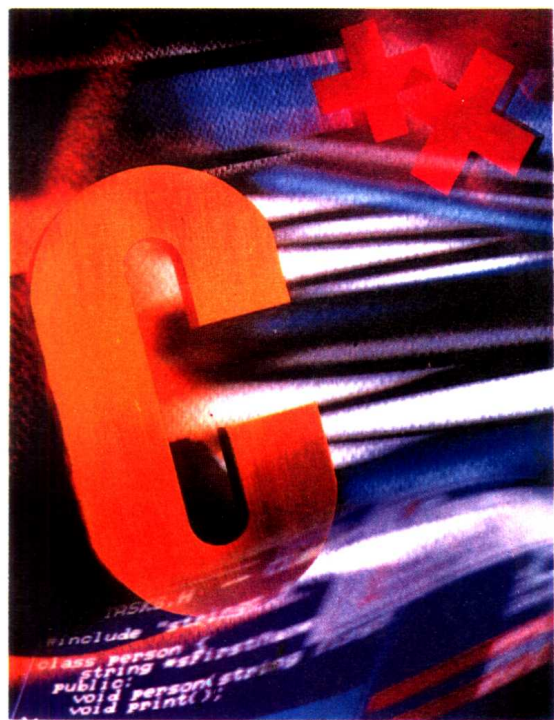


万能程序调试器

TURBO Debugger 进阶教程



海洋出版社

梁帆 郭平平 编
张鹏 刘宏程

北京希望电脑公司计算机技术丛书

万能程序调试器 Turbo Debugger 进阶教程

梁 帆 郭平平 编
张 鹏 刘宏程

海洋出版社

1993·北京

内 容 简 介

本书主要介绍如何使用 Turbo Debugger 调试 MS-DOS 环境中的 C、Pascal 和汇编语言程序,如何将比较好的调试方法应用于实际调试中。全书共四部分,共二十二章,内容涉及参考指南、调试艺术、高级调试技巧及数据结构指南。内容丰富、实用,可供程序员学习 TD 各种命令和功能、学会调试技巧时参考、使用。

需要本书的读者,请直接与北京 8721 信箱联系,邮码 100080,电话 2562329。

(京)新登字 087 号

北京希望电脑公司计算机技术丛书

万能程序调试器

Turbo Debugger 进阶教程

梁帆 郭平平 编
张鹏 刘宏程

*

海洋出版社出版(北京市复兴门外大街1号)

海洋出版社发行 兰空印刷厂印刷

开本:787×1092 1/16 印张:26.4 字数:646千字

1993年10月第一版 1993年10月第一次印刷

印数:1—5000册

*

ISBN 7-5027-3885-7/TP·240 定价:39元

目 录

第一部分 参考指南

前言

第一章 简介	(2)
1.1 软硬件配置	(2)
1.1.1 硬件需求.....	(2)
1.1.2 软件需求.....	(3)
1.1.3 知识需求.....	(3)
1.1.4 可选硬件.....	(3)
1.2 如何使用本书	(3)
1.3 各章节简介	(4)
1.3.1 第一部分:参考指南	(4)
1.3.2 第二部分:调试的艺术	(4)
1.3.3 第三部分:高级调试技巧	(5)
1.3.4 第四部分:数据结构指南	(5)
1.4 源程序列表	(5)
1.5 键盘上的键	(5)
1.6 字体和字型	(6)
1.7 文件名	(6)
1.8 阅读顺序	(6)
小结.....	(6)
第二章 准备程序进行调试	(7)
2.1 如何使用本章	(7)
2.2 设计-编译-调试	(8)
2.3 编译器的升级	(9)
2.4 用 TLIB 管理目标库	(9)
2.5 编译 C 和 C++ 程序并进行调试	(9)
2.5.1 C 语言程序范例.....	(9)
2.5.2 Lattice C 3.3	(10)
2.5.3 Microsoft C 5.1	(11)
2.5.4 Microsoft C 6.0	(11)
2.5.5 Quick C 2.01	(11)
2.5.6 Turbo C 2.0 集成开发环境	(12)
2.5.7 Turbo C 2.0 命令行编译器	(12)
2.5.8 Turbo C++ 1.0 程序员平台	(13)
2.5.9 Turbo C++ 1.0 命令行编译器	(14)
2.5.10 Zortech C++ 1.07	(14)

