

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

## **ОТЧЁТ**

### **ПО ЛАБОРАТОРНОЙ РАБОТЕ №8**

**Дисциплина: архитектура компьютера**

Студент: Ванюшкина Т.В.

Группа: НКАбд-01-24

Студ.билет: 1132246713

**МОСКВА**

**2024г**

## **СОДЕРЖАНИЕ**

<b>1. Цель работы.....</b>	<b>3</b>
<b>2. Теоретическое введение.....</b>	<b>4</b>
<b>3. Выполнение лабораторной работы.....</b>	<b>5-11</b>
<b>3.1 Реализация циклов в NASM .....</b>	<b>5-8</b>
<b>3.2 Обработка аргументов командной строки .....</b>	<b>8-10</b>
<b>3.3 Задание для самостоятельной работы.....</b>	<b>11</b>
<b>4. Выводы.....</b>	<b>12</b>
<b>5. Список литературы.....</b>	<b>13</b>

## **1.Цель работы**

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

## 2. 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 нём 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, помещённое в стек последним, извлекается первым. При помещении значения в стек указатель стека уменьшается, а при извлечении—увеличивается.

## 3. Выполнение лабораторной работы

### 3.1 Реализация циклов в NASM

Создаю каталог для программ лабораторной работы №8:

```
tatyana@vbox:~$ mkdir ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab08
```

Рис.1: Создание каталога

Перехожу в него :

```
tatyana@vbox:~$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab08
```

Рис.2: Переход в каталог

Создаю файл lab8-1.asm:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ touch lab8-1.asm
```

Рис.3: Создание файла

Ввожу в него текст программы из листинга 8.1:

```
mc [tatyana@vbox]:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08
lab8-1.asm  [-M--]  0 L: [ 1+27 28/ 28] *(675 / 675b) <EOF>
%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'
loop label    ; 'ecx=ecx-1' и если 'ecx' не '0'
call quit    ; переход на 'label'
```

Рис.4: Ввод текста программы

Создаю исполняемый файл и проверяю его работу:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ nasm -f elf lab8-1.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ./lab8-1
Введите N: 8
8
7
6
5
4
3
2
1
```

Рис.5: Создание и проверка файла

Изменяю текст программы добавив изменение значение регистра есх в цикле:

```
mc [tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/a
lab8-1.asm [-M--] 11 L:[ 1+26 27/ 29] *(559 / 606b) 0010 0x00A
#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:
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
loop label
call quit ; переход на `label`
```

Рис.6: Замена текста программы

Создаю исполняемый файл и проверяю его работу:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ nasm -f elf lab8-1.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ./lab8-1
Введите N: 8
7
5
3
1
```

Рис.7: Создание и проверка файла

Вношу изменения в текст программы добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop:

```
mc [tatyana@vbox]:~/work/study/2024-2025/Архитектура компьютера/arch-p
lab8-1.asm [-----] 8 L: [ 1+27 28/ 31] *(566 / 625b) 0010 0x00A
%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:
push ecx
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
pop ecx
loop label
call quit ; переход на 'label'
```

Рис.8: Замена текста программы

Создаю исполняемый файл и проверяю его работу:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ nasm -f elf lab8-1.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ./lab8-1
Введите N: 8
7
6
5
4
3
2
1
0
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$
```

Рис.9: Создание и проверка файла

## 3.2 Обработка аргументов командной строки

Создаю файл lab8-2.asm:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ touch lab8-2.asm
```

Рис.10: Создание файла

Ввожу в него текст программы из листинга 8.2:

```
mc [tatyana@vbox]:~/work/study/2024-2025/Архитектура компь
lab8-2.asm [-M--] 10 L: [ 1+14 15/ 15] *(174 / 174b) <EOF>
#include 'in_out.asm'
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx, 1
next:
cmp ecx, 0
jz _end
pop eax
call sprintLF
loop next
_end:
call quit
```

Рис.11: Ввод текста программы



Создаю исполняемый файл и запускаю его,указав аргументы:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ nasm -f elf lab8-2.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
```

Рис.12: Создание и запуск файла

Создаю файл lab8-3.asm:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ touch lab8-3.asm
```

Рис.13: Создание файла

Ввожу в него текст программы из листинга 8.3:

```
lab8-3.asm [-M--] 0 L: [ 1+22 23/ 23] *(288 / 298b) 0032 0x
%include 'in_out.asm'
SECTION .data
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
add esi,eax
loop next
_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис.14: Ввод текста программы

Создаю исполняемый файл и запускаю его,указав аргументы:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ nasm -f elf lab8-3.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
```

Рис.15: Создание и запуск файла

Изменяю текст программы из листинга 8.3 для вычисления произведения аргументов командной строки:

```
lab8-3.asm      [-M--] 12 L:[ 1+16 17/ 24] *(221 / 306b) 0010 0x00
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi,1
next:
cmp ecx,0h
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
```

Рис.16: Замена текста программы

Создаю исполняемый файл и запускаю его:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ nasm -f elf lab8-3.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ./lab8-3 12 13 7 10 5
Результат: 54600
```

