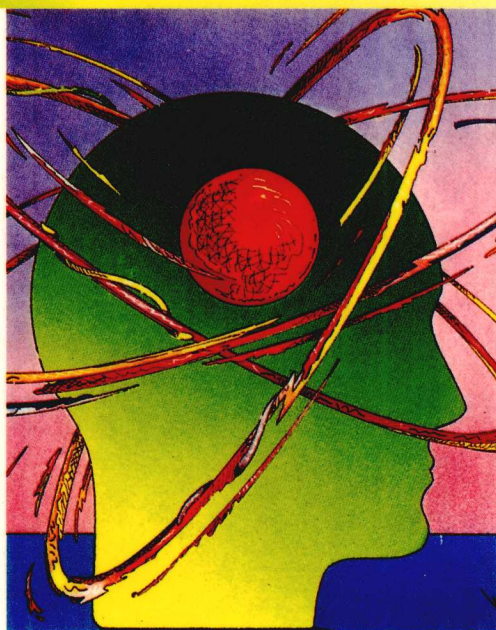




自己設計

SIDEKICK



SoftTip 尖端電腦

瑩園電腦軟體研究開發部

NO. 39360

開數: 16開 頁數: 544頁 譯/著: 編譯

適合讀者: 組合語言初學者或有經驗者

所需硬體: IBM PC/ XT或 PC AT

所需軟體: DOS (含 DEBUG、LINK、EXE2BIN), MASM 或 ASM
組譯器

書附磁片: 無

內容概要:

- 1、以程式來說明各項功能與指令的用法
 - 2、凡是第一次出現的指令或功能都加以追蹤或詳解
 - 3、設計 SIDEKICK 必須用到之程式的技巧 ;如記憶體駐留、鍵盤巨集指令
 - 4、隨時呼叫之計算器模擬程式之設計
-

相關書籍:

深入了解 IBM PC
透視 IBM PC XT/AT
IBM PC組合語言應用程式庫
IBM PC DOS & BIOS 實例與學習
IBM PC/MS DOS 技術手冊
IBM PC/MS DOS 系統函數呼叫辭典
MS MASM 5.X 使用手冊
MS MASM 5.X 速查手冊

目 錄

第一章 組合語言的回顧.....	1
組合語言的心臟.....	2
MOV.....	2
CMP.....	4
THE Jumps.....	6
從基礎到多樣化.....	7
INC.....	8
PUSH與 POP.....	8
ADD 與 SUB.....	10
SHR 與 SHL.....	11
AND 與 OR.....	11
岔斷 (Interrupt).....	12
在我們繼續深入之前.....	14
第二章 COM 檔案.....	17
EXE 與 COM.....	18
在 COM檔案裡沒有堆疊.....	20
假如你已經看過一個.....	21
程式節段前置 (Program Segment Prefix).....	22
資料.....	24

程序本身.....	25
結束程序節段.....	26
使用 COM.....	27
最後叮嚀.....	28

第三章 CLEANUP.COM -- 探討..... 29

搜尋檔案.....	31
程式節段前置控制區.....	32
一個位元接一個位元.....	33
剩下的區域.....	38

第四章 CLEANUP..... 39

關於檔案控制區段.....	39
搜尋欲刪除的檔案.....	40
輸入: YES 或 NO?	44
列印檔名的迴圈.....	49
讀取答案.....	53
刪除.....	54
從 TOP處繼續執行.....	56

第五章 PC與磁碟..... 61

UNDEL.....	61
打開磁碟機.....	63
邏輯上的區段.....	64

叢集 (CLUSTER).....	64
目錄與檔案配置表.....	65
磁碟的內部.....	66
檔案配置表作些什麼事.....	67
讀取一個檔案.....	68

