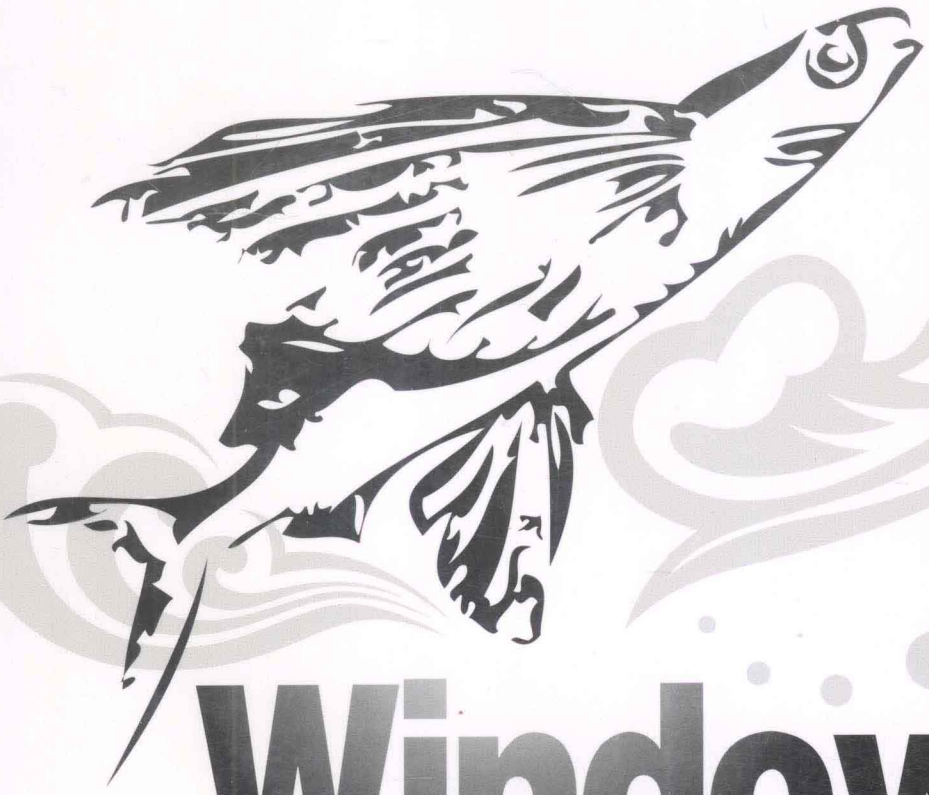
Broadview
www.broadview.com.cn

畅销10年，经典再现！



典藏版

Windows环境下32位汇编语言程序设计

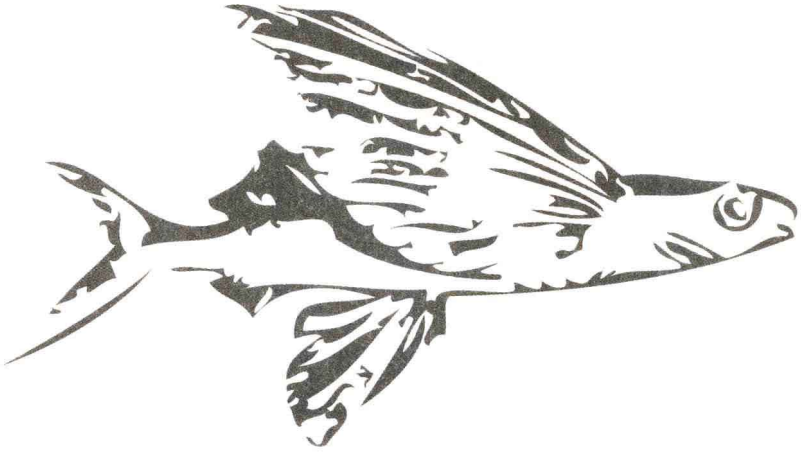
罗云彬 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

典藏版

Windows环境下32位汇编语言程序设计

罗云彬 著



電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

Windows 环境下 32 位汇编语言是一种全新的编程语言。它使用与 C++语言相同的 API 接口，不仅可以开发出大型的软件，而且是了解操作系统运行细节的最佳方式。

本书从编写应用程序的角度，从“Hello，World!”这个简单的例子开始到编写多线程、注册表和网络通信等复杂的程序，通过 70 多个实例逐步深入 Windows 环境下 32 位汇编语言编程的方方面面。

