

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**ІКНІ**  
**Кафедра ПЗ**

**ЗВІТ**

до лабораторної роботи № 8

**на тему:** “ Використання функцій BIOS для роботи з відео в текстовому та графічному режимах”

**з дисципліни:** “*Архітектура комп’ютера*”

**Лектор:**  
доц. каф. ПЗ  
Крук О. Г.

**Виконав:**  
ст. гр. ПЗ-22  
Ясногородський Н.В.

**Прийняв:**  
доц. каф. ПЗ  
Крук О. Г.

« \_\_\_\_ » \_\_\_\_\_ 2022р.  
Σ = \_\_\_\_ \_

Тема: використання функцій BIOS для роботи з відео в текстовому та графічному режимах.

Мета: опанувати функції BIOS для роботи з відео в текстовому та графічному режимах; розвинути навички складання програм для виведення різнокольорових рядків символів та графічних зображень; відтранслювати і виконати в режимі відлагодження програми, складені відповідно до свого варіанту.

### Теоретичні відомості

У текстовому режимі прикладна програма може вивести інформацію на екран одним з таких способів.

- **За допомогою функцій MS DOS.** Якщо на комп'ютері встановлена система MS DOS або її емулятор, для виведення текстових даних на екран можна скористатися функціями переривання INT 21h. Дані функції дозволяють перенаправити потоки введення-виведення на будь-який інший пристрій, такий як принтер або диск. Виведення на екран за допомогою функцій переривання INT 21h виконується досить повільно, і колір символів змінити не можна.

- **За допомогою функцій BIOS.** Вивести символи на екран можна також за допомогою функцій переривання INT 10h, оброблення якого виконується системою BIOS, а не DOS. Вони виконуються набагато швидше, ніж функції переривання INT 21h, і дозволяють змінити колір тексту на екрані. При заповненні символами великих областей на екрані за допомогою функцій переривання INT 10h, можна помітити невелику затримку виведення. Крім того, дані, що виводяться на екран, не можна перенаправити на інший пристрій.

- **Прямий доступ в відеопам'ять.** Вивести символи на екран можна також шляхом переміщення їх безпосередньо в область пам'яті відеоадаптера. При цьому досягається максимальна швидкість виведення, проте дані також можна перенаправити на інший пристрій. На зорі розвитку ПК, коли основною операційною системою була MS DOS, в прикладних програмах, таких як текстові процесори і електронні таблиці, використовувався саме цей метод виведення даних на екран. Слід зазначити, що даний метод також можна використовувати при роботі програми в повноекранному режимі під керуванням операційних систем Windows NT, 2000 і XP.

Залежно від поставлених завдань, в застосунку може використовуватися один з трьох запропонованих вище способів виведення даних на екран. Якщо на перше місце ставиться швидкість виведення на екран, то потрібно скористатися прямим виведенням у відеопам'ять. В інших випадках слід віддати перевагу функціям BIOS. Функціями DOS варто користуватися тільки тоді, коли вихідний потік даних може бути перенаправлений на інший пристрій або коли екран спільно використовується кількома програмами. Слід зазначити, що для виведення даних на екран у функціях MS DOS використовуються функції BIOS, а у функціях BIOS - прямий доступ до відеопам'яті.

### Змішування основних кольорів

Колір кожного пікселя зображення визначається значенням струму трьох незалежних променів електронно-променевої трубки монітора: червоного, зеленого і синього. Існує ще один, четвертий канал керування кольором, який змінює загальну інтенсивність (тобто яскравість) всіх пікселів. Таким чином, значення всіх доступних кольорів в текстовому режимі можна визначити у вигляді одного 4-бітового двійкового числа, заданого в наступному форматі: I = інтенсивність, R = червоний, G = зелений, B = синій. На рис. 2 показано, як формується піксель білого кольору на екрані.

