Part D 資料處理

第11章 字串的處理
第12章 二進制算術
第13章 十進制算術
第14章 表格的處理

第11章 字串的處理

字串指令
Repeat 前置指令

字串指令

動作	指令	隱含的運算元
移動	MOVS	ES:DI, DS:SI
載入	LODS	AX, DS:SI
貯存	STOS	ES:DI, AX
比較	CMPS	DS:SI, ES:DI
掃描	SCAS	ES:DI, AX

方向旗標 DF

- 一個字串指令只處理一個資料
- 字串指令的最後一個字母(B, W 或D)指示資料的大小
- 字串指令執行後, 相關的索引暫存器將增加或減少, 指到下一個資料
- DF=0 時, 索引增加; DF=1 時, 索引減少
- 指令STD和CLD用以設定和清除DF

REP 前置指令

- 放在字串指令之前, 使那字串指令重覆執行。次數放在CX
- REP: CX減1。若CX不是0, 重覆字串指令; 否則, 停止字串指令。放在 MOVS, LODS, STOS 之前
- REPE: CX減1。若CX不是0且ZF=1, 重覆字串指令; 否則, 停止字串指令。放在 CMPS 之前
- REPNE: CX減1。若CX不是0且ZF=0, 重覆字串指令; 否則, 停止字串指令。放在 SCAS 之前

MOVS 指令

```
MOV     CX,10
LEA     DI,STRING2
LEA     SI,STRING1
REP     MOVSW
```

```
        JCXZ        J90
J30:
        MOV         AX,[SI]
        MOV         [DI],AX
        ADD         DI,2
        ADD         SI,2
        LOOP        J30
J90:
```

MOVS 的例子

```
string1   db   'Interstellar'   ;data items
string2   db   12 dup (' ')
string3   db   12 dup (' ')
; ...
  mov     ax,@data    ;initialize
  mov     ds,ax       ; segment
  mov     es,ax       ; registers
                      ;use of MOVSB:
  cld                 ;left to right
  mov     cx,12       ;move 12 bytes
  lea     di,string2  ; string 1 to string2
  lea     si,string1
  rep     movsb
;
  cld                 ;use of MOVSW
  mov     cx,6        ;left t right
  lea     di,string3  ;move 6 words
  lea     si,string2  ;string2 to string3
  rep     movsW
; ...
```

LODSB 的例子

```
.DATA
HEADG1      DB    ‘Cybernauts’
HEADG2      DB    10 DUP (20H)
.CODE
A10MAIN     PROC
      MOV         AX,@DATA
      MOV         DS,AX
      MOV         ES,AX
      CLD
      MOV         CX,10
      LEA         SI,HEADG1
      LEA         DI,HEADG2+9
A20:
      LODSB
      MOV         [DI],AL
      DEC         DI
      LOOP        A20
      MOV         AX,4C00H
      INT         21H
A10MAIN     ENDP
      END         A10MAIN
```

STOSW 的例子

```
.DATA
HEADG1    DB   ‘Cybernauts’.CODE
A10MAIN   PROC
     MOV       AX,@DATA
     MOV       DS,AX
     MOV       ES,AX
     CLD
     MOV       AX,2020H
     MOV       CX,5
     LEA       DI,HEADG1
     REP       STOSW
     MOV       AX,4C00H
     INT       21H
A10MAIN   ENDP
     END       A10MAIN
```