作者罗云彬拥有 10 余年汇编语言编程经验，本书是作者多年来编程工作的总结，适合于欲通过 Windows 环境下 32 汇编语言编写 Windows 程序的读者。

图书在版编目（CIP）数据

Windows 环境下 32 位汇编语言程序设计：典藏版 / 罗云彬著. —北京：电子工业出版社，2013.8
ISBN 978-7-121-20759-4

Ⅰ. ①W… Ⅱ. ①罗… Ⅲ. ①汇编语言—程序设计 Ⅳ. ①TP313

中国版本图书馆 CIP 数据核字（2013）第 137514 号

责任编辑：李 冰
印 刷：北京中新伟业印刷有限公司
装 订：三河市皇庄路通装订厂
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1092 1/16 印张：47.25 字数：1210 千字
印 次：2013 年 8 月第 1 次印刷
印 数：3000 册 定价：99.00 元（含光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

从 Windows 出现开始，汇编语言似乎在慢慢地销声匿迹，但本书可以让人放弃这个观点，其实在 Win32 环境下，汇编语言依然强大。

Why——为什么选择 Win32 汇编

选择 Win32 汇编的理由是什么呢？

在 DOS 时代，学习汇编就是学习系统底层编程的代名词，若要成为一名入门级的汇编程序员，就需要学习从 CPU 结构、CPU 工作方式、各种硬件的编程方法到 DOS 工作方式等范围很广的知识。随着 Windows 时代的到来，Windows 像一堵巨大的墙，把我们和计算机的硬件隔离开。对于 DOS 的汇编程序员来说，就像在一夜之间，我们发现自己曾经学过的几乎所有的东西都被 Windows 封装到内核中去了，由于保护模式的存在，我们又无法像在 DOS 下那样闯入系统内核为所欲为。在 Windows 下用任何语言编程都必须遵循 Windows 的规范，汇编也不例外，也就是说，汇编不再是一种“有特权”的语言。面对汹涌而来的 Visual C++，Visual Basic，PowerBuilder 和 Java 等各个领域的猛将，从 DOS 时代“为所欲为”的“系统警察”岗位下岗，在其他领域又没有一技之长，汇编语言似乎失去了生存的意义，有很多人在 DOS 转向 Windows 的时候放弃了汇编语言。

但是经过短暂的失落，摆正了自己在系统中的位置，我们发现从“系统警察”转换到遵循 Windows 规范的“好市民”后，汇编语言又慢慢地在这个世界流行起来了。毕竟，不能为所欲为也可以有好的一面，我们可以不必再考虑一些老大难的问题，如程序运行时会面对什么样的显示卡，如何驱动不同的打印机，内存不够了如何用磁盘交换，等等。我们也可以在了解更少硬件知识的情况下就可以掌握 Win32 的汇编编程。而且，我们惊喜地发现，做了“好市民”以后，我们反而拥有了和其他语言同样的权利——为了做图形和界面等方面的功能，汇编程序员在 DOS 时代连做梦都在羡慕 C 语言庞大的函数库，而现在，Windows 为我们提供了比这还要多得多的函数，以至于其他大部分语言可以做出来的功能，汇编都可以做，而其他语言做不到的功能，汇编照样可以做！所以这就是理由之一：**Win32 汇编可以当做一种功能强大的开发语言使用，使用它完全可以开发出大型的软件来。**

正因为 Win32 汇编看上去不再那样低级，于是有读者曾经提出：Win32 汇编讲的都是用 API 来写程序，和高级语言差不多，以前在 DOS 下使用的中断什么的都不能用，所以没有什么新奇的了。还有读者认为本书只不过是 MSDN 的汇编版本而已。言下之意就是：学汇编就是为了了解高级语言底下一层的功能，但现在 Win32 汇编却使用和 C++等语言相同的 API 接口，既然和高级语言处于同一个级别，我们为什么还要去和机器指令打交道呢，还不如去学 Visual C++方便。

但是我们可以这样问一问自己：

问：在 DOS 汇编中我们为什么用中断功能？

答：为了使用 DOS 内核提供的功能。

问：在 DOS 中我们常常自己用操作 I/O 端口的方法读写硬盘或操作显卡吗？

答：不，我们用系统提供的 int 13h 和 int 10h。

……

同样，在 Win32 汇编里使用 API 也是为了使用 Windows 内核提供的功能。只不过使用的方式不再是中断方式而已，这不是 Win32 汇编语言“高级化”了，而是高级语言因为使用 Windows 的 API 接口而“低级化”了，其代价就是无法移植到其他系统，用 Visual C++写的程序是无法移植到其他操作系统平台上的，只有和平台无关的 ANSI C++等才能算是真正意义上的高级语言。

