Motto:

“Nu calea conteaza, ci modul cum o parcurgem. ” **Fer. V. Ghika**

C9

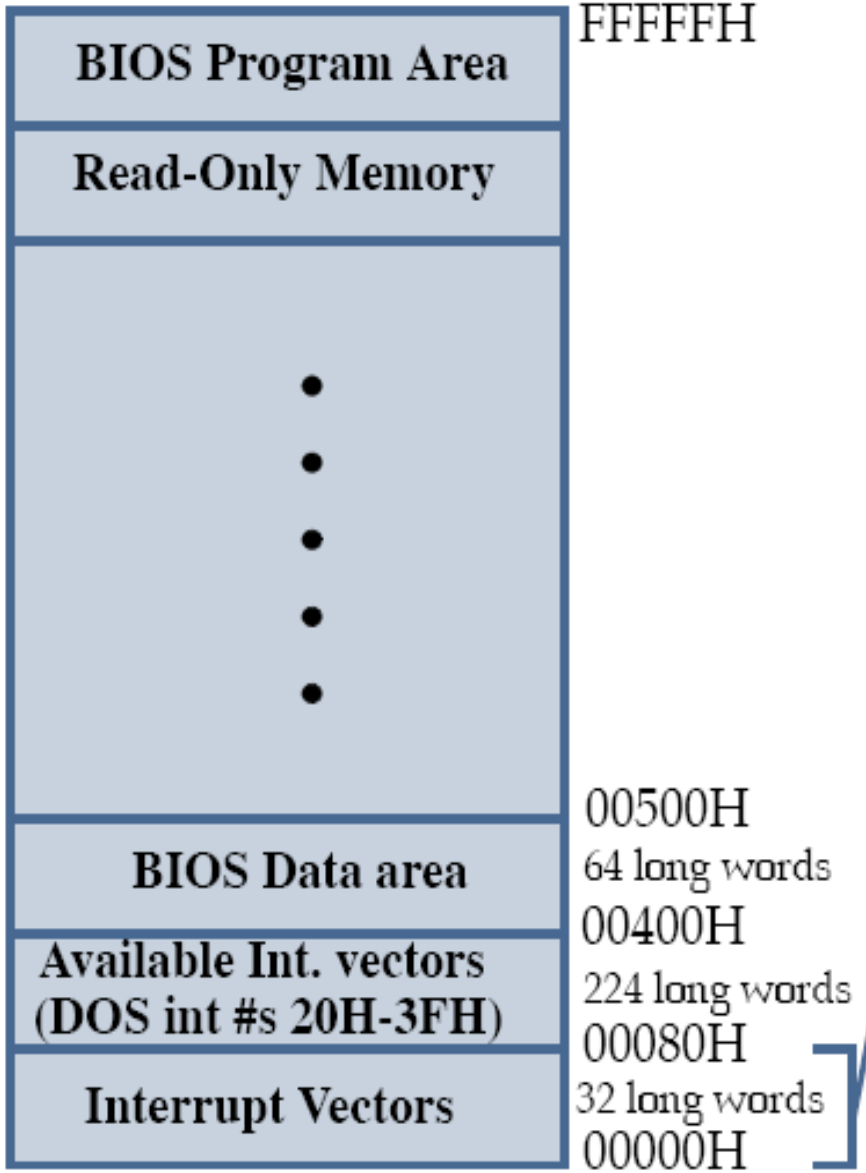
1. Intreruperi BIOS/DOS
2. INT 10h – servicii video
3. INT 16h – servicii tastatura
4. INT 1Ch

Aplicatii

Intreruperi

- Fiecarui eveniment “x” i se asociaza o rutina (procedura) de tratare
- Ptr. o tratare sistematica intreruperile sunt vectorizate , fiecarei intreruperi >> tip (byte)
- In momentul aparitiei intreruperii *procesorul trebuie sa primeasca tipul* ptr. a sti modul in care sa o trateze
- Adresele rutinelor se afla intr-o tabela plasata la inceputul memoriei *(0:0) TVI* – tabela vectorilor de intreruperi
- Adresa rutinei (tip x) se afla la (0: 4*tipx) (CS:IP)
- Pentru asigurarea portabilitatii aplicatiilor si compatibiliatatea software, exista *un set de intreruperi sistem* predefinite care corespund anumitor tipuri >> care executa aceiasi functie, chiar daca adresa din tabela difera
- Prioritatea intreruperilor depinde de tip – intr. Tip 0 este c.m.p.

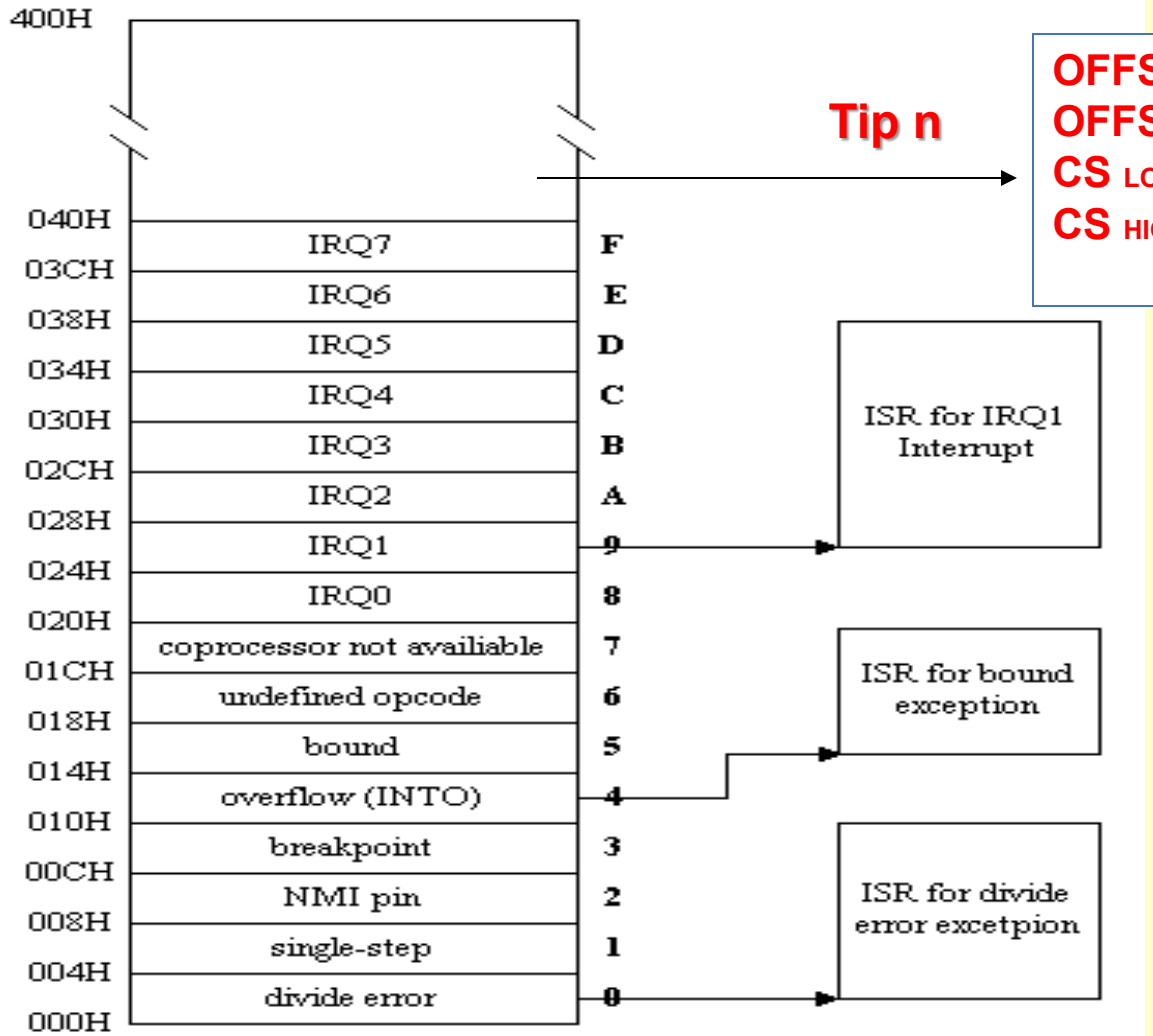
Interrupt Vectors (DOS PC)



