**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»**

|  |  |  |
| --- | --- | --- |
| Студенты |  |  |
| Преподаватель |  | Гречухин М.Н. |

Санкт-Петербург

2022

**Цель работы**

Изучить работу с видеосистемой в графическом режиме, вывод графика заданной функции с масштабированием и разметкой осей

**Задание (вариант №13)**

1. Разработать, написать и отладить программу управления постоянным перемещением символа "\*" в пределах заданного на экране окна с координатами (10,12,70,23). Для управления использовать клавиши "стрелка вверх", "стрелка вниз".

Для ввода использовать стандартные функции языка C++ (getch, getche, kbhit, bioskey). Сохранить отлаженную программу.

2. Изменить программу, заменив стандартные функции библиотеки C++ своими. Для написания функций используйте заданное прерывание INT 16h.

Если его возможностей не достаточно, то замените его по своему усмотрению. Сохраните отлаженную программу.

3. Две отлаженные программы предъявить преподавателю.

**Программная среда**

Вся программа написана на языке программирования Turbo С++, компилировалась в DOS BOX.

**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. Ввод информации в компьютер может быть выполнен на трех уровнях: обращением к функциям MS-DOS; обращением к функциям BIOS; физическим доступом к аппаратным средствам.

Ввод информации на уровне MS-DOS позволяет "пропустить" клавиатурный ввод через инсталлируемые драйверы, обеспечивает отслеживание нажатия комбинации клавиш Ctrl-C (Ctrl-Break), стандартную для MS-DOS обработку ошибок.

Доступ к клавиатуре на уровне BIOS позволяет программе отслеживать нажатие всех, а не только символьных клавиш, выполнять управление аппаратурой клавиатуры и пр. Интерфейсом Turbo С с BIOS является функция bioskey().

Непосредственный доступ к буферу клавиатуры резко повышает производительность программы. В некоторых случаях необходима имитация нажатий клавиш клавиатуры с записью кодов непосредственно в буфер. При этом физически нажатия клавиш не происходят.

Клавиатура персонального компьютера содержит специальный встроенный микропроцессор. Он при каждом нажатии и отпускании клавиши определяет ее порядковый номер и помещает его в порт 60h специальной электронной схемы, название же этого кода — скан-код. Когда скэн-код записан в порт 60h, схема ППИ выдает сигнал "подтверждения", уведомляя микропроцессор клавиатуры о принятии кода. Когда скэн-код принят схемой ППИ, аппаратура компьютера генерирует аппаратное прерывание с номером 9. Стандартный обработчик прерывания 9 - это программа, входящая в состав BIOS (BIOS ISR). BIOS ISR анализирует скэн-код и по специальным правилам преобразует его. Отметим, что по скэн-коду всегда можно установить, вследствие чего ISR получила управление: из-за нажатия или из-за отпускания клавиши.

Буфер BIOS для записи кодов клавиш занимает 32 байта оперативной памяти с адреса 40:lEh до 40:3Eh. Запись информации в буфер выполняет ISR BIOS прерывания 9, чтение - функции ISR BIOS прерывания 16h. Буфер клавиатуры рассчитан на 15 нажатий клавиш, генерирующих двухбайтовые коды и поэтому имеет 30 байт для кодов клавиш и еще два дополнительных байта, которые резервируются под двухбайтовый код для клавиши ENTER.

Буфер организуется как кольцевая очередь, доступ к которой осуществляется с помощью указателя «головы» (head pointer), адрес которого 40:1Ah, и указателя «хвоста» (tail pointer), адрес которого 40:1Ch.

Указатель "хвоста" задает смещение до слова, где будет записан обработчиком прерывания 9 код буферизуемой клавиши, т.е. первое свободное слово буфера. Указатель "головы" задает смещение слова, которое будет возвращено запросу буфе­ризованного ввода с клавиатуры, сделанного операционной системой или BIOSoм.

MS-DOS имеет целую группу функций прерывания 21h для выполнения ввода информации с клавиатуры. Последовательность действий системы при вводе с клавиатуры такова. Функция MS-DOS вызывает драйвер клавиатуры, передавая ему запрос на ввод одного символа из буфера клавиатуры. Драйвер, выполняя запрос, обращается к нужной функции прерывания 16h BIOS. ISR BIOS прерывания 16h читает из буфера клавиатуры нужное слово и передает в драйвер. Драйвер возвращает байт (обычно младший) в MS-DOS. Таким образом, функции MS-DOS и опирающиеся на них функции библиотеки Turbo С слабо зависят от особенностей аппаратуры, поскольку система от нее изолирована двумя слоями программного обеспечения - драйверами и BIOSом.

