Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Отчёт

по лабораторной работе № 1

на тему

«Последовательный порт»

|  |  |
| --- | --- |
| Студент группы № 950503 | Гуринович А. В. |
| Преподаватель | Одинец Д. Н. |

Минск 2021

**Задача**

Разработать  программный  модуль  реализации  процедуры  передачи (приема) байта информации через последовательный интерфейс.

Программа должна демонстрировать программное взаимодействие с последовательным интерфейсом с использованием следующих механизмов:

1. прямое взаимодействие с портами ввода-вывода (write, read)
2. использование BIOS прерывания 14h,
3. работа с COM-портом через регистры как с устройством ввода-вывода.

**Алгоритм**

Передача данных через последовательный интерфейс состоит из следующих фаз:

• инициализация порта;

• передача данных;

• прием данных;

• анализ состояния порта.

Программное взаимодействие с последовательным интерфейсом может осуществляться с использованием различных механизмов. Самый прозрачный из них - это прямое взаимодействие с портами ввода-вывода. Более высокоуровневые методы: использование прерывания 14h, работа с COM-портом как с устройством ввода-вывода.

Далее приведены листинги программ, реализующие процедуры передачи (приема) байта информации через последовательный интерфейс.

**Листинг программы, использующей WinAPI:**

**–**

**Листинг программы, использующей BIOS прерывания 14h:**

**Int14h.exe:**

.model small

.code

.stack 100h

start:

mov ah, 0 ;interuption number for port initialization

mov al, 11100011b ;9600 bits per second (111), no paritete check (00), stop bit lenght = 1 (1), number of bits in word = 8 (11)

mov dx, 0 ;COM1 (0)

int 14h

main:

mov ah, 2 ;read symbol from port specified in DX

int 14h

test ah, ah ;if something recived (ah != 0)

jnz nothingWasRecived

int 29h ;print recived symbol

nothingWasRecived:

mov ah, 1

int 16h

jz main ;if nothing in stdin bufer

mov ah, 8

int 21h ;write symbol from stdin to al without echo

test al, al

jnz sendSymbol ;if symbol was typed in

int 21h

cmp al, '-' ;if '-' (close program)

jne sendSymbol

mov ax, 4C00h

int 21h ;return control to operating system

sendSymbol:

mov ah, 1

int 14h ;send symbol to port specified in DX

jmp main

end start

**Листинг программы, работающей с COM-портом через регистры как с устройством ввода-вывода:**

**RSender.com:**

.model tiny

org 100h

.data

pointerCOM1 dw ?

inputString db "Enter symbol: ", 0Ah, 0Dh, '$'

errorMessage db 0Ah, 0Dh, "Error occupied while sending", '$'

succeessSend db 0Ah, 0Dh, "Sending completed successfully", '$'

.code

getChar macro

mov ah, 01h ;writing char into al

int 21h

endm

readStringFrom macro pointer

mov ah, 09h

mov dx, offset pointer

int 21h

endm

main:

mov ax, 40h ;read address of standart input device (COM1 by default)

mov es, ax

mov dx, es:[00]

mov pointerCOM1, dx

;setup

add dx, 3

mov al, 80h

out dx, al

sub dx, 2

mov al, 0

out dx, al

dec dx

mov al, 18h

out dx, al

add dx, 3

mov al, 00011011b

out dx, al

dec dx

dec dx

mov al, 00h

out dx, al

mov dx, pointerCOM1

add dx, 5

wait:

in al, dx

test al, 40h

jz wait

sub dx, 5

mov al, ah

out dx, al

readStringFrom inputString

getChar

mov dx, pointerCOM1

add dx, 5

out dx, al

in al, dx ;sending char

test al, 80h

jz succeess

readStringFrom errorMessage

ret

succeess:

readStringFrom succeessSend

ret

end main

**RReciver:**

.model tiny

org 100h

.data

pointerCOM1 dw ?

input\_ok db "Reciving completed successfully", 0Ah, 0Dh, '$'

errorMessage db "Error occupied while reciving", '$'

.code

putChar macro symbol

mov ah, 06h

mov dl, symbol

int 21h

endm

readStringFrom macro pointer

mov ah, 09h

mov dx, offset pointer

int 21h

endm

main:

mov ax, 40h ;read address of standart input device (COM1 by default)

mov es, ax

mov dx, es:[02]

mov pointerCOM1, dx

add dx, 3

mov al, 80h

out dx, al

sub dx, 2

mov al, 0

out dx, al

dec dx

mov al, 18h

out dx, al

add dx, 3

mov al, 00011011b

out dx, al

dec dx

dec dx

mov al, 00h

out dx, al

mov dx, pointerCOM1

add dx, 5

wait:

in al, dx

test al, 01h

jz wait

sub dx, 5

in al, dx

push ax

add dx, 5

in al, dx

test al, 80h

jz ok

readStringFrom errorMessage

ret

ok:

