



บทที่ 2

## ระบบคอมพิวเตอร์ของเครื่อง ไอบีเอ็มพีซี

### รายละเอียดของเครื่อง ไอบีเอ็มพีซี

เครื่อง ไอบีเอ็มพีซี เป็นเครื่องคอมพิวเตอร์ระดับ ไมโครคอมพิวเตอร์ประกอบด้วยส่วนสำคัญ 2 ส่วนคือ ส่วนที่เป็นตัวเครื่อง (System Unit) และคีย์บอร์ด นอกจากนี้ก็ยังมีส่วนประกอบอื่น ๆ ที่ผู้ใช้สามารถเลือกต่อเข้ากับเครื่องได้เช่น ส่วนที่เป็นจอภาพ เครื่องพิมพ์ เป็นต้น

ในส่วนที่เป็นตัวเครื่องจะประกอบไปด้วย หน่วยประมวลผลกลาง, รม (ROM หรือ Read Only Memory), แรม (RAM หรือ Random Access Memory), ชิป (Chip) ที่ทำหน้าที่ควบคุมฟังก์ชันต่าง ๆ ของระบบ ฯลฯ ส่วนประกอบเหล่านี้บรรจุอยู่บนแผงที่เรียกว่า แผงระบบ (System Board)

### หน่วยประมวลผลกลาง

ในเครื่อง ไอบีเอ็มพีซีใช้ไมโครโปรเซสเซอร์เบอร์ 8088 (8088 Micro Processor) เป็นหน่วยประมวลผลกลาง ไมโครโปรเซสเซอร์เบอร์ 8088 เป็นไมโครโปรเซสเซอร์ขนาด 16 บิต มีสายกำหนดตำแหน่ง (Address Bus) 20 สาย สายทางเดินข้อมูล 8 สาย สามารถกำหนดขนาดของหน่วยความจำได้ถึง 1 ล้านไบต์

### หน่วยความจำ

รวม มีขนาด 48 กิโลไบต์ในรวมของเครื่อง ไอบีเอ็มพีซี ประกอบด้วยโปรแกรมต่อไปนี้

- เบสิค อินเตอร์พรีเตอร์ (BASIC Interpreter) 32 กิโลไบต์
- โปรแกรมเช็คระบบ (Selftest Program) 2 กิโลไบต์
- ไบออส (BIOS หรือ Basic Input /Output System) 6 กิโลไบต์
- ที่ว่างซึ่งผู้ใช้สามารถเขียนโปรแกรมเพิ่มเข้าไปได้ มีขนาด 8 กิโลไบต์