В работе же использовались следующие функции и прерывания:

* int bioskey(int cmd)

Обращается в зависимости от значения в cmd к функциям АН = 00 - 02h прерывания 16h. Возвращаемое функцией значение повторяет значение регистра АХ при выходе из прерывания.

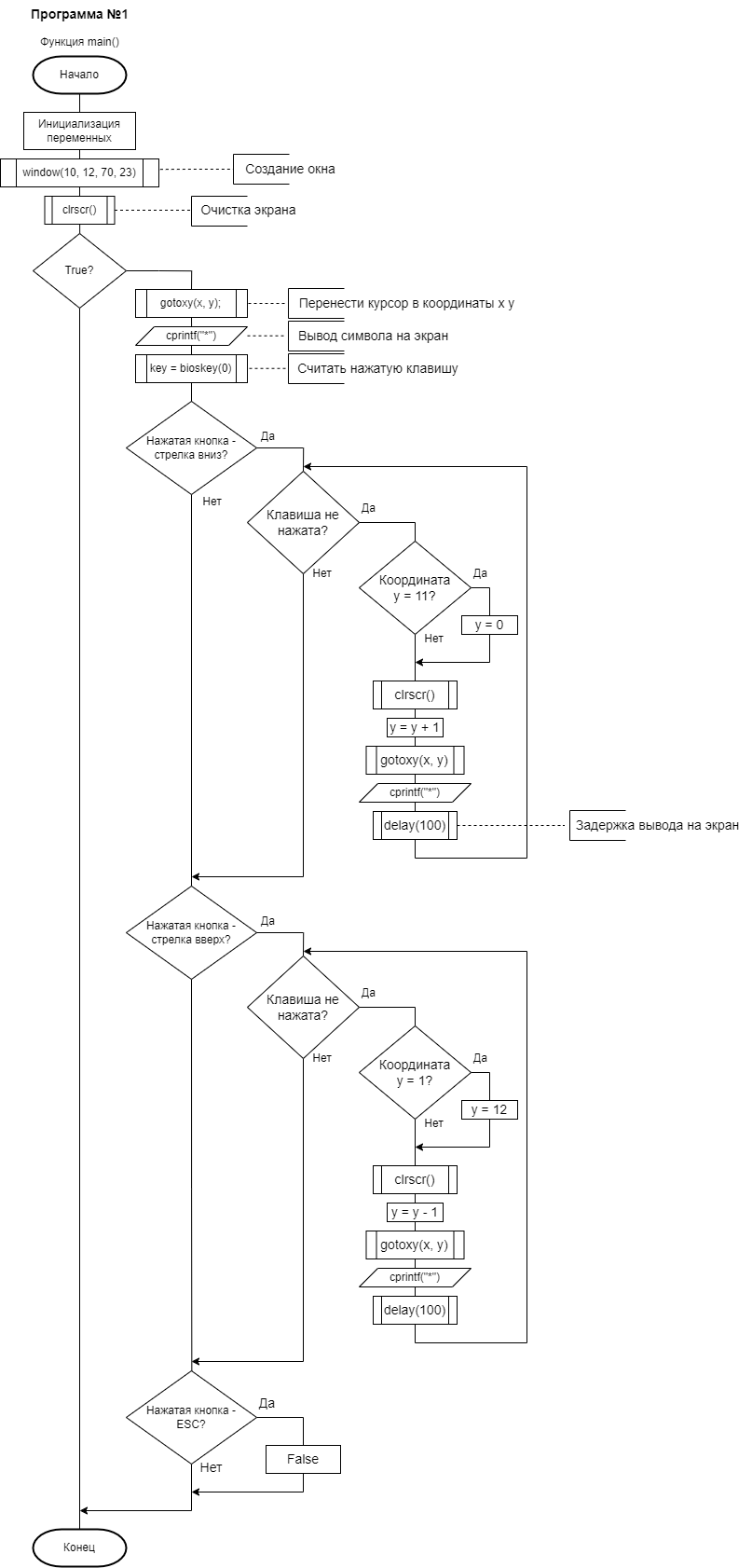
* Прерывание 16h с функциями 0, 1 , где:

