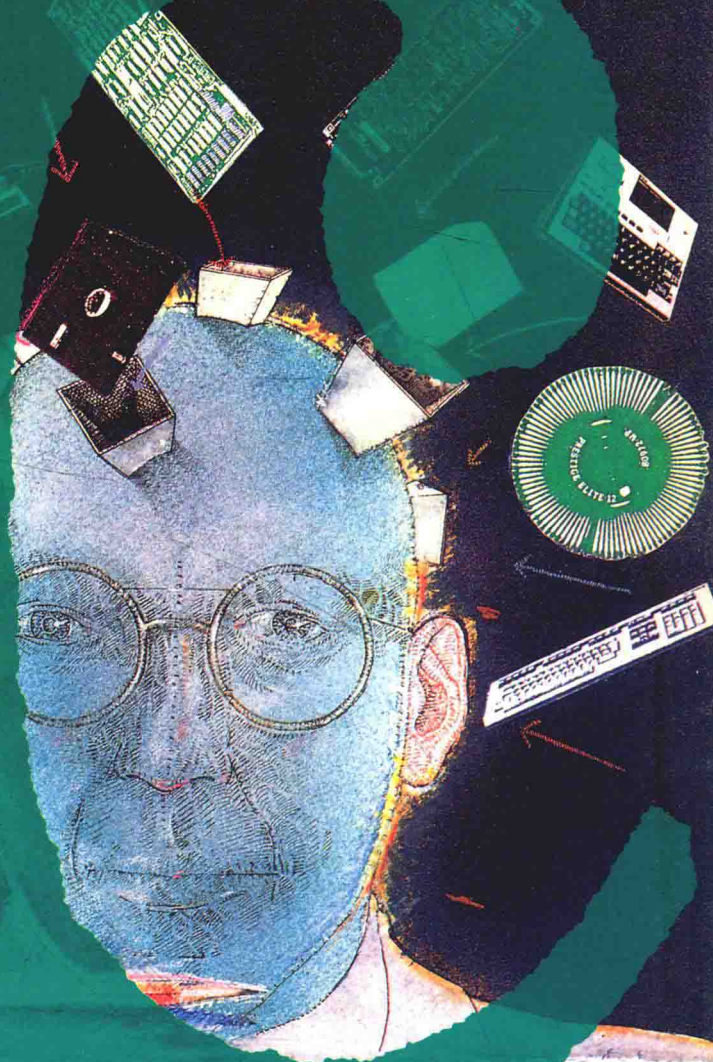


TURBO C

指令函數參考手冊

SoftTip 尖端電腦



榮園電腦軟體研究開發部

Borland C++指令函數字典 / 瑩園電腦軟體研究
開發部編。-- 臺北市：瑩園，民81
面；公分

1. C(電腦程式語言)

版
權
所
有



翻
印
必
究

製 作：瑩園電腦
發行者：周魏筱玉
發行所：台北市和平西路一段84號10樓(瑩園大廈)
信件請寄：台北郵政 6-52 號信箱
郵政劃撥帳號：07923480 戶名：魏筱玉
FAX：(02)365-6584

BIC 82320
82820
82920
83020

中華民國八十一年五月出版
出版登記證：局版臺業字第四二八九號

TRADEMARKS:

Microsoft, MS-DOS, OS/2, Multiplan, Softcard, XENIX, Microsoft Press, Microsoft Windows, QUICK C, QUICK BASIC, NASM are registered trademarks of Microsoft Corporation. FRED, Runtime+, DBASE III PLUS, DBASE CTOOLS, Micromaze, Framework, TimeFrame, FrameLock, and Framework II are trademarks of Ashton-Tate. Apple is a registered trademark of Apple Computer, Inc. MacBASIC is a trademark of Apple Computer, Inc. Macintosh is a trademark of Macintosh Laboratory, Inc. IBM, Personal Edit II, PC/XT, PC AT, PS/2, are registered trademarks of International Business Machines Co. Commodore 64 is a trademark of Commodore. UNIX is a trademark of Bell Laboratories. Tandy is a trademark of Tandy Corporation. Microsoft is a registered trademark of Microsoft, Inc. Lattice is a trademark of Lattice, Inc. Aztec is a trademark of Nanx software. Epson is a registered trademark of Epson. Hercules is a trademark of Hercules Computer Technology. Easy 3D is a trademark of Enabling Technologies, Inc. Wordstar is a registered trademark of MicroPro Int'l Corp. Lotus 1-2-3 is a registered trademark of Lotus Development Corp. R-base is a trademark of Microrim. Turbo, Turbo Basic, Turbo Pascal, Turbo C, TURBO C++, Microcalc, SideKick, Superkey, BORLAND are trademarks of Borland, Inc.

