บทที่ 10

การประมวลผลคีย์บอร์ดขั้นสูง

ADVANCED KEYBOARD PROCESSING

วัตถุประสงค์

หลังจากที่ท่านศึกษาบทนี้ท่านจะมีความเข้าใจดังต่อไปนี้

- การทำงานของคีย์บอร์ด
- การใช้ดอสสำหรับคีย์บอร์ดขั้นสูง
- การใช้ไบออสอินเตอร์รัพท์สำหรับคีย์บอร์ด
- การใช้ Extended function keys

CT 215 257

บทที่ 10
การประมวลผลคีย์บอร์ดขั้นสูง
ADVANCED KEYBOARD PROCESSING

บทนำ

ในบทที่ 8 ใช้บริการ 0AH และ 3FH สำหรับ INT 21H เพื่อเป็นการรับข้อมูลจากคีย์บอร์ด ฟังชั่นการทำงานเหล่านี้เป็น 2 แบบ ในหลายๆทางที่สามารถรับข้อมูลจากคีย์บอร์ด ส่วนบทนี้ใช้บริการ 0AH และ INT 16H สำหรับการรับข้อมูลจากคีย์บอร์ด รวมทั้งการทำงานของคีย์บอร์ดในส่วนของ SHIFT STATE SCAN CODE , KEYBOARD BUFFER ส่วน SHIFT STATE เป็นข้อมูลใน BIOS สามารถนำมาใช้ได้ถ้าต้องการ ตัวอย่าง CTRL , SHIFT หรือ ALT ถ้ามีการกดคีย์เหล่านี้ใช้งาน SCAN CODE คือจำนวนที่กำหนดการใช้คีย์สูงสุดบนคีย์บอร์ดสามารถกำหนดการทำงานของระบบถ้ามีการกดคีย์ และโปรแกรมสามารถตรวจเช็คสำหรับคีย์ที่ไม่ใช้รหัส ASCII เช่น HOME , PGUP , PGDN , ARROWS. ในภาคผนวกจะมีข้อมูลของการ SCAN CODE สำหรับแต่ละคีย์บนคีย์บอร์ด ส่วน KEYBOARD BUFFER เป็นหน่วยความจำสำหรับการเก็บข้อมูลก่อนโปรแกรมจะต้องทำงานตามอินพุท

SHIFT STATUS

พื้นที่ในส่วนของ BIOS อยู่ที่เซคเม้นต์แอดเดรส 40:[0] ซึ่งเป็นแอดเดรสข้อมูลที่จะนำไปใช้ประโยชน์ ข้อมูลเหล่านี้ไบท์แรกของ SHIFT STATUS อยู่ที่ตำแหน่ง 40:17H มี 1 บิตเป็นตัวชี้การทำงานต่อไปนี้

BIT	ACTION	BIT	ACTION
7	INSERT TOGGLED	3	ALT PRESSED
6	CAPSLOCK STATE TOGGLED	2	CTRL PRESSED
5	NUMLOCK STATE TOGGLED	1	LEFT SHIFT PRESSED
4	SCROLL LOCK STATE TOGGLED	0	RIGHT SHIFT PRESSED

ท่านอาจจะใช้ INT 16H บริการ 02 ในการตรวจสอบค่าเหล่านี้ ส่วนไบท์ที่ 2 ของคีย์บอร์ดอยู่ที่ตำแหน่ง 40:[18] มี 1 บิตชี้ค่าการทำงานต่อไปนี้

BIT	ACTION	BIT	ACTION
7	INSERT PRESSED	3	CTRL/NUMLOCK PRESSED
6	CAPSLOCK PRESSED	2	SYSREQ PRESSED
5	NUMLOCK PRESSED	1	LEFT ALT PRESSED
4	SCROLL LOCK PRESSED	0	LEFT CTRL PRESSED

บิต 0 , 1 , 2 บริษัท IBM ใช้เป็นส่วนขยายในการสร้าง 101 คีย์

SHIFT STATUS EXERCISE

ดูผลจากการใช้ CTRL , ALT และ SHIFT ในส่วนของการใช้ไบท์ SHIFT STATUS โดยการโหลดโปรแกรมดีบักเพื่อการเอ็กซีคิวส์ โดยมีการป้อนดังนี้ D 40:10 [enter] แสดงข้อมูลของ STATUS BYTE การกดคีย์ CAPSLOCK , NUMLOCK , SCRILOCK สามารถใช้ดีบักที่ D 40:17 เพื่อดูผลลัพธ์การทำงานของสภาวะทั้งสอง ไบท์ที่ 40:17 จะแสดงค่า 70 และไบท์ 40:18 แสดงค่า 00

KEYBOARD BUFFER

รายการที่น่าสนใจของ BIOS ที่ข้อมูลในตำแหน่ง 40:14H คือข้อมูลในส่วนของ KEYBOARD BUFFER พื้นที่ในส่วนนี้ท่านสามารถเก็บข้อมูลได้ถึง 15 ตัวอักขระ ก่อนที่โปรแกรมจะรับข้อมูลเพื่อทำงานต่อไป เมื่อท่านกดคีย์ ตัวโปรเซสเซอร์ที่ทำหน้าที่ควบคุมคีย์บอร์ด จะจ่ายคีย์ของ SCAN CODE และใช้ INT 09

