

# **Отчёт по лабораторной работе №6**

**Дисциплина: Архитектура компьютера**

**Машков Илья Евгеньевич**

# 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	Символьные и численные данные в NASM . . . . .	6
3.2	Выполнение арифметических операций в NASM . . . . .	10
<b>4</b>	<b>Ответы на вопросы по программе</b>	<b>15</b>
4.1	Выполнение заданий для самостоятельной работы . . . . .	16
<b>5</b>	<b>Список литературы</b>	<b>18</b>

## Список иллюстраций

3.1	Директория lab06. . . . .	6
3.2	Создание рабочего файла. . . . .	6
3.3	Код программы. . . . .	7
3.4	Результат выполнения программы. . . . .	8
3.5	Результат второго выполнения программы. . . . .	8
3.6	Создание файла lab6-2.asm. . . . .	8
3.7	Код программы. . . . .	9
3.8	Результат выполнения программы. . . . .	9
3.9	Результат второго выполнения программы. . . . .	10
3.10	Результат третьего выполнения программы. . . . .	10
3.11	Создание файла lab6-3.asm. . . . .	10
3.12	Код программы. . . . .	11
3.13	Результат выполнения программы. . . . .	11
3.14	Изменённый код программы. . . . .	12
3.15	Результат второго выполнения программы. . . . .	12
3.16	Создание файла variant.asm. . . . .	13
3.17	Результат второго выполнения программы. . . . .	13
3.18	Код программы. . . . .	14
4.1	Строки кода, выводящие сообщение о моём варианте. . . . .	15
4.2	Строки кода, вычисляющие вариант. . . . .	15
4.3	Строки кода, выводящие результат вычислений. . . . .	16
4.4	Создание файла lab6-4.asm. . . . .	16
4.5	Получение и запуск исполняемого файла. . . . .	17
4.6	Результаты выполнений программы. . . . .	17

# 1 Цель работы

Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM.

## 2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

## 3 Выполнение лабораторной работы

### 3.1 Символьные и численные данные в NASM

Ещё в прошлой лабораторной работе я по ошибке создал папку **lab06** (рис. [3.1]).

```
iemashkov@iemashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ ls
CHANGELOG.md lab04 labs Makefile README.en.md template
config lab05 lamkdir prepare README.git-flow.md
COURSE lab06 LICENSE presentation README.md
iemashkov@iemashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 3.1: Директория lab06.

Далее я перехожу в эту директорию и создаю файл **lab6-1.asm** с помощью команды **'touch'** (Рис. [3.2]).

```
iemashkov@iemashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ touch lab6-1.asm
iemashkov@iemashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ls
lab6-1.asm
iemashkov@iemashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.2: Создание рабочего файла.

Затем я ввожу код в .asm файл (Рис. [3.3]).

```
%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 sprintf
call quit
```

Рис. 3.3: Код программы.

Создаю исполняемый файл и запускаю программу (Рис. [3.4]). В выводе программы я получаю символ 'j', который по системе ASCII соответствует сумме двоичных кодов символов 6 и 4.

```

ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 lab6-1 lab6-1.o
ld: невозможно найти lab6-1: Нет такого файла или каталога
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-1
j
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ 

```

Рис. 3.4: Результат выполнения программы.

Теперь я убираю кавычки у символов **6** и **4**, создаю исполняемый файл и запускаю программу (Рис. [3.5]). В выводе я получаю неотображающийся символ с кодом **10** - это **символ перевода строки**.

```

ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-1

ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ 

```

Рис. 3.5: Результат второго выполнения программы.

Создаю файл **lab6-2.asm** (Рис. [3.7]).

```

ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ touch lab6-2.asm
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2.asm
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ 

```

Рис. 3.6: Создание файла lab6-2.asm.

Ввожу код программы (Рис. [??]).



```

1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7
8  mov eax, '6'
9  mov ebx, '4'
10 add eax, ebx
11 call iprintLF
12 call quit

```

Рис. 3.7: Код программы.

Создаю исполняемый файл и запускаю программу (Рис. [3.9]). В выводе получаю число **106**, т.к. программа позволяет вывести число, а не символ, хоть и по-прежнему происходит сложение кодов символов **6** и **4**.

```

tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
106
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ 

```

Рис. 3.8: Результат выполнения программы.

Теперь убираю кавычки и заново создаю исполняемый файл и запускаю программу и в выводе получаю **10**, т.к. в этом случае программа складывает сами

числа, а не их коды (Рис. [??]).

```
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.9: Результат второго выполнения программы.

Затем меняю параметр **'iprintln'** на **'iprint'**, создаю исполняемый файл и запускаю программу (Рис. [3.10]). В выводе получаю то же число, но без символа переноса строки.

```
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.10: Результат третьего выполнения программы.