2.5.11	Zortech C++ 2.0	(15)
2.6	编译 Pascal 程序并进行调试	(15)
2.6.1	Pascal 程序举例	(15)
2.6.2	Microsoft Pascal 4.0	(15)
2.6.3	QuickPascal 1.0	(17)
2.6.4	Turbo Pascal 4.0	(17)
2.6.5	Turbo Pascal 5.x 集成开发环境	(18)
2.6.6	Turbo Pascal 5.x 命令行编译器	(18)
2.7	准备汇编语言程序进行调试	(19)
2.7.1	汇编语言程序举例	(19)
2.7.2	Microsoft Macro Assembler 5.1	(20)
2.7.3	OptASM 1.5	(20)
2.7.4	QuickAssembler 2.01	(21)
2.7.5	Turbo Assembler 2.0	(21)
2.8	准备 .COM 程序进行调试	(22)
2.8.1	.COM 示范程序	(22)
2.8.2	汇编 .COM 程序	(22)
2.9	编译其它程序并进行调试	(23)
2.10	无源代码时的调试	(24)
2.11	用 TOUCH 修改文件日期	(24)
小结		(24)
第三章	建立并运行 Turbo Debugger	(26)
3.1	配置经验谈	(26)
3.1.1	黑白图形	(26)
3.1.2	建立目录	(26)
3.1.3	系统 RAM	(27)
3.1.4	扩充 RAM(EMS)	(27)
3.1.5	扩展 RAM(XMS)	(28)
3.1.6	将扩展 RAM 转换为扩充 RAM	(29)
3.1.7	基于 80286 系统的配置	(29)
3.1.8	基于 80386 系统的配置	(30)
3.1.9	异常	(31)
3.1.10	为环境变量保留空间	(31)
3.1.11	远程和双显示器模式的配置	(31)
3.2	安装鼠标	(31)
3.2.1	在 Microsoft Windows 中使用鼠标	(32)
3.3	最小配置	(32)
3.4	自己定义设置	(33)
3.4.1	编辑配置文件	(33)
3.4.2	恢复原始设置	(34)

3.5	TDINST 实用程序的命令	(34)
3.5.1	颜色	(34)
3.5.2	显示	(34)
3.5.3	选择项	(36)
3.5.4	显示模式	(38)
3.5.5	保存	(38)
3.5.6	退出	(38)
3.6	创建调试工作站	(38)
3.6.1	运行编辑器和其它程序	(38)
3.6.2	切换到 DOS	(39)
3.6.3	安装语言帮助信息	(39)
3.6.4	Microsoft Windows	(40)
3.7	运行 TD	(40)
3.8	TD 的命令行选择项	(40)
3.8.1	-c[file]	(41)
3.8.2	-do, -dp, -ds	(41)
3.8.3	-h, -?	(41)
3.8.4	-i[+ -]	(41)
3.8.5	-k[+ -]	(42)
3.8.6	-l	(42)
3.8.7	-m<#>	(42)
3.8.8	-p[+ -]	(42)
3.8.9	-r[+ -]	(42)
3.8.10	-rp<#>	(42)
3.8.11	-rs<#>	(42)
3.8.12	-sc[+ -]	(43)
3.8.13	-sd<dir>	(43)
3.8.14	-sm<#>	(43)
3.8.15	-vg[+ -]	(43)
3.8.16	-vn[+ -]	(43)
3.8.17	-vp[+ -]	(43)
3.8.18	-y<#>	(43)
3.8.19	-ye<#>	(44)
3.9	TD286 命令行选择项	(44)
3.10	TD386 命令行选择项	(44)
3.10.1	-b[+ -]	(44)
3.10.2	-e<#>	(44)
3.10.3	-fx000	(44)
3.10.4	-w	(45)
小结		(45)

