计算机 C/C++语言系列丛书

Turbo Assembler 4.X
程序设计与实例

陈忠仁 编著

希望

学苑出版社

计算机C/C++语言系列丛书

Turbo Assembler 4.X
程序设计与实例

陈忠仁　编著

万　博　审校

学苑出版社

(京)新登字 151 号

内 容 简 介

本书主要是为用 Turbo Assembler 4.0～4.5 进行汇编程序设计的人编写的。首先介绍 Turbo Assembler 4.0～4.5 的特色，说明如何编写简单的汇编程序；然后介绍 Turbo Assembler 的伪指令和开关及程序设计概念，说明如何建立面向对象的程序，如何使用表达式和符号值，如何选择处理器伪指令和符号，如何定义数据类型，如何使用程序模式和段，如何使用过程等；接着介绍条件伪指令、与链接器的接口、宏的使用等；之后介绍 Turbo Assembler 4.0～4.5 与 Borland C++和 Borland Pascal 的接口，说明如何用汇编语言编写 Windows 应用程序；附录给出了一些有用的参考信息。

需要本书的用户可直接与北京海淀 8721 信箱书刊部联系，电话 2562329，邮政编码 100080。

计算机 C/C++语言系列丛书

Turbo Assembler 4.X 程序设计与实例

编　　著：陈忠仁
审　　校：万　博
责任编辑：甄国宪
排　　版：万博图书创作社
出版发行：学苑出版社　　**邮政编码**：100036
社　　址：北京市海淀区万寿路西街 11 号
印　　刷：施园印刷厂
开　　本：787×1092　1/16
印　　张：26.875　　**字　　数**：669 千字
印　　数：1～3000 册
版　　次：1994 年 9 月北京第 1 版第 1 次
ISBN7-5077-0875-6/TP・24
本册定价：35.00 元

学苑版图书印、装错误可随时退换

引　言

欢迎使用 Borland 公司开发的多遍扫描汇编器 Turbo Assembler。Turbo Assembler 具有超前引用性，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 方式。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计设计的其它特征。以下仅列出其中的一些优点，在后续章节将有详细介绍：

- 面向对象的程序设计
- 32 位模式和堆栈框架支持
- 全 386、i486 和 Pentium 支持
- 简化的段指令
- 表支持
- 枚举
- 灵巧的 flag 指令
- 新的结构特性
- 快速立即乘法
- 多行定义支持
- VERSION 说明指令
- 嵌套指令
- 模拟 MASM 的 Quirks
- Turbo Debugger 的全源调试输出
- 内部交叉引用实用程序(TCREF)
- 配置文件和命令文件
- C 的.h 到 TASM.ash 文件转换实用程序

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

0.1　硬件和软件需求

Turbo Assembler 在包括 *XT*、*AT*、*PS/2* 以及所有兼容机在内的 *IBM PC* 序列机上运行。它要求 *MS—DOS*2.0 或以上版本，并至少配备 256*K* 内存。

Turbo Assembler 产生的是 8086、80186、80286、386 及 *i*486 指令代码。它也可产生 8087、80287、387 数字协处理器的浮点数指令(有关 80*x*86/80*x*87 序列指令集的更详细的内容，请参看 *Intel* 公司的其它有关资料)。

0.2 DPMI 支持

Turbo Assembler 支持 *DOS* 保护方式接口(*DPMI*)说明。*DPMI* 定义了能充分利用 80286、386 和 *i*486 处理器的优点的标准接口。这些处理器的保护方式和虚 8086 方式明显改变了我们进行计算的方式。现在可以使用多任务并使用扩展内存,而在过去,让应用程序和其它软件一起使用扩展内存进行多任务处理是很困难的。*DPMI* 标准解决了这个问题,用 *DPMI* 标准开发的应用程序可以在多任务操作系统下可靠地运行。

当前,*Turbo Assembler* 使用 *Microsoft Windows* 提供的 *DPMI* 服务,*Turbo Assembler* 也可以在使用这些 *DPMI* 服务的其它环境下运行。如果转换到新的环境中,但仍使用这些服务,那么你在软件上的投资将得到保护。

TASM.EXE 包含 *DPMI* 支持,详见第二章。

0.3 本书的内容

本书提供使用 *Turbo Assembler* 所需的基本指令,与其它语言的接口,详细描述算符,预定义符和 *Turbo Assembler* 使用的指令,并给出了编写 *Windows* 应用程序的示例。

下面详细介绍用户手册的内容:

第一章:"入门"介绍如何在你的系统上安装 *Turbo Assembler*。

第二章:"使用伪指令和开关"描述当使用伪指令和开关时如何控制 *Turbo Assembler* 的运行方式。

第三章:"程序设计概念"讨论 *Ideal* 方式和 *MASM* 方式之间的差异,如何使用预定义符、注释等等。

第四章:"建立面向对象程序"描述在汇编语言中如何用 *OOP* 技术。

第五章:"使用表达式和符号值"讨论表达式和算符的计算及定义。

第六章:"选择处理器伪指令和符号"讨论如何为特定处理器产生代码。

第七章:"使用程序模式和段"讨论程序模式,建立符号、简化段和段排序。

第八章:"定义数据类型"解释如何定义结构、联合、表、位字段记录和对象。

第九章:"设置和使用位置计数器"描述如何和为什么要用定义计数器、如何定义标号。

第十章:"声明过程"检查如何使用各种过程、如何定义和使用参数及局部变量。

第十一章:"控制符号作用域"讨论具有特定值的符号的作用域。

第十二章:"分配数据"介绍简单数据指令,如何建立结构、联合、记录、枚举数据类型、表和对象的实例。

第十三章:"高级编码指令"覆盖了 *Turbo Assembler* 的扩展指令,包含原型和调用语言过程。

第十四章:"使用宏"介绍如何在程序中使用宏。

第十五章:"使用条件伪指令"介绍条件伪指令。

第十六章:"与 *Linker* 的接口"介绍如何包含库和链接时印出哪些符号。

第十七章:"产生列表"介绍 *Turbo Assembler* 的列表文件和如何使用它们。

第十八章："*Turbo Assembler* 与 *Borland* C++的接口"介绍如何联合使用汇编语言和 *Borland* C++。

第十九章："再论与 *Borland* C++接口"通过示例对上一章讨论的技术加以运用。

第二十章："*Turbo Assembler* 与 *Turbo Pascal* 的接口"介绍与 *Turbo Pascal* 代码的接口。

第二十一章："汇编语言 *Windows* 程序示例"介绍用汇编语言编写 *Windows* 程序。

附录 *A*："程序设计蓝图"包含不同类型程序结构的实例。

附录 *B*："*Turbo Assembler* 语法概要"举例说明以修改的巴科斯范式给出的表达式(*Ideal* 和 *MASM* 模式下)。

附录 *C*："兼容性问题"包括 *MASM* 和 *Turbo Assembler MASM* 之间的差异。

附录 *D*："实用程序"主要介绍 *Turbo Assembler* 4.*X* 所附带的实用程序。

附录 *E*："出错信息"介绍当使用 *Turbo Assembler* 时所有的出错信息：消息性信息、致命错误信息和警告、出错信息。

0.4 约定

谈及 *IBM PCs* 或其兼容机时，指使用 8088，8086，80186，80286，386 和 *i*486 芯片的任何机型，涉及 *PC*—*DOS*，*DOS* 或 *MS*—*DOS* 时，指 2.0 版本或更高的版本。

目　录

第一章 入 门

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有杰出才干的人。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—— Intel80x86 系列的全部功能。

仅用汇编语言、C、Turbo Pascal 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。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。

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 InterfaceTechnical Reference》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。

1.1 安装 Turbo Assembler

Turbo Assembler 软件包由一系列执行程序、实用程序和示例程序组成。

关于安装 Turbo Assembler 的指令，请参安装盘上的 TSM_INST.TXT 文件。

Turbo Assembler 执行文件。

Turbo Assembler 4.X 软件包具有三个不同的汇编器，如下所示：

表 1.1 Turbo Assembler 的执行文件

TASM.EXE	实模式汇编器。汇编 16 位和 32 位使用 640K 内存的 DOS 应用程序。只产生 16 位调试信息。
TASMX.EXE	保护模式汇编器。汇编 16 位和 32 位使用多于 640K 内存的应用程序。只产生 16 位调试信息。
TASM32.EXT	保护模式汇编器。汇编 16 位和 32 位使用多于 640K 内存的应用程序。只产生 32 位的调试信息。

如果希望用 Turbo Assembler 代替 MASM 那么可参看第三章了解 Turbo Assembler 与 MASM 区别。

1.1.1 实用程序和执行程序

Turbo Assembler 软件包包含几个实用程序帮助用户建立汇编程序。所有实用程序的列表，请参阅 TSM_INST.TXT 文件。使用实用程序的指令，参阅文本文件 TSM_UTIL

