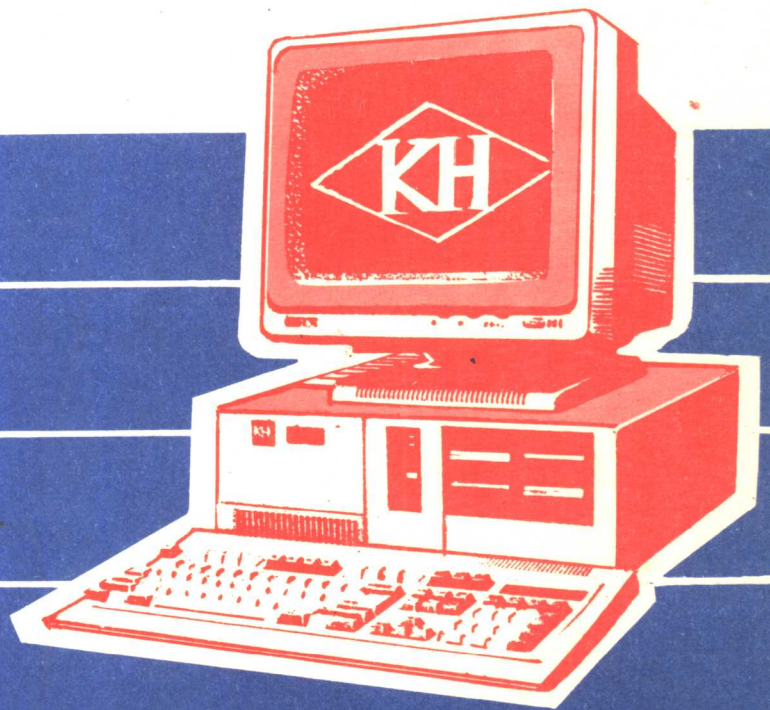


TURBO DEBUGGER 2.0 用户手册

科海培训中心
系列教材

作者 叶 新



876073
02

北京科海培训中心

Turbo Debugger(2.0 版)用户手册

林雪柏 荆立群 编译

北京科海培训中心

一九九〇年九月

目 录

前言	1
软件及硬件需求	1
术语解释	1
手册简介	2
第一章 启动	4
Turbo Debugger 软件包	4
程序盘	4
README 文件	4
HELPME!.DOC 文件	4
Turbo Debugger 实用程序	5
安装 Turbo Debugger	5
获得例子文件	5
LCD 和 B/W 显示器	6
硬件调试	6
现在在哪儿?	6
正在学习 Turbo 语言的程序员	6
已经使用过 Turbo 语言的程序员	6
第二章 调试和 Turbo 系列调试	7
什么是调试?	7
有错误吗?	7
错误在哪儿?	7
是什么错误?	7
改正错误	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
菜单和对话框	9
对话框	10
知道现在在哪里	11
局部菜单	11
历史机制	12
名字自动完成	13
渐增匹配(Incremental Matching)	13
宏生成(Making Macros)	13

窗口设置	14
视频窗口(View Window)	14
用户屏幕(User Screen)	16
检测窗口(Inspect Window)	16
活动窗口(Active Window)	16
窗口里有什么?	17
用窗口工作(Working with Window)	18
窗口切换(Window Hopping)	18
窗口移动和改变大小(Moving and Resizing Windows)	19
关闭和恢复(Closing and Recoving Windows)	19
保存屏幕配置(Saving Window Layout)	19
求得帮助	19
联机帮助	20
底部参考行	20
第三章 一个快速启动的例子	22
演示程序	22
使用 Turbo Debugger	23
菜单	23
状态行	23
窗口	23
使用 C 演示程序	24
使用 Pascal 演示程序	28
第四章 开始用 Turbo Debugger	34
准备待调试程序	34
准备 Turbo C 程序	34
准备 Turbo Pascal 程序	34
准备 Turbo 汇编程序	34
准备 Microsoft 程序	34
运行 Turbo Debugger	35
命令行选择项	35
配置文件	38
选择项菜单	38
在 Turbo Debugger 中运行 DOS	41
返回到 DOS	41
第五章 控制程序执行	42
检查当前程序状态	42
变量窗口	42