### Функції для роботи з відео переривання INT 10h

Табл. 2. Найбільш вживані функції переривання INT 10h

Функція	Опис
00h	Встановити текстовий або графічний режим відео з заданим номером
01h	Встановити форму і розмір курсору, вказавши номери початкового і кінцевого рядка відображення
02h	Перемістити курсор в зазначену позицію на екрані
03h	Визначити положення і розмір курсору
04h	Визначити положення і стан світового пера (застаріла)
05h	Відобразити на екрані відеосторінку із зазначеним номером (використовується рідко)
06h	Прокрутити вікно поточної відеосторінки вгору на вказану кількість рядків, замінюючи витіснені рядки пробілами
07h	Прокрутити вікно поточної відеосторінки вниз на вказану кількість рядків, замінюючи витіснені рядки пробілами
08h	Прочитати з екрану символ і його байт атрибутів, який визначається поточним положенням курсора
09h	Вивести на екран символ і його байт атрибутів в позицію, яка визначається поточним положенням курсора
0Ah	Вивести на екран тільки символ (без його байта атрибутів) в позицію, яка визначається поточним положенням курсора
0Bh	Встановити палітру кольорів відеоадаптера (використовується рідко)
0Ch	Вивести на екран піксель в графічному режимі
0Dh	Визначити колір пікселя в зазначеній позиції екрану
0Eh	Вивести символ на екран в графічному режимі і перемістити курсор на одну позицію вправо (використовується рідко)

0Fh	Визначити параметри поточного режиму відео
10h	Перемкнути режим миготіння відеоадаптера на режим керування інтенсивністю і навпаки
13h	Вивести рядок на екран в режимі емуляції телетайпа