● 本書資料曾參考下列書籍 ●

ADVANCE C FRIDIRNER ++
ADVANCED C
ADVANCED C PROGRAMMING
ADVANCED GRAPHIC IN C
ADVANCED TURBO C
BASIC TO C
C BY DISSECTION
C DATA BASE DEVELOPMENT
C FOR PROGRAMMER'S
C LANGUAGE FOR PROGRAMMER'S
C PRIMER PLUS
C THE COMPLETE REFERENCE
C TOOLBOX
C TRILOGY
CRAFTING C TOOLS
FROM BASIC TO C
GOING FROM BASIC TO C
IBM PC THE COMPLETE C
TURBO C++ PROGRAMMER'S GUIDE
TURBO C++ GETTING STARTED
USING TURBO C++

INTERACTIVE C
LATTICE TOP VIEW TOOLS BASKET
LEARNING TO PROGRAMMING IN C
MASTERING IN C
NSC 5.X QUICK REFERENCE
PASCAL TO C
PROFICIENT C
SUPERCHARGING C WITH ASSEMBLY
SUREFIRE PROGRAMMING IN C
THE C LIBRARY
THE C PROGRAMMER'S HANDBOOK
THE ILLUSTRATED C PROGRAMMING
THE SPIRIT OF C
TURBO C DOS UTILITIES
TURBO DEBUGGER USER'S GUIDE
USING TURBO C
VARIATION IN C
TURBO C++ USER'S GUIDE
TURBO C++ LIBRARY REFERENCE
GRAPHICS PROGRAMMING IN TURBO C

書號：82320/82820/82920/83020

開數：16 開

頁數：1120 頁

適合讀者：C 語言使用者

所需軟體：BORLAND C++、TURBO C、TBUBO C++

本書特色：

1. 本書將所有 BORLAND C++、TURBO C++ 及 TURBO C 之指令全數依字母順序詳解於書中，對於每一指令，依【用途 / 功能摘要】、【Includes】、【參數說明】、【語法範例】、【語法】、【描述】、【傳回值】、【範例】、【執行結果】、【注意事項及建議】、【可攜性】、【相關函數指令】、【功能分類】、【類別】等分項詳細說明。
2. 除了依字母分類外，為方便讀者易於查詢指令，故另分類如下：
 - A. 依 BORLAND C++ 新增之指令函數，對 TURBO C 及 TURBO C++ 指令函數之修改分類。
 - B. 依功能分類。
3. 內容豐富 — 本書係收集國外數十本相關書籍，取其精華融入而成。依正常一般電腦書籍之印刷將高達 2,000 頁，為保有原充實之內容，特別將書內之範例程式及執行結果縮小印刷，以避免書本過厚。
4. 物美價廉 — 瑩圃雖然所用紙張均為進口雪杉紙，價格較別人貴，印刷也採美西德海德堡 PS 版，印刷較一般精美，但因為瑩圃 C 語言書銷售量非常龐大，每本製作費也就大幅降低，“瑩圃”真正做

到薄利多銷之原則。

5. 含可執行範例 — 每一指令函數，最少均含一個可執行之範例，且每一範例不但均含有執行結果，且在關鍵處均加入註解。
 6. 指令 / 函數查詢 — 書內除依字母排序外，在本書前面也有依功能分類表，使讀者在撰寫程式時，易於查詢相關指令。
-

讀者注意：

本書因不同學校、補習班及訓練中心採用，因其各有不同之命名看法，為滿足客戶，故以『C 語言字典』、『BORLAND C++指令函數字典』、『TURBO C 字典』、『TURBO C指令函數參考手冊』等不同的書名出版，但內容完全相同，讀者選購時請勿重複購買。

相關書籍：

C 語言進階 C++

Borland C++ 程式設計入門手冊

Borland C++ 程式設計師手冊

Borland C++ 使用手冊

BORLAND C++ 新增指令函數

- `_atold`
把字串轉換成浮點數
- `_bios_disk`
提供 BIOS 磁碟機之服務
- `_bios_equiplist`
檢查設備
- `_bios_keybrd`
直接在 BIOS 中，提供鍵盤功能
- `_bios_memsizes`
計算記憶體的大小
- `_bios_printer`
在 BIOS 中，列表輸出 I/O
- `_bios_serialcom`
完成串列 I/O
- `_bios_timeofday`
讀取或設定 BIOS 計時器
- `_c_exit`
清除並離開程式
- `_cexit`
清除並離開程式
- `_chain_intr`
鏈結另一個中斷介面
- `_dos_getvect`
取得中斷向量
- `_dos_read`
讀取檔案內容
- `_dos_setblock`
修改先前配置的記憶空間之大小
- `_dos_setvect`
設定中斷向量
- `_dos_write`
寫到檔案中
- `_fmemccpy`
複製 n 位元組的資料
- `_fmemchr`
找尋字元 c (只找尋 n 位元組)
- `_fmemcmp`
比較兩個記憶區塊，它們都是 n 個位元組
- `_fmemcpy`
複製 n 位元組的資料
- `_fmemicmp`
比較兩個記憶區塊，它們都是 n 個位元組，而忽略大小寫
- `_fmemset`
將目的地址填滿 n 位元組的字元 c
- `_fstrcat`
在一字串中搜尋某特定字元第一次出現的位置
- `_fstrcat`
字串連接
- `_fstrespn`
掃描一字串，找出不包含另一字串中之子字串 (或任何其他字元) 的前端部分
- `_fstricmp`
不區分大小寫字母地比較兩字串
- `_fstrlen`
傳回字串長度
- `_fstrlwr`
將大寫字母轉換成小寫字母
- `_fstrncat`
字串連接
- `_fstrncmp`
比較兩字串的前 n 個字元
- `_fstrncpy`
由一字串拷貝特定數目字元至另一字串，若有需要，會將字串截斷或補白 (padding)
- `_chdrive`
設定正確的磁碟機
- `_disable`
放棄中斷
- `_dos_close`
關閉檔案
- `_dos_creatnew`
開啓新檔
- `_dos_findfirst`
搜尋磁碟所在目錄
- `_dos_findnext`
接續 `_dos_findfirst()` 搜尋
- `_dos_getdiskfree`
取得磁碟的於可用空間

