

Turbo Debugger 3.0

调试手册

李振格 编译

北京航空航天大学出版社

Turbo Debugger 3.0

调试手册

李振格 编译

北京航空航天大学出版社

(京)新登字 166 号

内 容 简 介

Turbo Debugger 3.0 是微机上新型的源级调试器,能调试 Borland 公司的系列语言: Turbo Pascal、Turbo C、以及用 Turbo Assembler 或宏汇编、Turbo C++、Borland C++、Turbo C++ for Windows 编写的程序,能调试任何. EXE、. COM、. SYS、. BIN 文件,能在源程序行上检查、监视、修改变量的值,进行表达式求值,设置条件和无条件断点,单步跟踪执行,能进行大程序双机虚拟调试、远程调试,能在 Microsoft Windows 中直接调试 Windows 应用程序。

Turbo Debugger 3.0

调试手册

Turbo Debugger 3.0

TiáoShì ShǒuCè

李振格 编译

责任编辑 冯学民

北京航空航天大学出版社出版

新华书店总店科技发行所发行 各地新华书店经销

朝阳科普印刷厂印刷

787×1092 1/16 印张:19.25 字数:492 千字

1992 年 9 月第一版 1992 年 9 月第一次印刷 印数:7000 册

ISBN 7-81012-340-8/TP·074 定价:15.30 元

前 言

随着计算机软件技术的不断发展,90年代迎来了面向对象范型(Object-Oriented Paradigm)的软件开发新时代。在C语言基础上发展起来的C++语言就是一种面向对象开发方法的程序设计语言。由于C++提供了把数据和在数据上的操作封装在一起的类、对象和方法的机制,并通过派生、继承、重载和多态性等特性,实现了人们追求已久的软件重用技术,使得软件,特别是大型复杂软件的构造和维护变得更加有效和容易,并使软件的开发能更自然地反映事物的本来面貌,从而大大提高软件开发的效率和质量。

1991年美国Borland公司在原Turbo C++基础上,推出了全新的Borland C++系列软件。Borland C++ 3.0除了实现了AT&T C++ 2.1版本的全部功能外,还支持ANSI C和Microsoft Windows应用软件的开发,因此成为当今国际上最受欢迎的面向对象程序设计软件。

为了让读者对Borland的C++系列产品有一个完整的了解,下面介绍一下Borland公司和它的C++系列产品。

Borland公司到目前为止已开发出了众多的语言产品及其相应的支持产品。在国内用得较多的语言产品有Borland C++、Turbo C++、Turbo C、Turbo Pascal、Turbo Prolog、Turbo Basic和Turbo Assembler。

Borland公司的语言产品除了Turbo Assembler之外,都具有可以归结为如下与众不同的特点:它们都是一个集编辑、编译、调试、运行和剖析优化等功能于一体的具有联机帮助和热键触发等特点的优秀的软件开发环境。

Borland公司与语言产品配套的实用程序也非常有名,Turbo Debugger和Turbo Profiler等程序与语言产品形成一个不可分割的整体。

到目前为止,Borland已经发行了许多种C++包装,其中有些只针对单个程序员,另外一些针对专业开发者。不同包装的编译器之间在功能上有差别。

☐ Turbo C++

Turbo C++编译器可以创建K&R、ANSI和C++程序,它是以AT&T C++ 2.0为模板实现的,软件包含了Borland类库,但不包含Turbo Vision和ObjectWindows。虽然它不失为一个非常好的个人编译器,但它不能产生运行于Windows下的应用程序,而且不包含独立的Turbo Debugger、Turbo Assembler和Turbo Profiler。

☐ Turbo C++ & Turbo Vision

该编译器不仅包含Turbo C++编译器所有成分,而且包含面向对象的Turbo Vision库。该库是编写以DOS为基础的面向文本模式用户界面的工具,利用该类库可以完成窗口、菜单、编辑控制和其它的界面元素设计。对于不关心Windows程序设计的个人来说,该软件包已经足够了。

☐ Borland C++ 2.0

该编译器保留了Turbo C++编译器的所有性能,并且还包含了独立的Turbo Debugger、

Turbo Assembler 和 Turbo Profiler。虽然 Borland C++ 2.0 不提供 Turbo Vision 和 ObjectWindows,但它足以编写各种 Windows 应用程序。

