Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №1

«Последовательный порт»

Выполнил: Проверил:

Студент группы 050504 Преподаватель

Крачковский А.В. Одинец Д.Н.

Минск, 2021

1. Постановка задачи

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

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

1. Прямое взаимодействие с портами ввода-вывода (wirte, read)
2. Использование BIOS прерывания 14h
3. Работа с COM-портом через регистры как с устройствами ввода-вывода.
4. Алгоритм

Программа состоит из нескольких подпрограмм (частей программы), представляющих собой некоторые функции. К ним относятся функции:

* Инициализация порта
* Запись байта информации в порт
* Чтение байта информации из порта
* Вывод результата на экран

1. Листинг программы

Далее приведены листинги программ, реализующие различные механизмы передачи (приёма) информации через последовательный интерфейс.

3.1. Листинг программы, взаимодействующей с портами ввода-вывода.

#include <stdio.h>   /\* Стандартные объявления ввода/вывода \*/   
#include <string.h>  /\* Объявления строковых функций \*/   
#include <unistd.h>  /\* Объявления стандартных функций UNIX \*/   
#include <fcntl.h>   /\* Объявления управления файлами \*/   
#include <errno.h>   /\* Объявления кодов ошибок \*/   
#include <termios.h> /\* Объявления управления POSIX-терминалом \*/   
#include <malloc.h>   
  
int entering\_line(char \*\*line){   
   int line\_size = 0, letter\_num = 0; char letter;   
   (\*line) = (char\*)malloc(1\*sizeof(char));   
   while(1){   
       scanf("%c", &letter);   
       if (letter == '\n'){   
           (\*line)[letter\_num] = '\0';   
           break;   
       }   
       (\*line)[letter\_num] = letter;   
       line\_size+=2; letter\_num++;   
       (\*line) = (char\*)realloc(\*line, line\_size);   
   }   
   return line\_size;   
}   
  
int main(){   
  
   int fd0 = open("/dev/tnt0", O\_RDWR | O\_NOCTTY | O\_NDELAY);   
   int fd1 = open("/dev/tnt1", O\_RDWR | O\_NOCTTY | O\_NDELAY);   
   int bytes;   
   if (fd0 == -1){   
       perror("open\_port: Unable to open");   
       return fd0;   
   }   
   else{   
       fcntl(fd0, F\_SETFL, 0);   
       printf("Opened successfully!\n");   
   }   
  
   char \*send\_message;   
   bytes = entering\_line(&send\_message);   
  
  
   int n;   
   n = write(fd0, send\_message, bytes);   
   if (n == -1){   
       fputs("write() of 1 bytes failed!\n", stderr);   
       free(send\_message);   
       return n;   
   } else {   
       printf("Success write!\n");   
   }   
  
   char accept\_letter = 'a';   
   while (accept\_letter != '\0'){   
       n = read(fd1, &accept\_letter, 1);   
       if(n == -1){   
           printf("Error read\n");   
           free(send\_message);   
           return -1;   
       }   
       printf("%c", accept\_letter);   
   }   
   puts("");   
   close(fd0);   
   close(fd1);   
   free(send\_message);   
   return 0;   
}

3.2. Листинг программы, использующей BIOS прерывание 14h.

.model small   
.stack 100h   
  
.data   
  
Error\_Write db "Write error!",0Dh,0Ah,'$'   
Error\_Read db "Read error!",0Dh,0Ah,'$'   
             
.code   
  
start:   
; connect data   
mov ax,DGROUP   
mov ds,ax   
; send byte   
mov dx,0   
mov ah,1   
mov al,'a'   
int 14h   
mov al,'r'   
  
;check if cool   
cmp ah,7   
je W\_Error   
jmp W\_Cont   
W\_Error:   
  mov dx,offset Error\_Write   
  mov ah,9   
  int 21h   
  mov ax,4C00h   
  int 21h   
  ret   
W\_Cont:   
; read   
xor dx,dx   
mov dx,1   
mov ah,02h   
int 14h   
;check if read   
cmp ah,0   
jne R\_Error   
jmp R\_Continue   
R\_Error:   
  mov dx,offset Error\_Read   
  mov ah,9   
  int 21h   
  mov ax,4C00h   
  int 21h   
  ret   
R\_Continue:   
  mov ah,2   
  mov dl,al   
  int 21h   
  mov ax,4C00h   
  int 21h   
  
end start

3.3. Листинг программы, работающей с COM-портами через регистры как с устройствами ввода-вывода.