TVI

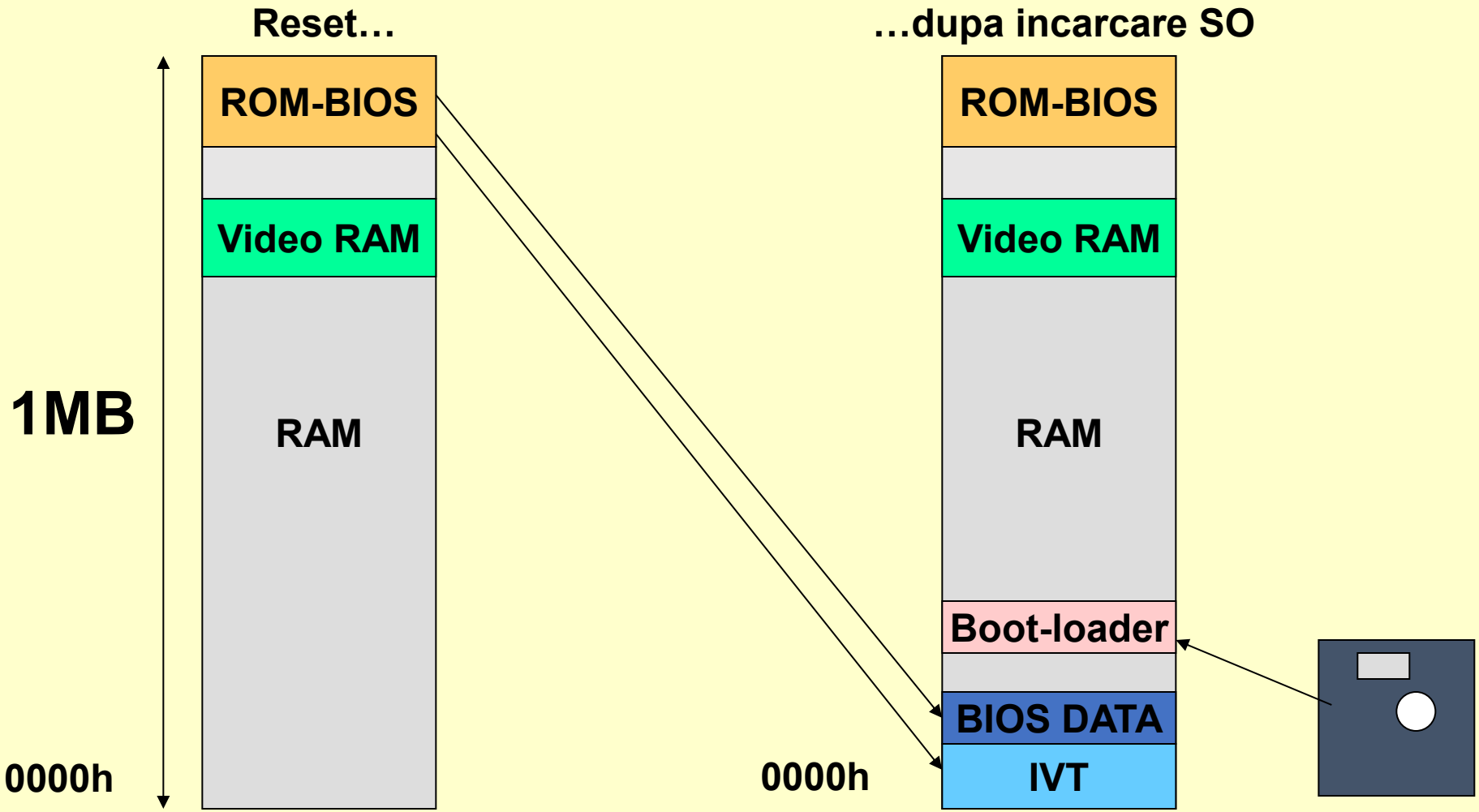
Address			Interrupt #	
7C-7F	Video Graphic Chars		1FH	Pts to Data
78-7B	Diskette Parameters		1EH	Pts to Data
74-77	Video Initialization		1DH	Pts to Data
70-73	Timer Tick	(18.2/sec)	1CH	Software Interrupts
6C-6F	Keyboard Break		1BH	Software Interrupts
68-6B	Time of Day		1AH	Software Interrupts
64-67	Bootstrap		19H	Software Interrupts
60-63	Resident BASIC		18H	Software Interrupts
5C-5F	Printer		17H	Software Interrupts
58-5B	Keyboard		16H	Software Interrupts
54-57	Cassette		15H	Software Interrupts
50-53	Communications		14H	Synchronous
4C-4F	Diskette/Disk		13H	Synchronous
48-4B	Memory		12H	Synchronous
44-47	Equipment Check		11H	Synchronous
40-43	Video		10H	Synchronous
3C-3F	Printer		FH	Hardware Interrupts
38-3B	Diskette		EH	Hardware Interrupts
34-37	Disk		DH	Hardware Interrupts
30-33	Communications		CH	Asynchronous
2C-2F	Communications		BH	Asynchronous
28-2B	Reserved		AH	Asynchronous
24-27	Keyboard		9H	8259A
20-23	Time of Day		8H	8259A
1D-1F	Reserved		7H	8259A
18-1B	Reserved		6H	8259A
14-17	Print Screen		5H	8259A
10-13	Overflow	(CPU)	4H	Microprocessor Interrupts
C-F	Breakpoint	(CPU)	3H	Microprocessor Interrupts
8-B	Non-maskable	(8087)	2H	Microprocessor Interrupts
4-7	Single Step	(CPU)	1H	Microprocessor Interrupts
0-3	Divide by zero	(CPU)	0H	Microprocessor Interrupts

#Intreruperi



Intel 80x86 Interrupt Vector Table in Real Mode

Harta Memoriei PC – in mod real



1. Intreruperi si servicii BIOS/DOS

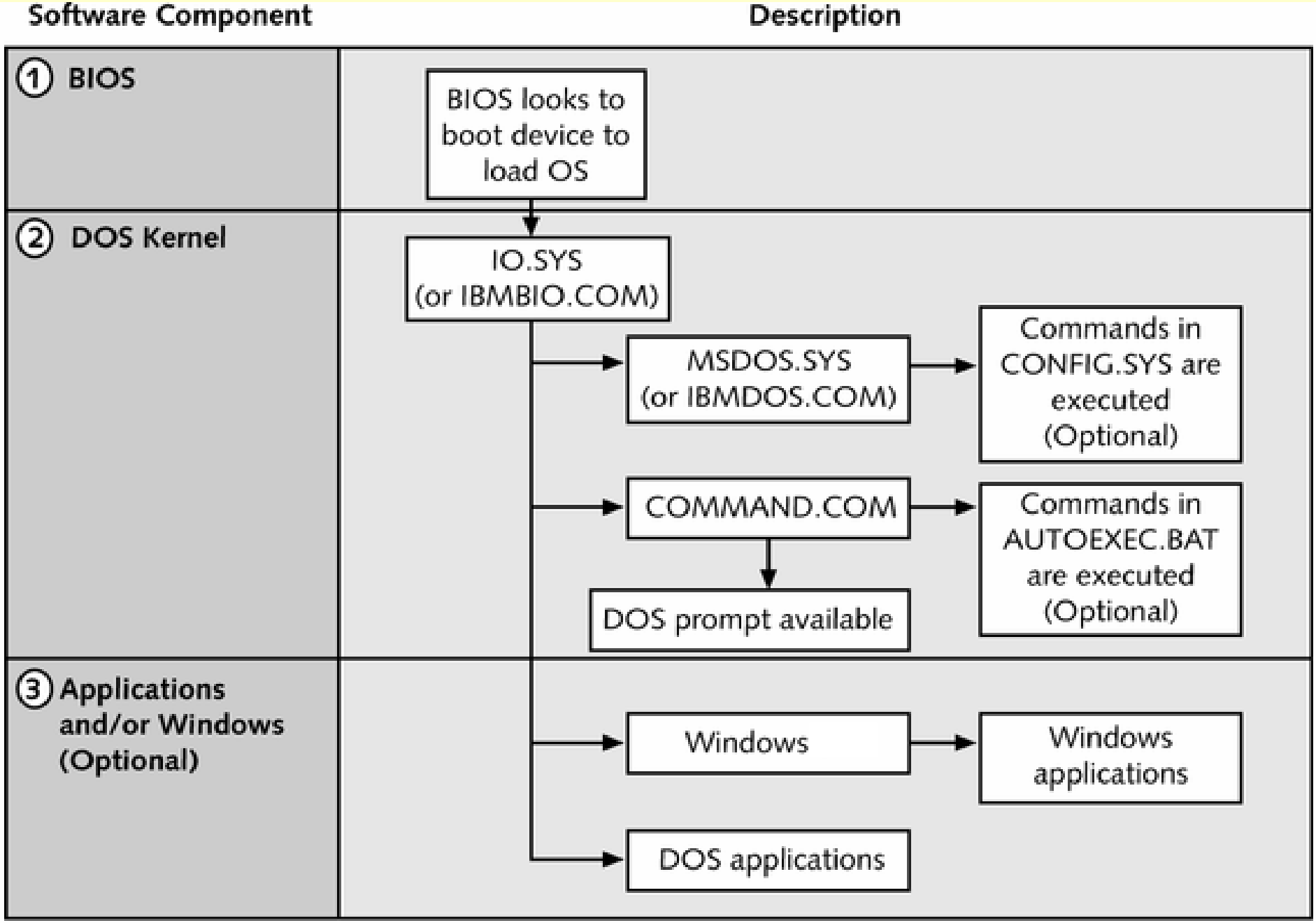
- Scopul SO este de a facilita accesul utilizatorului la resursele sistemului
- SO – o colectie de programe si proceduri
- Procedurile SO *sunt in general apelabile* de catre programele utilizatorilor si asigura accesul acestora la resursele sistemului (periferice, memorie) si la o serie de informatii care descriu contextul functionarii sistemului de calcul

