**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

**отчет**

**по лабораторной работе №4**

**по дисциплине «Организация ЭВМ и Систем»**

**Тема: «Клавиатура IBM PC. Использование прерываний»**

**Вариант 4**

|  |  |  |
| --- | --- | --- |
| Студент гр. 0302 |  | Чёрный Я.И. |
| Преподаватель |  | Жандаров В.В. |

Санкт-Петербург

2021

**Оглавление**

[1 Постановка задачи 3](#_Toc90506627)

[1.1 Дополнительная задача 3](#_Toc90506628)

[2 Краткие сведения о подсистеме ввода информации с клавиатуры, используемых прерываниях, буфере клавиатуры и функциях обслуживания ввода с клавиатуры 3](#_Toc90506629)

[3 Блок-схема программы 5](#_Toc90506630)

[4 Пример работы 6](#_Toc90506631)

[5 Текст программы 6](#_Toc90506632)

[6 Вывод 9](#_Toc90506633)

# Постановка задачи

Целью данной задачи является разработка и отладка программы управления пе­ремещением 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. Для ввода использовать стандартные функции языка C++.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № варианта | X1 | Y1 | X2 | Y2 | Вид движения | Клавиши управления | Номер прерывания |
| 4 | 25 | 5 | 55 | 15 | Пошаговое | Все направления | INT 21h |

## Дополнительная задача

Изменить программу, заменив стандартные функции библиотеки C++ своими. Для написания функций используйте заданное прерывание, если его возможностей достаточно. Если его возможностей недостаточно, то замените его по своему усмотрению. Сохраните отлаженную программу.

# 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().

Действия BIOS ISR при нажатии и отпускании одной и той же клавиши различны. Клавиши в зависимости от алгоритма обработки их скэн-кода можно разделить на:

1. шифт-клавиши (Right-Shift, Left-Shift, Alt, Ctrl);
2. триггерные клавиши (NumLock, ScrollLock, CapsLock);
3. клавиши с буферизацией расширенного кода;
4. специальные клавиши (клавиша PrnScr, комбинация Alt-Ctrl-Del, комбинация Ctrl-C (Ctrl-Break)).

За каждой шифт- или триггерной клавишей закреплен свой бит в ячейках памяти по адресам 40: 17h и 40: 18h (табл. 4.1). При каждом нажатии или отпускании шифт-клавиши ISR BIOS инвертирует соответствующий бит. Таким образом, текущее состояние бита шифт-клавиши говорит о том, нажата она в данный момент или отпущена. За триггерными кла­вишами закреплены два бита: один из них инвертируется только при нажатии клавиши ("фиксирует" состояние "Вкл/Выкл"), другой - при нажатии и отпускании, отслеживая текущее состояние клавиши.

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.

Далее приводится характеристика функций MS-DOS, используемых для ввода с клавиатуры.

AH=01h - ввод с ожиданием со стандартного устройства ввода (клавиатуры). Выполняется "эхо" на экран вводимых символов. ASCII-код прочитанного символа помещается в AL. Если нажимается специаль­ная клавиша, в AL возвращается 0, а второе обращение к функции возвращает расширенный скэн-код клавиши.

AH=06h - ввод-вывод с консоли. Если DL = FFh, выполняется ввод со стандартного устройства ввода без ожидания. Если буфер пуст, функция сообщает об этом установленным в 1 флагом нуля (ZF). В противном случае в регистре AL возвращается ASCII-код прочитанного символа.

AH=07h - ввод с консоли с ожиданием без "эха" на экран. ASCII-код прочитанного символа возвращается в AL. Если нажимается специальная клавиша, передаваемое в AL значение равно нулю, а второе обращение к функции возвращает расширенный скэн-код клавиши. Функция не выполняет "фильтрацию" ввода с клавиатуры. Это значит, что нажатие клавиши Backspace не стирает символ на экране, а только сдвигает курсор. Нажатие ENTER не переводит строку, а только перемещает курсор на начало строки.

AH=08h - подобна АН=07h, за исключением того, что если обнаруживается нажатие комбинации клавиш Ctrl-Break, вы­зывается прерывание 23h.