其实，任何汇编语言都是和操作系统密切相关的，不管是 DOS 汇编、Win32 汇编，还是 Linux 汇编，都是基于特定的操作系统的，如果一定要绕过操作系统，那么就不会有 DOS 汇编和 Win32 汇编的区别了，但是这样的话我们不是在学汇编，而是在自己开发操作系统。高级语言在不同的操作系统上看起来都差不多，但作为一种低级语言，不同操作系统上的汇编就是不同的世界。所以，既然 Windows 和 DOS 是两个完全不同的操作系统，我们就必须抛弃 DOS 汇编中的大部分概念从头开始学习 Win32 汇编。这就是理由之二：**Win32 汇编是 Windows 环境下一种全新的编程语言。**

Win32 环境下的很多高级语言，如 Visual C++和 Visual Basic 等，一如既往地对实现的细节进行了或深或浅的封装，就连最能表现 Windows 特征的部分，如消息循环和多线程的处理等内容也都被隐藏封装，使我们在使用它们进行可视化编程的同时，无法全面了解 Win32 程序运行的具体方式。在学习 Win32 汇编以后，这些隐藏在高级语言后面的细节就暴露出来了。

由于封装的关系，各种高级语言或多或少存在某种“缺陷”，比如 VB 不支持指针，结果很多需要使用指针的 API 用起来就很不方便，像多线程一类的特征在 VB 中就无法实现，PowerBuilder 也是如此；C 语言已经是最灵活的高级语言了，但还是无法在代码级别处理某些需求；而汇编语言见到的是一个最真实的操作系统，它可以用最灵活的方式使用各种系统功能，第 13 章中有关进程隐藏的内容就是最好的写照。所以理由之三就是：**使用 Win32 汇编语言是了解操作系统运行细节的最佳方式。**

最后的理由根本不是理由，而是必然的选择，当我们在 Windows 环境下进行加密解密、逆向工程，还有病毒、木马等有害代码的分析和防治工作时，Win32 汇编是唯一的选择。在任何讨论这方面内容的书籍中，汇编代码的篇幅总是很大的。因此，深入了解这些内容的前提就是深入汇编编程。

How——如何学习 Win32 汇编

以往的汇编书籍往往把重点放在硬件结构和指令上，讲述了一大堆电路框图和指令列

表，把大家搞得晕晕乎乎后，再举出一些重量级的例子，不是一些像数组、矩阵计算一类的复杂运算，就是开始图形模式画图，以至于大家看完以后就再也找不到北了！实际上，这些例子不是太难，而是太枯燥了。有人说，学汇编就像考大学，千军万马过独木桥，太多的人中途放弃了，只有少数人坚持到最后。

笔者认为：学习汇编应该在轻松的环境下进行，在学习中使用的例子不一定太复杂，但一定要有吸引力。用汇编写复杂的运算程序固然会比 C 更有效率，但同样的事在 C 中用一个表达式就全部搞定了，从这里开始学汇编，给人的感觉就像从复杂的公式开始学算术，要知道，加法还没有学会呢！而对于高级语言封装起来的系统功能，用汇编解释起来就非常直接，非常自然，也更容易懂。以笔者自己学汇编的过程来说，那时候是 1990 年，刚好是中国第一次病毒大流行，大家的计算机上都是那个病毒的开山鼻祖——乒乓病毒，在流行 DOS 的时期，看着在屏幕上蹦的小球，心中就有一个问题：如何编出这样一个玩意来呢？要知道 DOS 是单任务的，而那个球在别的程序运行的时候照样蹦！这用当时流行的 FORTRAN、C 等课程中学到的任何知识都无法解释，因为这些课程中不可能有 TSR、中断、引导区等内容。带着这样一个疑问学习汇编，在分析乒乓病毒的过程中啃一条条不懂的指令，病毒分析完了，汇编课也学完了，而且反过来看那些复杂的计算程序都是那么顺理成章，不攻自破了。实际上，从一些实用的系统功能开始学习汇编远比学矩阵计算容易理解。