แรม มีขนาดอยู่ระหว่าง 64-640 กิโลไบต์

### อุปกรณ์ช่วยในการทำงานของระบบ

#### ชิพเบอร์ 8255

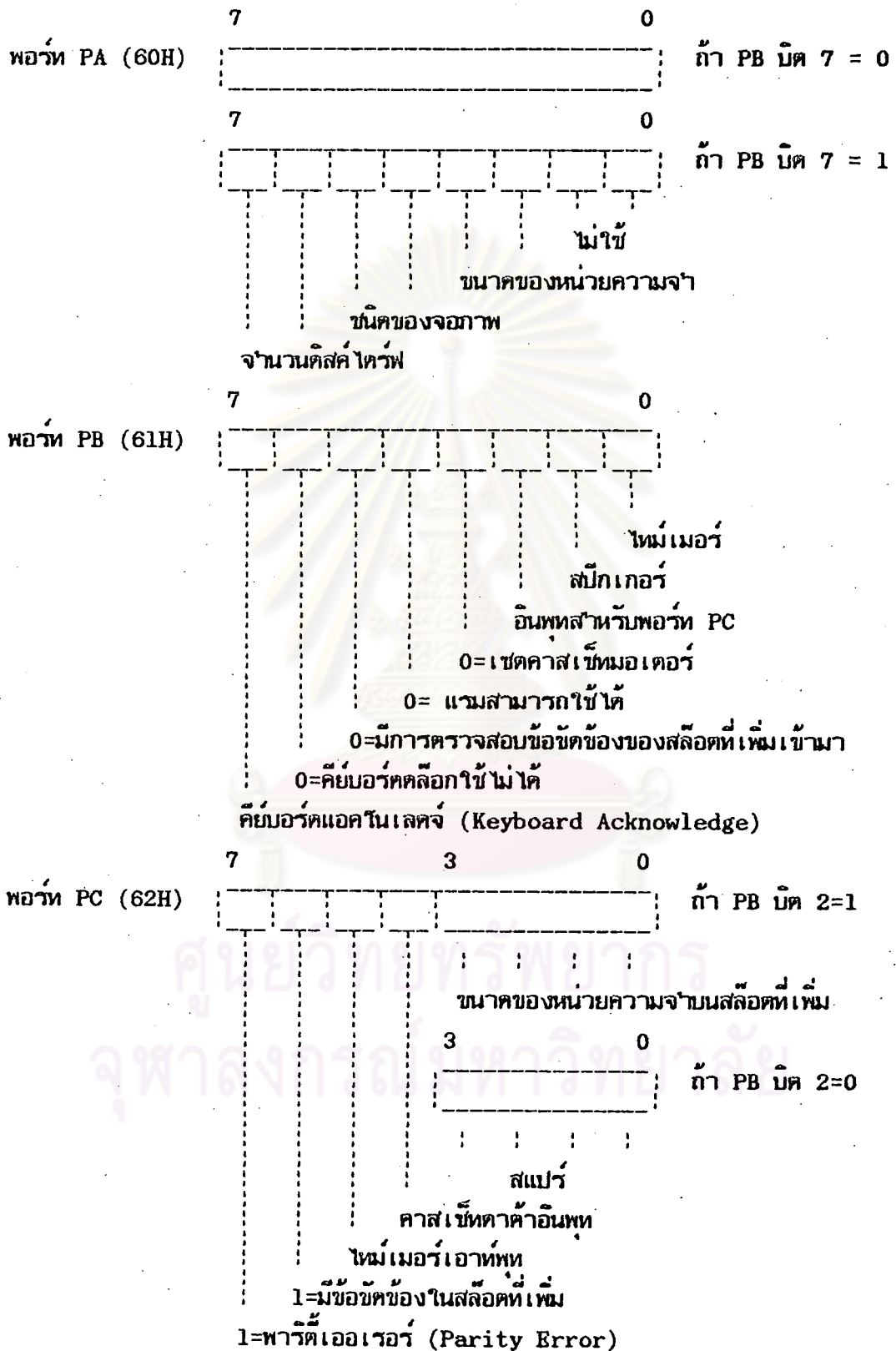
เป็นชิพที่ใช้ติดต่อกับอุปกรณ์ภายนอกบางตัวของเครื่อง อุปกรณ์ดังกล่าวคือ คีย์บอร์ด, สปีกเกอร์ (Speaker), คอนฟิกูเรชันสวิตช์ (Configuration Switch) ภายในชิพ 8255 ประกอบด้วยพอร์ต 3 พอร์ตคือ พอร์ต PA, พอร์ต PB, พอร์ต PC โดยตรงกับตำแหน่งที่ 60H, 61H, และ 62H ตามลำดับ และรีจิสเตอร์คำสั่งขนาด 1 ไบต์ซึ่งตรงกับตำแหน่งที่ 63H เมื่อเปิดเครื่องโปรแกรมในไบออสจะกำหนดค่าเริ่มต้นให้กับชิพนี้โดย พอร์ต PA และพอร์ต PC จะเป็นพอร์ตรับข้อมูล พอร์ต PB จะเป็นพอร์ตส่งข้อมูล ความหมายของแต่ละพอร์ตจะเป็นดังรูปที่ 2.1

การโปรแกรมเพื่อติดต่อกับชิพ 8255 ทำได้ดังนี้

IN	AL, 61H	; อ่านค่าปัจจุบันของพอร์ต PB
AND	AL, 7FH	; กำหนดค่าพอร์ต PB บิต 7=0
OUT	61H, AL	;
IN	AL, 60H	; AL=ค่าของสแกนโค้ด

