

BORLAND® C++ 3.0

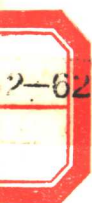
Turbo Assembler 3.0

用 户 手 册

袁 荣 等 编 译
希 望 审 校



海洋出版社



北京希望电脑公司 Borland C++ 3.0 系列丛书之十一

Turbo Assembler 3.0

用户手册

袁 荣 等编译
希 望 审校

11

海洋出版社

1992 年 · 北京

内 容 简 介

Turbo Assembler 支持 Borland C++ 3.0 平台的汇编语言, 支持编写低层功能和优化代码的一切特性, 支持 i486 的所有指令。

本书全面系统地介绍了 Turbo Assembler 的功能和用法。第一部分共分十九章, 介绍了 Turbo Assembler 命令行的指令和开关; 讨论了利用汇编进行程序设计的基本概念; 特殊地, 介绍了如何用汇编编写面向对象的程序; 介绍了如何使用表达式、符号值、处理器指令和符号、程序模型和分段、数据类型、位置计数器、过程、符号作用域、数据、宏、条件指令等汇编语言要素来编写程序; 说明了 Turbo Assembler 与连接器的接口; 讲述了如何产生各种不同的列表; 最后, 介绍了 Turbo Assembler 与 Borland C++、Turbo Pascal 的接口。第二部分是附录, 介绍了各种程序的不同设置(blueprints), 罗列了 Turbo Assembler 的语法; 讨论了兼容性问题; 说明了实用程序的功能和用法; 列出了错误信息。

本书是 Borland C++ 3.0 系列丛书之一, 需要本书和整套丛书者请与北京 8721 信箱联系, 邮编 100080, 电话 2562329。

(京)新登字 087 号

责任编辑: 阎世尊

Turbo Assembler 3.0 用户手册

袁 荣 等编译
希 望 审校

海洋出版社(北京市复兴门外大街 1 号)

海洋出版社发行 双青印刷厂印刷

开本: 787x1092 1/16 印张: 15.625 字数: 362 千字

1992 年 1 月第一版 1992 年 1 月第一次印刷

印数: 1-3000

ISBN 7-5027-2609-8/TP.90 定价: 12.0 元

郑 重 说 明

Borland(宝兰国际有限公司)授权我中科院九州计算机网络公司为 Borland 程序语言和应用软件之中国代理。

北京希望公司此中文手册的出版，已通过我公司与 Borland 的商讨，得到 Borland 的认可。

该手册仅限于和相应的 Borland 软件配套发行，严禁私自翻印和单独发行。

中科院九州计算机网络公司

1992 年 9 月

地 址：北京海淀白石桥路25号

邮 编：100081

电 话：8311822 8420320

特此函告

中国科学院计算中心
九州计算机网络公司

1992.9.19

目 录

内容简介	1
0.1 硬件和软件需求	1
0.2 DPMI 支持	2
0.3 有关本手册	2
0.4 符号约定	3
第一章 初启	4
1.1 安装 Turbo Assembler	4
1.2 联机正文文件	4
1.2.1 README 文件	5
1.2.2 HELPMEL.TSM 文件	5
1.2.3 H2ASH.TSM 文件	5
1.2.4 TCREF.TSM 文件	5
1.3 联机帮助	5
1.4 编写第一个 Turbo Assembler 用户程序	5
1.4.1 汇编第一个程序	6
1.4.2 链接第一个程序	7
1.4.3 运行第一个用户程序	7
1.4.4 发生了什么?	7
1.5 修改第一个 Turbo Assembler 程序	8
1.5.1 将输出送往打印机	9
1.6 编写第二个 Turbo Assembler 用户程序	10
1.6.1 运行 REVERSE.ASM	11
1.7 推荐的读物	12
第二章 使用伪指令和开关	13
2.1 在 DOS 中启动 Turbo Assembler	13
2.2 命令行选择项	15
2.3 /a	15
2.4 /b	16
2.5 /c	16
2.6 /d	16
2.7 /e	16
2.8 /h 或 /?	17
2.9 /i	17

