

(新一代调试器)

Turbo Debugger 用户指南

叶 欣 编译

- Turbo族程序的源级调试
- 菜单导引、热键启动、联机帮助
- 双监视器、双机调试
- 单步跟踪、断点执行、宏定义

给您带来编程效益的飞跃

北京科海培训中心

一九八九年十月

目 录

前言	1
软件及硬件要求	1
术语解释	1
手册简介	1
第一章 启动	4
Turbo Debugger 软件包	4
程序盘	4
安装 Turbo Debugger	5
在两软盘驱动器中使用 Turbo Debugger	5
在硬盘中建立 Turbo Debugger	5
硬件调试	5
现在在哪儿	5
正在学习 Turbo 语言的程序员	
已经使用过 Turbo 语言的程序员	
第二章 调试和 Turbo 系列	
什么是调试?	7
Turbo Debugger 能做什么	8
Turbo Debugger 不能做什么	8
Turbo Debugger 是怎样工作的	9
Turbo Debugger 的优点	9
使用主菜单	9
知道现在在哪儿	10
局部菜单	10
历史机制	11
宏生成	11
窗口设置	12
求得帮助	16
第三章 一个快速启动的例子	18
样例程序	18
使用 Turbo Debugger	18
帮助行	19

窗口	19
用 C 样例程序	20
使用 Pascal 样例程序	24
第四章 开始用 Turbo Debugger	29
准备待调试程序	29
运行 Turbo Debugger	29
命令行选择	30
配置文件	31
选择项菜单	31
在 Turbo Debugger 中运行 DOS	32
返回到 DOS	33
第五章 控制程序执行	34
考察当前程序状态	34
变量窗口	34
全局菜单命令	35
局部菜单命令	36
局部菜单	37
局部菜单	37
局部菜单	37
局部菜单	37
观看执行状态: Get Info 命令	38
运行菜单	39
中断程序执行	41
程序运行完	41
重开一调试节	41
重装用户程序	42
键入记录和再调用	42
加载一新的待调试程序	42
改变程序参数	42
第六章 考察和修改数据	43
数据菜单	43
指向源文件的数据项	44
观察窗口	45
观察窗口局部菜单	45
检测窗口	46
C 数据检测窗口	46

Pascal 数据检测窗口	50
汇编语言数据检测窗口	50
检测窗口局部菜单	57
第七章 断点	59
断点菜单	59
断点表达式范围	60
断点窗口	60
断点窗口局部菜单	61
记载窗口	64
记载窗口局部菜单	65
简单断点	66
条件断点和通过计数	66
全局断点	66
根据数据目标变化而中断点	67
记载变量值	67
执行表达式	67
第八章 考察和修改文件	68
考察源程序文件	68
模块窗口	68
模块窗口局部菜单	69
考察其他磁盘文件	71
文件窗口	71
文件窗口局部菜单	71
第九章 表达式	74
选择表达式计算语言	74
代码地址、数据地址和行号	74
在当前范围外存取符号	74
范围超越句法	75
表达式的隐含范围	75
字节列表	76
C 表达式	76
Pascal 表达式	79
汇编表达式	81
格式控制	82

第十章 汇编级调试	83
当源程序级调试不够时	83
CPU 窗口	83
代码区	84
反汇编	85
代码区局部菜单	85
寄存器局部菜单	88
标志区局部菜单	88
数据区	89
数据区局部菜单	89
堆栈区局部菜单	92
汇编	93
操作数地址大小溢出	93
存贮区和立即数	93
操作数数据大小溢出	94
串指令	94
转贮窗口	94
寄存器窗口	94
Turbo C 代码生成	94
第十一章 80X87 协处理器芯片和模拟器	96
80X87 芯片和模拟器	96
数字处理机窗口	96
80 位浮点寄存器	97
状态位	97
控制位	97
寄存器区局部菜单	97
状态区局部菜单	98
控制区局部菜单	98
第十二章 命令参数	100
热键	100
主菜单下的命令	101
文件菜单	101
视频菜单	102
运行菜单	102
断点菜单	102
数据菜单	103

