**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 |  | Шарпинский Д. А. |
| Преподаватель |  | Гречухин М. Н. |

Санкт-Петербург

2024

**Цель работы**

Изучение возможностей работы с клавиатурой, ознакомление со стандартными средствами библиотеки C++ и средствами системы прерываний DOS и BIOS, обслуживающими клавиатуру.

**Задание на лабораторную работу**

1. Разработать, написать и отладить программу управления пе¬ремещением символа (например, "\*") в пределах заданного на экране окна. Для управления использовать клавиши из набора: "стрелка вверх" (СтВВ), "стрелка вниз" (СтВН), "стрелка вправо" (СтВП), "стрелка влево" (СтВЛ) или функциональные клавиши Fl - F12 (варианты см. в таблице 4.2). 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 или INT 16h (см. таблицу). Если его возможностей не достаточно, то замените его по своему усмотрению. Сохраните отлаженную программу.

3. Две отлаженные программы предъявить преподавателю.

Вариант 17:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № варианта | X1 | Y1 | X2 | | Y2 | Вид движения | Клавиши управления | Номер прерывания |
| 17 | 10 | 15 | 70 | 20 | | Постоянное | СтВВ, СтВН | INT 21h |

**Алгоритм и подход**

1. Настройка окна:

Окно отображается в пределах экрана, ограниченное координатами (X1, Y1) и (X2, Y2). Для рисования окна используются функции вывода символов в нужных координатах.

1. Перемещение символа:

Символ, например "\*", перемещается по экрану с помощью клавиш стрелок. В каждой итерации проверяется, была ли нажата клавиша, и в зависимости от направления клавиши изменяются координаты символа.

1. Программирование с использованием стандартных функций:

Для реализации стандартной программы использовались функции getch() и kbhit(), которые позволяют получать символ с клавиатуры и проверять, была ли нажата клавиша.

1. Переписывание программы с использованием прерываний INT 21h:

После отладки программы с использованием стандартных функций была переписана версия программы с использованием прерываний INT 21h для получения символа с клавиатуры и проверки нажатия клавиш.

**Текст программы**

#include <dos.h>

#include <stdio.h>

#define X1 10

#define Y1 15

#define X2 70

#define Y2 20

#define DELAY 150

// KoopDuHaTbI u HaI7paBJIeHue DBu}|{eHu9I

typedef struct {

int x;

int y;

int dx;

int dy;

} Point;

int my\_getch() {

union REGS in, out;

in.h.ah = 0x07;

int86(0x21, &in, &out);

return out.h.al;

}

int my\_kbhit() {

union REGS in, out;

in.h.ah = 0x0B;

int86(0x21, &in, &out);

return out.h.al != 0;

}

void draw\_window() {

int i;

for (i = X1; i <= X2; i++) {

gotoxy(i, Y1);

printf("-");

gotoxy(i, Y2);

printf("-");

}

for (i = Y1; i <= Y2; i++) {

gotoxy(X1, i);

printf("|");

gotoxy(X2, i);

printf("|");

}

}

void render(Point p) {

gotoxy(p.x, p.y);

printf("\*");

}

void clear(Point p) {

gotoxy(p.x, p.y);

printf(" ");

}

void update\_direction(Point \*p) {

int ch = my\_getch();

if (ch == 27) { // ESC key to exit

p->dx = 0;

p->dy = 0;

return;

}

// Arrow keys => two bytes

if (ch == 0) {

ch = my\_getch();

if (ch == 72) { // Up arrow

p->dx = 0;

p->dy = -1;

} else if (ch == 80) { // Down arrow

p->dx = 0;

p->dy = 1;

} else if (ch == 75) { // Left arrow

p->dx = -1;

p->dy = 0;

} else if (ch == 77) { // Right arrow

p->dx = 1;

p->dy = 0;

}

}

}

void update\_position(Point \*p) {

clear(\*p);

if (p->dy == -1 && p->y > Y1 + 1) { // Upward

p->y--;

} else if (p->dy == 1 && p->y < Y2 - 1) { // Downward

p->y++;

} else if (p->dx == -1 && p->x > X1 + 1) { // Left

p->x--;

} else if (p->dx == 1 && p->x < X2 - 1) { // Right

p->x++;

}

render(\*p);

}

int main() {

Point point = {

(X1 + X2) / 2,

(Y1 + Y2) / 2,

0,

0

};

clrscr();

draw\_window();

render(point);

while (1) {

if (my\_kbhit()) {

update\_direction(&point);

if (point.dx == 0 && point.dy == 0) {

break; // Exit if ESC was pressed

}

}

if (point.dx != 0 || point.dy != 0) {

update\_position(&point);

delay(DELAY);

}

}

clrscr();

return 0;

}

**Выводы**

Использование прерываний для обработки ввода с клавиатуры позволяет более эффективно управлять взаимодействием с пользователем на низком уровне. В результате выполнения работы были достигнуты все поставленные цели, и программа была успешно отлажена и протестирована.

Также, поскольку в задании дано очень узкое по высоте окно, было принято решение расширить функционал программы и добавить возможность движения не только вверх-вниз, но и влево-вправо! Ведь техническая реализация не отличается практически ничем.