Intreruperi si servicii BIOS/DOS

- Componentele SO DOS:

  - ROM BIOS-ul rezident in FLASH/EPROM/ROM
  - Loader ptr. incarcarea SO (S1,T0, cap0-disk sistem >> 0:7C00h)
  - progr. IO.SYS (IBMIO.SYS)
  - progr. MSDOS.SYS
  - COMMAND.COM
  - progr. utilitare (comenzi externe – Format, Diskcopy, Chkdsk..)

Procesul de Bootare



Intreruperi si servicii BIOS/DOS

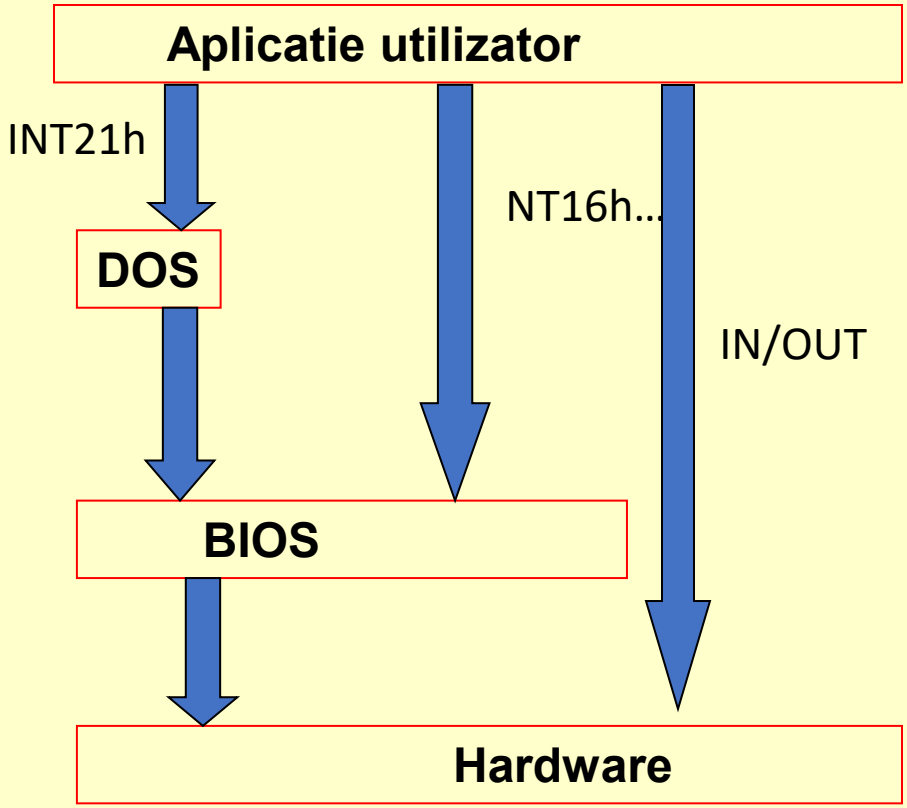
- BIOS-ul – Sistemul de programe de I/O de baza si contine:

- programe de test ptr. toate resursele din configuratia sistemului

- proceduri ptr. tratarea cererilor de intrerupere de la perifericele standard (ceas, tastatura, disk, ecran, …)

- Accesul utilizatorului la procedurile din BIOS se face prin ***INTRERUPERILE software,*** fiecare dispozitiv avand cate o intrerupere rezervata !!!

Intreruperi si servicii BIOS/DOS (cont.)



Intreruperi si servicii BIOS/DOS

Interup.	Adresa TVI	Tip	Descriere
10h	0000:0040h	Software	Video service routine
11h	0000:0044h	Software	Equipment list service routine
12h	0000:0048H	Software	Memory size service routine
13h	0000:004Ch	Software	Hard disk drive service
14h	0000:0050h	Software	Serial communications service routines
15h	0000:0054h	Software	System services support routines
16h	0000:0058h	Software	Keyboard support service routines
17h	0000:005Ch	Software	Parallel printer support services
18h	0000:0060h	Software	Load and run ROM BASIC
19h	0000:0064h	Software	DOS loading routine
1Ah	0000:0068h	Software	Real time clock service routines
1Bh	0000:006Ch	Software	CRTL - BREAK service routines
1Ch	0000:0070h	Software	User timer service routine
1Dh	00000074h	Software	Video control parameter table
1Eh	0000:0078h	Software	Floppy disk parameter routine
1Fh	0000:007Ch	Software	Video graphics character routine
20h-3Fh	0000:0080h - 0000:00FCh	Software	DOS interrupt points

Intreruperi si servicii BIOS/DOS

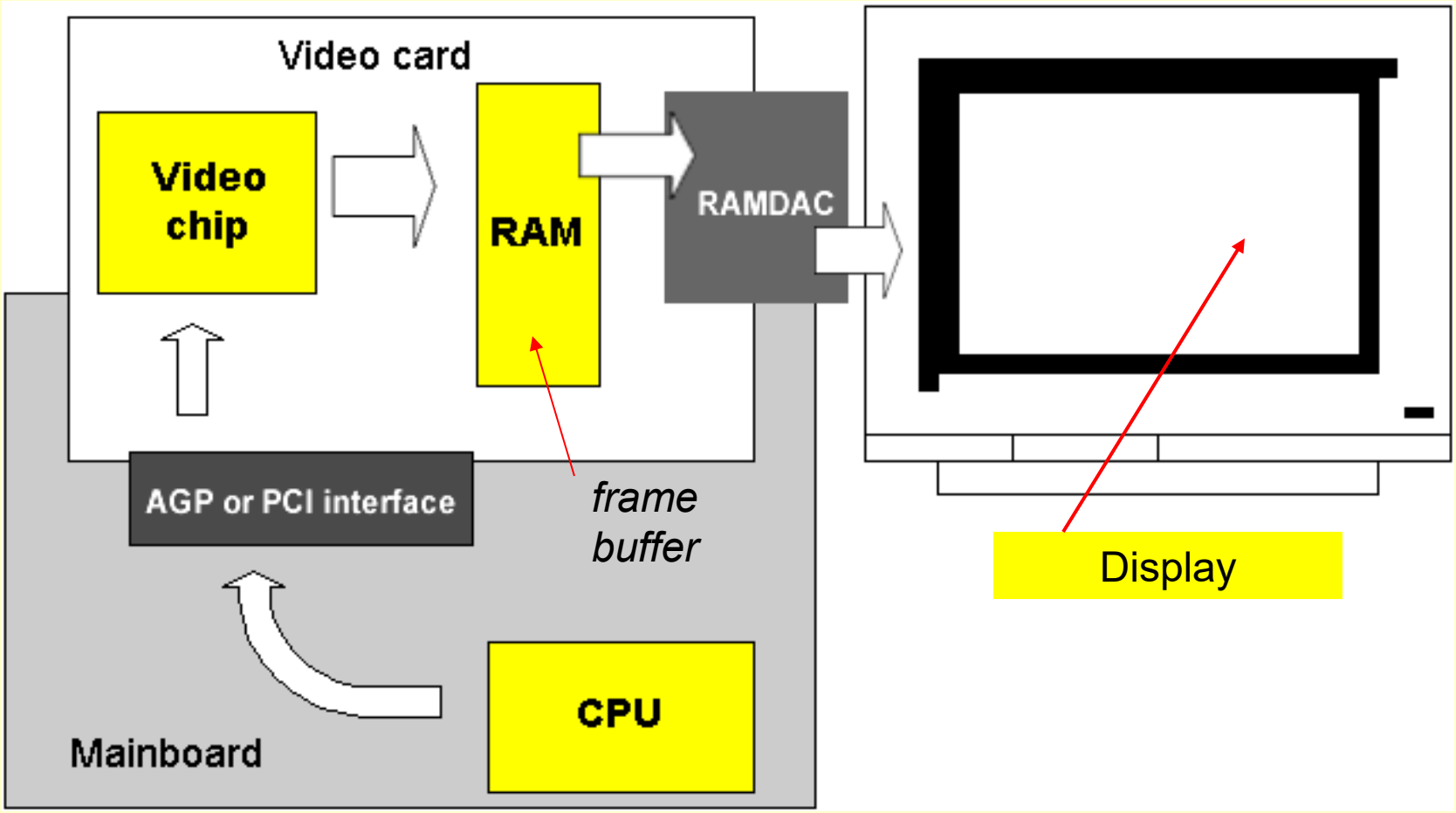
- Fiecare INTRERUPERE software are o paleta de servicii (functii) care se apeleaza prin incarcarea registrului :

  AH=nr. functie + parametrii in reg.

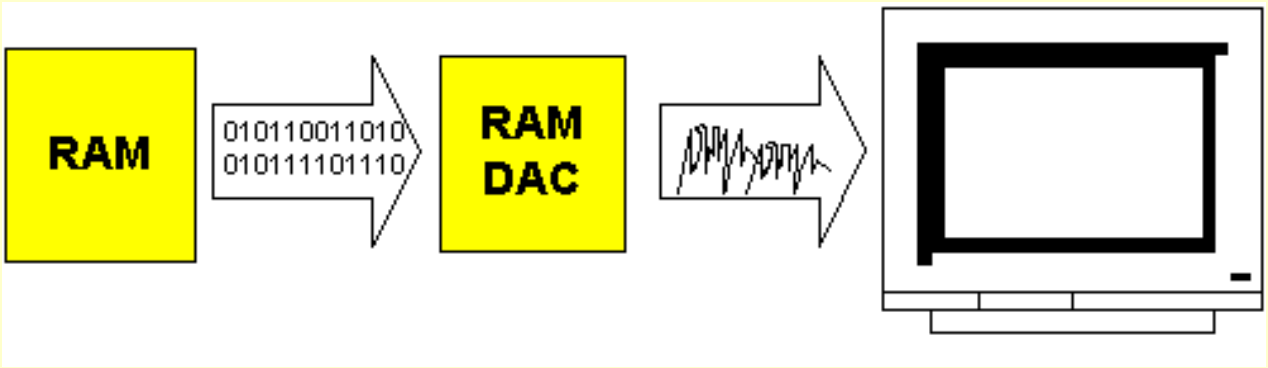
  inainte de apelul prin INT x.

- Serviciile DOS in general apelabile prin **INT 21h** , sunt mai lente, dar superioare d.p.v. al compatibilitatii software si facilitatilor de redirectare
- Serviciile BIOS si DOS ofera avantajul *independentei fata de implementa-rile hard particulare*

2. INT 10h – Servicii Video



INT 10h – Servicii Video

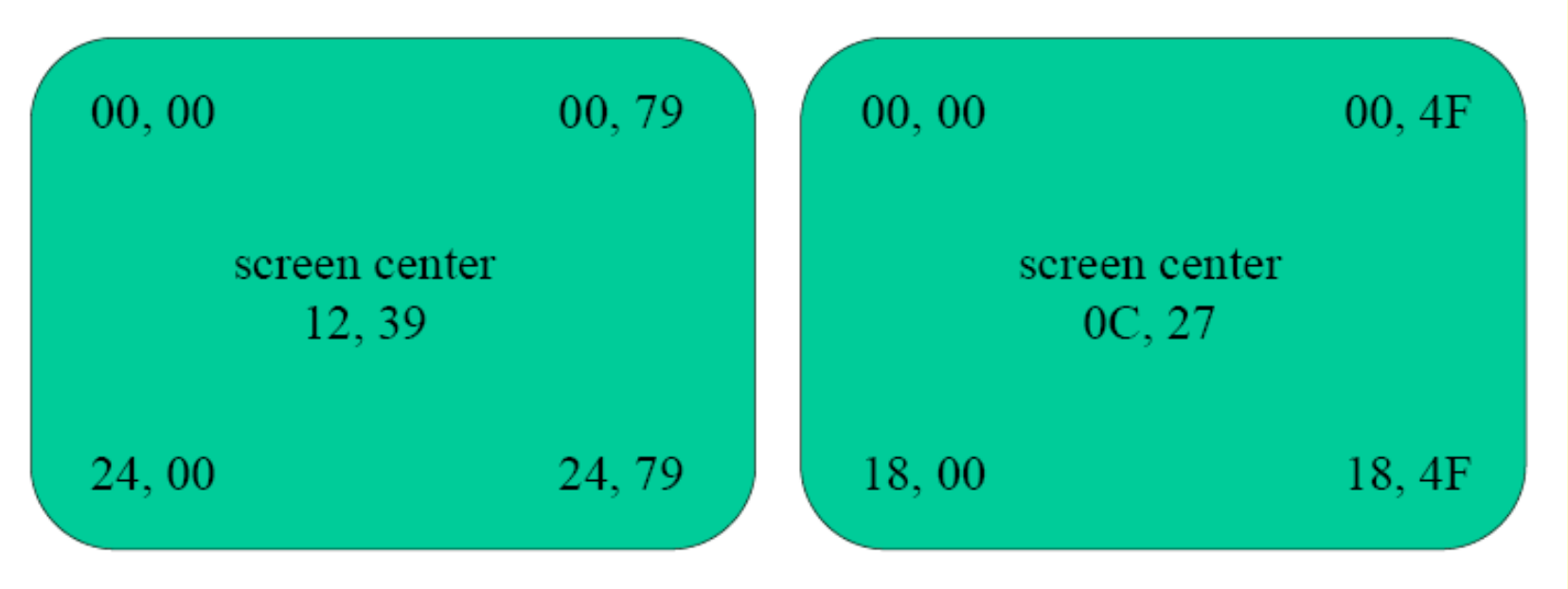


Rezolutie	Dimensiune Bit map cu 2 biti culoare	RAM necesar pe video card
640 x 480	614,400 bytes	1 MB
800 x 600	960,000 bytes	1.5 MB
1024 x 768	1,572,864 bytes	2 MB
1152 x 864	1,990,656 bytes	2.5 MB
1280 x 1024	2,621,440 bytes	3 MB
1600 x 1200	3,840,000 bytes	4 MB

INT 10h – Servicii Video- Mod text

- In mod text monitorul afiseaza uzual 80 x 25 caractere
- Tipuri de monitoare: MDA, CGA, EGA, VGA, SVGA....
- Memoria video (0B0000h; 0A0000h; 0B8000h)
- Fiecare caracter are in memoria video 2 octeti (cod ASCII + octet de atribut) pe baza lor, CRTC gen. semnalul video corespunzator)

