

Wojskowa Akademia Techniczna im. Jarosława Dąbrowskiego

Laboratorium z przedmiotu:
Interfejsy komputerów cyfrowych

Sprawozdanie z ćwiczenia laboratoryjnego nr 5: Programowanie adaptera VGA z poziomu BIOS

Prowadzący:
mgr inż. Krzysztof Szajewski

Wykonał: Radosław Relidzyński

Grupa: WCY20IY4S1

Data laboratoriów: 13.06.2021 r.

Spis treści

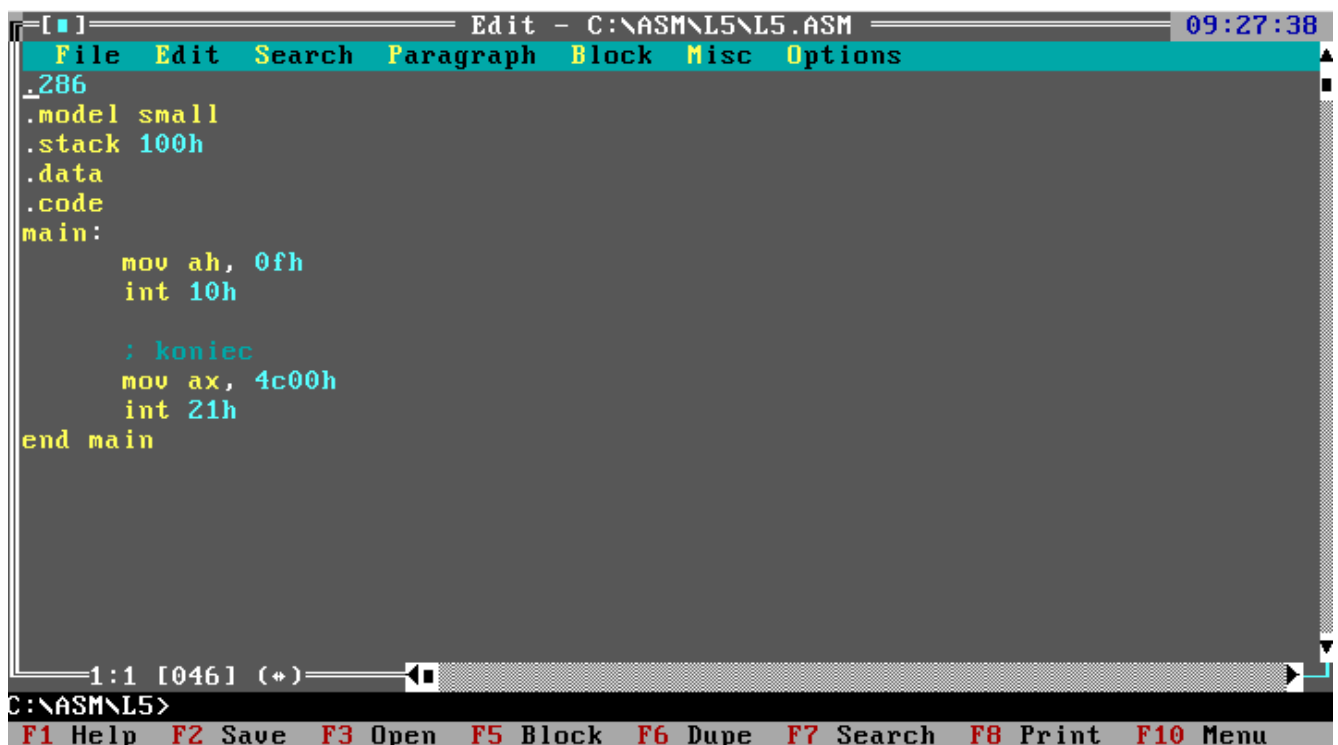
A.	Treść zadania	2
B.	Zadanie 1 odczyt trybu graficznego.....	2
C.	Zadanie 2 ustawienie trybu 80x25x16.....	3
D.	Zadanie 3 ustawienie kształtu kursora	4
E.	Zadanie 4 ustawienie pozycji kursora w 12 linii i 40 kolumnie	5
F.	Zadanie 5 odczytanie pozycji kursora.....	6
G.	Zadanie 6 wyświetlenie znaku w pozycji kursora	8
H.	Zadanie 7 odczytanie znaku na danej pozycji kursora	9
I.	Zadanie 8 ustawienie trybu graficznego 320x200x256	10
J.	Zadanie 9 wyświetlenie białego piksela w 100 linii i 160 kolumnie (funkcja 0CH)	12
K.	Zadanie 10 wyświetlenie białego piksela w 100 linii i 160 kolumnie (modyfikacja określonego bajtu segmentu 0A000H).....	13
L.	Zadanie 11 odczytanie kolor piksela w 100 linii i 160 kolumnie	15

A.Treść zadania

Napisać następujące programy realizujące następujące funkcje:

