

● 微型计算机软件丛书



● 刘甲耀 严桂兰 著

Turbo C
语言程序设计

电子工业出版社

Turbo C 语言程序设计

刘甲耀 严桂兰 著

电子工业出版社

(京)新登字 055 号

内容提要

本书借助软件设计表现法 PAD, 阐述 Turbo C 语言程序设计方法。内容包括 Turbo C 的使用; Turbo C 基础知识; Turbo C 基本程序结构; Turbo C 子程序结构; 构造型数据类型; 文件管理; 存储管理; 进程管理; 软件接口, 以及 Turbo C 高级程序设计。书末附有 C 语言的 PAD 标准图式, 以及 Turbo C 库函数。

本书可供各行各业从事计算机工作的人员使用, 也可作为大、专院校计算机及相关专业的教材及参考书。

Turbo C 语言程序设计

刘甲耀 严桂兰 著

责任编辑 王惠民 徐云鹏

*

电子工业出版社出版(北京海淀区万寿路)

电子工业出版社发行 各地新华书店经售

电子工业出版社计算机排版室排版

北京顺义李史山胶印厂印刷

*

开本: 787×1092 毫米 1/16 印张: 28.125 字数: 720 千字

1991 年 10 月第一版 1991 年 10 月第一次印刷

印数: 1-10100 册 定价: 13.50 元

ISBN7-5053-0877-7/TP·137

前 言

C语言是一种结构式、模块式、编译式通用程序设计语言,是当今世界上最有影响的程序设计语言之一,其数据类型丰富,运算符众多而灵活,控制流和数据结构新颖,既具有高级程序设计语言的功能,也具有低级汇编语言的功能,适宜于编制各种软件,包括系统软件与应用软件。而 Turbo C 则是美国 Borland 国际有限公司于 1987 年 5 月首次推出的一种具有现代编程风格的 C 语言,其使用容易,编译速度快(每分钟可编译 7000 行),效率高,能在 IBM PC 及其兼容机上运行,而且完全支持 ANSI 的 C 语言标准,完全支持 Kernighan 和 Ritchie 的 C 语言定义,包括开发 PC 上所使用的混合语言和混合模式程序设计的某些任意的扩充。如果把 Turbo C 应用与 Turbo prolog 进行连接,还能增加 Turbo C 应用人工智能的能力,并能大大减少软件开发时间,增加代码的透明度和程序的适应性。

本书特点是:

(1)编程技术新颖——应用当前先进的软件设计表现法 PAD 编制 Turbo C 程序,程序逻辑透明、可视、易学、易读、易记、易理解;

(2)熔 PAD、程序、技巧三者为一体,并以顺序、选择、循环三种基本程序结构为中心组织教材内容,逻辑思路清晰,别具一格;

(3)取材广泛,由浅入深,PAD/C 程序并举,适合于多层次读者。书中所有 Turbo C 程序均在 IBM PC 机上通过。

在这里,应该指出,Turbo C 语言程序设计是一门实践性很强的课程,应通过 PAD 转换成 C 语言程序和上机实践,这样,才能真正学会编制 C 语言程序,以达到应用 PAD 高效率地正确地编制 C 程序之目的。

在本书编写过程中,谷惠英和刘浩两同志编写和调试了部分程序,并对全书进行校对,在此谨向她(他)们表示感谢。

由于作者水平所限,书中错误之处在所难免,请读者批评指正。

作者 1989.10

目 录

第一章 Turbo C 的使用	(1)
第一节 Turbo C 的适用范围与系统的安装	(1)
第二节 Turbo C 的集成开发环境	(5)
第三节 Turbo C 命令行的使用与任选项的选择	(8)
第四节 Turbo C 的热键与编辑命令	(14)
第二章 Turbo C 基础知识	(17)
第一节 Turbo C 的基本符号与源程序结构	(17)
第二节 基本数据类型与存储类型	(20)
第三节 运算符与表达式	(38)
第三章 Turbo C 基本程序结构	(56)
第一节 顺序结构	(56)
第二节 选择结构	(79)
第三节 循环结构	(97)
第四章 Turbo C 的子程序结构	(130)
第一节 Turbo C 库子程序	(130)
第二节 函数的定义与调用	(141)
第三节 函数的递归与分程序结构	(147)
第四节 C 预处理程序	(154)
第五章 构造型数据类型	(165)
第一节 指针类型	(165)
第二节 数组类型	(179)
第三节 结构类型	(205)
第四节 联合类型	(223)
第六章 文件管理	(229)
第一节 流式文件	(229)
第二节 标准文件	(244)
第三节 设备文件	(257)
第四节 目录文件	(258)
第七章 存储管理	(269)
第一节 存储分配函数及其应用	(269)
第二节 存储操作函数及其应用	(278)
第八章 进程管理	(285)
第一节 DOS 命令的发布与程序的终止	(285)
第二节 其它程序的装入与运行	(286)
第三节 子进程的建立与运行	(287)
第九章 软件接口	(290)

