

## บทที่ 9

### การประมวลผลจอภาพขั้นสูง

#### ADVANCED SCREEN PROCESSING

##### วัตถุประสงค์

หลังจากที่ท่านศึกษาบทนี้ท่านจะมีความเข้าใจดังต่อไปนี้

- การจัดการจอภาพแบบต่างๆ
- การกำหนดอาร์ทิมิต์ของจอภาพ
- การใช้จอสีและกราฟิค

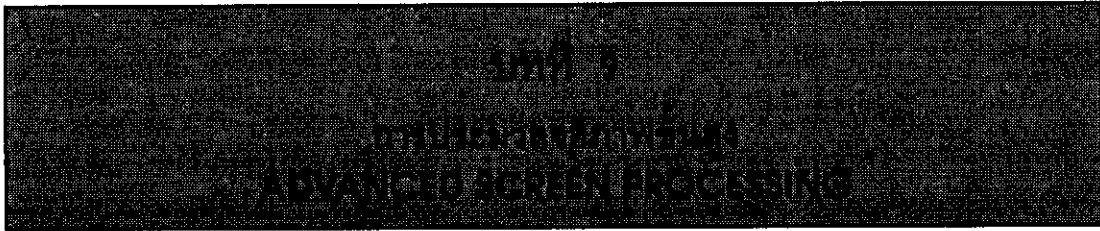
CT 215

217

CT 215

217





## บทนำ

วัตถุประสงค์ของบทนี้ เป็นการขยายการทำงานของจอภาพและคีย์บอร์ดเพิ่มขึ้น ในส่วนของจอภาพ จะมีการเซ็ทโหมด(text or graphics) 2 ชนิด

ในส่วนของ text mode นั้น ท่านสามารถแสดงในจอโรมันโรคมหรือจอสี ขึ้นอยู่กับชนิดของ adapter ที่ติดตั้ง เราจะอธิบายการใช้ attribute สำหรับสี, กะพริบ (blinking) และความเข้ม(high intensity) เหมือนกับคำสั่งที่ใช้เซ็ทขนาดของcursor ตำแหน่ง cursor การ scroll up or scroll down ของจอภาพ และการแสดงผลตัวอักษร หลังจากนั้นจะใช้ graphics mode กับคำสั่งต่างๆที่ใช้ในจอสี

## วิดีโอ VIDEO ADAPTERS

video adapters ที่ใช้ทั่วไปคือ

MDA	MONOCHROME DISPLAY ADAPTER
HGC	HERCULES GRAPHICS CARD
CGA	COLOR GRAPHICS ADAPTER
EGA	ENHANCED GRAPHICS ADAPTER
MCGA	MULTICOLOR GRAPHICS ARRAY (PS/2 MODEL 25,30)
VGA	VIDEO GRAPHICS ARRAY

video adapter ประกอบด้วยหน่วยพื้นฐาน 3 ส่วน

1. VIDEO CONTROLLER หรือ workhorse unit เป็นตัวจ่ายสัญญาณการสแกนให้มอนิเตอร์ สำหรับเลือก text or graphics ตัวโปรเซสเซอร์ของคอมพิวเตอร์ จะส่งคำสั่งไปยังรีจิสเตอร์ควบคุม และอ่านสถานะของข่าวสารจากรีจิสเตอร์ควบคุม

2. VIDEO RAM เป็นหน่วยความจำที่ใช้เก็บข่าวสารสำหรับการแสดงผลของมอนิเตอร์ การอินเทอร์รัพท์ จะเป็นการเคลื่อนย้ายข่าวสาร ที่จะแสดงผลไปยังหน่วยความจำนี้โดยตรง ตามแอดเดรสเซกเมนต์ของหน่วยความจำ สำหรับแอดเดรสหลักของ video adapter คือ

B000:[0] MONOCHROME TEXT MODE FOR MDA, EGA AND VGA

B100:[0] FOR HGC

B800:[0] TEXT MODE (COLOR OR BLACK AND WHITE) FOR CGA MCGA AND VGA และในส่วนของ GRAPHICS MODES (COLOR OR BLACK AND WHITE) FOR CGA, EGA, MCGA AND VGA

A000:[0] ใช้สำหรับ font descriptor เมื่ออยู่ใน text mode และบัฟเฟอร์ video รุ่นใหม่ที่มี high - resolution graphics สำหรับ EGA, MCGA, VGA พื้นที่ของหน่วยความจำดังกล่าวขึ้นอยู่กับ video mode โหมดเหล่านี้แสดงในรูปแบบที่ 9-1

3. VIDEO BIOS ซึ่งทำหน้าที่เชื่อม BIOS กับ video adapter เช่นข้อมูลในการเซต cursor และแสดงผลตัวอักษร

#### INT 10H SERVICE 00 : SETTING THE MODE

ในระบบจะมีการทำงานของอินเทอร์รัพท์ในการเซตโหมด text or graphics การทำงานนี้จะสามารถเซตโหมดได้ในโปรแกรมหรือสวิตช์ระหว่าง text and graphics ในจอสี การเซตโหมดจะต้องมีการเคลียร์จอภาพ ในรูป 9-1 การเซตโหมดต่างๆจะมีตัวเลขอยู่ซ้ายมือ ตัวอย่างในโหมด 03 ซึ่งเป็น text mode, color และค่า screen resolution ขึ้นอยู่กับชนิดของมอนิเตอร์ การเซตโหมดใช้ INT 10h และการบริการ 00 เก็บใน AH และโหมดใหม่อยู่ใน AL ตัวอย่างต่อไปนี้เป็นการเซต video mode สำหรับ standard color text

```
mov    ah,00          ; request set mode
mov    al,03          ; 80x25 standard color text
int    10h           ; call BIOS
```

```

MOV AH,00 ;Request set mode
MOV AL,03 ;80x25 standard color text
INT 10H ;Call BIOS

```

<i>Mode</i>	<i>Text/ Graphics</i>	<i>Mono/ Color</i>	<i>Adapter</i>	<i>Resolution</i>	<i>Colors</i>
00	Text (25 rows, 40 cols.)	Mono	CGA EGA MCGA VGA	320 × 200 320 × 350 320 × 400 360 × 400	
01	Text (25 rows, 40 cols.)	Color	CGA EGA MCGA VGA	320 × 200 320 × 350 320 × 400 360 × 400	16 colors 16 of 64 16 of 262,144 16 of 262,144
02	Text (25 rows, 80 cols.)	Mono	CGA EGA MCGA VGA	640 × 200 640 × 350 640 × 400 720 × 400	
03	Text (25 rows, 80 cols.)	Color	CGA EGA MCGA VGA	640 × 200 640 × 350 640 × 400 720 × 400	16 colors 16 of 64 16 of 262,144 16 of 262,144
04	Graphics	Color	CGA, EGA, MCGA, VGA	320 × 200	4 colors
05	Graphics	Mono	CGA, EGA, MCGA, VGA	320 × 200	
06	Graphics	Mono	CGA, EGA, MCGA, VGA	640 × 200	
07	Text (25 rows, 80 cols)	Mono	MDA EGA VGA	720 × 350 720 × 350 720 × 400	
0D	Graphics	Color	EGA, VGA	320 × 200	16 colors
0E	Graphics	Color	EGA, VGA	640 × 200	16 colors
0F	Graphics	Mono	EGA, VGA	640 × 350	
10	Graphics	Color	EGA, VGA	640 × 350	16 colors
11	Graphics	Color	MCGA, VGA	640 × 480	2 of 262,144
12	Graphics	Color	VGA	640 × 480	16 of 262,144
13	Graphics	Color	MCGA, VGA	320 × 200	256 of 262,144

Note 1: Modes 08-0A are for the PCjr  
Note 2: MDA: monochrome display adapter  
CGA: color graphics adapter  
MCGA: multicolor graphics array  
VGA: video graphics array

9-1 Modes for Video Displays

ถ้าท่านเขียนซอฟต์แวร์โดยที่ท่านไม่ทราบชนิดของมอนิเตอร์ ท่านสามารถใช้ BIOS INT 11h เพื่อตรวจสอบอุปกรณ์กับระบบ การทำงานจะดูได้จาก AX ว่าบิตที่ 5 และ 4 เป็นตัวกำหนด video mode