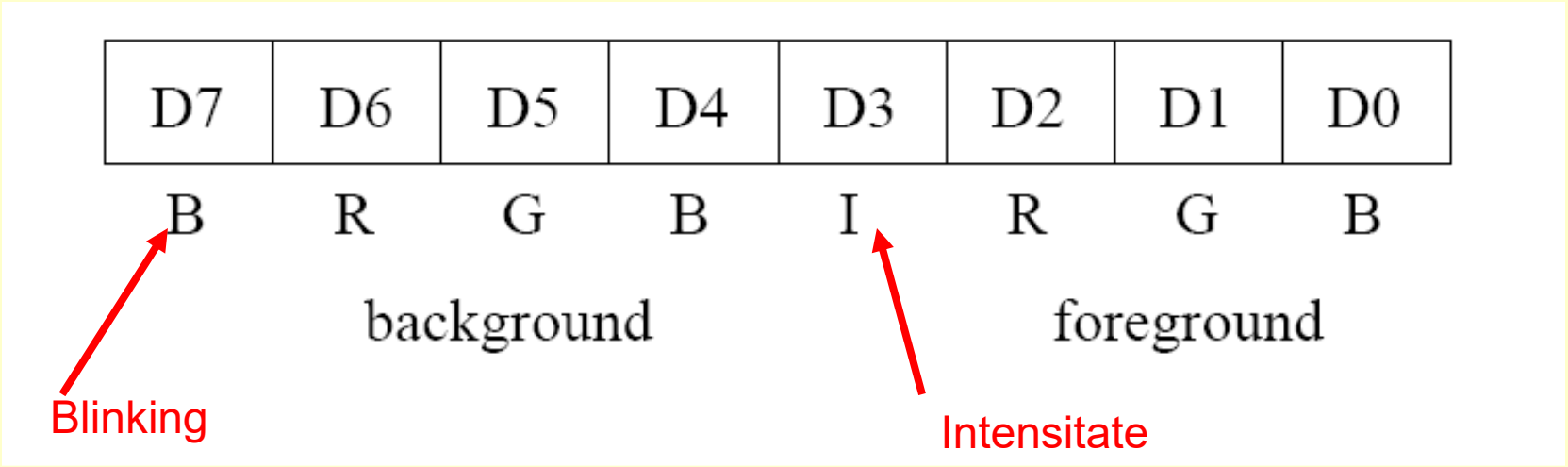
– CRTC = Catod Ray Tube Controller



INT 10h – Servicii Video

Atribut de culoare pentru mod CGA

Pag. video = 25x80x2 Bytes = 4000 ~ 4KB



INT 10h – Servicii Video

Pentru monitor color RGB :

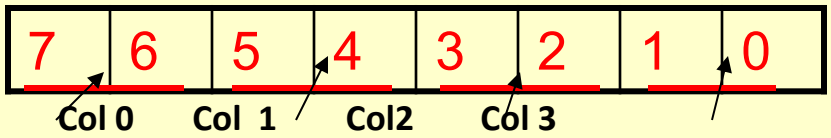
Binary	Color	
0000	Black	
0001	Blue	
0010	Green	
0011	Cyan	
0100	Red	
0101	Magenta	
0110	Brown	
0111	Light Gray	
1000	Dark Gray	
1001	Light Blue	
1010	Light Green	
1011	Light Cyan	
1100	Light Red	
1101	Light Magenta	
1110	Yellow	
1111	White	

INT 10h – Servicii Video- Mod grafic

- **Mod grafic** CGA : 320x200 / 4 culori /2biti/pixel - 2 palete
  (4 pixeli/octet)

Memoria video (CGA) incepe la B800h:0000h

Octet 0 → col 0,1,2,3



Daca X=col. (0….319) si Y=linia (0…199)

Offset= 2000h*(Ymod2)+80*int(Y/2)+int(X/4)

Nr. Bit= 6 - 2*(X mod 4)

Ex. Pixel: X=100; Y=160

Offset = 6425; bit= 6

Offset	
0000h	LINIA 0 (80 octeti)
0050h	LINIA 2 (80 octeti)
00A0h	LINIA 4 (80 octeti)
……..	
1EF0h	LINIA198 (80 octeti)
1F40h	nefolosit
2000h	LINIA 1 (80 octeti)
2050h	LINIA 3 (80 octeti)
20A0h	LINIA 5 (80 octeti)
……	
3EF0h	LINIA199 (80octeti)
3F40h	nefolosit

INT 10h – Servicii Video

- Mod grafic CGA : 640x200 / 2 culori /1bit/pixel

  Octet 0 ⟶ col 0,1,….7

Offset= 2000h*(Ymod2)+80*int(Y/2)+int(X/8)

Nr. Bit= 7 - (X mod 8)

Ex. X=100; Y=160. Offset=6412, bit=3

INT 10h – Servicii Video

Servicii Video

AH=00h	Set Video Mode
AH=01h	Set Cursor Shape
AH=02h	Set Cursor Position
AH=03h	Get Cursor Position And Shape
AH=04h	Get Light Pen Position
AH=05h	Set Display Page
AH=06h	Clear/Scroll Screen Up
AH=07h	Clear/Scroll Screen Down
AH=08h	Read Character and Attribute at Cursor
AH=09h	Write Character and Attribute at Cursor
AH=0Ah	Write Character at Cursor
AH=0Bh	Set Border/Palette Color
AH=0Eh	Write Character in TTY Mode
AH=0Fh	Get Video Mode
AH=13h	Write String

INT 10h – Servicii Video

- **AH=0: set video mode (clear screen)**

AL= 0: text 40*25 16 color (mono)
1: text 40*25 16 color
2: text 80*25 16 color (mono)
3: text 80*25 16 color
4: CGA 320*200 4 color
5: CGA 320*200 4 color (m)
6: CGA 640*200 2 color
……

0Dh: EGA 320*200 16 color
0Eh: EGA 640*200 16 color
0Fh: EGA 640*350 mono
10h: EGA 640*350 16 color
11h: VGA 640*480 16 color
12h: VGA 640*480 16 color
13h: VGA 320*200 256 color

```
        .MODEL TINY
        .CODE
        .STARTUP

MOV AH, 0               ;set function number
MOV AL, 3               ;mode 3, standard text mode
INT 10H                 ;the screen is found to be cleared
.EXIT
END
```

INT 10h – Servicii Video

- **AH=2: set cursor location**

BH= display page (normal=0)

DH,DL= row,col (0..24, 0..79)

```
        .MODEL TINY
        .CODE
        .STARTUP

MOV AH, 0           ;set function number
MOV AL, 3           ;mode 3, standard text mode
INT 10H             ;the screen is cleared with this.

MOV AH, 02          ;function number for setting cursor position
MOV BH, 0           ;specify page (which is the default page
MOV DH, 09          ;row co-ordinate
MOV DL, 35          ;column co-ordinate
INT 10H
        .EXIT
         END
```

- **AH=5: select active page,** AL= page 0..
- **AH=6: scroll up (clear screen) ; AH=7: scroll down**

AL= lines (=0: clear screen)

BH= attribute (=7: white on black)

CH= upper line (=0: top)

CL= left col (=0: far left)

DH= lower line (=18h(24): jos)

DL= right col (=4Fh(79): far right) DX=184Fh(6223)

Ex1. Deplaseaza ecranul cu 0 linii adica tot ecranul. Tot ce se va tipari va fi negru/ cyan.

```
MOV AH, 06          ;function number
MOV AL, 00          ;number of lines to be scrolled up
MOV BH, 30H         ;cyan background, black foreground
MOV CH, 0           ;starting row co-ordinate = 0
MOV CL, 0           ;starting column co-ordinate = 0
MOV DH, 24          ;ending  row co-ordinate = 24
MOV DL, 79          ;ending  column co-ordinate = 79
INT 10H
```