1. odczyt aktualnego trybu graficznego
2. ustawienie trybu graficznego 80x25x16
3. ustawienie kształtu kursora (wielkość 4 linie od dołu, wielkość 4 linie od góry, wielkość 2 linie w środku)
4. ustawienie pozycji kursora w 12 linii i 40 kolumnie
5. odczytanie pozycji kursora
6. wyświetlenie znaku w pozycji kursora
7. odczytanie znaku na danej pozycji kursora
8. ustawienie trybu graficznego 320x200x256
9. wyświetlenie białego piksela w 100 linii i 160 kolumnie (funkcja 0CH)
10. wyświetlenie białego piksela w 100 linii i 160 kolumnie (modyfikacja określonego bajtu segmentu 0A000H)
11. odczytanie kolor piksela w 100 linii i 160 kolumnie
12. narysowanie kwadratu o boku 50 pikseli (kontur i wypełniony) – dla chętnych

B.Zadanie 1 odczyt trybu graficznego



```
[■] Edit - C:\ASM\L5\L5.ASM 09:27:38
File Edit Search Paragraph Block Misc Options
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 0fh
    int 10h

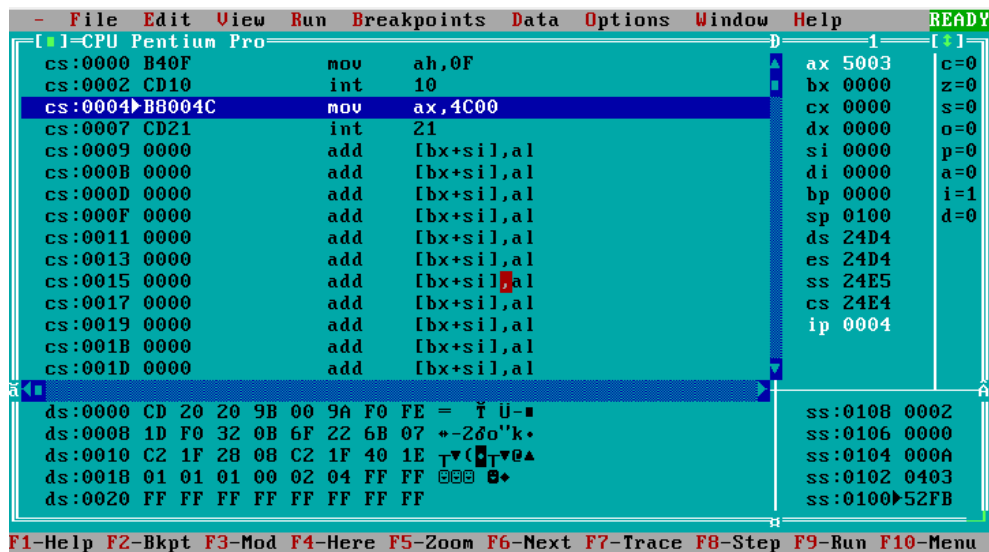
    ; koniec
    mov ax, 4c00h
    int 21h
end main

1:1 [046] (*)
C:\ASM\L5>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu
```

MOV ah, 0fh – pobranie numeru trybu pracy sterownika

INT 10h – wywołanie przerwania w celu wykonania polecenia

Odczytany tryb:



The screenshot shows a debugger window with the following content:

Address	Disassembly	Comment	Register/Value
cs:0000 B40F	mov ah,0F		ax 5003
cs:0002 CD10	int 10		bx 0000
cs:0004 B8004C	mov ax,4C00		cx 0000
cs:0007 CD21	int 21		dx 0000
cs:0009 0000	add [bx+si],al		si 0000
cs:000B 0000	add [bx+si],al		di 0000
cs:000D 0000	add [bx+si],al		bp 0000
cs:000F 0000	add [bx+si],al		sp 0100
cs:0011 0000	add [bx+si],al		ds 24D4
cs:0013 0000	add [bx+si],al		es 24D4
cs:0015 0000	add [bx+si],al		ss 24E5
cs:0017 0000	add [bx+si],al		cs 24E4
cs:0019 0000	add [bx+si],al		ip 0004
cs:001B 0000	add [bx+si],al		
cs:001D 0000	add [bx+si],al		

Registers (right side):

c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0

Memory dump (bottom left):

ds:0000 CD 20 20 9B 00 9A F0 FE = ħ Ů-■
ds:0008 1D F0 32 0B 6F 22 6B 07 +-2đo''k•
ds:0010 C2 1F 28 08 C2 1F 40 1E T▼(T▼@▲
ds:0018 01 01 01 00 02 04 FF FF 000 0•
ds:0020 FF FF FF FF FF FF FF FF

Memory dump (bottom right):

