高等院校计算机教育系列教材

Win32汇编语言实用教程

北银科文　冉林仓　编著

01

清华大学出版社

高等院校计算机教育系列教材

Win32 汇编语言实用教程

北银科文　冉林仓　编著

清华大学出版社
北京

内 容 简 介

本书在介绍 Win32 汇编语言指令和基本语法的基础上，重点介绍如何使用汇编语言和 Windows SDK API 开发 Win32 应用程序，同时还探讨了汇编语言和 Visual C++的混合编程、驱动程序的开发、COM 组件的使用和开发、数据库开发、代码优化、异常处理以及程序跟踪调试等问题。

对于每个主题，书中都提供了开发要领及应用的实例和技巧，本书主要面向具备一定汇编语言基础和初步的 Win32 编程经验的用户。

图书在版编目(CIP)数据

Win32 汇编语言实用教程/北银科文，冉林仓编著.—北京：清华大学出版社，2004
ISBN 7-302-07954-4
(高等院校计算机教育系列教材)

Ⅰ.W… Ⅱ.①北…②冉… Ⅲ.汇编语言—程序设计—高等学校—教材 Ⅳ.TP313

中国版本图书馆 CIP 数据核字(2003)第 001608 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：应　勤
文稿编辑：桑任松
封面设计：陈刘源
印 刷 者：北京市人民文学印刷厂
装 订 者：三河市化甲屯小学装订二厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：22.25　**字数**：534 千字
版　　次：2004 年 2 月第 1 版　2004 年 2 月第 1 次印刷
书　　号：ISBN 7-302-07954-4/TP·5777
印　　数：1～5000
定　　价：29.00 元

丛 书 序

人类进入新千年时，计算机科学已经具有一块极其活跃的、崇尚发展与创新的领地，并成为我们这一时代决定性的先驱技术。计算机是现代文化构成不可或缺的组成部分，是现代科学技术发展的先导，并且是世界经济巨量增长的根本引擎。同时，计算机技术的发展日新月异，它的快速发展对计算机科学与工程的教育产生了深刻的影响。因此，计算机教育在世界各国备受重视，计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

伴随着计算机新技术的不断涌现，已有技术几年内即变得陈旧。与此同时，计算机教育也被其所在的专业、文化和社会范围的改变影响着。计算机学科已变得更宽广、内容更丰富，其应用领域不断飞速增长。因此，无论在教学体系、教学内容还是教学方法、教学手段上都必须进行深化改革，与时俱进。CC2001 的出现，无疑是对计算机学科课程体系一个崭新的完整的引导。我们工科院校无论计算机专业还是非计算机专业的计算机科学与工程的教育应该紧密有机地与工科学生的培养目标相结合，注重知识、能力、素质教育三方面的综合教育，加强计算机系统的认知、分析、设计和应用能力，算法设计与分析能力，程序设计能力以及计算思维能力等方面的培养。

原化工部部属高校计算机教育协作组结合工程教育的特点，大力开展计算机教育协作与交流，十年来开展了务实的、全方位的、卓有成效的教学研讨及教研观摩等活动，极大地促进了交流并推动了各校计算机教育的发展。同时，协作组不断地扩展，吸收了许多其他领域的高校参加，共同为我国计算机高等教育事业的发展与完善进行广泛的交流探索。

目前参加这个协作组的主要高校有：

清华大学	南京大学	天津工业大学	北京化工大学
南京工业大学	青岛科技大学	郑州大学	武汉化工学院
沈阳化工学院	南京师范大学	华南理工大学	河北行政学院
南京工程学院	淮海工学院	北京石油化工学院	江苏石油化工学院