- 01 : 40x25 monochrome using a color adapter
- 10 : 80x25 monochrome using a color adapter
- 11 : 80x25 monochrome using a monochrome adapter

ท่านสามารถตรวจสอบค่า AX เพื่อชนิดของมอนิเตอร์ และเซ็ทโหมดได้ตามต้องการ

### TEXT (ALPHANUMERIC) MODE

text mode คือการประมวลผลปกติในรูปของ ตัวอักษรและตัวเลขบนจอภาพ การประมวลผลคล้ายกันทั้ง monochrome และ color ยกเว้นจอสีไม่มีการขีดเส้นใต้(underline) ในส่วนของ text mode จะเข้าถึงรหัส ASCII ได้ 256 ตัว

### ATTRIBUTE BYTE

การกำหนด attribute byte ในส่วนของ monochrome และจอสีที่เป็น text เพื่อกำหนดลักษณะของการแสดงตัวอักษร บนจอภาพและเครื่องพิมพ์ attribute byte อาจมีรูปแบบดังต่อไปนี้ ตามตำแหน่งของบิต

	BACKGROUND					FOREGROUND		
ATTRIBUTE :	BL	R	G	B	I	R	G	B
BIT NUMBER :	7	6	5	4	3	2	1	0

บิต 0-2 เป็นบิตที่ใช้ควบคุม foreground ของจอภาพ(สีของตัวอักษร) และบิต 4-6 ควบคุม background (สีพื้น) ตัวอักษร R,G และ B เป็นการชี้ตำแหน่งของบิตสำหรับ red, green and blue สำหรับจอสี ส่วนบิต 3(I) เป็นการเซ็ท high intensity และบิต 7(BL)

เป็นการเซ็ตตัวกระพริบ(blinking) สำหรับจอโมโนโครม บิต 0 เป็นการกำหนด attribute ของ underline ทั้งจอสีและโมโนโครม 000 คือดำและ 111 คือขาว ตัวอย่าง 1000 0111 หมายความว่า ตัวขาวบนพื้นดำและกระพริบ และ 0100 0001 หมายความว่า น้ำเงินบนพื้นแดง

การกำหนด attribute จะต้องแน่นอน สำหรับการแสดงในรูป text ของจอโมโนโครมและจอสี เมื่อโปรแกรมมีการกำหนด attribute มันจะจำจนกระทั่งสั่งเปลี่ยนแปลงเป็น attribute อื่น ท่านสามารถใช้ INT 10h ในการจ่าย attribute ของจอภาพ และการทำงานอื่น เช่น scroll up , scroll down การอ่าน attribute/character หรือแสดงผล attribute/character ถ้าท่านใช้โปรแกรม debug แสดง video ท่านจะเห็นข้อมูลแต่ละไบท์จะมี attribute ตามหลังขนาด 1 ไบท์

### MONOCHROME DISPLAY

มอนิเตอร์ชนิดโมโนโครมส่วนมาก ตัวอักษรที่แสดงจะเป็นสีเขียว เทา หรืออำพัน และสีพื้นจะเป็นสีดำ ถึงแม้ว่าหนังสือเล่มนี้จะอ้างถึงดำและขาว (black and white) ตั้งนั้นในการเปลี่ยน attribute ท่านอาจทำได้ดังต่อไปนี้

ATTRIBUTE	Blackground				Foreground				HEX
	BL	R	G	B	I	R	G	B	
black on black(non display)	0	0	0	0	0	0	0	0	00
underline	0	0	0	0	0	0	0	1	01
white on black,normal video	0	0	0	0	0	1	1	1	07
white on black,blinking	1	0	0	0	0	1	1	1	87
white on black,intense	0	0	0	0	1	1	1	1	0F
black on white,reverse video	0	1	1	1	0	0	0	0	70
black on white,reverse blinking	1	1	1	1	0	0	0	0	F0

## COLOR DISPLAY

สำหรับจอสีส่วนมาก ตัวอักษรที่แสดงสามารถกำหนดได้ถึง 16 สี และพื้นสามารถกำหนดได้ 8 สี การกระพริบและแสงเข้มสามารถแสดงผลได้ ท่านสามารถเลือก 1 ใน 16 สีสำหรับทำขอบ จอสีจะไม่มีการขีดเส้นใต้ การขีดทับสำหรับขีดเส้นใต้จะเป็นการเลือกสีน้ำเงินในการแสดงตัวอักษรของสีน้ำเงินบนพื้นดำ

attribute byte ที่ใช้ในจอสีก็เหมือนกับการใช้งานจอโมโนโครม (BW) มีสีพื้นฐาน 3 สี คือ แดง เขียว น้ำเงิน ท่านสามารถรวมสีเหล่านี้ในส่วนของ attribute byte ได้ 8 สี รวมขาวดำ และสามารถขีดความเข้มได้ 16 สี

COLOR	I	R	G	B	COLOR	I	R	G	B
black	0	0	0	0	gray	1	0	0	0
blue	0	0	0	1	light blue	1	0	0	1
green	0	0	1	0	light green	1	0	1	0
cyan	0	0	1	1	light cyan	1	0	1	1
red	0	1	0	0	light red	1	1	0	0
magenta	0	1	0	1	light magenta	1	1	0	1
brown	0	1	1	0	yellow	1	1	1	0
white	0	1	1	1	high-intensity	1	1	1	1
					white				



ATTRIBUTE	Background				foreground				HEX
	7	6	5	4	3	2	1	0	
	BL	R	G	B	I	R	G	B	
black on black	0	0	0	0	0	0	0	0	00
blue on black	0	0	0	0	0	0	0	0	01
red on blue	0	0	0	1	0	0	0	0	14
cyan on green	0	0	1	0	0	0	1	1	23
light magenta on white	0	1	1	1	1	1	0	1	7D
gray on green, blinking	1	0	1	0	1	0	0	0	A8

ท่านสามารถใช้ INT 11h ในการตรวจสอบชนิดของมอนิเตอร์ ถ้าเป็นโรมานครมาซ์ 07 ในการเซ็ท normal attribute และจอสีท่านสามารถใช้สีใดสีหนึ่งก็ได้ ระบบสีที่ใช้ใน text mode จะสนับสนุนจอภาพในส่วนของ page 0-3 ขณะที่ 0 คือ normal screen

ตัวอย่างต่อไปนี้ใช้ INT 10h การบริการ 09 เป็นการแสดง asterisks 5 ตัว สีเขียว อ่อนมนสีม่วงและการกระพริบ

```

MOV     AH,09           ; request display
MOV     AL,'*'         ; asterisk
MOV     BH,00          ; page number 0
MOV     BL,0DAH        ; color attribute
MOV     CX,05          ; five times
INT     10H            : call BIOS

```

#### BIOS INTERRUPT 10H FOR TEXT MODE

INT 10h ซึ่งสามารถนำมาใช้ในการเซ็ทโหมดได้ง่าย การทำงานของอินเทอร์รัพท์ชนิดนี้ เพียงแต่เซ็ทค่าของรหัสบริการไว้ใน AH ตามฟังก์ชันที่ต้องการของอินเทอร์รัพท์ การอินเทอร์รัพท์จะต้องเตรียมข้อมูลของ BX,CX,DX,DI,SI and BP