2.10	/j	18
2.11	/kh	18
2.12	/l	18
2.13	/la	19
2.14	/m	19
2.15	/ml	19
2.16	/mu	19
2.17	/mv#	20
2.18	/mx	20
2.19	/n	20
2.20	/o	21
2.21	/op	21
2.22	/p	21
2.23	/q	21
2.24	/r	21
2.25	/s	22
2.26	/t	22
2.27	/v	22
2.28	/w	22
2.29	/X	23
2.30	/Z	24
2.31	/zd	24
2.32	/zi	24
2.33	/zn	24
2.34	间接命令文件	25
2.35	配置文件	25
第三章 程序设计概念		26
3.1	Turbo Assembler Ideal 方式	26
3.1.1	为什么要使用 Ideal 方式?	26
3.1.2	进入和退出 Ideal 方式	27
3.1.3	MASM 方式与 Ideal 方式之间的区别	28
3.1.4	段和段组	29
3.2	注释程序	31
3.2.1	在行尾加注释	31
3.2.2	注释伪指令	31
3.3	扩充行	31
3.4	使用 INCLUDE 文件	32
3.5	预定义符号	33
3.6	符号赋值	33

3.7 通用模块结构	34
3.7.1 VERVION 伪指令	34
3.8 汇编期间的消息显示	35
3.9 显示警告信息	35
3.10 多个错误信息报告	36
第四章 建立面向对象程序	37
4.1 术语	37
4.2 在 Turbo Assembler 为何用对象?	37
4.3 对象是什么?	37
4.3.1 样本对象	38
4.3.2 声明对象	38
4.4 声明一个方法过程	40
4.5 虚方法表	41
4.5.1 初始化虚方法表	42
4.6 调用对象方法	42
4.6.1 调用静态方法	42
4.6.2 调用虚方法	43
4.6.3 调用祖先虚方法	45
4.6.4 关于调用方法	46
4.7 建立对象实例	46
4.8 编程格式	46
4.9 程序示例	47
第五章 使用表达式和符号值	48
5.1 常数	48
5.1.1 数字常数	48
5.1.2 串常数	49
5.2 符号	49
5.2.1 符号名	49
5.2.2 符号类型	49
5.2.3 简单的地址子类型	50
5.2.4 描述复杂的地址子类型	51
5.3 表达式	52
5.3.1 表达式精度	52
5.3.2 表达式中的常数	53
5.3.3 表达式中的符号	53
5.3.4 通用算术算子	56
第六章 选择处理器伪指令和符号	63

6.1	iApX86 处理器伪指令	63
6.2	预定义符	64
6.3	@CPU	64
6.4	@Wordsize	65
6.5	8087 协处理器伪指令	65
6.6	协处理器仿真伪指令	66
第七章	使用程序模式和段	67
7.1	MODEL 伪指令	67
7.1.1	MODEL 伪指令建立的符号	70
7.1.2	简化的段伪指令	71
7.1.3	STARTUPCODE 伪指令	72
7.2	定义类段和组	72
7.2.1	SEGMENT 伪指令	73
7.2.2	ENDS 伪指令	75
7.2.3	GROUP 伪指令	75
7.3	ASSUME 伪指令	75
7.3.1	段顺序	76
7.3.2	改变栈长度	77
第八章	定义数据类型	78
8.1	定义枚举数据类型	78
8.2	定义位域记录	79
8.3	定义结构和联合	80
8.3.1	打开结构或联合定义	80
8.3.2	指定结构和联合成员	80
8.3.3	关闭结构或联合定义	81
8.3.4	嵌套结构和联合	81
8.3.5	在其它结构中包含一个命名的结构	82
8.3.6	在表达式中使用结构名	83
8.4	定义表	83
8.4.1	覆盖表成员	85
8.5	定义命名类型	85
8.6	定义对象	85
8.6.1	TBLPTR 伪指令	86
8.6.2	扩充的 STRUC 伪指令定义的符号	86
第九章	设置和使用位置计数器	87
9.1	\$ 位置计数器符号	87
9.2	位置计数器伪指令	87