窗口菜单	103
选择项菜单	104
局部菜单命令	104
断点窗口局部菜单	104
CPU 窗口局部菜单	105
代码区局部菜单	105
数据区局部菜单	106
堆栈区局部菜单	107
寄存器区局部菜单	107
标志区局部菜单	107
文件窗口菜单	107
记载窗口菜单	108
模块窗口菜单	108
数字处理机窗口菜单	108
寄存器区局部菜单	109
状态区局部菜单	109
控制区局部菜单	109
堆栈区局部菜单	109
变量窗口菜单	109
全局符号区局部菜单	109
局部符号区局部菜单	110
观察窗口菜单	110
检测窗口局部菜单	110
文本区	110
列表区	111
提示箱命令	112
窗口移动命令	113
简写匹配	113
文件列表	113
完整的菜单树	114
 第十三章 如何调试程序	 117
当出现问题时	117
调试风格	117
运行整个程序	117
逐步测试	117
错误类型	118
C 特有的错误	118

一般性错误	121
Pascal 特有的错误	122
汇编特有的错误	127
精确度测试	129
测试界限条件和极端情况	130
错误数据输入	130
空数据输入	130
把调试当作程序设计的一部分	130
样例调试节	130
C 调试节	130
寻找错误	130
确定解决步骤	131
开始 Turbo Debugger 调试	131
检测	132
改正	132
Pascal 调试节	133
寻找错误	133
确定解决步骤	134
开始 Turbo Debugger 调试	134
检测	135
观察	136
最后一个错误	136
第十四章 80386 处理机上的虚拟调试	138
虚拟调试所需要的设备	138
安装虚拟调试设备驱动程序	138
开始虚拟调试	138
一般调试和虚拟调试间的区别	139
TD386 错误信息	139
TD386.SYS 错误信息	140
附录 A 命令选择项	141
附录 B Turbo Debugger 工具	142
附录 C 技术资料	149
附录 D 内在汇编关键字	152
附录 E 设置 Turbo Debugger 参数	155
附录 F 硬件调试接口	162
附录 G 远程调试(双机调试)	170

附录 H 提示和错误信息.....	174
附录 I 在不同语言下使用 Turbo Debugger.....	183
附录 J 字汇表.....	187

前 言

Turbo Debugger 是一个精巧的源程序级调试器,它是为使用 **Borland Turbo** 系列语言的程序员和使用其它编译器并需要有一个更强有力调试环境的程序员提供的。其重叠式窗口系统以及下拉和弹出选单为用户提供了一个快速的交互式接口,在线或上下文敏感的帮助系统可在操作的各个阶段提供服务。以下是它的几大特点:

- * 使用扩展内存规范(EMS)调试大程序
- * 完全的C表达式求值
- * 完全的Pascal表达式求值
- * 完全的汇编表达式求值
- * 屏幕布局可重新设置
- * 必要时可进行汇编语言/CPU一级的调试
- * 强有力的断点和记载(logging)设施
- * 按键记录宏
- * 使用双机系统调试很大的程序
- * 支持80386及其他销售商的调试硬件

软件及硬件需求

Turbo Debugger 在 IBM PC 系列上运行,包括 XT,AT,PS/2 系列及其它所有真正的 IBM 兼容机。它需要 DOS2.0 或更高版本以及至少 384K 的 RAM,可在任何彩/单色 80 列监视器上运行。虽然带有两个软盘驱动器的系统也工作得不错,我们建议还是使用一硬盘为好。

Turbo Debugger 不需要 8087 协处理器芯片。

要把 **Turbo Debugger** 用于 **Borland** 的产品,必须使用 **Turbo c 2.0**(或更高版本),**Turbo Pascal 5.0**(或更高版本),**Turbo Assembler 1.0**(或更高版本)。在使用 **Turbo Debugger** 时必须已经把源文件编译成可执行的 **.EXE** 文件,以得到全部的调试信息。

术语解释

为了说明方便,本手册中使用了一些与平常意义不同的术语,包括模块(Module),函数(Function)和参数(Argument)。

本手册中的模块不仅包括通常的 C 和汇编模块,还包括 Pascal 中的单元(Unit)。同样,本手册中的函数同时包括 C 函数和 Pascal 函数或过程。在 C 中,函数既可以返回值(类似 Pascal 的函数),也可不返回值(类似 Pascal 的过程)。另外,本手册中的参数(Argument)一词可以和参数(Parameter)互用,该词用于命令行或过程和函数中。