## INT 10H SERVICE 01: SET CURSOR SIZE

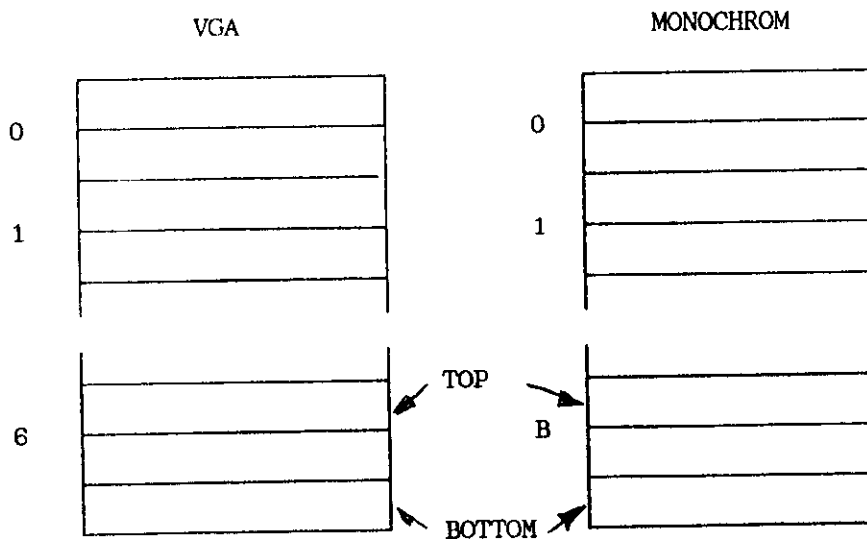
cursor ไม่ใช้รหัส ASCII และใช้ใน text mode ตามปกติแล้วสัญลักษณ์ของ cursor จะเหมือนกับ \_ (underline or break) ท่านสามารถให้คำสั่ง INT 10h ในการปรับขนาดของ cursor ในแนวตั้ง โดยการเซตค่าของ CH (บิต 4-0) สำหรับยอดของ cursor (start scan line) และเซตค่า CL (บิต 4-0) สำหรับส่วนล่างของ cursor (end scan line) ท่านสามารถปรับขนาดของ cursor ระหว่างยอด (top) และส่วนล่าง (bottom) ในช่วง 0:13 สำหรับจอโมโนโครม และ enhanced graphics และช่วง 0:7 สำหรับจอสี ดังตัวอย่างของจอโมโนโครมและ EGA

```

mov    ah,01          ; request set cursor size
mov    ch,00          ; start scan line
mov    cl,13          ; end scan line
int    10h            ; call BIOS

```

หลังจากโปรแกรมนี้ทำงาน cursor จะกระพริบและเป็นแท่งสี่เหลี่ยม ท่านสามารถปรับขนาดได้ทุกๆขนาด ระหว่างรูปร่างของมัน เช่น 04:08,03:10 และอื่นๆตามต้องการ การเปลี่ยนรูปร่างของ cursor จะต้องตรวจสอบดูว่า adapter ของจอภาพเป็นชนิดอะไร การให้ 12:13 (monochrome or EGA) หรือ 6:7 (CGA) เป็นการเซต cursor ให้ใช้ปกติ ท่านไม่แน่ใจว่ามอนิเตอร์ของท่านจะมี cursor อย่างไร ให้ใช้ execute โปรแกรมโดยใช้การบริการ 03 ภายใต้อุปกรณ์ debug ตรวจสอบ



การซ่อนเคอร์เซอร์ (Cursor Invisible) ซึ่งอาจจะทำได้ 2 ทางคือ Starting scan line คือเซตบิตที่ 5 ของ CH ไม่ว่าจะ เป็น Monochrom หรือ Color โดยการซ่อนและเรียกกลับมาซ้ำใหม่ดังนี้

```

MOV AH,3           ;GET CURRENT CURSOR
MOV BH,0           ;VIDEO PAGE 0
INT 10H           ;CALL BIOS
MOV SAVECURSOR,CX ;SAVE ORIGINAL CURSOR LINES
MOV AH,1           ;SET CURSOR LINE
OR CH,00100000B   ;SET BIT 5 IS INVISIBLE
INT 10H

```

เราสามารถเรียกเคอร์เซอร์กลับมาแสดงผลได้โดยวิธี

```

MOV AH,1           ;SET CURSOR LINE
MOV CX,SAVECURSOR ;ORIGINAL VALUES
INT 10H           ;CALL BIOS

```

ส่วนอีกวิธีหนึ่งคือการเซตเคอร์เซอร์ไปที่ตำแหน่งจุดสุดท้ายของจอภาพ ใช้ฟังก์ชัน 2 Row 0-24 โดยการเซตเคอร์เซอร์ให้มากกว่า 24 บรรทัด ทำให้มองไม่เห็นเคอร์เซอร์

#### INT 10H SERVICE 02 : SET CURSOR POSITION

การใช้บริการชนิดนี้ เราสามารถเซต cursor ได้ทุกตำแหน่งในจอภาพ ตามบรรทัดและคอลัมน์ที่กำหนด page ปกติจะใช้ page 0 แต่สามารถกำหนดได้ 0-3 สำหรับ 80 คอลัมน์ใน text mode ตำแหน่ง cursor ของแต่ละ page จะเป็นอิสระจาก page อื่นๆ การบริการจะโหลดค่า 02 ใน AH ค่าของpageจะอยู่ในBH ตำแหน่งของบรรทัดและคอลัมน์อยู่ใน DX

```

mov    ah,02          ; request set cursor
mov    bh,00          ; page number 0
mov    dh,row         ; row
mov    dl,col         ; column
int    10h           ; call BIOS

```

### INT 10H SERVICE 03 : READ CURSOR POSITION

โปรแกรมสามารถใช้ในการค้นหาตำแหน่งของ cursor ว่าปัจจุบันอยู่ในบรรทัดและคอลัมน์ใด และมีขนาดเท่าไร จากนั้นจะหาว่าอะโปรแกรมทำงานใช้ page ใดอยู่ แล้ว reset page นั้น แล้วจึง set page ที่ระบุใน BX มาใช้แทน

```

mov    AH,03          ;REQUEST CURSOR LOCATION
mov    BH,00          ;SET PAGE NUMBER 0 (NORMAL)
int    10H           ;CALL BIOS

```

หลังจากโปรแกรมทำงาน CH จะเก็บค่า START SCANLINE ของ CURSOR CL เก็บค่า END SCANLINE และ DH เก็บ ROW ,DL เก็บ COLUMN.

### INT 10H SERVICE 05 : SELECT ACTIVE PAGE

สำหรับการทำงานของ TEXT MODE 0-3 และ 13-16 การบริการนี้จะเป็นการ set page ที่จะแสดงผล PAGE อาจจะเป็น 0-3 ในโหมด 80 column หรือ 0-7 ใน โหมด 40 column

```

mov    ah.05          ;request active page
mov    al,page#       ;page number
int    10h           ;call bios

```

ตัวอย่าง การเปลี่ยน VIDEO PAGES

```
TITLE    VIDEO PAGES EXAMPLE
;THIS PROGRAM SWITCHES BACK AND FORTH BETWEEN
;VIDEO PAGES 0 AND 1 ON CGA OR EGA DISPLAY.
DOSSEG

.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC
MOV AX,@DATA           ;INITIALIZE DS REGISTER
MOV DS,AX
MOV AH,9               ;DISPLAY MESSAGE
MOV DX,OFFSET PAGE0
INT 21H
MOV AH,1               ;GET A KEYBOARD STROKE
INT 21H

TO_PAGE_1:
MOV AH,5               ;SET VIDEO PAGE
MOV AH,1               ;TO PAGE 1
INT 10H
MOV AH,9
MOV DX,OFFSET PAGE1
INT 21H
MOV AH,1
INT 21H

TO_PAGE_0:
MOV AH,5               ;SET VIDEO PAGE
MOV AL,0
INT 10H
MOV AX,4C00H           ;RETURN TO DOS
INT 21H
```

```

MAIN      ENDP
          .DATA
PAGE0     DB  'THIS IS VIDEO PAGE ZERO.$'
PAGE1     DB  'THIS IS VIDEO PAGE ONE.$'
          END  MAIN

```

### INT 10H SERVICE 06 : SCROLL UP SCREEN

ท่านสามารถใช้บริการนี้กับบทที่ 8 ในการเคลียร์จอภาพสำหรับ text mode โดยการเซตค่า AL เป็น 00 เป็นการเลื่อนจอภาพขึ้นไป 1 หน้า ผลของการเคลียร์จะทำให้จอภาพว่าง บรรทัดสูงสุดเป็นของ scroll off และบรรทัดว่างจะเกิดอยู่ข้างล่าง

