



研讀：精通並專門使用“C”及“dBASE”的軟體公司
為您：挑選及編譯有關“C”及“dBASE”之軟體書籍，
您每天都有更大的成就！ Daily BONANZA To YOU!

高等C 程式庫設計範例

定價 320元

本書簡介

書號：50420

開數：16/32開 頁數： 頁

適合讀者：1. C語言已具基礎，擬有更深了解者。

2. C程式設計師。

所需硬體：IBM PC AT/XT或相容機型

所需軟體：各種 C語言軟體

書附磁片：無

本書對讀者之貢獻：

一般 "C" 語言書談 "排序"、"BTREE" 均偏重於理論，而本書完全以實例來講解 "排序" 及 "BTREE"，並且對於程式設計每一步驟均予詳細解說。

另本書後半節對於鍵盤、螢幕、DOS、BIOS 之管理、終端機之模擬、執行速度之考慮、錯誤之避免等，均有完整範例程式，加以詳細的理論來說明。故本書不但帶你進入 "C" 的更奧妙的領域，同時一個 "C" 程式設計師撰寫程式常到的問題，在本書內範例及說明均能得到解答，甚至該範例幾乎都可直接拿來略做修改即可加入你的程式間使用。

故本書實為 "C" 程式設計必須擁有的好參考書。

內容概要：

本書共分五章及二個附錄，每章都介紹一個主題，並附有完整的程式範例。

第一章根據排序演算法 (sort algorithms)，建立了好幾個工具程式，其中包含了兩個很有名的方法，一個是插入排序法 (insertion sort)，另一是快速排序法 (quicksort)，我們將把它們寫成能適用於各種資料的庫存函數。接著我們舉了一個簡易程式範例，利用這些函數為 ASCII 文字敘述來排序。然後我們會介紹合併 (merging) 的技巧，用來處理超過 RAM 記憶體空間的大型檔案，原先的 MERGE 程式被普遍化，以便能在程式執行時，才指定要做排序的記錄 (record) 格式，和排序的鍵 (key) 之位置及形式。

第二章建立了 BTREE 模組，來處理索引項 (indexed entry) 資料檔案，本模組很長，但提供了一些關於多重索引 (multiple indexed)，重複鍵 (duplicate keys) 和不定長度鍵 (variable-length keys) 等的特性，本章的應用範例，包含了由收信人的名字地址、發信日期，及主題 (subject) 來產生通信記錄索引。

第三章介紹前幾章所用的工具程式。雖然 C 能滿足並適用於我們大多數的要求，但還是有些功能要用到組合語言，其他 MS-DOS 或 IBM PC 的特有模組，也將於本章中介紹，這些模組包括了單鍵輸入 (single-keyinput)、螢幕輸出 (screen output)，和 DOS 及 BIOS 存取之功能。

第四章要介紹終端機模擬程式 (terminal emulator) 的特性，例如，如果程式沒有緊密保存住所有的輸入，那麼從各種輸入來源，不可預期的資料就可能會有所遺漏，本章就要解決這些問題，以循環檢查輸入狀態 (polling input status)，用資料緩衝區來延緩時序之要求，還有可使用中斷驅動輸入處理 (interrupt-driven input handling) 等技巧。

第五章包含了一些比較雜碎的東西--如：利用 C 的獨有特性，來求執行速度最佳化，以及處理較嚴重的錯誤 (critical errors) 等。本章也將以前幾章的程式設計之觀念做個總結。

附錄 A 中我們解說了各種記憶體模式 (memory modules)，以及和 8088 (譯者按：8088 是 IBM PC 的組合語言) 結構間的關係。

附錄 B 中我們介紹了各種工具 SOURCE 程式。

* 註 *

本書第一版 76 年 12 月出版後，短短一年印了六版，而其中近一半之數量係大專院校由老師介紹同學們選購參考。此次承蒙張太乙、黃一橫老師對於內容大幅的修改，並將所有程式一一測試加入附錄 B，使讀者在研讀時可同時實際操作收到事半功倍之效果。