在清华大学出版社的大力支持下，本协作组 2001 年年会决定组织出版一套最新的计算机系列教材，第一期出版 11 部有关程序设计与软件应用方面的教材。它们是：《计算机导论》、《C 语言程序设计》、《Visual Basic 语言程序设计》、《Java 程序设计》、《面向对象程序设计——C++》、《SQL Server 数据库原理及应用教程》、《C#编程及应用程序开发教程》、《组网技术与配置》、《现代语音技术基础与应用》、《计算机图形学基础教程》和《Win32 汇编语言实用教程》等。

本系列教材依据 CC2001 框架，精心策划、准确定位，概念清晰，例题丰富，深入浅出，内容翔实，体系合理，重点突出，是一套面向高等学校计算机和非计算机专业学生的

计算机基础与应用系列教材，也可供从事计算机应用和开发的各类人员学习使用。

本系列教材源于十几所全国重点大学和普通高等院校计算机教育的教学改革与实践，凝聚了工作在教学第一线的任课教师的教学经验与研究成果。我们期望本系列教材的出版，并在教学实践中不断完善与更新，为我国高校计算机教育事业做出新的贡献。

编委会

2003 年 9 月

编委会名单

前　言

随着操作系统和商业软件的复杂化，传统的结构化编程方法越来越不能满足人们对软件开发的需要，面向对象、面向组件等编程方法，已经成为软件开发的主流。传统的开发工具已经逐渐淡出软件开发的行列。

汇编语言在DOS时代因为它能够直接存取硬件，有着得天独厚的优势，当时受到几乎所有程序员的重视。但是在Win32体系架构下，使用汇编语言直接开发的用户端应用程序，不能直接存取硬件，所有的硬件存取必须通过系统提供或者用户开发的驱动程序实现，从而使汇编语言不再具有至高无上的特权，再加上汇编语言可读性差、资料少等问题，使其一度受到人们的冷落。

尽管汇编语言存在上述种种缺点，但其优点还是显而易见的。汇编代码在程序优化、代码调试、解密加密、系统维护等方面，其优势还是其他语言所无法企及的，目前，可以使用汇编语言配合其他语言实现混合编程，从而使Win32汇编语言的优势可以得到充分发挥。

本书共分为17章，各章的主要内容分别介绍如下：

第1章回答了为什么要使用汇编语言、什么情况下使用汇编语言两个基本问题，接着介绍了32位汇编语言的特征，Win32汇编环境的安装和设置、通过一个应用程序的创建以及和C++程序的比较，旨在帮助读者快速进入Win32 汇编世界。

第2章和第3章主要解决汇编语言的语法问题，首先回顾了常用的80x86汇编指令、然后介绍MASM32自带的辅助库函数、分支、循环以及宏的使用。

第4章和第5章介绍了如何使用汇编语言创建窗口应用程序，如何使用窗口程序中的各种类型的资源。

第6章和第7章主要介绍汇编语言在系统开发方面的应用，包括动态链接库的开发和使用、钩子函数、文件管理、内存管理、管道、进程、线程、事件同步等。

第8章到第10章主要介绍COM对象的使用和创建、Windows NT服务创建的框架和安装实现、使用ODBC实现数据库的存取。

第11章和第12章介绍Windows NT虚拟设备驱动程序和基于Windows 9x内核的虚拟设备驱动程序开发的相关步骤及注意事项。

第13章介绍如何实现汇编语言和Visual C++的混合语言编程。

第14章介绍Win32本机应用程序的存储格式。

第15章到第17章主要介绍代码优化、程序调试、异常处理方面的内容。

本书所介绍的内容力图全面，旨在帮助读者能够熟练地驾驭Win32汇编语言。

本书主要由冉林仓编著，另外，尹建民、薛年喜、刘伟、徐日强、赵磊、张江涛、李志伟、李士良、刘旭、宋利军、刘咏、郑砚、张海霞、范翠丽、冯冬梅、向登宁、王军茹、李东玉、周松建等也参加了部分内容的编写，在此一并表示感谢。