<code>_getdrive, _dos_setdrive</code>	將目前的時間轉換為字串
取得並設定正確的磁碟編號	<code>_strtold</code>
<code>_dos_getfileattr, _dos_setfileattr</code>	將字串轉換成長倍精度值
改變檔案存取模式	<code>alloca</code>
<code>_dos_getftime, _dos_setftime</code>	在堆疊中配置一記憶體區塊
取得及設定檔案日期和時間	<code>arc</code>
<code>_dos_gettime, _dos_settime</code>	繪圖用，畫一弧
取得及設定系統時間	<code>closedir</code>
<code>_fstrnicmp</code>	關閉所有目錄
不區分大小寫地比較兩字串的前 n 個字元	<code>inp</code>
<code>_fstrnset</code>	從硬體設備中讀取一位元組
將一字串的前 n 個字元設定為某特定字元	<code>inpw</code>
<code>_fstrpbrk</code>	從一硬體中讀取字組
找出指定字元	<code>lfind</code>
<code>_fstrrchr</code>	執行對數值 key 的線性搜尋
反向找出指定的字元位置	<code>locking</code>
<code>_fstrrev</code>	設定或解除檔案共享 (fine-sharing) 鎖定
倒置給定之字串的字元順序	<code>log10l</code>
<code>_fstrset</code>	計算對數值
設定字串的字元為指定的字元	<code>logl</code>
<code>_fstrspn</code>	計算對數值
傳回指定之字元索引	<code>lsearch</code>
<code>_fstrstr</code>	執行對數值 key 的線性搜尋
傳回一字字串在另一字串中的指標值	<code>mblen</code>
<code>_fstrtok</code>	傳回多位元組字串之長度
依參數格式搜尋語法單元	<code>mbstowcs</code>
<code>_fstrupr</code>	將多位元組字串轉換為 <code>wchar_t</code> 陣列
將小寫字母轉換成大寫字母	<code>mbtowc</code>
<code>_getdcwd</code>	將多位元組字元轉換為 <code>wchar_t</code> 碼
在指定的磁碟中取得正確目錄	<code>movedata</code>
<code>_getdrive</code>	複製 n 位元組的資料
取得正確的磁碟機編號	<code>open</code>
<code>_makepat</code>	開啓所指定的檔案以供讀寫
由各個元素建立路徑	<code>opendir</code>
<code>_matherrl</code>	開啓所指定的目錄以供讀取
可以處理由數學程式館中的函數所產生的錯誤	<code>outp</code>
<code>_searchenv</code>	將指定的位元組寫到硬體輸出埠
由環境變數路徑找尋檔案	<code>outputb</code>
<code>_splitpath</code>	將指定的 byte 位元組寫到輸出埠
將一個完整的路徑分割成其組成單元	<code>outpw</code>
<code>_strdate</code>	將指定的字組寫到硬體輸出埠
將目前日期轉換為字串型態	<code>poly1</code>
<code>_strerror</code>	根據其各個引數 (方程式係數) 值, 產生一個多項式的值
傳回指向錯誤訊息的指標	<code>powl0l</code>
<code>_strtime</code>	計算十的次方值