AH=0Bh - проверка состояния стандартного ввода. Возвращает в регистре AL значение FFh, если буфер клавиатуры не пуст, и 0 в противном случае. Функцию следует использовать перед выполнением функций АН=01h, 07h и 08h для того, чтобы избежать ожидания ввода, если он отсутствует. Кроме того, функция используется как средство проверки того, нажата ли комбинация клавиш Ctrl-Break, если программа долгое время выполняет работу, не связанную с обращением к функциям MS-DOS. Периодическое выполнение функции позволяет аварийно завершить программу, например, в случае ее зацикливания.

Функции АН = 00 - 02h прерывания 16h BIOS положены в основу функции bioskey() библиотеки Turbo С. Далее следует описание этой функции.

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.

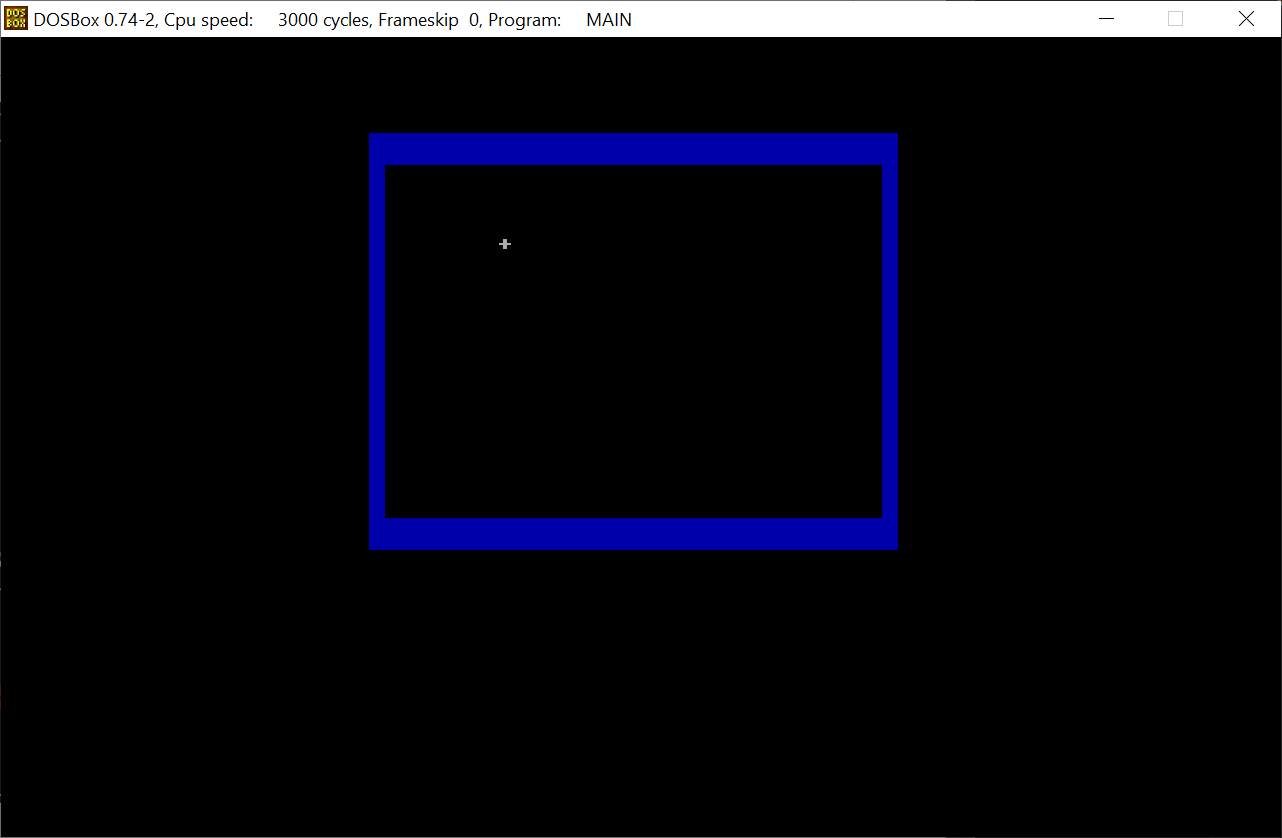
int getch (void)

Выполняет ввод с клавиатуры через функцию MS-DOS АН=07h. Она не выполняет "эхо" вывода на экран. В этой связи полезна для организации интерфейса с пользователем, при котором нажатие той или иной клавиши вызывает немед­ленную реакцию программы без отображения введенного символа на экране.