由于时间仓促，作者水平有限，不妥之处还希望读者朋友给予批评指正。

作者

2003 年 9 月

目　录

第 1 章　快速进入 Win32 汇编世界

本章主要介绍使用汇编语言的原因和场合、32 位汇编语言的特点、汇编语言环境的安装和设置、汇编语言与 C++应用程序的比较以及汇编语言程序的调试。

1.1　使用汇编语言的意义

在以前的 DOS 年代，汇编语言因为其小巧灵活、可以直接存取端口和使用中断，曾经一度受到广大编程爱好者的青睐。然而 Windows 的普及让 DOS 彻底退出了历史舞台。DOS 的急剧而下使得汇编语言失去了“用武之地”，而汇编语言对 Windows 复杂的体系结构复杂应用的支持越来越力不从心，再加上新生代的程序员热衷于追捧一些可视化的快速应用开发工具，这样这种结构化语言类型的底层开发工具就被人们扔进了遗忘的角落，其优势地位逐渐被面向对象的 C++语言取代。

造成汇编语言低迷的原因除了它艰深晦涩、难以移植、不支持面向对象之外，另外一个重要的原因是，在 Windows 特别是 Win32 环境下，汇编语言和其他语言一样不再具有至高无上的“特权”，Windows 系统对应用程序进行了权限划分，操作系统底层以及驱动程序运行在 0 级，即 Ring0 环；而其他 Win32 应用程序则运行在 Ring3 环，后者不能运行操作系统限定的特权指令，包括对一些重要端口的直接存取操作。否则就会引发程序异常。汇编语言尽管提供了各种端口存取的指令，也必须遵守这个约定。这样以前汇编语言具有的语言优势将不复存在。

不过，汇编语言也不是一无是处，了解和学习汇编语言可以获得高级语言所不具备的灵活性，可以跳出语言限定的各种框框。尽管汇编语言的开发效率无法和 C++及其他快速应用开发工具相提并论，但是其程序的代码短小精悍、运行高效却是其他语言无法比拟的。在一些对运行速度要求比较高的场合仍然是汇编的天下，比如 C++中的很多字符串操作函数都是采用汇编语言编写，因为这些函数采用汇编语言编写具有代码小、运行速度快的优势，频繁地调用这些函数，其程序的总体运行效率相当可观。比如用户需要开发一个每秒上亿级访问量的搜索引擎，显然，千方百计提高检索速度，是用户重点考虑的问题所在。或者用户需要开发一个国家电网监控系统，这个系统的响应速度是设计的关键，因为每秒钟的响应滞后都可能给国家造成上百万的财产损失。在这种情况下，汇编语言有着其当仁不让的优势。用户可以不用它来设计程序界面，但是完全可以利用它来设计优化算法或者编写响应代码。

另外用汇编语言编写 Win32 应用程序并不是从最底层的 80x86 指令开始的，它可以和 C 语言一样直接使用 Win32 SDK 编程，可以直接调用 Windows 系统提供的 API 函数，可以说，任何一个采用 Win32 SDK 开发的 C++应用程序，都可以很容易地转换成汇编语言。也就是说汇编语言在 Win32 开发中具有它的一席之地。

1.2 汇编语言的使用场合

用户不能简单地判断汇编语言和 C++等开发工具的谁长谁短，因为各种语言都有它自身的长处，有些语言易用性比较强，但是它的灵活性可能就比较差；有些语言开发速度比较快，但是运行效率可能就差一些。但是这些语言之间可以优势互补。可以通过汇编语言和高级语言混合编程的方法，取长补短，更好地解决实际问题。

一般地，汇编语言主要应用在下列场合：

- 代码要求占用较小的存储空间，而且要求具有很快的响应速度，比如操作系统核心代码实现、嵌入设备的监控程序、要求实时性比较高的控制软件。
- 需要和硬件打交道的代码，比如编写端口操作的底层驱动程序。
- 出于提高性能考虑，需要对代码进行优化的代码。比如频繁调用的子程序、软件解压算法等。
- 为了支持最新的处理器指令，高级语言尚未针对该处理器进行优化的情况。因为软件的推出总是滞后于硬件，用户通过汇编语言不必等待针对该硬件优化的高级语言函数库出现。
- 用于加密、解密、计算机病毒分析预防等安全领域。