9.2.1	ORG 伪指令	87
9.2.2	EVEN 和 EVENDATA 伪指令	88
9.2.3	ALIGN 伪指令	89
9.3	定义标号	89
9.3.1	: 算子	90
9.3.2	LABEL 伪指令	90
9.3.3	:: 伪指令	90
第十章	声明过程	92
10.1	过程定义语法:	92
10.1.1	声明 NEAR 或 FAR 过程	92
10.1.2	声明过程语言	94
10.1.3	指定语言修饰符	95
10.2	定义参数和局部变量	96
10.2.1	ARG 和 LOCAL 语法	96
10.2.2	ARG 和 LOCAL 变量名的作用域	97
10.2.3	保存寄存器	98
10.3	嵌套过程和作用域规则	98
10.4	声明对象的方法过程	100
第十一章	控制符号作用域	101
11.1	可重定义符号	101
11.2	块的作用域	101
11.2.1	LOCALS 和 NOLOCALS 伪指令	101
11.2.2	MASM 块的作用域	102
11.3	MASM 风格的局部标号	102
第十二章	分配数据	104
12.1	简单数据伪指令	104
12.2	建立结构或联合的实例	107
12.2.1	初始化联合或结构实例	107
12.3	建立记录的实例	110
12.3.1	初始化记录实例	110
12.4	建立枚举的实例	111
12.4.1	初始化枚举数据类型实例	111
12.5	建立表的实例	111
12.5.1	初始化表实例	111
12.6	建立和初始化命名表实例	112
12.7	建立对象的实例	112
12.8	建立对象虚方法表的实例	113

第十三章 高级编码指令	114
13.1 灵巧代码生成: SMART 和 NOSMART	114
13.2 扩充跳转	114
13.3 附加的 80386 LOOP 指令	115
13.4 附加的 80386 ENTER 和 LEAVE 指令	115
13.5 附加返回指令	115
13.6 扩充的 PUSH 和 POP 指令	116
13.6.1 多 PUSH 和 POP	116
13.6.2 指针 PUSH 和 POP	116
13.6.3 在 8086 处理器下压入常数	117
13.7 扩充移位指令	117
13.8 强制段重载: SEGxx 指令	117
13.9 附加的灵巧标志指令	118
13.10 附加的域值操作指令	118
13.10.1 SETFIELD 指令	118
13.10.2 GETFIELD 指令	119
13.11 附加的快速立即乘指令	119
13.12 80386 处理器指令的扩充	120
13.13 利用栈的调用序列	120
13.13.1 调用包含 RETURNS 的过程	121
13.13.2 CALL...Method	121
13.13.3 JMP...METHOD	122
13.14 附加的面向对象程序设计指令	122
第十四章 使用宏	123
14.1 正文宏	123
14.1.1 用 EQU 伪指令定义正文宏	123
14.1.2 串宏操作伪指令	123
14.2 多行宏	125
14.2.1 多行宏体	125
14.2.2 通用多行宏	127
14.2.3 计数重复宏	131
14.2.4 WHILE 伪指令	131
14.2.5 串重复宏	131
14.2.6 %立即宏伪指令	132
14.2.7 列表文件中包含多行宏扩展	133
第十五章 使用条件伪指令	134
15.1 通用条件伪指令语法	134