# Блок-схема программы



# Пример работы



# Текст программы

MAIN.CPP

|  |
| --- |
| #include <dos.h>  #include <conio.h>  #include <stdio.h>  #define KEY\_UP 72  #define KEY\_DOWN 80  #define KEY\_RIGHT 77  #define KEY\_LEFT 75  #define KEY\_ESC 27  #define KEY\_UPRIGHT 73  #define KEY\_UPLEFT 71  #define KEY\_DOWNRIGHT 81  #define KEY\_DOWNLEFT 79  void SetCursorVisibility(int value)  {  //use 01h function of int 10h  asm mov ah, 0x01  if (value > 0)  asm mov cx, 0x0607  else  asm mov cx, 0x2607  asm int 0x10  }  char GetINT21Key()  {  //use 07h function int 21h  union REGS regs;  regs.h.ah = 0x07;  int86(0x21, &regs, &regs);  return regs.h.al;  }  class Window  {  int x1, x2, y1, y2;  //functions to clamp cursor within the window  void clampY(int& y)  {  if (y > y2 - y1 + 1)  y = y2 - y1 + 1;  if (y < 1)  y = 1;  }  void clampX(int& x)  {  if (x > x2 - x1 + 1)  x = x2 - x1 + 1;  if (x < 1)  x = 1;  }  void ClearWindow(char l\_row, char l\_col, char r\_row, char r\_col, char attr)  {  union REGS regs;  regs.h.al = 0x00;  regs.h.ah = 0x06;  regs.h.ch = l\_row - 1;  regs.h.cl = l\_col - 1;  regs.h.dh = r\_row - 1;  regs.h.dl = r\_col - 1;  regs.h.bh = attr;  int86(0x10, &regs, &regs);  }  public:  Window(int x1, int y1, int x2, int y2)  {  this->x1 = x1;  this->x2 = x2;  this->y1 = y1;  this->y2 = y2;  clrscr();  window(x1, y1, x2, y2);  //draw border --> blue foreground black text  DrawBorder(0x10);  SetCursorVisibility(0);  InputCycle();  }  void DrawBorder(char colorAttr)  {  char borderSize = 1;  ClearWindow(y1 - borderSize, x1 - borderSize, y2 + borderSize, x2 + borderSize, colorAttr);  //draw window black foreground white text  ClearWindow(y1, x1, y2, x2, 0x07);  }  void Dispose()  {  ClearWindow(0, 0, 255, 255, 0x07);  }  void InputCycle()  {  int cursorX = 0, cursorY = 0, exitFlag = 0;  while (exitFlag == 0)  {  //cpp realization  //char input = getch();  // int21 realization  char input = GetINT21Key();  switch (input)  {  case KEY\_ESC: exitFlag = 1; break;  case KEY\_UP: cursorY--; break;  case KEY\_DOWN: cursorY++; break;  case KEY\_RIGHT: cursorX++; break;  case KEY\_LEFT: cursorX--; break;  case KEY\_UPRIGHT: cursorY--; cursorX++; break;  case KEY\_DOWNRIGHT: cursorY++; cursorX++; break;  case KEY\_UPLEFT: cursorY--; cursorX--; break;  case KEY\_DOWNLEFT: cursorY++; cursorX--; break;  }  clampX(cursorX);  clampY(cursorY);  clrscr();  gotoxy(cursorX, cursorY);  cputs("+");  }  Dispose();  }  };  int main()  {  clrscr();  int x1 = 25, x2 = 55, y1 = 5, y2 = 15;  Window mainWindow(x1, y1, x2, y2);  return 1;  } |

# Вывод

В результате выполнения данной работы был получен опыт работы с методами ввода с клавиатуры, системными прерываниями.