

# 第10章 鍵盤處理功能

## 本章介紹的INT 21H功能

07H: 直接鍵盤輸入, 無回應

08H: 鍵盤輸入, 無回應

0AH: 緩衝鍵盤輸入

0BH: 檢查鍵盤狀況

0CH: 清緩衝區, 呼叫另一鍵盤輸入功能

# 本章介紹的INT 16H功能

03H: 設定 Typematic 速率

05H: 寫到鍵盤 (模擬輸入)

10H: 讀一個字

11H: 檢查是否有按鍵輸入

12H: 取目前的shift狀況

## 鍵盤的按鍵

分三類:

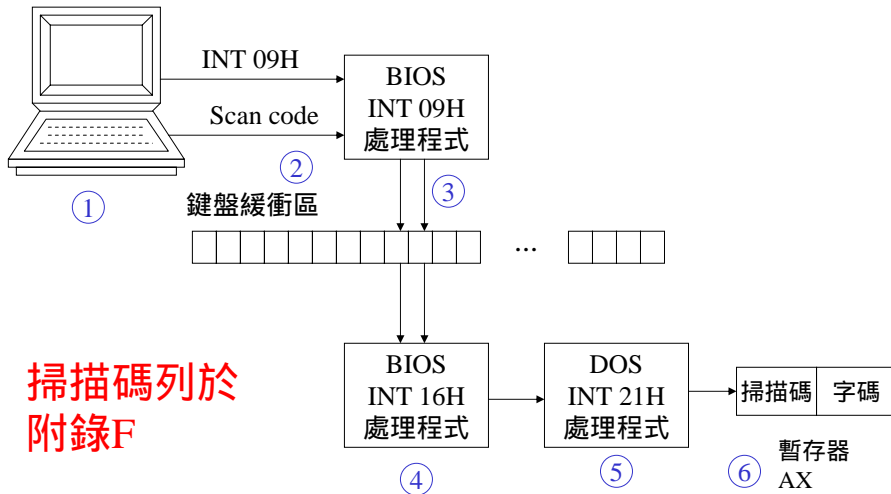
標準字元: 字母、數字和標點符號

延伸功能鍵: <F1>, <F2>, ..., <Del>, <Ins>, ...

控制鍵: <alt>, <ctrl> 和 <shift>

最早的PC只有83個按鍵, 後來增加單獨的游標控制鍵和其它, 增加到104個鍵

# 按鍵輸入的處理



## 鍵盤緩衝區

循環緩衝區(Circular Buffer)

由記憶體地址 **41EH** 起, 32個位元組

**41AH** 存目前緩衝區的開頭位址 (尚未取的第一個字)。每讀走一個字, 41AH內容增加2

**41CH** 存目前緩衝區的結束位址 (下一個字的存入位置)。每寫入一個字, 41CH內容增加2

開頭位址和結束位址相同時, 緩衝區內無資料

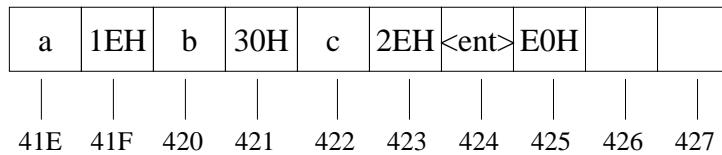
最多可放15個字

# 例子

按入 'abc<enter>'

41AH存41EH

41CH存426H

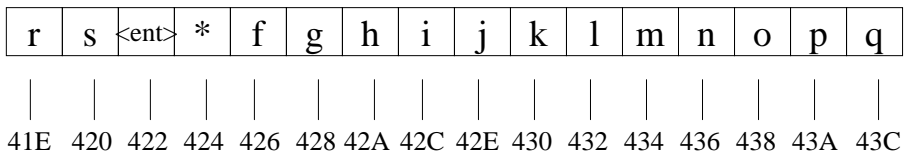


# 緩衝區滿載

上例中4個字已讀走, 又輸入 'fgh...' 15個字

41AH存426H

41CH存424H



## INT 21H function 08H 鍵盤輸入, 無回應

等待鍵盤輸入; 回來時, AL放按鍵的ASCII碼

若是特殊鍵, AL=0; 應再呼叫一次, 可得 Scan code

## INT 21H function 0BH 檢查鍵盤狀況

回來時, AL=0FFH 表示有按鍵  
A=0 表示無按鍵

須呼叫 function 08 (或其它)輸入字碼

## INT 21H function 0CH 清緩衝區, 呼叫另一鍵盤輸入功能

```
MOV    AH,0CH
MOV    AL,function
INT    21H
```

AL的值: 01H, 06H, 07H, 08H, 0AH

## INT 16H function 03H 設定 Typematic 速率

```
MOV    AH,03H
MOV    AL,05H
MOV    BH,repeat_delay
MOV    BL,repeat_rate
INT    16H
```