.TXT。

在盘上还有许多示例程序。为了让读者更方便学习，在本书的最后，列出了非常有用的程序，包含 16 位和 32 位 Windows 汇编程序。

1.2 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。

进入正文编辑器，敲入以下行建立名为 HELLO.ASM 的程序。

```
.MODEL SMALL
.STACK 100h
.DATA

TimePromot              DB    'Is it after 12 noon (Y/N)? $'
GoodMorningMessage      DB    13,10,"Good morning, World!',13,10,' $'
GoodAfternoonMessage    DB    13,10,"Good afternoon, World!',13,10,' $'
DefaultMessage          DB    13,10,"Greetings , World!',13,10,' $'

.CODE
start:
  mov     ax,@data
  mov     ds,ax                          ;set DS to point to the data
segment
  mov     dx,OFFSET TimePrompt           ;point to the time prompt
  mov     ah,9                           ;DOS: print string
int       21h                            ;display the time prompt
mov       ah,1                           ;DOS:get character
  int     21h                            ;get a single—character response
  or      al,20h                         ;force character to lower case
  cmp     al,'y'                         ;typed Y for afternoon?
  je      IsAfternoon
  cmp     al.'n'                         ;typed N for morning?
  je      IsMorning
  mov     dx,OFFSET DefaultMessage       ;default greeting
  jmp     DisplayGreeting
IsAfternoon:
  mov     dx,OFFSET GoodAfternoonMessage ;afternoon greeting
  jmp     DisplayGreeting
IsMorning:
  mov     dx,OFFSET GoodMorningMessage   ;before noon greeting
DisplayGreeting:
  mov     ah,9                           ;DOS: print string
```

```
    int     21h                     ;display the appropriate greeting
    mov     ah,4ch                  ;DOS: terminate program
    mov     al,0                    ;return code will be 0
    int     21h                     ;terminate the program
END start
```

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和 C、C++或 Pascal 不同,汇编语言可让用户对计算机编程,以便让计算机做它力所能及的任何事情,为此要输入一些附加行,这往往是值得的。

1.2.1 汇编第一个程序

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,即如下步骤进行:

汇编一个程序的流程如下:

- 汇编源文件 HELLO. ASM;
- 通过汇编生成目标文件 HELLOW. OBJ;
- 通过链接生成可执行文件 HELLO. EXE;
- 运行,以验证程序是否正确;
- 如果不正确,则修改程序,并重新从第 1 步开始执行,直到正确。

上面流程描述了编辑、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 可输入:

```
TASM hello
```

如果不指明其它文件名,HELLO. ASM 将被汇编成 HELLO. OBJ(注意,用户不必输入文件扩展名;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 4.5 Copyright (C) 1992 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。

1.2.2 链接第一个程序

在成功地汇编了 HELLO. ASM 之后,离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。在命令行中输入：

TLINK hello

同样没有必要输入文件扩展名；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。

1.2.3 运行第一个用户程序

现在可以运行第一个程序了。在 DOS 提示符下输入 hello，这时信息：

Hello,World

1 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！

1.2.4 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。

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，当在提示符后输入 hello 时，HELLO.EXE 被执行。

现在可输入：

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。

1.3 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      DB 13,10,'Good morning,World!',13,10,' $'
```

```
GoodAfternoonMessage    DB 13,10,'Good afternoon,World', 13,10,' $'
DefaultMessage          DB 13,10,'Grreetings, World!',10,13,' $'
.CODE
start:
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
cmp     al,'y'                          ;typed lowercase y for after noon?
je      IsAfternoon                     ;yes,it's after noon
cmp     al,'Y'                          ;typed uppercase Y for after noon?
je      IsMoring                        ;no,it's before noon
IsAfternoon:
mov     dx,OFFSET GoodAfternoonMessage  ;point to afternoon greeting
jmp     DiaplayGreeting
IsMoring:
mov     dx,OFFSET GoodMoringMessage     ;pointe to 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 start
```

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。如果输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提示符下输入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,但最好能作检查并明白其原因。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

能出现的错误和警告信息被收集在附录 E 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。

1.3.1 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
Hello Message DB"Hello,World",13,10,12
HELLO_MESSAGE_LENGTH EQU $ -HelloMessage
start:
.CODE
mov     ax,@data
mov     ds,ax
mov     ah,40h
mov     bx,4
mov     cx,HELLO_MESSAGE_LENGTH
mov     dx,OFFSET HelloMessage
int     21h
mov     ah,4ch
int     21h
END start
```

在该“Hello，World”程序中，不是用 DOS 功能在屏幕上打印一个串，而是将该串送往一个选定的设备或文件中——此处指送往打印机。输入并运行该程序，这时打印纸上出现我们所熟悉的“Hello，World”信息（记住，只有保存了该程序才可能运行它。修改过的代码被存入 HELLO.ASM 文件中，源文件版本被刷新）。

可以修改程序，使之将输出结果送往屏幕，重新在屏幕上显示“Hello，World”，而不是在打印机上打印它。为完成这种功能，仅需将：

```
mov bx,4;Printer handle
```

改为：

```
mov bx,1;Standard output handle
```

修改之后，重新汇编并重新链接，最后运行该程序。用户可以看到，当在屏幕上显示输出结果时，被显示的最后一个字符是“女性”的通代符（♀）。事实上，它是一个走纸符。因为屏幕显示与走纸无关，屏幕也处理不了走纸符，所以当程序授意屏幕打印一个走纸符时，PC 字符集