ถ้าค่าใน AL ไม่ใช่ค่า 0 ดังนั้นจะเป็นจำนวนของบรรทัดที่จะ scroll up โปรแกรมจะแสดงผลในส่วนล่างของจอภาพ บรรทัดต่อไปก็จะเลื่อนขึ้นไปแทนที่บรรทัดบนสุด แต่ถ้าการอินเทอร์รัพท์กำหนดคอลัมน์ 0 ก็เป็นบรรทัดใหม่ บัญชาการ scroll ของจอภาพคือการโหลดจำนวนบรรทัดใน AL ,attribute ใน BH บรรทัดเริ่มต้นและคอลัมน์ใน CX บรรทัดสิ้นสุดและคอลัมน์ใน DX

```

mov     ax,0601h      ; scroll up one line
mov     bh,30h        ; black on cyan
mov     cx,0000       ; from 00,00
mov     dx,184Fh      ; to 24,79(full screen)
int     10h           ; call BIOS

```

มาตรฐานของจุดมุ่งหมายการเลื่อนขึ้น 1 บรรทัด

1. กำหนดรายการของ row เซ็ตที่ 0 สำหรับการกำหนดตำแหน่งบรรทัดของ cursor
2. แสดงผลของบรรทัดและบรรทัดล่องหน้าของ cursor
3. ตรวจสอบถ้า row 1 กลัจุดล่างของจอภาพ (CMP ROW,22)
4. ถ้าไม่ใช่ เพิ่ม row ทีละ 1 และ exit
5. ถ้าใช่ เลื่อนขึ้น 1 บรรทัด 1x row ในการเซต cursor และเคลียร์ row 1x 00 รีจิสเตอร์ cx และ dx จะเลื่อนได้ทุกๆจุดของจอภาพ แต่จะต้องสัมพันธ์กับ AL ซึ่งเป็นระยะของ CX:DX ที่อ้างถึงในแต่ละส่วนของจอภาพ คำสั่งต่อไปนี้จะเลื่อน 5 บรรทัด ในการสร้าง window ที่จุดกลางของจอภาพ

```

mov    ax,0605      ; scroll five lines
mov    bh,61h       ; blue on brown
mov    cx,0A1Ch     ; from row 10,column 28
mov    dx,0E34h     ; to row 14,column 52
int    10h          ; call BIOS

```

REGISTER	CONTAINS
AH	6 TO SCROLL UP OR 7 TO SCROLL DOWN
AL	NUMBER OF LINES (0=ALL)
CH,CL	ROW AND COLUMN OF UPPER LEFT WINDOW CORNOR
DH,DL	ROW AND COLUMN OF THE LOWER RIGHT WINDOW CORNER
BH	VIDEO ATTRIBUTE GIVEN TO EACH BLANK LINE

### ตัวอย่าง 9.3

```

MOV    AH,6        ;SCROLL WINDOW UP
MOV    AL,0        ;ENTRIE WINDOW
MOV    CH,0        ;UPPER LEFT ROW
MOV    CL,0        ;UPPER LEFT COLUMN
MOV    DH,24       ;LOWER RIGHT ROW
MOV    DL,79       ;LOWER RIGHT COLUMN
MOV    BH,7        ;NORMAL ATTRIBUTE
INT    10H         ;CALL BIOS

```

### INT 10H SERVICE 07 : SCROLL DOWN SCREEN

สำหรับการทำงานของ text mode การเลื่อนลงของจอภาพและเว้นที่ว่างเกิดขึ้นที่ส่วนบน โดยการใช้ AH=07 และใช้ AL,BH,CX and DX เหมือนกับการบริการ 06, scroll up

### INT 10H SERVICE 08 : READ ATTRIBUTE/CHARACTER AT CURSOR POSITION

การบริการ 08 นี้ จะทำการอ่านตัวอักษรและ attribute จากพื้นที่แสดงผลในส่วนของ text and graphics, page 0 แต่สามารถอ่านได้ page 0-3 สำหรับ 80 คอลัมน์ของ text mode การไหลคค่าของ page จะอยู่ใน BH ดังตัวอย่าง

```
mov    ah,08        ; read attribute/character
mov    bh,00        ; page number 0
int    10h          ; call BIOS
```

การทำงานนี้ค่าของตัวอักษรอยู่ใน AL และค่า attribute อยู่ใน AH ในส่วน graphics การทำงานค่า 00h สำหรับ on ASCII แต่การอ่านตัวอักษรแต่ละตัวในหนึ่งเวลา ทำมึรหัส การวนรอบในการอ่านจะพบตัวอักษร

### INT 10H SERVICE 09 : DISPLAY ATTRIBUTE/CHARACTER AT CURSOR POSITION

การทำงานของบริการ 09 เป็นการแสดงตัวอักษรในโหมด text and graphics พร้อมกับการกระพริบ (blinking), reverse และอื่นๆ ข้อมูลใน AL เป็นตัวอักษรที่แสดงผลใน ทุกๆเวลา BH จะมีค่าของ page และ BL เป็นค่าของ attribute ส่วนค่าใน CX เป็นค่าของจำนวนครั้งที่แสดงโดยเอาข้อมูล AL แสดงผล ดังนี้

```
mov    ah,09        ; request display
mov    al,char      ; character to display
mov    bh,page#     ; page number
mov    bl,attribute ; attribute(text) or color(graphics)
mov    cx,repettition ; number of repeated character
int    10h          ; call BIOS
```

การทำงานปรแกรมนี้ cursor จะไม่เลื่อน หรือตอบสนองต่อรหัส bell , cr , lf or tab ดังตัวอย่างที่แสดงรูปภาพที่เป็น reverse และกระพริบ



```

mov    ah,09          ; request display
mov    al,03h        ; heart
mov    bh,00         ; page number 0
mov    bl,0F0h       ; blink reverse video
mov    cx,05         ; five times
int    10h           ; call BIOS

```

กรณีของ text จะแสดงตัวอักษรโดยอัตโนมัติ จากบรรทัดหนึ่งไปยังบรรทัดต่อไป ข้อมูลใน CX เป็น loop ในการแสดงอักษรแต่ละเวลา จากในหน่วยความจำไว้ที่ AL เมื่อมีการแสดงตัวอักษรแต่ละตัว ท่านจะใช้การบริการ 02 16 ที่ cursor เลื่อนไป 1 ตำแหน่ง

สำหรับ graphics จะใช้ค่าใน BL กำหนดสีของตัวอักษร ถ้าบิต 7 เป็น 0 เป็นการกำหนดสี ถ้าบิต 7 เป็น 1 กำหนดสีที่นำมาผสมกับตัวมัน

#### INT 10H SERVICE 0Ah ; DISPLAY CHARACTER AT CURSOR POSITION

การทำงานของอินเทอร์รัทซ์ชนิดนี้ ใช้แสดงตัวอักษรใน text or graphics ความแตกต่างระหว่าง 0Ah และ 09h ใน text คือ การบริการ 0Ah ใช้ในลักษณะ attribute ปัจจุบัน

```

mov    ah,0Ah        ; request display
mov    al,char       ; character to display
mov    bh,page#     ; page number
mov    cx,repettition ; number of repeated characters
int    10h           ; call BIOS

```

#### INT 10H SERVICE 0Eh ; WRITE TELETYPE

การทำงานของอินเทอร์รัทซ์ชนิดนี้ จะใช้การแสดงผลที่เทอร์มินอลโดยการเซตค่า AH=0Eh ตัวอักษรที่แสดงอยู่ใน AL page อยู่ใน BH และสีตัวอักษรอยู่ใน BL

```

mov    ah,0Eh      ; request display
mov    al,char     ; character to display
mov    bh,page#    ; page number
mov    bl,color    ; foreground color
int    10h         ; call BIOS

```

#### INT 10H SERVICE 0FH : GET CURRENT VIDEO MODE

การทำงานของอินเทอร์รัทซ์ชนิดนี้ จะเป็นการย้อนกลับโหมดเดิม(ดูจากการบริการ 00) โดยป้อนข้อมูลใน AL ตัวอักษรต่อบรรทัดใน AH และจำนวน page ใน BH

#### INT 10H SERVICE 11H : CHARACTER GENERATOR

สำหรับจอภาพ EGA,MCGA and VGA การทำงานนี้จะเป็นการบริการเซ็ตสภาวะโหมดและรีเซ็ต video environment

