

# **Лабораторная работа №7**

**Арифметические операции в NASM**

Киньябаева Аиша Иделевна

# 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>
<b>4</b>	<b>Выводы</b>	<b>14</b>

## Список иллюстраций

3.1	lab7-1.asm_вывод . . . . .	6
3.2	код . . . . .	6
3.3	lab7-1.asm_вывод . . . . .	7
3.4	lab7-2.asm_вывод . . . . .	7
3.5	lab7-2.asm_вывод . . . . .	7
3.6	код . . . . .	7
3.7	lab7-2.asm_вывод . . . . .	8
3.8	код . . . . .	8
3.9	lab7-3.asm_вывод . . . . .	9
3.10	код . . . . .	9
3.11	lab7-3.asm_вывод . . . . .	10
3.12	код . . . . .	10
3.13	variant.asm_вывод . . . . .	10
3.14	код . . . . .	12
3.15	вывод . . . . .	13

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

## 2 Задание

Освоение арифметических инструкций языка ассемблера NASM и написание программ для вычисления заданных выражений.

### 3 Выполнение лабораторной работы

Написание первой программы lab7-1.asm, которая работает с числами, представленными через ASCII. В результате мы видим знак “j”, так как программа сложила двоичные коды введенных нами значений. (рис. 3.1)

```
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ ./lab7-1
j
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 3.1: lab7-1.asm\_вывод

Далее мы преобразовываем программу и заменяем значения с кавычками на обычные цифры. В ответе получаем пустую строку, что соответствует символу STX (рис. 3.2), (рис. [fig:fig3])

```
%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. 3.2: код

```

aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ vim lab7-1.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ nasm -f elf lab7-1.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ ./lab7-1

```

Рис. 3.3: lab7-1.asm\_вывод

Пишем еще одну программу lab7-2.asm с небольшими изменениями. В результате получаем число 106, т.к. данная программа преобразовывает полученный код в число (рис. 3.4)

```

aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ vim lab7-2.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ nasm -f elf lab7-2.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ ./lab7-2
106
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$

```

Рис. 3.4: lab7-2.asm\_вывод

Опять же меняем значения с кавычками на обычные числа и получаем нужный нам ответ в виде суммы введенных чисел (рис. 3.5)

```

aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ vim lab7-2.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ nasm -f elf lab7-2.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$ ./lab7-2
10
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/Lab07$

```

Рис. 3.5: lab7-2.asm\_вывод

Поменяв функцию `iprintLF` на `iprint` видим, как программа выводит значение без последующего переноса строки, из-за чего строка ввода “налезла” на вывод (рис. 3.6), (рис. 3.7)

```

#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprint

call quit
~

```

Рис. 3.6: код

```

aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ vim lab7-2.asm
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
10aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$

```

Рис. 3.7: lab7-2.asm\_вывод

Пишем программу, выполняющую более сложные арифметические операции  
(а конкретно  $(5 \times 2 + 3) / 3$ ) (рис. 3.8), (рис. 3.9)

```

#include 'in_out.asm'

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

SECTION .text
GLOBAL _start
_start:

mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
~

```

Рис. 3.8: код



```

aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ vim lab7-3.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ ./lab7-3
Результат: 4
Остаток от деления: 1
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ █

```

Рис. 3.9: lab7-3.asm\_вывод

Пишем аналогичную программу, вычисляющую  $(4 \times 6 + 2) / 5$  (рис. 3.10), (рис. 3.11)

```

#include 'in_out.asm'

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

SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
~

```

Рис. 3.10: код

```

aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ vim lab7-3.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ ./lab7-3
Результат: 5
Остаток от деления: 1
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$

```

Рис. 3.11: lab7-3.asm\_вывод

Далее пишем программу, которая выдает номер варианта в соответствии с остатком от деления номера студенческого билета на 20 (у меня вышел 4 вариант) (рис. 3.12), (рис. 3.13)

```

#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
call atoi

xor edx,edx
mov ebx,20
div ebx
inc edx

mov eax,rem
call sprintf
mov eax,edx
call iprintLF

call quit
~

```

Рис. 3.12: код

```

aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ touch variant.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ vim variant.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf variant.asm
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o variant variant.o
aikinjabaeve@aikinjabaeve-VirtualBox:~/work/arch-pc/lab07$ ./variant
Введите № студенческого билета:
1132226463
Ваш вариант: 4

```

Рис. 3.13: variant.asm\_вывод

## ОТВЕТЫ НА ВОПРОСЫ:

1. За вывод на экран сообщения “Ваш вариант:” отвечают строки

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

2. Данные инструкции используются для:

```
mov ecx, x - записывает адрес вводимого “x” в ecx  
mov edx, 80 - задает длину вводимой строки  
call sread - в целом позволяет вводить строку с клавиатуры
```

3. Инструкция “call atoi” используется для преобразования ASCII кода в число

4. За вычисление варианта отвечают строки:

```
xor edx,edx  
mov ebx,20  
div ebx  
inc edx
```

5. Остаток от деления записывается в регистр edx

6. Данная инструкция используется для добавления единицы к значению edx

7. За вывод результата вычислений отвечают строки:

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

## САМОСТОЯТЕЛЬНАЯ РАБОТА

Из полученного ранее варианта 4 пишу код, вычисляющий выражение  $\frac{4}{3}(x-1)+5$  (рис. 3.14), (рис. 3.15)

```

#include 'in_out.asm'

SECTION .data
exm: DB 'Пример: 4/3(x-1)+5',0
msg: DB 'Введите значение: ',0
rem: DB 'Ответ: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax,exm
call sprintf

mov eax,msg
call sprintf

mov ecx, x
mov edx, 80
call sread

mov eax,x
call atoi

dec eax

mov ebx,4
mul ebx

xor edx,edx
mov ebx,3
div ebx

add eax,5
mov edi,eax

mov eax,rem
call sprintf
mov eax,edi
call iprintf

call quit
~

```

Рис. 3.14: код

```
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ vim mywork.asm
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf mywork.asm
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o mywork mywork.o
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ ./mywork
Пример: 4/3(x-1)+5
Введите значение:
4
Ответ: 9
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$ ./mywork
Пример: 4/3(x-1)+5
Введите значение:
10
Ответ: 17
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 3.15: вывод

Загрузка всех файлов на Git

Далее создается отчет по 7й лабораторной работе с помощью Markdown.

## 4 Выводы

В ходе данной лабораторной работы были приобретены навыки работы с языком ассемблера NASM и изучены некоторые команды для вычисления арифметических операций.