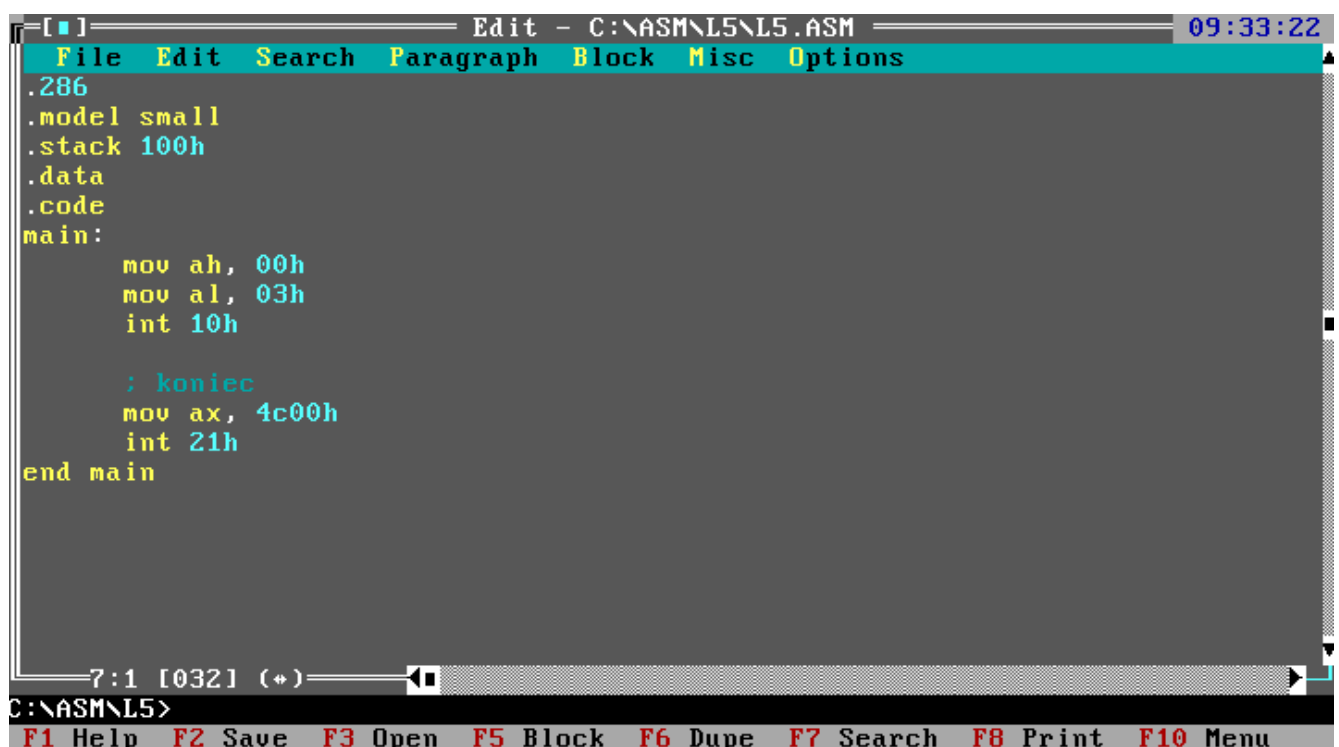
ss:0108 0002
ss:0106 0000
ss:0104 000A
ss:0102 0403
ss:0100 52FB

Footer: F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

AL = 03h, tryb 80x2516 - tekstowy

C. Zadanie 2 ustawienie trybu 80x25x16

Tryb 80x25x16, czyli AL = 03h



The screenshot shows an assembly editor window with the following content:

```
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 00h
    mov al, 03h
    int 10h

    ; koniec
    mov ax, 4c00h
    int 21h
end main
```


Footer: F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu

MOV ah, 00h – ustawienie trybu pracy sterownika karty graficznej

MOV al, 03h – ustawienie konkretnego trybu pracy sterownika karty graficznej

INT 10h – wywołanie przerwania w celu wykonania polecenia

D. Zadanie 3 ustawienie kształtu kursora



```

[ ] Edit - C:\ASM\L5\L5.ASM 09:44:03
File Edit Search Paragraph Block Misc Options
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 01h
    mov ch, 03h
    mov cl, 04h
    int 10h

    ; koniec
    mov ax, 4c00h
    int 21h
end main
1:1 [046] (*)
C:\ASM\L5>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu

```

MOV ah, 01h – ustawienie postaci kursora

MOV ch, 03h – ustawienie numeru górnej linii wiersza

MOV cl, 04h – ustawienie numeru dolnej linii wiersza

INT 10h – wywołanie przerwania w celu wykonania polecenia

Wykonanie programu:

```

C:\ASM\L5>dn
Dos Navigator Version 1.51 Copyright (C) 1991-99 RIT Research Labs

C:\ASM\L5>comp 15

C:\ASM\L5>tasm 15.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: 15.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 352k

C:\ASM\L5>tlink /v 15.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

C:\ASM\L5>

C:\ASM\L5>15

C:\ASM\L5>15.exe

C:\ASM\L5>—

```

E. Zadanie 4 ustawienie pozycji kursora w 12 linii i 40 kolumnie

```

Edit - C:\ASM\L5\L5.ASM 09:55:55
File Edit Search Paragraph Block Misc Options
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 02h
    mov bh, 00h
    mov dh, 0Ch
    mov dl, 28h
    int 10h

    ; koniec
    mov ax, 4c00h
    int 21h
end main

1:1 [046] (↔)
C:\ASM\L5>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu

```

MOV ah, 02h – ustawienie pozycji kursora na ekranie

MOV bh, 00h – określenie numeru strony

MOV dh, 0Ch – ustawienie pozycji kursora, wiersz 12

MOV dl, 28h – ustawienie pozycji kursora, kolumna 40

INT 10h – wywołanie przerwania w celu wykonania polecenia

Wykonanie programu:

```
C:\ASM\L5>15.exe

C:\ASM\L5>dn
Dos Navigator Version 1.51 Copyright (C) 1991-99 RIT Research Labs

C:\ASM\L5>comp 15

C:\ASM\L5>tasm 15.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International


Assembling file: 15.asm
Error messages: None
C:\ASM\L5>ssages: None
Passes: 1
Remaining memory: 352k

C:\ASM\L5>tlink /v 15.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

C:\ASM\L5>

C:\ASM\L5>15
```

F. Zadanie 5 odczytanie pozycji kursora



```
[ ] Edit - C:\ASM\L5\L5.ASM 10:03:22
File Edit Search Paragraph Block Misc Options
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 03h
    mov bh, 00h
    int 10h

    ; koniec
    mov ax, 4c00h
    int 21h
end main
6:1 [109] (*)
C:\ASM\L5>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu
```

MOV ah, 03h – pobranie pozycji kursora

MOV bh, 00h – określenie numeru strony

INT 10h – wywołanie przerwania w celu wykonania polecenia

```
- File Edit View Run Breakpoints Data Options Window Help READY
[ ] CPU Pentium Pro
cs:0000 B403      mov     ah,03
cs:0002 B700      mov     bh,00
cs:0004 CD10      int      10
cs:0006 B8004C    mov     ax,4C00
cs:0009 CD21      int      21
cs:000B 0000      add     [bx+si],al
cs:000D 0000      add     [bx+si],al
cs:000F 0000      add     [bx+si],al
cs:0011 0000      add     [bx+si],al
cs:0013 0000      add     [bx+si],al
cs:0015 0000      add     [bx+si],al
cs:0017 0000      add     [bx+si],al
cs:0019 0000      add     [bx+si],al
cs:001B 0000      add     [bx+si],al
cs:001D 0000      add     [bx+si],al

ds:0000 CD 20 20 9B 00 9A F0 FE =  T Ü-
ds:0008 1D F0 32 0B 6F 22 6B 07 +-28o" k.
ds:0010 C2 1F 28 08 C2 1F 40 1E T ( T e
ds:0018 01 01 01 00 02 04 FF FF 000 0+
ds:0020 FF FF FF FF FF FF FF FF

ax 0300  c=0
bx 0000  z=0
cx 0607  s=0
dx 1800  o=0
si 0000  p=0
di 0000  a=0
bp 0000  i=1
sp 0100  d=0
ds 24D4
es 24D4
ss 24E5
cs 24E4
ip 0006

ss:0108 0002
ss:0106 0000
ss:0104 000A
ss:0102 0403
ss:0100 52FB

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```

Bieżący wiersz: 18h, wiersz 24

Bieżąca kolumna: 00h, kolumna 0

G. Zadanie 6 wyświetlenie znaku w pozycji kursora



```

[ ] Edit - C:\ASM\L5\L5.ASM 10:11:10
File Edit Search Paragraph Block Misc Options
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 0ah
    mov al, 58h
    mov bh, 00h
    mov cx, 0001h
    int 10h

    ; koniec
    mov ax, 4c00h
    int 21h
end main

4:1 [046] (+)
C:\ASM\L5>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu

```

MOV ah, 0ah – wypisanie znaku na pozycji wskazywanej przez kursor

MOV al, 58h – kod ASCII znaku ('X')

MOV bh, 00h – określenie numeru strony

MOV cx, 0001h – określenie ilości kopii znaku (1)

INT 10h – wywołanie przerwania w celu wykonania polecenia

Wykonanie programu:


```

C:\ASM\L5>15
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C:\ASM\L5>dnXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Dos Navigator Version 1.51 Copyright (C) 1991-99 RIT Research LabsXXXXXXXXXXXX

C:\ASM\L5>comp 15

C:\ASM\L5>tasm 15.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: 15.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 352k

C:\ASM\L5>tlink /v 15.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

C:\ASM\L5>

C:\ASM\L5>15
X
C:\ASM\L5>_

```

H. Zadanie 7 odczytanie znaku na danej pozycji kursora

```

Edit - C:\ASM\L5\L5.ASM 10:13:59
File Edit Search Paragraph Block Misc Options
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 08h
    mov bh, 00h_
    int 10h

    ; koniec
    mov ax, 4c00h
    int 21h
end main
* 8:18 [000] (+)
C:\ASM\L5>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu

```

MOV ah, 08h – pobranie kodu i atrybutu znaku na pozycji wskazywanej przez kursor

MOV bh, 00h – określenie numeru strony

INT 10h – wywołanie przerwania w celu wykonania polecenia