第四章 窗口、菜单和热键	(46)
4.1 程序范例	(46)
4.2 作用域	(46)
4.3 Turbo debugger 的显示	(47)
4.4 窗口	(48)
4.5 菜单	(48)
4.6 观察窗	(51)
4.6.1 窗口中有什么?	(51)
4.6.2 鼠标窗口命令	(52)
4.6.3 使用滚标	(53)
4.6.4 窗口分区	(53)
4.7 键盘窗口命令	(54)
4.7.1 移动窗口	(54)
4.7.2 上下文敏感性	(55)
4.8 对话框	(55)
4.8.1 对话框内有什么?	(55)
4.8.2 关闭对话框	(57)
4.8.3 用鼠标项	(57)
4.8.4 用键盘设置选择项	(57)
4.8.5 输入正文	(58)
4.8.6 历史列表	(58)
4.8.7 信息对话框	(59)
4.8.8 提示对话框	(59)
4.9 检查窗	(60)
4.9.1 局部检查菜单	(61)
4.9.2 检查对象	(63)
4.9.3 对象检查菜单	(63)
4.9.4 检查对象类型	(64)
4.10 全局菜单	(64)
4.10.1 系统菜单(≡)	(65)
4.10.2 文件菜单(File)	(65)
4.10.3 观察菜单(View)	(68)
4.10.4 运行菜单(Run)	(68)
4.10.5 断点菜单(Breakpoints)	(71)
4.10.6 数据菜单	(72)
4.10.7 选择项菜单(Options Menu)	(74)
4.10.8 Window 菜单	(77)
4.10.9 Help 菜单	(79)
4.11 热键	(81)
小结	(82)

第五章 观察(View)菜单及其命令	(83)
5.1 如何使用本章	(84)
5.2 缺省命令	(84)
5.3 Another 观察窗	(84)
5.4 断点(Breakpoints)观察窗	(84)
5.4.1 断点观察窗局部命令	(85)
5.5 CPU 观察窗	(88)
5.5.1 CPU 窗口的使用	(88)
5.5.2 打开 CPU 窗口	(89)
5.5.3 CPU 代码区	(89)
5.5.4 CPU 代码区局部命令	(90)
5.5.5 CPU 的寄存器区	(93)
5.5.6 CPU 的标志区	(93)
5.5.7 CPU 的堆栈区	(93)
5.5.8 CPU 堆栈区的局部命令	(93)
5.5.9 CPU 的信息转储区	(95)
5.6 信息转储观察窗	(95)
5.6.1 滚屏信息转储窗口	(95)
5.6.2 信息转储观察窗局部菜单	(95)
5.7 执行史(Execution History)观察窗	(100)
5.7.1 反向执行代码	(100)
5.7.2 对反向执行命令的限制	(101)
5.7.3 执行史局部菜单	(101)
5.8 文件观察窗(File View)	(101)
5.8.1 文件观察窗局部菜单	(102)
5.9 谱系(Hierarchy)观察窗	(103)
5.9.1 C++与 Turbo Pascal 的对比	(104)
5.9.2 谱系观察窗局部菜单	(104)
5.10 Log(记录)观察窗	(105)
5.10.1 记录观察窗局部菜单	(106)
5.11 Module(模块)观察窗(<F3>)	(107)
5.11.1 模块观察窗标题	(107)
5.11.2 模块观察窗局部菜单	(108)
5.12 数学处理器观察窗(Numeric Processor View)	(111)
5.12.1 NDP 的堆栈值	(111)
5.12.2 数学处理器观察窗的窗口标题	(111)
5.12.3 数学处理器观察窗局部菜单	(112)
5.13 寄存器(Register)观察窗	(113)
5.13.1 寄存器观察窗局部菜单	(113)
5.14 堆栈(Stack)观察窗	(114)