ตัวอย่าง การอินเตอร์รัพท์ของ BIOS จะใช้ค่า SCAN CODE จากคีย์บอร์ดเปลี่ยนเป็นตัวอักขระ ASCII และส่งไปยังคีย์บอร์ดบัฟเฟอร์ การทำงานในส่วนย่อยต่อไป BIOS INT 16H จะอ่านตัวอักขระจากบัฟเฟอร์และส่งไปยังโปรแกรม

DOS INT 21H FOR KEYBOARD INPUT

ต่อไปนี้จะเป็นการอธิบายการทำงานของ DOS สำหรับการรับข้อมูลจากคีย์บอร์ด การทำงานทั้งหมดยกเว้นการบริการ 0AH จะรับข้อมูลที่ละตัวอักขระ แต่ถ้าเป็นชุดสตริงเราก็สามารถใช้ในการวนรอบในการรับข้อมูลตัวอักขระชุดสตริง และมีการตรวจสอบ Backspace และ Enter keys ในการแสดงผลที่จอภาพ

พร้อมเลื่อนเคอร์เซอร์ไป 1 ตำแหน่ง สำหรับ DOS ใช้บริการ 0AH และ INT 21H

INT 21H SERVICE 01 : KEYBOARD INPUT WITH ECHO

การทำงานในการรับข้อมูลตัวอักขระจากคีย์บอร์ดบัฟเฟอร์ หรือถ้าไม่มีการรับข้อมูลก็จะรอคอยการรับข้อมูลจากคีย์บอร์ด การทำงานจะมี 1 หรือ 2 ชนิด ถ้า AL ไม่ใช่ 0 ก็จะเป็นข้อมูลมาตราฐานของรหัส ASCII เช่นตัวอักขระ ตัวเลข ซึ่งข้อมูลเหล่านี้จะแสดงผลที่จอภาพ ถ้าเป็นค่า 0 ใน AL หมายความว่าผู้ใช้กดฟังชั่นคีย์เช่น HOME , F1 , PGUP เป็นจุดเริ่มของการ SCAN CODE ใน AL จะทำให้เกิดการทำงานของการบริการซ้ำ ดังตัวอย่างการทำงานของ CTRL/BREAK ดังต่อไปนี้

```
MOV     AH,01       ;REQUEST KEYBOARD INPUT
INT     21H
CMP     AL,00       ;EXTENDED FUNCTION KEY INPUT PRESSED
JNZ     ....        ;NO ASCII CHARACTER
INT     21H         ;YES--REPEAT OPERATION
```

INT 21H SERVICES 06:DIRECT CONSOLE I/O

การทำงานในการเคลื่อนย้ายข้อมูลของตัวอักขระที่เป็นไปได้และการควบคุมรหัสที่ไม่มีการอ้างถึง DOS มีการทำงานอยู่ 2 ชนิด สำหรับอินพุตและเอ้าพุต สำหรับตัวอย่างอินพุต คือการโหลดค่า 0FFH ไว้ใน DL ถ้าไม่มีตัวอักขระในบัฟเฟอร์ การทำงานก็จะทำให้ ZF = 1 และไม่มีการรอคอยอินพุต ถ้ามีตัวอักขระอยู่ในบัฟเฟอร์มันจะรอคอยการทำงานเพื่อส่งไบไว้ใน AL และจะทำให้ ZF = 0 การทำงานชนิดนี้จะไม่แสดงผลที่จอภาพ และไม่มีการตรวจสอบสำหรับ CTRL/BREAK หรือ CTRL/PRTSC ถ้า AL มีค่าไม่ใช่ 0 มันก็จะแทนด้วยรหัส ASCII เช่น ตัวอักขระ ตัวเลข ถ้าค่าเป็น 0 ใน AL หมายความว่าผู้ใช้กดฟังชั่นพิเศษเช่น HOME , F1 , PGUP เพื่อนำในการ SCAN CODE ใน AL เพื่อทำงานซ้ำ

```
K10:    MOV     AH,06
        MOV     DL,0FFH     ;REQUEST KEYBOARD INPUT
        INT     21H
        JZ      K10         ;BUFFER EMPTY--REPEAT
        CMP     AL,00       ;EXTENDED FUNCTION KEY PRESSED
        JNZ     K30         ;NO--ASCII CHARACTER
        INT     21H         ;YES--REPEAT OPERATION
```

ท่านสามารถให้การทำงานในการรอคอยข้อมูลจากคีย์บอร์ดโดยใส่รหัส INT ไว้ในการวนรอบดังนี้

```
K20:    INT  21h
        JZ   K20        ;BUFFER EMPTY ,TRY AGAIN
```

สำหรับการทำงานแสดงผลที่เอ้าพุตเก็บค่าตัวอักขระ ASCII ไว้ใน DL

INT 21H SERVICE 07: DIRECT KEYBOARD INPUT WITHOUT ECHO

การทำงานชนิดนี้เหมือนกับการบริการ 01 ยกเว้นตัวอักขระที่ป้อนเข้ามาไม่ไปปรากฏที่จอภาพ และการทำงานนี้ไม่ตอบสนองของ CTRL/BREAK ท่านสามารถใช้การทำงานนี้สำหรับ PASSWORD INPUT หรือไม่ต้องการแสดงที่จอภาพ