#### INT 10H SERVICE 12H : SELECT ALTERNATE SCREEN ROUTINE

การบริการนี้จะสนับสนุน EGA and VGA การนำข่าวสารของ EGA โดยการโหลด 10H ลงใน BL ดังนี้

BH	00h	for color and 01h for monochrome
BL	00h	for 64K,01h for 128K,02h for 192K and 03h for 256K
CH		adapter bits
CL		switch setting

#### INT 10H SERVICE 13H : DISPLAY CHARACTER STRING

สำหรับ EGA และ VGA การทำงานในการแสดงชุดสตริง การเซ็ต attribute การเคลื่อนย้าย cursor และการทำงานของ backspace bell CR LF

```

mov     ah,13h      ; request display
mov     al,function ; 0,1,2 or 3
mov     bh,page#    ; page number
mov     bl,attribute ; screen attribute
lea     bp,address  ; address of string in ES:BP
mov     cx,length   ; length of string
mov     dx,screen   ; relative starting location on screen
int     10h         ; call BIOS

```

ฟังก์ชัน 4 ใน AL คือ

- 0 แสดง attribute และสตริง ไม่เลื่อน cursor
- 1 แสดง attribute และสตริง เลื่อน cursor
- 2 แสดงตัวอักษร และ attribute ไม่เลื่อน cursor
- 3 แสดงตัวอักษร และ attribute เลื่อน cursor

### USING BIOS TO DISPLAY THE ASCII CHARACTER SET

ตัวอย่าง 8-1 ใช้ DOS แสดงผล ASCII ในตัวอย่าง 9-5 แสดงการใช้ BIOS ในการแสดงผล ASCII ดังนี้

- 08H อ่าน attribute ที่ตำแหน่งของ cursor สำหรับการใช้บริการ 06H
- 06H scroll up จอภาพ เพื่อการเคลียร์จอภาพ โดยการอ่าน attribute และสร้าง window 16 บรรทัด กับตัวหนังสือสีน้ำตาลและพื้นสีน้ำเงิน
- 02H set cursor เริ่มแรก และแสดงผลเลื่อนไปเรื่อยๆ
- 0AH แสดงตัวอักษรแต่ละตัว ณ จุด cursor รวมทั้งตัวอักษรควบคุม

```

TITLE      BIOASC (COM) Use BIOS to display ASCII character set
.MODEL     SMALL
.CODE
ORG        100H
BEGIN:     JMP          SHORT MAIN
CTR        DB          00          ;Counter for ASCII characters

```

```

COL      DB      24      ;Column of screen
ROW      DB      04      ;Row of screen
;
;      Main procedure:
;      -----
MAIN     PROC      NEAR
        CALL      B10CLR      ;Clear screen
A20:    CALL      C10SET      ;Set cursor
        CALL      D10DISP     ;Display characters
        CMP       CTR,0FFH    ;Last character displayed?
        JE        A30         ; yes -- exit
        INC       CTR         ;Increment ASCII counter
        ADD       COL,02      ;Increment column
        CMP       COL,56      ;At end of column?
        JNE       A20         ; no -- bypass
        INC       ROW         ; yes - increment row
        MOV       COL,24      ; and reset column
        JMP       A20
A30:    MOV       AH,4CH      ;Exit to DOS
        INT       21H
MAIN     ENDP
;
;      Clear screen and create window:
;      -----
B10CLR   PROC      NEAR
        MOV       AH,08      ;Request get current
        INT       10H        ; attribute in AH
        MOV       BH,AH      ;Move it to BH
        MOV       AX,0600H   ;Scroll whole screen
        MOV       CX,0000    ;Upper left location
        MOV       DX,184FH   ;Lower right location
        INT       10H
        MOV       AX,0610H   ;Create 16-line window
        MOV       BH,16H     ;Brown on blue

```

```

                MOV        CX,0418H           ;Upper left corner 04:24
                MOV        DX,1336H           ;Lower right corner 19:54
                INT        10H
                RET
B10CLR          ENDP
;              Set cursor to row and column:
;              -----
C10SET          PROC    NEAR
                MOV        AH,02             ;Request set cursor
                MOV        BH,00             ;Page 0 (normal)
                MOV        DH,ROW           ;New row
                MOV        DL,COL           ;New column
                INT        10H
                RET
C10SET          ENDP
;              Display ASCII characters:
;              -----
D10DISP        PROC    NEAR
                MOV        AH,0AH           ;Display
                MOV        AL,CTR           ;ASCII char
                MOV        BH,00             ;Page 0
                MOV        CX,01             ;One character
                INT        10H
                RET
D10DISP        ENDP
                END          BEGIN

```

จากโปรแกรมตัวอักษรที่แสดงในคอลัมน์ 16 บรรทัด 16 โปรแกรมนี้แสดงการทำงานอย่างชัดเจนมากกว่ามีประสิทธิภาพ ท่านสามารถพัฒนาโปรแกรมให้มีประสิทธิภาพได้ เช่น การใช้รีจิสเตอร์เก็บค่าบรรทัด, คอลัมน์ และตัวอักษร ASCII คำสั่ง INT 10h จะเปลี่ยนค่าใน AX อย่างเดียว ค่าในรีจิสเตอร์อื่นไม่เปลี่ยนแปลง

การสาธิตการใช้การบริการของ INT 10H ในการเขียนโปรแกรมเพื่อให้งานตามขั้นตอนที่กำหนดดังต่อไปนี้

1. เซ็ตจอภาพให้อยู่ในโหมด 3 (80 x 25) 16 Color text
2. เคล็จอภาพ Upper left corner = 26x8 และ Lower right corner = 52x16
3. เลื่อนเคอร์เซอร์ไปที่คอลัมน์ 39 บรรทัดที่ 12
4. พิมพ์อักษร 'A' สี ม่วง ที่ตำแหน่งเคอร์เซอร์กำหนด

สามารถเขียนโปรแกรมได้ดังนี้

```
.MODEL    SMALL
.STACK   100H
.CODE

MAIN     PROC
; SET MODE 80 X 25
        MOV  AH,0      ;SELECT MODE FUNCTION
        MOV  AL,3      ;80X25 COLOR TEXT
        INT  10H      ;SELECT MODE

;CLEAR WINDOW TO RED
        MOV  AH,6      ;SCROLL UP FUNCTION
        MOV  CX,081AH  ;UPPER LEFT CONNER (1AH,08H)
        MOV  DX,1034H  ;LOWER RIGHT CONNER (34H,10H)
        MOV  BH,43H    ;CYAN CHAR AND RED GROUND
        MOV  AL,0      ;SCROLL ALL LINE
        INT  10H      ;CLEAR WINDOW

;MOVE CURSOR
        MOV  AH,2      ;MOVE CURSOR FUNCTION
        MOV  DX,0C21H  ;CENTER OF SCREEN
        XOR  BH,BH     ;PAGE 0
        INT  10H      ;MOVE CURSOR
```

```

;DISPLAY CHAR WITH ATTRIBUTE
        MOV  AH,9      ;DISPLAY CHAR FUNCTION
        MOV  BH,0      ;PAGE 0
        MOV  BL,0C3H   ;BLINKING CYAN CHAR , RED BACK
        MOV  CX,1      ;DISPLAY ONR CHAR
        MOV  AL,'A'    ;CHARACTER IS 'A'
        INT  10H

;DOS EXIT
        MOV  AH,4CH
        INT  21H

MAIN    ENDP
        END  MAIN

```

### BLINKING REVERSE VIDEO AND SCROLLING

โปรแกรมในรูป 9-3 เป็นการรับชื่อจากคีย์บอร์ดและแสดงผลที่จอภาพ โปรแกรมนี้จะแสดงผลต่อจาก prompt ซึ่งเป็น reverse video (blue on white) รับชื่อเป็นปกติ (white on blue) และแสดงผลที่คอลัมน์ 40 ในบรรทัดเดียวกันในลักษณะ reverse and blinking ดังนี้

```

        Name?Francis Bacon   Francis Bacon(blinking)
        |                    |
        |                    |
column 0                    column 40

```

### EXTENDED ASCII CHARACTERS

ตัวอักษร ASCII 128-255(80H-FFH) เป็นจำนวนของตัวอักษรพิเศษที่ใช้แสดง prompt menus and logos สำหรับตัวอย่างที่ใช้ตัวอักษรเหล่านี้วาดสี่เหลี่ยมและเส้นตรง

