**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

Солтисюк Д.А.

**Прийняв:**

доц. каф. ПЗ

Крук О. Г.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2022р.

∑= \_\_\_\_\_

Львів – 2022

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 навики 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 ХР.

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, В = 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. Сформуйте 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: 22

**Хід роботи**

Результати роботи програми:pasted-image.tiff

pasted-image.tiff

**Код програми**

*; vim: ft=masm*

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

Mode\_03 = 03h *80x25, 16 â€“úâ€“Ã¦â€“Âªâ€”Ã¥â€“Ã¦â€”Àâ€”Öâ€“ò, 8 â€”Áâ€”Ã‡â€“Ã¦â€”Àâ€”Öâ€“Î©â€“Ã¦â€“ú*

Mode\_06 = 06h *; 640 â€”Ã– 200, 2*

Mode\_0D = 0Dh *; 320 â€”Ã– 200, 16 â€“úâ€“Ã¦â€“Âªâ€”Ã¥â€“Ã¦â€”Àâ€”Öâ€“ò*

Mode\_0E = 0Eh *; 640 â€”Ã– 200, 16 â€“úâ€“Ã¦â€“Âªâ€”Ã¥â€“Ã¦â€”Àâ€”Öâ€“ò*

Mode\_0F = 0Fh *; 640 â€”Ã– 350, 2*

Mode\_10 = 10h *; 640 â€”Ã– 350, 16 â€“úâ€“Ã¦â€“Âªâ€”Ã¥â€“Ã¦â€”Àâ€”Öâ€“ò*

Mode\_11 = 11h *; 640 â€”Ã– 480, 2*

Mode\_12 = 12h *; 640 â€”Ã– 480, 16 â€“úâ€“Ã¦â€“Âªâ€”Ã¥â€“Ã¦â€”Àâ€”Öâ€“ò*

Mode\_13 = 13h *; 320 â€”Ã– 200, 256 â€“úâ€“Ã¦â€“Âªâ€”Ã¥â€“Ã¦â€”Àâ€”Öâ€“ò*

Mode\_6A = 6Ah *; 800 â€”Ã– 600, 16 â€“úâ€“Ã¦â€“Âªâ€”Ã¥â€“Ã¦â€”Àâ€”Öâ€“ò*

.data

saveMode BYTE ?

mySurname BYTE 'S', 1Ch, 'o', 5Dh, 'l', 1Fh, 't', 5Ah, 'y', 1Ah, 's', 4Ah, 'i', 7Ch, 'u', 3Fh, 'k', 3Ch

myName BYTE 'D', 0Eh, 'm', 4Bh, 'y', 3Ah, 't', 3Dh, 'r', 7Eh, 'o', 1Fh

myMiddleName BYTE 'A', 3Bh, 'n', 6Eh, 'd', 5Ah, 'r', 6Eh, 'i', 3Ch, 'i', 6Dh, 'o', 2Bh, 'v', 7Eh, 'y', 3Ah, 'c', 2Ah, 'h', 7Dh

*; variant = 22*

row WORD 2*; variant % 10*

column WORD 44*; variant + group number*

xVal WORD ?

yVal WORD ?

rectangleWidth WORD 132*; 3 \* (group number + variant) = 3 \* (22 + 22) = 3 \* 44 = 132*

rectangleHeight WORD 74*; 2 \* (group number + variant) = 2 \* (22 + 22) = 2 \* 44 = 88*

.code

main **PROC**

**mov** ax, @data

**mov** ds, ax

*; â€“Ã³â€“Â±â€“Âµâ€”Àâ€“Âµâ€“öâ€“Âµâ€“üâ€“Ã¦ â€“Î©â€“Ã¦â€“üâ€“Âµâ€”À â€“Ã¸â€“Ã¦â€”Ã‡â€“Ã¦â€”Ã¡â€“Î©â€“Ã¦â€“óâ€“Ã¦ â€“òâ€”Öâ€“Â*

**mov** ah, 0Fh*; tells "int 10h" to return current video mode*

**int** 10h

**mov** saveMode, al

*; â€“Òâ€”Áâ€”Ã‡â€“âˆžâ€“Î©â€“Ã¦â€“òâ€“Âªâ€“Âµâ€“Î©â€“Î©â€”Ã¨ â€”Ã‡â€“Âµâ€“úâ€”Áâ€”Ã‡â€“Ã¦â€“òâ€“Ã¦â€“óâ€“Ã¦ â€”Àâ€“Âµâ€“öâ€“âˆ*

