计算机软件开发系列丛书

Turbo Assembler 2.5 使用与参考手册

金子方
碧　河　编写

本书介绍 Turbo Assembler 2.5的安装,程序设计入门,命令行参考、与Borlard C++、Turbo Pascal等的接口

学苑出版社

Turbo Assembler 2.5 使用与参考手册 金子方 等编写 学苑出版社

计算机软件开发系列丛书

Turbo Assembler 2.5 使用与参考手册

金子方　碧河等　编写
吴　君　　审校

学苑出版社
1993.

(京)新登字 151 号

内 容 提 要

本书主要介绍美国 Borland 公司的一次性扫描汇编器的安装、程序设计入门、命令行参考、扩充与改进、与 Borland C++、Turbo PASCAL 的接口技术等。

欲购本书的用户，请直接与北京 8721 信箱联系，电话 2562329，邮码 100080。

计算机软件开发系列丛书

Turbo Assembler 2.5 使用与参考手册

编　　写：金子方　碧河等
审　　校：吴　君
责任编辑：甄国宪
出版发行：学苑出版社　　邮政编码：100032
社　　址：北京市西城区成方街 33 号
印　　刷：施园印刷厂
开　　本：787×1092　1/16
印　　张：25.625　　字　数：607 千字
印　　数：1～3000 册
版　　次：1993 年 11 月北京第 1 版第 1 次
ISBN7－5077－0779－2/TP・11
本册定价：17.00 元

学苑版图书印、装错误可随时退换

目　录

内容简介

欢迎使用 Borland 公司开发的一次性扫描汇编器 Turbo Assembler。Turbo Assembler 具有超前引用特征，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(理想)模式。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序设计的其它特征。以下列出其中的一些优点，在本书的后续章节中将有更详细的介绍：

* DOS 保护模式接口(DPMI)支持在 Mircosoft Windows 下以保护模式运行 Turbo Assembler
* 全 386 和 i486 支持
* 改进的语法类型检查
* 简化的段指令
* 改进的列表控制
* PUSH、POP 扩展
* 扩展的带变元和可选语言参数的 CALL 语句
* 局部标号
* 局部的栈符号及过程参数调用
* 结构和联合
* 嵌套指令
* 模拟 MASM 的 Quirks 模式
* Turbo Debugger 的全源调试输出
* 内部交叉引用工具(TCREF)
* 配置文件和命令文件

Turbo Assembler 是一个功能强大的命令行汇编器，它接受用户的源文件(.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)

Turbo Assembler 在 80x86 和 80x87 系列处理器环境下工作。

§0.1 硬件和软件需求

Turbo Assembler 在包含 XT、AT、PSA/以及所有兼容机在内的 IBM PC 系列机上运行。它要求 MS—DOS 2.0 或更高版本的操作系统，并至少配备 256K 内存。

Turbo Assembler 产生的是 8086、80186、80286、386 及 i486 处理器的指令。它也可产生 8087、80287、80387 数字协处理器的浮点数指令。

§0.2 本书内容

第一部分 使用 Turbo Assembler

第一章："安装 Turbo Assembler"说明如何在系统上安装 Turbo Assembler

第二章："Turbo Assembler 程序设计入门"简单介绍了汇编语言程序设计并给出了一些例子。

第三章："命令行参考"详细讨论了所有命令行选项，并说明了如何使用配置文件和命令文件。

第四章："Turbo Assembler 要点"说明了 Turbo Assembler 对 MASM 的增强功能。

第五章："Turbo Assembler 与 Borland C++的接口"说明如何在 Borland 的 C++编译器中使用汇编语言。我们详细讨论了如何将汇编模块与 C++连接，以及如何在 C++中使用 Turbl Assembler 函数。

第六章："Turbo Assembler 与 Turbo Pascal 的接口"说明了汇编代码如何与 Turbo Pascal 代码交互，并提供了样例程序。

第七章："Turbo Assembler 的 Ideal 模式"介绍了 Ideal 模式，以及如何使用它。

第二部分 参考

第八章："预定义符号"介绍了 Turbo Assembler 的预定义符号

第九章："操作符"介绍了 Turbo Assembler 提供的各种操作符。

第十章："指令"以字母顺序详细讨论所有的 Turbo Assembler 指令。

第三部分 附录

附录 A："Turbo Assembler 语法摘要"以修改的 BNF 形式说明了 Turbo Assembler 表达式(MASM 和 Ideal 模式)。

附录 B："兼容性问题"讨论了 MASM 与 Turbo Assembler 的 MASM 模式之间的区别。

附录 C："实用工具"讨论了软件包中提供的四个工具：MAKE，TLINK，TLIB 和 THELP，关于 GREP、TCREF 和 OBJXREF 的信息在磁盘的文件中。