程式例子: 簡單的Editor

- 只有一行, 30個字
- 游標左右移動、移到頭尾
- 刪除字, 更改(加入)字

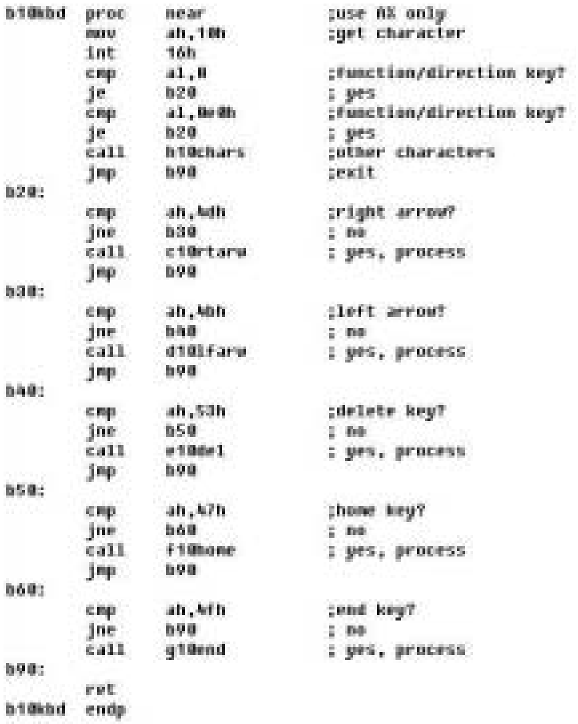
資料定義

```
        title   a11edit (EXE) editing features
        .model small
        .stack 64
        .data
indent  equ     24              ;screen indent
leftlim equ     0               ;left limit of data
ritlim  equ     29              ;right limit of data
nochars equ     30              ;length of data
col     db      0               ;screen column
row     db      10              ;screen row
datastr db      'abcdefghijklmno'       ;area for editing data
        db      'pqrstuvwxyzABCD',20h
.386
```

主程式

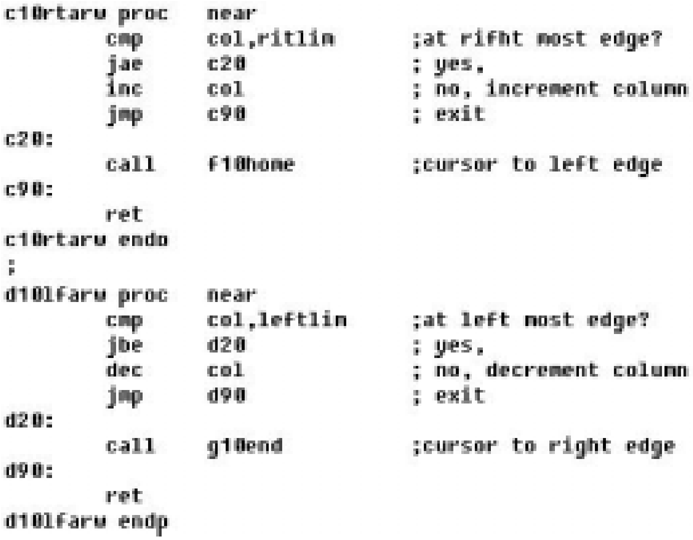
```
        .code
a10main proc    far
        mov     ax,@data        ;initialize segment
        mov     ds,ax           ; registers
        mov     es,ax
        call    q10clr          ;clear screen
        call    q20curs         ;set cursor start
        call    q30disp         ;display string
a30:
        call    q20curs         ;reset cursor start
        call    b10kbd          ;get KB character
        cmp     ah,01h          ;ESC key?
        jne     a30             ; no, continue
        mov     ax,0600h        ; yes, quit
        call    q10clr          ;clear screen
        mov     ax,4c00h        ;end of processing
        int     21h
a10main endp
```