BH: 0=1/4秒, 1=1/2秒 (Default), 2=3/4秒, 3=1秒  
BL: 0 到 31 (最慢)

## INT 16H function 05H 寫到鍵盤 (模擬輸入)

插入字到鍵盤緩衝區  
CH放ASCII碼, CL放掃描碼

## INT 16H function 10H 讀一個字

檢查鍵盤緩衝區。若有字, ASCII碼放在AL, 掃描碼放在AH。若沒有字, 則等待按鍵輸入

特殊鍵沒有ASCII碼, 只有掃描碼。讀到的ASCII碼是0或0E0H

## INT 16H function 11H 檢查是否有按鍵輸入

- 回來時, ZF=0表示有按鍵輸入; ASCII碼在AL, 掃描碼在AH
- 仍須呼叫讀鍵盤的功能讀入按鍵值
- 若不使用讀鍵盤的功能呼叫讀入按鍵值, 按鍵值仍放在緩衝區內

## INT 16H function 12H 取目前的shift狀況

將BIOS資料區40:17H和40:18H的內容讀到  
AL和AH

AH的位元

- 7: SysReq 按下, 6: Capslock 按下,
- 5: Numlock 按下, 4: ScrollLock 按下
- 3: 右 Alt 按下, 2: 右 Ctrl 按下,
- 1: 左 Alt 按下, 0: 左 Ctrl 按下



## AL的位元

7: Insert active, 6: Capslock active,  
5: Numlock active, 4: ScrollLock active  
3: 右 Alt 按下, 2: 右 Ctrl 按下,  
1: 左 Shift 按下, 0: 右 Shift 按下

例子: 檢查左或右邊的 Shift 是否被按下

```
MOV     AH, 12H
INT     16H
AND     AL, 00000011B
JZ      exit
```

## 打入完整的ASCII字元集合

鍵盤上可打入的只有字母、數字和標點符號等

如何輸入PC增加的特殊符號, 如笑臉、紅心、方塊和表格框線等?

按住ALT鍵, 按右方數字鍵輸入三位數十進制的ASCII碼 (0到255)

讀鍵盤時, 得到的掃描碼是0

# 游標控制鍵的掃描碼

: 48H

: 50H

: 4BH

: 4DH

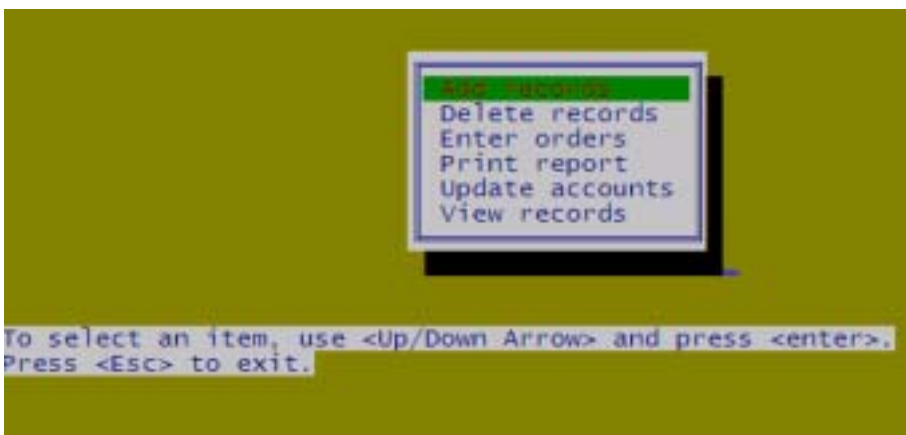
Home: 47H

End: 4FH

Del: 53H

Ins: 52H

## 程式例子: Menu的操作



# 資料定義

```
title a10selmu (exe) select item from menu
;-----
.model small
.stack 256
;-----
.data
toprow equ 8 ;top row of menu
botrow equ 15 ;bottom row of menu
leftcol equ 26 ;left column of menu
attrib db ? ;screen attribute
row db 0 ;screen row
shadow db 19 dup (0dbh) ;shadow characters
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 07, 'To select an item, use <Up/Down Arrow>'
db ' and press <enter>.'
db 13,10, 'Press <Esc> to exit.'
len equ $-prompt
```

# 主程式

```
.code
.386
a10main proc far
mov ax,@data ;initialize segment
mov ds,ax ;registers
mov es,ax
call q10clear ;clear screen
; mov row,botrow+4 ;set row
a20:
call b10menu ;display menu
mov row,toprow+1 ;set row of top item
mov attrib,24h ;set reverse video
call d10disp ;highlight current menu line
call c10input ;provide for menu selection
cmp al,1bh ;escape key pressed?
jne a20 ;no, continue
; mov ax,0600h ;esc pressed (indicates end)
call q10clear ;clear screen
mov ax,4c00h ;end of processing
int 21h
a10main endp
```