第六章 刪除與 FAT..... 71

寫入一個檔案.....	71
轉到 FAT上.....	72
實際上的 FAT.....	75
刪除.....	77
在我們掌握中的目錄.....	78
找出它在那一磁碟.....	81
讀入目錄.....	83
假如沒有符合的檔案.....	85
CLUSTER_I_0.....	86

第七章 尋找被刪除之檔案..... 89

上個論題.....	89
副程式.....	90
叢集的輸出入.....	92
資料.....	95
尋找被刪除的檔案.....	98
尋找 δ	101
使用 REPE CMPS來查驗檔名.....	103

第八章 完成救援的工作..... 109

重建目錄.....	110
寫入磁碟.....	112
進入 FAT的關鍵.....	113
共有多少個叢集？.....	114
修復 FAT.....	117
讀入 FAT.....	117
此檔案是否被覆寫 (write over) 了.....	120
假如我們遺失了資料.....	121
填入 FAT.....	122
將此 FAT寫到磁碟上.....	124

第九章 進入 FAT中..... 127

PUT_FAT_ENTRY.....	127
GET_NEXT_ZERO.....	131
檢查 /A(或 /a).....	135
列印一段提示.....	138
結論.....	141
假如你使用的是 DOS 1.1.....	142

第十章 時鐘..... 157

劇作.....	157
解救的辦法.....	158

什麼使它執行.....	159
岔斷.....	159
硬體岔斷.....	162
遮斷 (Interception).....	164
一個新節段.....	165
改變 DOS的跳越位址.....	167
配置我們新的位址.....	170
將 THE_PROG 附著於 DOS.....	171
冒充一個岔斷.....	171

第十一章 如何在螢幕上顯示..... 175

BOOSTER.....	175
螢幕.....	181
設定游標.....	183
視訊控制器.....	185
SCREEN節段.....	187
將 CLOCK固定於記憶體.....	190
如 BIOS 般的工作.....	190
將其放在螢幕上.....	192
完成顯示的工作.....	195

第十二章 CLOCK與 CALC..... 199

現在是什麼時間？.....	199
資料節段.....	201
程式節段.....	202

取得時間.....	205
使用 CALC 來計算時間.....	207
ASCII 的乘法調整.....	210
進入 DISPLAY.....	215
分鐘.....	216

第十三章 ONEKEY.....229

簡化.....	230
這不是 DOS.....	231
鍵盤暫存區.....	232
掃描碼及鍵盤.....	233
ONEKEY.....	235
ONEKEY的第二部分.....	236
區段 (Segment).....	237
啓動程式 (Booster).....	239

第十四章 接收字元.....245

讀取字元.....	245
INTERCEPT_KEYBOARD_INT.....	246
觸發鍵.....	252
捲繞 (Wrapping).....	253
是否我們要的？.....	255
匹配打入的鍵.....	258
指到命字串.....	261

第十五章 載入鍵盤暫存區.....265

STUFF.....	265
暫存區滿了嗎？.....	267
填入鍵盤暫存區.....	270
定時器接收.....	273
INTERCEPT_TIMER 的剩餘部分.....	278
運用之前.....	280
把你的字元放入 ONEKEY.....	280

第十六章 NPAD.....295

筆記板.....	95
NPAD看起來像什麼.....	297
簡易的儲存.....	298
NPAD的程序.....	299
模組 (Modules).....	302
切換 (Toggling).....	304
如果不是 ^N.....	305
^N的測試模組.....	306
記憶體內的筆記板.....	307
MOV ATTRIBUTE,7.....	309
MOV PAD_OFFSET,250.....	309
PAD_CURSOR.....	311
螢幕復原.....	311

第十七章 NPAD的螢幕 I/O.....313

完成 IO 的準備工作.....	313
消除筆記板.....	314
IO.....	314
BIOS之資料區域.....	320
IO的顯示迴路.....	322
IO的迴路.....	326
結束.....	327
IO_CHAR.....	328
PUT_CHAR.....	330
離開.....	334

