

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**ОТЧЕТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

*дисциплина: Архитектура компьютера*

Студент: Истомин Андрей

Группа: НБИбд-01-25

**МОСКВА**

2025 г.

## **СОДЕРЖАНИЕ**

- 1. Цель работы**
- 2. Теоретическое введение**
- 3. Ход работы**
- 4. Самостоятельная работа**
- 5. Выводы**

**Цель:** Освоение арифметических инструкций языка ассемблера NASM, изучение способов адресации данных и освоение преобразования между ASCII-символами и числовыми значениями.

## ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

### Способы адресации в NASM

- Регистровая адресация – операнды хранятся в регистрах
- Непосредственная адресация – значение операнда задаётся в команде
- Адресация памяти – операнд указывает на адрес в памяти

### Основные арифметические команды

- add – сложение
- sub – вычитание
- inc, dec – инкремент/декремент
- mul, imul – умножение беззнаковое/знаковое
- div, idiv – деление беззнаковое/знаковое
- neg – изменение знака

## Ход работы

1. Создаю каталог для программ лабораторной работы №6, перехожу в него и создаю файл lab 6-1.asm (рис.1.1)

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

(рис.1.1)

2. Ввожу в файл lab6-1.asm текст программы вывода значения регистра eax (рис.2.1)

```
GNU nano 7.2                                     lab6-1.asm *
%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.2.1)

3. Создаю исполняемый файл и запускаю его (рис.3.1)

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

(рис.3.1)

4. Изменяю текст программы и вместо символов, записываю в регистры числа.  
(рис.4.1)

```
GNU nano 7.2                                lab6-1.asm
%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.4.1)

5. Создаю исполняемый файл и запускаю его (рис.5.1)

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

(рис.5.1)

6. Создаю файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввожу в него текст программы (рис.6.1)

```
GNU nano 7.2                                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.6.1)

7. Создаю исполняемый файл и запускаю его (рис.7.1)

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

(рис.7.1)

8. Изменяю символы на числа, создаю исполняемый файл и запускаю его (рис.8.1)

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

(рис.8.1)

9. Создаю файл lab6-3.asm в каталоге ~/work/arch-pc/lab06, ввожу текст программы (рис.9.1)

```
GNU nano 7.2                                     lab6-3.asm
%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.9.1)

10. Создаю исполняемый файл и запускаю его (рис.10.1)

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

(рис.10.1)

11. Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06 и ввожу программу вычисления варианта задания по номеру студенческого билета (рис.11.1)

```
GNU nano 7.2                                         variant.asm *
%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
    call atoi

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

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

    call quit
```

(рис.11.1)

12. Создаю исполняемый файл и запускаю его (рис.12.1)

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

(рис.12.1)

13. Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06 и ввожу текст программы (рис.13.1)

```
GNU nano 7.2                                         variant.asm *
%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
    call atoi

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

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

    call quit
```

(рис.13.1)

14.Создаю исполняемый файл и запускаю его (рис.14.1)

```
aeistomin@aeistomin:~/work/arch-pc/lab06$ nasm -f elf variant.asm
aeistomin@aeistomin:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
aeistomin@aeistomin:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032251949
Ваш вариант: 10
```

(рис.14.1)

Ответы на вопросы:

1.mov eax, rem и call sprint

2.Чтение строки с клавиатуры в буфер x

3.Преобразование ASCII-строки в число

4.Строки от xor edx, edx до inc edx

5.В регистр edx

6.Для увеличения остатка на 1 (т.к. варианты с 1)

7.mov eax, rem ... call iprintLF

## Самостоятельная работа

1. Создаю файл sam\_rab.asm и ввожу в него программу вычисления выражения  $y=f(x)$  для варианта 1 (рис.1.1)

```
GNU nano 7.2                                         sam_rab.asm *
%include 'in_out.asm'

SECTION .data
    msg_x    DB 'Введите x: ', 0
    res_msg  DB 'Результат: ', 0

SECTION .bss
    x RESB 10

SECTION .text
    GLOBAL _start
_start:
    mov eax, msg_x
    call sprint

    mov ecx, x
    mov edx, 10
    call sread

    mov eax, x
    call atoi

    mov ebx, 2
    mul ebx

    div ebx
    inc edx

    mov eax, res
    call sprint
    mov eax, edx
    call iprintLF

    call quit
```

(рис.1.1)

2. Создаю исполняемый файл (рис.2.1)

```
aeistomin@aeistomin:~/work/arch-pc/lab06$ touch sam_rab.asm
aeistomin@aeistomin:~/work/arch-pc/lab06$ nano sam_rab.asm
aeistomin@aeistomin:~/work/arch-pc/lab06$ nasm -f elf sam_rab.asm
aeistomin@aeistomin:~/work/arch-pc/lab06$ ld -m elf_i386 -o sam_rab sam_rab.o
```

(рис.2.1)

3. Запускаю исполняемый файл (рис.3.1)

```
aeistomin@aeistomin:~/work/arch-pc/lab06$ ./sam_gab
Введите x: 1
Результат: 4
aeistomin@aeistomin:~/work/arch-pc/lab06$ ./sam_gab
Введите x: 10
Результат: 10
aeistomin@aeistomin:~/work/arch-pc/lab06$
```

(рис.3.1)

## Выводы

В ходе выполнения лабораторной работы:

-Освоены арифметические инструкции NASM: изучены команды сложения, вычитания, умножения и деления, а также команды инкремента и декремента.

-Изучены способы адресации: освоена регистровая, непосредственная и адресация памяти.

-Освоено преобразование данных: изучены различия между символьными и числовыми данными, освоено использование функций iprint, iprintLF и atoi для преобразования между ASCII-символами и числами.

-Разработаны программы: созданы программы для выполнения арифметических операций, вычисления выражений и обработка пользовательского ввода.

-Выполнена самостоятельная работа: написана программа для вычисления заданной функции с вводом параметра от пользователя.

-Результаты всех программ соответствуют ожидаемым значениям, что подтверждает правильность выполнения работы и понимание материала.