手册简介

以下是本手册各章内容简介:

第一章：启动。描述程序盘内容，介绍如何在系统里安装 Turbo Debugger，还对以下几章的学习提供一些有益建议。

第二章：调试和 Turbo 系列调试。介绍 Turbo Debugger 用户接口，菜单及窗口。并告诉用户如何对提示符和错误信息作出响应。

第三章：一个快速启动的例子。通过例子(Pascal 和 C 程序)，演示 Turbo Debugger 的各种功能。

第四章：开始用 Turbo Debugger。介绍如何在 DOS 提示符下运行 Turbo Debugger，何时使用命令行选择以及如何在配置文件中设置参数。

第五章：控制程序运行。介绍执行和停止程序的各种方法，并说明如何重新调试原来的程序或调试新的程序。

第六章：考察及修改程序数据。介绍 Turbo Debugger 特有的考察和修改程序中数据的方法。

第七章：断点。介绍动作(Action)的概念及它所包括的断点，观察点和跟踪点的概念。对于条件或无条件动作都作了解释，并指明了一个动作被捕获时出现的多种情况。

第八章：考察和修改源文件。介绍如何考察和修改源文件，包括任意的文本或二进制形式的盘文件。

第九章：表达式。介绍 Turbo Debugger 所能接受的 C, Pascal, 汇编表达式句法及修改显示格式时所用到的控制符。

第十章：汇编级调试。介绍如何观察或改变存储器内容，如何使用内在汇编和反汇编，如何检测或修改 CPU 寄存器内容和标志值。

第十一章：8087/80287 数学协处理器芯片和模拟器。介绍如何考察和修改一浮点硬件或模拟器的内容。

第十二章：命令参考。所有主菜单命令及对应于每一窗口类型的局部菜单命令列表。

第十三章：如何调试程序。介绍有效地调试程序的方法。

第十四章：80386 处理器的虚拟调试。介绍如何充分利用 80386 内存扩充的能力，使被调试的程序运行在 640K 以下的全地址中，就象没有安装 Turbo Debugger 一样。

附寻 A：命令行选择。对第四章所介绍命令行选择作一总结。

附寻 B：Turbo Debugger 工具。介绍如何使用 Debugger 提供的各类工具。这些工具程序包括和 Debugger 同时使用的 Codeview 程序和其他一些影响附加到 .EXE 文件调试信息的程序，还有用于显示部分文件内容的 TDUMP 程序。

附寻 C：技术资料。本章向有经验的程序员介绍 Turbo Debugger 的部分实现细节，说明它是如何和用户程序及 DOS 打交道的。

附寻 D：行汇编关键字。列出 8086/80286 所有指令助记符和其他特殊字。

附寻 E：设置 Turbo Debugger 参数。介绍如何利用安装程序(TDINST)设置屏幕颜色和改变缺省选择。

附寻 F：硬件 Debugger 接口。介绍在 Turbo Debugger 工作时如何写设备驱动器。

附录 G：远程调试(双机调试)。介绍如何利用双机系统调试大程序。

附寻 H：提示和错误信息。列出所有可能出现的提示和错误信息，并对如何

响应提了一些建议。

附寻 I: 在不同语言下使用 Turbo Debugger, 介绍一些调试用 C, Pascal 或汇编编写的程序的一些方法。

附寻 J: 词汇表。按字母顺序列出本手册中经常出现的术语。

第一章 启 动

Turbo Debugger 软件包

Turbo Debugger 软件包包括三张程序盘和 Turbo Debugger 用户手册(本手册)。程序盘包括了所有程序, 文件和工具, 以便调试用 Turbo C, Turbo Pascal, Turbo 汇编和其它一些用 Microsoft 编译器生成的程序。