第一节 与 DOS、8086、BIOS 的接口	(290)
第二节 Turbo C 与 Turbo prolog 的接口	(316)
第十章 Turbo C 高级程序设计	(327)
第一节 存储模式与近、远、巨型指针	(327)
第二节 混合模式程序设计	(333)
第三节 混合语言程序设计	(338)
第四节 低级程序设计	(347)
第五节 C 语言程序设计风格	(360)
附录 A C 语言的 PAD 标准图式	(363)
附录 B Turbo C 库函数	(364)
参考文献	(444)

第一章 Turbo C 的使用

Turbo C 是在 IBM PC 及其兼容机上实现的一种通用程序设计语言。它拥有完好的集成开发环境,交互式的编辑程序。编译、连接、运行能放在一起连续进行,而且起动快,编译速度快,易学易用。在介绍 Turbo C 程序设计之前,先介绍 Turbo C 的使用。

第一节 Turbo C 的适用范围与系统的安装

一. Turbo C 的适用范围

C 语言是一种结构式、模块式、编译式通用程序设计语言,具有好的可移植性,适宜于编制各种软件(包括系统软件和应用软件),特别是系统软件,也就是说,各式各样的程序设计任务几乎都可以用 C 语言来完成。而 Turbo C 则是美国 Borland 国际有限公司于 1987 年首次推出的一种 C 语言,它具有 300 多个库子程序(与 optimizing C86(2.3 版本)相比,多了 170 多个库函数),其中包括分类子程序、目录子程序、进程子程序、转换子程序、诊断子程序、输入/输出子程序、接口子程序(DOS, 8086, BIOS)、处理子程序(字符串,存储器)、数学子程序、存储分配子程序、杂子程序、标准子程序、时间和日期子程序等 12 类。以及 stdio.h、alloc.h 等 26 个包含文件(详见附录 B)。在 C 程序中,可用这些函数和包含文件来完成各式各样的程序设计任务,包括低级和高级的输入/输出,字符串和文件的处理,存储分配,进程控制,数据转换,数学计算等等。与传统的语言相比,它库函数丰富,起动快,使用容易,编译速度快(每分钟可编译 7000 行),效率高。欲建立和运行 C 语言程序,不需要使用单独的编辑程序、编译程序、连接程序以及 MAKE 软件,因为所有这些都已装进了 Turbo C 之内,构成了 Turbo C 集成的开发环境。

Turbo C 可以在 IBM PC,包括 XT、AT 以及所有 IBM 兼容机上实现,只要 DOS 2.0 或 DOS 2.0+,以及 384K 的 RAM 即可运行。整个 Turbo C 程序包有四个配置盘,它含有建立、编译、连接和运行 Turbo C 所必须的程序、文件和库,同时,还含有简单的程序,孤立的 MAKE 实用程序,以及上下文有关的帮助文件。用户可根据自己的需要生成用于只有一个驱动器、或者两个驱动器、或者一个驱动器和一个硬盘的 Turbo C 系统软件。因而,不管是一个软盘驱动器,或者两个软盘驱动器、还是一个软盘驱动器和一个硬盘的情况,均能运行 Turbo C 程序。

Turbo C 支持美国国家标准学会(ANSI)提出的 C 标准,也完全支持克尼汉(Kernighan)和里奇(Ritchie)的定义,包括开发 PC 机所使用的混合语言和混合模式的程序设计的某些扩充。正因为 Turbo C 具有上述这样明显的优越性,正在引起人们广泛的重视,如果把 Turbo C 应用与 Turbo prolog 进行连接,还能增加 Turbo C 应用人工智能的能力,并能大大减少软件开发时间,增加代码的透明度和程序的适应性。

