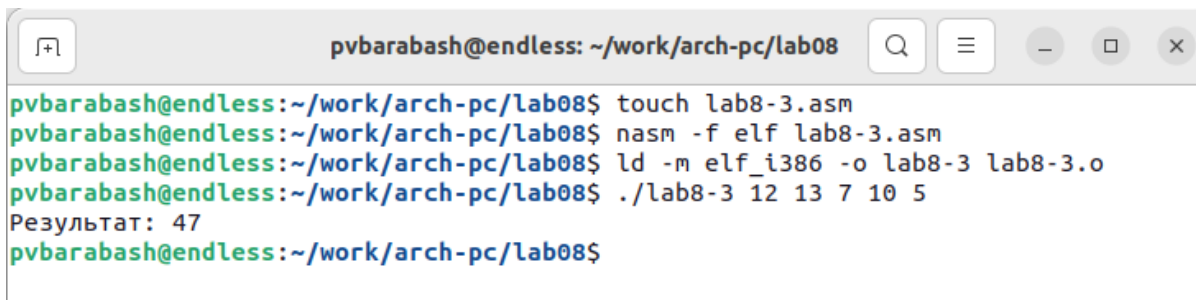
```
pvbarabash@endless: ~/work/arch-pc/lab08
pvbarabash@endless:~/work/arch-pc/lab08$ touch lab8-2.asm
pvbarabash@endless:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
pvbarabash@endless:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
pvbarabash@endless:~/work/arch-pc/lab08$ ./lab8-2 аргумент1 аргумент 2 'аргумент
3'
аргумент1
аргумент
2
аргумент 3
pvbarabash@endless:~/work/arch-pc/lab08$
```

Рис. 2.4: Создание файла и запуск программы, обрабатывающей введенные аргументы

Программой было обработано 4 аргумента, что является ошибкой, ведь аргументов всего 3. Так происходит потому, что “аргумент 2” не задан как единое целое и пробел воспринимается, как разделитель.

**Задание №5.** Рассмотрите еще один пример программы, которая выводит сумму чисел, которые передаются в программу как аргументы. Создайте файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 и введите в него текст программы из листинга 8.3. Создайте исполняемый файл и запустите его, указав аргументы.

Я внимательно изучила программу, создала файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 с помощью touch. После того, как я ввела текст программы в файл, я создала исполняемый файл и проверила его работу на данном в руководстве примере. Программа работает верно (рис. 2.5).



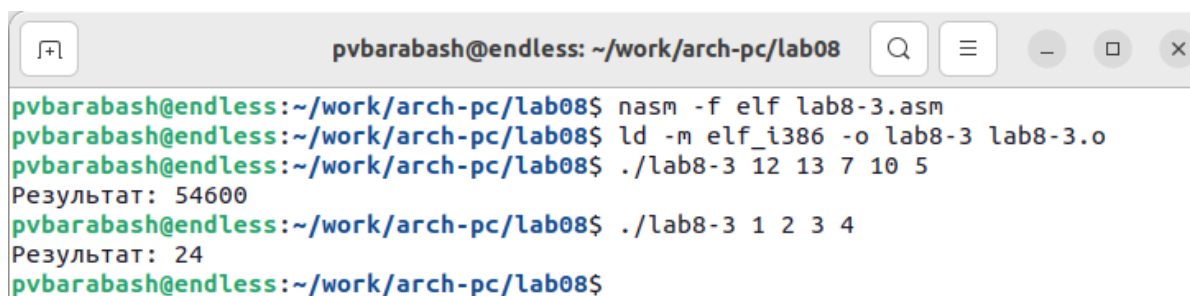
```
pvbarabash@endless: ~/work/arch-pc/lab08
pvbarabash@endless:~/work/arch-pc/lab08$ touch lab8-3.asm
pvbarabash@endless:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
pvbarabash@endless:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
pvbarabash@endless:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10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

Рис. 2.5: Создание файла, работа программы, выполняющей сложение введенных чисел



**Задание №6.** Измените текст программы из листинга 8.3 для вычисления произведения аргументов командной строки.

Я изменила текст программы для вычисления произведения аргументов командной строки. Создала исполняемый файл и проверила его работу (рис. 2.6).