Ex2. Se creaza o noua fereastra pe ecran de dimensiune diferita, si culoare diferita.

```
MOV AH, 06        ;function number
MOV AL, 9         ;number of lines to be scrolled up
MOV BH, 61H       ;brown background, blue foreground
MOV CH, 15        ;starting row co-ordinate = 15
MOV CL, 20        ;starting column co-ordinate = 20
MOV DH, 23        ;ending row co-ordinate = 23
MOV DL, 60        ;ending column co-ordinate = 60
```

INT 10h – Servicii Video

- **AH=9: write char&attrib**

AL=char

BH=display page (0)

BL=attribute

CX=no. of repetitions =2000 (7D0h) to fill 80*25 screen

```
MOV AH, 09          ;function number
MOV AL, 'S'         ;display 'S' at the cursor position
MOV BH, 0           ;page 0
MOV BL, 16H         ;blue background, brown foreground
MOV CX, 9           ;display the character 9 times
INT 10H
```

Ex. Sa se scrie un program care sa :

- seteze modul video 3 si sa curete ecranul
- sa genereze o fereastra de o marime si culoare specificata (albastru pe maro)
- seteaza cursorul intr-o pozitie specificata in fereastra
- afiseaza de 10x caracterul * la pozitia cursorului.

```
        .MODEL TINY
        .CODE
        .STARTUP

MOV AH, 0           ;set video mode
MOV AL, 3           ;mode 3, normal text mode
INT 10H             ;clear screen

MOV AH, 06          ;function number for scrolling up
MOV AL, 10          ;no of lines = 10
MOV BH, 61H         ;brown background, blue foreground
MOV CH, 5           ;starting row co-ordinate = 5
MOV CL, 20          ;starting column co-ordinate = 20
MOV DH, 14          ;ending row co-ordinate = 14
MOV DL, 60          ;ending column co-ordinate = 60
INT 10H             ;create the scroll window

MOV AH, 02          ;function number to set the cursor
MOV BH, 0           ;page number = 0
MOV DH, 09          ;row co-ordinate of cursor
MOV DL, 39          ;column co-ordinate of cursor
INT 10H             ;cursor at (9, 39)

MOV AH, 09          ;function to display character
MOV AL, '*'         ;character to be displayed = '*'
MOV BH, 0           ;page number
MOV BL, 16H         ;blue with brown text
MOV CX, 10          ;count = 10
INT 10H             ;display the character 10 times
        .EXIT
        END
```

- **AH=0Bh: set border/palette**

BH=0,color (0-15): Background/border color (border only in text modes)
BH=1, BL=0/1, palette ID(0-1):
CGA 320x200: palette
00h background, green, red, and brown/yellow
01h background, cyan, magenta, and white

- **AH=0Eh Write Character in TTY Mode**

input: AL = char; BH = pag. ; folosesc atribut existent si incrementez pozitia

Exemplu:
```
mov ah, 0eh
mov bh, 0
mov al, '$'
int 10h     ; scrie $ la pozitia curenta
```

INT 10h – Servicii Video

- **AH=0Ch: write pixel**

AL=pixel color
(if bit 7 is 1, the new pixel color bits will be XOR-ed with the color bits of the current pixel)
CX=column 0..319
DX=row 0..199

- **INT 10h** / **AH = 0Ch** - change color for a single pixel.

input: A**L** = pixel color; C**X** = column.; **DX** = row.

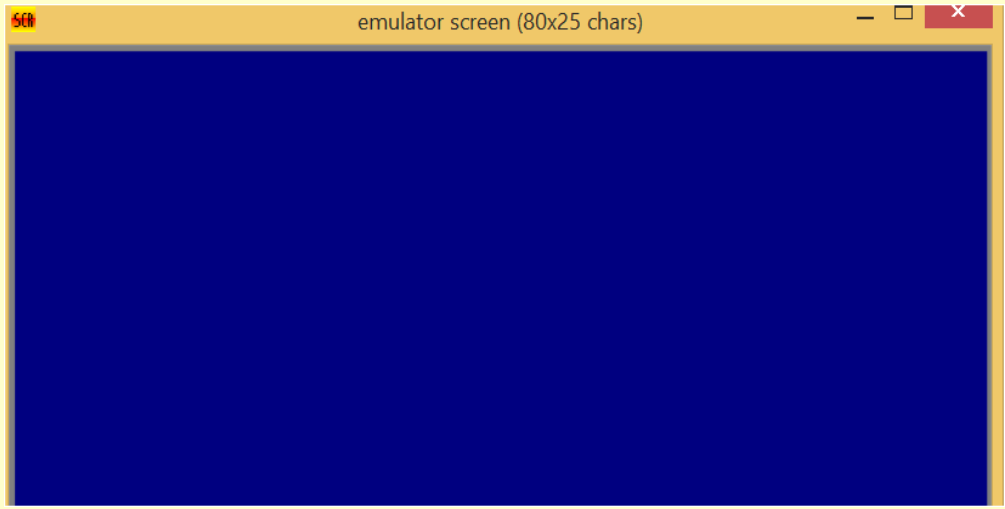
```
Ex.:    mov al, 13h     ; VGA mode
        mov ah, 0
        int 10h         ; set graphics video mode to VGA
        mov al, 00001100b
        mov cx, 10      ; col=10
        mov dx, 20      ; row=20
        mov ah, 0ch
        int 10h         ; set pixel.
```

INT 10h – Servicii Video

Exemplu. Scrieti un program care pune 20H (spatiu) pe intreg ecranul (curate ecranul). Folositi alb intens pe fond albastru, ca atribut de culoare.

```
MOV AH, 00          ;set mod video
MOV AL, 03          ;mod 3, CGA color text, 80 x 25
INT 10H
MOV AH, 9           ;servici display
MOV BH, 0           ;pag. Video 0
MOV AL, 20H         ;cod ASCII ptr. space
MOV CX, 2000         ;repet de 2000 x
MOV BL, 1FH         ;atribut alb intens
INT 10H              ; pe fond albastru
```

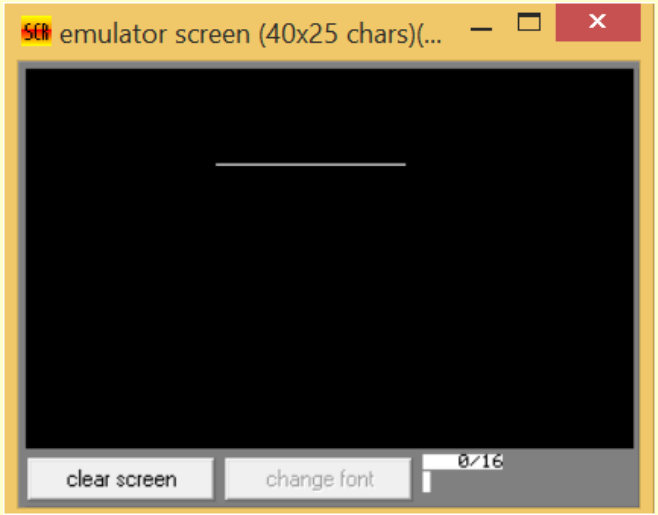
- Scrieti alta secventa de program care sa aiba acelasi efect



INT 10h – Servicii Video

- **Exemplu.** Scrieti un program care sa :(a) stearga ecranul, (b) seteaza mod video CGA rezolutie 640 x 200,(c) deseneaza o linie orizontala alba pornind din col = 100, rind = 50, pana la col= 200, rind= 50.

```
        MOV AX, 0600H   ;scroll up ecran
        MOV BH, 07      ; atribut normal alb/negru
        MOV CX, 0       ;de la rnd = 00, col= 00 (colt stanga sus)
        MOV DX, 184FH   ;la rnd = 18H, col = 4FH (colt dreapta jos)
        INT 10H         ; intrerupere ptr.clear screen
        MOV AH, 00      ;set mod video
        MOV AL, 06      ;mod = 06 (CGA high resolution)
        INT 10H         ;intrerupere ptr setare mod video
        MOV CX, 100     ;start linie de la col.100
        MOV DX, 50      ;rnd 50
BACK:   MOV AH, 0CH     ;AH = 0CH ptr. desenare
        MOV AL,01       ;pixeli albi
        INT 10H ;       ;int 10h ptr desenare linie
        INC CX          ;increment pozitie orizontala
        CMP CX, 200     ;desenez linie pina la col 200
        JNZ BACK
```



INT 10h – Servicii Video

- **Exemplu.** Scrieti un program care sa:

(a) schimbe modul video in CGA 320 x 200 grafic , 4 culori (mod AL=4);

(b) sa coloreze ecranul albastru.