```

- File Edit View Run Breakpoints Data Options Window Help READY
[ ] CPU Pentium Pro
cs:0000 B408 mov ah,08 ax 0720 c=0
cs:0002 B700 mov bh,00 bx 0000 z=0
cs:0004 CD10 int 10 cx 0000 s=0
cs:0006 B8004C mov ax,4C00 dx 0000 o=0
cs:0009 CD21 int 21 si 0000 p=0
cs:000B 0000 add [bx+si],al di 0000 a=0
cs:000D 0000 add [bx+si],al bp 0000 i=1
cs:000F 0000 add [bx+si],al sp 0100 d=0
cs:0011 0000 add [bx+si],al ds 24D4
cs:0013 0000 add [bx+si],al es 24D4
cs:0015 0000 add [bx+si],al ss 24E5
cs:0017 0000 add [bx+si],al cs 24E4
cs:0019 0000 add [bx+si],al ip 0006
cs:001B 0000 add [bx+si],al
cs:001D 0000 add [bx+si],al

ds:0000 CD 20 20 9B 00 9A F0 FE = T U-
ds:0008 1D F0 32 0B 6F 22 6B 07 *-2do''k*
ds:0010 C2 1F 28 08 C2 1F 40 1E T ( T e
ds:0018 01 01 01 00 02 04 FF FF 000 0
ds:0020 FF FF FF FF FF FF FF FF

ss:0108 0002
ss:0106 0000
ss:0104 000A
ss:0102 0403
ss:0100 52FB

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```

AH = 07h – atrybut znaku

AL = 20h – kod znaku (spacja)

I. Zadanie 8 ustawienie trybu graficznego 320x200x256

```

[ ] Edit - C:\ASM\L5\L5.ASM 10:19:04
File Edit Search Paragraph Block Misc Options
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 00h
    mov al, 13h
    int 10h

    ; koniec
    mov ax, 4c00h
    int 21h
end main

1:1 [046] (+)
C:\ASM\L5>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu

```

MOV ah, 00h – ustawienie trybu pracy sterownika karty graficznej

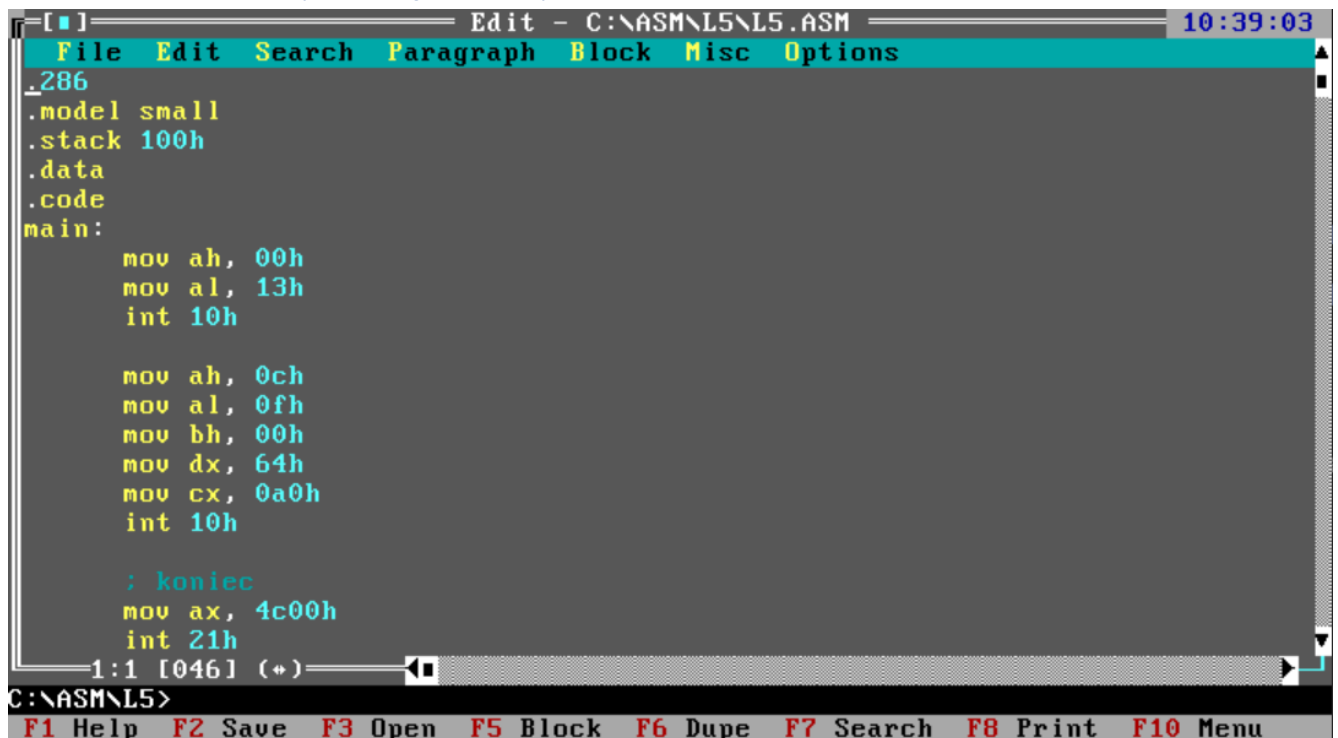
MOV al, 13h – ustawienie konkretnego trybu pracy sterownika karty graficznej