附录 D："错误信息"讨论了 Turbo Assembler 可能会出现的四类错误信息：消息性信息，致命错误信息，警告信息，及错误信息。

§0.3 符号约定

谈及 IBM PCs 或其兼容机时，指使用 8088、8086、80186、80286、80386 和 i486 芯片(所有这类芯片常称作 80x86)的任何机型。涉及 PC—DOS、DOS 和 MS—DOS 时，指 2.0 版本或更高版本的操作系统。

第一章　安装 Turbo Assembler

在进行汇编语言程序设计之前，用户务必完成以下事项。取出 Turbo Assembler 盘并为每张盘创建一个拷贝(用 DOS 命令)，以形成用户自己的“工作”拷贝。在此之后可保存好原程序盘。

如果希望用 Turbo Assembler 代替 MASM，那么可参看附录 B，以了解 Turbo Assembler 与 MASM 的不同之处。

注意：使用 Turbo Assembler 之前务必阅读 README 文件，该文件囊括了有关程序及纠错和(或)手册附加内容的最新信息。

§1.1　安装 Turbo Assembler

INSTALL 盘上有一个名为 INSTALL.EXE 的程序，它将帮助用户安装 Turbo Assembler。有两种可选择的安装方式：

1. 硬盘用户：该可选项允许用户选择装载文件的子目录。

2. 软盘用户：该可选项将使用 Turbo Assembler 时所需要的文件安装到双软盘驱动器系统中。在开始安装前务必准备几张已格式化的软盘。

要开始安装时，将当前驱动器改换到含有 INSTALL 程序的驱动器并打入 INSTALL。对屏幕底框中出现的每一提示，系统都给出相应的说明。例如，如果在 A 驱动器中安装 Turbo Assembler，则打入

```
INSTALL
```

在安装 Turbo Assembler 之前，务必阅读 README 文件以得到与该方面有关的更进一步的信息。

注意：如果希望在便携式电脑或使用 LCD 显示器的其它系统中运行 INSTALL 程序，那么在运行 INSTALL 之前应当将系统设置成黑白模式。可在 DOS 环境下用下列命令实现上述功能：

```
Mode bw80
```

用户亦可使用/b 开关强制 INSTALL 在黑白模式下工作：

```
INSTALL /b
```

第二章　Turbo Assembler 程序设计入门

如果用户以前从未使用汇编语言设计程序，那么可从此处开始阅读。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。

仅用汇编语言就可书写出完整的程序。用户也可将汇编语言嵌插在用 Turbo C、Turbo Pascal、Turbo prolog、Turbo Basic 及其它语言编写的程序中。无论在哪种方式下，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。首先，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。其次，本章将涉及常用的计算机及特殊的 8086 处理器的一些特点，以便用户了解 8086 汇编程序的特殊功能。本章也讨论了汇编语言程序设计中尤其与 IBM PC 有关的一些问题。

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》和《Personal Computer XT Technical Reference》三本手册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。

进一步阅读之前，用户可能希望阅读第三章"命令行参考"，以熟悉命令行选择项。如果还没有安装 Turbo Assembler，则应该按照第一章的介绍安装它。

最后一点：汇编语言是一种复杂的语言，要编写甚至是相对而言很简单的汇编语言程序也需要了解许多知识。有时在例子中使用了尚未介绍的语言特征，也是出于上述原因。在后续章节中将解释所有的语言特征。无论何时，只要用户对某一特征感到迷惑不解，就可查阅第十章"指令"部分。

阅读了这些说明之后用户就可以创建自己的第一个汇编语言程序了。

§2.1　编写第一个 Turbo Assembler 用户程序

在程序设计领域里，第一个程序通常是一个显示信息的程序"Hello, World"，这是一个最好的起点。

```
.MODEL small
.STACK 100h
.DATA
HelloMessage DB 'Hello, World', 13, 10, '$'
.CODE
mov as, @data
```

```
mov dx,ax                       ;set DS to point to the data segment
mov ah,9                        ;DOS print string function
mov dx,OFFSET HelloMessage      ;point to "Hello,World"
int 21h                         ;display "Hello ,World"
mov ah,4ch                      ;DOS terminate program function
int 21h                         ;terminate the program
END
```

