

Lecture 16

Advanced Keyboard Processing

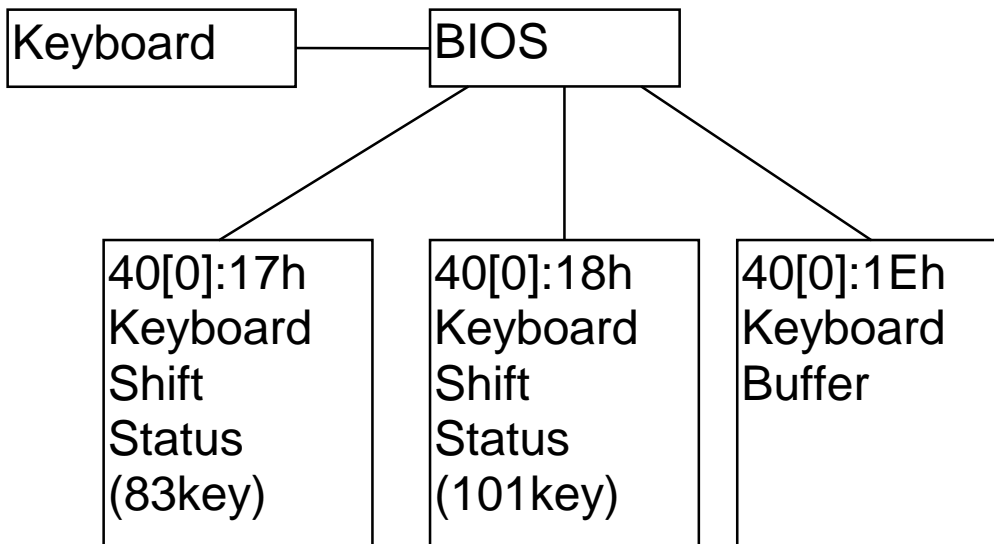
Text:

4th Edition: Chapter 11

5th Edition: Chapter 10

KEYBOARD KEYS

- Letters, numbers and punctuation marks
- Extended function keys
- Control keys



Some DOS Keyboard functions (INT 21h)

01h Keyboard Input With Echo

08h Keyboard Input With No Echo

0Ah Buffered Keyboard Input

0Bh Check Keyboard Status

0Ch Clear Keyboard Buffer, Invoke Function (01,08,0A)

Some BIOS Keyboard Functions (INT 16h)

00h Read a Character (83-key)

02h Return Current Shift Status (83-key)

05h Write to Keyboard Buffer

10h Read Keyboard Character (101-key)

12h Return Current Shift Status (101-key)

Extended Keys and Scan Codes

ASCII codes are used for the characters.

There are other keys (e.g., F2, HOME, ↑)

There are duplicates (+, *, DEL, ...)

When a key is pressed, the keyboard delivers an ASCII code and a SCAN code to the BIOS.

INT 16h, Function 10h would deliver:

	AH SCAN Code	AL ASCII
Lower case 'a'	1E	61
Upper case 'Q'	10	51
Control-x	2D	18
Top row "1"	02	31
Keypad "1"	4F	00
Function key F5	3F	00
Shift-F5	58	00

Extended characters have ASCII values (put in AL) of either 00h or 0Eh.

See Appendix F of the text

[sample of appendix F]

EXAMPLE:

Is a key an extended key?
if not, draw a green box,
if yes, draw a red box.
Exit if the user types "q".

```
include    c:\bp\bin\cs201\clearscr.lib

again:
    mov     ah,10h
    int     16h
    cmp     al,71h    ; if "q"
    je      exit     ; then stop
    cmp     al,00h
    jz      extnd
    cmp     al,0Eh
    jz      extnd
;
; not extended function key- set green
;
    ClearScr 23h    ; set color green
    jmp     again
extnd:
;
; extended function key
;
    ClearScr 43h    ; set color red
    jmp     again
exit:
```

EXAMPLE: Selecting from a menu

```
page 60,132
TITLE P11SELMU (EXE) Select item 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
NINETEEN DB 19 ;Width of menu
MENU DB 0C9H, 17 DUP(0CDH), 0BBH
DB 0BAH, ' Add records ', 0BAH
DB 0BAH, ' Delete records ', 0BAH
DB 0BAH, ' Enter orders ', 0BAH
DB 0BAH, ' Print report ', 0BAH
DB 0BAH, ' Update accounts ', 0BAH
DB 0BAH, ' View records ', 0BAH
DB 0C8H, 17 DUP(0CDH), 0BCH

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 segment
      MOV     DS,AX      ; registers
      MOV     ES,AX
      CALL    Q10CLR     ;Clear screen
      MOV     ROW,BOTROW+2
      MOV     COL,00
      CALL    Q20CURS   ;Set cursor
      MOV     AH,40H    ;Request display
      MOV     BX,01     ;Handle for screen
      MOV     CX,75     ;Number of characters
      LEA    DX,PROMPT ;Prompt
      INT     21H

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     AX,4C00H  ;Exit to DOS
      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, 09H        ;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,10H      ;Request keyboard
           INT      16H        ; input
           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      NINETEEN      ;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,09H        ;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,02H
MOV          BH,00      ;Page 0
MOV          DH,ROW     ;Row
MOV          DL,COL     ;Column
INT          10H
RET
Q20CURS     ENDP
END          BEGIN
```

The Keyboard Buffer and BIOS INT 09h

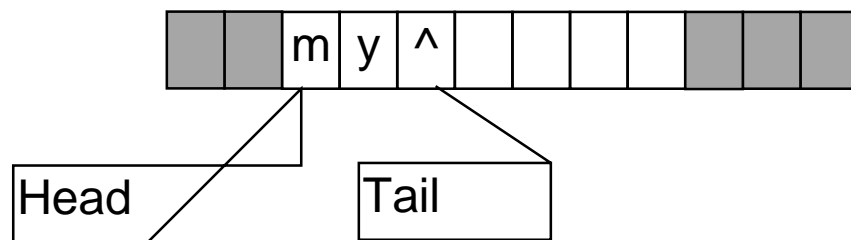
see figure 11-2 (4th edition), 10-3 (5th edition)

The keyboard buffer is a CIRCULAR QUEUE.

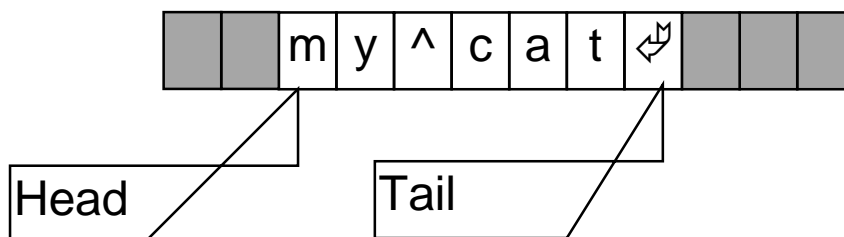
It contains a sequence of 16 words (32 bytes) for the characters typed in and the scan code.

Two other locations to tell where the first and last characters are.

If you type “my^”, the buffer will contain



If you type “cat↵”, the buffer will contain



Or in hex:

