

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**  
**Факультет физико-математических и естественных наук**  
**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**  
*дисциплина: Архитектура компьютеров и операционные системы*

Студент: Галиев Самир Салаватович

Группа: НКАбд-02-25

**МОСКВА**

2025 г.

## **Оглавление**

1. Цель работы - стр. 3
2. Результаты выполнения лабораторной работы - стр. 3
  - 2.1 Символьные и численные данные в NASM- стр. 3
  - 2.2 Выполнение арифметических операций - стр. 6
3. Задание для самостоятельной работы - стр. 8
4. Выводы - стр. 10
5. Список литературы - стр. 11

## 1. Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2. Результаты выполнения лабораторной работы

### 2.1 Символьные и численные данные в NASM

**Задание:** Изучение различий между работой с символьными и численными данными в NASM, освоение преобразования данных.

**Выполнение:**

Программа lab6-1.asm — Сложение символьных данных

```
GNU nano 8.6
#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
■
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % cp ../lab05/in_out.asm .
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % touch lab6-1.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nano lab6-1.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nasm -f elf32 lab6-1.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ld -m elf_i386 -o lab6-1 lab6-1.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ./lab6-1
j
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 %
```

*Рисунок 1: Результат lab6-1 с символами*

**Комментарии:** При сложении символов «6» и «4» программа складывает их ASCII-коды ( $54 + 52 = 106$ ), что соответствует символу «j». Данная операция с символами производится с их числовыми кодами.

## Программа lab6-1.asm — Сложение числовых данных

```
GNU nano 8.6
#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

ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nano lab6-1.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nasm -f elf32 lab6-1.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ld -m elf_i386 -o lab6-1 lab6-1.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ./lab6-1

ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 %
```

Рисунок 2: Результат lab6-1 с числами

**Комментарии:** При сложении чисел 6 и 4 программа выводит символ с кодом 10(перевод строки), так как функция sprintfF интерпритирует числовое значение как ASCII-код.

## Программа lab6-2.asm — Использование iprintLF

```
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % touch lab6-2.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nano lab6-2.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nasm -f elf32 lab6-2.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ld -m elf_i386 -o lab6-2 lab6-2.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ./lab6-2
106
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 %
```

Рисунок 3: Результат lab6-2 при запуске

**Комментарии:** Функция iprintLF корректно выводит числовое значение 106, полученное при сложении ASCII-кодов символов, демонстрируя правильное преобразование числа в строку для вывода

## Сравнение iprint и iprintLF

```
GNU nano 8.6
#include 'in_out.asm'

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

ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nasm -f elf32 lab6-2.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ld -m elf_i386 -o lab6-2.asm lab6-2.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ./lab6-2
10%
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 %
```

Рисунок 4: Результат lab6-2 с использованием iprint

**Комментарии:** Функция iprintLF добавляет перевод строки после вывода числа, в то время как iprint выводит число без перевода строки, что может приводить к появлению символа «%» на выходе терминала.

## 2.2 Выполнение арифметических операций

**Задание:** Освоение арифметических инструкций NASM на примере вычисления выражений.

### Программа lab6-3.asm — Вычисление выражения

```
GNU nano 8.6
#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

ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % touch lab6-3.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nano lab6-3.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nasm -f elf32 lab6-3.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ld -m elf_i386 -o lab6-3 lab6-3.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ./lab6-3
Результат: 4
Остаток от деления: 1
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 %
```

Рисунок 5: Код и результат lab6-3

**Комментарии:** Программа успешно вычисляет выражение  $(5*2+3)/3 = 4$  с остатком 1, демонстрируя работу инструкций умножение (mul) и деление (div) с получением частного и остатка.

## Программа variant.asm — Вычисление варианта

```
GNU nano 8.6
#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 printf

    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 iprintf

    call quit

ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % touch variant.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nano variant.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nasm -f elf32 variant.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ld -m elf_i386 -o variant variant.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ./variant
Введите № студенческого билета:
1032252590
Ваш вариант: 11
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % █
```

Рисунок 6: Вычисление варианта по номеру студенческого билета

**Комментарии:** Программа успешно вычисляет вариант задания по номеру студенческого билета, используя операцию деления с остатком и преобразование ASCII-строки в число с помощью функции `atoi`.

### 3. Задание для самостоятельной работы

**Задание:** Написать программу вычисления выражения  $f(x)=10(x+1)-10$  для варианта 11.

**Выполнение:**

```
GNU nano 8.6
#include 'in_out.asm'

SECTION .data
msg: DB 'Вычисление выражения f(x) = 10(x + 1) - 10',0
input: DB 'Введите x: ',0
res: DB 'Результат: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:
    ; Вывод заголовка
    mov eax, msg
    call sprintf

    ; Ввод значения x
    mov eax, input
    call sprintf

    mov ecx, x
    mov edx, 80
    call sread

    ; Преобразование введенной строки в число
    mov eax, x
    call atoi

    ; Вычисление выражения 10(x + 1) - 10
    ; Шаг 1: x + 1
    mov ebx, eax    ; сохраняем x в ebx
    add ebx, 1      ; ebx = x + 1

    ; Шаг 2: 10 * (x + 1)
    mov eax, 10     ; eax = 10
    mul ebx         ; eax = 10 * (x + 1)

    ; Шаг 3: 10(x + 1) - 10
    sub eax, 10     ; eax = 10(x + 1) - 10

    ; Сохранение результата
    mov edi, eax

    ; Вывод результата
    mov eax, res
    call sprintf
    mov eax, edi
    call iprintLF

    call quit
```

*Рисунок 7: Код программы для самостоятельной работы*

```

ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nano lab6-4.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % nasm -f elf32 lab6-4.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ld -m elf_i386 -o lab6-4 lab6-4.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ./lab6-4
Вычисление выражения  $f(x) = 10(x + 1) - 10$ 
Введите x: 1
Результат: 10
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % ./lab6-4
Вычисление выражения  $f(x) = 10(x + 1) - 10$ 
Введите x: 7
Результат: 70
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab06 % █

```

*Рисунок 8: Результат выполнения программы при вводе двух разных значений переменных  $x_1$  и  $x_2$ .*

**Комментарий:** Программа успешно выясняет заданное выражение для тестовых значений  $x_1$  и  $x_2$ , при помощи использования арифметических инструкций add(сложение), mul(умножение) и sub(вычитание) для вычисления составного выражения.

#### **4. Выводы:**

В ходе выполнения лабораторной работы были достигнуты следующие результаты:

- 1) Освоение арифметических инструкций NASM.
- 2) Изучены различия символьных и численных данных.
- 3) Реализованы программы вычисления выражений.
- 4) Освоен ввод данных с клавиатуры.
- 5) Изучено деление с остатком.
- 6) Выполнено задание для самостоятельной работы.

**Цель работы достигнута полностью** — освоены арифметические инструкции языка ассемблера NASM и их практическое применение для вычисления выражений различной сложности.

### Список литературы:

1. **The NASM documentation** — 2021. — URL: <https://www.nasm.us/docs.php>
2. **Расширенный ассемблер: NASM** — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>
3. **Столяров А.** Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix)
4. **Колдаев В. Д., Lupin С. А.** Архитектура ЭВМ. — М. : Форум, 2018.