二. Turbo C 的配置盘与系统的安装

1. Turbo C 的配置盘

Turbo C 程序包放置于四片配置盘上。#1 盘含有 Turbo C 使用的核心文件,即 Turbo C 集

成环境, Turbo C 帮助文件, 以及程序包的最新注解与错误; #2 盘含有 Turbo C 的命令行以及命令行所要使用的其它实用程序; #3 盘含有浮点库, 包含头文件, 以及用于微型存储模式、小型存储模式、大型存储模式的运行时间库; #4 盘含有用于压缩型存储模式、中型存储模式、以及巨型存储模式的运行时间库, 建立用户启动文件的支持文件, 以及一些杂文件。这四张配置盘的具体内容如下所示:

#1 盘: 集成环境

README	COM	4066	5-13-87	1,00a
TC	EXE	231257	5-13-87	1,00a
TCINST	COM	51288	5-13-87	1,00a
TCHelp	TCH	42229	5-13-87	1,00a
HELLO	C	104	5-13-87	1,00a
README		13239	5-13-87	1,00a
NONAME	OBJ	139	1-01-80	1,34a

7 File(s) 16384 bytes free

#2 盘: 命令行与实用程序

TCC	EXE	169098	5-13-87	1,00a
CPP	EXE	231257	5-13-87	1,00a
MAKE	EXE	18108	5-13-87	1,00a
TLINK	EXE	9753	5-13-87	1,00a
TOUCH	COM	3992	5-13-87	1,00a
CNVTCFG	EXE	14102	5-13-87	1,00a
BUILD-CO	BAT	662	5-13-87	1,00a
CO	ASM	9904	5-13-87	1,00a
RULES	ASI	7047	5-13-87	1,00a
SETENV	ASM	2208	5-13-87	1,00a
SETARGV	ASM	5061	5-13-87	1,00a
MAIN	C	1050	5-13-87	1,00a

12 File(s) 59392 bytes free

#3 盘: 包含文件与库 1

ALLOC	H	902	5-13-87	1,00a
ASSERT	H	275	5-13-87	1,00a
BIOS	H	530	5-13-87	1,00a
CONIO	H	520	5-13-87	1,00a
CTYPE	H	1348	5-13-87	1,00a
DIR	H	1225	5-13-87	1,00a
DOS	H	7322	5-13-87	1,00a
ERRNO	H	2651	5-13-87	1,00a
FCNTL	H	994	5-13-87	1,00a
FLOAT	H	2085	5-13-87	1,00a
IO	H	2410	5-13-87	1,00a
LIMITS	H	760	5-13-87	1,00a
MATH	H	2990	5-13-87	1,00a
MEM	H	909	5-13-87	1,00a

PROCESS	H	1785	5-13-87	1:00a
SETJMP	H	545	5-13-87	1:00a
SHARE	H	437	5-13-87	1:00a
SIGNAL	H	676	5-13-87	1:00a
STDARG	H	467	5-13-87	1:00a
STDDEF	H	670	5-13-87	1:00a
STDIO	H	4745	5-13-87	1:00a
STDLIB	H	2302	5-13-87	1:00a
STRING	H	1984	5-13-87	1:00a
TIME	H	839	5-13-87	1:00a
VALUES	H	994	5-13-87	1:00a
COT	OBJ	1389	5-13-87	1:00a
COS	OBJ	1497	5-13-87	1:00a
MATHS	LIB	15872	5-13-87	1:00a
CS	LIB	79872	5-13-87	1:00a
COL	OBJ	1411	5-13-87	1:00a
MATHL	LIB	16384	5-13-87	1:00a
CL	LIB	85504	5-13-87	1:00a
EMU	LIB	20480	5-13-87	1:00a
FP87	LIB	4096	5-13-87	1:00a
MCALC	C	3460	5-13-87	1:00a
MCALC	H	8164	5-13-87	1:00a
MCALC	PRJ	129	5-13-87	1:00a
MCALC	DOC	1012	5-13-87	1:00a
MCOMMAND	C	19393	5-13-87	1:00a
MCDISPLY	C	8190	5-13-87	1:00a
MCINPUT	C	3698	5-13-87	1:00a
MCPARSER	C	12547	5-13-87	1:00a
MCUTIL	C	14925	5-13-87	1:00a
MCMVSMEM	C	1012	5-13-87	1:00a
MCMVSMEM	OBJ	257	5-13-87	1:00a
SYS	(DIR)		5-13-87	1:00a