INT 21H SERVICE 08: KEYBOARD INPUT WITH ECHO

การทำงานชนิดนี้เหมือนกับการบริการ 01 แต่ไม่แสดงข้อมูลที่จอภาพ

INT 21H SERVICE 0AH: BUFFERED KEYBOARD INPUT

วิธีการของการบริการชนิดนี้ทำงานได้อย่างมีประสิทธิภาพ และใช้ประโยชน์ในส่วน DOS KEYBOARD รายละเอียดกลับไปดูบทที่ 8

INT 21H SERVICE 0BH: CHECK KEYBOARD STATUS

การทำงานจะให้ค่า 0FFH ไว้ใน AL ถ้าข้อมูลตัวอักขระที่ป้อนเข้ามามีค่าหรือ 00 ถ้าไม่มีตัวอักขระที่ป้อนเข้ามา การบริการนี้จะไม่มีการรอคอยการรับข้อมูลจากคีย์บอร์ด

INT 21H SERVICE 0CH CLEAR KEYBOARD BUFFER AND INVOKE SERVICE

ท่านสามารถใช้การบริการชนิดนี้ร่วมกับการบริการชนิดอื่นๆ เช่น 01,06,07,08 หรือ 0AH โดยการโหลดค่าบริการไว้ใน AL

```
MOV     AH,0CH          ;REQUEST KEYBOARD INPUT
MOV     AL,SERVICE      ;SERVICE CALL
MOV     DX,KBAREA       ;KEYBOARD AREA FOR SERVICE 0AH
INT     21H
```

การทำงานในส่วนแรกจะเป็นการเคลียบัฟเฟอร์ของคีย์บอร์ด การเอ็กซิคิวส์การบริการใน AL และรับ (หรือรอคอย) ตัวอักขระ ท่านสามารถใช้การทำงานสำหรับโปรแกรมในการพิมพ์ล่วงหน้า เช่นโปรแกรม DOS ที่ท่านพิมพ์ล่วงหน้า

BIOS INTRRUPT 16H FOR KEYBOARD INPUT

BIOS INT 16H ซึ่งเป็นพื้นฐานการทำงานของคีย์บอร์ดในส่วนของ BIOS ใช้ในการพัฒนาซอฟแวร์คอมมานด์ในการรับข้อมูลของการบริการไว้ใน AH แต่ต้องขึ้นอยู่กับ IBM ROM BIOS

INT 16H SERVICE 00: READ CHARACTER

การทำงานนี้เป็นการตรวจสอบบัฟเฟอร์สำหรับตัวอักขระ ถ้าไม่มีตัวอักขระจะรอคอย ถ้ามีตัวอักขระการทำงานจะย้อนกลับไปที่ AL และมันจะ SCAN CODE ใน AH ถ้ากดีย์ฟังชั่นเพิ่มเช่น HOME , F1 ตัวอักขระคือ 00 ลักษณะนี้มีความเป็นไปได้ 2 ทางคือ

Key pressed	AH	AL
Regular ASCII character:	scan code	ASCII character
Extended function key:	scan code	ZERO

ในกรณีนี้ ท่านสามารถตรวจสอบรีจีสเตอร์ AL สำหรับค่า 0 เมื่อผู้ใช้กดฟังชั่นคีย์ การทำงานชนิดนี้จะไม่แสดงผลที่จอภาพ

```
MOV     AH,00           ;REQUEST BIOS INPUT
INT     16H
CMP     AL,00           ;EXTENDED FUNCTION KEY?
```

INT 16H SERVICE 01: DETERMINE IF A CHARACTER IS AVAILABLE

การทำงานส่วนนี้เหมือนกับการบริการ 00 ถ้าตัวอักขระมีอยู่ในคีย์บอร์ดบัฟเฟอร์ การทำงานก็เคลีย ZF = 0 และเก็บค่าตัวอักขระใน AL และ SCAN CODE ใน AH จากบัฟเฟอร์ ถ้าไม่มีตัวอักขระการทำงานก็เซ็ต ZF = 1 และไม่รอคอย สังเกตการทำงานจะทำต่อไปเรื่อยๆ ของตัวอักขระในคีย์บอร์ดบัฟเฟอร์จนกระทั่งบริการ 00 อ่านข้อมูล

INT 16H SERVICE 02:RETURN THE CURRENT SHIFT STATUS

การทำงานจะส่งค่า AL ที่แสดงสถานะของ KEYBOARD SHIFT จากข้อมูล BIOS ตำแหน่ง 417H (40:17) ไบท์สถานะอธิบายไปแล้ว การตรวจสอบค่า SHIFT LEFT และ SHIFT RIGHT ของการกดคีย์มีดังนี้

```
MOV     AH,02           ;REQUEST SHIFT STATUS
INT     16H
OR      AL,00000011B    ;LEFT OR RIGHT SHIFT
JE      XXXX
```

INT 16H SERVICE 05H: KEYBOARD WRITE

การทำงานนี้จะเก็บค่าตัวอักขระในคีย์บอร์ดบัฟเฟอร์ ถ้าผู้ใช้มีการกดคีย์ จะโหลดรหัสตัวอักขระ ASCII ใน CH และ SCAN CODE ใน CL การทำงานนี้ท่านต้องป้อนข้อมูลในบัฟเฟอร์จนเต็ม