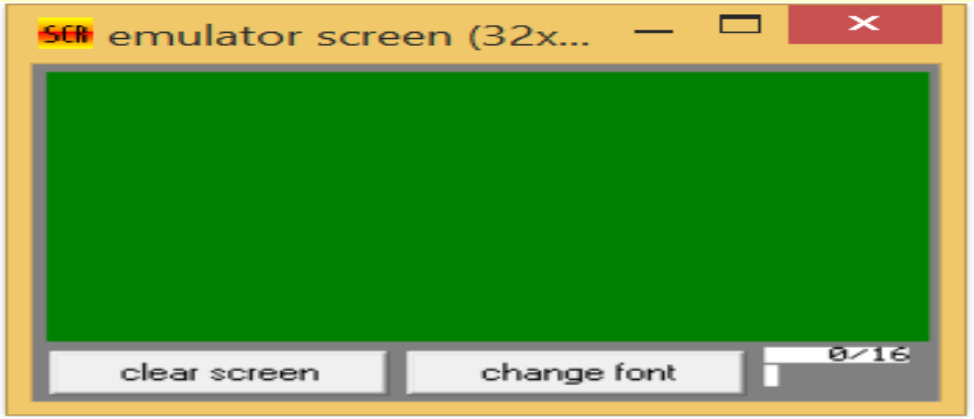
```
MOV AH, 00      ;set mod video
MOV AL, 4       ;CGA standard 320x200 , 4 culori
INT 10H

MOV AH, 0Bh     ;set paleta culoare
MOV BH, 0       ;set culoare background paleta 0
MOV BL, 1       ;albastru
INT 10H
```

INT 10h – Servicii Video

- **Exemplu.** Analizati secventa de program si stabiliti efectul.

```
        ;Afisez  ce????
        mov si,82h          ;parametrii din PSP
mess:   mov al,[si]  ;fiecare caracter
        cmp al,0Dh          ;pana la CR
        jz quit
        mov bh,0            ;pag. 0
        mov ah,0Eh
        int 10h             ; INT 10H
        inc si              ; next caracter
        jmp mess
quit:
```

Ex. Afisarea prin accesarea memoriei video direct. Pentru CGA avem memoria video incepand cu B800:0h. Se cere sa se faca intreg ecranul verde.




```
        .MODEL SMALL
        .CODE
        .STARTUP
    MOV AH, 0
    MOV AL, 3
    INT 10H             ;this is used to clear the screen

    CLD                 ;clear direction flag
    MOV AX, 0B800H      ;starting address of video memory
    MOV ES, AX          ;make video memory the extra segment
    MOV DI, 0           ;to point to first location in ES
    MOV CX, 25*80       ;characters on the screen (2000)
    MOV AH, 29H         ;attribute for green background
    MOV AL, 20H         ;ASCII code of space bar key
REP STOSW               ;store this data in the 2000 locations
            .EXIT
            END
```

3. INT 16h – Servicii Tastatura

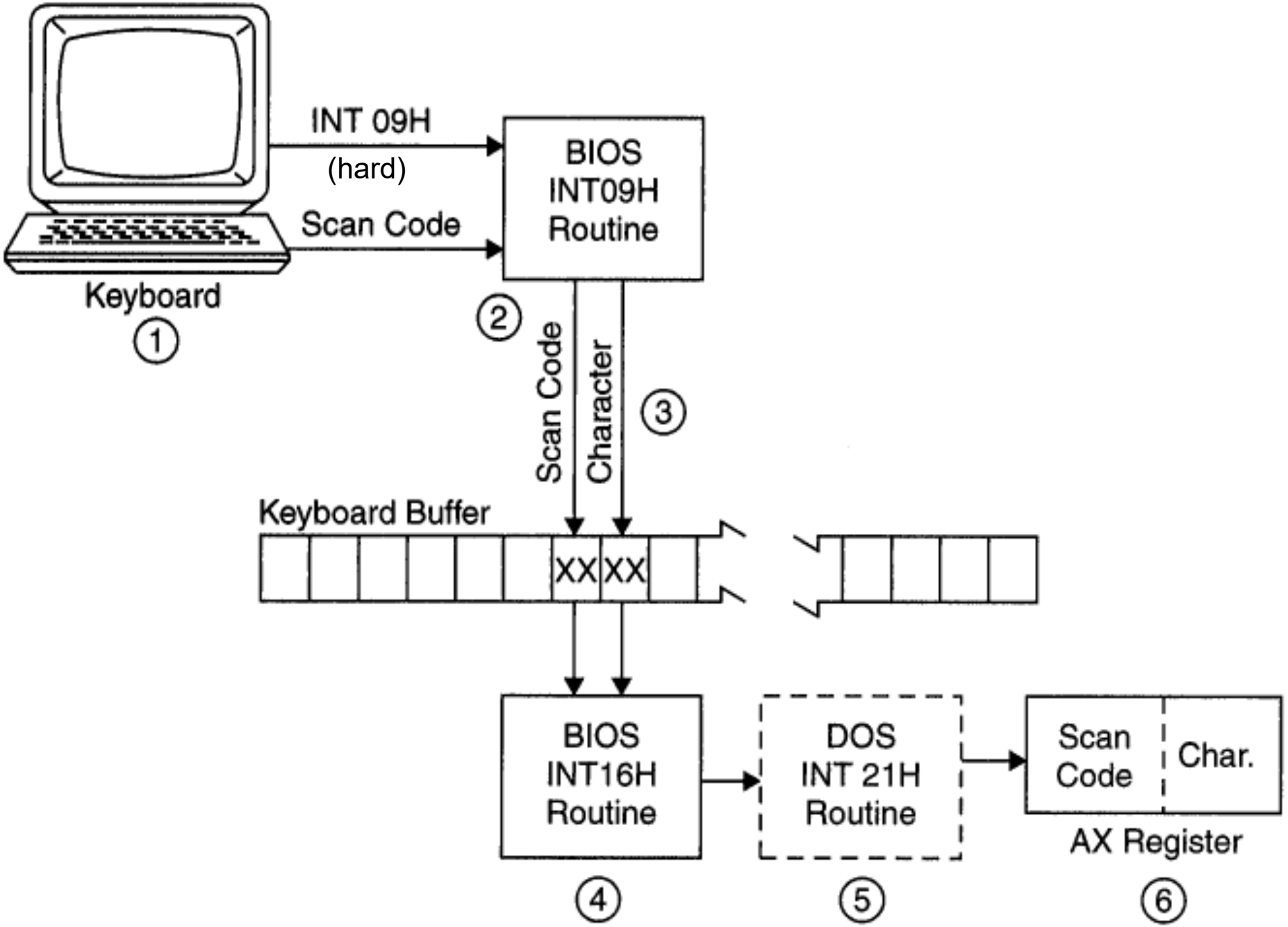
Tastatura are rezervate 2 intreruperi : tip: 9h si 16h

- La apasarea unei taste se genereaza intr. hard IRQ1 (tip 9h) la PIC 8259A
- Accesul la rutinele care trateaza tastatura se face prin INT16h (port 60h)
- La tastare se trimite cod MAKE, iar la eliberare cod BREAK (80h+MAKE)
- Rutina BIOS (09h)

  – transforma cod Make+Breake in cod ASCII si interpreteaza

  CTRL+ALT+DEL – reset warm;
  SHIFT+PrtScr - ecran=> LPT
  Ctrl+NumLock - oprire executie pina la tastare

- Interpretarea codului de scanare e fct. de tastele Shift (Alt,Ins, Caps, Numlock, Ctrl,…)
- Codurile =>> Buffer circular 0: 41Eh (BIOS data)
- Starea tastelor Shift (2 octeti) 0:417h, 0:418h


INT 09H
(hard)
Scan Code
Keyboard
1
BIOS
INT09H
Routine
2
Scan Code
Character
3
Keyboard Buffer
XX
XX
BIOS
INT16H
Routine
4
DOS
INT 21H
Routine
5
Scan
Code
Char.
AX Register
6

