

# **Отчет по лабораторной работе №8**

**Дисциплина: Архитектура компьютера**

Морошан Матвей Корнелиович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Реализация циклов в NASM . . . . .	6
3.2	Обработка аргументов командной строки . . . . .	9
3.3	Выполнение заданий для самостоятельной работы . . . . .	12
<b>4</b>	<b>Выводы</b>	<b>14</b>

## Список иллюстраций

3.1	Создание каталога, переход и создание файла . . . . .	6
3.2	Копирование файла . . . . .	6
3.3	Файл lab8-1.asm . . . . .	7
3.4	Создание исполняемого файла и проверка . . . . .	7
3.5	Изменение текста программы . . . . .	8
3.6	Создание исполняемого файла и проверка . . . . .	8
3.7	Изменение текста программы . . . . .	9
3.8	Создание исполняемого файла и проверка . . . . .	9
3.9	Создание lab8-2.asm . . . . .	9
3.10	Файл lab8-2.asm . . . . .	10
3.11	Создание исполняемого файла и запуск его . . . . .	10
3.12	Создание lab8-3.asm . . . . .	10
3.13	Файл lab8-3.asm . . . . .	11
3.14	Результат работы файла . . . . .	11
3.15	Изменение текст программы lab8-3.asm . . . . .	12
3.16	Результат работы файла для произведения . . . . .	12
3.17	Создание lab8-4.asm . . . . .	12
3.18	Текст программы lab8-4.asm . . . . .	13
3.19	Проверка работы программы . . . . .	13

# 1 Цель работы

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

## **2 Задание**

1. Реализация циклов в NASM
2. Обработка аргументов командной строки
3. Выполнение заданий для самостоятельной работы

## 3 Выполнение лабораторной работы

### 3.1 Реализация циклов в NASM

Создаю каталог для программ по лабораторной работе №8, перехожу в него и создаю файл lab8-1.asm (рис. 3.1)

```
mkmoroshan@dk2n21 ~ $ mkdir ~/work/arch-pc/lab08
mkmoroshan@dk2n21 ~ $ cd ~/work/arch-pc/lab08
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ touch lab8-1.asm
```

Рис. 3.1: Создание каталога, переход и создание файла

Копирую файл in\_out.asm из загрузок в соответствующую папку для дальнейшей работы (рис. 3.2)

```
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ cp ~/Загрузки/in_out.asm in_out.asm
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ls
in_out.asm  lab8-1.asm
```

Рис. 3.2: Копирование файла

Ввожу в lab8-1.asm текст программы из листинга 8.1 (рис. 3.3)

```
> mc [mkmoroshan@dk2n21.dk.sci.pfu.edu.ru:~/work/arch-pc/lab08]
lab8-1.asm [-M--] 9 L:[ 1+27 28/ 28] *(636 / 636b) <EOF> [*][X]
#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 iprintf ; Вывод значения 'N'
loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
; переход на 'label'
call quit
```

Рис. 3.3: Файл lab8-1.asm

Создаю исполняемый файл и проверяю его работу (рис. 3.4)

```
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 4
4
3
2
1
```

Рис. 3.4: Создание исполняемого файла и проверка

Изменяю текст программы (рис. 3.5)

```
> mc [mkmoroshan@dk2n21.dk.sci.pfu.edu.ru:~/work/arch-pc/lab08]
lab8-1.asm [-M--] 9 L: [ 1+22 23/ 29] *(482 / 646b) 0010 0x00A [*][X]
#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 ; Вывод значения 'N'
loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
; переход на 'label'
call quit
```

Рис. 3.5: Изменение текста программы

Создаю исполняемый файл и проверяю работу. Число проходов цикла не соответствует значению N введенному с клавиатуры (рис. 3.6)

```
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 4
3
1
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 6
5
3
1
```

Рис. 3.6: Создание исполняемого файла и проверка

Снова изменяю текст программы (рис. 3.7)



```
> mc [mkmoroshan@dk2n21.dk.sci.pfu.edu.ru:~/work/arch-pc/lab08]
lab8-1.asm [-M--] 7 L: [ 1+27 28/ 31] *(571 / 663b) 0010 0x00A [*][X]
#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 iprintf ; Вывод значения 'N'
pop ecx
loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
; переход на 'label'
call quit
```

Рис. 3.7: Изменение текста программы

Создаю исполняемый файл и проверяю его работу. В данном случае число проходов соответствует числу N введенному с клавиатуры (рис. 3.8)

```
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 6
5
4
3
2
1
0
```