5.14.1	堆栈观察窗局部菜单.....	(115)
5.15	变量(Variables)观察窗	(115)
5.15.1	变量观察窗局部菜单.....	(116)
5.16	监视(Watches)观察窗	(117)
5.16.1	观察变量.....	(117)
5.16.2	向监视窗中加入标号.....	(117)
5.16.3	监视观察窗局部菜单.....	(118)
小结		(119)
第六章	TD 的实用程序	(120)
6.1	显示联机帮助信息	(120)
6.2	错误信息	(120)
6.3	语法结构的描述	(120)
6.4	TDCONVRT.EXE	(121)
6.5	TDCONVRT 的语法结构及选择项	(121)
6.5.1	-c	(121)
6.5.2	-sw	(121)
6.6	TDDEV.EXE	(122)
6.7	TDDEV 的语法结构及选择项	(122)
6.7.1	-r	(122)
6.8	TDINST.EXE	(122)
6.9	TDINST 的语法结构及选择项	(122)
6.9.1	-c<file>	(122)
6.10	TDMAP.EXE	(122)
6.11	TDMAP 的语法结构及选择项.....	(123)
6.11.1	-b	(123)
6.11.2	-c	(123)
6.11.3	-e<ext>	(123)
6.11.4	-q	(124)
6.12	TDMEM.EXE	(124)
6.13	TDMEM 的语法结构及选择项	(124)
6.13.1	-v	(124)
6.14	TDNMI.COM	(124)
6.15	TDNMI 的语法结构及选择项	(124)
6.15.1	-p[<#>]	(124)
6.16	TDPACK.EXE	(124)
6.17	TDPACK 的语法结构	(125)
6.18	TDREMOTE.EXE	(125)
6.19	TDRF.EXE	(125)
6.20	TDSTRIP.EXE	(125)
6.21	TDSTRIP 的语法结构及选择项	(125)

6.21.1	-s	(126)
6.21.2	-c	(126)
6.22	TDSTRIP 用法举例	(126)
6.23	TDUMP.EXE	(127)
6.24	TDUMP 的语法结构及选择项	(127)
6.24.1	-a	(127)
6.24.2	-a7	(127)
6.24.3	-b<#>	(127)
6.24.4	-e	(127)
6.24.5	-el	(127)
6.24.6	-er	(127)
6.24.7	-elr	(128)
6.24.8	-h	(128)
6.24.9	-l	(128)
6.24.10	-m	(128)
6.24.11	-o	(128)
6.24.12	-oc	(128)
6.24.13	-oiID	(128)
6.24.14	-oxID	(129)
6.24.15	-v	(129)
6.25	UNZIP.EXE	(129)
6.26	UNZIP 的语法结构及选择项	(129)
6.26.1	-c, -cm	(130)
6.26.2	-o	(130)
6.26.3	-p	(130)
6.26.4	-t	(130)
6.26.5	-v[b c d e n p s r]	(139)
小结		(130)

第二部分 调试的艺术

第七章	制定一种调试策略	(131)
7.1	调试风格的基础	(132)
7.2	Turbo Debugger 的工具	(132)
7.2.1	断点	(132)
7.2.2	代码跟踪	(133)
7.2.3	数据检查	(133)
7.2.4	表达式求值	(134)
7.3	错误分类	(135)
7.3.1	语法错误(syntax errors)	(135)
7.3.2	运行时序错误(runtime errors)	(135)