INT 10h – wywołanie przerwania w celu wykonania polecenia

Wykonanie programu:

```
C:\ASM\L5>
```

J. Zadanie 9 wyświetlenie białego piksela w 100 linii i 160 kolumnie (funkcja 0CH)



The screenshot shows a DOS-based assembly editor window titled "Edit - C:\ASM\L5\L5.ASM" with a timestamp of 10:39:03. The menu bar includes File, Edit, Search, Paragraph, Block, Misc, and Options. The code is as follows:

```
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 00h
    mov al, 13h
    int 10h

    mov ah, 0ch
    mov al, 0fh
    mov bh, 00h
    mov dx, 64h
    mov cx, 0a0h
    int 10h

    : koniec
    mov ax, 4c00h
    int 21h
```

The status bar at the bottom shows "1:1 [046] (↔)" and a function key menu: F1 Help, F2 Save, F3 Open, F5 Block, F6 Dupe, F7 Search, F8 Print, F10 Menu.

MOV ah, 0ch – ustawienie trybu pracy sterownika karty graficznej

MOV al, 0fh – ustawienie koloru punktu

MOV bh, 00h – ustawienie numeru strony

MOV dx, 64h – ustawienie numeru linii (100)

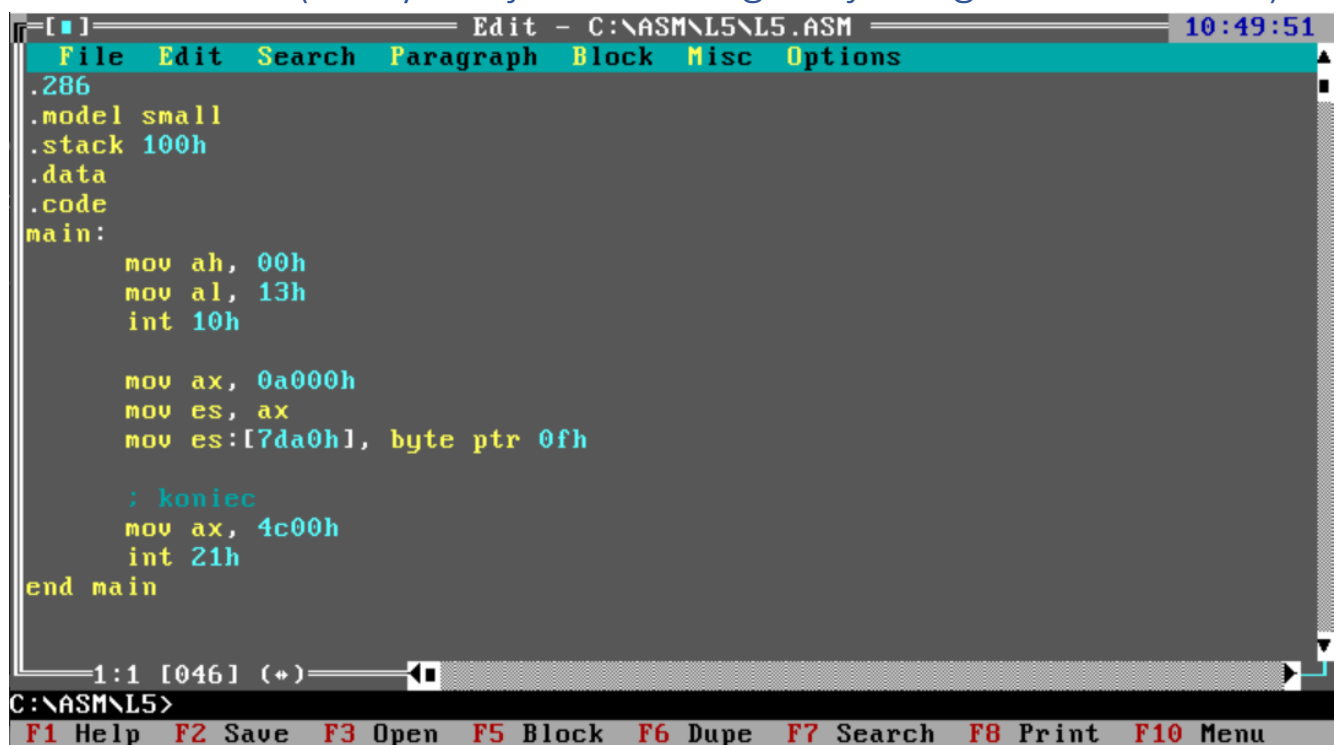
MOV cx, 0a0h – ustawienie numeru kolumny (160)

INT 10h – wywołanie przerwania w celu wykonania polecenia

Wykonanie programu:

C:\ASM\L5>

K. Zadanie 10 wyświetlenie białego piksela w 100 linii i 160 kolumnie (modyfikacja określonego bajtu segmentu 0A000H)



The screenshot shows a text editor window titled "Edit - C:\ASM\L5\L5.ASM" with a timestamp of 10:49:51. The menu bar includes File, Edit, Search, Paragraph, Block, Misc, and Options. The code is as follows:

```
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 00h
    mov al, 13h
    int 10h

    mov ax, 0a000h
    mov es, ax
    mov es:[7da0h], byte ptr 0fh

    ; koniec
    mov ax, 4c00h
    int 21h
end main
```