而下面的场合不适合使用汇编语言开发：

- 运行速度和存储空间没有限制，但是对开发速度要求较高的场合。
- 跨平台运行，随时需要向另外一个环境进行移植的场合。
- 利用软件工程进行团队开发的主要开发平台。
- 刚刚学习 Windows 编程的初学者。

总之，完全使用汇编语言编程的情况越来越少，一般地，用户应该尽可能地使用 C 或者 C++来代替汇编语言，对于特殊的情况可以借助于混合编程来解决。

本书假定用户已经具备 32 位 Windows API 编程的经验，并且从事过汇编语言编程。初学者应该从 C/C++ 、Delphi 或者 Visual Basic 学起，以便积累寄存器、数据类型、数据大小、系统 API 调用、调用规范的概念的经验，为使用汇编语言打好基础。

1.3 32 位汇编语言的简单介绍

编写汇编语言是一个复杂枯燥的过程，但是这种语言却能够提供高级语言所不具有的性能。它可以作为高级语言的有益补充，高级语言具有优雅高效的编程方式，但是当项目的规模越来越大时，可以使用汇编语言编写一些适当的模块来对项目进行补充。

汇编语言在代码编写上相对是比较自由的，一方面它可以用于编写结构化模块代码；另一方面它还可以编写没有约束的自由风格代码。二者都有其自身的优点。前者有利于对代码进行组织，尤其是那些规模较大的项目，而后者便于对程序进行优化。

汇编语言尽管代码短小但是却能提供优越的性能，汇编语言一方面利用了高级语言才有的代码重复使用技术，同时在必要的时候，它还可以提高关键代码的运行速度。

32 位汇编语言要比 DOS 和 16 位 Windows 代码要简单清晰得多，特别是它避免了段重新定位的复杂性，不再使用段寄存器进行寻址。同时用户也不再需要纠缠于 AX:DX 寄存器对来访问长整型变量。也不再受限于 16 位应用软件段结构所施加的 64 KB 边界。

编写 32 位的 Windows 软件难点在于 Windows 复杂的结构和它极其复杂的 API 函数集。它与 DOS 代码最大的不同在于它的参数传递是采用堆栈而不是 DOS 中断的寄存器。

使用汇编语言另外一个优势在于它能够很方便地处理“C”风格的 Windows API 函数，包括以零结尾的字符串、结构、指针和数据大小等都可以用汇编语言进行处理。

1.4　安装和设置汇编语言环境

MASM32 是一个免费软件，它可以被任何热衷于编写汇编语言的程序员自由使用，但是 MASM32 包中的所有软件都受国际版权法保护，它既不能被买卖也不能被包含在其他任何商业性质的软件包中。

在使用汇编语言之前，用户首先需要从 MASM 的主页 http://www.masm32.com/下载最新的 MASM V7.0 版本的软件包。然后把它解压缩到一个临时目录，再运行 install.exe 进行安装。安装的界面如图 1.1 所示。

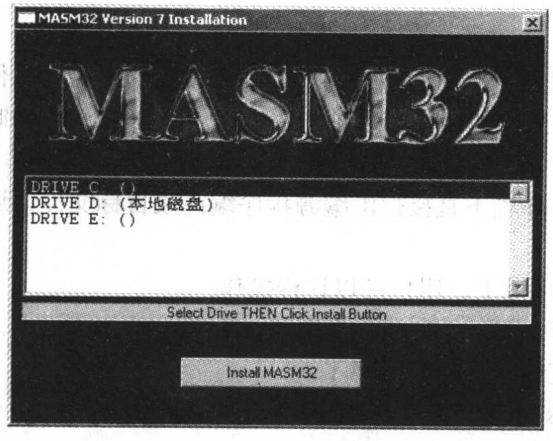

图 1.1　MASM 的安装

程序安装后，用户应该对安装目录有所熟悉，特别是要首先阅读该目录下的 HTML 和 HELP 目录下的内容，这些内容对用户尽快熟悉 MASM32 汇编语言很有帮助。另外在这个目录下还有大量例子程序和向导程序，阅读代码并尝试对这些例子程序进行编译链接，都是很好的快速入门途径。

