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 №2**

«Адресация межсегментных

переходов»

Выполнил:

студент группы 213 Файтельсон А.А.

Проверил:

доцент кафедры ПОиАИС Кривонос А.В.

Курск, 2023

**Цели работы:**

1) изучение принципов функционирования памяти и микропроцессора компьютера при выполнении межсегментных переходов;

2) приобретение навыков использования команд сдвига при написании ассемблерных программ;

3) получение представления об особенностях обработки данных и режимах доступа к данным при выполнении операций сдвига над данными.

**Листинг программы**

section .data

; Первый сегмент данных: однобайтовое число

seg1\_data:

db 0b11001010 ; Число в двоичной системе счисления

; Второй сегмент данных: адрес перехода на первый сегмент кода

seg2\_data:

dq first\_code\_segment ; Адрес перехода на первый сегмент кода

; Третий сегмент данных: ещё одно однобайтовое число

seg3\_data:

db 0b10110100 ; Число в двоичной системе счисления

section .bss

; Сегмент стека: зарезервировать 20 байт

stack\_res:

resb 20

section .text

global \_start ; Начало программы

; Третий сегмент кода

third\_code\_segment:

; Умножение числа из первого сегмента данных на 2

mov al, [seg1\_data] ; AL = число из первого сегмента

shl al, 1 ; Линейный сдвиг влево на 1 (умножение на 2)

; Деление числа из третьего сегмента данных на 4

mov bl, [seg3\_data] ; BL = число из третьего сегмента

shr bl, 2 ; Линейный сдвиг вправо на 2 (деление на 4)

; Переход на метку, определенную во втором сегменте данных

mov rax, [seg2\_data] ; Адрес перехода в rax

jmp rax ; Межсегментный переход на метку в первом сегменте кода

; Метка завершения работы

end\_label:

; Корректное завершение программы

mov rax, 60 ; Системный вызов для выхода (sys\_exit)

xor rdi, rdi ; Код завершения 0

syscall

; Первый сегмент кода

first\_code\_segment:

; Получить значение третьего бита числа из первого сегмента данных

mov bl, [seg1\_data] ; BL = число из первого сегмента

rol bl, 5 ; Циклический сдвиг влево на 5 бит (чтобы третий бит оказался на месте старшего)

and bl, 0b1 ; Извлекаем третий бит

; Переход во второй сегмент кода

jmp second\_code\_segment

; Второй сегмент кода

second\_code\_segment:

; Получить значение пятого бита числа из третьего сегмента данных

mov bh, [seg3\_data] ; BH = число из третьего сегмента

rol bh, 3 ; Циклический сдвиг влево на 3 бит (чтобы пятый бит оказался на месте старшего)

and bh, 0b1 ; Извлекаем пятый бит

; Переход к метке завершения работы

jmp end\_label

\_start:

jmp third\_code\_segment

**Вывод:** изучили принципы функционирования памяти и микропроцессора компьютера при выполнении межсегментных переходов; приобрели навыки использования команд сдвига при написании ассемблерных программ; получили представления об особенностях обработки данных и режимах доступа к данным при выполнении операций сдвига над данными.