Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: **«**Конструирование программ**»**

**ОТЧЕТ**

к лабораторной работе №4

на тему:

**«СОЗДАНИЕ ПРОСТОЙ ПРОГРАММЫ НА ЯЗЫКЕ АССЕМБЛЕР. ОБРАБОТКА СИМВОЛЬНЫХ ДАННЫХ»**

БГУИР 1-40 04 01

|  |
| --- |
| Выполнил студент группы 253505 БЕКАРЕВ Станислав Сергеевич |
|  |
| (дата, подпись студента) |
| Проверил ассистент кафедры информатики РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2023

**Цель работы:** Ознакомиться с директивами определения данных, изучить команды пересылки данных и передачи управления, изучить строчные операции и прерывания консольного ввода-вывода высокого уровня.

В рамках выполнения лабораторной работы должно быть выполнены следующее задание:

Написать программу, которая может инвертировать каждое слово в строке.

**Ход работы:** Для выполнения задания был написан программный код на языке Assembler, представленный в листинге №1.

Данный код сначала предлагает пользователю ввести строку со словами, затем считывает её и заносит в память. Затем программа начинает сохранять в стэк символы пока не встретит пробел, после чего возвращается на начало слово и начинает выгружать символы из стека. Доходя до пробела программа начинает проделывать тоже самое со следующим словом и т.д. В момент достижения конца строки мы получаем новую строку с инвертированными словами и выводим ее в консоль.

Для проверки программы запустим симулятор микроконтроллера, введем тестовую строку со словами. На рисунке 1 показан ввод и результат работы программы..



Рисунок 1 – Результат работы программы.

Листинг №1 – Исходный код задания

org 100h

newline macro:

push ax

push dx

mov ah, 02h

mov dl, 0Dh

int 21h

mov dl, 0Ah

int 21h

pop dx

pop ax

endm

start:

mov ah, 09h

mov dx, offset enter\_mess

int 21h

mov ah, 0Ah

mov dx, offset in\_mess

int 21h

newline

mov bx, offset in\_mess + 1

mov cx, 0

lp:

inc cx

inc bx

cmp [bx], 0Dh

jne lp

cmp cx, 01h

jne point

mov dx, offset empty

mov ah, 09h

int 21h

mov ah, 4Ch

int 21h

point:

mov di, offset in\_mess + 2

mov si, di

mov ah, 0h

lp1:

mov al, [si]

push ax

inc si

cmp [si], 0Dh

je lp3

cmp [si], ' '

jne lp1

mov si, di

lp2:

pop ax

mov [si], al

inc si

cmp [si], 0Dh

je lp3

cmp [si], ' '

jne lp2

inc si

mov di, si

jmp lp1

lp3:

mov si, di

pop ax

mov [si], al

inc si

cmp [si], 0Dh

mov di, si

jne lp3

mov ah, 09h

mov dx, offset end\_mess

int 21h

mov ah, 40h

mov bx, 01h

mov dx, offset in\_mess + 2

int 21h

mov ah, 4Ch

int 21h

in\_mess db 200 DUP('$')

enter\_mess db 'Enter your string with words:', 0Dh, 0Ah, '$'

end\_mess db 'Result string:', 0Dh, 0Ah, '$'

empty db 'Empty string!!!$'

end start

**Вывод:** Быловыполнено написание программы, которая может инвертировать каждое слово в строке. Были изучены директивы определения данных, команды пересылки данных и передачи управления, строчные операции и прерывания консольного ввода-вывода высокого уровня.