**mov** ax, Mode\_03

**int** 10h

**mov** ah, 05h

**mov** al, 1

**int** 10h

*”Ã‡â€“Âµâ€“úâ€”Áâ€”Ã‡â€“Ã¦â€“òâ€“Ã¦â€”Ã³ â€“Ã¸â€“âˆžâ€“ü'â€”Ã¨â€”Ã‡â€”Ö*

**mov** ax, 0B900h

**mov** es, ax

**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

**mov** cx, SIZEOF myName

**mov** si, OFFSET myName

**xor** di, di

**mov** bx, row

**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

**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

*”Ã³ â€“úâ€“Âªâ€“âˆžâ€“òâ€”Öâ€”Èâ€”Ö*

**mov** ah, 10h

**int** 16h

*€“ü*

**mov** ah, 00h*; â€“äâ€”Ã‰â€“Î©â€“úâ€”Ãœâ€”Öâ€”Ã¨ â€“òâ€”Áâ€”Ã‡â€“âˆžâ€“Î©â€“Ã¦â€“òâ€“Âªâ€“Âµâ€“Î©â€“Î©â€”Ã¨ â€“òâ€”Öâ€“Â¥â€“Âµâ€“Ã¦â€”Àâ€“Âµâ€*

**mov** al, Mode\_13

**int** 10h

*; int 10h*

**push** 0A000h*; â€“áâ€“Âµâ€“óâ€“üâ€“Âµâ€“Î©â€”Ã‡â€“Î©â€“âˆž â€“âˆžâ€“Â¥â€”Àâ€“Âµâ€”Áâ€“âˆž â€“òâ€”Öâ€“Â¥â€“Âµâ€“Ã¦â€“Â±â€”Ã‰â€”Äâ€“Âµâ€”Àâ€“âˆž*

**pop** es*; ES = A000h (â€“òâ€”Öâ€“Â¥â€“Âµâ€“Ã¦â€”Áâ€“Âµâ€“óâ€“üâ€“Âµâ€“Î©â€”Ã‡)*

**call** DRAW\_RECTANGLE

*”Ã³ â€“úâ€“Âªâ€“âˆžâ€“òâ€”Öâ€”Èâ€”Ö*

**mov** ah, 10h

**int** 16h

**mov** ah, 0*; â€“äâ€”Ã‰â€“Î©â€“úâ€”Ãœâ€”Öâ€”Ã¨ â€“òâ€”Áâ€”Ã‡â€“âˆžâ€“Î©â€“Ã¦â€“òâ€“Âªâ€“Âµâ€“Î©â€“Î©â€”Ã¨ â€“òâ€”Öâ€“Â¥â€“Âµâ€“Ã¦â€”Àâ€“Âµâ€*

**mov** al, saveMode*”Ã‰*

**int** 10h

exit

main **ENDP**

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

LOOP\_FIRST\_PAGE\_OF\_TEXTMEMORY:

**lodsb** *; â€“âˆ‘â€“âˆžâ€“òâ€“âˆžâ€“Î©â€”Ã‡â€“âˆžâ€“öâ€”Ã‰â€”Ã® si â€“Â¥â€“Ã¦ al*

**mov** es:[di], al

**inc** di

**loop** LOOP\_FIRST\_PAGE\_OF\_TEXTMEMORY

**ret**

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

DRAW\_RECTANGLE **PROC**

*; â€“Òâ€”Áâ€”Ã‡â€“âˆžâ€“Î©â€“Ã¦â€“òâ€“Âªâ€”Ã©â€”Ã® â€“úâ€“Ã¦â€“Âªâ€”Öâ€”À â€“Ã¦â€“úâ€”Àâ€“Âµâ€“üâ€“Ã¦â€“óâ€“Ã¦ â€“Âµâ€“Âªâ€“Â*