MASM32 提供了一个集成开发环境，但是这个集成开发环境无法和 Visual Studio 相提并论，它只是一个比较简单的文本编辑器外加一个可定制的菜单，通过配置，用户可以通过菜单来执行一些 Shell 命令，对程序代码进行编译和链接。

另外用户也可以在命令行进行程序的编译和链接。汇编代码的编译和链接主要依赖 Bin 目录的汇编编译器 ml.exe、资源编辑器 rc.exe 和链接器 Link.exe 等执行文件。用户可以手工运行这些程序进行编译，也可以建立一个 Makefile 文件，使用 nmake.exe 进行编译链接。不过 MASM32 软件包并没有提供 nmake.exe 文件，用户需要手工从 Visual Studio 安装目录

下 bin 子目录复制这个文件到 masm32\bin 目录。

另外为了在任何目录下都能够执行编译链接程序，用户最好建立一个设置环境变量的批处理文件。下面是一个典型的批处理文件。它指出了包含文件、库文件的搜索目录，同时它还给出了程序文件的查找路径。

```
@Echo off
set include=e:\masm32\include
set lib=e:\masm32\lib;e:\masm32\m32lib; D:\Program Files\Microsoft SDK\lib
set path=e:\masm32\bin;e:\masm32;%path%; D:\Program Files\Microsoft SDK\bin
start cmd.exe /k
@echo on
```

如果用户把 MASM32 安装到其他驱动器，可以把 e:换成安装的驱动器。

如果用户想了解 makefile 的编写，可以参考“masm32/ICZTUTES”目录下各个例子程序中的 makefile 文件。下面是一个典型的 makefile 文件。

```
NAME=dialog
$(NAME).exe: $(NAME).obj $(NAME).res
Link /SUBSYSTEM:WINDOWS /VERSION:4.0 /LIBPATH:c:\masm32\lib $(NAME).obj
$(NAME).res
$(NAME).res: $(NAME).rc
       rc $(NAME).rc
$(NAME).obj: $(NAME).asm
       ml /c /coff /Cp $(NAME).asm
```

这个 makefile 配置文件指出了项目的名称，编译后的可执行文件需要链接的目标文件和资源文件、以及编译采用的命令和选项，用户在创建 makefile 的时候可以利用这些现有的文件为模板。

如果在 Visual C++环境下直接把汇编源程序编译成目标文件，还需要对 Visual C++进行设置。

在 Visual C++ 6.0 环境下，用户可以这样操作：

(1) 在菜单中选择 Project | Setting 命令。

(2) 选中指定的汇编文件(单击即可)。

(3) 选中Custom Build页。

(4) 在Commands文本框中进行输入。

如果是 DEBUG 模式，则输入：

```
path e:\masm32\bin
ml /c /coff /Zi /FoDEBUG\$(InputName).obj $(InputPath)
```

如果是 RELEASE 模式，则输入：

```
path e:\masm32\bin
ml /c /coff /FoRELEASE\$(InputName).obj $(InputPath)
```