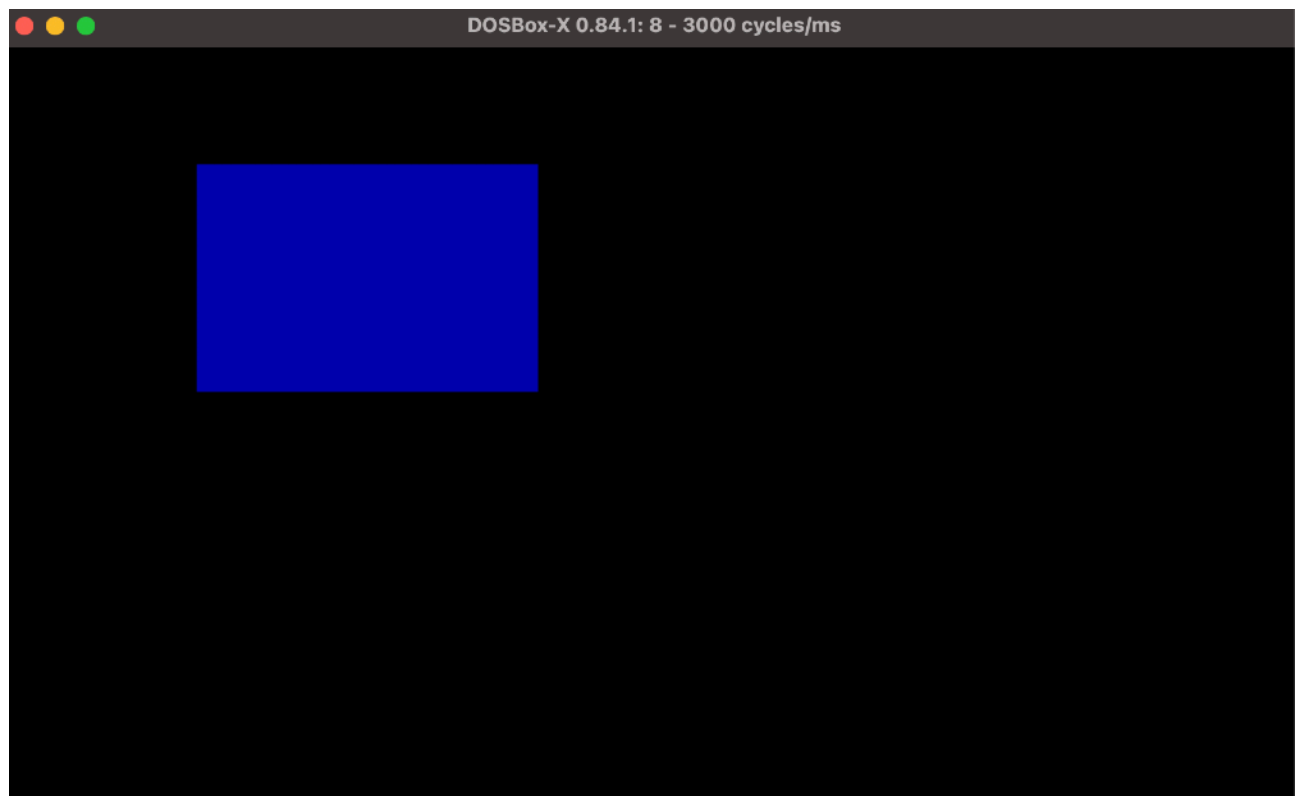
### Індивідуальне завдання

1. Опишіть рядки символів, в яких вкажіть прізвище, ім'я, по батькові.
2. Сформууйте байти атрибутів (різні) для кожного символу в кожному рядку символів.
3. Очистіть екран.
4. В текстовому режимі початковий номер рядка на екрані визначається як остача від ділення номера в списку групи на 10.
5. Початковий номер стовпця на екрані визначається як сума номера групи і номера в списку групи.
6. Шляхом безпосереднього записування тексту в першу сторінку текстової відеопам'яті виведіть рядок символів з прізвищем у початковий рядок, починаючи з початкового стовпця.
7. Рядок символів з іменем виведіть в рядок, номер якого дорівнює початковому+2, починаючи з стовпця, номер якого дорівнює початковому+3.
8. Рядок символів з по батькові виведіть в рядок, номер якого дорівнює початковому+6, починаючи з стовпця, номер якого дорівнює початковому+8.
9. Зробіть копію екрану.
10. В графічному режимі 13h побудуйте прямокутник, ліва верхня вершина якого розміщується в рядку, номер якого дорівнює кількості літер у прізвищі, і в стовпці, номер якого дорівнює кількості літер в по батькові. Довжина горизонтальної сторони прямокутника дорівнює потроєному номеру групи+номер в списку групи. Довжина вертикальної сторони прямокутника дорівнює подвоєному номеру групи+номер в списку групи. Колір прямокутника виберіть за остачею від ділення номера групи на 3: 0 – червоний; 1 – зелений; 2 – синій.
11. Перевірте результат роботи програми.
12. Зробіть копію екрану.
13. У звіті наведіть текст програми.

Номер в списку: 30

## Хід роботи

Написав код програми та вивів результати:



## Код програми

```
INCLUDE D:\Irvine\Irvine16.inc
```

```
; Константи відеорежимів
```

```
Mode_03 = 03h ; Текстовий режим 80x25, 16 кольорів, 8 сторінок
```

```
Mode_06 = 06h ; 640 x 200, 2 кольори
```

```
Mode_0D = 0Dh ; 320 x 200, 16 кольорів
```

```
Mode_0E = 0Eh ; 640 x 200, 16 кольорів
```

```
Mode_0F = 0Fh ; 640 x 350, 2 кольори
```

```
Mode_10 = 10h ; 640 x 350, 16 кольорів
```

```
Mode_11 = 11h ; 640 x 480, 2 кольори
```

```
Mode_12 = 12h ; 640 x 480, 16 кольорів
```

```
Mode_13 = 13h ; 320 x 200, 256 кольорів
```

```
Mode_6A = 6Ah ; 800 x 600, 16 кольорів
```

```
.data
```

```
saveMode BYTE ? ; Збережений поточний відеорежим
```

```
mySurname BYTE 'Y', 1Ch, 'a', 5Dh, 's', 1Fh, 'n',  
5Ah, 'o', 1Ah, 'g', 4Ah, 'o', 7Ch, 'r', 3Fh, 'o', 3Ch, 'd',  
7Ah, 's', 1Bh, 'k', 5Ch, 'y', 2Ah, 'i', 1Fh
```

```
myName BYTE 'N', 0Eh, 'i', 4Bh, 'k', 3Ah, 'i',  
3Dh, 't', 7Eh, 'a', 1Fh
```

```
myMiddleName BYTE 'V', 3Bh, 'i', 6Eh, 'c', 5Ah, 't',  
6Eh, 'o', 3Ch, 'r', 6Dh, 'o', 2Bh, 'v', 7Eh, 'i', 3Ah,  
'c', 2Ah, 'h', 7Dh
```

```
; variant = 15
```

```
row WORD 5 ; variant % 10
```

```
column WORD 37 ; variant + group number
```

```
xVal WORD ? ; Координата X
```

```
yVal WORD ? ; Координата Y
```

```
rectangleWidth WORD 111 ; 3 * (group number +  
variant) = 3 * (22 + 15) = 3 * 37 = 111
```

```
rectangleHeight WORD 74 ; 2 * (group number +  
variant) = 2 * (22 + 15) = 2 * 37 = 74
```

```
.code
```

```
main PROC
```

```
mov ax, @data
mov ds, ax
```

```
; Збережемо номер поточного відеорежиму
mov ah, 0Fh ; tells "int 10h" to return
current video mode
int 10h
mov saveMode, al
```

```
; Встановлення текстового режиму (Очищення екрану)
mov ax, Mode_03
int 10h
```

```
; Обрання першої сторінки
mov ah, 05h
mov al, 1
int 10h
```

```
; Запис тексту в першу сторінку текстової пам'яті
mov ax, 0B900h ; адреса першої сторінки
mov es, ax
```

```
; Виведення прізвища
mov cx, sizeof mySurname
mov si, OFFSET mySurname
xor di, di ; di = 0, but faster
```

```
mov bx, row
imul bx, 2
imul bx, 80
add di, bx
mov bx, column
imul bx, 2
add di, bx
```

```
call WRITE_TO_FIRST_PAGE_OF_TEXTMEMORY
```

```
; Виведення імені
mov cx, sizeof myName
mov si, OFFSET myName
xor di, di
```

```
mov bx, 6
add bx, 2
imul bx, 2
imul bx, 80
add di, bx
```

```
mov bx, column
add bx, 3
imul bx, 2
add di, bx
```

```
call WRITE_TO_FIRST_PAGE_OF_TEXTMEMORY
```

```
; Виведення по-батькові
mov cx, sizeof myMiddleName
mov si, OFFSET myMiddleName
xor di, di
```

```
mov bx, row
add bx, 6
imul bx, 2
imul bx, 80
add di, bx
mov bx, column
add bx, 8
imul bx, 2
add di, bx
```

```
call WRITE_TO_FIRST_PAGE_OF_TEXTMEMORY
```

```
; Чекаємо натиснення будь-якої клавіші
mov ah, 10h
int 16h
```

```
; Побудова прямокутника
; Перемкнемося в графічний режим
mov ah, 00h ; Функція встановлення відеорежиму
mov al, Mode_13
int 10h
; int 10h
push 0A000h ; Сегментна адреса відеобуфера
pop es ; ES = A000h (відеосегмент)
```

```
call DRAW_RECTANGLE
```

```
; Чекаємо натиснення будь-якої клавіші
mov ah, 10h
int 16h
```

```
; Відновимо попередній відеорежим
mov ah, 0 ; Функція встановлення відеорежиму
```



```
mov al, saveMode      ; Номер збереженого відеорежиму
int 10h
exit
```

