

# Отчет по лабораторной работа №8

Дисциплина: архитектура компьютера

Ширинкин Т. Б.

## Содержание

Цель работы .....	1
Выполнение лабораторной работы .....	1
Выполнение заданий для самостоятельной работы.....	6
Выводы .....	7

## Цель работы

Изучение циклов в Nasm, получения умений и пониманию особенностей работы циклов.

## Выполнение лабораторной работы

Создал каталог и первый файл (Рис. [-@fig:000])

```
tbshirinkin@Shiza:~$ mkdir ~/work/arch-pc/lab08
tbshirinkin@Shiza:~$ cd ~/work/arch-pc/lab08
tbshirinkin@Shiza:~/work/arch-pc/lab08$ touch lab8-1.asm
```

*Рис. 0 Создал каталог и первый файл*

Ввёл первый листинг (Рис. [-@fig:001])







```
%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. 7 Создал новый файл и ввёл в него новый листинг*

Вывел все 4 аргумента, что вписали в стек программы (они идут через пробел). (Рис. [ @fig:008])

```
tbshirinkin@Shiza:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ./lab8-2 аргумент1 аргумент2 'аргумент 3'
аргумент1
аргумент2
2
аргумент3
```

Рис. 8 Вывел все 4 аргумента, что вписали в стек программы (они идут через пробел).

Создал новый файл и ввёл в него новый листинг (Рис. [-@fig:009])



```

#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
mov ebx,8
mul ebx
mov ebx,3
sub eax,ebx
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
mov eax, esi ; записываем сумму в регистр 'eax'
call iprintLF ; печать результата
call quit ; завершение программы
~
~
~

```

36,1 All

Рис. 11 На основе последнего листинга написал новую программу (19 номер)

Работает как изначально задумано (Рис. [-@fig:012])

```

tbshirinkin@Shiza:~/work/arch-pc/lab08$ nasm -f elf lab8-sm19-1.asm
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-sm19-1 lab8-sm19-1.o
tbshirinkin@Shiza:~/work/arch-pc/lab08$ ./lab8-sm19-1 1 2 1
Результат: 23

```

Рис. 12 Работает как изначально задумано

## Выводы

Цель достигнута: изучены циклы в Nasm, получены умения и понимания особенностей работы циклов.