INT 16H SERVICE 10H READ A CHARACTER

การทำงานนี้เหมือนกับการบริการ 00 ยกเว้นการเพิ่มฟังชั่นขยาย เช่น F11 และ F12 จาก IBM extended functions เป็นงานบริการสำหรับ AT , XT 286 ,ps/2

INT 16H SERVICE 11H: DETERMINE IF A CHARACTER IS AVAILABLE

การทำงานนี้เหมอนกับการบริการ 01 ยกเว้นการเพิ่มฟังชั่นขยาย เช่น F11 และ F12 จาก IBM extended functions เป็นงานบริการสำหรับ AT , XT 286 ,ps/2

INT 16H SERVICE 12H: RETURN THE CURRENT SHIFT STATUS

การทำงานนี้เหมอนกับการบริการ 02 ยกเว้นว่ามันจะส่งให้ AH คือ extended shift status งานบริการสำหรับ AT , xt-286 , ps/2

BIT	KEY	BIT	KEY
7	SYS REQ PRESSED	3	RIGHT ALT PRESSED
6	CAPS LOCK PRESSED	2	RIGHT CTRL PRESSED
5	NUM LOCK PRESSED	1	LEFT ALT PRESSED
4	SCROLL LOCK PRESSED	0	LEFT CTRL PRESSED

CT 215 265

EXTENDED FUNCTION KEYS

คีย์บอร์ดแบ่งชนิดของคีย์ออกเป็น 3 ชนิด

1. Character เช่น ตัวอักษร A - Z ตัวเลข 0 - 9 และ %,$,#,@,&
2. Exyended function keys เช่น Home,End,Backspace,Arrows,Enter,Del.Ins PgUp,PgDn และโปรแกรมฟังชั่นคีย์
3. Control keys สำหรับ Alt,Ctrl และ Shift ซึ่งทำงานร่วมกับฟังชั่นอื่นๆ

Extended function keys การทำงานจะมีลักษณะคล้ายกับการส่งตัวอักขระแต่ไม่มีอะไร การออกแบบคีย์จะเป็นการทำงานตามรูปแบบที่กำหนด ดังตัวอย่างถ้าท่านกดคีย์ Home จะเป็นการเซ็ตเคอร์เซอร์ไปที่ตำแหน่งบนขวาสุดของจอภาพ หรือการกดคีย์ End จะเซ็ตเคอร์เซอร์ที่จุดสุดท้ายของ Text บนจอภาพ ในแต่ละคีย์การออกแบบสามารถ SCAN CODE จำนวนเริ่มต้นค่า 01 สำหรับ ESC ที่อ้างในภาคผนวก ที่แสดงรายละเอียดของคำสั่งเหล่านี้ ความหมายของ SCAN CODE เหล่านี้ โปรแกรมอาจจะต้องการ Source ของทุกๆ คีย์ ตัวอย่าง ในการรับข้อมูลตัวอักษร 1 ตัว ของ INT 16H การบริการ 00 การทำงานอาจจะมี 1 หรือ 2 ทาง ขึ้นอยู่กับการกดคีย์ตัวอักขระหรือ Extended function keys สำหรับตัวอักขระเช่น ตัวอักษร A การทำงานจะส่งรายการ 2 ค่า

1. ตัวอักขระ ASCII A (41H) ใน AL
2. SCAN CODE สำหรับตัวอักษร A (1EH) ใน AH

1E	41
AH	AL

ถ้าเรากดคีย์ประเภท Extebded function keys เช่น Ins การทำงานจะส่งค่า 2 ค่าดังนี้

1. ค่า 00 ในรีจีสเตอร์ AL
2. SCAN CODE สำหรับ Ins คือ 52H ในรีจีสเตอร์ AH

52	00
AH	AL

กรณีนี้หลังจากการทำงานของคำสั่ง INT 16H (บางครั้งก็ใช้ INT 21H) ท่านสามารถตรวจสอบค่าในรีจีสเตอร์ AL ถ้ามีค่าเป็น 00 ก็จะเป็นประเภท Exended function code ถ้าไม่ใช่ 0 การทำงานก็จะเป็นรหัสตัวอักขระ ASCII ต่อไปนี้เป็นการตรวจสอบ Extended function:

```
MOV     AH,00     ;REQUEST INPUT
INT     16H       ;CALL BIOS
CMP     AL,00     ;EXTENDED FUNCTION?
JZ      EXIT      ;YES--EXIT
```

ข้อมูลของคีย์บอร์ด 2 คีย์สำหรับตัวอักขระเช่น - , + และ * การกดคีย์ Asterisk ในตัวอย่างจะมีรหัส ASCII คือ 2AH จะเก็บไว้ใน AL และ 1 ใน 2 ค่าที่เป็น SCAN CODE เก็บใน AH ขึ้นอยู่กับชนิดของคีย์ที่กด ค่า 09H ถ้าเป็นรหัส Asterisk ในคีย์บอร์ดที่อยู่บนเลข 8 หรือค่า 29H ถ้ารหัส Asterisk อยู่บนคีย์ PrtSc การตรวจสอบรหัสการทำงานของ SCAN CODE ของ Asterisk มีดังนี้

```
CMP     AL,2AH    ;ASTERISK?
JNE     EXIT1     ;NO--EXIT
CMP     AH,09H    ;WHICH SCAN CODE?
JE      EXIT2
```