#### ชิพเบอร์ 8259

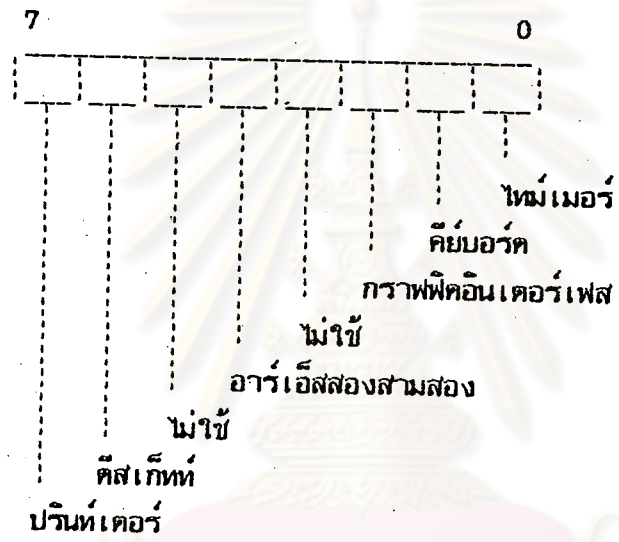
เป็นชิพทำหน้าที่ควบคุมการอินเตอรัพท์ ทั้งนี้เนื่องจากสัญญาณอินเตอรัพท์ที่ส่งมายังหน่วยประมวลผลกลางมีคัยกันมาก แต่หน่วยประมวลผลกลางสามารถรับสัญญาณอินเตอรัพท์นั้นได้เพียงทางเดียว ชิพ 8259 จะเป็นตัวจัดการรับสัญญาณอินเตอรัพท์เหล่านั้นแล้วจัดส่งไปให้หน่วยประมวลผลกลาง ชิพ 8259 สามารถรับสัญญาณอินเตอรัพท์ที่เป็นอิสระต่อกันได้ทั้งหมด 8 ทางจากอุปกรณ์ 8 ชนิด แต่จะจัดส่งไปให้หน่วยประมวลผลกลาง 1 ชนิดในแต่ละครั้ง วิธีการเลือกจะกำหนดค่าตามลำดับความสำคัญ ให้กับอินเตอรัพท์ที่ได้รับ โดยไทม์เมอร์อินเตอรัพท์ (Timer Interrupt) จะได้รับค่าลำดับความสำคัญสูงสุด รองลงมาตามลำดับคือสัญญาณอินเตอรัพท์จากคีย์บอร์ด, กราฟฟิคอินเตอรัพส์ (Graphics Interface), อาร์เอสสองสามสอง (RS232), หน่วยบันทึกข้อมูล และเครื่องพิมพ์เป็นลำดับสุดท้าย หลังจากหน่วยประมวลผลกลางทำงานตอบสนอง



รูปที่ 2.1 8255 พอร์ท

อินเทอร์พท์ที่ได้รับแล้ว จะส่งสัญญาณเสร็จสิ้นการอินเทอร์พท์ (EOI) มายังพอร์ท 8259 เพื่อให้จัดส่งสัญญาณอินเทอร์พท์ถัดไป

ในการโปรแกรมเพื่อติดต่อกับพอร์ท 8259 สามารถกระทำได้โดยผ่านพอร์ท 21H ซึ่งเป็นอินเทอร์พท์มาสเตอร์รีจิสเตอร์ (Interrupt mask register) ขนาด 1 ไบต์ โดยการกำหนดค่าให้กับบิตใดบิตหนึ่งของรีจิสเตอร์นั้น และส่งสัญญาณเสร็จสิ้นการอินเทอร์พท์มายังพอร์ท 20H



รูปที่ 2.2 อินเทอร์พท์มาสเตอร์รีจิสเตอร์

เอ็มเอสคอส

เอ็มเอสคอส เป็นโปรแกรมควบคุมระบบงานที่ใช้กับเครื่องไอบีเอ็มพีซี เอ็มเอสคอสสามารถแบ่งออกเป็น 4 ส่วนคือ

1. บูทเรคคอร์ด (Boot Record) เป็นส่วนของโปรแกรมที่จะอ่านส่วนอื่นๆ เข้ามาในระบบ
2. โปรแกรมไอบีเอ็มไบโอ (IBMBIO) เป็นโปรแกรมที่เรียกใช้โปรแกรมในไบออสเพื่อควบคุมการทำงานของอุปกรณ์ของระบบ
3. โปรแกรมไอบีเอ็มคอส (IBMDOS) เป็นโปรแกรมติดต่อกับโปรแกรมใช้งานทั่วไป จะจัดการเกี่ยวกับการจัดไฟล์ การจัดข้อมูลเพื่อเตรียมพร้อมให้โปรแกรมไอบีเอ็มไบโอสามารถนำมาใช้

