

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Вычислительной техники**

**ОТЧЕТ**  
**по лабораторной работе № 4**  
**по дисциплине «Организация ЭВМ и систем»**  
**ТЕМА: КЛАВИАТУРА IBM PC. ИСПОЛЬЗОВАНИЕ ПРЕРЫВАНИЙ**

Студент гр. 3311

Баймухамедов Р. Р.

Преподаватель

Гречухин М. Н.

Санкт-Петербург

2024

## **Краткие сведения о подсистеме ввода информации с клавиатуры, используемых прерываниях, буфере клавиатуры и функциях обслуживания ввода с клавиатуры.**

подавляющее большинство программ выполняют ввод информации с клавиатуры на трех уровнях: обращением к функциям MS-DOS, обращением к функциям BIOS и физическим доступом к аппаратным средствам.

- Ввод информации на уровне MS-DOS: Позволяет "пропустить" клавиатурный ввод через устанавливаемые драйверы, обеспечивает отслеживание нажатия комбинации клавиш Ctrl-C (Ctrl-Break) и стандартную обработку ошибок.
- Ввод информации на уровне BIOS: Позволяет программе отслеживать нажатие всех клавиш, выполнять управление аппаратурой клавиатуры и пр. Интерфейсом Turbo C с BIOS является функция bioskey().
- Физический доступ к буферу клавиатуры: Резко повышает производительность программы. В некоторых случаях необходима имитация нажатий клавиш с записью кодов непосредственно в буфер. При этом физически нажатия клавиш не происходят.

Аппаратные и программные средства ввода информации с клавиатуры

- Аппаратные средства: Клавиатура содержит специальный микропроцессор, который при каждом нажатии и отпускании клавиши определяет ее порядковый номер и помещает его в порт 60h. Скэн-код содержит порядковый номер нажатой клавиши и бит, указывающий на нажатие или отпускание клавиши.
- Анализ и преобразование скэн-кода: BIOS ISR анализирует скэн-код и преобразует его по специальным правилам. Клавиши делятся на шифт-клавиши, триггерные клавиши, клавиши с буферизацией расширенного кода и специальные клавиши.
- Буфер клавиатуры: Буфер BIOS для записи кодов клавиш занимает 32 байта оперативной памяти с адреса 40:1Eh до 40:3Eh. Буфер организуется как кольцевая очередь, доступ к которой осуществляется с помощью указателей "головы" и "хвоста".

Ввод информации с клавиатуры средствами MS-DOS

MS-DOS имеет группу функций прерывания 21h для выполнения ввода информации с клавиатуры:

- AH=01h: Ввод с ожиданием со стандартного устройства ввода с эхом на экран.
- AH=06h: Ввод-вывод с консоли без ожидания.

- АН=07h: Ввод с консоли с ожиданием без эха на экран.
  - АН=08h: Подобна АН=07h, но вызывает прерывание 23h при нажатии Ctrl-Break.
  - АН=0Bh: Проверка состояния стандартного ввода.
  - АН=0Ch: Ввод с клавиатуры с очисткой буфера.
- Ввод информации с клавиатуры средствами BIOS
- Интерфейсом программ с клавиатурой является прерывание 16h BIOS:
- АН=00h: Чтение с ожиданием двухбайтового кода из буфера клавиатуры.
  - АН=01h: Чтение без ожидания двухбайтового кода из буфера клавиатуры.
  - АН=02h: Определение состояния шифт- и триггерных клавиш.
  - АН=05h: Имитация нажатий клавиш в демонстрационных программах.

### Задание на лабораторную работу

1. Разработать, написать и отладить программу управления перемещением символа ("\*") в пределах заданного на экране окна с координатами (x1 - 20, y1 - 5, x2 - 60, y2 - 15) с постоянным видом движения. Для управления использовать клавиши F1 (для перемещения влево) и F2 (для перемещения вправо). Для ввода использовать стандартные функции языка C++ (getch, getche, kbhit, bioskey). Сохранить отлаженную программу.

2. Изменить программу, заменив стандартные функции библиотеки C++(getch, getche, kbhit, bioskey) своими. Для написания функций используйте заданное прерывание INT 21h. Если его возможностей недостаточно, то замените его по своему усмотрению. Сохраните отлаженную программу.

3. Две отлаженные программы предъявить преподавателю.

### Алгоритмы и тексты отлаженных программ

Программа с использованием стандартных функций C++

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>

int main(){
    int x1=20, y1=5, x2=60, y2=15; // window coordinates
    int screen_x, screen_y, key = 0; // coordinates of * and value of entered key
    clrscr();
    textbackground(0);
```

```

window(x1, y1, x2, y2); // Set the window with coordinates (x1, y1) to (x2,
y2)
screen_x = (x2 - x1) / 2; // middle x coordinates
screen_y = (y2 - y1) / 2; // middle y coordinates
gotoxy(screen_x, screen_y); // cursor to middle coordinates
cprintf("*");
gotoxy(screen_x, screen_y); // go cursor under the symbol

while(key != 27){ // 27 - key ESC
    key = getch();
    if(key == 60){ // F2
        while(wherex() < (x2-x1)){ // while right border not reached
            if(kbhit()){ // if we interrupt the move
                key = getch();
                if(key == 59){ // F1 - to move will be stopped
                    key = 0;
                    break;
                }
                else if (key == 27) return 0; // ESC
            }
            cprintf(" *");
            gotoxy(wherex() - 1, wherey()); // go cursor under the symbol
            delay(100);
        }
    }
    if(key == 59){ // F1
        while(wherex() > 1){ // while left border npt reached
            if(kbhit()){ // if we interrupt the move
                key = getch();
                if(key == 60) break; // F2 - to move will be stopped
                else if (key == 27) return 0; // ESC
            }
            cprintf(" "); // change current symbol on empty
            gotoxy(wherex() - 2, wherey()); // get back on 2 symbols (cause
cprintf(" ") change position of cursor on +1)
            cprintf("*");
            gotoxy(wherex() - 1, wherey()); // go cursor under the symbol
            delay(100);
        }
    }
}
return 0;
}

```

## Программа с использованием собственных функций