main ENDP

WRITE\_TO\_FIRST\_PAGE\_OF\_TEXTMEMORY PROC

```
LOOP_FIRST_PAGE_OF_TEXTMEMORY:
    lodsb                ; завантажує si до al
    mov es:[di], al
    inc di
    loop LOOP_FIRST_PAGE_OF_TEXTMEMORY
ret
```

WRITE\_TO\_FIRST\_PAGE\_OF\_TEXTMEMORY ENDP

DRAW\_RECTANGLE PROC

```
    ; Встановлює колір окремого елемента палітри кольорів i
    ; креслить на екрані кілька пікселів
    ; Змінимо колір елемента палітри, що визначається
    індексом 1
```

```
    mov dx, 3c8h        ; Порт індексу палітри кольорів
    (3C8h)
```

```
    mov al, 1           ; Встановимо індекс 1
    out dx, al
```

```
    mov dx, 3c9h        ; Значення кольорів виводяться у
    порт 3C9h
```

```
    mov al, 2           ; Синій колір
    out dx, al
```

```
    ; Обчислимо зміщення першого пікселя у відеобуфері.
    ; Воно характерне для поточного відеорежиму 13h,
    роздільна здатність якого
    ; складає 320x200.
```

```
    mov xVal, (sizeof myMiddleName)/2
```

```
    mov yVal, (sizeof mySurname)/2
```

```
    mov ax, 320         ; Кількість пікселів у рядку
```

```
    mov bx, 0
```

```
    mul yVal            ; множимо на координату Y,
```

```
    add ax, xVal        ; і додаємо координату X.
```

```
    ; Помістимо значення індексу кольору у відеопам'ять.
```

```
    mov cx, rectangleWidth
```

```
    imul cx, rectangleHeight
```

```
    mov di, ax          ; в AX – зміщення відеобуфера
```

```
    ; Рисуємо пряму завдовжки 10 пікселів.
```

```
DPI:
```

```
    mov BYTE PTR es:[di], 1 ; Записуємо індекс кольору
    ; В регістрі ES зберігається сегментна адреса
    відеобуфера.
```

```
    add di, 1
    add bx, 1
    .IF bx == rectangleWidth
        mov bx, 0
        add di, 320
        sub di, rectangleWidth
    .ENDIF
Loop DPI
ret
DRAW_RECTANGLE ENDP

END main
```

---

### Висновки

Я опанував функції BIOS для роботи з відео в текстовому та графічному режимах; розвинув навички складання програм для виведення різнокольорових рядків символів та графічних зображень