筆記欄

雙圖板您：每天都有更大的成就 (DAILY BONANZA TO YOU)

目 錄

第一章 排序的工具

1.1	內部排序演算法：插入與快速排序.....	1-3
1.1.1	虛擬碼.....	1-4
1.1.2	快速分類：較迅速的演算法.....	1-7
1.2	一般化內部排序函數：memsort.....	1-15
1.3	memsort 執行效率的分析.....	1-25
1.4	加強 memsort之功能.....	1-27
1.5	應用：文字行的排序.....	1-28
1.5.1	虛擬碼.....	1-30
1.5.2	SORTTEXT的原始檔.....	1-31
1.5.3	sorttext.c.....	1-33
	檢查命令行是否標明了兩個檔名，再呼叫 dosort 對檔案進行排序	
1.5.4	fillarea.c.....	1-33
	以一行接一行的方式讀取文字檔，並以傳送引數的方式，將其放入儲存區域	
1.5.5	outfile.c.....	1-38
	使用經 memsort排序過的指標陣列，將文字行依序寫出	
1.5.6	sortio.c.....	1-39
1.6	外部排序演算法.....	1-44
1.6.1	外部排序的虛擬碼.....	1-46
1.6.2	資料結構.....	1-49
1.6.3	例外.....	1-50
1.7	MERGE1原始檔案.....	1-50
1.7.1	merge1.h.....	1-50
	最多可合併的記錄群個數，以及合併階段的兩個函數所用到的 WDATA資料結構	
1.7.2	merge1.c.....	1-51
	merge1.c 是根據 sorttext.c 所建立的程式	

1.7.3	formruns.c.....	1-55
	控制從輸入檔建立起始的記錄群，類似 SORTTEXT 的 dosort 函數	
1.7.4	domerge.c.....	1-58
	控制程式的合併階段	
1.7.5	dopass.c.....	1-59
	執行單一合併處理的巡迴，它的輸入引數是輸入與輸出檔名的字首	
1.7.6	smerge.c.....	1-62
	執行實際的合併工作，其邏輯如同虛擬碼	
1.7.7	resort.c.....	1-65
	將最後一個元素插入陣列中	
1.7.8	util.c.....	1-66
	util.c 檔案包括了一些公用函數：swap_str, mknname 及 defile	
1.7.9	程式結構.....	1-68
1.8	MERGE2：一般化的外部排序程式.....	1-69
1.9	MERGE2：原始檔案.....	1-71
1.9.1	sortcomp.h.....	1-71
	定義了標明鍵值欄位的常數及資料型態	
1.9.2	sortspec.h.....	1-72
	定義了輸入模組所需的檔案與記錄型態常數	
1.9.3	sortcomp.c.....	1-72
	比較兩個記錄，它以 getspec 所定義的鍵值欄位做迴圈	
1.9.4	sortio2.c.....	1-79
	顯示對 SORTTEXT 與 MERGE1 程式所用的輸出入函數所作之修改	
1.9.5	getspec.c.....	1-85
	掃描命令行，找出檔案型態，記錄型態或排序鍵的規格	
1.9.6	merge2.c.....	1-91
1.10	測量 MERGE2 的執行效率.....	1-94
1.11	加強 MERGE2 之功能.....	1-95
1.11.1	較好的使用者介面.....	1-96
1.11.2	處理大檔案.....	1-96
1.11.3	處理更多的資料型態.....	1-98
1.11.4	最佳的執行效率.....	1-99

