

第14章表格的處理

資料的安排

查表

排序

定義表格

爲方便處理, 通常每一項的格式相同, 且經過排序

例子:

MONTH_TBL DB 'Jan', 'Feb', 'Mar', ..., 'Dec'

CUST_TBL DB 205, 208, 209, 212, 215, 224, ...

每一項又分成多個小項, 各小項有不同的格式

STOCK_TBL DB 12, 'Computers', 14, 'Paper....', 17, 'Diskettes', ...

```

STOCK_NO  DB   '05','10','12',...
STOCK_DESC DB   'Excavators','Lifters  ','Presses  '

STOCK_TBL DB   '05','Excavators'
           DB   '10','Lifters  '
           DB   '12','Presses  '

```

TYPE、LENGTH和SIZE運算

```

RAIN_TBL DW  12 DUP (?)
;   ...
MOV AX, TYPE RAIN_TBL      ;2
MOV BX,LENGTH RAIN_TBL    ;12
MOV CX, SIZE RAIN_TBL     ;24

```

直接指到一項的地址

```
LEN_ENTRY EQU 9
MONTH_IN DB '12'
MONTH_TBL DB 'January ', 'February ', 'March '
          DB 'April ', 'May ', 'June '
          DB 'July ', 'August ', 'September'
          DB 'October ', 'November ', 'December '
.386 ;
; Convert ASCII month to binary
XOR WORD PTR MONTH_IN, 3030H
MOVZX AX, MONTH_IN
IMUL AX, 10
ADD AL, MONTH_IN+1
; Locate month in table
DEC AL
IMUL AX, LEN_ENTRY
LEA BP, MONTH_TBL
ADD BP, AX
; Display alpha month
MOV AX, 1301H
MOV BX, 0016H
MOV CX, LEN_ENTRY
MOV DX, 0812H
INT 10H
; ...
```

例子 2: 月和星期名稱表

```
LEN_ENTRY EQU 9
DAYOFMON DW 0
SAVEDAY DB ?
SAVEMON DB 0
TEN DB 10
ROW DB 10
COLUMN DB 30
DAYS_TBL DB 'Sunday ', 'Monday '
          DB 'Tuesday ', 'Wednesday'
          DB 'Thursday ', 'Friday '
          DB 'Saturday '
MONTH_TBL DB 'January ', 'February '
          DB 'March ', 'April '
          DB 'May ', 'June '
          DB 'July ', 'August '
          DB 'September', 'October '
          DB 'November ', 'December '
```

```

.386
    MOV  AH, 2AH
    INT  21H
    MOV  SAVEMON, DH
    MOV  SAVEDAY, DL
;
    MOV  AH, 0
    IMUL AX, LEN_ENTRY
    LEA  BP, DAYS_TBL
    ADD  BP, AX
    MOV  CX, LEN_ENTRY
    CALL B10DISPLAY
;
    MOVZX AX, SAVEMON
    DEC  AX
    IMUL AX, LEN_ENTRY
    LEA  BP, MONT_TBL
    ADD  BP, AX
    MOV  CX, LEN_ENTRY
    CALL B10DISPLAY

    MOVZX AX, SAVEDAY
    DIV  TEN
    OR   AX, 3030H
    MOV  DAYOFMON, AX
    LEA  BP, DAYOFMON
    MOV  CX, 2
    CALL B10DISPLAY
; ...
B10DISPLAY PROC      NEAR
    PUSHA
    MOV  AX, 1301H
    MOV  BX, 0016H
    MOV  DH, ROW
    MOV  DL, COLUMN
    INT  10H
    INC  ROW
    POPA
    RET
B10DISPLAY ENDP

```

查表

若表格已排序, 要找尋特定的一項。

比較的結果決定是否找到或要不要繼續找

- 要找的值比表中的小, 表示找不到。表格後面的各項都大於要找的, 不須繼續找了。
- 要找的值與表中的相等, 表示找到了。
- 要找的值比表中的大, 表示須繼續找

```
LEN_STKNO EQU 2
LEN_DESCR EQU 10
STOCKN_IN DB '12'
STOCK_TBL DB '05', 'Excavators'
           DB '10', 'Lifters'
           DB '12', 'Presses'
           DB '15', 'Valves'
           DB '23', 'Processors'
           DB '27', 'Pumps'
;-----
; ...
MOV CX, 6
LEA SI, STOCK_TBL
A20:
MOV AL, STOCKN_IN
CMP AL, [SI]
JNE A30
MOV AL, STOCKN_IN+1
CMP AL, [SI+1]
JE A50
```

```

A30:      JB   A40
          ADD  SI, LEN_STKNO
          ADD  SI, LEN_DESCR
          LOOP A20
A40:      ;
          ;...
          JMP  A90
A50:      INC  SI
          INC  SI
          MOV  AX, 1301H
          MOV  BP, SI
          MOV  BX, 0061H
          MOV  CX, LEN_DESCR
          MOV  DX, 0812H
          INT  10H
A90:      ;...