## 3.2 Выполнение арифметических операций в NASM

Создаю файл **lab6-3.asm** (при создании я забыл дать ему расширение .asm, но потом переименовал его) (Рис. [3.11]).

```
10ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ touch lab6-3
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2 lab6-2.asm lab6-2.o lab6-3
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.11: Создание файла lab6-3.asm.

Теперь ввожу в него код программы, который будет вычислять выражение  $f(x) = (5 \cdot 2 + 3) / 3$  (Рис. [3.12]).

```

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8
9 ; ---- Вычисление выражения
10
11 mov eax,5 ; EAX=5
12 mov ebx,2 ; EBX=2
13 mul ebx ; EAX=EAX*EBX
14 add eax,3 ; EAX=EAX+3
15 xor edx,edx ; обнуляем EDX для корректной работы div
16 mov ebx,3 ; EBX=3
17 div ebx ; EAX=EAX/3, EDX=остаток от деления
18 mov edi,eax ; запись результата вычисления в 'edi'
19
20 ; ---- Вывод результата на экран
21
22 mov eax,div ; вызов подпрограммы печати
23 call sprint ; сообщения 'Результат: '
24 mov eax,edi ; вызов подпрограммы печати значения
25 call iprintLF ; из 'edi' в виде символов
26 mov eax,rem ; вызов подпрограммы печати
27 call sprint ; сообщения 'Остаток от деления: '
28 mov eax,edx ; вызов подпрограммы печати значения
29 call iprintLF ; из 'edx' (остаток) в виде символов
30 call quit ; вызов подпрограммы завершения

```

Рис. 3.12: Код программы.

Далее создаю исполняемый файл и запускаю программу (Рис. [3.13]).

```

tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-3.asm
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -n elf_i386 -o lab6-3 lab6-3.o
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ 

```

Рис. 3.13: Результат выполнения программы.

Изменяю код программы так, чтобы она вычисляла выражение  $f(x) = (4*6 + 2)/5$  (Рис. [3.14]).

```

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8
9 ; ---- Вычисление выражения
10
11 mov eax,4 ; EAX=4
12 mov ebx,6 ; EBX=6
13 mul ebx ; EAX=EAX*EBX
14 add eax,2 ; EAX=EAX+2
15 xor edx,edx ; обнуляем EDX для корректной работы div
16 mov ebx,5 ; EBX=5
17 div ebx ; EAX=EAX/5, EDX=остаток от деления
18 mov edi,eax ; запись результата вычисления в 'edi'
19
20 ; ---- Вывод результата на экран
21
22 mov eax,div ; вызов подпрограммы печати
23 call sprint ; сообщения 'Результат: '
24 mov eax,edi ; вызов подпрограммы печати значения
25 call iprintLF ; из 'edi' в виде символов
26 mov eax,rem ; вызов подпрограммы печати
27 call sprint ; сообщения 'Остаток от деления: '
28 mov eax,edx ; вызов подпрограммы печати значения
29 call iprintLF ; из 'edx' (остаток) в виде символов
30 call quit ; вызов подпрограммы завершения

```

Рис. 3.14: Изменённый код программы.

Создаю исполняемый файл и запускаю программу (Рис. [3.15]).

```

ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-3.asm
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ 

```

Рис. 3.15: Результат второго выполнения программы.

Создаю файл **variant.asm** с помощью команды **'touch'** (Рис. [3.16]).

```

lemashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ touch variant.asm
lemashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2 lab6-2.asm lab6-2.o lab6-3 lab6-3.asm lab6-3.o variant.asm
lemashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$

```

Рис. 3.16: Создание файла variant.asm.

Теперь ввожу в него код программы, которая по номеру студенческого должна мне выдать номер моего варианта (Рис. [3.17]).

```

#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 sprintf

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 sprintf
mov eax, edx
call iprintLF
call quit

```

Рис. 3.17: Результат второго выполнения программы.

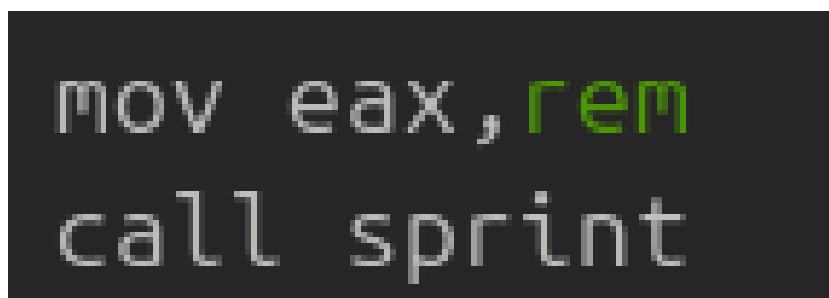
Создаю исполняемый файл и запускаю его (Рис. [4.1]). Мой вариант - 5.