1.11.5	更具彈性.....	1-100
1.11.6	改進演算法.....	1-101
1.12	結論.....	1-101

第二章 BTREE : 索引檔模組

2.1	觀念的發展.....	2-2
2.2	BTREE 模組的功能規格.....	2-10
2.3	BTREE 虛擬碼.....	2-17
2.4	例外情形與設計選擇.....	2-22
2.5	BTREE 程式列表.....	2-25
2.5.1	btree.h.....	2-29
	btree.h 定義的 BTREE 函數, 負責傳回成功或失敗的代號	
2.5.2	bt_macro.h.....	2-33
	bt_macro.h 的定義只在 BTREE 模組會用到	
2.5.3	bt_space.c.....	2-35
	定義一些 BTREE 的整體變數, 將所有整體變數宣告於一個不包含任何函數的檔案中	
2.5.4	bt_low2.c.....	2-37
	提供一些處理區段中索引項的低階工具	
2.5.5	bt_file.c.....	2-40
	負責索引檔的輸入輸出	
2.5.6	bt_util.c.....	2-43
	支援 BTREE 的公用函數	
2.5.7	bt_free.c.....	2-46
	管理一未用的索引區段的串列 (list)	
2.5.8	blockio.c 與 bt_cache.c.....	2-49
	提供索引區段的輸入與輸出	
2.5.9	openix.c.....	2-58
	集合了有關開啓與關閉索引檔的函數	
2.5.10	bt_first.c.....	2-63
	定位於索引的開始 / 結束之處	
2.5.11	bt_get.c.....	2-67
	get_next 和 get_previous 皆將位於新的目前位置的項目, 拷貝到輸入引數所提供的位址	

2.5.12	bt_block.c.....	2-73
在同一個區段中完成找尋、插入，以及取代的功能		
2.5.13	bt_find.c.....	2-78
找尋函數		
2.5.14	bt_ins.c.....	2-81
插入函數		
2.5.15	bt_del.c.....	2-88
包含了幾個有關刪除項目的函數		
2.5.16	bt_low.c.....	2-97
BTREE 模組的低階函數		
2.5.17	bt_top.c.....	2-100
find_exact, find_ins find_del 函數		
2.6	分析 BTREE.....	2-103
2.7	測試 BTREE.....	2-106
2.8	加強 BTREE之功能.....	2-107
2.8.1	去除限制條件.....	2-107
2.8.2	較好的錯誤處理.....	2-108
2.8.3	公用程式.....	2-108
2.8.4	儲存區更好的用途.....	2-109
2.8.5	組合 BTREE與資料記錄管理.....	2-109
2.8.6	鍵的壓縮.....	2-110
2.8.7	檔案管理應用.....	2-110
2.9	簡單應用：書信索引.....	2-111
2.10	結論.....	2-129

第三章 IBM PC特定用途之低階語言工具庫

3.1	組合語言工具程式.....	3-4
3.1.1	傳輸埠 I/O.....	3-4
3.1.2	存取所有 PC的記憶體.....	3-13
3.1.3	產生軟體岔斷.....	3-23
3.1.4	備註.....	3-35

3.2	測試組合語言函數.....	3-35
3.3	修改工具庫程式以適用於其他的編譯器與組譯器.....	3-53
3.4	使用其他的記憶體模式(memory model).....	3-64
3.5	支援 swint 函數.....	3-67
3.6	DOS 的利用.....	3-71
3.6.1	產生 DOS 呼叫.....	3-78
3.7	鍵盤輸入.....	3-81
3.7.1	備註.....	3-91
3.8	VIDEO 輸出函數.....	3-92
3.9	直接螢幕輸出.....	3-108
3.9.1	設計方法.....	3-112
3.9.2	完成 SCREEN 模組.....	3-114
3.9.3	測量 SCREEN 模組的執行速度.....	3-130
3.10	計時器函數.....	3-131
3.11	建立檔案輸出入庫存函數.....	3-134
3.12	使用並修改工具庫函數.....	3-142
3.13	結論.....	3-143
3.13.1	應用程式的彈性和品質.....	3-143
3.13.2	可攜帶性.....	3-144
3.13.3	發展技巧.....	3-144

第四章 終端機模擬程式

4.1	終端機模擬程式的功用.....	4-3
4.1.1	非同步通訊.....	4-4
4.1.2	非同步通訊轉接器.....	4-4
4.1.3	8250 UART 晶片.....	4-5
4.1.4	RS-232 界面.....	4-5
4.1.5	中斷訊號.....	4-6
4.1.6	飽率.....	4-6
4.2	基本的終端機模擬程式.....	4-7
4.2.1	非同步通訊輸出入支援程式.....	4-9