Turbo Debugger 软件也包括一些本手册没有介绍的文档资料。

用户手册提供了 Turbo Debugger 能力的直接介绍和一个完整说明。

在使用 Turbo Debugger 以前, 应该对原始程序盘做一个完整的复制, 并把源盘放在安全的地方, 用复制盘去运行 Turbo Debugger。工作过程中, 各种情况随时都会发生。

程序盘

当你要在系统里安装 Turbo Debugger 时, 把程序盘中的文件拷贝到工作软盘或硬盘中。源程序盘没有被加密, 因此不需要运行特定安装程序。该盘为双面双密度软盘, 可以在 IBM PC 及其兼容机上运行。在程序盘中有以下文件:

TD.EXE	Turbo Debugger	运行程序
TDINST.EXE	Turbo Debugger	安装程序(用于设置Debugger参数)
TDHELP.TDH	Turbo Debugger	帮助文件
README.COM	最后更新的文件阅读程序	
README	最新信息	
TDH386.SYS	80386设备驱动程序	
TD386.EXE	虚拟调试程序	
TDRF.EXE	远程文件传送程序	
TDREMOTE.EXE	远程调试工具	
TDSTRIP	符号表抽取工具	
TDUMP.EXE	通用模块反汇编工具	
TDCONVRT.EXE	把CodeView程序转化为Turbo格式的工具	
TDMAP	把.MAP文件信息追加到.EXE文件末尾的工具	
TDPACK	缩小.EXE文件中调试信息大小的工具	
TCDEMO.*	C 语言演示程序	
TCDEMOB.*	第十三章所述的C错误演示程序	
TPDEMO.*	Pascal语言演示程序	
TPDEMOB.*	第十三章所述的Pascal错误演示程序	

注: 在原始盘中, 最后四类程序被压缩在 TDEXMPL.ARC 中, 在 DOS 提示符下使用

UNPACK TDEXMPL.ARC 即可得到这些程序。

安装 Turbo Debugger

应该把程序盘中的所有文件都拷到你的工作软盘或硬盘中(若空间不够,则至少应拷贝 TD.EXE 文件)。

在两软盘驱动器中使用 Turbo Debugger

对于两软盘驱动器的系统,应该使用不同的两张盘。一张装入 Debugger(TD.EXE)程序,另一张作为工作盘。

把原始程序盘中的所有文件拷贝到你的运行程序盘中,把你编制的源文件和待调试的可执行程序装入到工作盘中。使用时,把运行程序盘插在 A 驱动器中,把工作盘插到 B 中,确信 A 为当前驱动器后,打入 TD,通过 File/Change Dir 命令打入 B,把 B 改为当前驱动器,接着用 File/Load 命令装入待调试的程序。

在硬盘中建立 Turbo Debugger

在硬盘中使用 Turbo Debugger,可以极大地提高运行效率。

如果你想把 Turbo Debugger 放在经常使用的某一目录中,那么进入该目录并把程序盘中的所有文件都拷贝到该目录下。例如,你使用 Turbo C 并打算在该目录下应用 Debugger,提示符下打 C:\tc\copy a: ***,其中 TC 为你的 Turbo C 目录。同样 Turbo Pascal 用户可以把 Turbo Debugger 盘的内容拷到他们的 Pascal 目录下。也可以单独为 Debugger 建一目录,如命名为 DEBUG,把程序盘中的所有文件都拷到该目录下。为运行 Debugger,可以改变当前目录,也可以把 Debugger 目录包含在 PATH 环境变量中,以便不管在系统那个地方,都能加以调用。通过使用 PATH 环境变量, Turbo Debugger 可以自动发现帮助和配置文件。

硬件调试

如果用户是在 80386 系统中运行 Turbo Debugger,可以安装随 Turbo Debugger 一起提供的 TDH386.SYS 设备驱动程序,该驱动程序可以大大加快利用断点观察内存变化的过程。把本程序拷贝到保存设备驱动器的目录下(如\SYS),在 CONFIG.SYS 文件中增加一行

```
DEVICE=\SYS\TDH386.SYS
```

等你再次启动系统时, Turbo Debugger 就能发现并使用本设备驱动程序。

现在在哪儿

现在,你已经安装好了 Turbo Debugger 文件,并开始学习使用它。由于本手册是为二种不同类型的用户所写,你可能对其中某些章节更感兴趣,以下的提示将给你以指导。

正在学习 Turbo 语言的程序员:

如你刚开始学习 Turbo 系列的某一语言,那么在开始学习 Debugger 以前请先学会如何用该语言编制小程序.学习 Turbo Debugger 的最好方法是你自己确实有一程序需要调试.当你掌握了该语言后,再阅读第三章的指导,以便对 Turbo Debugger 的主要功能有一个很快的了解,从而学到为调试第一个程序所必须的知识.在以后各章中,我们将对 Debugger 的更复杂能力作进一步的介绍.

