计算机 C/C++语言系列丛书

C语言保护方式编程捷径

李增泰　编著

学苑出版社

计算机 C/C++语言系列丛书

C 语言保护方式编程捷径

李增泰　编著

学苑出版社
1994.

(京)新登字 151 号

内 容 提 要

当今，计算机 CPU 性能不断翻新，而软件开发却略显迟滞。例如微处理器 80386、80486 崭新而卓越的工作方式——“保护方式”，其性能就远远没有得到充分发挥，因为好多程序还只是局限于 386 对 8086 的兼容方式——“实方式”之下进行操作。本书则力图打破这种局面，向读者介绍“保护方式”下 C 语言程序的开发，以及 80386 汇编语言的使用。

计算机 C/C++语言系列丛书

C 语言保护方式编程捷径

编　　著：李增泰
责任编辑：甄国宪
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：14.875　　**字 数**：342 千字
印　　数：1～5000 册
版　　次：1994 年 6 月北京第 1 版第 1 次
ISBN7－5077－0875－6/TP・24
本册定价：17.00　元

学苑版图书印、装错误可随时退换

前　言

一. 内容介绍

使用 C 语言编写保护方式程序，便是本书向读者讲述的内容。另外书中还介绍了一些相关的技术，如 80386CPU 的工作方式、以及 DOS-Extender 的必备知识。

80386 的保护方式

80386CPU 最关键的进步就在于改变了存贮管理思想，并丰富了各种工作方式，它们包括实地址方式、保护方式、虚拟实方式。而对于一般计算机人员来说，最熟悉的莫过于实方式，因为 DOS 一直都是一个实地址方式的操作系统，而从未在保护方式方面作过什么设计。

80386 本身的数据宽度是 32 位，直接寻址能达到 4GB，而实方式所依据的是 16 位数据宽度，直接寻址仅仅有 1MB。若要充分发挥 386CPU 的高级特性，那只有选择保护方式。而且 80386 的先进的存贮管理，作为并行多任务的根据，只有在保护方式下才能得以体现。

DOS-Extender

DOS-Extender 属于 DOS 下的一个实用程序，它为 DOS 扩展了大量的保护方式功能，使系统能够支持保护方式程序的开发和运行。

C 语言保护方式编程

C 语言是当今最最流行的编程语言，但用它直接编写保护方式程序却是一个新鲜的领域。本书就将介绍这一技术，使众多 C 程序员的工作迅速进入一个更高的境界。

二. 本书的意图

用 C 语言编写保护方式程序，尚属崭新的技术。以前 386 或 386 以上微机，其 CPU 的卓越性能纵然为大家所了解，但由于缺乏必要的手段和工具，也只能望洋兴叹。

微机从 8088XT、80286AT 到 80386、80486 的进步，让计算机用户直接体会到的只有运行速度的提高，而其最具突破性的进步是 CPU 工作方式的改变，这一点却不能为大家随意驾驭，甚至大寻址空间的好处都不能方便地在自己的程序中得以体现。

与此技术相关的软件和资料在国内市场已经出现，例如 Watcom C、DOS-Extender，但

是它们所引起的注意，却远远没能达到其理应达到的程度。关键在于保护方式还没有被普遍了解，保护方式编程技术更未被一般计算机人员所掌握。

《C 语言保护方式编程捷径》意欲为改变这一状况提供必要的动力，帮助您尽快走进保护方式编程领域，使 386、486 的卓越性能为您尽情享用。

目　录

第一章　80386 工作方式

本章内容提要

- 80386 芯片的设计特色
- 80386 工作方式探讨

不管您是否熟悉 80386 的细节，在学习用 C 语言编写保护方式程序之前，了解一下 80386 的工作方式总是件受益匪浅的事情。

第一节　80386 的设计特色

80386 微处理器的设计有很多特色，如：并行多任务、大寻址空间、软件保护、分页虚存、以及与早期软硬件的向上兼容。

80386 的内部寄存器与数据通路，宽度都是 32 位，有效地址也是 32 位，因此能一次处理或传输 32 位的数据，并提供高达 4GB 的线性地址空间。

80386 包括多个密切配合的功能单元，如：指令预先提取单元、指令译码单元、执行单元、分段单元、分页单元、总线接口单元。在一条指令被执行的同时，其它指令被预先提取、译码，而且芯片上的地址计算和分页换算也都同时进行。多个单元的同时动作，正是多任务并行处理的基础。