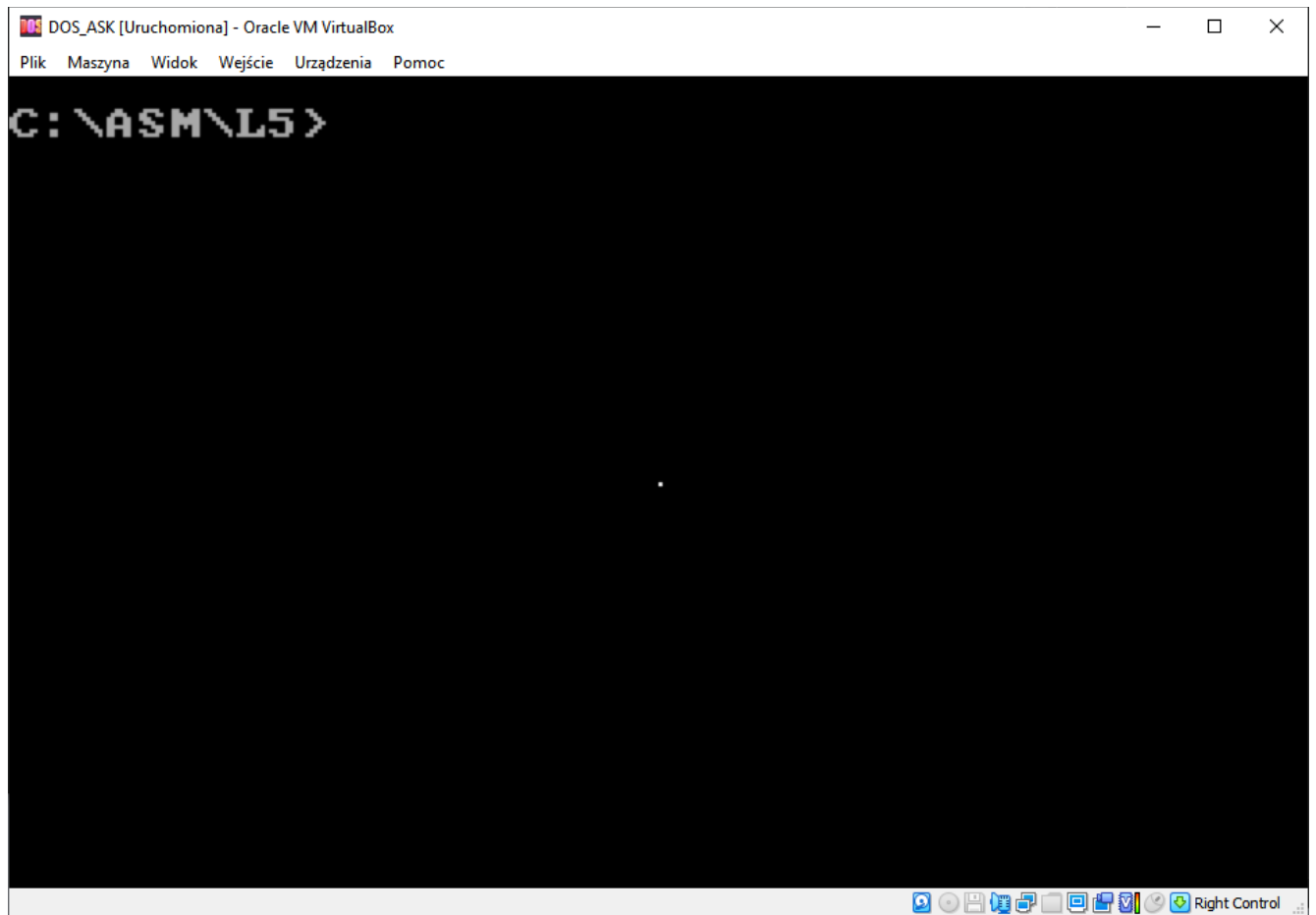
The status bar at the bottom shows "1:1 [046] (+)" and a keyboard shortcut menu: F1 Help, F2 Save, F3 Open, F5 Block, F6 Dupe, F7 Search, F8 Print, F10 Menu.