```
pvbarabash@endless: ~/work/arch-pc/lab08
pvbarabash@endless:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
pvbarabash@endless:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
pvbarabash@endless:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 54600
pvbarabash@endless:~/work/arch-pc/lab08$ ./lab8-3 1 2 3 4
Результат: 24
pvbarabash@endless:~/work/arch-pc/lab08$
```

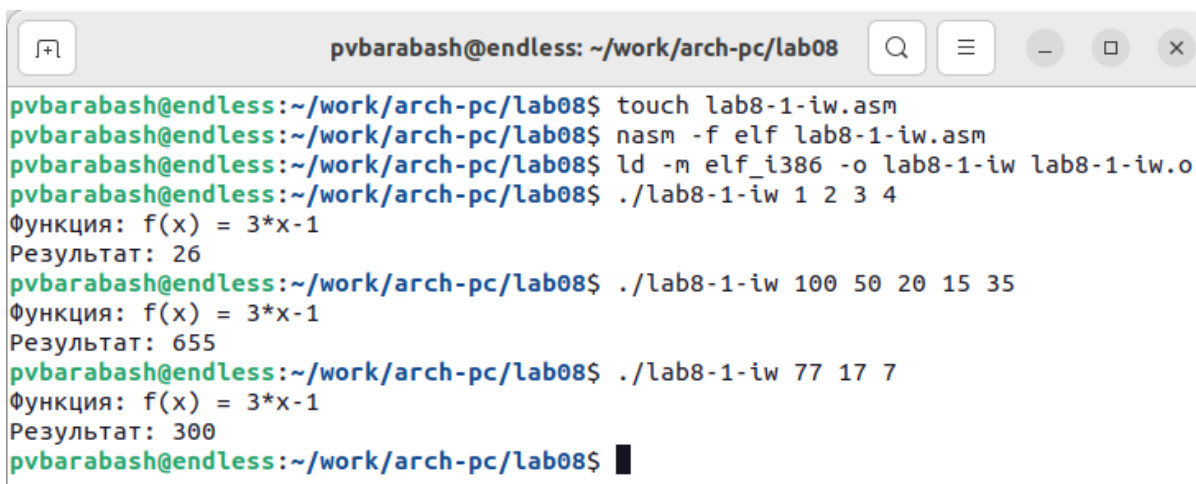
Рис. 2.6: Работа изменённой программы для вычисления произведения чисел

Проверив вручную, я убедилась, что программа работает верно. Изменённый код Вы найдете в прикреплённом файле lab8-3.asm.

### 3 Выполнение самостоятельной работы

**Задание №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  $f(x_1) + f(x_2) + \dots + f(x_n)$ . 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  $x = x_1, x_2, \dots, x_n$ .

Мой вариант второй. Следовательно мне необходимо написать программу вычисления функции  $3 \cdot x - 1$ . Я создала файл `lab8-1-iw.asm` и написала в нём программу. Я создала исполняемый файл и проверила работу программы на трёх наборах (рис. 3.1).



```
pvbarabash@endless: ~/work/arch-pc/lab08
pvbarabash@endless:~/work/arch-pc/lab08$ touch lab8-1-iw.asm
pvbarabash@endless:~/work/arch-pc/lab08$ nasm -f elf lab8-1-iw.asm
pvbarabash@endless:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1-iw lab8-1-iw.o
pvbarabash@endless:~/work/arch-pc/lab08$ ./lab8-1-iw 1 2 3 4
Функция: f(x) = 3*x-1
Результат: 26
pvbarabash@endless:~/work/arch-pc/lab08$ ./lab8-1-iw 100 50 20 15 35
Функция: f(x) = 3*x-1
Результат: 655
pvbarabash@endless:~/work/arch-pc/lab08$ ./lab8-1-iw 77 17 7
Функция: f(x) = 3*x-1
Результат: 300
pvbarabash@endless:~/work/arch-pc/lab08$
```

Рис. 3.1: Работа программы по вычислению значения функции

Проверим вычисления вручную.

$3 * 1 - 1 = 2; 3 * 2 - 1 = 5; 3 * 3 - 1 = 8; 3 * 4 - 1 = 11; 2 + 5 + 8 + 11 = 26$  (ответ верный)

$3 * 100 - 1 = 299; 3 * 50 - 1 = 149; 3 * 20 - 1 = 59; 3 * 15 - 1 = 44; 3 * 35 - 1 = 104; 299 + 149 + 59 + 44 + 104 = 655$  (ответ верный)

$3 * 77 - 1 = 230; 3 * 17 - 1 = 50; 3 * 7 - 1 = 20; 230 + 50 + 20 = 300$  (ответ верный)

## **4 Выводы**

Я приобрела навыки написания программ с использованием циклов и обработки аргументов командной строки. Узнала, как работают push и pop.