```

有範圍的表格

例子: 所得稅率表

收入	稅率	調整值
0-1,000	.1	0
1,000.01-2,500	.15	50
2,500.01-4,250	.18	125
4,250.01-6,000	.20	260
6,000.01以上	.23	390

所得稅=收入×稅率－調整值

```

TAXTBL      DD  100000,10,0
                DD  250000,15,5000
                DD  425000,18,12500
                DD  600000,20,26000
                DD  999999,23,39000

```

查表時, 比較輸入值與表格中的收入欄
 若輸入值較高, 應比較下一項
 輸入值較低, 那一項就是適用值

使用字串比較指令來查表

```

LEN_STKNO   EQU 3
LEN_DESCR   EQU 10
STOCKN_IN   DB  '123'
STOCK_TBL   DB  '035', 'Excavators'
                DB  '038', 'Lifters'
                DB  '049', 'Presses'
                DB  '102', 'Valves'
                DB  '123', 'Processors'
                DB  '127', 'Pumps'
                DB  '999', 10 DUP (' ')

```

```

        CLD
        LEA     DI, STOCK_TBL
A20:    MOV     CX, LEN_STKNO
        LEA     SI, STOCKN_IN
        REPE   CMPSB
        JE     A30
        JB     A40
        ADD    DI, CX
        ADD    DI, LEN_DESCR
        JMP    A20

```

```

A30:    MOV     AX, 1301H
        MOV     BP, DI
        MOV     BX, 0061H
        MOV     CX, LEN_DESCR
        MOV     DX, 0812H
        INT    10H
        JMP    A90
A40:    ; DISPLAY ERROR MESSAGE
;
A90:    ; ...

```


每一項不固定長度

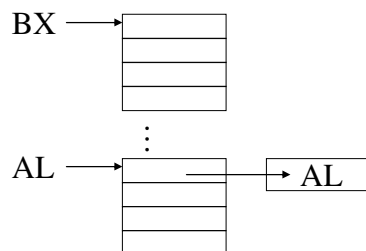
- 可以節省記憶體位置
- 兩項之間要放一個間隔符號(不會出現的數碼, 如0, 0FFH等)
- 查表時, 若不符合, 應先找到下一項的開始位置(由當時位置一個字一個字往下找, 找到間隔符號後, 下一個字即是下一項的開始位置)

XLAT 指令

[label:] XLAT

隱含的運算元: **BX** 指到轉換表的第一個位置, **AL** 指示第幾項

結果: 將指定位置的表格內容放到**AL**



例子: ASCII 數字轉成EBCDIC

```
.MODEL SMALL
.CODE
ORG 100H
BEGIN:
    JMP A10MAIN
;-----
ASCNO DB '-31.5 '
EBCNO DB 6 DUP (' ')
XLATAB DB 45 DUP (40H)
        DB 60H,4BH,40H
        DB 0F0H,0F1H,0F2H,0F3H,0F4H
        DB 0F5H,0F6H,0F7H,0F8H,0F9H
        DB 198 DUP (40H)

A10MAIN PROC NEAR
    LEA SI,ASCNO
    LEA DI,EBCNO
    MOV CX,6
    LEA BX,XLATAB
A20:
    LODSB
    XLAT
    STOSB
    LOOP A20
    ;...
    MOV AX,4C00H
    INT 21H
A10MAIN ENDP
```

例子: 顯示十六進值和ASCII字

- 在螢幕上顯示0-255的十六進值和所代表的文字
- 先在資料區中準備好一行16個數碼的顯示(80個字; 2個十六進值, 空一格, 一個ASCII字, 再空一格)
- 控制碼 (如換行和回頭等, 數碼 07-0FH)

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F

資料定義和主程式

```
ROW      DB      2
DISPROW DB      16 DUP(5 DUP (' '))
HEXCTR   DB      0
XLATAB   DB      '0123456789ABCDEF'
.386     ;...
         CALL    clear_screen
         LEA    SI,DISPROW
A20:
         CALL    B10HEX
         CALL    C10DISPLY
         CMP    HEXCTR,0FFH
         JE     A90
         INC   HEXCTR
         JMP   A20
         ;...
```

轉換成十六進制

```
B10HEX PROC NEAR
          MOVZX AX,HEXCTR
          SHR  AX,4
          LEA  BX,XLATAB
          XLAT
          MOV  [SI],AL
          MOV  AL,HEXCTR
          AND  AL,0FH
          XLAT
          MOV  [SI]+1,AL
          RET
B10HEX ENDP
```