การเซ็ตเคอร์เซอร์ไปที่ บรรทัด 0 คอลัมน์ 0 ถ้าผู้ใช้กด Home key (SCAN CODE = 47H)

```
MOV     AH,00     ;REQUEST INPUT
INT     16H
CMP     AL,00     ;EXENDED FUNCTION?
JNE     EXIT1     ;NO--EXIT
CMP     AH,47H    ;SCAN CODE FOR HOME?
JNE     EXIT2     ;NO--EXIT
MOV     AH,02
MOV     BH,00     ;SET CURSOR
MOV     DX.00     ;TO 0,0
INT     10H       ;CALL BIOS
```

ฟังชั่นคีย์ F1 - F10 จะจ่ายรหัส SCAN CODE 3BH - 44H ดังตัวอย่างการตรวจสอบฟังชั่น F10

```
        CMP     AH,44H      ;FUNCTION KEY F10
        JE      EXIT1       ;YES--EXIT
```

ที่ตำแหน่ง EXIT1 โปรแกรมจะไปทำงานตามต้องการ

SELECTING FROM A MENU

ส่วนหนึ่งของโปรแกรมในตัวอย่าง 10-1 แสดงการทำงานของเมนู และตัวลูกศรในการเลื่อนขึ้นเลื่อนลงเพื่อเลือกรายการ มีรูปแบบแต่ละ Procedure ทำงานดังต่อไปนี้

BEGIN การเรียกโปรแกรม Q10CLR ในการเคลียจอภาพ การเรียก B10MENU แสดงรายการเมนูและเซ็ตรายการแรกเป็น Reverse video และการเรียก D10INPT เป็นการรับข้อมูลจากคีย์บอร์ด

B10MENU การเรียก H10DISP แสดงแต่ละรายในเมนู

D10INPT การใช้ INT 16H สำหรับอินพุต ลูกศรลงเป็นการเลื่อนรายการลง ลูกศรขึ้นเป็นการเลื่อนรายการขึ้น คีย์ Enter เป็นการรับรายการข้อมูล และคีย์ ESC ออกจากรายการ ส่วนคีย์อื่นๆทั้งหมดเมื่อป้อนเข้ามาจะไม่สนใจ การทำงานของ Reverse video จะทำแบบวนรอบ

H10DISP การแสดงการเลือกบรรทัดตาม Attribute

Q10CLR เคลียจอภาพให้ตัวหนังสือสีน้ำเงินและพื้นสีน้ำตาล

```
                page    60,132
                TITLE     SELMENU (EXE) Select line from menu
; ------------------------------------------------
                .MODEL SMALL
                .STACK 64
; ------------------------------------------------
                .DATA
    TOPROW      EQU         00      ;Top row of menu
    BOTROW      EQU         07      ;Bottom row of menu
    LEFCOL      EQU         16      ;Left column of menu
    COL         DB          00      ;Screen column
```

```
ROW       DB        00        ;Screen row
COUNT     DB        ?         ;Characters per line
LINES     DB        ?         ;Lines displayed
ATTRIB    DB        ?         ;Screen attribute
NINTEEN   DB        19        ;Width of menu
MENU      DB        '┌─────────────────┐'
          DB        '│ Add records     │'
          DB        '│ Delete records  │'
          DB        '│ Enter orders    │'
          DB        '│ Print report    │'
          DB        '│ Update accounts │'
          DB        '│ View records    │'
          DB        '└─────────────────┘'
PROMPT    DB        09, 'To select an item, use up/down arrow'
          DB        ' and press Enter.'
          DB        13, 10, 09, 'Press Esc to exit.'
;-------------------------------------------------
          .CODE
BEGIN     PROC   FAR
          MOV     AX,@data          ;Initialize data segment
          MOV     DS,AX
          MOV   ES,AX
          CALL    Q10CLR            ;Clear screen
          MOV   ROW,BOTROW+2
          MOV   COL,00
          CALL    Q20CURS           ;Set cursor
          MOV   AH,40H              ;Request display for prompt
          MOV   BX,01               ;Handle for screen
          MOV   CX,75               ;Number of characters
          LEA   DX,PROMPT
          INT   21H
```