15.1.1	IFxxx 条件汇编伪指令	134
15.1.2	ELSEIFxxx 条件汇编伪指令	135
15.1.3	ERRxxx 出错伪指令	136
15.2	特定伪指令说明	136
15.2.1	非条件出错伪指令	136
15.2.2	表达式条件伪指令	137
15.2.3	符号定义条件伪指令	137
15.2.4	正文串条件伪指令	138
15.2.5	汇编遍条件	140
15.3	将条件包括在列表文件中	141
第十六章 与 Linker 的接口		142
16.1	定义外部符号	142
16.1.1	特定语言的约定	142
16.1.2	声明公共符号	142
16.1.3	声明库符号	143
16.1.4	声明外部符号	143
16.1.5	声明全程符号	143
16.1.6	定义公共变量	143
16.2	包含库	144
第十七章 产生列表		145
17.1	列表格式	145
17.2	通用列表伪指令	145
17.3	包含文件列表伪指令	146
17.4	条件列表伪指令	146
17.5	宏列表伪指令	147
17.6	交叉引用列表伪指令	147
17.7	改变列表格式参数	148
第十八章 Turbo Assembler 与 Borland C++ 的接口		151
18.1	在 Borland C++ 中调用 Turbo Assembler 函数	151
18.1.1	Borland C++ 与 Turbo Assembler 的接口框架	152
18.1.2	Turbo Assembler 与 Borland C++ 的交互性	164
18.1.3	从 Borland C++ 中调用 Turbo Assembler 函数	172
18.1.4	用汇编语言编写 C++ 成员函数	175
18.1.5	Pascal 调用约定	177
18.2	在 Turbo Assembler 中调用 Borland C++	178
18.2.1	链入 C 的启动码	178
18.2.2	段的设置	179

18.2.3	执行调用	179
18.2.4	在 Turbo Assembler 调用 Borland C++ 函数	180
第十九章	Turbo Assembler 与 Turbo Pascal 的接口	183
19.1	与 Turbo Pascal 共享信息	183
19.1.1	编译伪指令和外部子程序	183
19.1.2	PUBLIC 伪指令	184
19.1.3	EXTRN 伪指令	184
19.1.4	使用段定位	187
19.1.5	无效代码消除	187
19.2	Turbo Pascal 参数传递约定	188
19.2.1	值参	188
19.2.2	变量参数	189
19.2.3	栈的维护	189
19.2.4	存取参数	189
19.3	Turbo Pascal 中的函数结果	192
19.4	为局部数据分配空间	192
19.4.1	分配私有静态存储区	192
19.4.2	分配动态存储区	192
19.5	由 Turbo Pascal 调用汇编语言子程序的例子	194
19.5.1	通用 16 进制转换子程序	194
19.5.2	交换两个变量	197
19.5.3	扫描 DOS 环境	199
附录 A	程序设计蓝图	204
A.1	简化段描述	204
A.2	DOS 程序	206
A.2.1	EXE 程序	206
A.2.2	COM 程序	207
A.3	Windows 程序	207
A.3.1	Windows DLL	208
A.3.2	Windows 应用程序	208
附录 B	Turbo Assembler 语法概要	209
B.1	词法	209
B.2	MASM 方式下的表达式语法	210
B.3	Ideal 方式下的表达式语法	212
附录 C	兼容性问题	215
C.1	一遍与两遍汇编	215

C.2	环境变量	215
C.3	MicroSoft 二进制浮点格式	216
附录 D	Turbo Assembler 实用程序	217
D.1	H2ASH 转换程序	217
附录 E	出错信息	219
E.1	信息性信息	219
E.2	警告和出错信息	219
E.3	致命错误信息	236

内容简介

欢迎使用 Borland 公司开发的多遍扫描汇编器 Turbo Assembler Turbo Assembler 具有超前引用性,其汇编速度可达每分钟 48000 行(在 IBM PS/2 60 模式下)。它与 MASM 兼容,并带有可选的扩展语法的 Ideal 方式。无论是初学者还是有经验的编程者都会欣赏这些特征及其提供的便于汇编语言程计设计的其它特征。以下仅列出其中的一些优点,在后续章节将有详细介绍:

- 面向对象的程序设计
- 32 位模式和栈框架支持
- DOS 保护方式接口 (DPMI) 支持在 Microsoft Windows 下以保护方式运行的 Turbo Assembler
- 全 386 和 i486 支持
- 简化的段指令
- 表支持
- 枚举
- 灵巧的 flag 指令
- 新的结构特性
- 快速立即乘法
- 多行定义支持
- VERSION 说明指令
- 嵌套指令
- 模拟 MASM 的 Quirks
- Turbo Debugger 的全源调试输出
- 内部交叉引用实用程序 (TCREF)
- 配置文件和命令文件
- .h 到 .ash(inc) 文件转换实用程序

Turbo Assembler 是一个功能强大的命令行汇编器,它接收用户的源文件(.asm)并输出目标模块(.OBJ)。用户可使用 Borland 公司开发的高速链接程序 TLINK.EXE 对目标模块进行链接并生成可执行文件(.EXE)。

0.1 硬件和软件需求

Turbo Assembler 在包括 XT、AT、PS/2 以及所有兼容机在内的 IBM PC 序列机上运行。它要求 MS-DOS2.0 或以上版本,并至少配备 256K 内存。

Turbo Assembler 产生的是 8086、80186、80286、386 及 i486 指令代码。它也可产生 8087、80287、387 数字协处理器的浮点数操作(有关 80x86/80x87 序列指令集的更详细的内容,请参看 Intel 公司的其它有关资料)。

0.2 DPMI 支持

Turbo Assembler 支持 DOS 保护方式接口 (DPMI) 说明。DPMI 定义了能充分利用 80286、386 和 i486 处理器的优点的标准接口。这些处理器的保护方式和虚 8086 方式明显改变了我们进行计算的方式。现在可以使用多任务并使用扩展内存, 而在过去, 让应用程序和其它软件一起使用扩展内存进行多任务处理是很困难的。DPMI 标准解决了这个问题, 用 DPMI 标准开发的应用程序可以在多任务操作系统下可靠地运行。

当前, Turbo Assembler 使用 Microsoft Windows 提供的 DPMI 服务, Turbo Assembler 也可以在使用这些 DPMI 服务的其它环境下运行。如果转换到新的环境中, 但仍使用这些服务, 那么你在软件上的投资将得到保护。

TASM.EXE 包含 DPMI 支持, 详见第二章。

0.3 有关本手册

Turbo Assembler 参考资料共有两本: "Turbo Assembler User's Guide" (本书) 和 "Turbo Assembler Quick Reference Guide"。本书提供使用 Turbo Assembler 所需的基本指令, 与其它语言的接口, 详细描述算符。预定义符和 Turbo Assembler 使用的伪指令, 另一本书提供伪指令及其它处理器指令的快速参考。

下面详细介绍用户手册的内容:

