

title: "лабораторная работа №6" author: "Сархан хабиба осама"
date: " 17 ноября 2025 г." — Цель работы

Изучение различий между символьными и численными данными в ассемблере NASM, освоение арифметических операций и использование подпрограмм преобразования данных.

Теоретическая часть

Различие между символьными и численными данными

В ассемблере NASM существует принципиальная разница между:

- Символьными данными: '6' - код ASCII символа (54 в десятичной системе)
- Численными данными: 6 - числовое значение

Функции преобразования из in_out.asm

- iprint - вывод чисел в формате ASCII
- iprintLF - вывод чисел с переводом строки
- atoi - преобразование ASCII-кода символа в целое число
- sprint - вывод строки
- sread - чтение строки ввода

Выполненные программы

1. Программа lab6-1.asm

Цель: Понять разницу между обработкой символов и чисел

```
%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
```

Результат выполнения: Символ 'j'

Объяснение:

- Код ASCII символа '6' = 54
- Код ASCII символа '4' = 52
- $54 + 52 = 106$
- Код 106 в ASCII таблице соответствует символу 'j'

Программа выводит символ, а не число

2. Программа lab6-2.asm

Цель: Использование функций `iprint` и `iprintLF` для преобразования данных

Код:

```
%include 'in_out.asm'
```

```
SECTION .text
global _start
_start:
    mov eax, '6'
    mov ebx, '4'
    add eax, ebx
    call iprintLF
    call quit
```

Результат выполнения: Число 106

Объяснение: Функция `iprintLF` преобразует числовое значение в ASCII-строку и выводит его как число, а не как символ.

Дополнительные эксперименты:

· При замене `mov eax, '6'` на `mov eax, 6` результат: 10 · При замене `call iprintLF` на `call iprint` вывод происходит без перевода строки

3. Программа lab6-3.asm

Цель: Выполнение арифметических операций с выводом результата и остатка

Код:

```
%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

```

Результат выполнения:

Результат: 4

Остаток от деления: 1

Объяснение вычислений:

· $5 \times 2 = 10$ · $10 + 3 = 13$ · $13 \div 3 = 4$ (целая часть) · Остаток от деления = 1

Модификация для выражения $(4*6+2)/5$:

· Результат: 5 · Остаток: 1

4. Программа variant.asm

Цель: Вычисление варианта задания по номеру студенческого билета

Код:

```

#include 'in_out.asm'

```

```

SECTION .data

```

```

msg DB 'Введите номер студенческого билета: ',0
res DB 'Ваш вариант: ',0

```

```

SECTION .bss

```

```

x: RESB 80

```

```

SECTION .text

```

```

global _start

```

```

_start:

```

```

mov eax, msg
call sprint

```

```

mov ecx, x
mov edx, 80
call sread

```

```

mov eax, x
call atoi

xor edx, edx
mov ebx, 20
div ebx
add edx, 1

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

call quit

```

Алгоритм работы:

1. Вывод приглашения для ввода
2. Чтение номера студенческого билета
3. Преобразование введенной строки в число (atoi)
4. Вычисление варианта по формуле: $(S \% 20) + 1$
5. Вывод результата

Анализ результатов

Сравнение способов вывода данных

Способ вывода Результат для '6' + '4' Результат для 6+4 Вывод как символ 'j' (код 106) Символ с кодом 10 Вывод как число (iprint) 106 10

Особенности арифметических операций

- Для корректного деления необходимо обнулять регистр EDX
- Функция mul умножает EAX на операнд, результат в EAX
- Функция div делит EDX:EAX на операнд, частное в EAX, остаток в EDX

Выводы

1. Изучена принципиальная разница между символьными и численными данными в ассемблере NASM
2. Освоены арифметические операции (сложение, умножение, деление) и их реализация в NASM
3. Научены использовать подпрограммы преобразования данных из файла in_out.asm
4. Получен практический опыт работы с функциями ввода-вывода и преобразования типов данных

5. Разработаны и протестированы программы для выполнения арифметических операций и вычисления варианта задания

Все программы работают корректно, результаты соответствуют ожидаемым. Работа demonstrates понимание основных принципов обработки данных в ассемблере NASM.