第十八章 按鍵識別..... 335

顯示筆記板.....	335
備存螢幕上的重疊區域.....	336
DISPLAY.....	339
要接受字元嗎？.....	342
模組.....	345
DEL 模組.....	345
擦字模組 (Rubout Module).....	347
返回鍵 (Carriage Returns).....	350
字元 (Characters).....	355
你的 NPAD 版本.....	358

第十九章 DEBUG 基礎入門摘要..... 367

PROTECT#.....	368
利用 DEBUG對 DOS進行探討.....	368

第二十章 PROTECT#; PC 上的檔案保護功能.... 373

中途截斷並移去刪除指令.....	373
驅動器 (The Booster).....	375
移去刪除功能的步驟.....	376
PROTECT_SHARP.....	378
測試.....	380
我們自行刪除檔案.....	381
使用 CLEANUP的部份功能.....	382
DOS 傳遞給我們的 FCB.....	382
DEL_CHECK.....	386
開始搜尋合格的檔案.....	388
假使目錄內找不到合格的檔案.....	390

第二十一章 搜尋與刪除..... 393

DOS 及系統的刪除工作.....	394
TOP.....	398
印出檔案的名稱.....	399
THELOOP 迴路.....	401
二個重要的旗標.....	403
刪除抑或繞道而行.....	407
刪除.....	408
尋找下一個合格的檔案.....	409

恢復正常狀態.....	411
-------------	-----

第二十二章 DEBUG..... 419

DEBUGSCAN.....	420
一個應用實例.....	421
另一個範例.....	425
DEBUG 指令群介紹.....	427
追蹤程式的執行 (Trace).....	428
由 I/O埠輸入及輸出資料 (Input and Output)...	430
搬移內容 (Move).....	431
計算十六進制 (Hex).....	431
顯示暫存器內容 (Register).....	432
執行程式 (Go).....	432
填入 (Fill).....	434
比較指定位置的內容 (Compare).....	435
總結.....	435

第二十三章 偵錯 (Debugging) 的實際運作.... 437

檔案的屬性 (Attribute).....	437
初次嘗試.....	439
使用 DEBUG開始偵錯.....	440

第二十四章 2 的補數 (Two's Complementing). 453

負號.....	454
---------	-----

符號位元.....	454
進位旗標.....	456
新的跳越 (Jumps)指令，非新的數值.....	457
2 的補數.....	459

第二十五章 部份精確度分析..... 463

加法.....	463
減法.....	464
乘法.....	465
除法.....	466
將範例電腦程式化.....	469
程式碼的內容.....	472
比較 DX:AX和 BX:CX之內容.....	474
CMP DX:AX, BX:CX.....	476

第二十六章 磁碟監管(Watchdog)..... 479

DSKWATCH.....	479
磁碟機之旅.....	480
讀取磁碟片的結果正確否？.....	481
磁碟錯誤一覽表 (Catalog).....	481
DSKWATCH程式.....	483
監視磁碟機.....	485
何時會產生磁碟錯誤.....	492
NEC 錯誤碼之解碼.....	496
停止 DSKWATCH.....	497

使用前說明.....	498
------------	-----

第二十七章 圖案設計家字元 (Designer Characters)..... 503

添加另一半之字元表.....	503
----------------	-----

螢幕上的顯示.....	507
-------------	-----

結語.....	508
---------	-----

第一章

組合語言的回顧

假如每件事都能符合我們的須要，那本書充其量也不過如坊間給 PC 使用的書一般沒有什麼特色。我們的注意力是集中在發展 PC 上更刺激的能力，並使其為我們所用。我們的主要目的在於能使用電腦裡所有複雜且特殊的功能，以使我們能從泛泛之輩一躍而成為此方面的頂尖高手。