.model small   
.stack 100h   
.data   
  
.code   
start:   
  
  
mov dx,3F8h   
mov al,'r'   
out dx,al   
  
mov al,'a'   
  
mov dx,2F8h   
  
in al,dx   
mov ah,2   
mov dl,al   
int 21h   
  
mov ax,4C00h   
int 21h   
end start

1. Тестирование программ

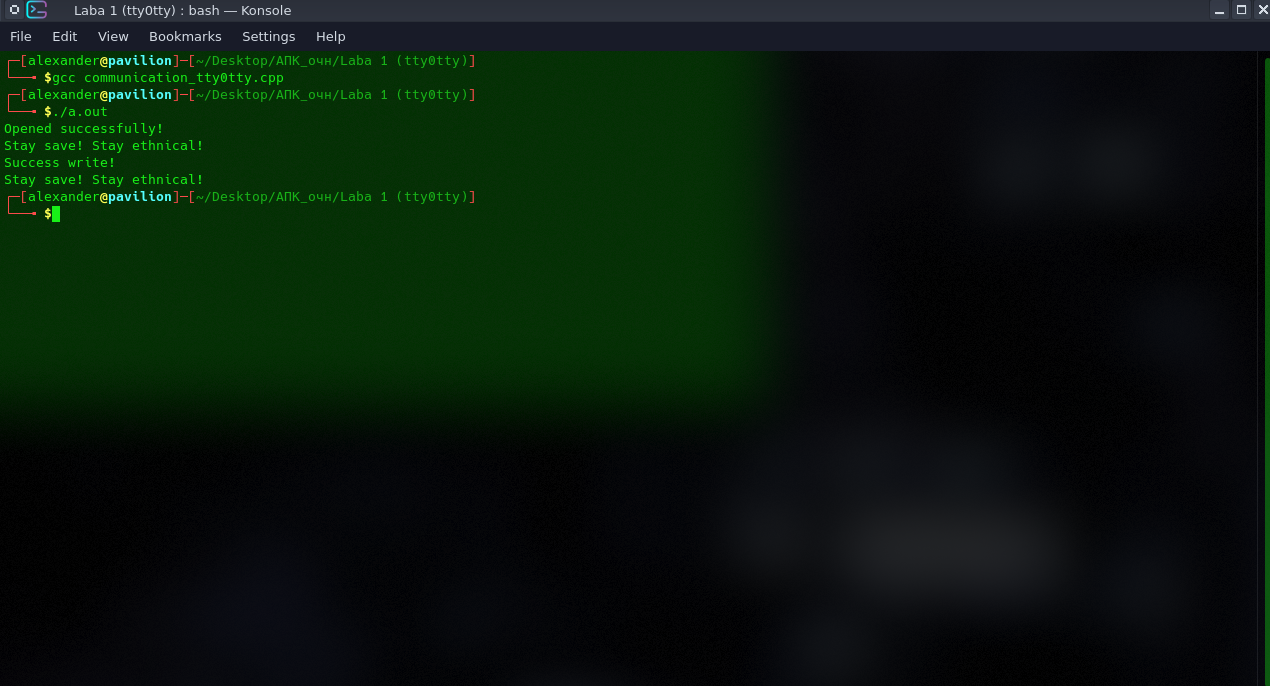


Рисунок 4.1 – Результат работы программы, взаимодействующей с портами ввода-вывода.

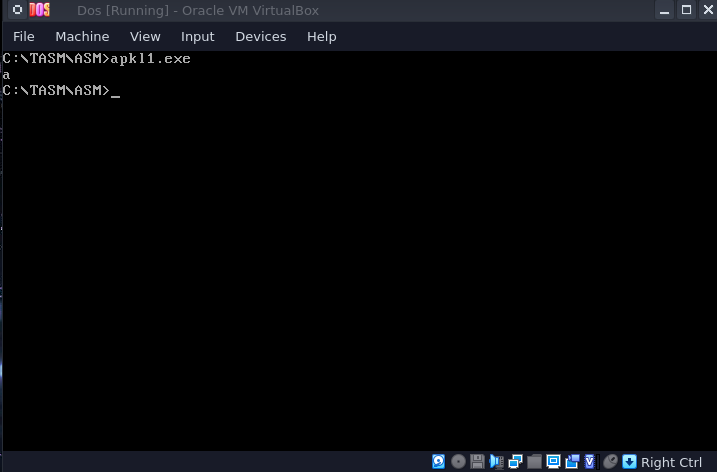
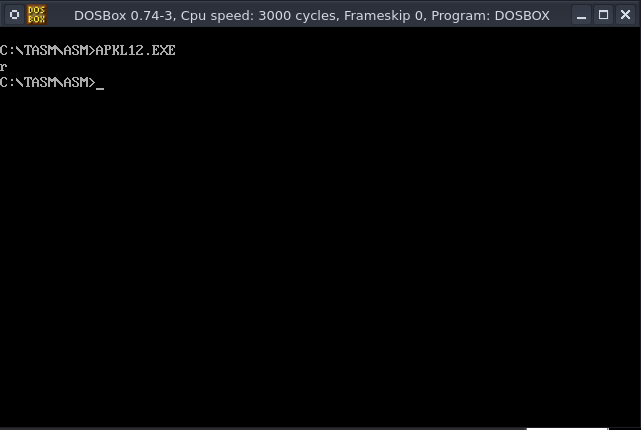


Рисунок 4.2 – Результат работы программы, использующей BIOS прерывание 14h.

Рисунок 4.3 – Результат работы программы, работающей с COM-портами через регистры как с устройствами ввода-вывода.



1. Заключение

В ходе лабораторной удалось передать 1 байт информации через последовательный порт с использованием различных механизмов.

Для эмуляции COM портов использовался tty0tty, для эмуляции DOS используется Oracle Virtual Box и dosbox на хосте Linux debian.