readStringFrom input\_ok

pop ax

putChar al

ret

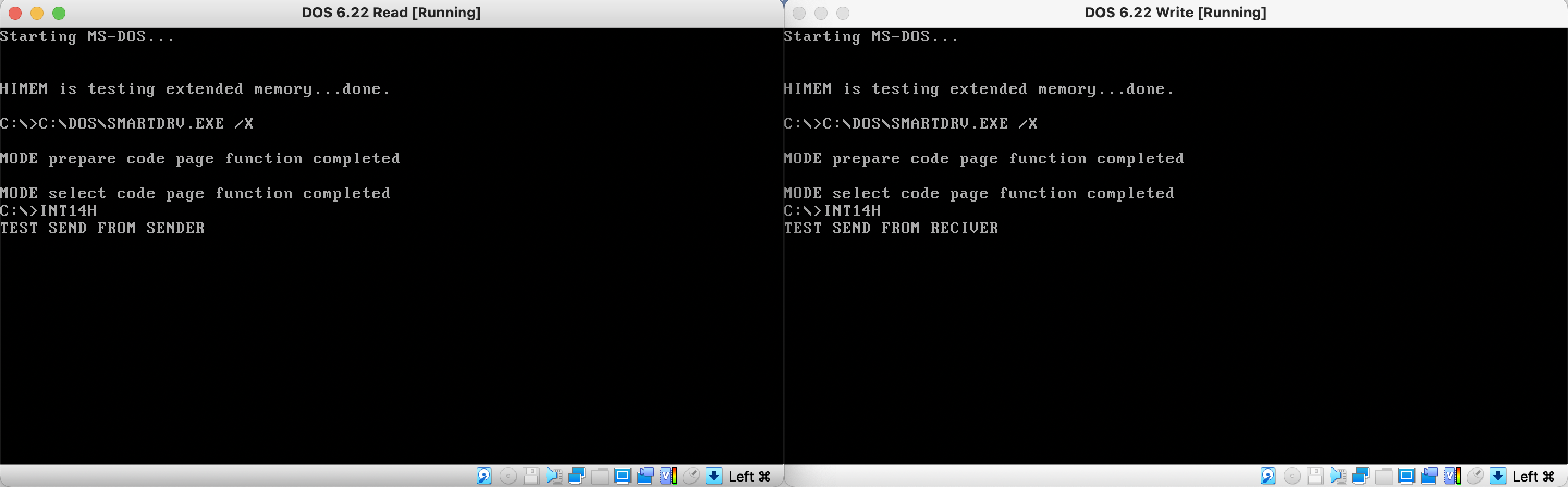
end main

**Демонстрация работы:**

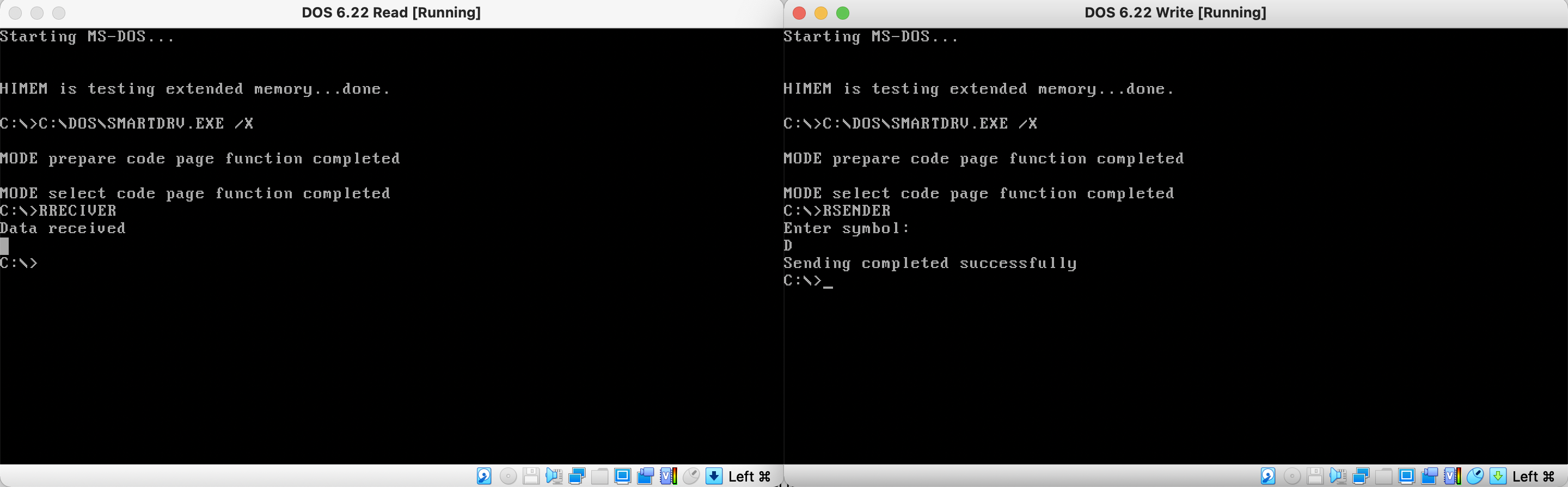
**1)Программа, использующая WinAPI:**

**–**

**2) программа, использующая BIOS прерывания 14h:**

****

**3) программа, работающая с COM-портом через регистры как с устройством ввода-вывода:**

****

**Заключение**

В ходе данной лабораторной работы были изучены возможности использования последовательных интерфейсов RS-232, разработаны алгоритмы передачи данных. Программное взаимодействие с последовательным интерфейсом осуществлялось с использованием различных механизмов. Самый прозрачный из них - это прямое взаимодействие с портами ввода-вывода. Более высокоуровневые методы: использование прерывания 14h, работа с COM-портом как с устройством ввода-вывода.

Для эмуляции для эмуляции DOS использовался Oracle Virtual Box, COM порты были перенаправлены из обоих виртуальных машин в UNIX FIFO pipe.