INT 16h – Servicii Tastatura

1. Tastatura genereaza intreruperea 09h (hard)
2. Rutina INT 09h primeste codul de scanare de la tastatura si gaseste caracterul asociat (cod ASCII daca $\exists$)
3. INT 09h pune caracterul si codul de scanare in buffer-ul tastaturii (zona de date BIOS/DOS)

4/5. Aplicatia utilizator cere serviciu prin INT 16h sau INT 21h

6. INT 16h acceseaza buffer-ul si pune in AL =codul ASCII si in AH= codul de scanare

INT 16h – Servicii Tastatura

AH=00h	Read Character
AH=01h	Read Input Status
AH=02h	Read Keyboard Shift Status
AH=03h	Set typematic rates
AH=10h	Read Character Extended
AH=11h	Read Input Status Extended
AH=12h	Read Keyboard Shift Status Extended

INT 16h – Servicii Tastatura

- Servicii INT 16h

- AH=0 ; citesc tasta, sterge buffer
  AL ← ASCII code
  AH ← scan (sau second code)

- AH=1 ; verific daca s-a tastat ceva, ramane in buffer
  ;ZF ← 0 daca s-a tastat ceva
  ; 1 daca buffer e gol

- AH=2 ; Copiez shift status byte (0040:0017h) in AL

INT 16h – Servicii Tastatura

- AH=3; Setez parametrii de Tastare (BH=00…11h, delay ;
  BL=00…1fh, rep.rate)

Typematic Initial Delay Values	
bh Register Content	Delay (milliseconds)
00h	250
01h	500
02h	750
03h	1000
04h through **ffh**	RESERVED

Typematic Character Repeat Rates (**bl** Value = Rate in Char/Sec)		
00h = 30.0	0bh = 10.9	16h = 4.3
01h = 26.7	0ch = 10.0	17h = 4.0
02h = 24.0	0dh = 9.2	18h = 3.7
03h = 21.8	0eh = 8.6	19h = 3.3
04h = 20.0	0fh = 8.0	1ah = 3.0
05h = 18.5	10h = 7.5	1bh = 2.7
06h = 17.1	11h = 6.7	1ch = 2.5
07h = 16.0	12h = 6.0	1dh = 2.3
08h = 15.0	13h = 5.5	1eh = 2.1
09h = 13.3	14h = 5.0	1fh = 2.0
0ah = 12.0	15h = 4.6	20h through ffh RESERVED

INT 16h – Servicii Tastatura

AH= **10h** – Extended Keyboard Read -Similar Serviciului **00h,**
dar prelucreaza taste Speciale, Returneaza codurile ASCII/Scan ptr toate tastaturile de 101/102 taste

AH= **11h** – Get Extended Keystroke Status – Similar Serviciului **01h,**
ptr tastaturile cu 101/102 taste

AH= **12h** – Get Extended Shift Status

- facilitati suplimentare ptr tastaturi cu 101/102 taste
- Returneaza **0040:0017** in **al**
- Returneaza **0040:0018** in **ah**

Table 78: BIOS Keyboard Support Functions

Function # (AH)	Input Parameters	Output Parameters	Description
0		al- ASCII character ah- scan code	Read character. Reads next available character from the system's type ahead buffer. Wait for a keystroke if the buffer is empty.
1		ZF- Set if no key. ZF- Clear if key available. al- ASCII code ah- scan code	Checks to see if a character is available in the type ahead buffer. Sets the zero flag if not key is available, clears the zero flag if a key is available. If there is an available key, this function returns the ASCII and scan code value in ax. The value in ax is undefined if no key is available.
2		al- shift flags	Returns the current status of the shift flags in al. The shift flags are defined as follows: bit 7: Insert toggle bit 6: Capslock toggle bit 5: Numlock toggle bit 4: Scroll lock toggle bit 3: Alt key is down bit 2: Ctrl key is down bit 1: Left shift key is down bit 0: Right shift key is down
3	al = 5 bh = 0, 1, 2, 3 for 1/4, 1/2, 3/4, or 1 second delay bl = 0..1Fh for 30/sec to 2/sec.		Set auto repeat rate. The bh register contains the amount of time to wait before starting the autorepeat operation, the bl register contains the autorepeat rate.
5	ch = scan code cl = ASCII code		Store keycode in buffer. This function stores the value in the cx register at the end of the type ahead buffer. Note that the scan code in ch doesn't have to correspond to the ASCII code appearing in cl. This routine will simply insert the data you provide into the system type ahead buffer.

Ex. Scrieti un program care umple ecranul cu caracterul introdus de la tastatura. Daca nu se tasteaza nimic, se asteapta apasarea unei taste.

```
                .MODEL TINY
                .CODE
                .STARTUP
NO_KEY:  MOV AH, 11H      ;function for waiting for key
         INT 16H
         JZ NO_KEY        ;if ZF = 1, no key press, keep waiting
   KEY:  MOV AH, 10H      ;function for taking into AL the key
         INT 16H
         MOV CX, 2000     ;number of characters to fill the screen
DISP:    MOV DL, AL       ;move the ASCII character to DL
         MOV AH, 02       ;function number to display DL content
         INT 21H          ;DOS interrupt 21H
         LOOP DISP        ;display until CX = 0
                .EXIT
                 END
```

4. Intreruperea 1Ch
Executia periodica a unor operatii

- Implementare: prin ceasul de timp-real (timer)

- contor (timer 8253/54) genereaza periodic intreruperi - la fiecare 18.2ms
- rutina de tratare a intreruperii continue si instructiunea INT 1Ch
- Rutina de tratare a intreruperii 1Ch contine o singura instructiune : IRET - revenire din rutina
- Prin redirectarea intreruperii 1Ch catre o anumita rutina, aceasta va fi apelata la fiecare intrerupere a ceasului de timp real
- rutina trebuie sa fie rezidenta in memorie

EX. Sa se preia de la tastatura un sir de caractere si sa se afiseze pe ecran (incepand din randul din mijlocul ecranului, col.0) pana la tasta ‘ESC’.

```
        Mov ah,0
        Mov al,3 ;80x25
        INT 10h ; setez mod text 3
        Mov ah,2
        Mov bh,0
        Mov dx,0d00h
        INT 10h ; pozitionez cursorul
STAI:   Mov ah,1
        INT 16h  ; astept tasta
        Jz STAI
        Mov ah,0
        INT 16h ; preiau tasta
        Cmp al, 1bh  ; este ‘ESC’ ?
        Jz GATA
        Mov bl,1
        Mov bh,0
        Mov ah,0eh
        INT 10h  ; afisez caracter
        JMP STAI
GATA:   RET
```

Tema. Desenare linie orizontala

Analizati aplicatia de mai jos.

Programul deseneaza o linie orizontala pe ecran de la locatia (170, 240) la (470,240) scriind pixeli pe ecran folosind functia (AH=0Ch) a INT 10h.

```
TITLE "Program to draw a horizontal line on the screen"
.MODEL SMALL ; this defines the memory model
.STACK 100 ; define a stack segment of 100 bytes
.DATA ; this is the data segment
.CODE ; this is the code segment
MOV AX,@DATA ; get the address of the data segment
MOV DS, AX ; and store it in DS register
MOV AH, 0Fh ; get current video mode
INT 10h
PUSH AX ; save current video mode
MOV AH, 00h ; set video mode
MOV AL, 12h ; graphics 640x480
INT 10h
; draw a green color line from (170, 240) to (470, 240)
        MOV CX, 170 ; start from column 170
        MOV DX, 240 ; and row 240
BACK:   MOV AX, 0C02h ; AH=0Ch and AL = pixel color (green)
        INT 10h ; draw pixel
        INC CX ; go to next column
        CMP CX, 470 ; check if column=470
        JB BACK ; if not reached column=470, then continue
        MOV AH, 07h ; wait for key press to exit program
        INT 21h
        POP AX ; retrieve original video mode
        MOV AH, 00h
        INT 10h ; restore original video mode
        MOV AX, 4C00H ; Exit to DOS function
        INT 21H
        END ; end of the program
```