<code>powl</code>	計算次方值	<code>spawnl, spawnle, spawnlp, spawnvp</code>	執行子行程 (Child process)
<code>readdir</code>	讀取所指定的目錄	<code>sqrtl</code>	計算長倍精確度 (long double) 平方根值
<code>real</code>	將複數的實數部份傳回，或是將 BCD 數字轉回實數型態	<code>stime</code>	設定日期系統之日期時間
<code>rewinddir</code>	將開啟的目錄讀寫指標重新指向開頭	<code>strcpy</code>	複製字串到另一處
<code>rmtmp</code>	刪除暫存檔	<code>strxfrm</code>	複製部份字串
<code>set_new_handler</code>	設定當記憶體配置錯誤時所要執行的函式	<code>tanl</code>	計算長倍精確度正切函數值
<code>setaspecratio</code>	更改形狀比率調整因子 (aspect ratio correction factor)	<code>tempnam</code>	從指定目錄中產生唯一檔名
<code>setblock</code>	修改先前配置的記憶體空間之大小	<code>time</code>	傳回自格林威治標準時間 1970 年 1 月 1 日 00.00:00 起至目前所經過的秒數
<code>setdta</code>	設定磁碟傳輸位址 (disk-transfer address)	<code>umask</code>	設定檔案的讀寫權
<code>setmode</code>	設定開啓檔案的模式	<code>utime</code>	設定檔案的時間和日期
<code>sinhl</code>	傳回雙倍精確度的雙曲線正弦函數值	<code>vsscanf</code>	由管道中格式化讀取資料
<code>sinl</code>	傳回長倍精確度正弦函數值	<code>wcstombs</code>	將 <code>wchar_t</code> 陣列轉換為多字元組之字串
<code>spawnlpe, spawnv, spawnve, spawnvpe,</code>		<code>wctomb</code>	將 <code>wchar_t</code> 碼轉換為多字元組字元

BORLAND C++ 對 TURBO C++ 指令函數之修改

<code>_OvrlnitExt</code>	設置延伸記憶體 (Extended Memory) 給覆疊處理器 (Overlay manager) 使用		產生複數
<code>_OvrlntitEms</code>	設置延伸記憶體 (Expanded memory) 給覆疊處理器 (Overlay manager) 使用	<code>conj</code>	傳回複數的共軛複數
<code>_setcursortype</code>	選擇本文狀態下 (Text Mode) 的游標形狀	<code>farheapcheck</code>	檢查及驗證遠程堆積 (heap)
<code>arg</code>	計算一個複數在複數平面上的角度	<code>farheapcheckfree</code>	檢查堆積的未使用區塊，是否皆為一個常數
<code>atan2, atan2l</code>	傳回 y/x 的反正切值	<code>farheapchecknode</code>	檢查及驗證在遠程堆積中的一個節點 (node)
<code>bcd</code>	將一個數字轉換成 BCD 數	<code>farheapfillfree</code>	以一個常數填入遠程堆積的未使用區塊
<code>complex</code>		<code>farheapwalk</code>	一個接著一個地走完所有的節點，以

取得遠程堆積的相關資訊	isxdigit
fputchar	判別字元是否為十六進位的數字
輸出一個字元到標準輸出設備 (stdout)	line
getmaxy	在二指定點間繪一直線
取得螢幕座標 y 的最大值	localeconv
getpid	傳回現在的列印格式 (formats)
取得程式的識別碼 ID	mktime
heapcheck	傳回自 1970 年 1 月 1 日 00:00:00 至某年某月某日某時所經的秒數，並計算出這天是星期幾，以及是一年中的第幾天
檢查及驗證堆積 (heap)	norm
heapcheckfree	傳回複數的絕對值的平方
檢查堆積的未使用區塊，是否皆為一個常數	polar
heapchecknode	轉換一極座標成為複數座標
檢查及驗證在堆積中的一個節點 (node)	read
heapfillfree	讀取檔案內的資料
以一個常數填入堆積的未使用區塊	setlocale
heapwalk	設定現在的列印格式
一個接著一個地走完所有的節點，以取得堆積的相關資訊	strchr
imag	在一字串中搜尋某特定字元第一次出現的位置
取得複數的虛部	strcmp
installuserdriver	比較兩字串的大小
裝置一個外加的設備驅動程式 (device driver)	strcoll
installuserfont	根據某特定格式，比較兩字串的大小
載入一個非內定的字型檔 (CHR) 到 BGI 系統內	strcpy
isalpha	將一字串的內容拷貝至另一字串
判別一字元是否為英文字母 ('A'~'Z'或'a'~'z')	strcspn
isgraph	字串處理
判別一字元是否為可印出的字元 (<0x21~0x7E)	strftime
islower	將時間值轉換為格式化的字串以供輸出之用
判別一字元是否為小寫字母 ('a'~'z')	stricmp
isprint	不區分大小寫字母地比較兩字串
判別一 ASCII 字元是否為可印出的字元 (<0x21~0)	strncmp
ispunct	比較兩字串的前 n 個字元
判別一字元是否為標點符號 (0x00~0x20和 0x7F)	strncpy
isspace	字串複製
測定字元型態是否為調位字元 (0x09-0x0d或0x20)	strnicmp
isupper	不區分大小寫地比較兩字串的前 n 個字元
判別字元是否為大寫字母 ('A'~'Z')	strnset
	將一字串的前 n 個字元設定為某特定字元
	vprintf
	將格式化的資料輸出至標準輸出檔 (stdout)
	vsprintf
	將格式化的資料輸出至字串緩衝區

