**Лабораторна робота №4**

**Дослідження роботи відеоадаптера**

**Мета:** засвоїти принципи роботи відеоадаптера; отримати навики використання переривання відеосервісу для виводу інформації.

**Теоретичні відомості:**

***Відеоадаптер*** – це пристрій, призначений для обробки, генерації зображень з подальшим їх виведенням на екран. Сучасні відеоадаптери не обмежуються лише звичайним виведенням зображень, вони мають вбудований графічний процесор, котрий може проводити додаткову обробку, звільняючи від цих задач центральний процесор. Відеоадаптер може бути окремим компонентом ПК або інтегрованим до складу материнської плати.

***Відеопам'ять*** – це доступна відеоадаптеру область оперативної пам'яті, в якій розміщені дані, що відповідають зображенню на екрані. Відеопам'ять може бути локальною для відеоадаптеру або виділена з основної оперативної пам'яті ПК (характерно для інтегрованих відеоадаптерів). У відеопам'яті може міститися як безпосередньо растровий образ зображення (екранний кадр), так і окремі фрагменти як в растровій (текстури), так і у векторній (геометричні примітиви) формах. В залежності від кількості інформації, необхідної для формування кадру зображення, відеопам'ять може містити певну кількість відеосторінок. Одна з відеосторінок є активною (тобто її зміст використовується для формування зображення, що виводиться у поточний час). Зміна інформації на неактивних відеосторінках не призводе до зміни поточного зображення, що дозволяє одночасно з виведенням поточного зображення формувати наступне зображення. Відеоадаптери підтримують два основних варіанта роботи – ***текстовий та графічний***. У текстовому режимі найменшим елементом зображення є *знакомісце*, яке може відображати один символ. У графічному режимі найменшим елементом зображення є *піксель*. Нумерація елементів починається з 0 від лівого верхнього краю екрана. Конкретний режим роботи характеризується кількістю елементів (знакомісць або пікселів) по вертикалі та горизонталі, а також додатковими параметрами елементів (атрибутом для знакомісць або кольором для пікселів).

Для роботи з зображенням використовуються різні способи, у тому числі переривання відеосервісу, яке надає вбудоване програмне забезпечення ПК. У таблиці наведено функції переривання відеосервісу int 10h.

|  |  |
| --- | --- |
| Код  функції | Опис функції |
| 00*h* | Встановлення режиму роботи відеоадаптера  Вхідні дані:  · AL – код режиму (якщо старший біт дорівнює 0, то  відбувається очистка екрану);  Отримані дані:  · Немає. |
| 02*h* | Встановлення позиції курсору  Вхідні дані:  · BH – номер відеосторінки;  · DL – номер рядка;  · DH – номер стовпчика;  Отримані дані:  · Немає. |
| 09*h* | Запис символу з атрибутами у поточну позицію курсору  Вхідні дані:  · AL – ASCII-код символу;  · BL – атрибут символу , де  R/G/B – червоний/зелений/синій колір символу, Y –  яскравість символу, r/g/b – червоний/зелений/синій  колір фону, M – режим миготіння символу;  · BH – номер відеосторінки;  · CX – кількість символів;  Отримані дані:  · Немає. |
| 0*Ah* | Запис символу з тим атрибутом, який був у позиції курсору  Вхідні дані:  · AL – ASCII-код символу;  · BH – номер відеосторінки;  · CX – кількість символів;  Отримані дані:  · Немає. |
| 0*Fh* | Визначення поточного режиму роботи відео адаптера  Вхідні дані:  · Немає;  Отримані дані:  · AL – код режиму;  · AH – кількість символів у рядку;  · BH – номер відеосторінки. |

**Завдання:**

1. Очистити екран;

2. Вивести рядок символів, який завдано користувачем (кількість символів – від 3 до 76), зі зміною кольору символів;

3. Сформувати рамку з символів без зміни кольору навколо виведеного рядка.

**Код:**

uses crt;  
var  
sen:string;  
code:char;  
l,i,w:integer;  
page,atr,posx,posy:byte;  
label  
step1;  
begin  
clrscr;  
step1:  
write('Input your sentance [ 3 < symbols < 76 ] = '); readln(sen);  
l:=length(sen);  
if (l<3)or(l>76) then goto step1;  
posx:=round((80-l)[/2](tg://bot_command?command=2));  
posy:=12;  
  
asm  
mov ah,$0F  
int $10  
mov page,bh  
end;  
  
atr:=1;  
w:=l+2;  
For i:=1 to 3 do  
begin  
asm  
mov ah,2  
mov bh,page  
mov dh,posy  
mov dl,posx  
int $10  
  
mov ah,9  
mov al,'-'  
mov cx,w  
mov bh,page  
mov bl,atr  
int $10  
end;  
posy:=posy+1;  
end;  
  
posy:=posy-2;  
posx:=posx+1;  
For i:=1 to l do  
begin  
code:=sen[i];  
asm  
mov ah,2  
mov bh,page  
mov dh,posy  
mov dl,posx  
int $10  
  
mov ah,9  
mov al,code  
mov cx,1  
mov bh,page  
mov bl,atr  
int $10  
end;  
posx:=posx+1;  
atr:=atr+1;  
if atr>15 then atr:=1;  
end;  
readln;  
end.

**Результат:**

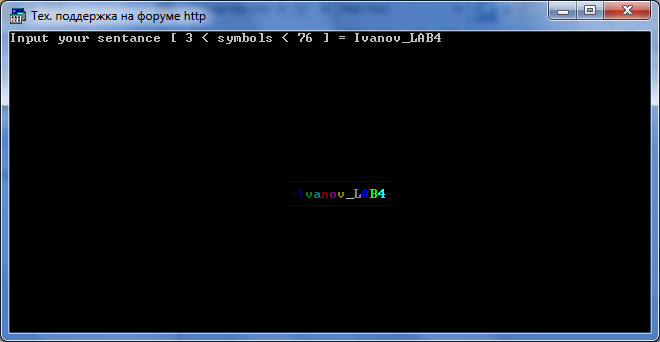


Рис.1. Результат роботи

**Висновок:** Я засвоїв принципи роботи відеоадаптера та отримав навики використання переривання відеосервісу для виводу інформації.