第一章: "初启" 介绍如何在你的系统上安装 Turbo Assembler

第二章: "使用伪指令和开关" 描述当使用伪指令和开关时如何控制 Turbo Assembler 的运行方式。

第三章: "程序设计概念" 讨论 Ideal 方式和 MASM 方式之间的差异。如何使用预定义符, 注释等等。

第四章: "建立面向对象的程序" 描述在汇编语言中如何用 OOP 技术。

第五章: "使用表达式和符号值" 讨论表达式和算符的计算及定义。

第六章: "选择处理器伪指令和符号" 讨论如何为特定处理器产生代码。

第七章: "使用程序模式和段" 讨论程序模式, 建立符号、简化段和段排序。

第八章: "定义数据类型" 解释如何定义结构、联合、表、位字段记录和对象。

第九章: "设置和使用定位计数器" 描述如何和为什么要用定义计数器、如何定义标号。

第十章: "声明过程" 检查如何使用各种过程、如何定义和使用参数及局部变量。

第十一章: "控制符号的作用域" 讨论具有特定值的符号的作用域。

第十二章: "分配数据" 介绍简单数据指令、如何建立结构、联合、记录、枚举数据类型、表和对象的实例。

第十三章: "高级编码指令" 覆盖了 Turbo Assembler 的扩展指令。

第十四章: "使用宏" 介绍如何在程序中使用宏。

第十五章: "使用条件伪指令" 介绍条件伪指令。

第十六章: "与 Linker 的接口" 介绍如何包含库和链接时印出哪些符号。

第十七章: "产生列表" 介绍 Turbo Assembler 的列表文件和如何使用它们。

第十八章: "Turbo Assembler 与 Borland C++ 的接口" 介绍如何联合使用汇编语言

和 Borland C++。

第十九章：“Turbo Assembler 与 Turbo Pascal 的接口”介绍与 Turbo Pascal 代码的接口。

附录 A：“程序蓝图”包含不同类型程序结构的实例。

附录 B：“Turbo Assembler 语法概要”举例说明以修改的巴科斯范式给出的表达式 (Ideal 和 MASM 模式下)。

附录 C：“兼容问题”包括 MASM 和 Turbo Assembler MASM 之间的差异。

附录 D：“实用程序”主要介绍.h 到.inc 的转换实用程序。

附录 E：“错误信息”介绍当使用 Turbo Assembler 时所有的出错信息：消息性信息。致命错误信息和警告、出错信息。

0.4 符号约定

谈及 IBM PCs 或其兼容机时，指使用 8088, 8086, 80186, 80286, 386 和 i486 芯片的任何机型，涉及 PC-DOS, DOS 或 MS-DOS 时，指 2.0 版本或更高的版本。

第一章 初 启

你可能听说过,汇编语言程序设计是一种只适于训练有素者或只适于天才的“未知项”。切勿相信!汇编语言只是机器本身具有的一种高级语言。正如你所希望的一样,计算机语言是高度逻辑化的语言;也正如你可能希望的一样,汇编语言具有极其强大的功能——事实上,也只有汇编语言才能触及到 IBM PC 及其兼容机系列的核心处理器——Intel 80x86 系列的全部功能。

仅用汇编语 C、Turbo Pascal 及其它语言编写的程序中。无论在何种方式下,都可用汇编语言写出灵活快速的小型程序。汇编语言代码对计算机各方面操作的控制能力与其执行速度有同样的重要性,它可低级控制到系统时钟的最近一次变化。

本章介绍汇编语言并剖析汇编语言程序设计的独特优点。用户将运行一些汇编语言程序,以便获得对该语言的感性认识,并逐渐习惯使用汇编语言进行编程。

很显然,这几章的讲解并不能使用户成为汇编语言程序设计专家;而只能引导用户熟悉汇编语言并开始编制自己的汇编语言程序。建议用户阅读有关汇编语言程序设计和 PC 机结构方面的专门书籍。另外,IBM 公司的《DOS technical Reference》、《BIOS Interface Technical Reference》是极有用的参考资料;这些手册讲述了汇编语言与 IBM 个人机的系统软件及硬件的接口。

1.1 安装 Turbo Assembler

在进行汇编语言程序设计之前,需用 DOS 拷贝 Turbo Assembler 盘并形成用户自己的“工作”拷贝,这些做完后,将源盘放至安全处。

如果希望用 Turbo Assembler 代替 MASM 那么可参看第三章了解 Turbo Assembler 与 MASM 区别。

INSTALL 盘上有一个名为 INSTALL.EXE 的程序,它将帮助用户安装 Turbo Assembler 3.0,安装方式有两种:

1) 硬盘用户:该项允许用户选择装载文件的子目录。

2) 软盘用户:该项将使用 Turbo Assembler 时所需要的文件安装到双软盘驱动器系统中。在开始安装前必须准备几种格式化的软盘。

在开始安装时,将当前驱动器改换到含有 INSTALL 程序的驱动器并打入 INSTALL,并对屏幕底框中出现的每一提示系统会给出解释。

如果你的计算机带 LCD 显示器,可用 /b 开关,那么它就在黑白模式下工作:

INSTALL /b

1.2 联机正文文件

Turbo Assembler 同时带了几个联机正文文件: FILELIST.DOC, README, HELPMES.TSM, MANUAL.TSM, H2ASH.TSM 和 TCREF.TSM, 头二个文件在标有“Installation Disk”