正如最经典的 C 程序就是那个“Hello，World!”一样，这个程序的有名并不是因为它用高深复杂的语句放倒了一大批人，而是它以最简单易懂的方式让人们走入 C 语言的大门。对于 Win32 汇编也是如此，从最简单的例子开始总是没错的，笔者建议读者跟随本书中从简到繁的例子，努力做到理解并灵活引用这些例子中的各种功能，正如“熟读唐诗三百首，不会写诗也会吟”，最后能够熟练地使用 Win32 汇编来解决各种编程需求就是最大的胜利。

另外，正如前面讲到的，汇编语言的学习必须和操作系统紧密结合。经过简单的调查，笔者发现很多高校使用的汇编教程还是停留在清华 91 版《IBM-PC 汇编语言程序设计》之类的教材上，虽然这些教材中基础知识部分永远不会过时，但涉及操作系统的部分还是停留在 DOS 阶段。随着 DOS 操作系统的悄然引退，继续把精力花在上面是一种浪费，因为任何语言都必须有应用的平台，否则课程学完之后会尴尬地发现没有地方可以应用。笔者认为，在《IBM-PC 汇编语言程序设计》之类传统教材中的基础部分学习完毕以后，重点就应该转向 Win32 汇编，以及保护模式方面的知识。

关于本书的内容

本书尝试从编写应用程序的角度，从“Hello，World”这个简单的例子开始到编写多线程、注册表和网络通信等复杂的程序，通过 70 多个从简单到复杂的例子，逐步深入 Win32 汇编编程的方方面面。笔者从事汇编编程已经有十几年的历史了，从 8086 时代的 DOS 汇编编程开始到当前的 Win32 汇编编程，从一个初学者到现在能利用 Win32 汇编来解决大部

分编程需求，中间也经过了很长时间的摸索和大量的挫折，所以笔者很清楚初学者在哪些地方会遇到问题，但是涉及 Win32 汇编的书籍却实在太少了。正是因为如此，笔者决心把本书的目标定为：能让读者入门并在最后能熟练掌握 Win32 汇编编程，而不是那种深入系统奥秘一类的书籍。

从这个目标出发，本书的选材中尽量去掉已经有其他书籍详细讨论的部分，因为要一本书涉及全部方面是不现实的。内容全面就必然不精，内容深刻就必须围绕一个中心点，所以本书的内容并不详细讨论一般汇编教材的基础部分，如处理器结构和保护模式等，也不准备涉及 Windows 驱动程序、COM 编程或者其他能够冠以“密技”头衔的内容。本书主要的内容将放在 32 位宏汇编对比 DOS 汇编所不同的部分，以及 Win32 应用程序的汇编实现上。不求全面，只求精也！（说句老实话，也不敢对自己不精通的地方妄加评论，以免破坏自己的良好形象。☺ ）

在一些汇编编程论坛上，经常有初学者问到 MASM 和 TASM 有什么不同，用哪个比较好，@@标号是什么意思，为什么用下载的汇编编译器无法编译程序等问题，虽然这些都属于最基本的问题，但是以前的确没有一个地方或者有一本书能系统全面地讲解这些问题。本书的**基础篇**就是因此而设的，它们是：

- 第 1 章　背景知识
- 第 2 章　准备编程环境
- 第 3 章　使用 MASM

当搭建编译环境和对编译器的使用不再成为绊脚石的时候，初学者的问题往往集中在对 Windows 程序结构的迷惑上，消息驱动体系、窗口过程、与硬件隔绝的图形接口及资源文件等相对于 DOS 程序来说都是全新的内容。接下来的 4 章将深入讨论这些内容，通过这几章，读者应该开始习惯以 Windows 的方式考虑问题了（脑海中的 DOS 逐渐远去……），这就是本书的**初级篇**：

- 第 4 章　第一个窗口程序
- 第 5 章　使用资源
- 第 6 章　定时器和 Windows 时间
- 第 7 章　图形操作

Windows 系统不像 DOS 系统，它的应用程序界面是规范化的，统一的界面来自大量统一的界面控件，学习这些控件就等于学习如何编写 Windows 界面。下面的**界面篇**中的两章将探讨这方面的内容：

- 第 8 章　通用对话框
- 第 9 章　通用控件

学到这里为止，读者应该可以写出界面规范的标准的 Win32 程序了。但还是无法用这些程序来解决一些具体问题，因为有关 Windows 系统的高级特征的介绍还没有开始，如内

