

# **РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук Кафедра  
прикладной информатики и теории вероятностей**

Архипов Олег Константинович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
2.1	Символьные и численные данные в NASM . . . . .	5
2.2	Арифметические операции в NASM . . . . .	12
<b>3</b>	<b>Самостоятельная работа</b>	<b>20</b>
<b>4</b>	<b>Выводы</b>	<b>23</b>
	<b>Список литературы</b>	<b>24</b>

## Список иллюстраций

2.1	Каталог и файл asm для новой ЛР . . . . .	5
2.2	Файл in_out.asm . . . . .	6
2.3	Программа вывода значения регистра eax . . . . .	7
2.4	Исполнение lab6-1.asm . . . . .	7
2.5	Изменение файла lab6-1.asm . . . . .	8
2.6	Исполнение и результат обновленного файла lab6-1.asm . . . . .	9
2.7	Файл lab6-2.asm . . . . .	9
2.8	Текст программы в файле lab6-2.asm . . . . .	9
2.9	Исполнение файла lab6-2 . . . . .	10
2.10	Изменение файла lab6-2.asm . . . . .	10
2.11	Исполнение измененного файла lab6-2 . . . . .	11
2.12	Замена iprintLF на iprint в файле lab6-2.asm . . . . .	11
2.13	Запуск файла lab6-2 без символа перевода строки . . . . .	11
2.14	Файл lab6-3.asm . . . . .	12
2.15	Текст файла lab6-3.asm . . . . .	13
2.16	Результат работы файла lab6-3 . . . . .	14
2.17	Измененная часть файла lab6-3.asm . . . . .	15
2.18	Результат работы измененного файла lab6-3 . . . . .	15
2.19	Файл variant.asm . . . . .	16
2.20	Текст в variant.asm . . . . .	17
2.21	Запуск variant . . . . .	18
2.22	Ввод номера студ. билета . . . . .	18
2.23	Мой вариант . . . . .	18
3.1	Файл для вычисления самостоятельного задания . . . . .	20
3.2	Код программы . . . . .	21
3.3	Результат работы программы . . . . .	22
3.4	Результат работы программы с альтернативным значением . . . . .	22

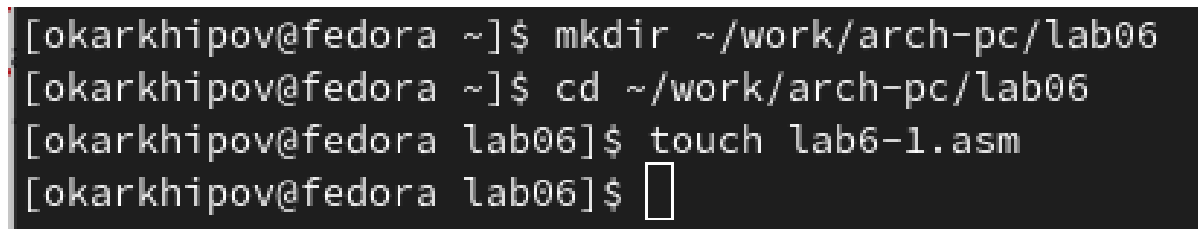
# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Выполнение лабораторной работы

### 2.1 Символьные и численные данные в NASM

Создаю каталог для программ ЛР №6, перехожу в него и создаю там файл lab6-1.asm (рис. 2.1).

A terminal window with a dark background and light-colored text. It shows four lines of commands and their outputs. The first line creates a directory, the second changes to it, the third creates a new file, and the fourth shows the prompt in the new directory.

```
[okarkhipov@fedora ~]$ mkdir ~/work/arch-pc/lab06
[okarkhipov@fedora ~]$ cd ~/work/arch-pc/lab06
[okarkhipov@fedora lab06]$ touch lab6-1.asm
[okarkhipov@fedora lab06]$
```

Рис. 2.1: Каталог и файл asm для новой ЛР

Копирую файл in\_out.asm в новую директорию ~/work/arch-pc/lab06 (рис. 2.2).