งานได้ โปรแกรมส่วนนี้ได้แก่โปรแกรมคอสฟังก์ชันคอลล์ (DOS Functions Call) โปรแกรมคอสอินเตอร์รัพ (DOS Interrupt)

4. โปรแกรมคอมมานด์ (COMMAND) ทำหน้าที่รับคำสั่งจากภายนอก แปลงคำสั่งนั้นแล้วเรียกใช้โปรแกรมไอบีเอ็มคอสเพื่อจัดการกับคำสั่งดังกล่าว โปรแกรมคอมมานด์แบ่งออกเป็น 4 ส่วนคือ

4.1 ส่วนที่ประจำอยู่ในหน่วยความจำตลอดเวลา เป็นส่วนที่อยู่ให้ส่วนของโปรแกรมไอบีเอ็มคอส มีส่วนของโปรแกรมที่ตอบสนองการเกิดอินเตอร์รัพ 22H อินเตอร์รัพ 23H และอินเตอร์รัพ 24H ดูแลการเกิดข้อขัดข้อง และส่งข้อความแสดงข้อขัดข้องนั้นทางจอภาพ นอกจากนี้ยังเป็นตัวโหลด (Load) ส่วนชั่วคราว ภาที่ส่วนชั่วคราวถูกโปรแกรมใช้งานทับ

4.2 ส่วนเริ่มต้น จะอยู่ถัดจากส่วนแรก และจะเป็นส่วนที่ได้รับการควบคุมการทำงานในขณะระบบเริ่มทำงาน โดยจะมีโปรแกรมควบคุมไฟล์ออโตเอ็กซ์เซค (AUTOEXEC) ส่วนนี้จะถูกทับไปหลังจากมีการโหลดโปรแกรมแรกเข้ามาทำงาน

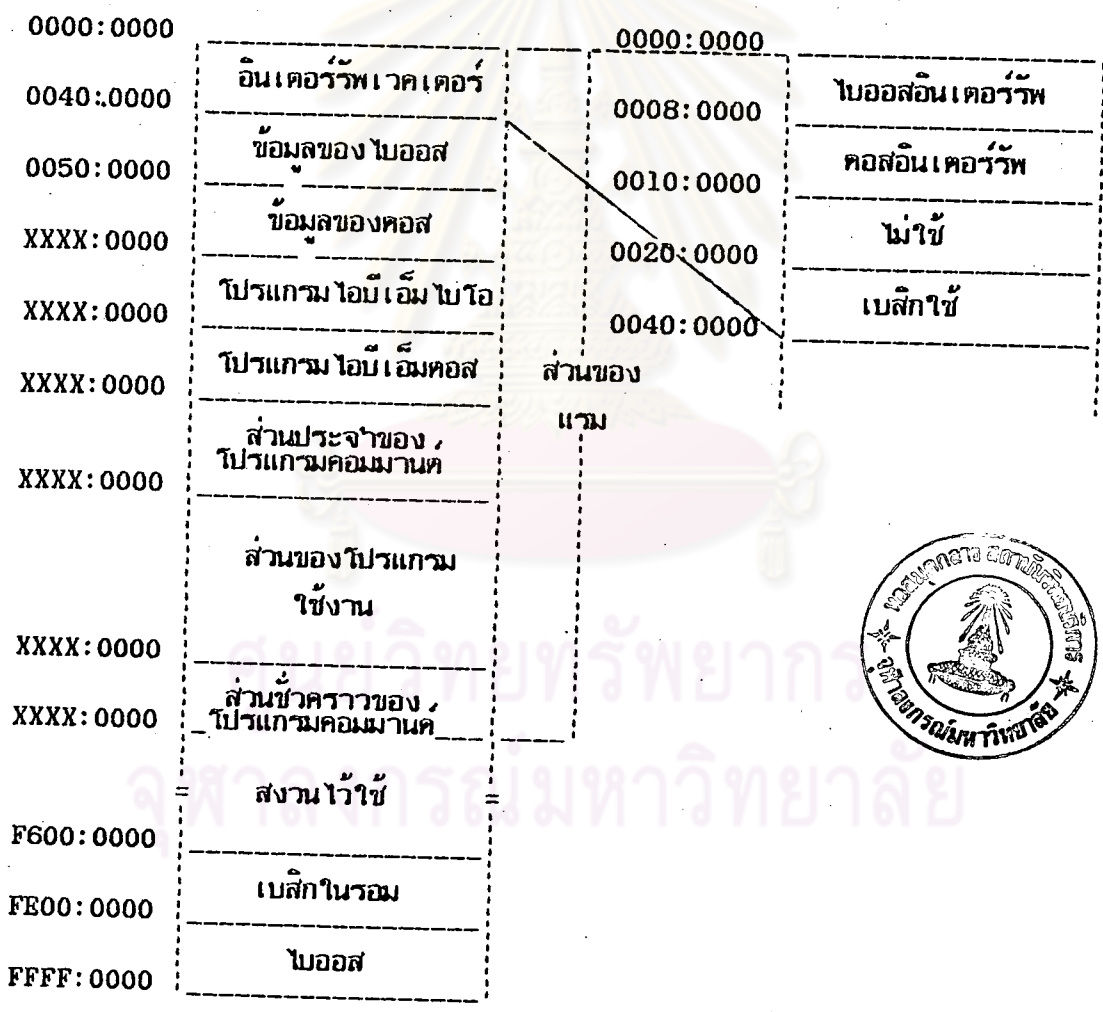
4.3 ส่วนชั่วคราว จะถูกโหลด ณ.ตำแหน่งสูงสุดของหน่วยความจำ ใต้คำสั่งภายในของเอ็มเอสคอสทั้งหมด และส่วนที่จัดการกับแบทไฟล์ (Batch File)