BORLAND C++ 對 TURBO C 指令函數之修改

<code>--emit--</code>	讀取指定的磁碟區段
<code>_chmod</code>	<code>abswrite</code> 將資料寫到指定的磁碟區段
<code>_clear87</code>	<code>access</code> 檢查檔案的可存取性
<code>_close</code>	<code>acos</code> 傳回反餘弦值
<code>_control87</code>	<code>allocmem</code> 傳回記憶體區段位址
<code>_creat</code>	<code>asctime</code> 產生 ASCII 碼的時間字串
<code>_dos_open</code>	<code>assert</code> <code>assert</code> 測試一個條件並有可能放棄
<code>_fpreset</code>	<code>atan</code> 計算反正切值
<code>_graphgetmem</code>	<code>atan2</code> 用以計算 y/x 的反正切
<code>_lrotl</code>	<code>atexit</code> 在一程式正常結束，回到作業系統 (OperatingSystem) 之前呼叫 <code>func</code>
<code>_lrotr</code>	<code>atof</code> 轉換 ASCII 碼的字串為浮點型態的值
<code>_matherr</code>	<code>atoi</code> 轉換 ASCII 碼的字串為整數值
<code>_open</code>	<code>atol</code> 轉換 ASCII 碼的字串為長整數值
<code>_rotl</code>	<code>bar</code> 畫柱狀圖
<code>_rotr</code>	<code>bdos</code> 執行基本上 DOS 所提供的系統函數
<code>_status87</code>	<code>bdosptr</code> 啟動 MS-DOS 系統呼叫
<code>_tolower</code>	<code>cbioscom</code> 根據引數 <code>port</code> 所指定的輸出執行各種 RS232 運作
<code>_toupper</code>	<code>biosdisk</code> 直接透過 BIOS 進行磁碟運作
<code>_write</code>	<code>biosequip</code> 檢查與系統連接的設備
<code>abort</code>	<code>bioskey</code> 控制鍵盤運作
<code>abs</code>	<code>biosmemory</code> 傳回記憶體的大小
<code>absread</code>	

biosprint	執行各種印表機的功能	country	根據 countrycode傳回相對應的格式化資料，其資料包括分割日期、時間、現鈔等符號
biostime	讀取或設定 BIOS 的計時器	cprintf	把格式化的輸出送到螢幕
brk	改變程式所配置的資料段空間	cputs	把字串直接輸出到螢幕上
bsearch	二分搜尋一個排序過的陣列	creat	產生一個第一階層的檔案
cabs	計算複數的絕對值	creatnew	產生一個第一階層的檔案
calloc	配置第三階層的記憶體	creattemp	產生一個第一階層的檔案
ceil, ceill	取得一個浮點數的極限值	cscanf	從控制台將資料讀到引數所指定的位置上
cgets	從主控台上直接輸入字串	ctime	把時間值轉換成字串形式
chdir	變換目前所在的目錄	ctrlbrk	設定 control-break處理函數
chmod	改變檔案的保護模式	delay	暫停執行片刻 (毫秒)
chsize	改變檔案大小	delline	刪除在本文視窗中的字列
circle	畫圓	detectgraph	藉檢查硬體以決定所要使用的圖型驅動器及模態
cleardevice	清除圖形螢幕	difftime	計算 time2 - time1 的值
clearerr	清除第二階層輸出入或檔案的錯誤旗標	disable	控制硬體中斷
clearviewport	清除目前的觀察窗	div	兩整數相除，傳回商及餘數
clock	決定處理機時間	dosexterr	產生擴充錯誤訊息存到 eblkp所指的位址
close	關閉第一階層的一個檔案	dostounix	由 getdate和 gettime所傳回的 date 和 time 值改成 UNIX 型式的表示法
closegraph	關掉圖型系統	drawpoly	畫一多邊形的輪廓
clreol	在本文窗中清除至最末一列	dup	建立開啓之檔案的第二個代碼
clrscr	清除本文視窗	dup2	迫使 newhandle代碼與 oldhandle代碼所代表的檔案相關
coreleft	計算可用記憶體的空間		
cos	傳回餘弦函數值		
cosh, coshl	傳回雙曲線餘弦函數值		