4.3	TTY1的執行效率	4-19
4.3.1	查詢	4-25
4.3.2	優先權定序方式	4-25
4.3.3	問題分析	4-26
4.4	改進 TTY1 程式的執行效率	4-26
4.5	定義 TTY 終端機模擬程式	4-29
4.6	TTY2 的原始程式檔	4-32
4.6.1	TTY2的標頭檔案	4-35
4.6.2	tty2.c.....	4-36
第二個 TTY終端機模擬程式		
4.6.3	tty2get.c.....	4-39
取得鍵盤輸入		
4.6.4	tty2exec.c.....	4-41
執行鍵盤命令		
4.6.5	tty2disp.c.....	4-43
disp_char 和 disp_msg 函數分別用來顯示單一字元或字串		
4.6.6	tty2rec.c.....	4-45
檔案的讀取 / 記錄函數		
4.6.7	tty2brk.c.....	4-51
送出中斷訊號		
4.6.8	tty2rcv.c.....	4-53
字元接收的處理		
4.6.9	queue.c.....	4-57
以先進入先處理的次序來存取資料的緩衝區		
4.6.10	async2.c.....	4-63
增加傳送一個中斷訊號，以及設定接收資料的岔斷等功能		
4.6.11	install.c.....	4-70
設定 / 去除中斷向量		
4.6.12	intcode.asm.....	4-71
處理岔斷		
4.7	編譯、測試，並評估 TTY2 程式	4-77
4.8	功能的改進	4-79

4.8.1	使用參數	4-79
4.8.2	設定通訊參數	4-79
4.8.3	撥接電話號碼	4-80
4.8.4	撥號目錄	4-80
4.8.5	報導狀態及錯誤	4-81
4.8.6	錯誤檢查協定	4-81
4.8.7	模擬特定的 CRT終端機.....	4-82
4.8.8	布告欄系統	4-83
4.8.9	更快的速度	4-83
4.9	結論.....	4-84

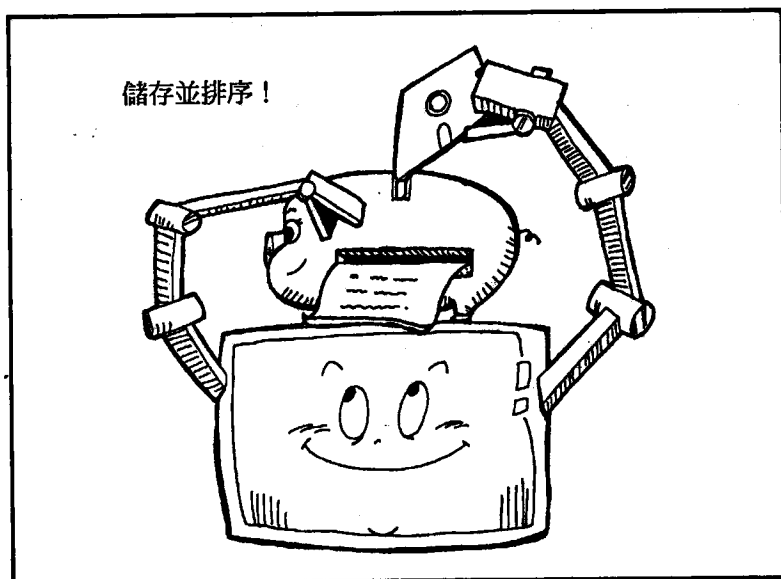
第五章 結語以及最後的一些考慮

5.1	使用 C語言的其他特性:程式最佳化	5-2
5.2	處理 Control-Break 的情況.....	5-8
5.3	處理嚴重錯誤.....	5-15
5.4	討論.....	5-29