Левая панель			Файл	Команда	Настройки	Правая панель		
<- ~/work/arch-pc/lab06						<- ~/work/arch-pc/lab05		
.и			Имя	Размер	Время правки	.и		
/..			-ВВЕРХ-	офт 28	21:00	/..		
in_out.asm			3942	офт 26	21:19	in_out.asm		
lab6-1.asm			0	офт 28	21:00	lab5-1.asm		
						lab5-1.o		
						*lab5-12		
						lab5-12.asm		
						lab5-12.o		
						*lab5-2		
						lab5-2.asm		
						lab5-2.o		
						*lab5-22		
						lab5-22.asm		
						lab5-22.o		

Рис. 2.2: Файл in\_out.asm

Прописываю в файле lab6-1.asm текст программы, которая запишет в регистр еах символ 6, в регистр ебх - символ 4, далее сложит эти два значения и выведет результат (для последнего запишет значение регистра еах в переменную buf1 (в квадратных скобках, т.к. это переменная) и затем в регистр еах - адрес переменной buf1 (адрес - без скобок) (рис. 2.3).

```

GNU nano 7.2      /hc
#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 sprintfLF
call quit

```

Рис. 2.3: Программа вывода значения регистра eax

Создаю и запускаю исполняемый файл и действительно получаю на выходе символ j, т.к. складываются двоичные представления кодов символов 6 и 4, после чего выводится символ, соответствующий этому суммарному коду (рис. 2.4).

```

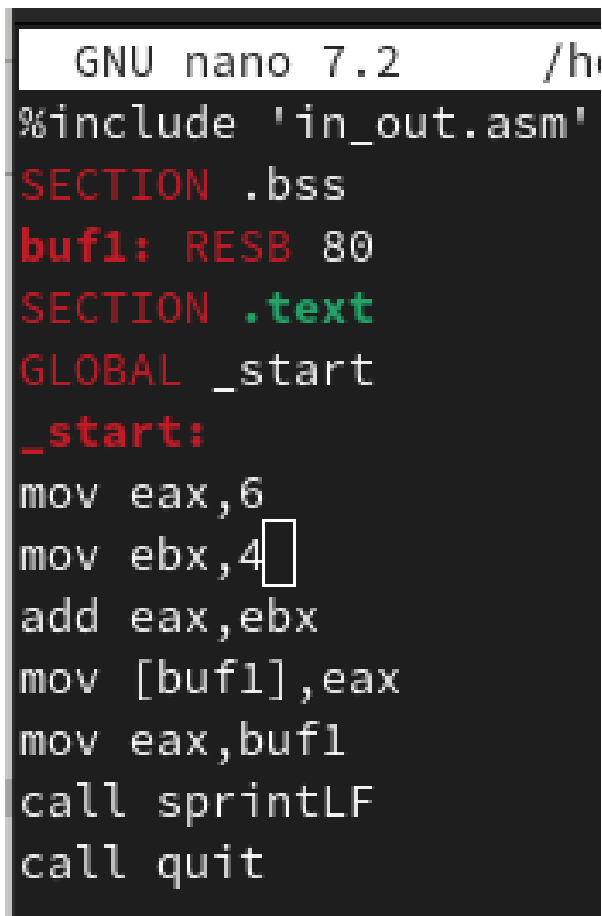
[okarkhipov@fedora lab06]$ nasm -f elf lab6-1.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[okarkhipov@fedora lab06]$ ./lab6-1
j
[okarkhipov@fedora lab06]$ 

```

Рис. 2.4: Исполнение lab6-1.asm

Изменяю mov eax,'6' и mov ebx,'4' на mov eax,6 и mov ebx,4 в исходном коде

(рис. 2.5).



```
GNU nano 7.2 /h
#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 printf
call _exit
```