ecvt	將浮點轉換成字串	fflush	清除管道的緩衝區
ellipse	畫一橢圓弧	fgetc	由管道中讀字元
enable	允許任何設備產生中斷	fgetchar	由 stdin 中讀字元
eof	偵測檔案終點記號	fgetpos	取得目前檔案指標
FP_OFF	取得遠指標的偏移位址	fgets	從管道中讀取一個字串
FP_SEG	取得遠指標的段位址	filelength	傳回檔案的長度
exec	載入並執行其他函數	fileno	傳回與目前指定之管道相關的檔案代碼
execl, execlx, execlp, execlpe,		fillemnipse	用來畫並填滿一橢圓形
execv, execve, execvp, execvpe	把指向字元字串的指標傳遞過去	fillpoly	畫並填滿一多邊形
exit - _exit	結束程式的執行，並關閉所有的檔案	findfirst	找出第一個符合條件的檔案
exp	計算指數的函數	findnext	尋找下一個符合條件的檔案
fabs, fabsf	計算浮點引數的絕對值	floodfill	塗滿一有界限的區域
farcalloc	由累堆所取得之記憶空間	floor, floorf	傳回浮點值的整數部份
farcoreleft	計算累堆未用的記憶空間	flushall	將所有緩衝區的內容寫入相關檔案
farfree	釋出記憶空間 (不在同一資料段的記憶空間)	fmod, fmodf	求浮點數相除後的餘數
farmalloc	取得記憶空間	fnmerge	取得完整的路徑名稱
farrealloc	將舊記憶空間資料複製到新的位址	fnsplit	分割路徑名稱
fclose	關閉某一開啓之管道	fopen	開啓一個緩衝式檔案
fcloseall	關閉所有開啓的管道	fprintf	輸出格式化的字元或數值到某一管道中
fcvt	將浮點數轉換成字串	fputc	輸出一個字元到某一管道中
fdopen	建立某輸出入管道與另一檔案代碼之關係	fputchar	輸出一個字元到標準輸出 (stdout)
feof	檢查檔案是否終止	fputs	輸出字串到某一管道
ferror	測試指定管道的讀取或寫入錯誤	fread	從管道中讀取資料

free	釋放記憶體	getcwd	取得工作目錄的完整路徑名稱
freemem	釋回記憶體	getdate	取得系統的日期
freopen	再次開啓一個緩衝式處理的檔案	getdefaultpalette	傳回調色板所定義的結構
frexp, frexpl	切割浮點數值為假數與指數兩個部份	getdfree	取得磁碟剩餘空間
fscanf	自檔案中讀取格式化的資料	getdisk	傳回使用的磁碟號碼
fseek	在檔案中尋找一個指定的新的位置來讀取或寫入	getdta	取得轉移位址
fsetpos	設定檔案中讀取或寫入的位置	getenv	搜尋和 envvar 相符的項目
fstat	取得開啓檔案的資料	getfat	取得檔案配置表上的資訊
ftell	傳回目前的檔案指標所在的位置	getfatd	取得預設磁碟機檔案配置表上的資訊
fwrite	將資料寫到輸出的管道	getfillpattern	將使用者定義的填圖型樣拷貝至記憶體內
gcvt	轉換浮點值為字串	getfillsettings	抓取有關目前填圖型樣及顏色的有關資訊
geninterrupt	產生中斷	getftime	取得檔案上一次的使用日期與時間
getarccoords	抓最後一個呼叫 arc 之座標	getgraphmode	傳回目前繪圖模態
getaspectratio	傳回目前圖形模態的縱橫比	getimage	將某一指定區域的位元影像存入記憶體內
getbkcolor	傳回目前背景色	getlinesettings	抓取線的目前形式、形樣及粗細
getc	從目前的管道中讀取單一字元	getmaxcolor	傳回最大顏色值
getcbrk	檢查 control-break 的旗標	getmaxmode	傳回當時驅動程式的最大模態編號
getch	從控制台上讀取單一字元	getmaxx	傳回最大 x 螢幕座標
getchar	從 stdin (通常為鍵盤) 讀取單一字元	getmoderange	抓取一給定圖形驅動器之模態範圍
getche	自鍵盤讀取字元, 回應至螢幕	getpalette	傳回有關目前調色板的資訊
getcolor	傳回目前的繪圖顏色	getpass	顯示引數 prompt 內存之字串並讀取輸入之 PASS4
getcurdir	取得某特定磁碟的目錄名稱	getpixel	抓指定圖元的顏色

getpsp	建立程式分段字首	imagesize	傳回儲存一“位元影像”(bit map)所須的位元
gets	標準輸入管道讀取一列資料	initgraph	起始設定繪圖系統
gettext	從本文模態螢幕上的本文拷貝至記憶體	inport	從指定的輸出入埠讀入兩個位元組
gettextinfo	抓取本文模態螢幕資訊	inportb	從指定的輸出入埠讀入一個位元組
gettextsettings	傳回有關目前本文設置的資訊	inline	在本文視窗中插入空白行
gettime	取得系統當時的時間	int86	8086軟體中斷界面
getvect	取得中斷向量	int86x	8086軟體中斷界面
getverify	取得驗證旗標之狀態	intdos	啟動暫存器值所指定的 DOS中斷
getviewsettings	傳回有關目前觀察窗的資訊	intdosx	啟動暫存器值所指定的 DOS系統呼叫
getw	讀取整數型別的二進位值	intr	產生軟體中斷
getx	傳回目前位置的 x座標	ioctl	I/O 裝置控制
gety	傳回目前位置的 y座標	isalnum	測試某一指定的整數值是否為('A'-'Z', 'a'-'z'或'0'-'9')等文字或數字之一
gmtime	以長整數值之型別儲存時間成為格林威治標準	isalpha	測試某一指定的整數值是否為('A'-'Z'或'a'-'z')等字母之一
gotoxy	放置本文視窗上的游標	isascii	測試某一指定的整數值
graphdefaults	將所有“圖形設置”重新設定為它們的預設置	isatty	測定裝置型態
grapherrormsg	傳回一個錯誤訊息字串	iscntrl	測定字元型態
graphresult	就最後一次不成功的圖形運算傳回一錯誤碼	isdigit	測定字元型態是否為('0'-'9')等數字
harderr	建立硬體錯誤處理函數	isgraph	測定字元型態是否為不含空格在內的可列印字元(0x21- 0x7e)
hardresume	讓處理函數回到 DOS	islower	測定字元型態是否為('a'-'z')等小寫字母
hardretn	讓處理函數回到程式	isprint	測定字元型態是否為(0x20-0x7e)等可列印字元
highvideo	選擇高強度本文字元		