HEX	CHARACTER
DA	top left corner angle
BF	top right corner angle
C0	bottom left corner angle
D9	bottom right corner angle
C4	solid horizontal line
B3	solid vertical line

page 60,132

TITLE NMSCROLL (EXE) Reverse video, blinking, scrolling

.MODEL SMALL

.STACK 64

-----

.DATA

NAMEPAR	LABEL	BYTE	
			;Name parameter list:
MAXNLEN	DB	20	; maximum length of name
ACTNLEN	DB	?	; no. of chars entered
NAMEFLD	DB	20 DUP(' ')	; name
COL	DB	00	
COUNT	DB	?	
PROMPT	DB	'Name? '	
ROW	DB	00	

-----



```

        .CODE
BEGIN   PROC   FAR
        MOV     AX,@data
        MOV     DS,AX
        MOV     ES,AX
        MOV     AX,0600H
        CALL    Q10SCR                ;Clear screen
A20LOOP:
        MOV     COL,00                ;Set column to 0
        CALL    Q20CURS
        CALL    B10PRMP                ;Display prompt
        CALL    D10INPT                ;Provide for input of name
        CMP     ACTNLEN,00            ;No name? (indicates end)
        JNE     A30
        MOV     AX,0600H
        CALL    Q10SCR                ;If so, clear screen,
        MOV     AH,4CH                ; and terminate
        INT     21H
A30:
        CALL    E10NAME                ;Display name
        JMP     A20LOOP
BEGIN   ENDP
:Display prompt:
;-----
B10PRMP PROC   NEAR
        LEA     SI,PROMPT            ;Set address of prompt
        MOV     COUNT,05
B20:
        MOV     BL,71H                ;Reverse video
        CALL    F10DISP                ;Display routine

```

```

                INC        SI            ;Next character in name
                INC        COL          ;Next column
                CALL     Q20CURS
                DEC        COUNT        ;Countdown
                JNZ       B20           ;Loop n times
                RET

B10PRMP        ENDP
;Accept input of name:
;-----
D10INPT        PROC    NEAR
                MOV        AH,0AH      ;Request input
                LEA        DX,NAMEPAR
                INT        21H
                RET
D10INPT        ENDP
;Display name with blinking reverse video:
;-----
E10NAME        PROC    NEAR
                LEA        SI,NAMEFLD  ;Initialize name
                MOV        COL,40      ;Set screen column
E20:
                CALL     Q20CURS      ;Set cursor
                MOV        BL,0F1H     ;Blink reverse video
                CALL     F10DISP      ;Display routine
                INC        SI          ;Next character in name
                INC        COL        ;Next screen column
                DEC        ACTNLEN     ;Countdown name length
                JNZ       E20         ;Loop n times
                CMP        ROW,20     ;Near bottom screen?
                JAE        E30
                INC        ROW        ; no - increment row
                RET

```

E30:

```
        MOV          AX,0601H          ; yes -
        CALL   Q10SCR          ; scroll screen
        RET
```

E10NAME ENDP

;Display character:

;-----

```
F10DISP      PROC   NEAR                                ;BL (attribute) set on entry
        MOV          AH,09                                ;Request display
        MOV          AL,[SI]                             ;Get name character
        MOV          BH,00                                ;Page number
        MOV          CX,01                                ;One character
        INT          10H                                 ;Call BIOS
        RET
```

F10DISP ENDP

;Scroll screen:

;-----

```
Q10SCR       PROC   NEAR                                ;AX set on entry
        MOV          BH,17H                                ;White on blue
        MOV          CX,0000
        MOV          DX,184FH
        INT          10H                                 ;Call BIOS
        RET
```

Q10SCR ENDP

;Set cursor row/col:

;-----

```

Q20CURS    PROC    NEAR
            MOV     AH,02
            MOV     BH,00
            MOV     DH,ROW      ;Row
            MOV     DL,COL      ;Column
            INT     10H
            RET
Q20CURS    ENDP
            END     BEGIN

```

ตัวอย่าง 9.6 ต่อบนนี้ใช้ INT 10h การบริการ 09 ในการขีดเส้น 25 ครั้ง

```

mov     ah,09      ; request display
mov     al,0C4h    ; solid line
mov     bh,00      ; page number 0
mov     bl,0Fh     ; white on black,intense
mov     cx,25      ; 25 repetitions
int     10h        ; call BIOS

```

อย่าลืมว่าการบริการนี้ไม่เลื่อน cursor ตัวอย่างที่ง่ายที่สุดคือการแสดงกล่องที่กำหนดใน data segment

HEX	CHARACTER
B0	one-quarter dots on(light)
B1	one-half dots on(medium)
B2	three-quarter dots on(dark)

## DOS SERVICE 02 TO DISPLAY

ท่านสามารถใช้DOS การบริการ 02 ในการแสดงตัวอักษรทีละตัว โดยการโหลดตัวอักษรใน DL และแสดงผลที่ตำแหน่ง cursor กำหนดและใช้ INT 21h TAB,CR,LFทำงานปกติ

```
mov    ah,02          ; request display character
mov    dl,char        ; character to display
int    21h
```

## GRAPHICS MODE

ในส่วนนี้จะอธิบายในการทำงานของกราฟิก จอภาพชนิด RGB ที่ใช้แสดงจอสีแบบกราฟิก จะรับสัญญาณเข้า และส่งสัญญาณออก 3 ชนิด ที่แยกโดยเป็นอิเล็กตรอน แดง เขียวและน้ำเงิน สำหรับเป็นแม่สี

ระบบ EGA(enhanced graphics adapter) และระบบ VGA(video graphics array) จะให้ความละเอียดของจอภาพดีกว่าระบบกราฟิกแบบเดิม CGA (color graphics adapter) รายละเอียดดูได้จากตาราง 9-1

การทำงานของกราฟิกมีพื้นฐานการทำงาน 2 อย่าง คือ text and graphics จอภาพที่สร้างขึ้นมานั้นจะออกแบบให้ทำงานได้ทั้ง 2 อย่าง ปกติจะทำงาน text แต่การทำงาน graphics จะใช้ BIOS INT 10h การบริการ 00 ดังตัวอย่างต่อไปนี้

### 1. set medium graphics mode

```
mov    ah,00          ; set mode
mov    al,04          ; medium graphics
int    10h
```

### 2. set text mode

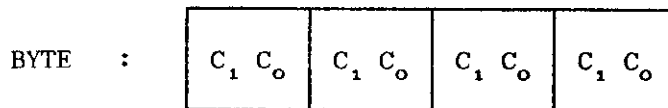
```
mov    ah,00          ; set mode
mov    al,03          ; color text
int    10h
```

โหมดกราฟิกที่ใช้ pixels ในการกำหนดสี ในระบบ CGA มีความเข้มของภาพ 3 ชนิด

1. low resolution โหมดนี้ ROM ไม่ได้สนับสนุน มันจะแสดงความเข้ม 100 บรรทัด แต่ละบรรทัด 160 pixels แต่ละ pixel จะประกอบด้วยสี 16 สี (นั่นคือ 4 บิตต่อ pixel)
2. medium resolution โหมดนี้จะแสดงความเข้ม 200 บรรทัด แต่ละบรรทัด 320 pixels แต่ละไบต์มี 4 pixels
3. high resolution โหมดนี้แบ่งออกเป็น 200 บรรทัด บรรทัดหนึ่งมี 640 pixels แต่มันต้องการหน่วยความจำ 16K-byte โหมดนี้ในระบบกราฟิกสามารถใช้ BW ได้ แต่ละไบต์จะมี 8 pixels นั่นคือ 1 บิตต่อ 1 pixels

### MEDIUM RESOLUTION MODE

ภายใต้การทำงานของโหมดนี้ แต่ละไบต์จะมี 4 pixels จำนวน 0 ถึง 3 มีดังนี้



PIXEL :            0            1            2            3

ทุกครั้งเราสามารถกำหนดสีได้ 4 สี โดยกำหนดค่า 0 ถึง 3 แต่โหมดนี้สามารถกำหนดได้ 4 สี เพราะใช้ 2 บิตต่อ pixel คือ 00,01,10 และ 11 ท่านสามารถเลือก pixel 00 สำหรับทุกๆสีใน 16 สี สำหรับสีพื้น และเลือก pixel 01,10 และ 11 สำหรับทุก 1 ใน 3 สี 2 ชุด