Whitewater Resource Toolkit(WRT)使 Borland C++ 2.0 能够创建 Windows 资源(图标、光标、位图、菜单、字体和字符串),它是一个能很好地与 Borland 产品和与 Windows 兼容的编译器一起工作的资源管理工具。

☐ Borland C++ 2.0 & Application Frameworks

该软件简称为 Borland C++ 2.0 & AF,虽然它仍然是运行于 DOS 的 IDE,但是它提供了专业的开发者用来创建 DOS 和 Windows 应用程序的任何工具和环境。

Borland C++ 2.0 & AF 包含了 Borland C++ 2.0 编译器、DOS 应用程序的类库 Turbo Vision 和 Windows 应用程序的类库 ObjectWindows,是 Borland C++ 2.0 平台上最完整的发行包装。

☐ Borland C++ 3.0

Borland C++ 3.0 提供了一个专业开发人员的集成环境,能生成高质量的 C 和 C++ 代码、DOS 和 Windows 代码。在 Borland C++ 3.0 中,C++ 支持提高到 AT&T C++ 2.1 版本的水平。这种包装既包含 Borland C++ 3.0 编译器,也包含 Turbo C++ for Windows 编译器,支持宿主 Windows 的用户界面,提供了所有的 Windows 和 DOS 开发工具、Turbo Assembler、Turbo Profiler 和 Turbo Debugger,但并不提供 Turbo Vision 和 ObjectWindows。Borland C++ 和独立的调试器仍使用 DOS 的文本接口,但其中的 Turbo C++ for Windows 是一个真正的 Windows 图形用户界面(GUI)的应用程序,并且新增了 Resource Workshop 资源管理程序。在 Windows 的 Program Manager 的 Borland C++ 3.0 程序组中,具有支持 Windows 的程序 Turbo C++ for Windows、Resource Workshop、Winsight、ImportLib、Turbo Debugger for Windows、WRremote、WRSetup、Turbo Profiler for Windows 和 FConvert。

☐ Borland C++ 3.0 & Application Frameworks

这种包装是读者最好的选择,它能运行于 Windows,并且提供了专业的开发者用来创建 DOS 和 Windows 应用程序的任何工具和环境,其中包含新的 Turbo Vision 和 ObjectWindows、类库和 C 库函数的源代码。

为了系统和全面地使用 Borland 在 C++ 3.0 平台上的系列产品的功能,在《Borland C++ & Turbo C++ 用户手册》、《Borland C++ & Turbo C++ 库函数参考手册》和《Borland C++ & Turbo C++ 程序员手册》(它们针对 Borland C++ 2.0 和 Turbo C++ 1.x)基础上,把 Borland 的 C++ 3.0 平台新增的内容整理成二本书,它们就是《Borland C++ 3.0 & Turbo C++ for Windows 3.0 用户手册》和《Borland C++ 3.0 & Turbo C++ for Windows 3.0 程序员手册》。

《Borland C++ 3.0 & Turbo C++ for Windows 3.0 用户手册》分成四篇,第一篇介绍了 Turbo C++ for Windows 集成环境的用法,讨论了预编译头文件功能和用法。第二篇介绍了 Borland C++ 3.0 集成环境和命令行编译器的用法。在 Borland C++ 3.0 的系列软件中用 Resource Workshop 资源管理程序代替了 Borland C++ 2.0 中的 Whitewater Resource Toolkit,第三篇全面系统地论述 Resource Workshop 的功能和用法,说明了如何创建和管理 Windows 的对话框(dialog box)、菜单(menu)、加速器(热键)(accelerator)、字符串表(string table)、位图(bitmap)、图标(icon)、光标(cursor)、字体(font)、用户自定义和 fcdData 资源。第四篇按字母顺序介绍了 Borland C++ 3.0 新增和修改的 C 库函数,罗列了它们的功能、用法、返回值和示例。

