

# **Отчёт по лабораторной работе**

**Арифметические операции в NASM.**

Югай Александр Витальевич

# 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>
3.1	Выполнение арифметических операций в NASM . . . . .	12
3.2	Ответы на вопросы по программе . . . . .	16
3.3	Задание для самостоятельной работы . . . . .	17
<b>4</b>	<b>Выводы</b>	<b>20</b>

## Список иллюстраций

3.1	Создаем каталог и файл . . . . .	7
3.2	Заходим в Midnight Commander . . . . .	7
3.3	Заполняем файл по листингу 6.1 . . . . .	8
3.4	Запускаем файл и смотрим на его работу . . . . .	8
3.5	Изменяем файл . . . . .	9
3.6	Запускаем файл и смотрим на изменения . . . . .	9
3.7	Создаем файл . . . . .	9
3.8	Заполняем файл . . . . .	10
3.9	Смотрим на работу программы . . . . .	10
3.10	Изменяем файл . . . . .	11
3.11	Смотрим на работу программы . . . . .	11
3.12	Изменяем файл . . . . .	12
3.13	Смотрим на работу программы . . . . .	12
3.14	Создаем файл . . . . .	12
3.15	Заполняем файл . . . . .	13
3.16	Смотрим на результат работы программы . . . . .	14
3.17	Редактируем файл . . . . .	14
3.18	Смотрим на результат работы программы . . . . .	14
3.19	Создаем файл . . . . .	14
3.20	Заполняем файл . . . . .	15
3.21	Проверяем результат работы программы . . . . .	16
3.22	Создаем файл . . . . .	17
3.23	Заполняем файл . . . . .	18
3.24	Проверяем работу программы . . . . .	19
3.25	Проверяем работу программы . . . . .	19

## Список таблиц

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

## 2 Задание

Написать программы для решения выражений

### 3 Выполнение лабораторной работы

Создаем каталог для программ Лаб6, и в нем создаем файл

```
avyugayj@ubuntu:~$ mkdir ~/work/arch-pc/lab06
avyugayj@ubuntu:~$ cd ~/work/arch-pc/lab06
avyugayj@ubuntu:~/work/arch-pc/lab06$ touch lab6-1.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$
```

Рис. 3.1: Создаем каталог и файл

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1

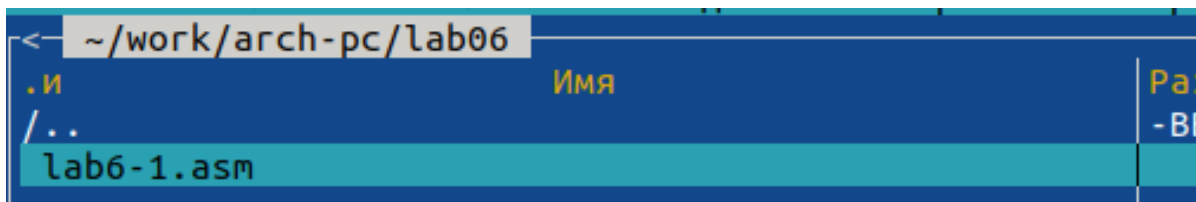


Рис. 3.2: Заходим в Midnight Commander

```
GNU nano 6.2
#include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF
call quit
```

Рис. 3.3: Заполняем файл по листингу 6.1

Создаем исполняемый файл и запускаем его

```
avyugayj@ubuntu:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./lab6-1
j
avyugayj@ubuntu:~/work/arch-pc/lab06$
```

Рис. 3.4: Запускаем файл и смотрим на его работу

Снова открываем файл для редактирования и убираем кавычки с числовых значений



```

%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, 6
mov ebx, 4
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintLF
call quit

```

Рис. 3.5: Изменяем файл

Создаем исполняемый файл и запускаем его

```

avyugayj@ubuntu:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./lab6-1

avyugayj@ubuntu:~/work/arch-pc/lab06$

```

Рис. 3.6: Запускаем файл и смотрим на изменения

Создаем новый файл в каталоге

```

avyugayj@ubuntu:~/work/arch-pc/lab06$ touch lab6-2.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$

```

Рис. 3.7: Создаем файл

Заполняем файл в соответствии с листингом 6.2

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 3.8: Заполняем файл

Создаем исполняемый файл и запускаем его

```
avyugayj@ubuntu:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./lab6-2
106
avyugayj@ubuntu:~/work/arch-pc/lab06$
```

Рис. 3.9: Смотрим на работу программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений

```
one more file  
%include 'in_out.asm'  
SECTION .text  
GLOBAL _start  
_start:  
mov eax, 6  
mov ebx, 4  
add eax, ebx  
call iprintLF  
call quit
```

Рис. 3.10: Изменяем файл

Создаем исполняемый файл и запускаем его

```
avyugayj@ubuntu:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm  
avyugayj@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o  
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./lab6-2  
10  
avyugayj@ubuntu:~/work/arch-pc/lab06$ █
```

Рис. 3.11: Смотрим на работу программы

Снова открываем файл для редактирования и меняем iprintLF на iprint

```

%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, 6
mov ebx, 4
add eax, ebx
call iprint
call quit

```

Рис. 3.12: Изменяем файл

Создаем исполняемый файл и запускаем его

```

avyugayj@ubuntu:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./lab6-2
10avyugayj@ubuntu:~/work/arch-pc/lab06$

```

Рис. 3.13: Смотрим на работу программы

Вывод функций `iprintLF` и `iprint` отличаются только тем, что `LF` переносит на новую строку.