全局区局部菜单命令	43
静态区局部菜单命令	43
堆栈窗口	44
堆栈窗口局部菜单	44
源(Origin)局部菜单命令	44
获取信息(Get Info)命令	44
运行菜单	46
执行历史窗口(Execution History Window)	48
中断程序执行(Interrupting Program Execution)	49
Ctrl-Break	49
程序结束(Program Termination)	50
重新开始一个调试(Restarting a Debugging Session)	50
重装用户程序(Reloading Your Program)	50
键入宏记录和再调用(Keystroke Macro Recording and Playback)	50
打开新的待调试程序(Opening a New Program to Debug)	50
修改程序参数	51
第六章 观察和修改数据	52
数据菜单(Data Menu)	52
指向源文件的数据项(Pointing at Data Objects in Source File)	54
观察窗口(Watch Window)	54
观察窗口局部菜单	55
检测窗口	55
C 数据检测窗口	56
Pascal 数据检测窗口	58
汇编语言数据检测窗口	59
检测窗口局部菜单	61
第七章 断点	63
断点菜单	63
断点表达式范围	64
断点窗口	64
断点窗口局部菜单	65
记载窗口	67
记载窗口局部菜单	67
简单断点	68
条件断点和遍计数	68
全局断点	68
观察数据目标变化的断点	69
记载变量值	69

执行表达式	69
第八章 观察和修改文件	70
观察源程序文件	70
模块窗口	70
模块窗口局部菜单	71
观察其他磁盘文件	72
文件窗口	72
文件窗口局部菜单	73
第九章 表达式	75
选择计算表达式的语言	75
代码地址、数据地址和行号	75
在当前范围外存取符号	76
范围忽略语法	76
表达式求值的隐含范围	77
字节表	78
C 表达式	78
Pascal 表达式	82
汇编表达式	83
格式控制	84
第十章 C++和面向对象 Pascal 的调试	85
层次窗口	85
对象类型列表区	85
层次树区	86
对象类型/集合检测窗口(Object Type/Class Inspect Window)	86
对象类型/集合检测窗口局部菜单	87
对象/集合实例检测窗口	87
对象/集合实例检测窗口局部菜单	88
第十一章 汇编级调试	90
当源程序级调试不够时	90
CPU 窗口	90
代码区	91
反汇编	91
代码区局部菜单	92
寄存器和标志区(Registers and Flags Pane)	94
寄存器局部菜单	94
标志区局部菜单	95

数据区	95
数据区局部菜单	95
堆栈区	97
堆栈区局部菜单	97
汇编	98
操作数地址大小忽视	98
内存操作数和立即数	98
操作数数据大小忽略	98
串指令	99
转储窗口	99
寄存器窗口	99
Turbo C 代码生成	100
第十二章 80×87 协处理器芯片和仿真程序	101
80×87 芯片和仿真程序	101
数学处理器窗口	101
寄存器区	101
寄存器区局部菜单	102
控制区	102
控制区局部菜单	102
状态区	102
状态区局部菜单	103
第十三章 命令参考	104
热键	104
主菜单下的命令	106
局部菜单命令	108
通配符搜索样板	118
菜单结构树	118
第十四章 如何调试程序	119
当出现问题时	119
调试风格	119
运行整个程序	119
逐步测试	119
错误类型	120
一般性错误	120
C 特有的错误	121
Pascal 特有的错误	124
汇编特有的错误	129