7.3.3	逻辑错误	(136)
7.4	运行时序错误和逻辑错误的类型	(136)
7.4.1	依赖数据的错误 data-dependent bugs	(136)
7.4.2	间歇性错误 intermittent	(137)
7.4.3	动靶错误 moving-target bugs	(137)
7.4.4	致命错误 fatal bugs	(137)
7.4.5	远距离错误 long-distance bugs	(138)
7.4.6	定时性发生错误 time-bomb	(138)
7.5	调试策略	(138)
7.5.1	测试错误	(139)
7.5.2	确定错误	(139)
7.5.3	分离错误	(140)
7.5.4	修改和再测试	(140)
	小结	(140)
第八章	断点和代码跟踪	(142)
8.1	断点、跟踪点和监视点	(142)
8.2	用代码断点进行调试	(142)
8.2.1	分离错误	(142)
8.2.2	在过程和函数中中断	(143)
8.2.3	检查程序退出条件	(143)
8.2.4	找出运行时序错误	(143)
8.2.5	在 OOP 方法中中断	(143)
8.2.6	代码断点技巧	(144)
8.3	用数据断点调试	(144)
8.3.1	输入表达式	(145)
8.3.2	硬件和软件的不同	(145)
8.3.3	加速软件数据断点	(145)
8.3.4	字符串比较	(146)
8.3.5	依靠寄存器值中断	(146)
8.3.6	数据断点技巧	(146)
8.4	记录表达式	(147)
8.4.1	记录多重变量	(147)
8.4.2	记录复杂表达式	(148)
8.4.3	在 OOP 代码中记录“self”	(148)
8.4.4	副作用	(148)
8.5	拼接代码	(148)
8.5.1	拼接 pascal 代码	(149)
8.5.2	拼接 C 代码	(149)
8.5.3	拼接过程和函数调用	(150)
8.6	设置通过计数(pass count)	(150)

8.6.1	校验一个循环索引	(150)
8.6.2	查找未被认可的变量赋值	(150)
8.6.3	找出不必要的递归	(151)
8.7	用代码跟踪调试	(151)
8.7.1	跟踪和单步	(151)
8.7.2	使用指令跟踪	(152)
8.7.3	激活运行	(152)
8.7.4	用反向跟踪调试 back tracing	(153)
8.7.5	反向跟踪机器代码	(153)
8.7.6	反向跟踪的限制	(153)
8.8	跟踪进入 DOS 和 BIOS 代码	(153)
小结		(154)
第九章	表达式求值	(155)
9.1	语言和形式	(155)
9.2	改变表达式的形式	(155)
9.3	行号	(156)
9.4	C 语言表达式	(157)
9.4.1	操作符	(157)
9.4.2	算术表达式	(157)
9.4.3	字符串表达式	(157)
9.4.4	类型转换	(158)
9.4.5	副作用	(158)
9.5	Pascal 表达式	(158)
9.5.1	操作符	(158)
9.5.2	算术表达式	(159)
9.5.3	字符串表达式	(159)
9.5.4	调用字符串函数	(159)
9.5.5	类型转换	(160)
9.5.6	副作用	(160)
9.6	汇编语言表达式	(160)
9.6.1	操作符	(161)
9.6.2	算术表达式	(161)
9.6.3	字符串表达式	(161)
9.6.4	副作用	(162)
9.7	面向目标的表达式	(162)
9.8	调用对象的方法	(162)
小结		(163)
第十章	C 语言常见错误	(164)
10.1	找出错误的根源	(164)
10.1.1	移动了注释符	(164)

10.1.2	大括号和小括号的不匹配	(165)
10.1.3	else 与 if 错误匹配	(165)
10.1.4	混淆 = 和 == 运算符	(166)
10.1.5	路径名故障	(166)
10.1.6	分号位置错误	(167)
10.1.7	意外的函数重定义	(168)
10.2	由变量引起的问题	(168)
10.2.1	未初始化的变量	(168)
10.2.2	查找未初始化变量	(169)
10.2.3	稳定一个变化的变量	(169)
10.2.4	全局变量误操作	(170)
10.2.5	混淆静态变量和自动变量	(170)
10.2.6	混淆静态变量和外部变量	(171)
10.3	数组	(172)
10.3.1	毫无意义的值	(172)
10.3.2	下标范围错	(172)
10.4	由指针引起的问题	(173)
10.4.1	未初始化的指针	(173)
10.4.2	查找未初始化的指针	(173)
10.4.3	查找空指针	(174)
10.4.4	指针未分配空间	(174)
10.4.5	指针计算错误	(174)
10.4.6	指针和自动变量	(175)
10.4.7	未释放分配的空间	(175)
10.4.8	使用已释放了的内存	(176)
10.5	函数	(176)
10.5.1	漏掉了返回值	(176)
10.5.2	混淆了调用值和调用地址	(177)
10.5.3	函数副作用	(177)
10.5.4	意外的递归	(178)
10.6	数值错误	(179)
10.6.1	操作符优先级使用错误	(179)
10.6.2	十六进制方式	(179)
10.7	文件处理	(179)
10.7.1	忘记关闭已打开的文件	(180)
10.7.2	没有检查 I/O 错误	(180)
10.7.3	在 Scanf() 中忽视了使用指针	(180)
10.8	Break 语句的错误	(180)
10.8.1	嵌套的 Break	(181)
10.8.2	Break 与 Continue 语句	(181)