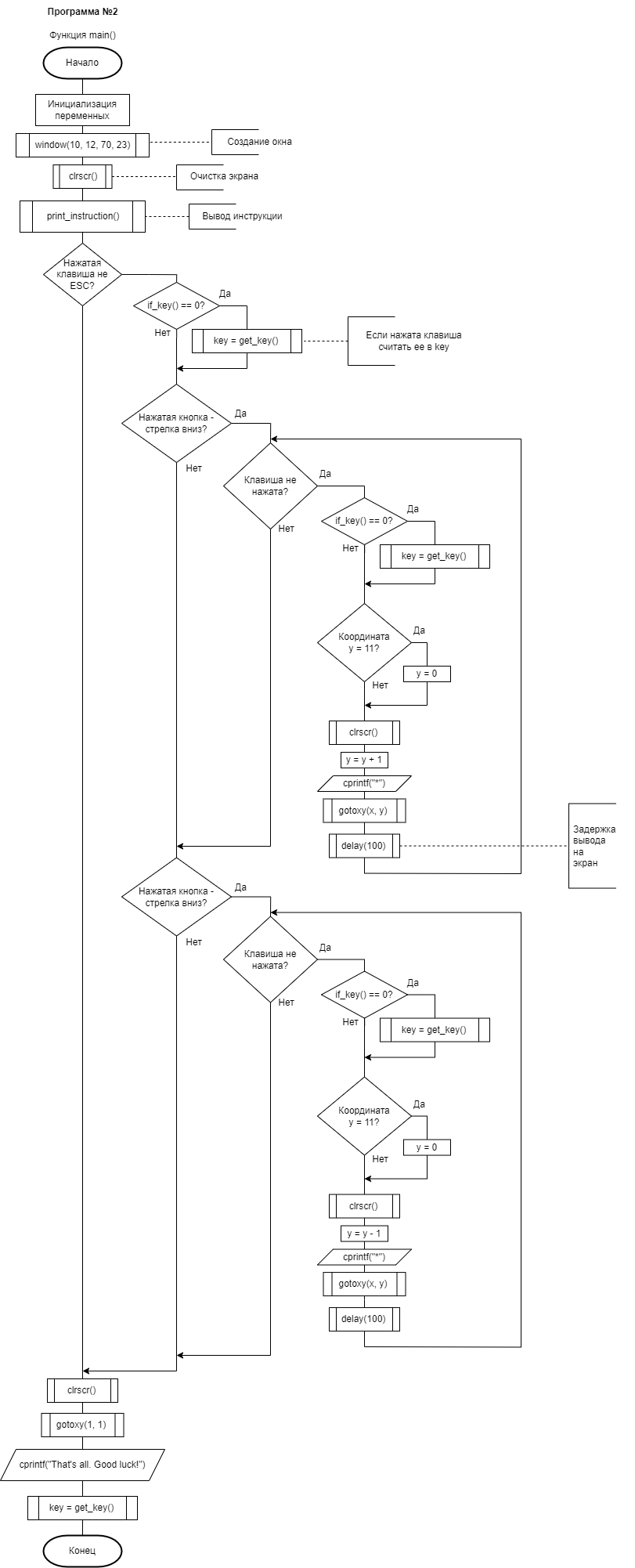
0 - 00H читать (ожидать) следующую нажатую клавишу