4.4 โหลดเคอร์ (Loader) เป็นส่วนทำการโหลดคำสั่งภายนอกของเอ็มเอสคอสและโปรแกรมใช้งานทั่วไป โหลดเคอร์เป็นส่วนหนึ่งของส่วนชั่วคราว

#### การทำงานเริ่มแรกของระบบ

เมื่อเราเปิดเครื่องไมโครโปรเซสเซอร์จะกำหนดค่า CS เป็น FFFFH และกำหนดค่า IP เป็น 0000H ดังนั้นคำสั่งแรกที่หน่วยประมวลผลกลางจะทำคือคำสั่ง ณ.หน่วยความจำ ณ.ตำแหน่งที่ FFFF0H ซึ่งเป็นตำแหน่งของรอม โปรแกรมไบออสซึ่งอยู่ในรอม ณ.ตำแหน่งนั้นจะเริ่มทำงาน โดยจะตรวจสอบว่ามีอุปกรณ์อะไรบ้างที่ต่อกับระบบในขณะนั้น พร้อมทั้งกำหนดค่าเริ่มต้นให้ หลังจากนั้นจะสร้างไบออสอินเตอร์รัพเวคเตอร์ (BIOS Interrupts Vector) เริ่มจาก

ตำแหน่งที่ 0 ถึง 7FH เสร็จแล้วจะทำการโหลดคอสจากดิสก์ โดยการอ่านบูทเรคคอร์ดเข้ามาในหน่วยความจำ บูทเรคคอร์ดจะตรวจสอบดิสก์แล้วทำการโหลดโปรแกรมไอบีเอ็มไบโอ และโปรแกรมไอบีเอ็มคอสเข้ามาในหน่วยความจำ แล้วจะสร้างคอสอินเตอร์รัฟเวคเตอร์ (DOS Interrupts Vector) ต่อจากไบออสอินเตอร์รัฟเวคเตอร์ จากนั้นจะโหลดโปรแกรมคอมมานด์เข้ามาในหน่วยความจำแล้วส่งการควบคุมการทำงาน ไปให้โปรแกรมคอมมานด์ ณ. จุดนี้ผู้ใช้เครื่องก็สามารถทำงานได้ การจัดหน่วยความจำขนาด 1 เมกกะ ไบต์ของระบบ จะเป็นดังนี้



