

# **Отчёт по лабораторной работе №8**

**Архитектура компьютера НММбд 03-24**

Топорова Дарья Сергеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выполнение самостоятельной работы</b>	<b>10</b>
<b>5</b>	<b>Выводы</b>	<b>11</b>

# Список иллюстраций

3.1 создаём каталог и файл . . . . . 7

## **Список таблиц**

# 1 Цель работы

Целью работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки в NASM.

## 2 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  $x = x_1, x_2, \dots, x_n$ .

## 3 Выполнение лабораторной работы

### 1.1 создаю каталог lab08 и файл для выполнения лабораторную №8

```
dstoporova@dk6n52 ~ $ mkdir ~/work/arch-pc/lab08
dstoporova@dk6n52 ~ $ cd ~/work/arch-pc/lab08
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ touch lab8-1.asm
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ mc
```

Рис. 3.1: создаём каталог и файл

### 1.2 Ввожу в файл lab8-1.asm программу вывода значения регистра eax из листинга 8.1. Запускаю исполняемый файл.

```
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
```

```
mc [dstoporova@dk6n52.dk.sci.pfu.edu.ru:~/work/arch-
Файл Правка Вид Поиск Терминал Справка
~/afs/.dk.sci.pfu.edu.ru/home/~work/arch-pc/lab08/lab8-1.asm
%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:
```

### 1.3 Изменим текст программы, добавив изменение значение регистра ecx в цикле. Запускаем изменённую программу. Теперь число проходов цикла не соответствует значению, введённому с клавиатуры.

```

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

```

```

dstoporova@dk6n52 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 10
9
7
5
3
1

```

1.4 Внесём изменения в текст программы добавив команды push и pop для сохранения значения счетчика цикла loop. Запускаем программу и проверим ее работу. Число проходов цикла соответствует числу, введённому с клавиатуры.

```

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

```

```

dstoporova@dk6n52 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0

```

2.1 создаём файл lab8-2.asm. Вводим программу выводящую на экран аргументы командной строки из листинга 8.2. Запускаю исполняемый файл.

```

mc [dstoporova@dk6n52.dk.sci.pfu.edu.ru:~/work/arch-pc/lab08]
Файл Правка Вид Поиск Терминал Справка
~/afs/.dk.sci.pfu.edu.ru/home/~work/arch-pc/lab08/lab8-2.asm 943/9
#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 sprintf ; вызываем функцию печати
loop next ; переход к обработке следующего
; аргумента (переход на метку 'next')
_end:
call quit

```

```

dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-2 аргумент1 аргумент2 аргумент3
аргумент1
аргумент2
аргумент3

```

2.2 создаём файл lab8-3.asm. Введём программу по вычислению суммы аргументов командной строки, запустим исполняемый файл.



```

mc [dstoporova@dk6n52.dk.sci.pfu.edu.ru]:~/work/arch-pc/lab08
Файл Правка Вид Поиск Терминал Справка
lab8-3.asm [----] 7 L: [ 4+10 14/ 29] *(585 /1428b) 1078 0x436
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)
mov esi, 0 ; Используем 'esi' для хранения
; промежуточных сумм
next:
cmp ecx,0h ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку '_end')
pop 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

```

```

dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-3 12 13 7
Результат: 47

```

2.3 Изменяем текст программы из листинга 8.3 для вычисления произведения аргументов командной строки.

```

_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)
mov esi, 1
next:
cmp ecx,0h ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку '_end')
pop eax ; иначе извлекаем следующий аргумент из стека
call atoi ; преобразуем символ в число
mov ebx, esi
mul ebx
mov esi, eax
loop next ; переход к обработке следующего аргумента

```

```

dstoporova@dk6n52 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-3 12 13 7 10 5
Результат: 54600

```

## 4 Выполнение самостоятельной работы

3.1 Напишем программу, которая находит сумму значений функции  $f(x)$  для  $x=x_1, x_2, \dots, x_n$ , т.е. программа должна выводить значение  $f(x_1) + f(x_2) + \dots + f(x_n)$ . Мой вариант - 16. Создадим исполняемый файл и проверим его работу на нескольких наборах  $x=x_1, x_2, \dots, x_n$ .

```
mc [dstoporova@dk6n52.dk.sci.pfu.edu.ru]:~/work/arch-pc/lab08
Файл  Правка  Вид  Поиск  Терминал  Справка
lab8-4.asm  [-----]  7  L: [ 12+15  27/ 40]  *(258 / 362b)  0010  0x00A
pop edx

sub ecx,1

mov esi,0

mov eax,prim
call sprintf
next:
cmp ecx,0
jz _end

mov ebx,30
pop eax
call atoi
mul ebx

add eax,-11

add esi,eax

loop next
```

```
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ nasm -f elf lab8-4.asm
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-4
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-4 2 4 5
f(x)=30x-11
Результат: 297
dstoporova@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-4 1 2 3
f(x)=30x-11
Результат: 147
```

## 5 Выводы

В результате выполнения лабораторной работы я приобрёл навыки написания программ с использованием циклов и обработкой аргументов командной строки в NASM.