Рис. 2.5: Изменение файла lab6-1.asm

Снова исполняю обновленный файл и получаю символ с кодом 10, или LF т.е. символ перевода строки, отображаться он не будет (если не учитывать пропущенную строку) (рис. 2.6).



```
[okarkhipov@fedora lab06]$ nasm -f elf lab6-1.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[okarkhipov@fedora lab06]$ ./lab6-1

[okarkhipov@fedora lab06]$
```

Рис. 2.6: Исполнение и результат обновленного файла lab6-1.asm

Создаю новый файл lab6-2.asm в той же директории (рис. 2.7).

```
[okarkhipov@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-2.asm
[okarkhipov@fedora lab06]$
```

Рис. 2.7: Файл lab6-2.asm

Ввожу следующую программу (рис. 2.8).

```
GNU nano 7.2 /home/okarkhipov/work/arch-pc/lab06/lab6-2.asm
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 2.8: Текст программы в файле lab6-2.asm

Компилирую исполняемый файл для lab6-2.asm и запускаю его, получаю число 106, которое является суммой кодов исходных символов (рис. 2.9).

```
[okarkhipov@fedora lab06]$ nasm -f elf lab6-2.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[okarkhipov@fedora lab06]$ ./lab6-2
106
[okarkhipov@fedora lab06]$
```

Рис. 2.9: Исполнение файла lab6-2

Заменяю в lab6-2.asm `mov eax,'6'` и `mov ebx,'4'` на `mov eax,6` и `mov ebx,4` как и в предыдущем файле lab6-1.asm (рис. 2.10).

```
GNU nano 7.2 /h
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 2.10: Изменение файла lab6-2.asm

Создаю объектный и исполняемый файлы программы и запускаю последний, получаю число 10 (рис. 2.11).

```
[okarkhipov@fedora lab06]$ nasm -f elf lab6-2.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[okarkhipov@fedora lab06]$ ./lab6-2
10
[okarkhipov@fedora lab06]$
```

Рис. 2.11: Исполнение измененного файла lab6-2

Заменяю функцию `iprintLF` на `iprint` и снова создаю исполняемый файл, после чего запускаю его, получаю то же число 10, без символа перевода строки, что и характеризует отличие функций `iprintLF` на `iprint` (рис. 2.12-2.13).

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

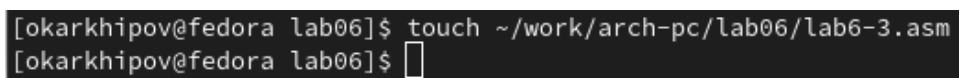
Рис. 2.12: Замена `iprintLF` на `iprint` в файле lab6-2.asm

```
[okarkhipov@fedora lab06]$ nasm -f elf lab6-2.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[okarkhipov@fedora lab06]$ ./lab6-2
10[okarkhipov@fedora lab06]$
```

Рис. 2.13: Запуск файла lab6-2 без символа перевода строки

## 2.2 Арифметические операции в NASM

Создаю файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 2.14).

A terminal window with a dark background. The first line shows the command 'touch ~/work/arch-pc/lab06/lab6-3.asm' being executed. The second line shows the prompt '[okarkhipov@fedora lab06]\$' with a cursor. The terminal has a title bar with 'okarkhipov@fedora lab06' and standard window controls.

```
[okarkhipov@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-3.asm  
[okarkhipov@fedora lab06]$
```

Рис. 2.14: Файл lab6-3.asm

Ввожу текст программы в файл lab6-3.asm (рис. 2.15).

```

%include 'in_out.asm'

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
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
call iprintLF  ; значения 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

```

Рис. 2.15: Текст файла lab6-3.asm

Создаю файлы: объектный lab6-3.o и исполняемый lab6-3 и запускаю программу, получаю 4 и 1 в остатке, как и должно быть (рис. 2.16).

```
[okarkhipov@fedora lab06]$ nasm -f elf lab6-3.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[okarkhipov@fedora lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
[okarkhipov@fedora lab06]$
```