鍵盤輸入



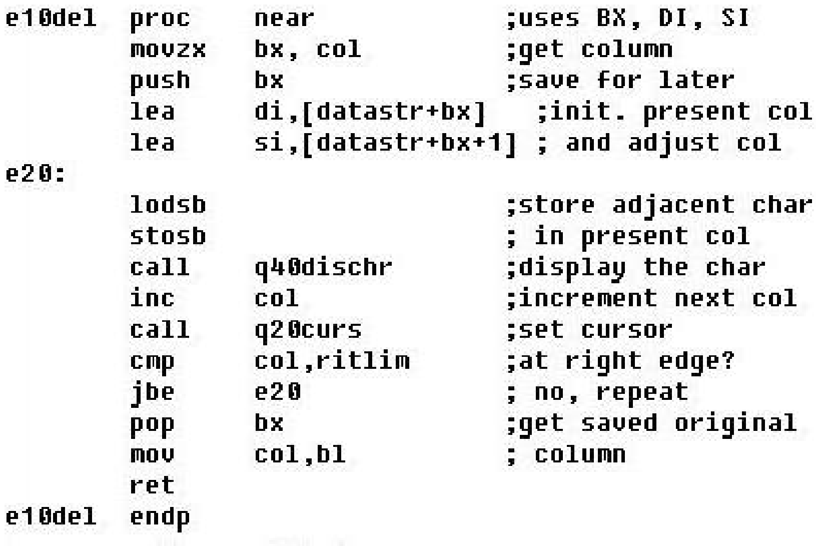
```
b10kbd  proc    near            ;use AX only
        mov     ah,10h          ;get character
        int     16h
        cmp     al,0            ;function/direction key?
        je      b20             ; yes
        cmp     al,0e0h         ;function/direction key?
        je      b20             ; yes
        call    b10chars        ;other characters
        jmp     b90             ;exit
b20:
        cmp     ah,4dh          ;right arrow?
        jne     b30             ; no
        call    c10rtaru        ; yes, process
        jmp     b90
b30:
        cmp     ah,4bh          ;left arrow?
        jne     b40             ; no
        call    d10lfaru        ; yes, process
        jmp     b90
b40:
        cmp     ah,53h          ;delete key?
        jne     b50             ; no
        call    e10del          ; yes, process
        jmp     b90
b50:
        cmp     ah,47h          ;home key?
        jne     b60             ; no
        call    f10home         ; yes, process
        jmp     b90
b60:
        cmp     ah,4fh          ;end key?
        jne     b90             ; no
        call    g10end          ; yes, process
b90:
        ret
b10kbd  endp
```

左右箭頭鍵的處理



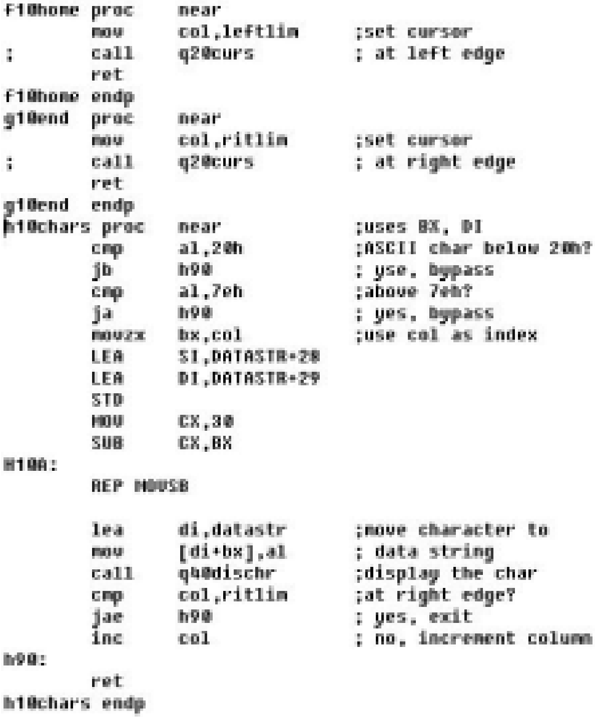
```
c10rtaru proc   near
        cmp     col,ritlin      ;at rifht most edge?
        jae     c20             ; yes,
        inc     col             ; no, increment column
        jmp     c90             ; exit
c20:
        call    f10home         ;cursor to left edge
c90:
        ret
c10rtaru endp
;
d10lfaru proc   near
        cmp     col,leftlin     ;at left most edge?
        jbe     d20             ; yes,
        dec     col             ; no, decrement column
        jmp     d90             ; exit
d20:
        call    g10end          ;cursor to right edge
d90:
        ret
d10lfaru endp
```