附錄 A IBM PC的結構與 C語言的記憶體模式

附錄 B 各種工具 SOURCE 程式

第一章 排序的工具



利用電腦的快速運算，我們可以很方便的進行數字或文字的排序，本書第一章將根據排序演算法而建立一些普遍性的工具程式庫，並設計了兩個完整的程式，本章的目標就是發展一般性的排序模組，以供讀者的程式使用。許多應用都需要將資料依序排好，但有相當多的程式設計者卻無法寫出一個好的排序函數，因此本章提出的模組可讓讀者不必花費心力設計排序函數，而能夠專注於應用發展上。雖然有些排序模組需要改進或加強功能，但總比從新設計簡單多了！

自從發明電腦以來，排序的演算法就一直是計算機科學中倍受矚目的問題。專家們利用數學分析法說明自己的方法，如果我們能廣泛使用這些排序程式，將會更具有實質效益。若能把實際應用和好的數學分析法加以組合，可以得到許多有用的演算法，我們的工作就是善用這用演算法。首先以整數陣列的排序來說明各演算法，然後我們將發展一套可適用於各型態資料的排序工具。

排序問題分為兩種：一種將所有資料一次放入隨機存取記憶體 (RAM) 中，另一種則不行。本章先討論第一種情況 -- 內部排序 (internal sorting)，並發展一般性的工具；接著討論第二種情況 -- 外部排序 (external sorting) -- 的演算法；本章的最後則提出可供不同型態檔案使用的排序 / 合併程式。

1.1 內部排序演算法： 插入與快速排序

將資料依序排列的演算法相當多，一開始，我們先看比較簡單的插入排序，它的技巧是根據玩橋牌時，如何將手上的牌加以排列而來的，亦即每次取出一張牌，將它插入適當的位置。我們排序整數陣列的演算法，就是利用這個方法，下列圖表說明了插入排序運用在含有四個整數陣列的過程，開始時只有第一個元素在排序過的區域內，完成之後，陣列已含有全部的元素：

排序前	—未排序— <u> 4 3 1 6 </u> ↑ 下一個放插入之元素
插入第二個元素之後	—已排序—未排序— <u> 3 4 1 6 </u> ↑ 下一個欲插入 元 素
插入第三個元素之後	—已排序 —未排序 <u> 1 3 4 6 </u> ↑ 下一個欲插 入元素

插入第四個元素之後

—已排序—

1	3	4	6
---	---	---	---

1.1.1 虛擬碼

描述插入排序的虛擬碼非常簡單：

```

insert
  repeat for each element from the second through last
    compare to sorted elements to find where
    to insert it.
    move sorted elements to make room for this one.
    insert it in order.
end

```

insert函數示於圖 1-1，它可以讓任何大小的整數陣列叫用，陣列元素的註標由 0 到 $na-1$ ，第 11 到 20 行的迴圈從元素 1 開始，也就是陣列的第二個元素。注意第 15 到 18 行的內迴圈包含兩種功能：將元素與欲插入之元素作比較，以決定所應插入的位置，並移動某些元素，以便空出位置，插入該元素。第 19 行就是將新元素，放進我們所找到的適當位置。

```
1  /* inserti.c - 整數陣列的插入排序檔案 */
2  #include "stdio.h"
3
4  int insert(a,na)
5      int a[] ;          /* 欲排序的整數陣列 */
6      int na ;           /* 欲排序的整數個數 */
7  {
8      int i , j ;        /* 迴圈的註標 */
9      int temp ;         /* 暫時保存陣列中的一個元素 */
10
11     for( i=1 ; i < na ; i = i + 1 )
12     { /* 插入第 i 個元素到已排序的陣列中 */
13         temp = a[i] ;
14         j = i - 1 ;
15         while( (j >= 0) && (temp < a[j]) )
16         { a[j+1] = a[j] ;
17           j = j - 1 ;
18         }
19         a[j+1] = temp ;
20     }
21 }
```

圖 1-1 inserti.c

<< 瑩 圃 電 腦 R & D >> 5