在输入 Hello. ASM 程序之后，将它存入磁盘。

熟悉 C、C++或 Pascal 语言的用户可能会想到，用以显示"Hello,World"的汇编语言程序看上去似乎稍长了一些。相对而言，汇编程序的确较长，因为每条汇编指令本身比 C、C++或 Pascal 指令完成的功能要少。另外，用户在组织这些汇编指令时有很大的灵活性，这意味着与 C 或 Pascal 不同，汇编语言可让用户对计算机编程，以让计算机做它力所能及的任何事情——要打入一些附加行时，这往往很有价值。

§2.1.1 汇编第一个程序

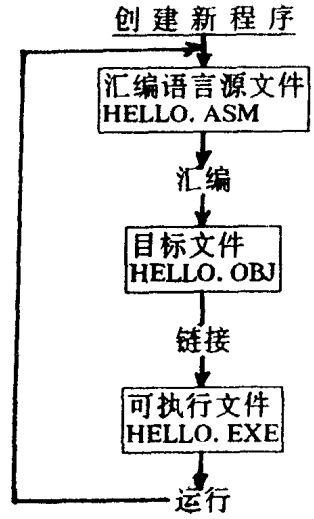

图 2.1 编辑、汇编、链接和运行循环

保存了 HELLO. ASM 文件之后，用户可能希望运行该文件。但只有将它转换成可执行的形式(可以运行或执行)后，该程序才可运行。这就要求要有两个附加的步骤，即如图 2.1 所示的汇编和链接。图 2.1 描述了编辑、汇编、链接和运行程序的循环开发过程。

通过汇编这一步可将源代码转换成称之为目标模块的中间形式，通过链接可将一或多个目标模块组合成一个可执行的程序代码。可在命令行中完成汇编和链接工作。

要汇编 HELLO. ASM，可打入

```
TASM hello
```

如果不指明其它文件名，HELLO. ASM 将被汇编成 HELLO. OBJ。(注意，用户不必打入文件扩展名；Turbo Assembler 会将. ASM 作为扩展名。)用户可看到屏幕上出现下列内容：

Turbo Assembler Version 2.5 Copyright (C) 1988,1991 by Borland International,Inc

Assembling file:HELLO.ASM

Error message:None

Warning message:None

Passes:1

Remaining memory:266K

如果完全按上述 HELLO.ASM 的内容准确地进行输入,那么屏幕上不会出现任何警告和错误。出现警告和错误时,警告和错误信息会显示在屏幕上,同时还会指出出错行的行号。如果出现错误,可检查程序代码并使之与上述内容完全一致,然后再次对其进行汇编。

§2.1.2 链接第一个程序

在成功地汇编了 HELLO.ASM 之后,与运行第一个汇编程序就只有一步之隔了。一旦链接了刚汇编过的目标模块并使之成为可执行的代码形式,用户就可以运行此程序。

要对程序进行链接,可使用 Turbo Assembler 自带的链接器 TLINK。在命令行中打入

TLINK hello

同样没有必要打入文件扩展名;TLINK 假定其扩展名为.OBJ。完成链接后(同样,至多需要几秒钟的时间),链接器自动以目标文件的名字命名.EXE 文件,除非用户规定其它的名字。如果链接成功,屏幕上出现如下信息:

Turbo Linker Version 3.0 Copyright(c)1987,1991by Borland International,Inc.

链接过程中可能会出现错误,但本程序样例在链接时不会出错。如果用户看到错误信息(出现在屏幕上),可修改源代码,使之与上述内容完全一致,然后重新汇编和链接。

§2.1.3 运行第一个用户程序

现在可以运行第一个用户程序了。在 DOS 提示符下打入 hello,这时信息

Hello,World

将被显示在屏幕上,并且这是程序所做的全部工作……你已经创建并运行了第一个汇编语言程序!

§2.1.4 发生了什么?

在创建并运行了 HELLO.ASM 程序后,请回想一下,从进入正文状态直到运行程序为止,究竟做了哪些工作。

当首次输入源汇编代码时,程序正文由正文编辑器保存在内存中。如果此时由于任何原因导致关机,那么源代码将会被丢失;所以建议用户经常尽快地保存源代码,以避免可能的灾难。将源代码保存到磁盘中时,形成正文的永久性拷贝并存入文件 HELLO.ASM 中,既使在关机之后文件 HELLO.ASM 也仍然存在。(但磁盘损伤时,文件 HELLO.ASM 也可能难逃厄运。)HELLO.ASM 是一个标准的 ASCII 正文文件;用户可在 DOS 提示符下通过打入

TYPE hello.asm

来显示它,用户也可使用任何正文编辑器对其进行编辑。

汇编 HELLO. ASM 时，Turbo Assembler 将 HELLO. ASM 中的正文指令转换成与其等价的二进制形式，并存入目标文件 HELLO. OBJ。HELLO. OBJ 是一种介于源代码与可执行文件之间的中间文件，其中包含有根据 HELLO. ASM 中的指令形成可执行代码时所需的全部信息，但它采用的是一种更有利于与其它目标文件链接组成一个完整程序的形式。