*€”Áâ€“Âµâ€“Âªâ€”Öâ€“ò*

*1*

**mov** dx, 3c8h*3C8h)*

**mov** al, 1*1*

**out** dx, al

**mov** dx, 3c9h*¨ â€”Ã‰ â€“Ã¸â€“Ã¦â€”Àâ€”Ã‡ 3C9h*

**mov** al, 1

**out** dx, al

*â€”Öâ€“úâ€”Áâ€“Âµâ€“Âªâ€”Ã¨ â€”Ã‰ â€“òâ€”Öâ€“Â¥â€“Âµâ€“Ã¦â€“Â±â€”Ã‰â€”Äâ€“Âµâ€”Àâ€”Ö.*

*; â€“Òâ€“Ã¦â€“Î©â€“Ã¦ â€”Ã–â€“âˆžâ€”Àâ€“âˆžâ€“úâ€”Ã‡â€“Âµâ€”Àâ€“Î©â€“Âµ â€“Â¥â€“Âªâ€”Ã¨ â€“Ã¸â€“Ã¦â€”Ã‡â€“Ã¦â€”Ã¡â€“Î©â€“Ã¦â€13h, â€”Àâ€“Ã¦â€“âˆ‘â€“Â¥â€”Öâ€“Âªâ€”Ã¥â€“Î©â€“âˆž â€“âˆ‘â€“Â¥â€“âˆžâ€”Ã‡â€“Î©â€”Öâ€”Áâ€”Ã‡â€”Ã¥ â€”Ã¨â€“úâ€“Ã¦â€“óâ€“Ã¦*

*; â€”Áâ€“úâ€“Âªâ€“âˆžâ€“Â¥â€“âˆžâ€”Ã® 320â€”Ã–200.*

**mov** xVal, (SIZEOF myMiddleName)/2

**mov** yVal, (SIZEOF mySurname)/2

**mov** ax, 320*; â€“Úâ€”Öâ€“Âªâ€”Ã¥â€“úâ€”Öâ€”Áâ€”Ã‡â€”Ã¥ â€“Ã¸â€”Öâ€“úâ€”Áâ€“Âµâ€“Âªâ€”Öâ€“ò â€”Ã‰ â€”Àâ€”Ã¨â€“Â¥â€“úâ€”Ã‰*

**mov** bx, 0

**mul** yVal

**add** ax, xVAl

*€“Ã¦â€“Âªâ€”Ã¥â€“Ã¦â€”Àâ€”Ã‰ â€”Ã‰ â€“òâ€”Öâ€“Â¥â€“Âµâ€“Ã¦â€“Ã¸â€“âˆžâ€“ü'â€”Ã¨â€”Ã‡â€”Ã¥.*

**mov** cx, rectangleWidth

imul cx, rectangleHeight

**mov** di, ax*; â€“ò â€“Ãªâ€“â€¢ â€šÀÓ â€“âˆ‘â€“üâ€”Öâ€”Éâ€“Âµâ€“Î©â€“Î©â€”Ã¨ â€“òâ€”Öâ€“Â¥â€“Âµâ€“Ã¦â€“Â±â€”Ã‰â€”Äâ€“Âµâ€”Àâ€“âˆž*

*10 â€“Ã¸â€”Öâ€“úâ€”Áâ€“Âµâ€“Âªâ€”Öâ€“ò.*

DPI:

**mov** BYTE PTR es:[di], 1

*; â€“Ò â€”Àâ€“Âµâ€“óâ€”Öâ€”Áâ€”Ã‡â€”Àâ€”Ö ES â€“âˆ‘â€“Â±â€“Âµâ€”Àâ€”Öâ€“óâ€“âˆžâ€”Ã®â€”Ã‡â€”Ã¥â€”Áâ€”Ã¨ â€”Áâ€“Âµâ€“óâ€“üâ€“Âµâ€“Î©â€”Ã‡â€“Î©â€“âˆž â€“âˆžâ€“Â¥â€”Àâ€“Âµâ€”Áâ€“âˆž â€“òâ€”Öâ€“Â¥â€“Âµâ€“Ã¦â€“Â±â€”Ã‰â€”Äâ€“Âµâ€”Àâ€“âˆž.*

**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**

На даній лабораторній роботі я опанував функції BIOS для роботи з відео в текстовому та графічному режимах; розвинув навики складання програм для виведення різнокольорових рядків символів та графічних зображень.