# 顯示選單

```
b10menu proc near
pusha ;preserve general register
mov ax,1301h ;request display shadow box
mov bx,060h ;page and attribute
lea bp,shadow ;shadow characters
mov cx,19 ;19 characters
mov dh,toprow+1 ;top row of shadow
mov dl,lefc0l+1 ;left column of shadow
b20:
int 10h
inc dh ;next row
cmp dh,botrow+2 ;all rows displayed?
jne b20 ; no, repeat
mov attrib,71h ;blue on white
mov ax,1300h ;request display menu
movzx bx,attrib ;page and attribute
lea bp,menu ;menu line
mov cx,19 ;length of line
mov dh,toprow ;row
mov dl,lefc0l ;column
b30:
int 10h
add bp,19 ;next menu line
inc dh ;next row
cmp dh,botrow+1 ;all rows displayed?
jne b30 ; no, repeat
;
mov ax,1301h ;request display prompt
movzx bx,attrib ;page and attribute
lea bp,prompt ;prompt line
mov cx,len ;length of line
mov dh,botrow+4 ;screen row,
mov dl,0 ; column
int 10h
popa ;restore registers
ret
b10menu endp
```

# 按鍵輸入處理

```
c10input proc near
; pusha ;preserve general registers
c20:
mov ah,10h ;request keyboard
int 16h ; input
cmp ah,50h ;down arrow?
je c30
cmp ah,40h ;up arrow?
je c40
cmp al,00h ;enter key?
je c90
cmp al,10h ;esc key?
je c90
jmp c20
c30:
mov attrib,71h ;blue on white
call #100isp ;set old line to normal video
inc row ;increment for next row
cmp row, botrow-1 ;past bottom row
jbe c50 ; no, ok
mov row, toprow+1 ; yes, reset
jmp c20
c40:
mov attrib,71h ;blue on white
call #100isp ;set old line to normal video
dec row
cmp row,toprow+1 ;below top row?
jae c50 ; no, ok
mov row,botrow-1 ; yes, reset
c50:
mov attrib,17h ;white on blue
call #100isp ;set new line to reverse side
jmp c20
c90:
popa ;restore registers
ret
c10input endp
```

# 顯示控制

```
d10disp proc near
    pusha                    ;preserve general registers
    mov     al,row           ;row tells which line to set
    sub     al,toprow
    mov     bl,19            ;imul ax,19
    mul     bl               ;multiply by length of line
    lea    si,menu+1        ; for select menu line
    add     si,ax
;
    mov     ax,1300h        ;request display prompt
    movzx  bx,attrib        ;page and attribute
    mov     bp,si           ;prompt line
    mov     cx,17           ;length of line
    mov     dh,row         ;screen row,
    mov     dl,lefcol+1    ; column
    int     10h
    popa                    ;restore registers
    ret
d10disp endp
q10clear proc near
    pusha                    ;preserve general registers
    mov     ax,0600h        ;blue on brown
    mov     bh,61h
    mov     cx,0
    mov     dx,184fh        ;full screen
    int     10h            ;call bios
    popa                    ;restore registers
    ret
q10clear endp
```

## 程式例子: 檢查鍵盤控制鍵狀態

```
title    a11kbsta (COM) testing alt, shift, and ctrl status
.model   small
.stack   64
.data

biodata segment at 40h ;locate bios data area
org      17h           ; and
kbstat   db           ? ; status byte
biodata ends
;-----
cr       equ          0dh ;carriage return
lf       equ          0ah ;line feed
altkey   db           'Alt key pressed ', cr,lf
ctrlkey  db           'Ctrl key pressed ',cr,lf
shiftkey db           'Shift key pressed',cr,lf
.286
```

# 主程式

```
.code
a10main proc far
mov ax,biodata ;initialize seg. address
mov es,ax ; of BIODATA in ES

a20:
mov ah,10h
int 16h
cmp al,0dh
je a90
mov bl,es:kbstat ;get keyboard status byte
test bl,00000011B ;shift+char pressed?
jz a30 ; no, bypass
lea bp,shiftkey ;request display
call d10display ; shift message

a30:
test bl,00000100B ;ctrl+char pressed?
jz a40 ; no, bypass
lea bp,ctrlkey ;request display
call d10display ; ctrl message

a40:
test bl,00001000B ;alt+char pressed?
jz a20 ; no, bypass
lea bp,altkey ;request display
call d10display ; alt message
jmp a20 ;repeat

a90:
mov ax,4c00h ;end processing
int 21h
a10main endp
```

## 顯示控制

```
; display message for alt, ctrl, and
; shift if key is pressed.
d10display proc near ;BP set on entry
pusha ;preserve registers
push es
mov ax,@data ;set up for ES:BP
mov es,ax ; for address of data
mov ax,1301h ;request display
mov bx,0071h ;page and attribute
mov cx,17 ;length of string
mov dx,1008h ;row and column
int 10h
pop es
popa ;restore registers
ret
d10display endp
```