顯示

```
C10DISPLY      PROC    NEAR
                MOV     AL,HEXCTR
                MOV     [SI]+3,AL
                CMP     AL,07H
                JB      C20
                CMP     AL,10H
                JAE     C20
                MOV     BYTE PTR [SI]+3,20H
C20:            ADD     SI,5
                LEA     DI,DISPROW+80
                CMP     SI,DI
                JNE     C90
                MOV     AX,1300H
                MOV     BX,0031H
                LEA     BP,DISPROW
                MOV     CX,80
                MOV     DH,ROW
                MOV     DL,0
                INT     10H
                INC     ROW
                LEA     SI,DISPROW
C90:            RET
C10DISPLY      ENDP
```

Bubble sort

- 依次比較相鄰兩項, 若順序不對則交換位置。完成一輪比較後, 最大(或最小, 視順序是由小到大或由大到小而定)項將移到最後
- 重覆比較動作。第二輪比較後, 第二大(或第二小)得項移到最後第二項。...

L20: 設 SWAP 為 NO
指標指到表格開始位置

L30: 這一項 > 下一項?
是: 交換兩項, 設SWAP為YES
指標增加1, 指到下一項
已指到最後一項?
否: 回到L30
是: SWAP=YES?
是: 跳回L20 (繼續排序)
否: 結束

例子

圖14-6

- 輸入多個名字
- 排序
- 顯示

資料定義

```
.MODEL SMALL
.STACK 64
.DATA
LEN_NAME EQU 20
ENDADDR DW ?
MESSG1 DB 'Name? '
NAMECTR DB 0
NAMESAVE DB 20 DUP (?)
NAME_TBL DB 30 DUP (20 DUP (' '))
ROW DB 0
SWAPPED DB 0
NAMEPAR LABEL BYTE
MAXNLEN DB 21
NAMELEN DB ?
NAMEFLD DB 21 DUP (' ')
.386 ;-----
```

主程式

```
.CODE
A10MAIN PROC FAR
MOV AX,@DATA
MOV DS,AX
MOV ES,AX
CLD
CALL Q10CLEAR
LEA DI,NAME_TBL
A20:
CALL B10ENTER
CMP NAMELEN,0
JE A30
CMP NAMECTR,30
JE A30
CALL C10STORE
JMP A20
A30:
CALL Q10CLEAR
CMP NAMECTR,1
JBE A90
CALL D10SORT
CALL F10NAMES
A90:
MOV AX,4C00H
INT 21H
A10MAIN ENDP
```

輸入名字

```
B10ENTER      PROC    NEAR
              LEA     BP,MESSG1
              MOV     CX,6
              CALL    G10DISPLY
              INC     ROW
              MOV     AH,0AH
              LEA     DX,NAMEPAR
              INT     21H
              MOVZX   BX,NAMELEN
              MOVZX   CX,MAXNLEN
              SUB     CX,BX

B20:          MOV     NAMEFLD[BX],20H
              INC     BX
              LOOP    B20
              RET

B10ENTER      ENDP
;
C10STORE      PROC    NEAR
              INC     NAMECTR
              CLD
              LEA     SI,NAMEFLD
              MOV     CX,LEN_NAME
              REP     MOUSB
              RET

C10STORE      ENDP
```

排序

```
D10SORT PROC    NEAR
          SUB     DI,40
          MOV     ENDADDR,DI

D20:     MOV     SWAPPED,0
          LEA     SI,NAME_TBL

D30:     MOV     CX,LEN_NAME
          MOV     DI,SI
          ADD     DI,LEN_NAME
          MOV     AX,DI
          MOV     BX,SI
          REPE   CMPSB
          JBE     D40
          CALL    E1-XCHNG

D40:     MOV     SI,AX
          CMP     SI,ENDADDR
          JBE     D30
          CMP     SWAPPED,0
          JNZ     D20
          RET

D10SORT ENDP
```


交換

```
E10XCHNG      PROC      NEAR
               MOV      CX,LEN_NAME
               LEA     DI,NAME$AVE
               MOV     SI,BX
               REP  MOUSB
               MOV     CX,LEN_NAME
               LEA     DI,BX
               REP  MOUSB
               MOV     CX,LEN_NAME
               LEA     SI,NAME$AVE
               REP  MOUSB
               MOV     SWAPPED,1
               RET
E10XCHNG      ENDP
```