(5) 在 Outputs 文本框中输入。

如果是DEBUG模式，则输入：

```
DEBUG\$(InputName).obj
```

如果是 RELEASE 模式，则输入：

```
RELEASE\$(InputName).obj
```

如果用户没有把 MASM32 安装在 E 盘，则要作相应的修改。以上的设置情况如图 1.2 所示。

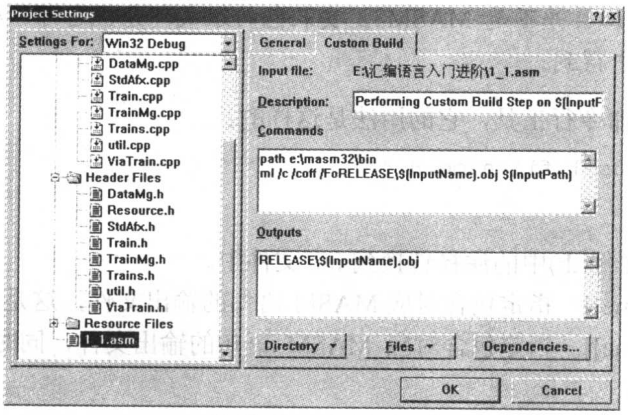

图 1.2　Visual C++ 6 环境设置

对于 Visual Studio .NET 环境，设置方法是一样的。步骤如下：

(1) 选中指定的汇编文件(单击即可)。

(2) 右击鼠标，从弹出的快捷菜单中选择【属性】命令。

(3) 从【配置属性】中选择【自定义生成步骤】节点，然后根据项目生成的模式定义编译命令行和输出的目标文件，如图 1.3 所示。

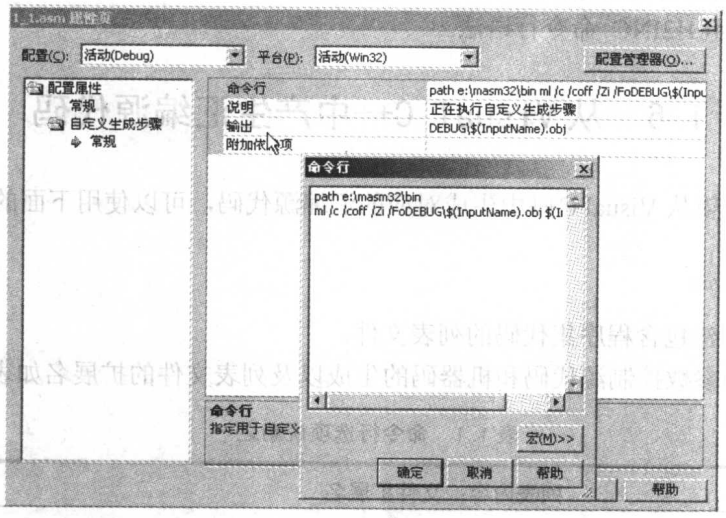

图 1.3　Visual Studio .NET 环境设置

1.5　H2INC 工具的使用

Windows SDK 提供的函数都是采用 C 或者 C++的风格进行声明的，这对于汇编开发者来说，使用这些函数的声明是不方便的，手工翻译这些头文件也是一个烦琐的而且容易出错的过程。好在微软为汇编程序员提供了一个转换工具 H2INC，通过这个转换工具，能够

把 C 风格的头文件转换成 MASM 兼容的包含文件(.inc)，这个工具只能转换常量、结构和函数声明，无法转换 C 代码。

注意：H2INC 工具并不是 MASM32 安装包的组成部分，它可以从微软提供的 SDK 开发包中得到。

H2INC 是一个命令行工具，它的用法是这样的：

```
H2INC [[options]] filename.H
```

参数列表如下：

- /C 把头文件(.h)中的注释转换到 inc 文件中。
- /Fa[[filename]] 指定包含对应 MASM 语句的输出文件，这是一个默认选项。
- /Fc[[filename]] 指定包含对应 MASM 语句的输出文件，同时包含了被注释掉了的 C 代码。
- /HELP 调用 QuickHelp 显示 H2INC 的帮助。
- /Ht 允许生成对应的文本，默认情况下文本不被转换。
- /Mn 告诉 H2INC 明确声明所有指针和函数的调用方式。
- /Ni 禁止嵌套头文件的展开。
- /Zn string 把 string 添加到 H2INC 产生的所有名称前面，以此消除生成的该文件与其他 H2INC 产生的包含文件造成的命名冲突。
- /Zu 强制所有的结构和联合命名标签惟一。
- /? 显示 H2INC 命令行语法。