精确度测试	131
把调试当作程序设计的一部分	131
样本调试过程	132
C 调试	132
Pascal 调试	135
第十五章 80386 处理机上的虚拟调试	140
虚拟调试所需要的设备	140
安装虚拟调试设备驱动程序	140
开始虚拟调试	140
普通调试和虚拟调试的区别	141
TD386 错误信息	142
TD386、SYS 错误信息	143
第十六章 用 80286 进行保护方式调试	144
保护方式调试器硬件要求	144
开始使用保护方式调试器	144
Turbo Debugger 和保护方式调试器的区别	144
在不同机器上运行 TD286	145
第十七章 调试 TSR 程序和设备驱动程序	146
什么是 TSR?	146
测试一个 TSR	146
什么是设备驱动程序?	148
调试一个设备驱动程序	148
结束调试过程	150
附录 A 命令行选择项	151
附录 B 技术资料	152
附录 C 内部汇编关键字	155
附录 D 设置 Turbo Debugger 参数	158
附录 E 远程调试	167
附录 F 对话框和错误信息	171
附录 G 在不同语言下使用 Turbo Debugger	187
术语表	191

前 言

Turbo Debugger 2.0 是美国 **Borland** 公司继 1.0 版后推出的功能更强的源程序级调试器，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大有力调试环境的程序员提供的。

重叠式窗口系统以及下拉和弹出菜单为用户提供了一个快速的交互式接口，联机或上下文有关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、Pascal 和汇编表达式求值
- 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
- 反向跟踪
- 远程系统调试很大的程序
- 支持 80386 及其他销售商的调试硬件
- 支持 Turbo Pascal 5.5 的面向对象编程
- 支持 C++ 和 Turbo C++ 调试
- TSR 和设备驱动程序调试

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 DOS 2.0 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。虽然在带有两个软盘驱动器的系统上也能工作得不错，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** 时必须先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，函数既可以有值返回(类似 Pascal 的函数)，也可不返回值(类似 Pascal 的过程)。另外，本书中参数(argument 或 parameter)一词除了在过程和函数中引用外，也可以在命令行里引用。

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, 还对后面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有的程序或调试新的程序。

第六章: 观察及修改程序数据。介绍 Turbo Debugger 特有的观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。对于条件动作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: 观察和修改文件。介绍如何观察和修改源文件, 包括任意的文本或二进制形式的磁盘文件。

第九章: 表达式。介绍 Turbo Debugger 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: C++和面向对象的 Pascal 调试。介绍让你观察 Turbo Pascal 程序中的对象和 Turbo C++程序中的集合。

第十一章: 汇编级调试。介绍如何观察或修改存储器内容, 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。

第十二章: 8087/80287 数学协处理器芯片和仿真程序。介绍如何考察和修改浮点硬件或仿真程序的内容。

第十三章: 命令参考。所有的主菜单命令及对应于每一窗口类型的局部菜单命令列表。

第十四章: 如何调试程序。介绍有效地调试程序的方法。

第十五章: 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。

第十六章: 用 TD286 进行保护方式调试。介绍如何在保护方式下运行 TD286。

第十七章: 调试 TSR 和设备驱动程序。介绍如何调试 TSR 和设置驱动程序。

附录 A: 命令行选择项。对第四章所介绍命令行选择项作一总结。

附录 B: 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。

附录 C: 内部汇编关键字。列出 8086/80286 所有指令助记符和其他特殊字。

附录 D: 设置 Turbo Debugger 参数。介绍如何利用安装程序(TDINST)设置屏幕颜色和更改选择项缺省值。

附录 E: 远程调试。介绍如何利用远程系统调试大程序。

附录 F：对话框和错误信息。列出所有可能出现的对话框和错误信息，并对如何响应提了一些建议。

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

术语表：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 软件包由一套程序盘和 Turbo Debugger 用户手册(本手册)组成。程序盘包含了所有程序、文件和工具,以便调试用 Turbo C、Turbo Pascal、Turbo 汇编和其它用 Microsoft 编译器生成的程序。另外在 README 和 HELPME!.DOC 文件中还包括一些手册中所没有的资料。

用户手册提供了 Turbo Debugger 性能的逐个介绍和完整的命令说明。

在使用 Turbo Debugger 以前,应该对源程序盘做一个完整的工作副本,并把源盘放在安全的地方,仅在备份时用源盘。用工作盘去运行 Turbo Debugger。因为源盘是你工作盘出问题时的唯一后备。