DEL鍵的處理



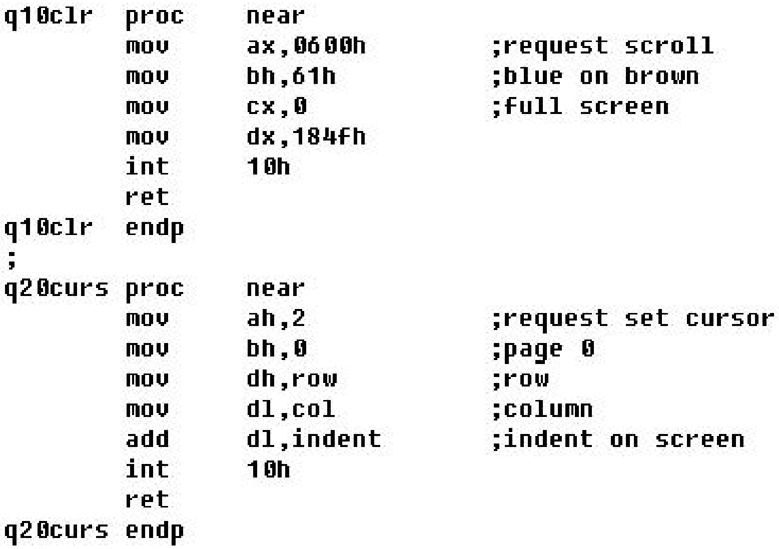
```
e10del  proc    near               ;uses BX, DI, SI
        movzx   bx, col            ;get column
        push    bx                 ;save for later
        lea     di,[datastr+bx]      ;init. present col
        lea     si,[datastr+bx+1] ; and adjust col
e20:
        lodsb                      ;store adjacent char
        stosb                      ; in present col
        call    q40dischr          ;display the char
        inc     col                ;increment next col
        call    q20curs            ;set cursor
        cmp     col,ritlim         ;at right edge?
        jbe     e20                ; no, repeat
        pop     bx                 ;get saved original
        mov     col,bl             ; column
        ret
e10del  endp
```

HOME、END 和文字鍵的處理



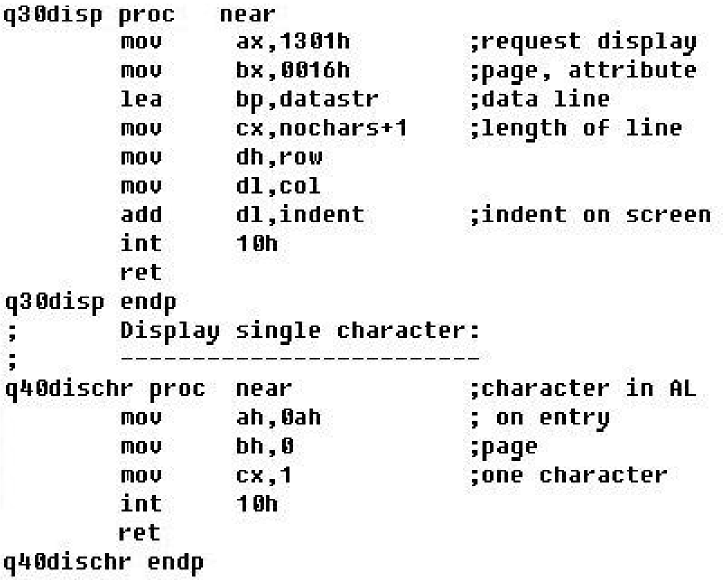
```
f10home proc    near
        mov     col,leftlim         ;set cursor
;       call    q20curs             ; at left edge
        ret
f10home endp
g10end  proc    near
        mov     col,ritlim          ;set cursor
;       call    q20curs             ; at right edge
        ret
g10end  endp
h10chars proc   near                ;uses BX, DI
        cmp     al,20h              ;ASCII char below 20h?
        jb      h90                 ; yse, bypass
        cmp     al,7eh              ;above 7eh?
        ja      h90                 ; yes, bypass
        movzx   bx,col              ;use col as index
        LEA     SI,DATASTR+28
        LEA     DI,DATASTR+29
        STD
        MOV     CX,30
        SUB     CX,BX
H10A:
        REP MOVSB

        lea     di,datastr          ;move character to
        mov     [di+bx],al          ; data string
        call    q40dischr           ;display the char
        cmp     col,ritlim          ;at right edge?
        jae     h90                 ; yes, exit
        inc     col                 ; no, increment column
h90:
        ret
h10chars endp
```