46 File(s) 2048 bytes free

#4 盘:库 2

C0C	OBJ	1401	5-13-87	1:00a
MATHC	LIB	16384	5-13-87	1:00a
CC	LIB	82944	5-13-87	1:00a
COM	OBJ	1507	5-13-87	1:00a
MATHM	LIB	15872	5-13-87	1:00a
CM	LIB	82944	5-13-87	1:00a
C0H	OBJ	1388	5-13-87	1:00a
MATHH	LIB	16384	5-13-87	1:00a
CH	LIB	84992	5-13-87	1:00a
MATHERR	C	2000	5-13-87	1:00a

FILECOMP	C	11073	5-13-87	1,00a
GETOPT	C	4228	5-13-87	1,00a
CPINTT	OBJ	1996	5-13-87	1,00a
BAR	C	3868	5-13-87	1,00a
PBAR	PRO	2139	5-13-87	1,00a

15 File(s) 28672 bytes free

2. Turbo C 系统的安装

要运行 Turbo C,首先要根据自己的计算机系统的软盘驱动器以及硬盘的配置情况,安装适合自己需要的 Turbo C 系统。

(1)对于使用一个驱动器的系统 对于只有一个驱动器的微机系统,只能使用 Turbo C 的集成环境(在#1 配置盘上名为 TC.EXE 的菜单驱动版本)。当然,虽然只有一个驱动器,也要使用两张盘片——一张程序盘,一张工作盘。

在程序盘中,要装入 TC.EXE 和 TCHELP.TCH 两个文件。

在工作盘中,除了要包含用户的 C 源程序、目标文件和可执行文件之外,还要建立两个分开的子目录——包含文件子目录 INCLUDE,以及库文件子目录 LIB。如果建立了这两个子目录并使用 TCC,则必须指定具有-I 选择项的包含目录和具有-L 选择项的库目录。如果使用 TC,则必须建立选择项/环境菜单中的包含文件目录和库目录。把#3 配置盘上的包含文件(*.H 和 SYS\STAT.H)放进 INCLUDE 子目录中,把下列文件放进 LIB 子目录中:

C0X.OBJ
EMU.LIB
FP87.LIB
MATHX.LIB
CX.LIB

其中,x 为欲用的特定存储模式的第一个字母,例如,如果使用小存储模式,则用 S 替换 x,即把 C0S.OBJ、MATHS.LIB 和 CS.LIB 放进/LIB 子目录中。若使用一个软盘驱动器运行 Turbo C,则要先将程序盘插入驱动器,并键入 TC,就把程序装入了内存。此时,把程序盘取出,将工作盘插进去。如果需要帮助,则在按下 F1 之前,需把程序盘再插进去。

(2)使用两个软盘驱动器的系统 对于具有两个驱动器的系统,既可选用 Turbo C 集成环境运行,也可选用 Turbo C 命令行版本运行。

①对于运行 Turbo C 版本:要运行 Turbo C 集成环境版本(名为 TC.EXE 的菜单驱动版本),则按上述步骤建立一个程序盘和一个工作盘。把程序盘插入 A 驱动器,工作盘插入 B 驱动器。

②对于运行 TCC 版本:要运行 Turbo C 命令行版本(名为 TCC.EXE),则必须建立两个新盘——一个用于 A 驱动器,一个用于 B 驱动器。

(a)在用于 A 驱动器的软盘上放置下列文件:

TCC.EXE
TLINK.EXE
名为\LIB 的子目录
名为\INCLUDE 的子目录

而在\LIB 子目录中放置下列文件:

C0X.OBJ

EMU. LIB
FP87. LIB
MATHX. LIB
CX. LIB

其中, x 含义与前述相同, 在 \INCLUDE 子目录中则放置 #3 盘上所有的包含文件(即 *. H)。

(b) 在用于 B 驱动器的软盘(工作盘)上放置用户用 Turbo C 命令行建立的. C, . OBJ 和 . EXE 文件;

(3) 对于使用硬盘的系统在硬盘上建立并使用 Turbo C 时, 允许用户从集成环境版本(TC. EXE)变换到命令行版本(TCC. EXE)。

首先要建立断开根目录的一个子目录 \TURBOC, 把所有的程序文件和用 Turbo C 建立的用户自己的所有文件以及. C 文件都放进这个子目录中。然后, 断开 \TURBO C 子目录, 再建立 LIB 和 INCLUDE 两个子目录, 把所有的库文件和启动文件(即. LIB 和 COX. OBJ 文件)都放进 LIB 子目录中。把所有头文件(#3 配置盘中的 *. H 文件)放进 INCLUDE 子目录中。