已经使用过 Turbo 语言的程序员:

如果你是一个熟练的 Turbo 系列程序员,那么请先阅读第二章,以便了解 Turbo Debugger 用户接口的新特点.若适合你的风格,则你可以跳过第三章,直接读第四章"开始用 Turbo Debugger",想对所有的命令有一个完整的了解,请看第十二章"命令参考".

第二章 调试和 Turbo 系列调试

以前有人认为自己从不犯错误,而事实上他错了,这就是为什么要有调试的原因。

最简单的道理是没有人是十全十美的,我们都会犯错误。不管是做象散步一样简单的事情还是编程序一样复杂的事情,我们有时都会感到困惑。

在编程中,困惑是一种正常现象。很少有程序员能一次编出一个没有错误的程序,对此不应感到奇怪,但困惑也意味着你暂停原来的工作而去测试,用程序设计的术语来说,就是调试。

什么是调试?

调试是发现和改进程序错误的过程。并不奇怪,在你开始写程序时,发现和改进错误的时间比写的时间还多。目前,调试还不是一门完善的科学,调试程序的最好工具是大脑,但有些建议可供参考(详见第十三章)。一般说来,调试可分为四大步:

- 1.发现有错误
- 2.找到该错误
- 3.查明错误原因
- 4.改正错误

有错误吗?

第一步可能是很明显的,运行着的计算机突然挂起了,或者由于出现一大堆乱七八糟的字符而崩溃了。但也可能不那么明显,也许只有当你超人一特定数(如零或负数)或仔细检查输出结果时,才能发现。

错误在哪儿?

第二步隔离错误可能是最困难的。在处理这个问题时,当然不可能把整个程序都记在脑子里(除非这个程序确实很小)。因此最佳的方法是分解--把程序分成几个部分后逐一加以调试。对于结构化编程尤其适用这种方法。

是什么错误?

第三步,找到错误原因可能是其次困难的一步。当找到错误后,通常都能知道程序为什么不正确运行。但有时,错误也可能是难以捉摸的,需要作进一步的追踪。

改正错误

最后一步是改正错误。当明白错误所在后,根据你所掌握的程序语言知识,很容易做到这一点。现在重新运行程序直到下一个错误出现。

在编程过程的许多时候,就在重复着以上四个步骤。例如,当出现句法错误时,编译就不会通过。Borland 语言有一内在的句法检查器会给你提供错误类型信息,以便你当场修改。

但还有一些错误可能是很隐含和微妙的。有可能只有当你输入一个负数时才出现。也有可能是如此难以捉摸以至于你陷入了困境,这时就需要有 Turbo Debugger 来帮忙。

Turbo Debugger 能做什么

使用独立的 Turbo Debugger 比语言编译器本身带有的调试程序具有更强大的能力(在语言编译器中加功能齐全的调试程序会使其变得相当庞大)。

使用 Turbo Debugger 可以调试用 Borland Turbo 系列及 Microsoft C, Pascal 和汇编等语言所编制的程序(在调试 Microsoft 的程序时需要有一转换工具)。也可以调试其他制造商的语言产品,但必须在汇编级进行(除非使用了附录 B 所介绍的 TDMAP 工具)。

使用 Turbo Debugger 有助于解决调试过程中最困难的两步:

找到错误及错误原因。

Turbo Debugger 以通过减慢程序执行和检查程序在任何时候的状态这一独特能力克服调试过程中的障碍。你甚至可以测试变量的值,以观察它是如何影响程序的。这些能力主要集中在:跟踪,单步,视频,检测,修改和观察中。