存管理、文件操作和多线程等。这些就是本书**系统篇**中将要介绍的内容，通过这些内容，读者将比较深入地了解 Windows 的工作方式：

- 第 10 章　内存管理和文件操作
- 第 11 章　动态链接库和钩子
- 第 12 章　多线程
- 第 13 章　进程控制
- 第 14 章　异常处理

相信到这里为止，读者对 Windows 的了解已经比较系统了。虽然 Windows 中还存在其他很多方面的内容，如管道，邮件槽，如何写控制面板程序、屏幕保护程序和驱动程序等。但是有了前面的基础以后，读者自己去了解这些内容就不成问题了，因为掌握了“渔”，得到“鱼”又有什么困难呢？在最后的几章中，本书将从应用的角度再补充介绍一些常用的网络编程、注册表、PE 文件和数据库操作方面的内容，这就是**应用篇**：

- 第 15 章　注册表和 INI 文件
- 第 16 章　WinSock 接口和网络编程
- 第 17 章　PE 文件
- 第 18 章　ODBC 数据库编程

在本书中，笔者特别以显著的方式标出了一些经验之谈，这些是笔者在长期的汇编编程中得到的体会，可能是任何一本教科书或者手册里都没有的。希望这些能给读者带来帮助！


用“灯泡”标出的部分表示一些小技巧，对编程的理解有促进作用。


用“惊叹号”标出的部分表示容易出错的部分，可以帮助读者避免一些难以理解的错误。

对读者的假设

有了内容的定位，读者的定位也就比较清楚了，本书适合于以下读者：

- 想用 Win32 汇编写 Windows 应用程序的读者。
- 想从 DOS 下的 16 位汇编转向 Windows 下 32 位汇编的读者。
- 欲了解 Win32 汇编，以便为 Windows 下的加密解密、系统安全、逆向工程等方面打基础的读者。
- 欲了解 Win32 汇编，以便为用汇编写 Windows 驱动程序打基础的读者。
- 正在学习汇编课程，需要补充汇编课程中 Win32 部分的学生。

在开始学习本书之前，读者应该有以下的基础知识：

- 计算机的基础知识，如进制转换、逻辑运算、变量类型和指针的概念等。
- 数据结构的基础知识，因为 Win32 编程涉及大量的数据结构。
- C 语言的基础知识，因为 Win32 编程的绝大部分参考资料都是以 C 的格式出现的。
- Intel 80x86 处理器的基础知识，如寻址方式和指令的使用等。

本书并不是为以下读者准备的：

- 欲详细了解保护模式的读者——因为 Windows 并不是一个开放的平台，Windows 的开放只限于应用程序接口，所以要用 Windows 做背景研究保护模式只能是自讨苦吃，如果读者需要深入了解这方面的内容，最好的方法就是去研究 Linux 的核心代码并在 Linux 上实验。
- 欲了解 Windows 核心“机密”的读者——汇编并不等同于深入操作系统的内部，所以本书不是《Windows 内核分析》。而真正意义上的《Windows 内核分析》除了 Microsoft，恐怕谁也写不出来。
- 欲了解 Windows 驱动程序编写的读者——要介绍清楚 Windows 驱动程序，需要的篇幅绝不会亚于本书的篇幅，本书不打算涉及这方面的内容，读者有兴趣的话，可以阅读《Programming WDM》和《System Programming for Windows 95》等书，前者讲述的是 Windows 2000/NT 下的 WDM 驱动程序，后者讲述的是 Windows 9x 下的 VxD 驱动程序。

典藏版有什么新的内容

本书第 1 版出版至今已经 10 年多了，第 3 版出版至今也已经 3 年多了，期间笔者收到了大量的读者来信，对本书提出了各种意见和建议，综合各方面的考虑，典藏版做了以下改进。

- 对第 3 版中已知的错误进行了修正，包括一些排版错误、错别字和例子中的 Bug。
- 对一些过时的内容进行了更新或删除。
- 根据读者的反馈，对部分章节进行了重写。

关于附书代码和读者反馈

为了更好地说明 Win32 汇编的编程方法，本书附带了 70 多个例子，这些例子的源代码全部可以在附书光盘中找到，代码全部采用 MASM 格式编写，推荐使用的编译软件为 MASM32 SDK 软件包。

MASM32 SDK 软件包可以在官方网站下载：http://www.masm32.com

本书中的例子代码已经经过了严格的防病毒测试，绝对不含任何病毒，但第 11 章的例子涉及钩子技术，第 17 章的例子涉及对 PE 文件进行操作，其中的小段代码与一些木马和病毒的特征码类似，以至于被一些杀毒软件误认为有未知病毒，请读者放心使用，不必顾虑。