269

```
A10LOOP:
            CALL    B10MENU          ;Display menu
            MOV     COL,LEFCOL+1
            CALL    Q20CURS          ;Set cursor
            MOV     ROW,TOPROW+1     ;Set row to top item
            MOV     ATTRIB,16H       ;Set reverse video
            CALL    H10DISP          ;Highlight current menu line
            CALL    D10INPT          ;Provide for menu selection
            CMP     AL,0DH           ;Enter pressed?
            JE      A10LOOP          ;  yes -- continue
            MOV     AX,0600H         ;Esc pressed (indicates end)
            CALL    Q10CLR           ;Clear screen
            MOV     AH,4CH           ;  and terminate
            INT     21H
BEGIN       ENDP
;           Display full menu:
;   ----------------
B10MENU     PROC    NEAR
            MOV     ROW,TOPROW       ;Set top row
            MOV     LINES,08         ;Number of lines
            LEA     SI,MENU
            MOV     ATTRIB,71H       ;Blue on white
B20:
            MOV     COL,LEFCOL       ;Set left column of menu
            MOV     COUNT,19
B30:
            CALL    Q20CURS          ;Set cursor next column
            MOV     AH,09            ;Request display
            MOV     AL,[SI]          ;Get character from menu
            MOV     BH,00            ;Page 0
            MOV     BL,71H           ;New attribute
            MOV     CX,01            ;One character
```

```
                INT     10H
                INC     COL             ;Next column
                INC     SI              ;Set for next character
                DEC     COUNT           ;Last character?
                JNZ     B30             ;No -- repeat
                INC     ROW             ;Next row
                DEC     LINES
                JNZ     B20             ;All lines printed?
                RET                     ;If so, return
    B10MENU     ENDP
;               Accept input for request:
;               -------------------------
    D10INPT     PROC    NEAR
                MOV     AH,0            ;Request input character
                INT     16H
                CMP     AH,50H          ;Down arrow?
                JE      D20
                CMP     AH,48H          ;Up arrow?
                JE      D30
                CMP     AL,0DH          ;Enter key?
                JE      D90
                CMP     AL,1BH          ;Escape key?
                JE      D90
                JMP     D10INPT         ;None -- retry
    D20:        MOV     ATTRIB,71H      ;Blue on white
                CALL    H10DISP         ;Set old line to normal video
                INC     ROW
                CMP     ROW,BOTROW-1    ;Past bottom row?
                JBE     D40             ;No -- ok
                MOV     ROW,TOPROW+1    ;Yes -- reset
                JMP     D40
    D30:        MOV     ATTRIB,71H      ;Normal video
```

```
                CALL    H10DISP         ;Set old line to normal video
                DEC     ROW
                CMP     ROW,TOPROW+1    ;Below top row?
                JAE     D40             ;No -- ok
                MOV     ROW,BOTROW-1    ;Yes -- reset
D40:            CALL    Q20CURS         ;Set cursor
                MOV     ATTRIB,16H      ;Reverse video
                CALL    H10DISP         ;Set new line to reverse video
                JMP     D10INPT
D90:            RET
D10INPT         ENDP
;               Set menu line to normal/highlight:
;               ----------------------------------
H10DISP         PROC    NEAR
                MOV     AH,00
                MOV     AL,ROW          ;Row tells which line to set
                MUL     NINTEEN         ;Multiply by length of line
                LEA     SI,MENU+1       ;  for selected menu line
                ADD     SI,AX
                MOV     COUNT,17        ;Characters to display
H20:
                CALL    Q20CURS         ;Set cursor next column
                MOV     AH,09           ;Request display
                MOV     AL,[SI]         ;Get character from menu
                MOV     BH,00           ;Page 0
                MOV     BL,ATTRIB       ;New attribute
                MOV     CX,01           ;One character
                INT     10H
                INC     COL             ;Next column
                INC     SI              ;Set for next character
                DEC     COUNT           ;Last character?
                JNZ     H20             ;No -- repeat
```

```
                MOV   COL,LEFCOL+1        ;Reset column to left
                CALL   Q20CURS            ;Set cursor
                RET
    H10DISP     ENDP
;               Clear screen:
;               ------------
    Q10CLR      PROC   NEAR
                MOV   AX,0600H
                MOV   BH,61H                          ;Blue on brown
                MOV   CX,0000
                MOV   DX,184FH
                INT   10H                             ;Call BIOS
                RET
    Q10CLR      ENDP

;               Set cursor row:column:
;               ----------------------
    Q20CURS     PROC   NEAR
                MOV   AH,02
                MOV   BH,00           ;Page 0
                MOV   DH,ROW          ;Row
                MOV   DL,COL          ;Column
                INT   10H
                RET
    Q20CURS     ENDP
                END     BEGIN
```