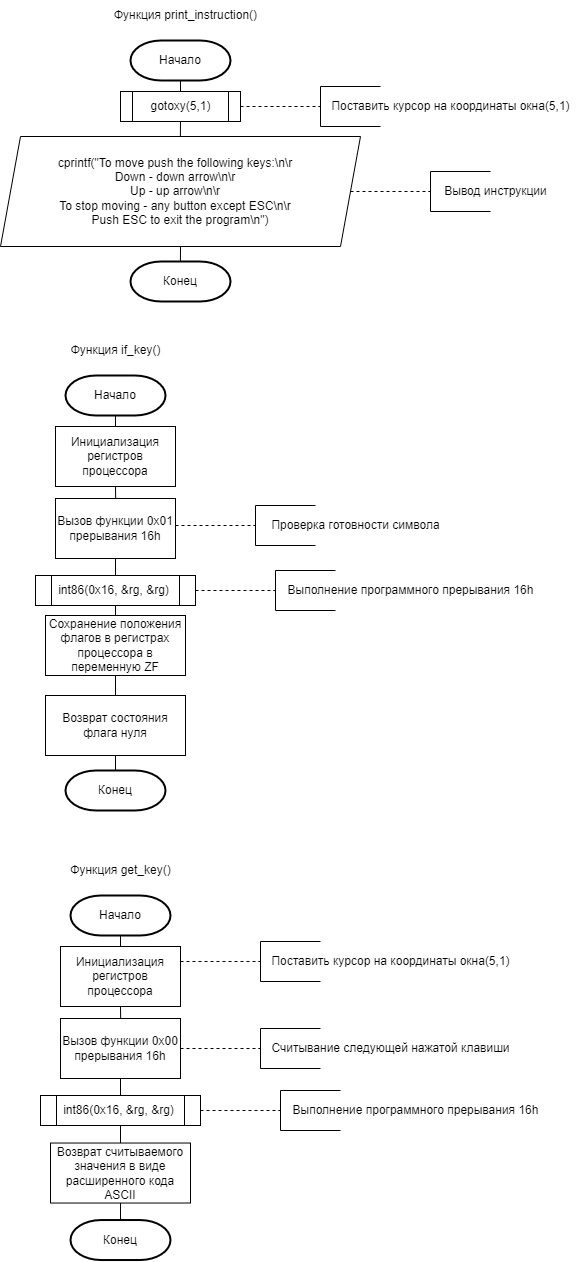
1 - 01H проверить готовность символа (и показать его, если так)

выход: ZF = 1 если символ не готов.

ZF = 0 если символ готов.

**Схемы алгоритмов**





**Текст программы**

**Программа №1**

#include <dos.h>

#include <conio.h>

#include <bios.h>

int main()

{

int key, x = 1, y = 1, status;

window(10, 12, 70, 23);

clrscr();

while (1)

{

gotoxy(x, y);

cprintf("\*");

while (bioskey(1) == 0);

key = bioskey(0);

if (key == 20480)

{

while(bioskey(1)==0)

{

if(y==11) y==0;

clrscr();

y++;

gotoxy(x, y);

cprintf("\*");

delay(100);

}

}

if (key == 18432)

{

while(bioskey(1)==0)

{

if(y==1) y==12;

clrscr();

y--;

gotoxy(x, y);

cprintf("\*");

delay(100);

}

}

if (key == 283) break;

}

return 0;

}

**Программа №2**

#include <dos.h>

#include <conio.h>

#include <bios.h>

#include <stdio.h>

int if\_key()

{

union REGS rg;

rg.h.ah = 0x01;

int86(0x16, &rg, &rg);

unsigned int ZF = 1<<6;

return rg.x.flags & ZF;

}

int get\_key()

{

union REGS rg;

rg.h.ah = 0x00;

int86(0x16, &rg, &rg);

return rg.h.ah;

}

void print\_instruction()

{

gotoxy(5,1);

cprintf("To move push the following keys:\n\r Down - down arrow\n\r Up - up arrow\n\r To stop moving - any button except ESC\n\r Push ESC to exit the program\n");

}

int main(){

int x = 1,y = 1;

char key = 0;

clrscr();

window(10,12,70,23);

print\_instruction();

gotoxy(x,y);

cprintf("\*");

while(key != 1)

{

if (if\_key() == 0) key = get\_key(); //If button pushed

if (key == 80)

{

while(if\_key() != 0)

{

if (if\_key == 0) key = get\_key();

if (y == 11) y = 1;

clrscr();

y++;

gotoxy(x, y);

cprintf("\*");

delay(100);

}

}

else

{

if (key == 72)

{

while(if\_key() != 0)

{

if (if\_key == 0) key = get\_key();

if (y == 1) y = 11;

clrscr();

y--;

gotoxy(x, y);

cprintf("\*");

delay(100);

}

}

}

}

clrscr();

gotoxy(1,1);

cprintf("That's all. Good luck!");

key = get\_key();

return 0;

}