### 3.1 Выполнение арифметических операций в NASM

Создаем новый файл в каталоге

```

avyugayj@ubuntu:~/work/arch-pc/lab06$ touch lab6-3.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$

```

Рис. 3.14: Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.3

```
;-----  
; Программа вычисления выражения  
;-----  
%include 'in_out.asm' ; подключение внешнего файла  
SECTION .data  
div: DB 'Результат: ',0  
rem: DB 'Остаток от деления: ',0  
SECTION .text  
GLOBAL _start  
_start:  
; ---- Вычисление выражения  
mov eax,5 ; EAX=5  
mov ebx,2 ; EBX=2  
mul ebx ; EAX=EAX*EBX  
add eax,3 ; EAX=EAX+3  
xor edx,edx ; обнуляем EDX для корректной работы div  
mov ebx,3 ; EBX=3  
div ebx ; EAX=EAX/3, EDX=остаток от деления  
mov edi,eax ; запись результата вычисления в 'edi'  
; ---- Вывод результата на экран  
mov eax,div ; вызов подпрограммы печати  
call sprint ; сообщения 'Результат: '  
mov eax,edi ; вызов подпрограммы печати значения  
call iprintLF ; из 'edi' в виде символов  
mov eax,rem ; вызов подпрограммы печати  
call sprint ; сообщения 'Остаток от деления: '  
mov eax,edx ; вызов подпрограммы печати значения  
call iprintLF ; из 'edx' (остаток) в виде символов  
call quit ; вызов подпрограммы завершения
```

Рис. 3.15: Заполняем файл

Создаем исполняемый файл и запускаем его

```

avyugayj@ubuntu:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
avyugayj@ubuntu:~/work/arch-pc/lab06$

```

Рис. 3.16: Смотрим на результат работы программы

Открываем файл и редактируем его для вычисления выражения  $f(x) = (5 \times 2 + 3)/3$

```

; вычисление выражения
mov eax,4 ; EAX=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'

```

Рис. 3.17: Редактируем файл

Компилируем файл и запускаем программу

```

avyugayj@ubuntu:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
avyugayj@ubuntu:~/work/arch-pc/lab06$

```

Рис. 3.18: Смотрим на результат работы программы

Создаем новый файл в каталоге

```

Остаток от деления: 1
avyugayj@ubuntu:~/work/arch-pc/lab06$ touch variant.asm

```

Рис. 3.19: Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.4

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x
xor edx,edx
mov ebx,20
div ebx
inc edx
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис. 3.20: Заполняем файл

Компилируем файл и запускаем его

```

avyugayj@ubuntu:~/work/arch-pc/lab06$ nasm -f elf variant.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132230302
Ваш вариант: 3
avyugayj@ubuntu:~/work/arch-pc/lab06$ █

```

Рис. 3.21: Проверяем результат работы программы

## 3.2 Ответы на вопросы по программе

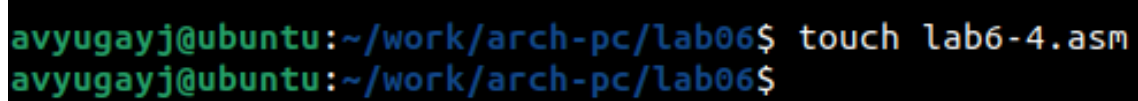
1. Строка “moveax,rem” и строка “call sprint” отвечают за вывод на экран сообщения ‘Ваш вариант:’.
2. Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре esx, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре edx. Затем вызывается процедура sread, которая выполняет чтение строки.
3. Инструкция “call atoi” используется для преобразования строки в целое число. Она принимает адрес строки в регистре eax и возвращает полученное число в регистре eax.
4. Строка “xoredx,edx” обнуляет регистр. edx перед выполнением деления. Строка “movebx,20” загружает значение 20 в регистр ebx. Строка “divebx” выполняет деление регистра eax на значение регистра ebx с сохранением частного в регистре eax и остатка в регистре edx,
5. Остаток от деления записывается в регистр edx.
6. Инструкция “inc edx” используется для увеличения значения в регистре edx на 1. В данном случае, она увеличивает остаток от деления на 1.
7. Строка “mov eax,edx” передает значение остатка от деления в регистр eax.  
36 Строка “call iprintLF” вызывает процедуру iprintLF для вывода значения



на экран вместе с переводом строки.

### 3.3 Задание для самостоятельной работы

Создаем новый файл в каталоге

A terminal window with a black background and green text. The first line shows the command 'touch lab6-4.asm' being executed. The second line shows the prompt after the command has finished.

```
avyugayj@ubuntu:~/work/arch-pc/lab06$ touch lab6-4.asm  
avyugayj@ubuntu:~/work/arch-pc/lab06$
```

Рис. 3.22: Создаем файл

Открываем его и заполняем, чтобы решалось выражение  $f(x) = (2+x)^2$

```

%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите x: ',0
div: DB 'Результат: ',0
SECTION .bss
rez: RESB 80
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprintf

mov ecx,x
mov edx,80
call sread
mov eax,x
call atoi

add eax,2
mul eax
mov [rez],eax

mov eax, div
call sprintf
mov eax,[rez]
call iprintLF
call quit

```

Рис. 3.23: Заполняем файл

Компилируем программу и проверяем для  $x=2$

```
avyugayj@ubuntu:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
avyugayj@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
2
Результат: 16
```

Рис. 3.24: Проверяем работу программы

Проверяем для  $x=8$

```
avyugayj@ubuntu:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
8
Результат: 100
```

Рис. 3.25: Проверяем работу программы

## 4 Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.