《Borland C++ 3.0 & Turbo C++ for Windows 3.0 程序员手册》分成二篇,第一篇从程序设计的角度说明了 Turbo C++ for Windows 3.0 中的C++的语法,论述了C++的要素和特性,讨论了预处理器的功能和用法,介绍了C++流的概念、作用、内涵和用法,讲述了创建 Windows 应用程序的过程;ObjectWindows 是 Borland C++ 3.0 中用来开发 Windows 应用软件的类库,利用该类库,用户用很少的代码就可以设计出完美的 Windows 应用程序的用户界面。第二篇的前面部分从介绍开发一个从简单到复杂 Windows 应用程序的步骤入手,全面地介绍了 ObjectWindows 功能和用法;后面部分按字母顺序列出了关于类库的类和流式类、它们的成员和用法的描述;最后还给出了全局变量和辅助类的描述。

Borland C++ 系列软件包含许多实用程序,《Turbo Debugger 3.0 试调手册》所描述的是其中最重要的实用程序 Turbo Debugger。书中介绍了如何使用 Turbo Debugger 菜单,如何在运行时在源程序上检查、监视和修改变量的值,如何进行表达式的求值,如何设置条件和无条件断点,如何进行单步跟踪,如何进行大程序双机虚拟调试、远程调试,如何调试设备驱动程序和 TSR,如何用 TDW 调试用 Borland 系列语言编写 Windows 应用程序等,最后给出了错误信息。

在本套书的编译过程中,张志辉、李毅、吕肖庆、黄磊光、吕良双、汪文、章忆文等同志给予了极大的支持。在此一并致谢。

由于编译者水平有限,加上时间仓促,书中难免有缺点和错误,欢迎广大读者给予批评和指正。

编译者

1992 年于北航计算中心

目 录

前 言

第 0 章 简 介	1
0.1 3.0 版的新性能和所作的修改	1
0.2 硬件及软件需求	2
0.3 术语解释	2
0.4 手册内容	3
第一章 起 步	5
1.1 产品磁盘和联机文本文件	5
1.1.1 README 文件	5
1.1.2 HELPMEL.TD 文件	5
1.1.3 MANUAL.TD 文件	5
1.2 Turbo Debugger 实用程序	6
1.3 安装 Turbo Debugger	6
1.3.1 安装 TDW 的 TDDEBUG.386	6
1.4 DOS 硬件调试	7
1.4.1 DOS 硬件调试	7
1.4.2 Windows 应用程序和硬件调试	7
1.5 阅读建议	7
1.5.1 正在学习 Turbo 语言的程序员	7
1.5.2 已经用过 Turbo 语言的程序员	8
第二章 调试与 Turbo Debugger	9
2.1 有错误吗	9
2.2 错误在哪儿	9
2.3 是什么错误	9
2.4 改正错误	9
2.5 Turbo Debugger 能帮用户做什么	10
2.5.1 Turbo Debugger 不能做什么	11
2.5.2 Turbo Debugger 是怎样工作的	11
2.6 Turbo Debugger 的优势	11
2.6.1 菜单与对话框	12
2.6.2 了解现在在哪里	13
2.6.3 历史经验	15
2.6.4 递增匹配	16
2.6.5 生成宏	16
2.6.6 窗口介绍	16

2.6.7	复制和传输	24
2.6.8	获取帮助	27
第三章	一个快速示例	31
3.1	演示程序	31
3.2	使用 Turbo Debugger	32
3.2.1	菜单(menus)	32
3.2.2	状态行(status line)	33
3.2.3	窗口(windows)	33
3.3	使用 C 演示程序	34
3.3.1	设置断点(Setting breakpoints)	35
3.3.2	使用监视	36
3.3.3	检查简单的 C 数据对象	37
3.3.4	检查复杂的 C 数据的对象	37
3.3.5	改变 C 数据值	38
3.4	使用 Pascal 示例程序	39
3.4.1	设置断点(Setting breakpoints)	40
3.4.2	使用监视(Using watches)	41
3.4.3	检查简单的 Pascal 数据对象	42
3.4.4	检查复杂的 Pascal 数据对象	42
3.4.5	改变 Pascal 数据值	43
第四章	启动 Turbo Debugger	45
4.1	准备待调试的程序	45
4.1.1	准备 Borland C++ 程序	45
4.1.2	准备 Turbo Pascal 程序	45
4.1.3	准备 Turbo 汇编程序	45
4.2	运行 Turbo Debugger	46
4.3	命令行选择项	46
4.3.1	装载配置文件(-c)	47
4.3.2	显示更新方式(-d)	47
4.3.3	获取帮助(-h 与-?)	47
4.3.4	进程 ID 转换(-i)	47
4.3.5	击键记录(-k)	47
4.3.6	汇编模式启动(-l)	47
4.3.7	设置堆大小(-m)	47
4.3.8	鼠标器支持(-p)	48
4.3.9	远程调试(-r)	48
4.3.10	源代码处理(-s)	48
4.3.11	视频硬件(-v)	48
4.3.12	远程窗口调试(-w)	49
4.3.13	覆盖池大小(-y)	49