也許你已經 K 過有關於 PC 內部的書籍而現正想將這些概念的片斷匯集在一起。也許你很喜歡在滿天雲霧裡探險，以找出 PC 的確實功能在那裡。或許你很懷疑 PC 到底能作些什麼事，或者你可讓它作些什麼事？假如你也想過，如何讓兩個程式在同一時間裡執行；或如何放一個記事本在螢幕上，或如何再取回已刪除的檔案... 等等。那筆者可以肯定的說，本書正是你必須常備在案頭的好書之一。

我們將要與專業的軟體發展者使用相同的工具來建立我們的程式，那就是說，我們將用組合語言。想要讀這本書，你必具備有組合語言的基本知識 -- 就算是不非常熟悉，你也應該有粗略的知識。

本章將針對這個主題作個快速的複習，但是我們的篇幅有限，我們實在沒有辦法說得太過於詳細，所以當你在讀完本章而仍然覺得觀念還

是並不怎麼清楚的時候，筆者以為，你應該多花一點時間與精神澈底的將組合語言弄懂才行。換句話說，如果你已經知道 JGE到底是什麼意思的話，那你可說已經懂得夠多而可提早從此章畢業了。

選擇使用組合語言並不是件奇怪的事情，對於大部份我們將要做的東西，比如能同時讓兩個程式執行，或於某個程式執行時放個計算器於螢幕上，或救回已被刪除的檔案等等，這些只是組合語言所能完成的強大功能之一部份而已。而且我們也會在使用新的東西之前作實際上詳細的討論。總之，學習的較好方式是使用範例，而我們也試著在本書提供一些強大而有用的範例供你參考使用。

但是你必須記住一點，本書並不是用來教授組合語言用的。組合語言只是我們拿來作為探測我們所要的東西之工具而已。我們要知道的是在 PC 裡面能使用些什麼有趣的功能之確實知識。也許，能瞭解視訊控制器的原理是件很好的事，但若能實際的應用它不是顯得更有趣味嗎？相同的道理也可以拿來形容磁碟目錄與鍵盤緩衝器。就因為使用 PC 的所有功能是本書的主旨，是故我們不該在未準備週全之前就作深入的討論，因此讓我們花些時間來談談我們所要的工具：

組合語言的心臟

底下是一小群組合成組合語言核心的簡短介紹：

MOV

MOV 是組合語言最基本的指令。假如你已經知道 MOV的確定意義，

那麼可以說你已經上路了。底下是些 MOV 的用法：

```
MOV BX,5CH  
MOV AX,[BX]  
MOV AX,CX
```

第一個命令是將 5CH 搬入暫存器 BX 中。第二個命令使用了 BX 的一個厲害功能，它被當作成一個註標 (index)。這個用法是 8088 或相類似的微處理器與上一代微處理器不同的功能之一，而這點對我們來說相當重要。若你想從記憶體裡位址 1234:5678 開始的地方取得一個 16 位元的內含值時，那麼 DS 必須設成 1234H 而 BX 必須放入 5678H。當 BX 被放於括弧內時，MOV AX, [BX] 會將此字組 (word) 內的值搬入 AX 之中。因為這種定址的方式只使用在程式裡的資料而不是命令，所以 PC 將會假定你正在使用資料節段。也就是說，MOV AX, [BX] 是 MOV AX, DS:[BX] 的縮寫。此種方式稱為間接定址 (indirect addressing)。形成這種定址方式會如此有彈性的原因是，假如持續的增加或減少 BX 的值，你可以向上或向下掃描整個資料節段的資料。如上所述。此種定址的方式非常的有效率，我們將會很快的應用到實例上。

MOV AX, CX 這個指令也許你已經知道是作什麼用的了，它只是很簡單的將 CX 裡的數值移到 AX 裡而已，而遺留在 CX 裡的數值並沒有被更改。

INTEL 的 8088 在記憶體裡儲存字 (word) 時，是以一種令人感到無奈的方式完成的。那就是低位元組先存，然後再存高位元組 (也就是說，較高的位元組放於記憶體的較高位址)。例如，欲存 1234H 時，將