```

#include <stdio.h>
#include <conio.h>
#include <dos.h>

int my_getch() { // function of reading the key

```

```

union REGS regs;
regs.h.ah = 0x00; // set the function 00h int16h interuption to read the key
int86(0x16, &regs, &regs); // call int16h intruption
return regs.x.ax; // return external code from register ax
}

int my_kbhit() { // function of pressing the key
    union REGS regs;
    regs.h.ah = 0x01; // set the function 01h int16 interuption to check is key
    pressed
    int86(0x16, &regs, &regs); // call int16h interuption
    return !(regs.x.flags & 0x40); // check zf flag (if 1 - key not pressed)
}

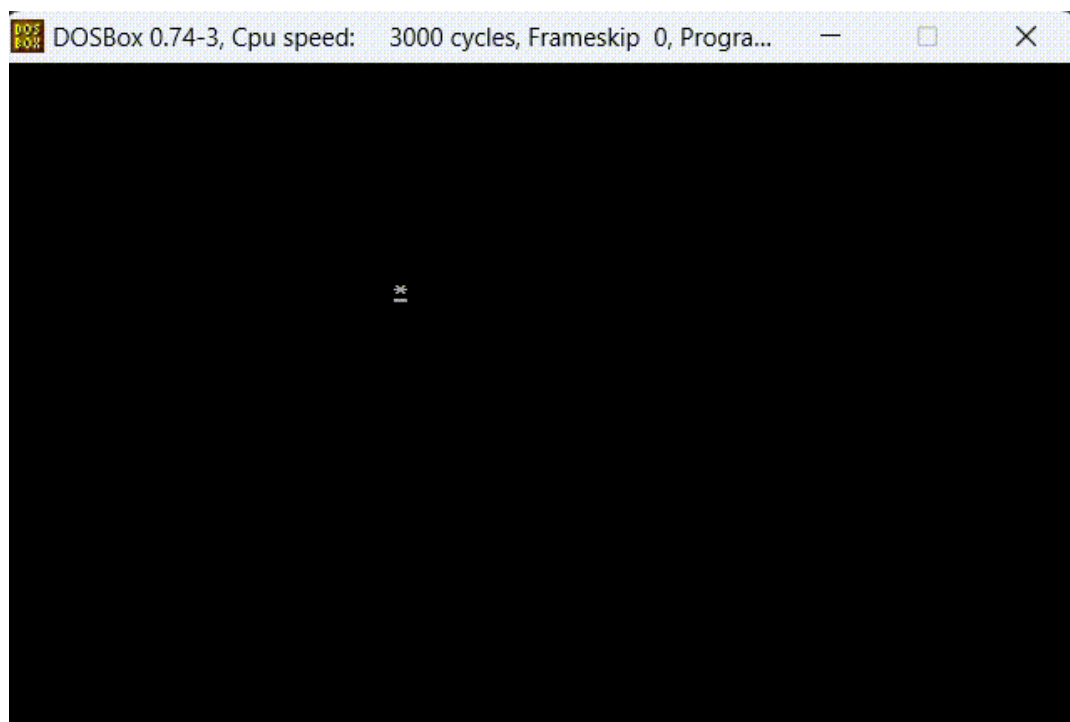
int main() {
    int x1 = 20, y1 = 5, x2 = 60, y2 = 15; // window coordinates
    int screen_x, screen_y, key = 0; // coordinates of * and value of entered key
    clrscr();
    textbackground(0);
    window(x1, y1, x2, y2); // set the window with coordinates (x1, y1) to (x2,
    y2)
    screen_x = (x2 - x1) / 2; // middle x coordinates
    screen_y = (y2 - y1) / 2; // middle y coordinates
    gotoxy(screen_x, screen_y); // cursor to middle coordinates
    cprintf("*");
    gotoxy(screen_x, screen_y); // go cursor under the symbol

    while (key != 0x011B) { // 0x011B - key ESC
        key = my_getch();
        if (key == 0x3C00) { // F2
            while (wherex() < (x2 - x1)) { // while right border not reached
                if (my_kbhit()) { // if we interupt the move
                    key = my_getch();
                    if (key == 0x3B00) { // F1 - to move will be stopped
                        key = 0;
                        break;
                    } else if (key == 0x011B) return 0; // ESC
                }
                cprintf(" *");
                gotoxy(wherex() - 1, wherey()); // go cursor under the symbol
                delay(100);
            }
        }
        if (key == 0x3B00) { // 0x3B00 - key F1
            while (wherex() > 1) { // while left border not reached
                if (my_kbhit()) { // if we interupt the move
                    key = my_getch();
                    if (key == 0x3C00) break; // F2
                    else if (key == 0x011B) return 0; // ESC
                }
                cprintf(" "); // change current symbol on empty
            }
        }
    }
}

```

```
        gotoxy(wherex() - 2, wherey()); // get back on 2 symbols (cause
cprintf(" ") change position of cursor on +1)
        cprintf("*");
        gotoxy(wherex() - 1, wherey()); // go cursor under the symbol
        delay(100);
    }
}
}
return 0;
}
```

## Пример запуска программы



## Выводы

В ходе выполнения лабораторной работы была достигнута цель изучения возможностей работы с клавиатурой, а также ознакомления со стандартными средствами библиотеки C++ и средствами системы прерываний DOS и BIOS, обслуживающими клавиатуру.