4.4	配置文件	49
4.5	选项菜单 (Option Menu)	50
4.5.1	语言命令	50
4.5.2	宏菜单	50
4.5.3	显示选项命令 (Display Options)	51
4.5.4	Path for Source 命令	52
4.5.5	保存选择项命令 (Save Options)	52
4.5.6	恢复选择项命令	53
4.6	在 Turbo Debugger 中运行 DOS	53
4.7	返回 DOS	53
第五章	控制程序执行	54
5.1	检查当前程序状态	54
5.1.1	变量窗口	54
5.1.2	堆栈窗口 (Stack Window)	57
5.1.3	源 (Origin) 局部菜单命令	59
5.1.4	获取信息 (Get Info) 命令	59
5.2	运行菜单	63
5.2.1	运行 (Run, F9)	63
5.2.2	执行到光标位置 (Goto Cursor, F4)	63
5.2.3	跟踪进入 (Trace Into, F7)	63
5.2.4	执行一步 (Step Over, F8)	64
5.2.5	执行到某处 (Execute to... , Alt-F9)	65
5.2.6	执行到返回 (Until Return, Alt-F8)	65
5.2.7	连续慢速执行 (Animate...)	65
5.2.8	反向跟踪 (Back Trace, Alt-F4)	65
5.2.9	指令跟踪 (Instruction Trace, Alt-F7)	65
5.2.10	参数 (Argument...)	65
5.2.11	程序复位 (Program Reset, Ctrl-F2)	65
5.3	执行历史窗口	66
5.3.1	指令区	67
5.3.2	击键记录区	67
5.4	中断程序执行	69
5.4.1	Ctrl-Break	69
5.5	程序终止	69
5.6	重新开始调试	69
5.6.1	重新加载用户程序	69
5.6.2	击键宏记录和重用	70
5.7	打开新的待调试程序	70
5.8	改变程序参数	71

第六章 检查和修改数据	72
6.1 数据菜单(Data Menu)	72
6.1.1 检查(Inspect...)	73
6.1.2 求值/修改(Evaluate/Modify...)	73
6.1.3 增加监视项(Add Watch...)	75
6.1.4 函数返回值	75
6.2 指向源文件的数据目标	75
6.3 监视窗口(Watch Window)	75
6.3.1 监视窗口的局部菜单	75
6.4 检查窗口	77
6.4.1 C 数据检查窗口	77
6.4.2 Pascal 数据检查窗口	80
6.4.3 汇编语言数据检查窗口	84
6.5 检查窗口的局部菜单	87
6.5.1 范围(Range...)	87
6.5.2 改变(Change...)	87
6.5.3 检查(Inspect...)	88
6.5.4 继承(Descend...)	88
6.5.5 新表达式(New Expression...)	88
6.5.6 强制类型转换(Type Cast...)	88
第七章 断点	89
7.1 断点菜单(The Breakpoints menu)	90
7.1.1 设置或清除断点(Toggle)	90
7.1.2 在...(At...)	90
7.1.3 全局存储变化(Changed memory global)	90
7.1.4 全局表达式为真(Expression True Global)	90
7.1.5 硬件断点(Hardware Breakpoint)	90
7.1.6 全清>Delete All)	91
7.2 断点窗口	91
7.2.1 断点窗口的局部菜单	91
7.2.2 断点选项(Breakpoint Options)对话框	94
7.2.3 条件和行为对话框(Conditions and Actions)	96
7.3 设置断点	98
7.3.1 简单断点	98
7.3.2 全局断点	99
7.3.3 改变内存的断点	99
7.3.4 条件表达式	99
7.3.5 硬件断点	100
7.3.6 记载变量值	100
7.3.7 断点和模板	100

