

# **Отчёт по лабораторной работе №8**

Цатурьян Лев Вячеславович НММбд-03-23

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
<b>3</b>	<b>Задание для самостоятельной работы</b>	<b>11</b>
<b>4</b>	<b>Выводы</b>	<b>13</b>

## Список иллюстраций

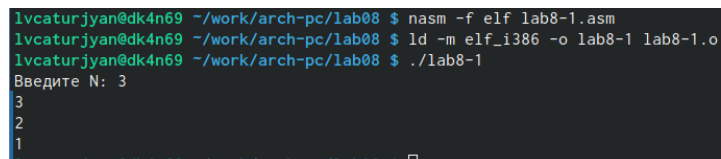
2.1	Запуск исполняемого файла . . . . .	5
2.2	Работа программы . . . . .	6
2.3	Изменённый текст программы . . . . .	7
2.4	Создание объектного и исполняемого файлов. Запуск программы	7
2.5	Работа программы . . . . .	7
2.6	Создание объектного и исполняемого файлов. Запуск программы	8
2.7	Изменённый текст программы для вычисления произведения аргументов . . . . .	9
2.8	Создание объектного и исполняемого файлов. Запуск программы	10
3.1	Программа в тс . . . . .	11
3.2	Работа программы . . . . .	12

# 1 Цель работы

Приобретение навыков написания программ с использованием циклов, работа со стеками.

## 2 Выполнение лабораторной работы

Сначала я создал каталог для программ лабораторной работы № 8, перешёл в него и создал файл lab8-1.asm: Далее я ввёл в созданный файл текст листинга 8.1, создал объектный и исполняемый файлы



```
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 3
3
2
1
```

Рис. 2.1: Запуск исполняемого файла

Программа работает некорректно

Далее я изменил текст программы добавив изменение значение регистра есх в цикле Для этого я добавил команду `sub есх,1`

Далее я создал объектный и исполняемый файлы и запустил программу

```
4292571292
4292571290
4292571288
4292571286
4292571284
4292571282
4292571280
4292571278
4292571276
4292571274
4292571272
4292571270
4292571268
4292^Z
[1]+  Остановлен    ./lab8-1
lvcaturljyan@dk4n69 ~/work/arch-pc/lab08 $
```

Рис. 2.2: Работа программы

Регистр `ecx` принимает бесконечно много значений, так как из него каждый раз вычитается единица. Число проходов цикла не соответствует значению введенному с клавиатуры.

Далее я изменил текст программы, добавив команды `push` и `pop` (добавление в стек и извлечение из стека).

```
mov ecx,[N]
; Счетчик цикла, 'ecx=N'
label:
push ecx
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
pop ecx
loop label
; Вывод значения 'N'
; 'ecx=ecx-1' и если 'ecx' не '0'
; переход на 'label'
call quit
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия

Рис. 2.3: Изменённый текст программы

```
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 3
2
1
0
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $
```

Рис. 2.4: Создание объектного и исполняемого файлов. Запуск программы

Все работает корректно

Далее я создал файл lab8-2.asm и ввёл в него текст из листинга 8.2

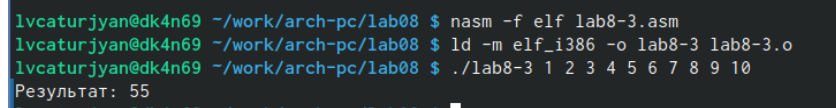
```
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ nasm -f elf lab8-2.asm
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o

lvcaturjyan@dk8n59 ~/work/arch-pc/lab08 $ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
lvcaturjyan@dk8n59 ~/work/arch-pc/lab08 $
```

Рис. 2.5: Работа программы

Команда обработала 4 аргумента и вывела каждый из них на отдельной строке

После этого я создал файл lab8-3.asm и ввел в него текст из листинга 8.3