在这里, 应当指出, 使用 TC 时, 应在选择项 \环境菜单中建立 INCLUDE(包含)目录和库目录; 而对使用 TCC 来说, 则要指定具有 -I 命令行选择项的包含目录和具有 -L 选择项的库目录。

第二节 Turbo C 的集成开发环境

应用 Turbo C 集成环境, 建立(编辑)和运行(包括编译、连接和运行)C 源程序很方便。其执行过程如下:

一. 建立 Turbo C 程序

在 DOS 提示符下键入 TC 文件名. C, 即进入编辑插入状态(在屏幕顶部显示主菜单横批, 中部是编辑窗口, 下部是信息窗口, 底部是快速行菜单), 此时, 即可在编辑窗口打入 Turbo C 程序。如键入名为 J1. C 的源程序:

```
/* file: j1. c */  
#include "stdio. h"  
main( )  
{  
    char name[30];  
    printf("what's your name?");  
    scanf("%s", name);  
    printf("Hello, %s\n", name);  
}
```

程序打完之后, 即可进入运行(包括编译、连接、运行一起来), 当然, 也可先存盘后运行。

二. 运行 Turbo C 程序(编译、连接、运行. C 程序)

Turbo C 程序编辑完毕后, 即可用下列方法之一对它编译、连接、运行:

(1)按 ALT-R

(2)按 F10, 再按 R

(3)按 F10, 用移动键把光标移到主菜单 RUN 处, 再按↵

如对所编辑的 j1. C 文件, 一旦编辑完毕, 按 ALT-R, 即在屏幕顶部显示 "What's your name?", 等待输入您的名字, 打入名字后显示 HELLO, 你的名字。按任何键又回到 Turbo C, 如果源程序没有存盘, 此时可把它存盘。

如果编译时发现错误, 则会在信息窗口显示错误信息, 此时, 可按 F10 E 回到编辑插入状态修改源程序。

如果在连接时出现错误, 一般是由于没有告诉库文件在什么地方, 此时, 应按 ALT-O 或 F10 O 进入菜单选择项去指定库目录, 然后再运行。

应当指出, 如果编辑完毕后, 只想把 C 源程序编译为 .OBJ 文件, 则可按下列方法之一进行编译:

(1)按 ALT-9

(2)按 F10, 按 C, 再按 C

(3)按 F10, 再用光标移动键把光标移到主菜单 compiler 处, 按↵(或按 C)编译完成后, 如无错误, 会在屏幕上显示 success, press any key, 即表示编译成功。此时, 按任何键又回到菜单活动状态。若编译出现错误, 同样会在信息窗口示出错误信息, 此时, 应回到编辑状态修改程序。

如果编译成功(即无发现错误)之后, 就可进入运行。因为主菜单在活动, 所以, 只要按 R 即可; 如果未返回到编辑程序, 则要按 ALT-R 或 F10 回到主菜单, 再按 R, 不管怎样, 都需要连接库子程序, 这时, 如果发现错误, 如前所述, 可能是由于没有告诉库文件在什么地方, 如果仍看到连接窗口, 则按 F10, 再按 O(或按 ALT-O)进入菜单选择项, 按 E 选择环境命令, 然后按 L 选择库目录命令; 当库目录窗口出现时, 打入库文件的完整的子目录路径名(也可能是 A:\TURBOC\LIB 或 C:\TURB\LIB), 按↵。

按 ESC 离开环境菜单, 再按 S(Store 选择项), 则出现配置文件提示框, 框中有 TCCONFIG. TC 名字, 只要按↵, 若 Turbo C 询问是否需要全写 TCCONFIG. TC, 则键入 Y, 最后按 ESC 离开选择菜单, 并按 R 再次运行。此时, Turbo C 应连接成功。若缺省存储模式, 则默认为小模式, 因而, Turbo C 将连接 CS. LIB 库子程序。程序连接之后, Turbo C 就对其运行。

三. 保存 Turbo C 程序(存盘)