10.8.3 开关语句(switch)中忘记用 break	(182)
小结.....	(182)
第十一章 C 语言调试实例	(183)
11.1 调试策略的回顾.....	(183)
11.2 程序.....	(183)
11.3 调试步骤说明.....	(192)
11.3.1 使用 LS.C	(192)
11.3.2 LS.C 的工作情况	(192)
11.4 错误 1	(193)
11.4.1 错误 1——测试	(193)
11.4.2 错误 1——确定	(193)
11.4.3 错误 1——分离	(194)
11.4.4 错误 1——修改	(195)
11.5 错误 2	(195)
11.5.1 错误 2——测试	(196)
11.5.2 错误 2——确定	(196)
11.5.3 错误 2——分离	(196)
11.5.4 错误 2——修改	(197)
11.6 错误 3	(197)
11.6.1 错误 3——测试	(198)
11.6.2 错误 3——确定	(198)
11.6.3 错误 3——分离	(198)
11.6.4 错误 3——修改	(199)
11.7 错误 4	(200)
11.7.1 错误 4——测试	(200)
11.7.2 错误 4——确定	(201)
11.7.3 错误 4——分离	(201)
11.7.4 错误 4——修改	(203)
11.8 错误 5	(203)
11.8.1 错误 5——测试	(204)
11.8.2 错误 5——确定	(204)
11.8.3 错误 5——分离	(204)
11.8.4 错误 5——修改	(205)
小结.....	(205)
第十二章 Pascal 常见错误	(206)
12.1 录入错误.....	(206)
12.1.1 注释中括号不匹配.....	(206)
12.1.2 ELSE 与 IF—THEN 出错	(207)
12.1.3 消失了的标过程.....	(208)
12.2 变量的困惑.....	(210)

12.3	全程变量之争.....	(211)
12.4	越界错误.....	(211)
12.4.1	如何查出运行时序错误.....	(211)
12.4.2	循环次数过多.....	(212)
12.5	过程的危机.....	(213)
12.5.1	混淆变参与值参.....	(213)
12.5.2	字符串长度问题.....	(213)
12.6	函数中的错误.....	(214)
12.6.1	副作用.....	(214)
12.6.2	函数退出的途径.....	(215)
12.6.3	多余的递归.....	(216)
12.6.4	间接递归调用.....	(217)
12.6.5	交互式副作用.....	(218)
12.7	指针.....	(218)
12.7.1	未初始化指针.....	(218)
12.7.2	找出 nil 指针.....	(219)
12.7.3	找出未初始化的指针.....	(219)
12.7.4	释放指针.....	(220)
12.7.5	非规范化指针.....	(220)
12.7.6	对 MemAvail 和 MaxAvail 的误解.....	(221)
12.7.7	内存不够.....	(222)
12.8	数字之迷.....	(222)
12.8.1	运算符优先级错位.....	(223)
12.8.2	负的字型值.....	(223)
12.8.3	加上十六进标志符.....	(224)
12.8.4	整数的环绕.....	(224)
12.9	文件操作错误.....	(224)
12.9.1	忘记关闭文件.....	(225)
12.9.2	延迟发生的文件错误.....	(225)
12.10	覆盖中存在的问题.....	(227)
12.10.1	初始化多次.....	(227)
12.10.2	缓慢的覆盖.....	(227)
	小结.....	(228)
第十三章	Pascal 调试实例.....	(229)
13.1	程序.....	(229)
13.2	调试实例.....	(237)
13.3	使用 CAL.PAS.....	(237)
13.4	错误 1.....	(238)
13.4.1	错误 1——测试.....	(238)
13.4.2	错误 1——确定.....	(239)