1.6 从 Visual C++中产生汇编源代码

如果用户希望从 Visual C++中生成对应的汇编源代码，可以使用下面的命令行选项：

```
/FA[c|s|cs]
/Fa pathname
```

/FA 选项创建包含程序集代码的列表文件。

此选项中的参数控制源代码和机器码的生成以及列表文件的扩展名如表 1.1 所列。

表 1.1 命令行选项说明之一

选 项	列表内容；文件扩展名
/FA	程序集代码；.asm
/FAc	机器码和程序集代码；.cod
/FAs	源代码和程序集代码；.asm
/FAcs	机器码、源代码和程序集码；.cod

默认情况下，列表文件获取与源文件相同的基名称。使用 /Fa 选项可以更改列表文件的名称和在其中创建列表文件的目录，如表 1.2 所列。

表 1.2　命令行选项说明之二

/Fa 用法	结　果
/Fa	为编译中的每个源代码文件创建一个源文件.asm
/Fa 文件名	将文件名.asm 放到当前目录中。仅在编译单个源代码文件时有效
/Fa 文件名.扩展名	将文件名.扩展名放到当前目录中。仅在编译单个源代码文件时有效
/Fa 目录\	为编译中的每个源代码文件创建一个源文件.asm，并将其放到指定目录中。请注意必须有后缀反斜杠。只允许使用当前磁盘上的路径

在 Visual C++ 6.0 及 Visual Studio .NET 开发环境中设置此编译器选项步骤如下：

(1) 打开此项目的【属性页】对话框(或 Project Settings 对话框)。

(2) 单击 C/C++文件夹(或 C/C++选项卡)。

(3) 单击【输出文件】属性页。

(4) 修改【ASM 列表位置】(/Fa) 或【汇编输出】(/FA) 属性(参考图 1.4 和图 1.5)。

示例

下列命令行产生名为 HELLO.cod 的组合源代码和机器码列表：

```
CL /FAcs HELLO.CPP
```

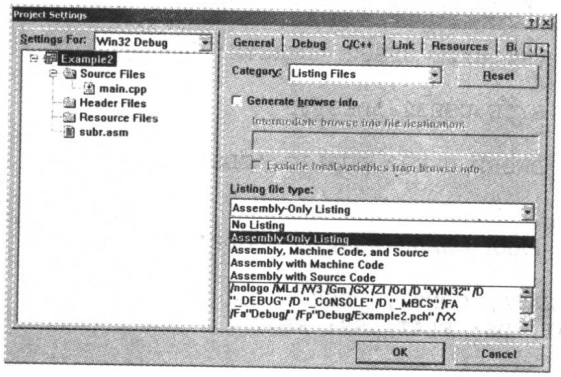

图 1.4　Visual C++6 环境中的编译器选项设置

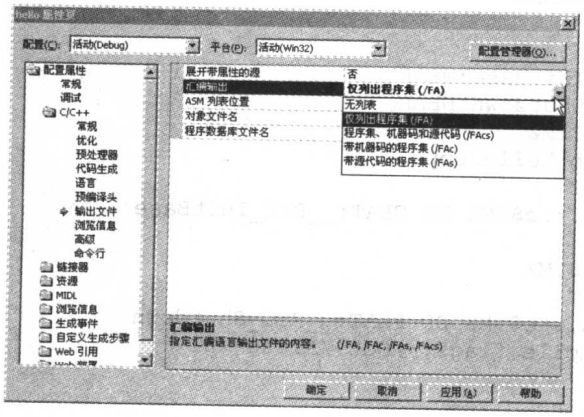

图 1.5　Visual C++.NET 环境中的编译器选项设置