Рис. 2.16: Результат работы файла lab6-3

Изменяю текст программы для вычисления выражения

$$f(x) = (4 * 6 + 2) / 5$$

(можно сравнить рис. 2.17 и рис. 2.15).

```

GNU nano 7.2 /home/okarkhipov/work/arch-pc/lab06/1
#include 'in_out.asm'

SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:

; Вычисление выражения

mov eax,4      ; EAX=4
mov ebx,6      ; EBX=6
mul ebx        ; EAX=EAX*EBX
add eax,2      ; EAX=EAX+2
xor edx,edx    ; Обнуляем EDX для корректной работы div
mov ebx,5      ; EBX=5
div ebx        ; EAX=EAX/5, EDX=остаток
mov edi,eax    ; Запись результата вычисления в 'edi'

```

Рис. 2.17: Измененная часть файла lab6-3.asm

Проверяю результат работы программы после внесенных изменений (рис. 2.18).

```

[okarkhipov@fedora lab06]$ nasm -f elf lab6-3.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[okarkhipov@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[okarkhipov@fedora lab06]$

```

Рис. 2.18: Результат работы измененного файла lab6-3

Для следующего задания по вычислению варианта задания по номеру студенческого билета создаю файл `variant.asm` в каталоге `~/work/arch-pc/lab06` (рис. 2.19).

```
[okarkhipov@fedora lab06]$ touch ~/work/arch-pc/lab06/variant.asm  
[okarkhipov@fedora lab06]$
```

Рис. 2.19: Файл `variant.asm`

Ввожу текст программы, которая запросит номер студ. билета, затем вычислит вариант по формуле  $(S_n \bmod 20) + 1$ , т.е. остаток от деления № билета на 20 и ещё плюс 1, после чего выведет ответ на экран (рис. 2.20).



```
GNU nano 7.2 /home/okarkhipov/work/arch-p
#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. 2.20: Текст в variant.asm

Создаю исполняемый файл и запускаю его (рис. 2.21), ввожу номер билета (рис. 2.22) и получаю в ответе 4, что верно (рис. 2.23).

```
[okarkhipov@fedora lab06]$ nasm -f elf variant.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[okarkhipov@fedora lab06]$ ./variant
Введите № студенческого билета:

```

Рис. 2.21: Запуск variant

```
[okarkhipov@fedora lab06]$ nasm -f elf variant.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[okarkhipov@fedora lab06]$ ./variant
Введите № студенческого билета:
1132236063
```

Рис. 2.22: Ввод номера студ. билета

```
[okarkhipov@fedora lab06]$ nasm -f elf variant.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[okarkhipov@fedora lab06]$ ./variant
Введите № студенческого билета:
1132236063
Ваш вариант: 4
[okarkhipov@fedora lab06]$
```

Рис. 2.23: Мой вариант

- 1) В этой программе за вывод на экран сообщения 'Ваш вариант:' отвечают строки:

```
rem: DB 'Ваш вариант: ',0
```

.

```
.  
.  
mov eax,rem  
call sprint
```

2) Строки:

```
mov ecx, x  
mov edx, 80  
call sread
```

отвечают за помещение адреса x , где x - вводимая строка (в данном случае номер билета) в регистр ecx , а в регистр edx - максимальную длину этой строки, call sread - вызов подпрограммы внешнего файла in\_out.asm , отвечающей за ввод сообщения с клавиатуры.

- 3) Инструкция “call atoi” используется для вызова функции преобразования ASCII - кода символа в целое число, после чего записывает результат в регистр eax .
- 4) За вычисление варианта отвечают команды: “xor edx,edx” - обнуление регистра edx , “mov ebx,20” - присвоение регистру ebx значения 20, “div ebx” - деление на 20 (значение регистра ebx) значения соответствующего регистра eax , в edx - остаток, “inc edx” - увеличение остатка, записанного в регистр edx на 1.
- 5) Как уже упоминалось выше остаток от деления при выполнении инструкции “div ebx” записывается в регистр edx .
- 6) Инструкция “inc edx” нужна для увеличения значения в регистре edx на 1.
- 7) За вывод на экран результата вычислений отвечают команды “mov eax,edx” и “call iprintLF”.

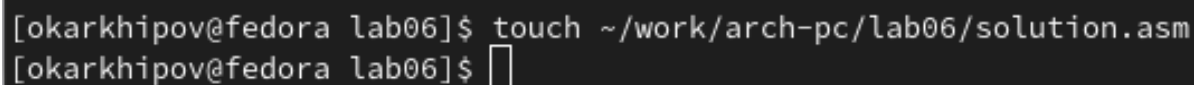
### 3 Самостоятельная работа

Мой номер варианта 4, значит беру функцию вида

$$f(x) = (4/3) * (x - 1) + 5,$$

где  $x_1 = 4, x_2 = 10$ .

Создаю в директории `~/work/arch-pc/lab06` файл `solution.asm` (рис. 3.1).

A terminal window with a dark background and light-colored text. The prompt is `[okarkhipov@fedora lab06]$`. The command `touch ~/work/arch-pc/lab06/solution.asm` is entered. The prompt is repeated on the next line with a cursor character (a small square) after it.

```
[okarkhipov@fedora lab06]$ touch ~/work/arch-pc/lab06/solution.asm
[okarkhipov@fedora lab06]$ █
```

Рис. 3.1: Файл для вычисления самостоятельного задания

Далее прописываю код. Он достаточно подробно описан на рис. 3.2.

```

GNU nano 7.2 /home/okarkhipov/work/arch-
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите переменную x: ',0
rem: DB 'Ответ: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:

mov eax, msg ; Вывод сообщения
call sprint ; 'Введите переменную x:'

; Действие 1: присвоение переменной x
mov ecx, x ;
mov edx, 80 ;
call sread ; Ввод переменной x
mov eax,x ; Вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`

; Действие 2: вычитание единицы
sub eax,1 ; eax=eax-1
mov edi,eax ; Освобождаю регистр eax

; Действие 3: деление 4/3
xor edx,edx ; Обнуляю edx для корректной работы div
mov ebx,3 ;
mov eax,4 ; eax=eax/ebx , т.е. 4 на 3,
div ebx ; остаток записывается в edx

; Действие 4: умножение результата Д3 на результат Д2
mul edi ; eax=edi*eax

; Действие 5: сумма результата Д4 и числа 5
mov ebx,5
add eax,ebx ; eax=eax+ebx=eax+5

; Вывод результата
mov edi,eax ; Перемещаю результат в регистр edi
mov eax,rem ; Вызов подпрограммы печати
call sprint ; сообщения 'Ответ: '
mov eax,edi ; Вызов подпрограммы печати значения
call iprintLF ; из 'edi' (результат) в виде символов
call quit

```

Рис. 3.2: Код программы

Создаю исполняемый файл и запускаю его, в качестве переменной ввожу 4, получаю ответ 8. Это результат без учета остатка в операции деления (рис. 3.3).

```
[okarkhipov@fedora lab06]$ nasm -f elf solution.asm
[okarkhipov@fedora lab06]$ ld -m elf_i386 solution.o -o solution
[okarkhipov@fedora lab06]$ ./solution
Введите переменную x: 4
Ответ: 8
```

Рис. 3.3: Результат работы программы

Запускаю программу еще раз с переменной 10, чтобы удостовериться в правильности работы программы, результат верный (рис. 3.4).

```
[okarkhipov@fedora lab06]$ ./solution
Введите переменную x: 10
Ответ: 14
```

Рис. 3.4: Результат работы программы с альтернативным значением

## 4 Выводы

Были освоены особенности арифметических операций в ассемблере NASM.

## **Список литературы**