Рис. 3.8: Создание исполняемого файла и проверка

## 3.2 Обработка аргументов командной строки

Создаю файл lab8-2.asm (рис. 3.9)

```
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ touch lab8-2.asm
```

Рис. 3.9: Создание lab8-2.asm

Ввожу в lab8-2.asm текст программы из листинга 8.2 (рис. 3.10)

```

> mc [mkmoroshan@dk2n21.dk.sci.pfu.edu.ru:~/work/arch-pc/lab08
lab8-2.asm          [~M--]  9 L:[ 1+19 20/ 20] *(943 / 943b) <EOF> [X][X]
%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[]

```

Рис. 3.10: Файл lab8-2.asm

Создаю исполняемый файл и запускаю его. Программа обрабатывает все 3 аргумента (рис. 3.11)

```

mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab8-2.asm
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3

```

Рис. 3.11: Создание исполняемого файла и запуск его

Создаю файл lab8-3.asm (рис. 3.12)

```

mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ touch lab8-3.asm

```

Рис. 3.12: Создание lab8-3.asm

Ввожу в lab8-3.asm текст программы из листинга 8.3 (рис. 3.13)

```

> mc [mkmoroshan@dk2n21.dk.sci.pfu.edu.ru:~/work/arch-pc/lab08
lab8-3.asm          [-M--] 32 L:[ 1+28 29/ 29] *(1428/1428b) <EOF>
#include "in_out.asm"
SECTION .data
msg db "Результат: ",0
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 sprintf
    mov eax, esi ; записываем сумму в регистр 'eax'
    call iprintf ; печать результата
    call quit ; завершение программы

```

Рис. 3.13: Файл lab8-3.asm

Создаю исполняемый файл и запускаю его, получив результат (рис. 3.14)

```

mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-3 12 13 7 10 5
Результат: 47

```

Рис. 3.14: Результат работы файла

Я изменил текст программы так, чтоб вместо суммы аргументов было произведение (рис. 3.15)

```
> mc [mkmoroshan@dk2n21.dk.sci.pfu.edu.ru:~/work/arch-pc/lab08]
lab8-3.asm [----] 12 L: [ 1+22 23/ 27] *(264 / 313b) 0010 0x00A [X]
%include "in_out.asm"
SECTION .data
msg db "Program: ",0
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi,1
mov eax,1
next:
cmp ecx,0
jz _end
pop eax
call atoi
mov ebx,eax
mov eax,esi
mul ebx
mov esi,eax
loop next
_end:
mov eax,msg
call sprint
mov eax,esi
call iprintf
call quit
```

Рис. 3.15: Изменение текст программы lab8-3.asm

Создаю исполняемый файл и запускаю его, получив результат для произведения (рис. 3.16)

```
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-3 12 5 7
Результат: 420
```

Рис. 3.16: Результат работы файла для произведения

### 3.3 Выполнение заданий для самостоятельной работы

Создаю файл lab8-4.asm для выполнения задания для самостоятельной работы (рис. 3.17)

```
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ touch lab8-4.asm
```

Рис. 3.17: Создание lab8-4.asm

У меня 4 вариант -  $f(x)=2(x-1)$ . Поскольку не было сказано, что нельзя преобразовывать выражение, то я раскрыл скобки и получил  $f(x)=2x-2$ . Пишу текст программы для вычислений (рис. 3.18)

```

> mc [mkmoroshan@dk2n21.dk.sci.pfu.edu.ru:~/work/arch-pc/lab08
lab8-4.asm [----] 9 L:[ 1+28 29/ 29] *(349 / 349b) <EOF> [X][X]
#include "in_out.asm"
section .data
prim db 'f(x)=2(x-1)',0
otv db 'Результат: ',0
section .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi,0
mov eax,prim
call sprintf
next:
cmp ecx,0
jz _end
mov ebx,2
pop eax
call atoi
mul ebx
add eax,-2
add esi,eax
loop next
_end:
mov eax,otv
call sprintf
mov eax,esi
call iprintf
call quit

```

Рис. 3.18: Текст программы lab8-4.asm

Проверяю работу файла для трех наборов (рис. 3.19)

```

mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-4 1 2 3 4
f(x)=2(x-1)
Результат: 12
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-4 1 3 5 6
f(x)=2(x-1)
Результат: 22
mkmoroshan@dk2n21 ~/work/arch-pc/lab08 $ ./lab8-4 1 2 3 4 5 7
f(x)=2(x-1)
Результат: 32

```

Рис. 3.19: Проверка работы программы

## 4 Выводы

При выполнении данной лабораторной работы я приобрел навыки написания программ с использованием циклов и обработкой аргументов командной строки