虽然本书中所有的例子代码都已经在 32 位的 Windows 98、Windows 2000、Windows XP、Windows Vista 和 Windows 7 下测试通过，但也有存在 Bug 的可能，如果发现代码存在错误或者发现书中有其他问题，请告知作者，以便在下一个版本中改进。如果读者有任何的反馈意见——不管是批评还是鼓励都请和作者联系，作者的 E-mail 是 luoyunbin@hotmail.com。

2013 年 5 月

致　　谢

首先感谢我的父母亲，如果没有你们从小到大对我的培养，就没有这一切。也感谢我的妹妹，在很多关键的时候，你总是给予我很多的帮助。

感谢我的妻子小猪猪，在本书创作的时候，没有你的理解和支持，我不可能完成这样一部作品；在本书发行后的日子里，要不是你将逛街、买衣服、旅游的时间慷慨地贡献出来，并盯紧四处乱跑的小宝宝，本书就不会有多次再版的机会。

感谢我的母校浙江大学，浙大“求是创新”的校训，“实事求是、严谨踏实、奋发进取、开拓创新”的校风让我能够有一个好的学习习惯，让我在毕业以后的这么多年里能够始终有一种动力去学习最新的知识。

感谢电子工业出版社博文视点资讯有限公司的郭立老师和李冰编辑，你们的支持、鼓励和专业的指点使本书能够按照进度按时完成。

非常高兴看到本书的再版，本书的前 3 个版本已经服务了几万读者，从本书的第 1 版发行到现在，我收到了很多读者和网友的反馈，使本书更加完善。在典藏版发行之际，再次感谢你们对我的关心和爱护，也感谢你们为我提了很多宝贵的意见和建议。

目　录

基础篇

初级篇

界面篇

系统篇

应用篇

第1章 背景知识

让我们在轻松的背景知识介绍中开始 Win32 汇编之旅。本章将对 Win32 平台的历史和现状做简要介绍，同时对 80386 处理器，以及 Windows 操作系统中涉及 Win32 汇编的基础知识部分做快速充电。

1.1 Win32 的软硬件平台

1.1.1 80x86 系列处理器简史

Win32 可以在多种硬件平台上运行，但使用最广泛的硬件平台是基于 Intel 公司 80x86 系列处理器的微型计算机。

自 1978 年 6 月 Intel 公司推出它的第一个 16 位微处理器 8086 以来，计算机技术就开始进入飞速发展的时期。8086 芯片的主频为 4.43 MHz，集成的晶体管数大约为 2.9 万个，运算器的位长为 16 位，采用了 20 条地址线，可以寻址的范围为 2^{20} 个字节地址，即 1 MB；1982 年，该公司发布了 80286 处理器，芯片上集成了 12 万个晶体管，主频提高到了 12 MHz。

1985 年 Intel 公司推出 32 位的 80386 处理器，芯片上集成的晶体管数为 27.5 万个，主频提高到了 33 MHz，地址线则扩展到 32 条，直接寻址的能力达到 4 GB。80386 处理器在设计的时候考虑了多用户及多任务的需要，在芯片中增加了保护模式、优先级、任务切换和片内的存储单元管理等硬件单元。80386 的出现使 Windows 和 UNIX 等多任务的操作系统可以在 PC 上运行。直到现在，运行于 80x86 处理器之上的多任务操作系统都是以 80386 的运行模式为基础的。

1989 年，Intel 公司推出 80486 处理器，在芯片内集成了浮点处理器和 8 KB 的一级缓存，片内的晶体管数达到了 118 万个，并把主频提高到 50 MHz～66 MHz。80486 处理器开始使用流水线技术，即在 CPU 中由 5～6 个不同功能的电路单元组成一条指令处理流水线，然后将一条指令分成 5～6 步后再由这些电路单元分别执行，由此提高 CPU 的运算速度。电路单元的数目就是流水线的深度。为了使计算机中的其他部件不至于成为 CPU 速度发展的瓶颈，80486 处理器开始使用了倍频技术，即让处理器速度（CPU 主频）数倍于系统总线速度（外频）。

从 80386 开始，在 Intel 公司向市场大量推出处理器芯片的同时，其他一些电脑公司和厂商如 AMD 和 Cyrix 等，也纷纷投入大量的人力财力进行处理器的开发和研制，并很快把研制出的产品推向市场。这些 CPU 芯片和 80386 芯片兼容，在编程上可以使用与 Intel 处理器相同的指令集。