跟踪(Tracing): 一次执行一行程序。

单步(Step) : 一次执行一行程序,并跳过任何过程或函数调用。当你确信过程或函数是正确时,跳过它有助于加快调试速度。

视频(View) : 可以用 Turbo Debugger 开一特殊的窗口以显示十二种内容: 变量,变量值,断点,堆栈,记载,数据文件,源文件, CPU 代码,内存,寄存器,数字处理机信息和程序输出。

检测(Inspector): 可以用 Turbo Debugger 深入到工作程序的内部,以检测象堆栈那样复杂数据结构的内容。

修改(Modify): 可以修改全局或特定变量的当前值。

观察(Watches): 可以隔离程序变量,观察它在程序执行过程中的值变化。

你可以使用这一有力的工具把程序分为几部分,然后逐一加以验证。这样,不管程序有多复杂,都能找到错误所在。也许错误在于函数返回一错误值,也许是由于程序陷入了死循环或无限递归中。不管是什么错误, Turbo Debugger 都能帮助你发现并改正它。

Turbo Debugger 不能做什么

根据 Turbo Debugger 的以上特点,也许会认为它能做调试及运行过程中的所有事情,但事实上,它至少不能做以下三件事:

* Turbo Debugger 并没有内在的编辑程序来修改你的源程序。大多数的程序员都有自己最喜欢的编辑器。因此在 Turbo Debugger 中另加一编辑器是一种存贮浪费,你可以很容易地通过文件窗口中的局部 Edit 命令把控制传到你的编辑器中。Turbo Debugger 使用你在 TDINST 安装程序中指定的编辑器。

* Turbo Debugger 不能重编译你的程序。你需要有原始程序的编译器(如 Turbo Pascal 或 C)。

* Turbo Debugger 不能代替你进行思考。在调试一程序时,最主要的是动脑子想。

Turbo Debugger 只是一个工具，如果你不动脑子而去使用它，就不可能节约时间或提高效率。

Turbo Debugger 是怎样工作的

Turbo Debugger 提供强大的复杂的调试能力，而且非常容易使用，它通过一个友好的用户接口(User Interface)完成其种种功能，以下介绍其接口优点。

Turbo Debugger 的优点

一旦你开始使用 **Turbo Debugger**，我们相信你也会被它所迷住，**Turbo Debugger** 设计得既方便又容易，它具有以下几个特点：

- * 方便的逻辑下拉菜单。
- * 对内容相关的弹出菜单，免去了用户记忆和敲入命令的麻烦。
- * 当你确实需要敲字符时，**Turbo Debugger** 把你敲入的内容存入一张表中，你可以从这张历史表中，选择编辑文本或写到一文件中。
- * 完整的宏机制加快命令或键入系列。
- * 方便完整的窗口管理。
- * 多类型的在线辅导。

本章其余部分讨论**Turbo Debugger** UI的这些特点。

使用主菜单

同其他许多Borland产品一样，**Turbo Debugger**在屏幕的顶部有一方便的可移动式主菜单存取系统。不管在那个窗口，主菜单都是可见的，在主菜单的每一条目下，都有一下拉菜单。

有三种方法可以进入这些菜单：

- 1.按F10把光标移到特定项目下按回车键。
- 2.按 F10，再按各菜单的头一个字母(F,V,R,E,W,O)
- 3.按 ALT 和各菜单的头一个字母(F,V,R,E,W,O)去激活特定的命令菜单。如在系统的任何地方，按下 ALT_F，就进入文件菜单。

为在各个分菜单之间移动，可以采取以下方法：

- * 按ESC键退出本菜单，返回到原先激活的菜单(一菜单被激活是指有光标在该菜单里)。
- * 按 F10 返回到原来的活动窗口。
- * 按→或←从一个菜单移到另一个菜单。
- * 按 Home 或 End 键分别返回到第一和最后一个菜单。

主菜单中的有些命令是短键命令，称为热键(HOTKEYS)，对应的热键出现在菜单命令的右面。

图12.1 12.2 12.3显示**Turbo Debugger**完整的下拉菜单树，表12.1列出所有热键。要