Turbo c 程序既可在运行成功之后再保存起来, 也可在编辑完毕(打完程序)之后先保存起来。如要在打完程序之后就保存起来, 则可按 F2, 或 ALT-F S, 或 F10 F S, 或者按 F10, 使用光标移动键把光标移到 File 处, 按↵, 出现下拉菜单, 再把光标移到 Save 处, 按↵即存盘, 存盘后按 F10, 又回到主菜单(活动状态)。

四. Turbo C 程序的修改

1. 进入编辑、插入状态

要修改(插入与删除)源文件, 首先需要进入编辑、插入状态, 这有两种情况:

(1)如果在编译中发现错误, 需要对程序进行修改, 则按 F10 E(或按 F10, 把光标移到 Edit 处, 再按↵), 即进入编辑插入状态, 此时, 就可根据需要对程序进行修改。

(2)如果要对在磁盘中保存的老文件进行修改, 则需要按下述方式先把老文件从磁盘装入

内存,然后再进行修改:

①在编辑状态下,按 F3(或按 F10 F L),出现 Load file name 提示框,此时,打入需要装入的文件名.C(注意,如果文件存于 A 盘,则需打入 A:文件名.C;如果存于 B 盘,则要打入 B:文件名.C)即装入文件。

②根据主菜单,把光标移到 File 处,按 ↵,即出现下拉菜单,再把光标移到 Load 处,按 ↵,出现如上所述的提示框,此时,按上述一样的方法键入所需调用的文件名.C 即可。

2. 修改 Turbo C 程序

在编辑、插入状态下,可用下列删除、插入命令修改程序(文件):

(1)插入:

①插入字:用上、下、左、右箭头键移动光标到所需插入字处的右边,键入所要插入的字符即可。

②插入行:把光标移到插入行之前,按 Ctrl-N 即可插入一行。

(2)删除

①删除字:把光标移到所需删除的字符处,按 Del 键。

②删除行:把光标移到所要删除的行首,按 Ctrl-Y。

(3)修改 Turbo C 程序示例

在 DOS 提示符下

C>tc J1.C ↵(把上述的老文件 J1.C 调出修改)

J1.C 文件调出后,可修改成如下程序:

```
/* file:j2.c */
#include "alloc.h"
#include "stdio.h"
main()
{
    char * name;
    char msg[10];
    name=(char *)malloc(10)
    printf("what's your name?");
    scanf("%s",name);
    strcpy(msg,"Hello,");
    printf("%s %s",msg,name);
}
```

程序修改完毕后,可按上述一样的方法存盘和运行,其运行结果与 J1.C 的一样。注意,如要把修改后的程序改名为 J2.C 存盘,则可在修改完成后按 F10 F W,再按照提示打入

J2.C ↵即可。

如果欲建立一个新程序(名为 sum.c),则可在修改完程序后按 F10 E(或 Alt-E)进入编辑窗口(或按 F10 E N 清除屏幕),即可打入程序如下:

```
/* file:sum.c */
#include "stdio.h"
main()
```

```

{
int a,b,sum;
printf("Enter two numbers:");
scanf("%d %d",&a,&b);
sum=a+b;
printf("The sum is %d/n",sum);
}

```

程序打完后,按上述方法运行,当运行后屏幕出现 Enter two number:,此时,键入两个整数,即得所求结果。注意,键入的数据项之间要用空格与/或 tabs 与/或回车来分隔。

五. 退出 Turbo C 系统

程序执行(编译或编译、连接、运行)完毕,可按 F10 Alt-X,或 F10 F Q,或 F10 F Alt 退出 Turbo C 系统,回到 DOS 状态。

第三节 Turbo C 命令行的使用与任选项的选择

一. 命令行的使用

1. 任选项的类型及其与菜单选择的关系

在 Turbo C 中,命令行任选项有三种类型:

- (1)编译程序任选项
- (2)连接程序任选项
- (3)环境任选项

其中,编译程序任选项又分为六类:

- ①存储模式任选项
- ②宏定义任选项
- ③处理器(CPU)任选项
- ④源代码任选项
- ⑤错误报告任选项
- ⑥段名任选项

命令行任选项与菜单选择的相互关系如表 1-1 所示。

表 1-1 命令行任选项与菜单选择的相互关系

命令行开关	菜单选择
-A	O/C 源只有 ANSI 关键字...ON(接通)
-a	O/C/代码生成/排列成行...字
-a- **	O/C/代码生成/排列成行...字节
-B	(未使用)
-C	O/C/源/嵌套注解...on(接通)
-c	编译/编译成 OBJ
-Dname	O/C/定义