รูปที่ 2.3 การจัดหน่วยความจำของระบบ



### การไหลคโปรแกรมใช้งานลงหน่วยความจำ

เมื่อโปรแกรมใช้งานจะถูกไหลคเข้าไปในหน่วยความจำ เอ็มเอสคอสจะจัดที่ส่วนล่างสุดของหน่วยความจำในส่วนของโปรแกรมใช้งานไว้ให้ ส่วนนี้เรียกว่า ส่วนของโปรแกรมเซกเมนต์ (PROGRAM SEGMENT) โปรแกรมเอ็กเซค्यूทีฟ (EXEC) ในโปรแกรมไอบีเอ็มคอส จะทำหน้าที่จัดเตรียมข้อมูล 256 ไบต์แรกของโปรแกรมเซกเมนต์เพื่อเตรียมพร้อมสำหรับการทำงานของโปรแกรมที่จะไหลคเข้ามา ส่วนดังกล่าวนี้เรียกว่า โปรแกรมเซกเมนต์พรีฟิกซ์ หรือ พีเอสพี (PROGRAM SEGMENT PREFIX หรือ PSP) ซึ่งมีรายละเอียดดังนี้

ออฟเซต	ขนาด	จัดเก็บ
0	2	คำสั่งของอินเตอร์พรีท 20H
5	5	จุดทางเข้าในการเรียกใช้คอส
A	4	ตำแหน่งบอกการจบโปรแกรม
E	4	ตำแหน่งของโปรแกรมที่จะทำงานเมื่อมีการกด CTRL-C
12H	4	ตำแหน่งของโปรแกรมที่จะทำงานเมื่อโปรแกรมมีข้อผิดพลาด
5CH	16	พารามิเตอร์ค่าที่ 1
6CH	16	พารามิเตอร์ค่าที่ 2
8CH	128	พื้นที่ใช้ในการส่งผ่านข้อมูลกับคิสก์ หรือ ดีทีเอ (DTA)

ตารางที่ 2.1 ลักษณะของพีเอสพี

### คอสอินเตอร์พรีทและฟังก์ชันคอลล์

เป็นส่วนของโปรแกรมที่ระบบสร้างขึ้นมาเพื่อควบคุมการใช้งานของอุปกรณ์ของระบบอินเตอร์พรีทของคอสเริ่มจาก อินเตอร์พรีทที่ 20H ถึง 3FH สำหรับอินเตอร์พรีทที่ 21H จะเป็นคอส

ฟังก์ชันคอลล์ รายละเอียดของคอสอินเตอร์พรีพและฟังก์ชันคอลล์ที่น่าสนใจดังนี้

- อินเตอร์พรีพ 20H - โปรแกรมเทอร์มินेट (Program Terminate) จบการทำงาน  
ของโปรแกรมปัจจุบัน แล้วส่งต่อความคุมการทำงานให้เอ็มเอสดอส
- อินเตอร์พรีพ 21H - เป็นคอสฟังก์ชันคอลล์ที่น่าสนใจดังนี้
- AH = 00H - จบการทำงานของโปรแกรมปัจจุบัน แล้วส่งต่อความคุมการทำงานให้เอ็มเอสดอส
  - AH = 01H - รับการกดตัวอักษรจากคีย์บอร์ด แล้วแสดงผลตัวอักษรบนจอภาพ
  - AH = 02H - แสดงตัวอักษรในรีจิสเตอร์ DL บนจอภาพ
  - AH = 05H - ส่งรหัสตัวอักษรในรีจิสเตอร์ DL ไปยังเครื่องพิมพ์
  - AH = 07H - รอรับการกดตัวอักษรบนแป้นคีย์บอร์ดเก็บไว้ใน AL โดยไม่มีการแสดงตัวอักษรบนจอภาพ และจะมีการเช็คการกด Ctrl-C ด้วย
  - AH = 08H - เหมือนฟังก์ชัน 07H แต่จะไม่เช็ค Ctrl-C
  - AH = 09H - แสดงผลกลุ่มของข้อมูลที่ชี้โดย DS:DX บนจอภาพทุกตัวจนกว่าจะพบตัวอักษร '\$'
  - AH = 0AH - รอรับการกดตัวอักษรจากคีย์บอร์ดเก็บไว้ในบัฟเฟอร์ที่ชี้โดย DS:DX โดยเริ่มเก็บตั้งแต่ไบต์ที่ 3 เป็นต้นไป สำหรับไบต์ที่ 1 จะเป็นจำนวนตัวอักษรที่สามารถเก็บในบัฟเฟอร์ได้ ไบต์ที่ 2 จะเก็บจำนวนตัวอักษรที่เก็บในบัฟเฟอร์จริง ๆ
- หลังจากการกดแครีเอจรีเทอร์น