รูป 11.1 Display MENU

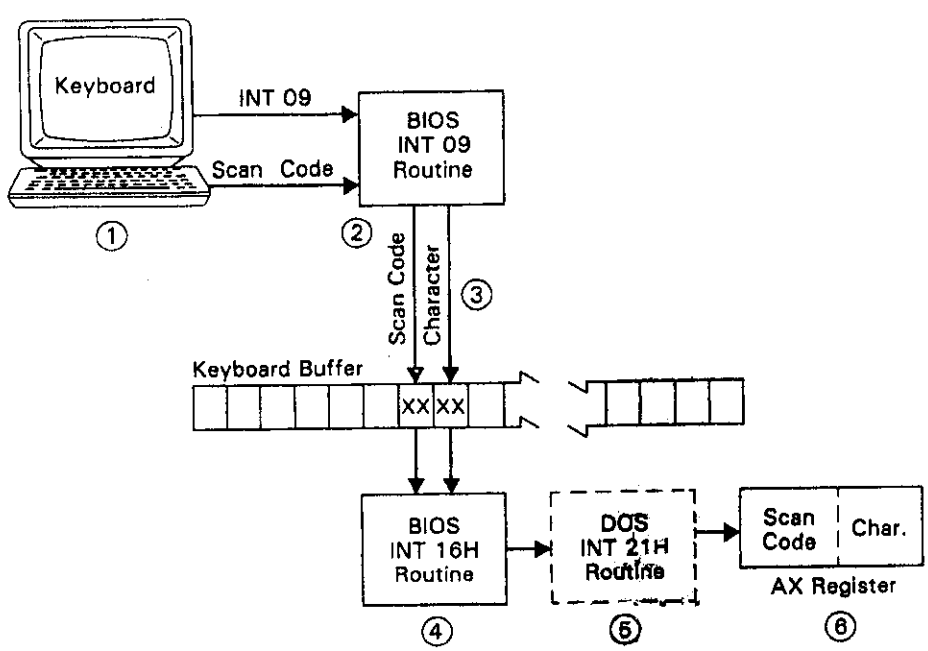
INTERRUPT 09 AND THE KEYBOARD BUFFER

เมื่อมีการกดคีย์ ตัวโปรเซสเซอร์ของคีย์บอร์ดจะจ่ายสัญญาณของการ SCAN CODE และใช้ INT 09 การอินเตอร์รัพท์ชนิดนี้ (ตำแหน่งที่ 36) เป็นการอินเตอร์รัพท์ของ ROM BIOS งานที่ทำคือการรับข้อมูลจาก Port 96 (60H) ดังนี้

```
IN      AL,60H
```

การอ่านงานของการ SCAN CODE จะทำการเปรียบเทียบค่าสแกนในตารางสำหรับรหัส ASCII งานนี้จะรวมถึงการ SCAN CODE และรหัส ASCII เพื่องส่งข้อมูล 2 ไบท์ไปยังคีย์บอร์ดบัฟเฟอร์ ตามรูป 10-2 แสดงวิธีการทำงาน

ข้อยกเว้น INT 09 จะตรวจสอบไบท์สถานะของคีย์บอร์ด 40:17 และ 40:18 สำหรับคีย์ SHIFT ALT , CTRL ถึงแม้ว่าสัญญาณเหล่านี้จะสร้างโดย INT 09 งานอินเตอร์รัพท์ก็จะไม่ส่งค่าไปยังบัฟเฟอร์ ดังนั้น INT 09 จะไม่สนใจคีย์ที่ไม่กำหนด เมื่อมีการกดคีย์ ตัวโปรเซสเซอร์ของคีย์บอร์ดจะจ่ายการสัญญาน SCAN CODE และ INT 09 เมื่อท่านกดคีย์ภายใน 1/2 วินาที มันจะจ่าย SCAN CODE ในส่วนที่ 2 (รหัสแรกบวก 128 บิตซ้ายสุดจะถูกเซ็ต) และจะทำงานอื่นใน INT 09 การ SCAN CODE ในส่วนที่ 2 จะบอกการบริการอินเตอร์รัพท์ว่า ท่านมีการกดคีย์ ถ้าท่านกดคีย์มากกว่า 1/2 วินาที คีย์บอร์ดถือว่าท่านกดคีย์ซ้ำ



รูป 10-2 KEYBOARD BUFFER

THE KEYBOARD BUFFER

การทำงานของคีย์บอร์ดบัฟเฟอร์ต้องการ 2 แอดเดรส แอดเดรสแรกบอก INT 09 ขณะที่รับตัวอักษรต่อไปและอีกแอดเดรสบอก INT 16H ขณะที่รวมกับตัวอักษรต่อไป แอดเดรสคือค่าออฟเซ็ตภายในเซคเม้นต์ 40[0]

Hex 41A แอดเดรสปัจจุบันที่จุดเริ่มต้นของบัฟเฟอร์ที่จะอ่านตำแหน่งต่อไป

Hex 41C แอดเดรสสุดท้ายของบัฟเฟอร์ ตำแหน่งต่อไปที่เก็บข้อมูลตัวอักขระ

Hex 41E คีย์บอร์ดบัฟเฟอร์ 16 เวิร์ด (32 ไบท์) สามารถกำหนดความยาวได้ 32 ไบท์ บัฟเฟอร์ของคีย์บอร์ดเก็บตัวอักขระและ SCAN CODE ที่ป้อนเข้ามาหลังจากการอ่านของ INT 16H 2 ไบท์คือรหัสตัวอักษรและ SCAN CODE

ADDRESS OF HEAD	ADDRESS OF TAIL	BUFFER
41A	41C	41E

เมื่อท่านพิมพ์ตัวอักอักขระ INT 09 จะอยู่ส่วนท้าย เมื่อ INT 16H อ่านค่าของตัวอักขระทำงานในส่วนต้น ในกรณีนี้โปรเซสเซอร์จะทำงานเป็นวงกลม ในส่วนต้นและส่วนท้ายสลับกันไปอย่างต่อเนื่อง เมื่อบัฟเฟอร์ว่าง ส่วนต้นและส่วนท้ายอยู่ที่ตำแหน่งแอดเดรสเดียวกัน ตามตัวอย่างต่อไปนี้ ผู้ใช้กดคีย์ 'abcd' และ Enter คำสั่ง INT 09 จะเก็บค่าตัวอักขระลงในบัฟเฟอร์และไปรออยู่ที่ส่วนท้ายที่ 428 โปรแกรม INT 16H ก็จะทำการอ่านตัวอักขระ 5 ครั้งและไปรอที่ 428 ดังนั้นบัฟเฟอร์ก็จะว่าง

a	b	c	d	0DH	....
¦	¦	¦	¦	¦	¦
41E	420	422	424	426	428

เมื่อบัฟเฟอร์เต็ม ส่วนท้ายก็จะเป็นส่วนหัวทันที ผู้ใช้ก็สามารถพิมพ์ 'fghijklmnopqrs' คำสั่ง INT 09 ก็จะเก็บค่าตัวอักขระเริ่มต้นด้วยที่ตำแหน่ง 428 ในส่วนท้าย จะทำงานเป็นวงกลมในการเก็บข้อมูล และจะเก็ฐค่า 's' ที่ 424 ทันทีก่อนส่วนต้นที่ 426

```
  p   q   r   s   0DH  e   f   g   h   i   j   k   l   m   n   o
  ¦   ¦   ¦   ¦   ¦    ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦
41E 420 422 424  426  428 42A 42C 42E 430 432 434 436 438 43A 43C
```