7.4	记载窗口(Log Window)	101
7.4.1	记载窗口局部菜单	102
第八章	审阅和修改文件	103
8.1	审阅源程序文件	103
8.1.1	模块窗口	103
8.1.2	模块窗口的局部菜单	104
8.2	审阅其它磁盘文件	105
8.2.1	文件窗口	105
8.2.2	文件窗口的局部菜单	106
第九章	表达式	108
9.1	为计算表达式选择语言	108
9.2	代码地址、数据地址和行号	109
9.3	存取当前作用域以外的符号	109
9.3.1	作用域重设句法	109
9.3.2	表达式的隐含范围	112
9.4	字节列表(Byte lists)	113
9.5	C表达式	113
9.5.1	C符号	113
9.5.2	寄存器伪变量	113
9.5.3	C常量和格式	114
9.5.4	转义序列	115
9.5.5	C运算符优先级	115
9.5.6	在程序中执行C函数	116
9.5.7	带副作用的C表达式	116
9.5.8	C关键字和类型转换	117
9.6	Pascal表达式	117
9.6.1	Pascal符号	117
9.6.2	Pascal常量和格式	117
9.6.3	Pascal字符串	118
9.6.4	Pascal运算符和优先级	118
9.6.5	调用Pascal函数和过程	118
9.7	汇编表达式	118
9.7.1	汇编符号	118
9.7.2	汇编常量	119
9.7.3	汇编操作码	119
9.8	格式控制	119
第十章	调试C++和面向对象的PASCAL程序	121
10.1	层次窗口(The Hierarchy Window)	121
10.1.1	对象类型列表(Object Type List)区	121
10.1.2	层次树区	122

10.2	对象类型/类检查窗口	123
10.2.1	对象类型/类检查窗口的局部菜单	124
10.3	对象设例检查窗口	125
10.3.1	对象/类设例检查窗口的局部菜单	126
10.3.2	中区和底区	128
第十一章	汇编级调试	129
11.1	当源程序级调试不够时	129
11.2	CPU 窗口	129
11.3	代码区(Code Pane)	130
11.3.1	反汇编	131
11.3.2	代码区局部菜单	131
11.4	寄存器区和标志位区	134
11.4.1	寄存器区的局部菜单	134
11.4.2	标志区局部菜单	136
11.5	选择符区(Selector pane)	136
11.5.1	选择符区局部菜单	136
11.6	数据区	137
11.6.1	数据区的局部菜单	137
11.7	堆栈区	141
11.7.1	堆栈区的局部菜单	141
11.8	汇编	142
11.8.1	操作数地址长度重置	142
11.8.2	操作数数据长度重置	143
11.8.3	串指令	143
11.9	转储窗口	144
11.10	寄存器窗口	144
第十二章	80x87 协处理器芯片和仿真器	145
12.1	80x87 芯片和仿真器	145
12.2	数学处理器窗口(Numeric Processor)	145
12.2.1	寄存器区	146
12.2.2	控制区	147
12.2.3	状态区	147
第十三章	命令一览表	150
13.1	热键(Hot Keys)	150
13.2	主菜单的命令	152
13.2.1	≡(系统)菜单(≡ System menu)	152
13.2.2	文件菜单(File menu)	152
13.2.3	浏览菜单(View menu)	152
13.2.4	运行菜单(Run menu)	153
13.2.5	断点菜单(Breakpoints menu)	153