_Dname==string	O/C/定义
_d * *	O/C/代码生成/合并完全一样的字串
efilename	工程/工程名
-f * *	O/C/代码生成/浮点...仿真
-f-	O/C/代码生成/浮点...什么也没有
-f87	O/C/代码生成/浮点...8087
-G	O/C/最优化/优化为...速度
-g#	O/C/错误/警告:在...#后停止
--lpathname	O/E/包含目录
--i#	O/E/标识符长度...#
-j#	O/C/错误/错误:在...#后停止
--k	O/C/代码生成/缺省字符类型...无符号的
_k- * *	O/C/代码生成/缺省字符类型...带符号的
_Lpathname	O/E/库目录
-M	O/L/映象文件
_mc	O/C/模式...压缩型
_mh	O/C/模式...巨型
_ml	O/C/模式...大型
_mm	O/C/模式...中型
_ms * *	O/C/模式...小型
_mt	O/C/模式...微型
-N	O/N/代码生成/测试堆栈溢出...on(接通)
_npathname	O/E/输出目录
--O	O/C/最优化/优化为...长度
_ofilename	(未使用)
-P	O/C/代码生成/调用规定...Pascal
-p- * *	C/代码生成/调用规定...C
_r * *	O/C/最优化/使用寄存器变量...on(接通)
--s	(未使用)
_Uname	(未使用)
-W	O/C/错误显示警告...on(接通)
-W-	O/C/错误/显示警告...off(断开)
_wxxx	O/C/错误/可移植性警告或违反 ANSI 或普通错误或少量的普通错误...on(接通)
-w-xxx	O/C/错误/可移植性警告或违反 ANSI 或 普通错误或少量普通错误...off(断开)
-Y	O/C/代码生成/标准的堆栈结构...on(接通)
-y	O/C/代码生成/行号...on(接通)
-z	O/C/最优化/寄存器优化...on(接通)
-zAname	O/C/名称/代码/类
-zBname	O/C/名称/数据/类
-zCname	O/C/名称/代码/段
-zDname	O/C/名称/BSS/段
-zGname	O/C/名称/数据/组
-zPname	O/C/名称/代码/组
-zRname	O/C/名称/数据/段

-zSname	O/C/名称/BSS/组
-zTname	O/C/名称/BSS/类
-l	O/C/代码生成...80186/80286
-l- * *	O/C/代码生成...8088/8086

O=任选 C=编译程序 E=环境 * *=接通缺省

2. 任选项的接通与断开

命令行任选项是通过列入一个破折号(-)紧跟任选字母来选定的(如-l)。要断开一个任选项,则在任选字母后面加上另一个破折号即可。例如-A 则接通 ANSI 关键字任选项,而-A -则断开了。这一特性对于禁止和启动单独的开关是有用的,因此,使得在配置文件中的相应设置无效。

3. 命令行的语法

通过 DOS 命令行选择 Turbo C 编译程序任选项,采用下列语法,tcc [任选项...] 文件名 文件名...

编译程序按照下列规则集编译文件:

文件名.asm 调用 MASM 来汇编成 OBJ
 文件名.obj 包含作为连接时间的目标
 文件名.lib 包含作为连接时间的库
 文件名 编译文件名.C
 文件名.C 编译文件名.C
 文件名.xyz 编译文件名.xyz

例如,给定下列命令行:

tcc -a -f -c -o -z -emyexe oldfile1.c oldfile2 nextfile.c TCC 就把 OLDFILE1.C、OLD-FILE2.C 和 NEXTFILE.C 编译成.OBJ,生成一个名为 MYEXE.EXE 的可执行程序,其带有所选定的字排列成行(-a)、浮点仿真(-f)、嵌套注解(-c)、跳转最优化(-o)和寄存器最优化(-z)任选项。

如果在命令行上给出一个.ASM 文件,或者如果一个.C 文件包含在线汇编,则 TCC 将要调用 MASM。开关 TCC 给 MASM 是

/mx/D-mdl-

其中 mdl 是 TINY、SMALL、MEDIUM、COMPACT、LARGE 或 HUGE 之一。/mx 开关告诉 MASM 用情况灵敏度接通来汇编。

二. 任选项的选择

1. 存储模式任选项

-mc 利用压缩型存储模式编译。
 -mb 利用巨型存储模式编译。
 -ml 利用大型存储模式编译。
 -mm 利用中型存储模式编译。
 -ms 利用小型存储模式编译(缺省)。
 -mt 利用微型存储模式编译,生成几乎象小型存储模式一样的代码,但在执行任何连接时利用 C0T.OBJ 来生成一个微型模式程序。