MOV ax, 0a000h – pobranie bajtu segmentu

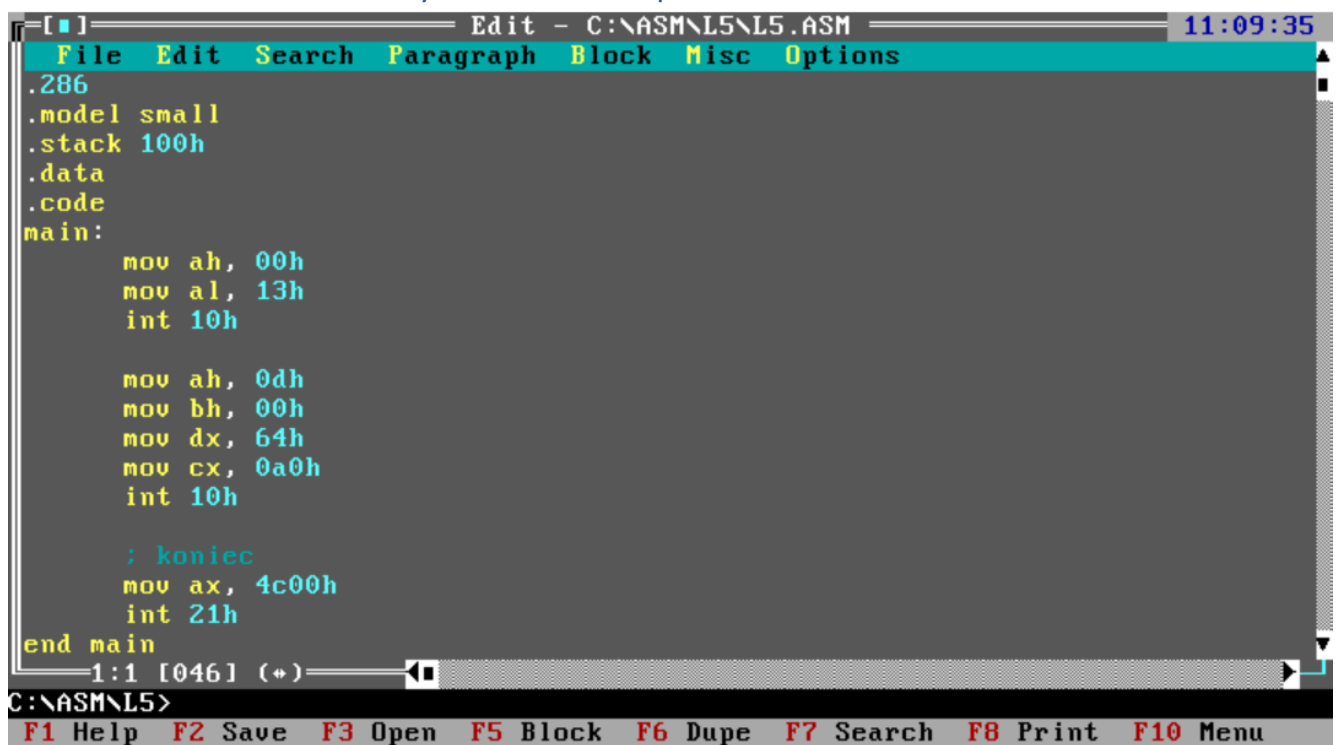
MOV es, ax – przekazanie wartości

MOV es:[7da0h], byte ptr 0fh – wyświetlenie piksela, 7da0h -> położenie (100, 160), kolor biały

Wykonanie programu:



L. Zadanie 11 odczytanie kolor piksela w 100 linii i 160 kolumnie



```

[ ] Edit - C:\ASM\L5\L5.ASM 11:09:35
File Edit Search Paragraph Block Misc Options
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 00h
    mov al, 13h
    int 10h

    mov ah, 0dh
    mov bh, 00h
    mov dx, 64h
    mov cx, 0a0h
    int 10h

    : koniec
    mov ax, 4c00h
    int 21h
end main
1:1 [046] (+)
C:\ASM\L5>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu

```

MOV ah, 0dh – ustawienie trybu pracy sterownika karty graficznej

MOV bh, 00h – ustawienie numeru strony

MOV dx, 64h – ustawienie numeru linii (100)

MOV cx, 0a0h – ustawienie numeru kolumny (160)

INT 10h – wywołanie przerwania w celu wykonania polecenia

```

- File Edit View Run Breakpoints Data Options Window Help READY
[ ]-CPU Pentium Pro D 1 [ ]
cs:0000 B400 mov ah,00 ax 0D00 c=0
cs:0002 B013 mov al,13 bx 0000 z=0
cs:0004 CD10 int 10 cx 00A0 s=0
cs:0006 B40D mov ah,0D dx 0064 o=0
cs:0008 B700 mov bh,00 si 0000 p=0
cs:000A BA6400 mov dx,0064 di 0000 a=0
cs:000D B9A000 mov cx,00A0 bp 0000 i=1
cs:0010 CD10 int 10 sp 0100 d=0
cs:0012 B8004C mov ax,4C00 ds 24D4
cs:0015 CD21 int 21 es 24D4
cs:0017 0000 add [bx+si],al ss 24E6
cs:0019 0000 add [bx+si],al cs 24E4
cs:001B 0000 add [bx+si],al ip 0012
cs:001D 0000 add [bx+si],al
cs:001F 0000 add [bx+si],al
ds:0000 CD 20 20 9B 00 9A F0 FE = T U-
ds:0008 1D F0 32 0B 6F 22 6B 07 +-28o'k+
ds:0010 C2 1F 28 08 C2 1F 40 1E T ( T T
ds:0018 01 01 01 00 02 04 FF FF 000 0+
ds:0020 FF FF FF FF FF FF FF FF ss:0108 0002
ss:0106 0000
ss:0104 000A
ss:0102 0403
ss:0100 52FB
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```

AL = 00h – kolor czarny