顯示

```
F10NAMES      PROC      NEAR
               MOV     ROW,0
               LEA     BP,NAME_TBL
F20:           MOV     CX,LEN_NAME
               CALL    G10DISPLY
               INC     ROW
               ADD     BP,LEN_NAME
               DEC     NAMECTR
               JNZ     F20
               RET
F10NAMES      ENDP
;
G10DISPLY     PROC      NEAR
               MOV     AX,1301H
               MOV     BX,0016H
               MOV     DH,ROW
               MOV     DL,10
               INT     10H
               RET
G10DISPLY     ENDP
```

地址表

高階語言的case

```
CODE DB    ?  
  
...  
CMP  CODE, 1  
JE   CODE1_RTNE  
CMP  CODE, 2  
JE   CODE2_RTNE  
CMP  CODE, 3  
JE   CODE3_RTNE  
  
...
```

可以寫成

```
ADDRTBL DW  CODE1_RTNE  
         DW  CODE2_RTNE  
         DW  CODE3_RTNE  
  
...  
MOVZX BX, CODE  
DEC   BX  
SHL  BX, 1  
JMP  [ADDRTBL+BX]
```

Linked list

Offset	Part No.	Price	Next item
0000	0103	12.50	0024
0008	1720	08.95	0016
0016	1827	03.75	0000
0024	0120	13.80	0032
0032	0205	25.00	0008

```
.MODEL    SMALL
.STACK    64
.DATA
PARTNO    DB    '1720'
LINKLST   DB    '0103'
          DW    1250,24
          DB    '1720'
          DW    0895,16
          DB    '1827'
          DW    0375,00
          DB    '0120'
          DW    1380,32
          DB    '0205'
          DW    2500,08
```

```
.CODE
A10MAIN   PROC FAR
          MOV    AX,@DATA
          MOV    DS,AX
          MOV    ES,AX
          CALL   B10LINK
          MOV    AX,4C00H
          INT    21H
A10MAIN   ENDP
```

```

B10LINK      PROC   NEAR
              CLD
              LEA   DI, LINKLST
B20:         MOV   CX, 4
              LEA   SI, PARTNO
              REPE CMPSB
              JE    B30
              JB    B40
              ADD   DI, CX
              ADD   DI, 2
              MOV   DX, [DI]
              LEA   DI, LINKLST
              ADD   DI, DX
              CMP   DX, 0
              JNE   B20
              JMP   B40
B30:         ; ITEM FOUND
              JMP   B90
B40:         ; DISPLAY ERROR MESSAGE
B90:         RET
B10LINK      ENDP
              EDN   A10MAIN

```

二維陣列

3×5 陣列

```
DATA_ARRAY  DW   3 DUP (5 DUP (?))
```

元素 [ROW, COL] 的位置 (ROW和COL
都由0開始)

$2 \times (\text{ROW} \times 5 + \text{COL})$

	0	1	2	3	4
0					
1					
2					

例子：
計算二維陣列各行和各列的和

圖14-7

資料定義

```
.MODEL SAML
.STACK 64
.DATA
SPRD_TBL      DW      05,03,06,04,10,0
               DW      04,05,01,09,06,0
               DW      06,02,0,10,11,0
               DW      10,07,03,05,02,0
               DW      0,0,0,0,0,0
COUNT      DB      ?
COLS        EQU     12
ROWS        EQU     10
NOCOLS      EQU     6
NOROWS      EQU     5
COLTOT      EQU     NOCOLS*(NOROWS-1)*2
;-----
.386
```

主程式

```
.CODE
A10MAIN PROC FAR
    MOV AX,@DARA
    MOV DS,AX
    MOV ES,AX
    CALL B10HORZL
    CALL C10VERTL
;    ...
    MOV AX,4C00H
    INT 21H
A10MAIN ENDP
```

水平相加

```
B10HORZL PROC NEAR
    PUSHA
    MOV COUNT,NOROWS-1
    LEA DI,SPRD_TBL
    LEA SI,SPRD_TBL
    ADD SI,ROWS
B10:
    MOV CX,NOCOLS-1
B20:
    MOV AX,[DI]
    ADD [SI],AX
    ADD DI,2
    LOOP B20
    ADD DI,2
    ADD SI,COLS
    DEC COUNT
    JNZ B10
    POPA
    RET
B10HORZL ENDP
```

垂直相加

```
C10VERTL      PROC      NEAR
              PUSHA
              MOV      COUNT,NOCOLS-1
              MOV      BX,0
              LEA     SI,[SPRD_TBL+COLTOT]
C10:          LEA     DI,SPRD_TBL
              ADD     DI,BX
              MOV     CX,NOROWS-1
C20:          MOV     AX,[DI]
              ADD     [SI],AX
              ADD     DI,COLS
              LOOP    C20
              ADD     BX,2
              ADD     SI,2
              DEC     COUNT
              JNZ     C10
              POPA
              RET
C10VERTL      ENDP
```