$C_1$ $C_0$	PALETTE 0	PALETTE 1
0   0	background	background
0   1	green	cyan
1   0	red	magenta
1   1	brown	white

การใช้ INT 10h การบริการ 0Bh ในการเลือก palette และสีพื้น เช่นถ้าท่านเลือกสีพื้นเป็นสีเหลืองใน palette 0 สีที่ท่านเลือกได้คือ เหลือง เขียว แดง และน้ำตาล โบทจะประกอบด้วย pixel คือ 10101010 จะแสดงสีแดงทั้งหมด ถ้าท่านเลือกสีพื้นมีน้ำเงินและ palette 1 สีที่ท่านเลือกได้คือ น้ำเงิน ม่วง แดง ขาว โบทจะประกอบด้วย pixel 00011011 จะแสดงสีน้ำเงิน ม่วง แดงสด และขาว

### BIOS INTERRUPT 10H FOR GRAPHICS

ในกลุ่มของ INT 10h สามารถกำหนดจอภาพเป็น text and graphics การทำงานทั้งสองเพียงแต่ให้ข้อมูลในรีจิสเตอร์ BX,CX,DX,DI,SI and BP แต่ไม่ใช้ AX จะอธิบายการทำงานแต่ละฟังก์ชันดังต่อไปนี้

#### INT 10H SERVICE 00 : SET MODE

การทำงานของ BIOS ชนิด INT 10h ที่ใช้ในส่วนของ text mode จะมีการบริการของการเซ็ทโหมดสำหรับจอสีหรือโมโนโครม ในรูปของ text and graphics การบริการ 00 ที่อยู่ใน AH และ 04 ใน AL เป็นมาตรฐานของกราฟิกจอสี

```

mov    ah,00          ; set mode for
mov    al,04          ; 320x200 resolution
int    10h

```

การเซ็ทกราฟิกเราจะมองไม่เห็น cursor

#### INT 10H SERVICE 04 : READ LIGHT PEN POSITION

การบริการชนิดนี้ใช้กับโหมดกราฟิก ที่ต้องการทราบสถานะของ light pen การทำงานจะให้ข่าวสารย้อนกลับ ดังนี้

```

AH    0 สถานะไม่มีการกระตุ้น (triggered)
      1 สถานะมีการกระตุ้น
DX    row อยู่ใน DH และ column อยู่ใน DL

```

CH/BX ตำแหน่งบรรทัดของ pixel กับ raster(horizontal) อยู่ใน CH และคอลัมน์/dot อยู่ใน BX

#### INT 10H SERVICE 08 : READ ATTRIBUTE/CHARACTER AT CURSOR POSITION

การบริการนี้สามารถอ่านได้ทั้งตัวอักษรและ attribute จากพื้นที่แสดงผลใน text และ graphics ดูจาก text mode

INT 10H SERVICE 09 : DISPLAY ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION ดูตัวอย่างจาก BIOS ใน text mode

INT 10H SERVICE 0AH : DISPLAY A CHARACTER AT CURSOR POSITION ดูตัวอย่างจาก BIOS ใน text mode

#### INT 10H SERVICE 0BH : SET CURSOR PALETTE

การให้บริการนี้เป็นการเซตสีและแสดงตัวอักษรกราฟิก ค่าที่อยู่ใน BH (00 หรือ 01) และค่าใน BL ตามจุดประสงค์ดังนี้

BH=00 (1) เลือกสีพื้นสำหรับ 320x200 ของกราฟิก

(2) เลือกขอบของสี หรือ

(3) เลือกสีที่แสดง ใน BL จะมีข้อมูลของสี (0-31)

BH=01 เลือกกลุ่มสีสำหรับ 320x200 ของกราฟิก ข้อมูลใน BL จะอยู่กลุ่มสี (0หรือ1)

ตัวอย่างในการเลือกกลุ่มสี 0 (palette:0)

```
mov    ah,0Bh        ; request color
mov    bh,01h        ; select palette
mov    bl,00         ; number 0 (green,red,brown)
int    10h           ; call BIOS
```

#### INT 10H SERVICE 0Ch : WRITE PIXEL DOT

การให้บริการ 0Ch ที่แสดงในการเลือกสี (background and palette) โดยการเซต CX สำหรับ column และ DX สำหรับ row ค่าต่ำสุดของ column and row คือ 0



และค่าสูงสุดขึ้นอยู่กับ video mode. ข้อมูลใน AL คือค่าสีของ pixel 0-3 สำหรับ medium resolution graphics และ 0-1 สำหรับ high resolution graphics

```

mov    ah,0Ch          ; request write dot
mov    al,color        ; color of pixel
mov    cx,column       ; horizontal position
mov    dx,row          ; vertical position
int    10h

```

INT 10H SERVICE 0DH : READ PIXEL DOT

การทำงานของบริการชนิดนี้ ตรงกันข้ามกับการบริการ 0Ch โดยการอ่าน dot ตามค่าของสี โดยการเซต CX สำหรับ column และ DX สำหรับ row ค่าต่ำสุดของ column และ row คือ 0 ค่าสูงสุดขึ้นอยู่กับ video mode การทำงานจะเก็บค่าสีของ pixel ใน AL

INT 10H SERVICE 0EH : WRITE TELETYPE ดูตัวอย่างจาก BIOS ใน text mode

ตัวอย่าง 9.7 จงทำสำเนา Pixel ที่ Col 50 Row 199 ไปยัง Pixel ที่ตำแหน่ง Col 20 , Row 199

วิธีทำ จะต้องมีการอ่าน Pixel ที่ Col 50 Row 199 และ เขียนใน Col 20 Row 40

```

MOV    AH,0DH          ;READ PIXEL
MOV    CX,50           ;COL 50
MOV    DX,199          ;ROE 199
INT    10H             ;AL GETS PIXEL VALUE
MOV    AH,0CH          ;WRITE PIXEL AL IS ALREADY SET
MOV    CX,20           ;COL 20
MOV    DX,40           ;ROE 40
INT    10H

```

ตัวอย่าง 9.8 จงเขียน Pixel จำนวน 256 ครั้ง ที่บรรทัด 100

วิธีทำ เลือกการทำงานโหมด 13H แล้วกำหนด LOOP ที่จะเขียนค่าใน AL จาก 0-255 ในคอลัมน์ที่ 0-255

```
                ;SET MODE
                MOV  AH,0
                MOV  AL,13H
                INT  10H

                ;DISPLAY 256 PIXEL IN ROW 100
                MOV  AH,0CH          ;WRITE PIXEL
                MOV  AL,0            ;START PIXEL
                MOV  BH,0            ;PAGE 0
                MOV  CX,0            ;COL 0
                MOV  DX,100          ;ROW 100
L1:             INT  10H            ;WRITE PIXEL
                INC  AL              ;NEXT COLOR
                INC  CX              ;NEXT COL
                CMP  CX,256
                JL   L1
```

**INT 10H SERVICE 10H : SET PALETTE REGISTERS**

```
INPUT          AH = 10H
                AL = 10H
                BX = COLOR REGISTER
                CH = GREEN
                CL = BLUE
                DH = RED
```

- การบริการชนิดนี้สำหรับระบบ EGA and VGA พังก์ชันที่อยู่ใน AL มีการทำงานดังต่อไปนี้
- 00 เป็นการเซ็ตรีจิสเตอร์ palette ซึ่งข้อมูลใน BH เป็นค่าที่เซ็ต และข้อมูลใน BL เป็นตัวเซ็ต
  - 01 เซ็ต overscan register ซึ่งค่าข้อมูล BH เป็นค่าที่เซ็ต
  - 02 การเซ็ตรีจิสเตอร์ palette ทั้งหมด และ overscan ES:DX ซึ่งตาราง 17 ไบท์ ซึ่งไบท์ 0-15 คือค่าของ palette และค่าไบท์ 16 คือค่า overscan
  - 03 toggle intensify/blinking bit ซึ่ง 00 ใน BL คือความเข้ม และ 01 กระทบ