清螢幕、設游標位置



```
q10clr  proc    near
        mov     ax,0600h        ;request scroll
        mov     bh,61h          ;blue on brown
        mov     cx,0            ;full screen
        mov     dx,184fh
        int     10h
        ret
q10clr  endp
;
q20curs proc    near
        mov     ah,2            ;request set cursor
        mov     bh,0            ;page 0
        mov     dh,row          ;row
        mov     dl,col          ;column
        add     dl,indent       ;indent on screen
        int     10h
        ret
q20curs endp
```

顯示文字



```
q30disp proc   near
        mov     ax,1301h        ;request display
        mov     bx,0016h        ;page, attribute
        lea     bp,datastr      ;data line
        mov     cx,nochars+1    ;length of line
        mov     dh,row
        mov     dl,col
        add     dl,indent       ;indent on screen
        int     10h
        ret
q30disp endp
;       Display single character:
;       -------------------------
q40dischr proc  near            ;character in AL
        mov     ah,0ah          ; on entry
        mov     bh,0            ;page
        mov     cx,1            ;one character
        int     10h
        ret
q40dischr endp
```

CMPS的例子

```
string1 db     'Interstellar'  ;dataitems
string2 db     'Interstellar'
string3 db     12 dup (' ')
; ...
     cld            ;left to right
     mov  cx,12     ;initialize for 12 bytes
     lea  di,string2;ES:DI
     lea  si,string1;DS:SI
     repe cmpsb     ;compare string1 : string2
     jne  exit      ; not equal, bypass
; ...          ; equal
```

比較string2和string3

```
     mov  cx,12     ;initialize for 12 bytes
     lea  di,string3
     lea  si,string2
     repe cmpsb     ;compare string2 : string3
     je   exit      ; equal, exit
; ...          ; not equal
```

比較文字的大小(用以排序)

- 應一字一字比 (使用CMPSB)

例子: 比較 John 和 Jean
第一字 ‘J’ 相等, 第二字 ‘o’ 比 ‘e’ 大

- 若一次比兩個字 (使用CMPSW)

第一次將比較 ‘oJ’ 和 ‘eJ’, 還是 John 較大

- 但若比較 ‘SAMUEL’ 和 ‘ARNOLD’

一字一字比, SAMUEL > ARNOLD
一次比兩字, ‘RA’>’AS’

SCAS的例子

```
string1 db'Interstellar'  ;data item
; ...
     cld            ;left to right
     mov  al,'r'    ;scan string1 for 'r'
     mov  cx,12     ; 12 characters
     lea  di,string1;ES:DI
     repne scasb
     je   exit      ;found
; ...          ;not found
```

以空格取代星號

```
datalen equ    13   ;length of test data
testdata db    'Extra*innings'
; ...
  cld            ;set left to right
  mov  al,'*'      ;search character
  mov  cx,datalen   ;length of test data
  lea  di,testdata ;address of testdata (ES:DI)
  repne scasb  ;scan testdata
  jne  exit         ;character found?
  mov  byte ptr[di-1],20h
```

複製一段字

```
pattern db '|****|'   ;pattern to be dulicated
disparea db    42 dup (?); display area
; ...
     cld            ;left to right operation
     mov  cx,21          ;21 words
     lea  di,disparea    ;destination
     lea  si,pattern     ;source
     rep  movsw          ;move characters
```