```
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ nasm -f elf variant.asm
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ ld -m elf_i386 -o variant variant.o
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ ./variant
Введите № студенческого билета:
1132231984
Ваш вариант: 5
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$
```

Рис. 3.18: Код программы.

## 4 Ответы на вопросы по программе

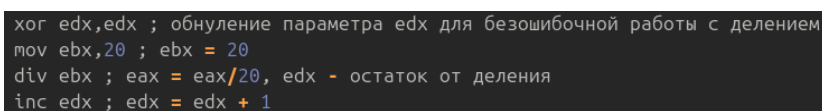
1. За вывод сообщения “Ваш вариант:” отвечают эти строки кода (Рис. [??]):



```
mov eax, gem
call sprint
```

Рис. 4.1: Строки кода, выводящие сообщение о моём варианте.

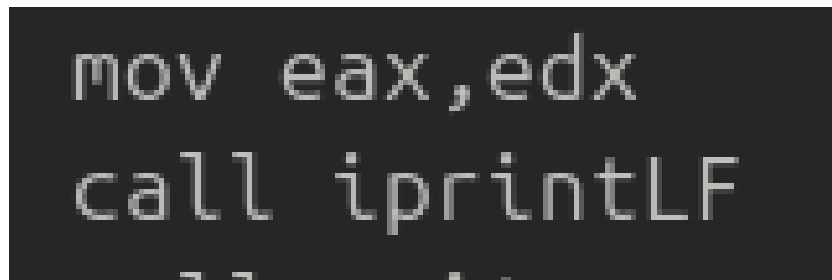
2. ‘**mov ecx, x**’ используется для того, чтобы внести адрес вводимой строки ‘**x**’ в регистр ‘**ecx**’. ‘**mov edx, 80**’ - запись в регистр ‘**edx**’ длины вводимой строки. ‘**call sread**’ - вызов подпрограммы из внешнего файла, которая обеспечивает ввод сообщения с клавиатуры.
3. ‘**call atoi**’ - подпрограмма из внешнего файла, преобразующая код символа в целое число и записывающая его в регистр ‘**eax**’.
4. За вычисление варианта отвечают строки (Рис. [4.2]):



```
xor edx,edx ; обнуление параметра edx для безошибочной работы с делением
mov ebx,20 ; ebx = 20
div ebx ; eax = eax/20, edx - остаток от деления
inc edx ; edx = edx + 1
```

Рис. 4.2: Строки кода, вычисляющие вариант.

5. При выполнении инструкции '**div ebx**' остаток от деления записывается в регистр '**edx**'.
6. Инструкция '**inc edx**' увеличивает значение регистра '**edx**' на 1
7. За вывод на экран результатов вычислений отвечают строки (Рис. [4.3]):

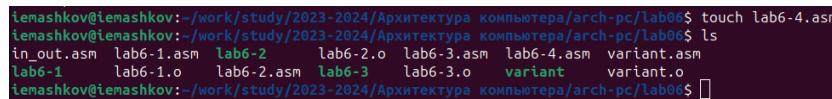


```
mov eax,edx
call iprintLF
```

Рис. 4.3: Строки кода, выводющие результат вычислений.

## 4.1 Выполнение заданий для самостоятельной работы

1. Создаю файл **lab6-4.asm** (Рис. [4.4]).



```
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ touch lab6-4.asm
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm  lab6-2      lab6-2.o    lab6-3.asm  lab6-4.asm  variant.asm
lab6-1      lab6-1.o    lab6-2.asm  lab6-3      lab6-3.o    variant     variant.o
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06$
```

Рис. 4.4: Создание файла lab6-4.asm.

Ввожу код программы, которая будет вычислять выражение ' $f(x) = (9x - 8)/8$ '. Именно это выражение стоит под номером 5 в таблице (Рис. [4.5]).



```

%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите значение x: ', 0
rem: DB 'Результат: ', 0
SECTION .bss
x: RESB 80 ; Переменная, значение которой запрашивается у пользователя
SECTION .text
GLOBAL _start
_start:

; ---- Вычисление выражения (9x - 8)/8
mov eax,msg ;
call sprint ;
mov ecx, x ;
mov edx, 80 ;
call sread ;
mov eax,x ; вызов подпрограммы преобразования
call atoi ;
mov ebx,9 ; EBX=9
mul ebx ; EAX=EAX*EBX
sub eax,8 ; EAX=EAX-8
xor edx,edx ;
mov ebx,8 ; EBX=8
div ebx ; EAX=EAX/8
mov edi,eax ; запись результата вычисления в 'edi'

; ---- Вывод результата на экран
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения

```

Рис. 4.5: Получение и запуск исполняемого файла.

Создаю исполняемый файл и запускаю его со значением '8', а потом и со значением '64' (Рис. [4.6]).

```

ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab6$ nasm -f elf lab6-4.asm
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab6$ ld -m elf_i386 -o lab6-4 lab6-4.o
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab6$ ./lab6-4
Введите значение x: 8
Результат: 8
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab6$ ./lab6-4
Введите значение x: 64
Результат: 71
ienashkov@ienashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab6$ 

```

Рис. 4.6: Результаты выполнений программы.

Эти результаты я проверил, а это значит, что программа отработала верно. #  
Выводы

При выполнении данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.

## 5 Список литературы

Архитектура ЭВМ

Таблица кодов ASCII