程序盘

当你要在系统里安装 Turbo Debugger 时,要把程序盘中的文件拷贝到工作软盘或硬盘上去,这只要运行程序盘上的 INSTALL.EXE 即可。源程序盘为双面双密度软盘,可以在 IBM PC 以及真正的兼容机上使用。

README 文件

请注意,在你安装 Turbo Debugger 之前应先阅读一下安装盘上的 README 文件。该文件中可能包含手册中没有的最新信息。另外,它还包含程序盘上所有文件的一个列表,并带有简单的说明。

为了读 README 文件,将安装盘插入 A 驱动器,键入 A:并回车转到 A 盘,然后再键入 README 和回车。阅读时,用 ↑ 和 ↓ 浏览文件。按 Esc 退出。

HELPME!.DOC 文件

安装盘上还包含 HELPME!.DOC 文件,它回答用户经常碰到的问题。如有困难则请参考该文件。另外,HELPME!.DOC 还解决下列问题:

- 对于基于图形和文本程序的屏幕输出
- 继续使用 Turbo Debugger 的同时运行其他程序
- 程序的中断
- Turbo Debugger 与其他 Turbo 语言的语义及词法区别
- 用 Turbo Debugger 调试多语言程序
- Tandy 1000A, IBM PC Convertible 或 NEC MultiSpeed 以及其他使用 NMI(不可屏蔽中断, nonmaskable interrupt)的计算机

Turbo Debugger 实用程序

Turbo Debugger 随带了几个实用程序。在程序盘上有这些程序的详细说明。参见 README 文件中有关如何存取这些基于磁盘的文档。

- 从 CodeView 到 Turbo Debugger 的实用程序, TDCONVRT.EXE, 使你可以调试用 Microsoft 编译器开发的 C 或汇编程序
- 远程文件传输实用程序, TDRF.EXE, 用于远程调试并提供远程系统中基本的文件维护命令
- 符号表去除实用程序, TDSTRIP.EXE, 从你的文件中去除调试信息(即符号表), 而不需重新连接
- TDPACK.EXE 压缩调试信息
- TDMAP.EXE 将调试信息附加到 .MAP 文件中
- 最后, TDUMP.EXE 是强有力的目标模块和 .EXE 文件的反汇编程序
- 另外, 还提供一个很小的 TSR 程序 TDNML.COM, 以便使用 PERICSOP E I 板时重新设置开关

注意:

如要获取 TDCONVRT.EXE、TDRF.EXE、TDSTRIP.EXE、TDPACK.EXE、TDMAP.EXE 或 TDUMP.EXE 的所有的命令行选择项, 只要键入程序名和回车即可。例如, 要看 TDMAP.EXE 的命令行选择项, 则键入

TDMAP

即可。

安装 Turbo Debugger

安装盘中有文件 INSTALL.EXE 帮助你安装 Turbo Debugger 2.0。

安装时, 将当前驱动器改为插有 INSTALL 程序的驱动器, 然后键入 INSTALL。屏幕底部的窗口中有每一步的说明。

INSTALL 将所有的 Turbo Debugger 文件拷贝到你的硬盘上的子目录里。缺省的子目录为:

Turbo Debugger 子目录: C:\TD

例子子目录: C:\TD

在缺省情况下, 程序盘上的所有文件都放在 Turbo Debugger 子目录下。如果想要一个单独的例子子目录, 则在选择开始安装之前编辑缺省的例子文件路径。

安装 Turbo Debugger 之后, 应阅读文件 README 以获取有关 Turbo Debugger 的更进一步的说明。

注意: 要获得 INSTALL.EXE 的所有命令行选择项, 键入: INSTALL -h。

获得例子文件

Turbo Debugger 程序盘上有文件 TDEXAMPL.ZIP。它包含一些压缩了的例子文件。

可以使用 UNZIP.EXE 获取这些例子。例如，键入

UNZIP TDEXAMPL

释放 TDEXAMPL.ZIP 中的所有文件到当前目录。INSTALL 允许你选择在安装过程中是拷贝整个.ZIP 文件还是所有单独的例子程序到你的硬盘上。

LCD 和 B/W 显示器

如果 INSTALL 显示有问题，则应使用 /B 命令行参数迫使其使用黑白(B/W80)方式：

```
A:INSTALL /B
```

如果你使用 LCD 显示器或在彩色图形卡上使用单色显示器也应使用 /B 参数。

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被修改的内存区域和 I/O 端口存取过程。把该文件拷贝到你的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 就能发现并使用该设备驱动程序。