2. 宏定义任选项

- Dxxx 定义指定的标识符 xxx 成为由单个空白符()组成的字符串。
- Dxxx=string 定义指定的标识符 xxx 成为等号后面的字符串 string。string 不能含有任何空格或制表符
- Uxxx 未定义任何上述的所指定标识符 xxx 的定义。

3. 处理器任选项

- i 使得 Turbo C 生成扩充的 80186 指令。当生成的 80286 程序在非保护方式(如用 IBM PC/AT 在 MS-DOS3.0 下)运行时,也使用这个任选项。
- a 强迫整数长度项在机器字范围定位。附加字节将要插入一个结构,以保证字段排列成行。自动变量和全程变量将要严格地定位。字符变量和无符号字符变量和字段可以放置于任何地址,所有其它变量必须放置于偶数地址。
- f87 利用在线 8087 指令而不是利用 8087 仿真库子程序产生浮点运算。指定浮点处理器可在运行时间使用,因此,用这个任选项编译的程序将不能在无浮点芯片的机器上运行。
- f 如果运行时间系统没有 8087,则在运行时间调用仿真 8087,如果有 8087,则调用 8087 以进行浮点计算(缺省)。
- f- 指定程序不包含浮点计算,因此,在连接阶段不连接浮点库。

4. 源代码任选项

- A 建立 ANSI 兼容代码:忽略任何一个 Turbo C 扩充关键字,而可用作正常的标识符。这些关键字包括:
near far huge cdecl
asm pascal interrupt
_es _ds _cs _ss
以及寄存器伪变量,如 _Ax, _Bx, _SI, 等。
- c 允许嵌套注解,注解可以不正常地嵌套。
- i# 使得编译程序只识别标识符的前 # 个字符。只有所有的标识符,不管是变量、预处理程序宏名字,还是结构成员名的前 # 个字符不同时,才作为不同的标识符来处理。用缺省, Turbo C 每个标识符使用 32 个字符。其它系统,包括 UNIX 系统,则忽略前 8 个字符以外的字符。
- k 使编译程序把所有字符说明作为无符号的字符类型来处理。这允许与作为无符号的字符说明处理的其它编译程序兼容。用缺省,则作为带符号的字符说明处理。

5. 代码任选项

- d 当一字符串与另一字符串匹配时,则合并字符串。这产生较小的程序(缺省)。
- G 使编译程序建立较快的子程序,而不是建立较小的子程序。
- N 在每个函数的入口产生堆栈溢出逻辑:这将使得当发现堆栈溢出时,显示堆栈溢出信息。这在程序长度和速度上都是昂贵的,但它是作为一个任选项提供,因为堆栈溢出可能很难发现。如果发现溢出,就打印信息“Stack overflow!”(堆栈溢出!),而且程序用出口代码 1 退出。
- y 包含在目标文件中供符号调试程序使用的行号。这增加目标文件的长度,但不影响可执行程序的长度和速度。这个任选项只与能使用信息的符号调试程序一起使用。
- O 优化删除过多跳转的长度,并识别循环和开关语句。
- P 强迫编译程序使用 pascal 参数传递顺序来产生所有子程序调用和所有函数。结果的函数调用较小和较快。函数必须传递参数的正确个数和类型,与正常的 C 语言用法不同,这允许可变的函数参数个数。可以使用 cdecl 语句来使这个任选项无效,和特别地说明函数是 C 类型。
- r- 抑制使用寄存器变量。在这个任选项下,忽略所有 register 关键字。如果具有不保持 SI 和 DI 寄存器值的一些汇编语言代码,则 -r- 将允许从 Turbo C 调用该代码。通常,抑制寄存器变量执行减少被生成代码的效率,但希望使用现有的子程序时,可能是需要的,应当注意,当使用 -r- 任选项时,用 -r- 编译的源文件,能调用没有 -r- 编译的源文件的代码,如:运行时间库中的子程序。然而,反之则不然,因此,用 -r- 编译的文件,也只能从 -r- 编译的文件中调用。
- r 启动使用寄存器变量(缺省)。