13.4.3	错误 1——隔离	(239)
13.4.4	错误 1——改正	(240)
13.5	错误 2	(241)
13.5.1	错误 2——测试	(241)
13.5.2	错误 2——确定	(242)
13.5.3	错误 2——隔离	(242)
13.5.4	错误 2——改正	(244)
13.6	错误 3	(244)
13.6.1	错误 3——测试	(244)
13.6.2	错误 3——确定	(244)
13.6.3	错误 3——隔离	(245)
13.6.4	错误 3——改正	(245)
13.7	错误 4	(246)
13.7.1	错误 4——测试	(247)
13.7.2	错误 4——确定	(247)
13.7.3	错误 4——隔离	(247)
13.7.4	错误 4——改正	(248)
13.8	错误 5	(248)
13.8.1	错误 5——测试	(248)
13.8.2	错误 5——确定	(248)
13.8.3	错误 5——隔离	(249)
13.8.4	错误 5——改正	(249)
13.9	错误 6	(250)
13.9.1	错误 6——测试	(250)
13.9.2	错误 6——确定	(250)
13.9.3	错误 6——隔离	(250)
13.9.4	错误 6——改正	(251)
小结		(251)
第十四章	汇编语言常见错误	(252)
14.1	录入错误	(252)
14.1.1	指令操作数顺序	(252)
14.1.2	寄存器出栈错误	(252)
14.1.3	混淆偏移量和变量	253()
14.2	程序常见错误	(254)
14.2.1	未返回 DOS	(254)
14.2.2	未设置堆栈或堆栈太小	(255)
14.2.3	理解初始化数据	(255)
14.2.4	错误理解 ASSUME	(256)
14.2.5	意外地修改了寄存器	(256)
14.2.6	未声明的寄存器	(257)

14.3	标志位上的错误	(257)
14.4	段错误	(257)
14.4.1	使用了错误的段寄存器	(257)
14.4.2	使用错误的缺省段寄存器	(258)
14.4.3	忽略了数据段起始偏移	(258)
14.4.4	异常的段环绕	(259)
14.5	过程的危机	(259)
14.5.1	异常跳出	(259)
14.5.2	未初始化寄存器参数	(260)
14.5.3	多个入口点和出口点	(260)
14.5.4	近返回和远返回	(261)
14.5.5	未保存寄存器	(261)
14.6	跳转上的错误	(261)
14.6.1	跳转条件有错	(261)
14.6.2	局部标号位置有错	(262)
14.7	串指令错误	(263)
14.7.1	依靠为零的 CX 检查整个段	(263)
14.7.2	串操作数	(264)
14.7.3	错误的方向标志设置	(265)
14.7.4	混淆缺省段寄存器	(265)
14.8	中断中的错误	(265)
14.8.1	破坏了寄存器的值	(265)
14.8.2	禁止中断	(266)
14.8.3	未恢复中断向量	(266)
14.9	数字错误	(266)
14.9.1	未扩展符号位	(266)
14.9.2	进制错误	(266)
14.10	调试混合语言代码	(267)
	小结	(267)
第十五章	用 TD 调试汇编语言程序	(268)
15.1	程序	(268)
15.2	准备调试	(276)
15.3	使用 FILLMEM 和 FILLTEST	(277)
15.4	第一个程序错误	(278)
15.4.1	第一个程序错误——测试并确定	(278)
15.4.2	第一个程序错误——隔离	(279)
15.5	第二个程序错误	(280)
15.5.1	第二个程序错误——隔离	(280)
15.5.2	第二个程序错误——修改	(281)
15.6	第三个程序错误	(281)