然后，在链接 HELLO. OBJ 时，TLINK 将它转换成可执行文件 HELLO. EXE。最后，当在提示符后打入 hello 时，HELLO. EXE 被执行。

现在可打入

```
dir hello. *
```

以列出用户盘上的 HELLO 文件。用户可看到屏幕上显示出 HELLO. ASM、HELLO. OBJ、HELLO. EXE 和 HELLO. MAP。

§2.2 修改第一个 Turbo Assembler 程序

现在可回到编辑状态并修改 HELLO. ASM 程序，使之可接受外界输入。将源代码修改成如下形式：

```
    .MODEL small
    .STACK 100H
    .DATA
TimePrompt DB 'Is it after 12 noon (Y/N) $'
GoodMorningMessage LABEL BYTE
DB 13,10,'Good morning,world!'13,10'$'
GoodAfternoonMessage LABEL BYTE
      DB 13,10,'Good afternoon,Worrld',13,10,'$'
    mov ax,@data
    mov ds,ax                          ;Set DS to point to data segment
    mov dx,OFFSET TimePrompt           ;point to the time prompt
    mov ah,9                           ;DOS print string function #
    int 21h                            ;display the time prompt
    mov ah,1                           ;DOS get character function  #
    int 21h                            ;get a single—character response
    cmp al,'y'                         ;typed lowercase y for after noon?
    jz IsAfternoon                     ;yes,it's after noon
    cmp al,'Y'                         ;typed uppercase Y for after noon?
    jnz IsMoring                       ;no,it's begore noon
IsAfternoon:
    mov dx,OFFSET GoodAfternoonMessage  ;point to the afternoon greeting
    jmp DiaplayGreeting
IsMoring:
    mov dx,OFFSET GoodMoringMessage     ;pointer to the before noon greeting
DisplayGreeting:
```

```
mov ah,9                    ;DOS print string function #
int 21h                     ;display the appropriate greeting
mov ah,4ch                  ;DOS terminate Program function #
int 21h                     ;terminate the program
END
```

这就为程序增加了两个重要的新功能:输入和决策。该程序询问是否是中午以后,然后从键盘上接受一个单字符响应。如果打入的字符是大写或小写的Y,程序就会显示出一句适用于下午的问候语;否则,程序显示出适用于上午的问候语。该程序代码包含了一个实用程序所必需的所有元素——从外界的输入、数据处理、决策、向外界的输出。

将修改后的程序存入盘中。(修改后的文件将代替原来的HELLO.ASM文件,所以原版本文件会丢失。)按前面例子中的操作步骤重新汇编并重新链接该程序,然后在DOS提示符下打入hello运行之。屏幕上显示出如下信息:

```
Is it after 12 noon(Y/N)?
```

光标在问号后闪烁,等待用户作出回答。按下Y键,程序响应如下:

```
Good afternoon,World!
```

现在,HELLO.ASM成为一种交互式的决策程序。

在汇编语言程序设计过程中,用户肯定会在输入和程序语法方面产生大量的错误。在汇编用户代码时,Turbo Assembler捕获这些错误并报告出错信息。报告的错误分作两类:警告和出错。检测出代码中有模棱两可、但不一定会引起错误的部分时,Turbo Assembler就显示出警告信息。有时可忽略警告,但最好能检查出来并理解之所以然。如果检测代码中有明显错误,致使不可能完成汇编并产生目标文件时,Turbo Assembler就显示出错信息。换言之,警告指非致命性错误,而出错时,只有在处理错误之后才可能运行程序。Turbo Assembler中可能出现的错误和警告信息被收集在附录D中。

与使用其它程序设计语言一样,Turbo Assembler不能捕获逻辑错误。Turbo Assembler可以告知用户代码能否被汇编,但它不能保证汇编过的代码能完成用户所期望的功能——只有用户本人才能确知这一点。

要列出或在打印机上打印用户程序,可参阅所用的正文编辑器的参考手册。Turbo Assembler源文件是标准的ASCII正文文件,所以也可在DOS提示符下用PRINT命令打印任何源汇编文件。

§2.2.1 将输出送往打印机

打印机是一种方便的输出设备;有时不仅希望将程序文件送往打印,也希望将程序的输出结果送往打印机。下面的程序将在打印机上打印而不是在屏幕上显示"Hello,world"

:

```
.MODEL small
.STACK 100h
.DATA
HellowMessage DB"Hello,world",13,10,12
```