ispunct	判定字元型態是否為標點符號字元 (isalnum(c) 或 isctrl(c) 均為真)	matherr	可以處理由數學程式館中的函數所產生的錯誤
isspace	判定字元型態是否為調位字元 (0x09-0x0d或0x20)	max	將兩個數值中的較大者傳回
isupper	判定字元型態是否為大寫字母('A'-'Z')	memcpy	從 source位置將的資料拷貝到 destin 位置
isxdigit	判定字元型態是否為十六進位數字 ('A'-'F', 'a'-'f'或'0'-'9')	memchr	搜尋指定之字元
itoa	將某一數字，轉換成字串	memcmp	將指定的位元組依ASCII碼的順序作比較
kbhit	檢查控制台是否有按鍵敲擊的動作	memcpy	將 source 的指定的位元組拷貝到 destin
keep	允許程式暫時跳回 DOS系統	memcmp	將指定的位元組作比較不考慮大小寫
labs	產生長整數引數 n的絕對值	memmove	將指定位元組複製到所指定的位址
ldexp	計算 $x \cdot 2^{(exp)}$ 的值	memset	將指定的位元組設定為引數所指定的字元
ldexpl	計算長浮點數 $x \cdot 2^{(exp)}$ 的值	min	將兩個數值中的較小者傳回
ldiv	將兩長整數相除，傳回商及餘數	MK_FP	根據段位偏移產生一個 4個位元組的遠 (far) 指
linerel	畫一條〈離目前位置 (CP) 一段距離〉的直線	mkdir	建立一個新的目錄
lineto	畫一條直線由 CP 至 (x,y)	mktemp	建立一個唯一的檔案名稱
localtime	根據變數 timezone 的值把 time 作適當的修正	modf	將浮點值分解成小數和整數兩部分
lock	鎖定共用檔案	movedata	來移動遠處的資料
log	計算對數值	movemem	複製位元組到指定的位址上
longjmp	恢復存在的環境條件	moverel	移動目前位置 (CP) 一個相對距離
lowvideo	選取低強度字元	movetext	拷貝螢幕上一矩形本文至另一塊地方
lseek	把指標重新定位	moveto	將現在的位置 (CP) 移到 (x, y)
ltoa	將長整數轉換成字串	normvideo	選取正常強度字元
		nosound	關掉 PC 喇叭
		open	開啓 pathname 所指定的檔案

outport
 將指定的 word 寫到輸出埠
 outportb
 將指定的 byte 位元寫到輸出埠
 outtext
 在觀察窗顯示一字串
 outtextxy
 將一字串送至指定位置
 parsfnm
 剖析一個字串成為檔案名稱
 peek
 用來檢查位址內容
 peekb
 用來檢查位址內容
 perror
 將一道錯誤訊息印至stderr中
 pieslice
 畫及塗滿圓餅圖切片
 poke
 將 value 的值存入位址
 pokeb
 將 value 的值存入位址
 pow
 計算次方值
 powl0
 計算次方值
 printf
 格式化的列印
 putc
 輸出字元
 putch
 輸出字元至螢幕
 putchar
 輸出字元至標準設備
 putenv
 增加或取代環境變數字串
 putimage
 將位元影像放到螢幕上
 putpixel
 在指定點畫一點
 puts
 輸出字串
 puttext
 自記憶體拷貝本文至螢幕
 putw
 輸出二進位整數
 qsort