- AH = 0BH - ตรวจสอบค่าในคีย์บอร์ดเพอร์ ถ้ามีคีย์จะเซตรีจิสเตอร์ AL เป็น FFH ถ้าไม่มี AL จะเป็น 0
- AH = 0CH - เคลียร์ค่าในคีย์บอร์ดเพอร์แล้วเรียกฟังก์ชันที่อยู่ใน AL มาทำงาน
- AH = 0DH - ยกเลิกคีย์บอร์ดของไฟล์ทั้งหมด
- AH = 0EH - เลือกคีย์บอร์ดที่กำหนดในรีจิสเตอร์ DL มาใช้งานโดยที่ 0 - หมายถึงคีย์ A , 1 - หมายถึงคีย์ B เป็นต้น และ AL จะเป็นจำนวนของคีย์บอร์ดที่จะใช้งานได้ทั้งหมด
- AH = 0FH - เปิดไฟล์ตามเอ็พซีบี (FCB) ที่กำหนดโดย DS:DX และ AL จะบอกผลของการทำงานดังนี้  
     AL = 00 การทำงานสำเร็จ  
     AL = FF หาไฟล์ไม่พบ
- AH = 10H - ปิดไฟล์ตามเอ็พซีบี ที่กำหนดโดย DS:DX ผลของการทำงาน เป็นเช่นเดียวกับฟังก์ชัน 0FH
- AH = 11H - ค้นหาไฟล์ตามเอ็พซีบี ที่กำหนดโดย DS:DX ผลการทำงานเป็นเช่นเดียวกับฟังก์ชัน 0FH
- AH = 12H - ค้นหาไฟล์เป็นฟังก์ชันที่ใช้ต่อจากฟังก์ชัน 11H
- AH = 13H - ลบไฟล์ตามเอ็พซีบี ที่กำหนดโดย DS:DX ผลการทำงาน เป็นเช่นเดียวกับฟังก์ชัน 0FH
- AH = 14H - อ่านไฟล์แบบต่อเนื่องตามเอ็พซีบีที่กำหนดโดย DS:DX ค่าของเรคคอร์ดปัจจุบันจะถูกอ่านเข้ามาเก็บในทิตีเอ ผลการทำงาน จะแสดงใน AL ดังนี้

- 0 - การทำงานสำเร็จ
- 1 - อ่านจนจบไฟล์ข้อมูล
- 2 - ขนาดของคิตีเอ ไม่พอ
- 3 - อ่านจนจบไฟล์ เรคคอร์ดสุดท้ายมีขนาดน้อยกว่าคิตีเอ ส่วนที่เหลือจะถูกเติมด้วย 0

AH = 15H - เขียนไฟล์ตามเอ็พซีบีที่กำหนดโดย DS:DX โดยเรคคอร์ดปัจจุบันที่กำหนดในเอ็พซีบี จะเขียนลงไฟล์ผ่านคิตีเอ และจะเพิ่มค่าตำแหน่งของเรคคอร์ดปัจจุบันอีก 1 ผลการทำงานจะแสดงใน AL ดังนี้

- 0 - การทำงานสำเร็จ
- 1 - คิส์ค์เต็ม
- 2 - ขนาดของคิตีเอ ไม่พอ

AH = 16H - สร้างไฟล์ตามเอ็พซีบีที่กำหนดโดย DS:DX

AH = 17H - เปลี่ยนชื่อไฟล์ในเอ็พซีบีที่กำหนดโดย DS:DX ผลการทำงานกำหนดใน AL ดังนี้

- 0 - การทำงานสำเร็จ

FFH - หาไฟล์ไม่เจอหรือชื่อที่เปลี่ยนใหม่มีอยู่แล้ว

AH = 19H - จะให้หมายเลขของคิส์ค์ไควท์ทำงานอยู่ในปัจจุบันไว้ใน AL

AH = 1AH - กำหนดคิตีเอตาม DS:DX ไว้เป็นบัฟเฟอร์ในการเขียนหรืออ่านข้อมูล

AH = 21H - อ่านไฟล์แบบแรนดอมตามเอ็พซีบีที่กำหนดโดย DS:DX โดยตำแหน่งของเรคคอร์ดที่จะอ่านจะกำหนดไว้ในเอ็พซีบี