Рис.17: Создание и запуск файла

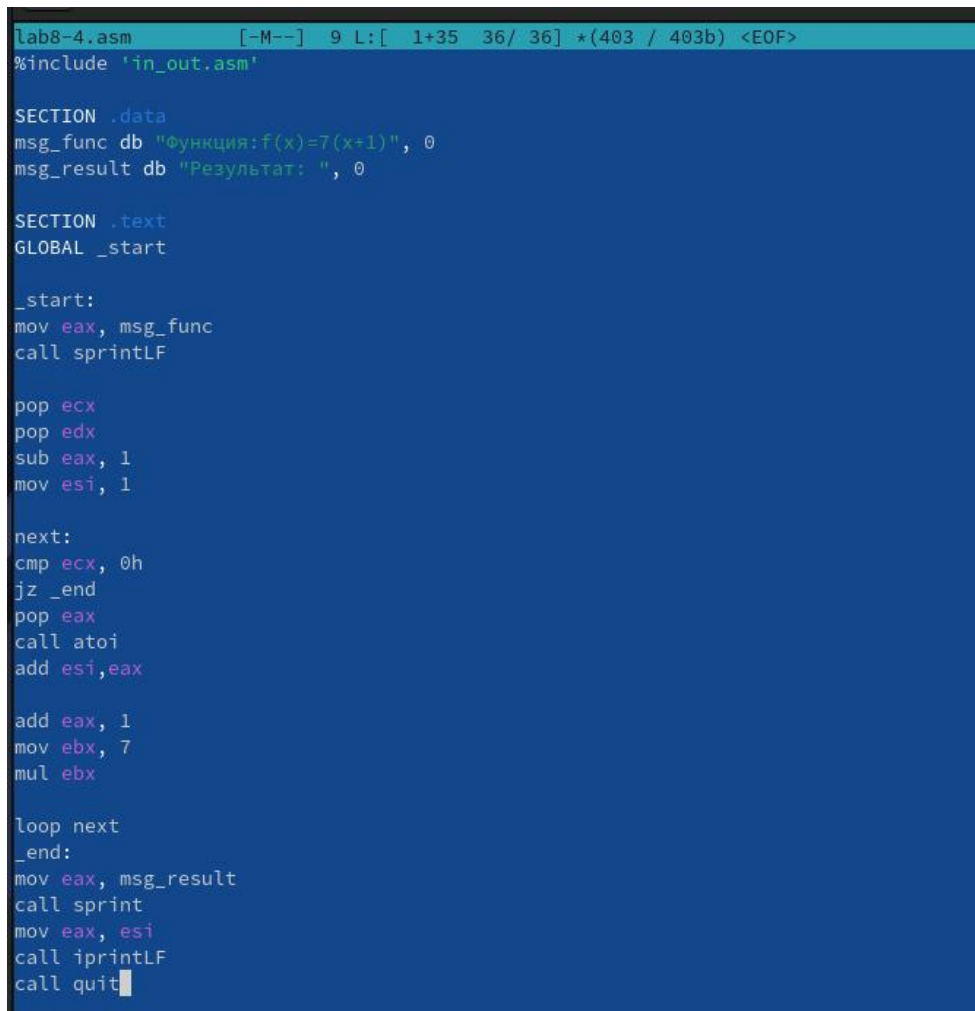
### 3.3 Задание для самостоятельной работы

Создаю файл lab8-4.asm:

```
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ touch lab8-4.asm
```

Рис.18: Создание файла

Ввожу в него текст программы для вычисления суммы значений функции  $f(x)$  для  $x = x_1, x_2, \dots, x_n$ . Вид функции: Вариант 14  $f(x) = 7(x + 1)$



```
lab8-4.asm [-M--] 9 L: [ 1+35 36/ 36] *(403 / 403b) <EOF>
#include 'in_out.asm'

SECTION .data
msg_func db "Функция: f(x)=7(x+1)", 0
msg_result db "Результат: ", 0

SECTION .text
GLOBAL _start

_start:
mov eax, msg_func
call sprintfLF

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

next:
cmp ecx, 0h
jz _end
pop eax
call atoi
add esi, eax

add eax, 1
mov ebx, 7
mul ebx

loop next
_end:
mov eax, msg_result
call sprintf
mov eax, esi
call iprintLF
call quit
```

Рис.19: Ввод текста программы

Создаю исполняемый файл и проверяю его работу:

```
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ nasm -f elf lab8-4.asm
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab08$ ./lab8-4 1 2 3 4 5
Функция: f(x)=7(x+1)
Результат: 140
```

Рис.20: Создание и запуск файла

## **4. Выводы**

В ходе выполнения лабораторной работы я приобрела навыки написания программ с использованием циклов и обработкой аргументов командной строки.

## **5. Список литературы**

Курс: Архитектура компьютеров и операционные системы. Раздел "Архитектура компьютеров" (02.03.00, УГСН) (rudn.ru)