```
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ./lab8-3 1 2 3 4 5 6 7 8 9 10
Результат: 55
```

Рис. 2.6: Создание объектного и исполняемого файлов. Запуск программы

Результат я проверил вручную. Программа работает правильно

Далее мне потребовалось изменить текст файла так, чтобы вычислялось произведение аргументов



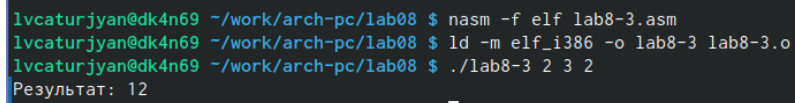
```
pop    edx
sub    ecx, 1
mov    esi, 1

next:
cmp    ecx, 0h
jz     _end

pop    eax
call   atoi
mov    edx, eax
mov    eax, esi
mul    edx
mov    esi, eax
```

Рис. 2.7: Изменённый текст программы для вычисления произведения аргументов

Вместо значения 0 в регистре esi теперь значение 1 - это нужно, чтобы результат умножения был отличен от нуля. Остальные добавленные мной команды умножают значение в регистре esi на введенный аргумент.



```
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ./lab8-3 2 3 2
Результат: 12
```

Рис. 2.8: Создание объектного и исполняемого файлов. Запуск программы

Программа работает правильно

### 3 Задание для самостоятельной работы

Мой вариант из прошлой лабораторной работы - 7

Моя программа подставляет введённые пользователем значения аргументов в функцию, запоминает значения функции от каждого аргумента и суммирует их

```
q.asm [----] 47 L: [ 1+ 0 1/ 39] *(63 / 470b) 0010 0x00A
#include "input.h" ; Цатурьян Лев ИММбд-03-23
SECTION .data
aff db "Вариант 7. Функция вида y=3(x+2)",0
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 edx, eax
mov eax, 2
add edx, eax
mov eax, 3
mul edx
add esi, eax

loop next

_end:
mov eax, aff
call sprintf

mov eax, msg
call sprintf

mov eax, esi
call iprintf
call quit
```

Рис. 3.1: Программа в mc

Её листинг:

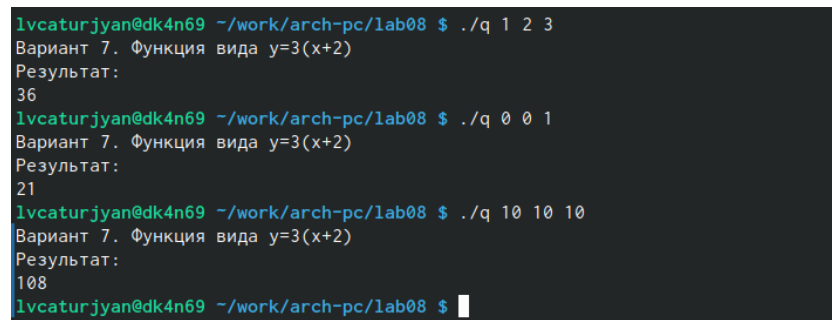
```

%include'in_out.asm'; Цатурьян Лев НММбд-03-23 SECTION .data aff db "Вариант
7. Функция вида  $y=3(x+2)$ ",0 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 edx, eax mov eax, 2 add edx, eax mov eax, 3 mul edx add esi,eax
    loop next
_end: mov eax, aff call sprintLF
    mov eax, msg call sprintLF
    mov eax, esi
    call iprintLF call quit

```

Программа извлекает значение аргумента из стека, выполняет над ним арифметические операции, после чего прибавляет полученное значение к регистру промежуточных сумм (esi), с помощью цикла loop эти действия повторяются, пока не кончатся аргументы

Далее я создал объектный и исполняемый файлы и запустил программу, вводя разные значения аргументов



```

lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ./q 1 2 3
Вариант 7. Функция вида  $y=3(x+2)$ 
Результат:
36
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ./q 0 0 1
Вариант 7. Функция вида  $y=3(x+2)$ 
Результат:
21
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $ ./q 10 10 10
Вариант 7. Функция вида  $y=3(x+2)$ 
Результат:
108
lvcaturjyan@dk4n69 ~/work/arch-pc/lab08 $

```

Рис. 3.2: Работа программы

Программа работает корректно, результаты вычислений были проверены мной вручную

## 4 Выводы

Я приобрёл навыки использования циклов в программе