ผลการทำงานเก็บไว้ใน AL เช่นเดียวกับฟังก์ชัน 14H

AH = 22H - เขียนไฟล์แบบแรมคอมตามเอ็พซีบที่กำหนดโดย DS:DX ผล  
การทำงานแสดงไว้ในรีจิสเตอร์ AL

AH = 23H - หาขนาดของไฟล์ตามเอ็พซีบที่กำหนดโดย DS:DX

AH = 2AH - ให้ค่าวันที่ของระบบตั้ง

CX - ปี

DH - เดือน

DL - วันที่

AL - วัน (0=วันอาทิตย์, 6=วันเสาร์)

AH = 2BH - กำหนดค่าวันที่ของระบบ ค่าของรีจิสเตอร์เหมือนฟังก์ชัน 2AH

AH = 2CH - ให้ค่าของเวลาของระบบ ตั้ง

CH - ชั่วโมง

CL - นาที

DH - วินาที

AH = 2DH - กำหนดค่าของเวลาของระบบ ค่าของรีจิสเตอร์เป็นเช่นเดียวกับฟังก์ชัน 2CH

อินเตอร์วอลล์ 27H - เทอร์มินัลสแตเรสสิเดนท์ (Terminate But Stay Resident) หลังจากเลิกสิ้นสุดการทำงานของโปรแกรม ปัจจุบัน จะเก็บโปรแกรมไว้ในหน่วยความจำตลอดเวลา โดย DX จะเก็บตำแหน่งของคำสั่งท้ายสุด

### ไบออส

เป็นส่วนของโปรแกรมที่จัดการการทำงานเกี่ยวกับอุปกรณ์ต่างๆ ที่ต่ออยู่กับระบบ ไบออส

สามารถเรียกใช้ได้โดยการอินเทอร์รัพไปยังระบบ ไบออสอินเทอร์รัพจะเริ่มตั้งแต่อินเทอร์รัพ 0H ถึง 1FH ไบออสอินเทอร์รัพที่น่าสนใจมีดังนี้

- อินเทอร์รัพ 00H - ติวบายซีโร (Divide By Zero)  
จะเกิดเมื่อผลการหารโดยใช้ค่าสั่งหารไม่สามารถเก็บค่าในรีจิสเตอร์ผลลัพธ์ได้
- อินเทอร์รัพ 01H - ซิงเกิลสเต็ป (Single Step)  
จะทำให้การทำงานของระบบทำงานไปครั้งละ 1 ขั้นตอน จะใช้ในการแก้ไขโปรแกรม
- อินเทอร์รัพ 05H - ปรีนทส์กรีน (Print Screen)  
จะส่งตัวอักษรที่ปรากฏบนจอภาพไปพิมพ์ยังเครื่องพิมพ์
- อินเทอร์รัพ 09H - คีย์บอร์ดอินเทอร์รัพ (Keyboard Interrupt)  
จะรอรับการกดแป้นบนคีย์บอร์ดแล้วเก็บค่าของอักษรที่ได้ไว้ในคีย์บอร์ดบัฟเฟอร์
- อินเทอร์รัพ 10H - วีดีโอไอโอ (Video I/O)  
จัดการเกี่ยวกับการแสดงผลบนจอภาพ เช่น การแสดงตัวอักษร การเลื่อนเคอร์เซอร์ เป็นต้น
- อินเทอร์รัพ 13H - ดิสเกตต์ไอโอ (Diskette I/O)  
จัดการเกี่ยวกับการอ่าน เขียนข้อมูลลงดิสก์
- อินเทอร์รัพ 16H - คีย์บอร์ดไอโอ (Keyboard I/O)  
จะส่งค่าของรหัสตัวอักษรที่เก็บในคีย์บอร์ดบัฟเฟอร์ไปให้โปรแกรมหลักโดยผ่านรีจิสเตอร์ AL
- อินเทอร์รัพ 17H - ปรีนทเตอร์ไอโอ (Printer I/O)  
ส่งค่าตัวอักษรในรีจิสเตอร์ AL ไปยังเครื่องพิมพ์