ส่วนรหัสฟังก์ชันใน AL สำหรับ VGA คือ

- 07 (read individual palette register)
- 08 (read overscan register)
- 09 (read all palette and overscan)
- 10h (set individual color register)
- 12h (set block of color register)
- 13h (select color page)
- 15h (read individual color register)
- 17h (read block of color register)
- 1Ah (read color page state)

ตัวอย่าง 9.9 การใส่ค่าสี 30 RED , 230 GREEN , 10 BLUE ในรีจิสเตอร์สี 5

```

MOV AH,10H          ;SET COLOR REGISTER
MOV AL,10H
MOV BX,15           ;REGISTER 5
MOV DH,30          ;RED
MOV CH,20          ;GREEN
MOV CL,10          ;BLUE
INT 10H

```

INT 10H SERVICE 1AH : READ/WRITE DISPLAY ใช้กับระบบ PS/2

INT 10H SERVICE 1BH : RETURN FUNCTIONALITY/STATE INFORMATION  
ใช้กับระบบ PS/2

INT 10H SERVICE 1CH : SAVE/RESTORE VIDEO STATE ใช้กับระบบ PS/2

### SETTING GRAPHICS MODE AND DISPLAYING COLOR

โปรแกรมในรูป 9-4 ในการใช้ INT 10H และการบริการ 10H สำหรับการเซตกราฟิครวม การบริการ 0BH สำหรับการเลือกพื้นสีเขียว (BACKGROUND) และการบริการ 0CH สำหรับการเขียน PIXEL DOT สำหรับ 640x350 โปรแกรมจะเพิ่มค่าสีในแต่ละบรรทัด สีจะเปลี่ยนเป็นค่าเดิมทุกๆ 7 บรรทัด การแสดงผลจะเริ่มที่คอลัมน์ 64 จากซ้ายมือ และสิ้นสุดคอลัมน์ 64 จากขวามือ หลังจากสิ้นสุดโปรแกรมโปรแกรมจะกลับสู่ TEXT MODE

#### Determining typed of video adaptor

Video graphics adaptor จะสนับสนุนการบริการต่างๆ ท่านจะต้องเข้าในอะเดปเตเตอร์ต่างๆที่ใช้อยู่ในระบบคอมพิวเตอร์ และเขียนโปรแกรมตรวจเช็คดังนี้

##### 1. VGA is installed

```
MOV     AH,1AH           ;REQUEST VGA SERVICE
MOV     AL,0             ;AND FUNCTION 0
INT     10H
CMP     AL,1AH           ;IF AL CONTIAINS 1AH ON RETURN'
JE      VGAFFOUND       ;SYSTEM CONTAINS A VGA
```

## ตัวอย่าง 9.10

```

TITLE      GRAPHIX (COM)  Display color/graphics
.MODEL    SMALL
.CODE
ORG       100H
BEGIN     PROC    NEAR
          CALL    B10MODE      ;Set mode and palette
          CALL    C10DISP     ;Display color/graphics
          CALL    D10RSET     ;Reset mode to text
          MOV     AH,4CH      ;Exit
          INT     21H
B10MODE   PROC    NEAR
          MOV     AH,00      ;Set EGA graphics mode
          MOV     AL,10H     ;640 X 350 (04H for CGA)
          INT     10H
          MOV     AH,0BH     ;Set color palette
          MOV     BH,00      ;Background
          MOV     BL,02      ;Green
          INT     10H
          RET
B10MODE   ENDP
C10DISP   PROC    NEAR
          MOV     BX,00      ;Set initial color.
          MOV     CX,64      ; column,
          MOV     DX,70      ; and row
C20:
          MOV     AH,0CH     ;Write pixel dot
          MOV     AL,BL      ;Set color
          INT     10H        ;BX, CX, & DX are preserved
          INC     CX         ;Increment column

```

```

        CMP     CX,576           ;Column at 576?
        JNE     C20             ; no -- loop
        MOV     CX,64           ; yes - reset column
        INC     BL              ;Change color
        INC     DX              ;Increment row
        CMP     DX,140         ;Row at 140?
        JNE     C20             ; no - loop
        RET                     ; yes - terminate
C10DISP ENDP
D1ORSET PROC NEAR
        MOV     AH,00           ;Reset text mode
        MOV     AL,03
        INT     10H
        RET
D1ORSET ENDP
BEGIN   ENDP
        END     BEGIN

```

## 2. EGA IS INSTALLED

```

        MOV     AH,12H         ;REQUEST EGA SERVICES
        MOV     BL,10H         ;AMOUNT OF EGA MEMORY
        INT     10H
        CMP     BL,10H         ;IF BL NO LONGER CONTAINS 10H,
        JNE     EGAFFOUND      ;SYSTEM CONTAINS AN EGA

```

## 3. CGA , MDA IS INSTALLED

ตรวจสอบที่ตำแหน่ง 40:0063 ซึ่งเป็นแอดเดรสฐานของหน่วยความจำที่ซัควคมุม ถ้าเป็นค่าของ 3BxH หมายความว่า เป็น MDA และ 3DxH เป็น CGA

## บทสรุป

- อาร์ทิวต์ไบนารีคือค่าที่กำหนดใน TEXT MODE สำหรับโหมดโรม และ จอสี มีดังนี้ BLINKING REVERSE , HIGH INTENSITY สำหรับจอสีสามารถเลือกสีต่างได้
- BIOS INT 10H เป็นการบริการประมวลผลจอภาพ เช่น กำหนดเคอร์เซอร์
- การบริการ INT 10H ที่ใช้ในการแสดงผลสามารถกำหนดเคอร์เซอร์ลวงหน้า
- โหมดกราฟิกสี LOW RESOLUTION (ROM ไม่สนับสนุน) MEDIUM RESOLUTION (สำหรับ สีปกติ) HIGH RESOLUTION (สำหรับ BW graphics)
- PIXEL คือ PICTURE ELEMENT ประกอบไปด้วยจำนวนของบิตที่กำหนดขึ้นอยู่กับกราฟิกอะแดปเตอร์ และ RESOLUTION

## แบบฝึกหัด

- 9-1. จงเขียนอาร์ทิวต์ของจอภาพโรมในรหัสเลขฐานสองสำหรับการทำงานต่อไปนี้
  - a) Underline only
  - b) white on black
  - c) normal intensity
  - d) reverse video , intens
- 9-2. จงเขียนอาร์ทิวต์ของจอภาพสีในรหัสเลขฐานสองสำหรับการทำงานต่อไปนี้
  - a) magenta on light cyan
  - b) red on blue
  - c) brown on yellow
  - d) green on red
- 9-3. จงเขียนรหัสการทำงานต่อไปนี้
  - a) เซ็ตโหมด 80 คอลัมน์โรม
  - b) เซ็ตโหมด 40 คอลัมน์
  - c) เซ็ตขนาดเคอร์เซอร์เริ่มที่บรรทัด 5 สิ้นสุดบรรทัด 12
  - d) Scroll up screen 10 lines
- 9-4. จงเขียนคำสั่งในการแสดง "\*" จำนวน 5 บรรทัด ใช้อาร์ทิวต์ Yellow on green
- 9-5. จงเขียนโปรแกรมในการรับข้อมูลจากคีย์บอร์ด โดยการไว้ DOS 21H และการบริการ 0AH และใช้การบริการที่ 9 ในการแสดงข้อมูล โปรแกรมจะเคลียและเซ็ตเคอร์เซอร์ตามที่ท่านตัดสินใจ
- 9-6. จงเขียนคำสั่งในการเซตกราฟิกโหมดต่างๆดังต่อไปนี้
  - a) 320x200 on CGA
  - b) 640x200 on EGA
  - c) 640x200 on monochrom
  - d) 320x200 on VGA
- 9-7. จงเขียนรหัสคำสั่งในการกำหนดสีพื้นของจอสีในการทำงานของกราฟิกโหมด
- 9-8. จงเขียนโปรแกรมในการอ่าน DOT จากบรรทัดที่ 12 คอลัมน์ที่ 13