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。由于本手册是为两种不同类型的用户编写，你可能对其中某些章节比较感兴趣，以下的提示将给你以指导。

正在学习 Turbo 语言的程序员

如果你刚开始学习 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需的知识。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 TurboDebugger 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，请看第十三章“命令参考”。

第二章 调试和 Turbo 系列调试

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？

调试是发现和改正程序错误的过程。并不奇怪，在你开始写程序时，发现和改正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，当然还有一些系统化的方法。一般说来，调试大致可分为四个步骤：

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，也许只有当你输入一个特定数(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，即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，即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，还需要作进一步的追踪。

改正错误

最后一步是改正错误。当明确错误所在后，根据你所掌握的程序语言知识，很容易做到这一点。现在可再次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(在语言编译器中增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 的 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(除非有 CodeView 信息)。

使用 Turbo Debugger 有助于解决调试过程中最困难的两步：找到错误及错误发生原因。

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):	一次执行一程序行。
反跟踪(Back Tracing):	往回检查已执行过的代码。
单步(Stepping):	一次执行一程序行，并跳过任何过程或函数调用。当你确信过程或函数是正确的时，跳过有助于加快调试速度。
视频(Viewing):	可以用 Turbo Debugger 开一个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、数值处理机信息和程序输出。
检测(Inspecting):	可以用 Turbo Debugger 深入到工作程序的内部，以检测象堆栈那样的复杂数据结构的内容。
修改(Changing):	可以修改全局或局部变量的当前值。
观察(Watching):	可以隔离程序变量，观察它们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个部分，然后逐一加以验核。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。

Turbo Debugger 2.0 可以调试 C++ 和面向对象的 Pascal 程序。它懂得对象和集合，正确处理虚拟方法或成员函数的以后连接，从而问题可以正确执行和显示代码。

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 并没有内部的编辑器来修改你的源程序。大多数的程序员都有自

己最喜欢的编辑器。因此在 Turbo Debugger 中另加一编辑程序是一种存贮浪费，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程序中。Turbo Debugger 使用你在 TDINST 安装程序中指定的编辑程序。如果是 Turbo C++，则可以使用新的控制传递特性，在 Turbo 语言的集成环境中运行 Turbo Debugger。

- Turbo Debugger 不能重新编译你的程序。你需要有源程序的编译器(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，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, UI)完成其种种功能，以下介绍其接口的优点。

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：

- 方便的具有逻辑性的下拉菜单
- 上下文有关的弹出式局部菜单，免去了用户记忆和键入命令的麻烦
- 在对话框中，可以选择、设置和切换选择项，键入各种信息
- 完整的宏机制加快命令或键入系列
- 支持鼠标器
- 方便完整的窗口管理
- 多类型的联机帮助
- 会话记录及反向执行

本章其余部分讨论 Turbo Debugger UI 的这些特点。

菜单和对话框

同许多 Borland 其他产品一样，Turbo Debugger 在屏幕的顶部有一方便的全局主菜单系统。不管在哪个窗口，主菜单都是可见的，在主菜单的每一项下，都有一个下拉菜单。通过下拉式菜单你可以：

- 执行一条命令
- 打开弹出式菜单。当你选择后面跟有菜单图标(menu icon)的菜单时，出现弹出式菜单
- 打开对话框。当你选择后面跟有对话图标(dialog icon)的菜单时，出现弹出式菜单

有四种方法打开这些菜单：

1. 按 F10，把光标移到特定项目下按回车键