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 №3**

«Команды условного и безусловного переходов. Организация ветвлений и циклов в программе»

Выполнил:

студент группы 213 Файтельсон А.А.

Проверил:

доцент кафедры ПОиАИС Кривонос А.В.

Курск, 2024

**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**

data segment ; Начало сегмента данных

a db 15 ; Переменная 'a' с начальным значением 15

b db 9 ; Переменная 'b' с начальным значением 9

nod db 0 ; Переменная для хранения результата НОД (изначально 0)

data ends ; Конец сегмента данных

code segment ; Начало сегмента кода

begin: ; Метка начала программы

; Инициализация сегмента данных

mov ax, data ; Загружаем адрес сегмента данных в регистр AX

mov ds, ax ; Устанавливаем сегмент данных (DS) на значение из AX

; Подготовка для алгоритма Евклида

mov al, a ; Загружаем значение переменной 'a' в регистр AL

mov bl, b ; Загружаем значение переменной 'b' в регистр BL

while\_loop: ; Начало цикла алгоритма Евклида

cmp al, bl ; Сравниваем значения AL (a) и BL (b)

je end\_loop ; Если AL равно BL, переходим к завершению цикла (НОД найден)

jl smaller ; Если AL меньше BL, переходим к обработке случая "меньшее значение"

jg bigger ; Если AL больше BL, переходим к обработке случая "большее значение"

smaller: ; Обработка случая, когда AL < BL

sub bl, al ; Вычитаем значение AL из BL (b = b - a)

jmp while\_loop ; Возвращаемся в начало цикла для повторной проверки

bigger: ; Обработка случая, когда AL > BL

sub al, bl ; Вычитаем значение BL из AL (a = a - b)

jmp while\_loop ; Возвращаемся в начало цикла для повторной проверки

end\_loop: ; Конец цикла алгоритма Евклида

mov nod, al ; Сохраняем найденный НОД (содержимое AL) в переменную nod

; Подсчет количества единиц в двоичном представлении НОД

mov cl, 8 ; Устанавливаем счетчик итераций на 8 (по числу битов в nod)

xor dl, dl ; Обнуляем DL для подсчета количества единиц

count\_one: ; Начало цикла подсчета единиц

shr nod, 1 ; Сдвигаем биты переменной nod вправо на 1, старший бит уходит в флаг переноса

adc dl, 0 ; Увеличиваем DL на 1, если был перенос (т.е. младший бит был равен 1)

loop count\_one ; Уменьшаем CL и повторяем цикл, пока CL > 0

; Завершение программы

mov ah, 4ch ; Устанавливаем код завершения программы для DOS (прерывание 21h)

int 21h ; Вызываем прерывание 21h для завершения программы

code ends ; Конец сегмента кода

end begin ; Завершение программы, точка входа — метка begin

**Вывод:** изучили принципы функционирования памяти и микропроцессора компьютера при выполнении ветвлений и циклов;