使用快速排序演算法做排序的工作
 raise
 將一個軟體信號送給正在執行的程式
 rand
 產生一個虛擬隨機亂數
 randbrd
 利用指標隨機讀取記錄
 randbwr
 利用指示隨機寫入記錄
 random
 亂數產生器
 randomize
 起始設定亂數產生器
 read
 讀一個檔案
 realloc
 重新配置主記憶
 rectangle
 繪矩形
 registerbgidriver
 登錄已鍵入之繪圖驅動器程式碼
 registerbgifont
 登錄已鍵入的鉛筆字形碼
 remove
 將檔案刪除
 rename
 改變檔案或目錄名稱
 restorecrtmode
 恢復螢幕狀態至先前 initgraph設定
 rewind
 將檔案指標重新設定在檔案的開端
 rmdir
 刪除目錄
 sbrk
 更改資料段空間配置
 scanf
 由標準輸入 (stdin) 管道讀入資料
 searchpath
 嘗試搜尋檔案
 sector
 畫出一個扇狀的派餅圖 (elliptical pie silce), 並將其填滿
 segread
 將段暫存器的值填入結構中
 setactivepage
 設定一工作頁面 (active page) 作為繪圖狀態下輸出之用

setallpalette

依指定的方式改變調色板所有顏色

setbkcolor

用調色板設定目前背景顏色

setblock

修改記憶段空間之大小

setbuf

設定緩衝區

setcbreak

設定 control-break 的旗標

setcolor

用調色板設定目前繪圖顏色

setdate

設定系統日期

setdisk

設定目前工作磁碟機

setfillpattern

選取使用者自定的填圖型樣

setfillstyle

設定填圖型樣及顏色

setftime

設定檔案的日期時間

setgraphmode

設定系統為繪圖模式並清除螢幕

setjmp

將目前的堆疊環境存入變數環境

setlinestyle

設定目前線的寬度及型式

setmode

設定模式

setpalette

改變調色板一個顏色

setrgbpalette

允許使用者能設定 IBM 8514 的顏色

settextjustify

設定文本版面

settextstyle

設定目前本文字元

settime

設定時間

setusercharsize

就鉛鑄字形而言，它是使用者自定的字元
擴大因子

setvbuf

設定緩衝區

setvect

改變中斷向量值

setverify

設定驗證旗標的狀態

setviewport

設定目前觀察窗為繪圖輸出

setvisualpage

設定由螢幕顯示的工作頁面

setwritemode

在繪圖模式中，為線性繪圖設定寫入模式

signal

設定信號處理的動作

sin

傳回正弦函數值

sinh

傳回雙曲線正弦函數值

sleep

暫停執行動作數秒鐘

sopen

開啓一共用檔案

sound

打開 PC 的喇叭，且以特定的頻率發聲

sprintf

格式化字元和數值輸出到一個字串

srand

設定產生一系統虛擬隨機數的起點

sscanf

讀取資料然後移至給定的位置

stat

開啓檔案目錄後貯存訊息

stime

設定日期系統之日期時間

strcat

字串連接

strcmp

字串處理

strcmpi

不考慮文字大小寫形式來比較兩字串

strdup

產生一塊記憶空間，並將字串放入

strerror

傳回指向錯誤訊息的指標

strlen

傳回字串長度

strlwr

將大寫字母轉換成小寫字母

strncat

字串連接

strncmpi	不考慮文字大小寫型式，將一字串部份和另一字串部份作比較	00.00:00 起至目前所經過的秒數	
strncpy	字串複製	tmpfile	在目前工作目錄中開啓一個臨時的緩衝式檔案
strpbrk	找出指定字元	tmpnam	產生一個臨時的檔案名稱
strrchr	反向找出指定的字元位置	toascii	將單一字元轉換成所指定的格式
strrev	倒置給定之字串的字元順序	tolower	將單一字元轉換成所指定的格式
strset	設定字串的字元為指定的字元	toupper	將單一字元轉換成所指定的格式
strspn	傳回指定之字元索引	tzset	設定整體變數 daylight, timezone 和 tzname 之值
strstr	傳回子字串在另一字串中的指標值	ultoa	將未具正負號的長整數數字轉換成以空字元結尾的字元串
strtod	將字串轉換成雙精度值	ungetc	還原字元至輸入管道
strtok	搜尋剖析單元	ungetch	將字元還原到控制台上
strtol	將字串轉換成長整數	unixtodos	把 unix 型式的時間表法轉換成 DOS型式
strtol	把一字串轉換成型態為無負號的長整數	unlink	指定檔案的刪除
strupr	將指定字串中的小寫字母全部轉換成大寫字母	unlock	解鎖被 lock 鎖住的檔案
swab	拷貝並且調換位元組	va_arg, va_end, va_start	存取函數引數
system	將指定的字串傳給命令解釋程式	vfprintf	將輸出資料格式化
tell	取得與 handle 相關連之檔案指標	vfprintf	格式化輸入資料
textattr	設定本文 (text) 的屬性	vprintf	將輸出資料格式化
textbackground	選擇新的本文背景顏色	vscanf	格式化標準輸入資料
textcolor	選擇本文狀態下顯示字元的顏色	wherex	傳回視窗內游標的水平位置
textheight	傳回字串的高度，以圖元 (pixels) 為單位	wherey	傳回視窗內游標的水平位置
textmode	將螢幕設定在本文狀態	window	定義目前本文工作狀態下視窗的大小
time	傳回自格林威治標準時間 1970 年 1月1	write	將資料寫到檔案中