13.2.6	数据菜单(Data menu)	153
13.2.7	选项菜单(Options menu)	153
13.2.8	窗口菜单(Window menu)	153
13.2.9	帮助菜单(Help menu)	154
13.3	局部菜单命令	154
13.3.1	断点窗口(Breakpoints Window)	154
13.3.2	CPU 窗口的菜单(CPU Window menus)	154
13.3.3	转贮窗口(Dump Window)	156
13.3.4	文件窗口(File Window)	156
13.3.5	记载窗口菜单(Log Window Menu)	157
13.3.6	模块窗口(Module Window)	157
13.3.7	数学处理器窗口(Numeric Processor Window)	158
13.3.8	层次窗口(Hierarchy Window)	158
13.3.9	寄存器窗口菜单(Register Window Menu)	158
13.3.10	堆栈窗口(Stack Window)	159
13.3.11	变量窗口(Variables Window)	159
13.3.12	监视窗口(Watches Window)	159
13.3.13	检查窗口(Inspector Window)	159
13.3.14	对象类型/类检查窗口(Object Type/Class Inspector Window)	160
13.3.15	对象/类设例检查窗口(Object/Class Instance Inspector Window)	160
13.4	文本区	160
13.5	列表区	161
13.6	输入框和历史列表框中的命令	162
13.7	窗口移动命令	162
13.8	通配符搜索模板	163
13.9	完整的菜单树	163
第十四章	调试技巧	165
14.1	当出现问题时	165
14.2	调试风格	165
14.2.1	运行整个程序	165
14.2.2	增量测量	166
14.3	错误类型	166
14.3.1	一般性错误	166
14.3.2	C 特有的错误	167
14.3.3	Pascal 特有的错误	170
14.3.4	汇编特有的错误	175
14.4	精确测试	178
14.4.1	测试边界条件	178
14.4.2	无效的数据输入	178
14.4.3	空数据输入	178

14.5	调试——程序设计的一部分	178
14.6	样例调试过程	178
14.7	C 调试过程	179
14.7.1	寻找错误	179
14.7.2	确定解决方案	179
14.7.3	启动 Turbo Debugger	179
14.7.4	检查	180
14.7.5	断点	180
14.7.6	监视窗口	180
14.7.7	求值/修改对话框	180
14.7.8	Eureka!	181
14.8	Pascal 调试过程	181
14.8.1	寻找错误	181
14.8.2	确定解决方案	182
14.8.3	启动 Turbo Debugger	183
14.8.4	巡查程序	183
14.8.5	求值/修改对话框	183
14.8.6	检查	183
14.8.7	监视 (Watches)	184
14.8.8	最后一个错误	185
第十五章	80386 处理器上的虚拟调试	186
15.1	虚拟调试所需设备	186
15.2	安装虚拟调试设备驱动程序	186
15.3	开始虚拟调试	186
15.4	一般调试和虚拟调试的区别	188
15.5	通信	188
15.6	TD386 错误信息	189
15.7	TDH386.SYS 错误信息	190
第十六章	用 TD286 在保护模式下调试	191
16.1	保护模式调试器的硬件要求	191
16.2	安装保护模式调试器	191
16.3	启动保护模式调试器	191
16.4	保护模式的调试器与 Turbo Debugger 的不同点	191
16.5	在不同的机器上运行 TD286	192
第十七章	面向 Windows 的 Turbo Debugger (TDW)	193
17.1	运行 TDW 的要求	193
17.2	安装 TDW	193
17.3	配置 TDW	194
17.3.1	使用 TDW 选择项	194
17.3.2	用 TDINST 设置 TDW	194

17.4 使用 TDW	195
17.4.1 记载窗口消息	196
17.4.2 获得内存和模块列表	203
17.4.3 调试动态连接库(DLL)	205
17.4.4 将内存句柄转变为地址	209
17.4.5 调试建议	209
17.5 TDW 错误信息	210
第十八章 调试一个标准的 Windows 应用程序	211
18.1 样例程序	211
18.2 编译和连接演示程序	211
18.3 调试 BCWDEMOA	212
18.3.1 决策	212
18.3.2 中止 BCDEMOA	212
18.3.3 记录消息	212
18.3.4 分析消息记录	213
18.3.5 发现错误	213
18.3.6 终止 BCWDEMOA	216
18.4 调试 BCWDEMOB	217
18.4.1 切换程序	217
18.4.2 测试程序	217
18.4.3 如何处理	217
18.4.4 比较全局内存列表	218
18.4.5 发现错误的卓有成效的方法	218
第十九章 调试 ObjectWindows 应用程序	221
19.1 关于程序	221
19.1.1 Color Scribble 窗口类型定义	222
19.1.2 创建应用程序	223
19.2 调试程序	223
19.2.1 找出第一个错误	223
19.2.2 找出笔颜色方面的错误	225
19.2.3 找出离开窗口时的错误	227
19.2.4 找出清除屏幕错误	229
第二十章 调试 TSRs 和设备驱动程序	231
20.1 什么是 TSR	231
20.1.1 调试 TSR	231
20.2 什么是设备驱动程序	233
20.2.1 调试设备驱动程序	234
20.3 结束调试过程	235
附录 A 命令行选择项	236

