Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №1**

«Линейное исполнение программ.

Арифметические и поразрядные логические операции над

байтами»

Выполнил:

студент группы 213 Файтельсон А.А.

Проверил:

доцент кафедры ПОиАИС Кривонос А.В.

Курск, 2023

**Цели работы:** изучение принципов функционирования памяти и микропроцессора компьютера при последовательном исполнении

команд программы. Приобретение навыков использования арифметических команд при написании ассемблерных программ. Приобретение навыков использования поразрядных логических команд при написании ассемблерных программ. Получение представления об особенностях обработки данных разных размерностей и режимах доступа к данным при выполнении арифметических операций.

**Листинг программы**

format ELF

section '.data' writeable

a dd 10

b dd -27

inputFormat db "%d%d", 0

outputFormat db "%d", 0

; 1.1. Данные

val1 dd 10 ; первое значение 10

val2 dd 27 ; второе значение 27

bin\_val db 11001100B ; произвольное двоичное значение

; 1.2. Резервирование памяти для результатов

sum\_res dd 0 ; результат сложения

diff\_res dd 0 ; результат вычитания

product dd 1 ; результат умножения (двухбайтовое значение)

quotient db 0 ; результат деления (частное)

remainder db 0 ; результат деления (остаток)

section '.text' executable

public main

main:

mov ebp, esp; for correct debugging

; mov ebp, esp; for correct debugging

; 1.3. Сложение 10 и 27

mov eax, [val1] ; загрузить значение 10 из памяти в регистр EAX

add eax, [val2] ; добавить к значению 10 значение 27

mov [sum\_res], eax ; сохранить результат сложения в памяти в переменной sum\_res

; 1.4. Вычитание 10 и 27

mov eax, [val1] ; загрузить значение 10 из памяти в регистр EAX

sub eax, [val2] ; вычесть 27 из значения 10

mov [diff\_res], eax ; сохранить результат вычитания в памяти в переменной diff\_res

; 1.5. Изменить знак второго числа (27) и выполнить операцию 10 - (-27)

mov eax, [val1] ; загрузить значение 10 в регистр EAX

neg [val2] ; изменить знак числа 27 (оно становится -27)

sub eax, [val2] ; вычесть -27, что эквивалентно сложению 10 + 27

mov [diff\_res], eax ; сохранить результат в переменной diff\_res

; 1.6. Умножение 10 на -27 с учетом знака

mov eax, [val1] ; загрузить значение 10 в регистр EAX

imul eax, [val2] ; умножить EAX (10) на значение -27 (val2) с учетом знака

mov [product], eax ; сохранить результат умножения в переменной product

; Беззнаковое умножение 10 на -27

mov eax, [val1] ; загрузить значение 10 в регистр EAX

mov ebx, [val2] ; загрузить значение -27 в регистр EBX

mul ebx ; выполнить беззнаковое умножение EAX на EBX (-27)

mov [product], eax ; сохранить результат в переменной product

; 1.7. Деление 27 на 10

mov eax, [val2] ; загрузить значение 27 в регистр EAX

mov ebx, [val1] ; загрузить значение 10 в регистр EBX

div ebx ; выполнить деление EAX (27) на EBX (10); результат деления в AL, остаток в AH

mov [quotient], al ; сохранить частное от деления в переменной quotient

mov [remainder], ah ; сохранить остаток от деления в переменной remainder

; 1.8. Логические операции над двоичным значением

mov al, [bin\_val] ; загрузить двоичное значение в регистр AL

or al, 00000011B ; установить два бита в 1 (OR с маской 00000011)

not al ; инвертировать все биты

and al, 00011100B ; сбросить (очистить) три бита, используя AND с маской 00011100

mov bl, al ; дублировать результат в регистр BL

xor al, bl ; сложить значение AL и BL по модулю 2 (XOR)

;write your code here

xor eax, eax

ret

**Вывод:** изучили принципы функционирования памяти микропроцессора компьютера, научились использовать арифметические команды при написании ассемблерных программ, приобрели навык использования поразрядных логических команд при написании ассемблерных программ, разобрались в особенностях обработки данных разных размерностей и режимах доступа к данным при выполнении арифметических операций.