คำสั่ง INT 09 จะไม่ยอมรับตัวอักขระที่พิมพ์มาเกินในแอดเดรสส่วนต้น จะรับเพียง 15 ตัวอักขระเท่านั้น ถึงแม้ว่าบัฟเฟอร์จะเก็บได้ 16 ตัว ในตัวอย่างต่อไปนี้ ถ้า INT 09 รับตัวอักขระอื่นๆ มันก็จะไปรอที่ส่วนท้ายแอดเดรสเดียวกับส่วนหัว และ INT 16H จะบอกว่าบัฟเฟอร์ว่าง

THE CTRL , SHIFT ,AND ALT KEYS

คำสั่ง INT 09 จะมีไบท์สถานะอยู่ที่ 40:17 ของ BIOS คีย์ Right Shift (bit 0), คีย์ Left shift (bit 1) , คีย์ CTRL (bit 2) และ คีย์ ALT (bit 3) เมื่อท่านกด 1 ในคีย์เหล่านี้ งานใน BIOS ก็จะทำการตรวจสอบบิตที่เป็น 1 เมื่อพบก็จะเคลียบิตเป็น 0 โปรแกรมของท่านสามารถตรวจสอบคีย์เหล่านี้ (Service 2) หรืออ้างอิงโดยตรงในไบท์สถานะ ในส่วนหนึ่งของโปรแกรม COM ที่อ้างโดยตรง

```
BIODATA     SEGMENT   AT 40H          ;LOCATE BIOS DATA AREA
            ORG       17H             ;AND
KBSTATE     DB        ?               ;SATATUS BYTE
BIODATA     ENDS
CODESG      SEGMENT       PARA
            ASSUME        CS:CODESG,DS:BIODATA
            ORG       100H
BEGIN:
            MOV       AX,BIODATA          ;INITIALIZE ADDRESS OF
            MOV       DS,AX               ;BIODATA IN DS
            MOV       AL,KBSTATE          ;GET KEYBOARD STATUS BYTE
            OR        AL,00000011B        ;TEST EITHER SHIFT PRESSED
            JE        XXX
            ...
```

โปรแกรมที่ใช้ SEGMENT AT เพื่อกำหนดข้อมูลใน BIOS ส่วน KBSTATE กำหนดตำแหน่งของคีย์บอร์ดที่ไบท์สถานะ 40:17H ในส่วนของ CODE SEGMENT จะต้องกำหนดแอดเดรสเริ่มแรกของ BIODATA ที่อยู่ใน DS และเก็บไบท์สถานะของคีย์บอร์ดใน AL คำสั่ง OR เป็นการตรวจสอบไบท์ของคีย์ SHIFT ทั้งสอง

สรุป

- การทำงานของคีย์บอร์ดในดอส จะใช้การลริการต่างๆในการแสดงผลหรือไม่แสดงผลบนจอภาพ
- BIOS INT 16H เป็นพื้นฐานการทำงานของคีย์บอร์ด สำหรับการรับข้อมูลจากคีย์บอร์ดมายังบัฟเฟอร์ สำหรับเก็บตัวอักขระ
- พื้นที่เก็บข้อมูล BIOS แอดเดรส 40:1EH เป็นข้อมูลของคีย์บอร์ดบัฟเฟอร์ พื้นที่นี้ใช้พิมพ์ล่วงหน้าได้ 15 ตัว
- INT 09 จะเป็นตัวสแกนรหัสจากคีย์บอร์ด และส่งไปยังลัฟเฟอร์ หรือเช็ต่า SHIFT , CTRL ALT

แบบฝึกหัด

10-1. ตำแหน่งไบท์แรกของของคีย์บอร์ดชิฟท์สถานะใน BIOS คือพื้นที่อะไร

10-2. จงอธิบายการทำงานของ INT 21H ซึ่งเป็นการรับข้อมูลดังต่อไปนี้

a) Srevice 01 b) Service 07

c) Service 08 d) Srevice 0AH

10-3. จงอธิบายถึงความแตกต่างของ INT 16H การบริการ 00H , 01H , 10H

10-4. สแกนรหัสของคีย์ต่อไปนี้คือ

a) UP Arrow b) Down Arrow

c) Home e) PgUp

f) F3 g) Numlock

10-5. ตำแหน่ง BIOS BUFFER ในหน่วยความจำคือตำแหน่งอะไร

10-6. จงเขียนรหัสคำสั่ง ของ ENTER A KEUSTROKE ถ้ากดคีย์ PgUp เซ็ตเคอร์เซอร์ไปที่ตำแหน่งบรรทัด 24 และคอมลัมน์ 0