附录 B 技术资料	237
B.1 变化了的加载地址和内存自由空间	237
B.2 系统崩溃	237
B.3 跟踪 DOS 和进程 ID 切换	237
B.4 使用 8027/80287 数学协处理器和仿真器	238
B.5 Turbo Debugger 使用的中断	238
B.6 用 INT 3 和 INT 1 调试	238
B.7 显示保存和模式切换	238
B.8 存储分配	239
B.9 EMS 支持	239
B.10 中断向量保存的恢复	239
附录 C 嵌入汇编关键字	240
附录 D 设置 Turbo Debugger 参数	243
D.1 运行 TDINST	243
D.1.1 TDINST 命令行选项	243
D.2 设置屏幕颜色	244
D.2.1 设置屏幕颜色 (Customizing screen colors)	244
D.2.2 缺省颜色 (The default Colors)	246
D.3 设置 Turbo Debugger 的显示参数	248
D.3.1 显示切换 (Display Swapping)	248
D.3.2 整数格式 (Integer Format)	249
D.3.3 初始显示 (Beginning Display)	249
D.3.4 屏幕行数 (Screen Lines)	249
D.3.5 制表长度 (Tab Size)	249
D.3.6 最多的监视项 (Maximum Tiled Watch)	249
D.3.7 快速屏幕更新 (Fast Screen Update)	249
D.3.8 允许 43/50 行显示 (Permit 43/50 Lines)	249
D.3.9 完整图形保存 (Full Graphics Saving)	250
D.3.10 用户屏幕更新 (User Screen Updating)	250
D.3.11 记载表长度	250
D.4 Turbo Debugger 的选项 (Turbo Debugger Options)	250
D.4.1 目录 (Directories ...)	250
D.4.2 输入与提示 (Input and Prompting ...)	250
D.4.3 源 (语言) 调试 (Source Debugging ...)	252
D.4.4 杂类选项 (Miscellaneous Options ...)	253
D.5 设置显示方式	255
D.6 命令行选择项及其对应的安装设置	256
D.7 完成设置后	257
D.7.1 保存修改	257
D.7.2 退出 TDINST	258

附录 E 远程调试	259
E.1 硬件和软件要求	259
E.2 调试远程 DOS 应用程序	260
E.2.1 建立远程系统	260
E.2.2 配置 TDREMOTE	260
E.2.3 远程 DOS 驱动程序	261
E.2.4 建立远程 DOS 连接	262
E.3 调试远程 Windows 应用程序	262
E.3.1 建立远程系统	262
E.3.2 配置 WREMOTE	263
E.3.3 启动远程 Windows 驱动程序	264
E.3.4 建立远程 Windows 链路	264
E.4 加载程序到远程系统	265
E.5 远程调试过程	265
E.6 调试建议	266
E.6.1 TDREMOTE 信息	266
E.6.2 WREMOTE 消息	267
附录 F 对话框与错误信息	268
F.1 提示框	268
F.2 错误信息	272
F.2.1 致命错误	272
F.2.2 其它错误信息	273
F.3 其它信息	283
附录 G 在不同语言下使用 Turbo Debugger	284
G.1 Borland C++ 的提示	284
G.1.1 作优化处理	284
G.1.2 存取指针数据	281
G.1.3 单步执行复杂表达式	284
G.2 Turbo 汇编提示	285
G.2.1 查看原始十六进制数	285
G.2.2 源程序级调试	285
G.2.3 检查和修改寄存器值	285
G.3 Turbo Pascal 的提示	285
G.3.1 单步执行初始化代码	285
G.3.2 单步执行出口过程	286
G.3.3 常量	286
G.3.4 字符串和集合在堆栈上的临时空间	286
G.3.5 灵活的类型强制转换(Type Casting)	286
G.3.6 Pascal 的 CPU 窗口注意点	287