80386 的分页单元负责存贮器管理、存放地址转换数据，并且提供虚拟存贮器能力，使得程序可以操作的地址空间比芯片的实际容量大得多。

由于完善的向上兼容特性，16 位机的实方式属性可以作为 80386 的一个子功能表现出来。

80386 的设计还有许多过人之处，留作此书以外的话题，供您研究。

第二节　80386 有哪些工作方式

80386 拥有完整的 32 位运算能力，同时具备作为 16 位 8086 以及 80286 的功能。其丰富的工作方式使得 80386 的计算能力极为强大且富有弹性。工作方式有如下三种：

- REAL，　　实地址方式
- PROTECTED，　　保护方式
- VIRTUAL8086，　　虚拟 8086 方式

三种方式的寻址方法不同、寻址空间大小不同。

在 PC 机的发展过程中，随着 CPU 的进步，其工作方式也经历了几次关键性的变化。

· 16 位的 8086/8088：只有实地址方式，寻址范围仅仅 1MB

· 16 位的 80286：加进了高级的 16 位保护方式，寻址存储器高达 16MB

· 80386：在保护方式中又加进了 32 位运算，寻址空间大于 4GB，并且添加了虚拟 8086 方式，以兼容现有的 8086 软件

这一发展过程可参见图 1—2—1

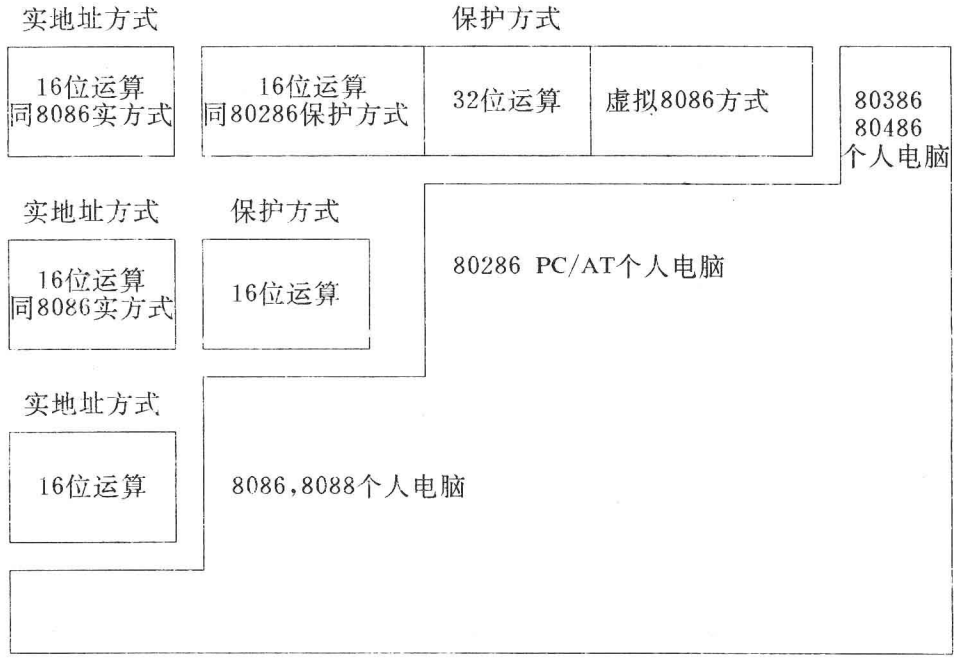

图1—2—1　CPU及相关个人电脑工作方式发展过程

保护方式是 80386 工作方式中非常优秀的一种，不仅具有 32 位的数据传输与寻址能力，而且能引用具有分页作用的虚拟存储器，具有坚强的软件安全防范措施。保护方式编程已经成为未来软件开发中一个不可忽视的方向。

80386 的众多先进特性只有在保护方式下才能得到充分体现。在保护方式下，可以构造一个多任务环境，任务之间是相互隔绝的，任务和操作系统之间也是隔绝和保护的。同时任务之间可以相互切换，8086 实方式程序就可以作为一个任务，运行于保护方式划分出来的一块空间之中，而且与其它任务之间互不影响。

80386 的硬件在保护方式下，提供完全的 32 位运算。不管是 16 位还是 32 位运算，只要在 80386 保护方式下，就能启动其分页单元，以完全支援虚拟存储器，使编程人员免于为实地址的寻址极限大伤脑筋。

实方式已经为大家所明了，可保护方式的情况也许不同，尤其对使用高级语言的编程人员，诸如 C 语言。如何轻而易举地充分发挥计算机硬件的先进性能，如何追随软件的时代潮流，希望这里能给您提供一个欣喜的开端。

第二章　保护方式下的内存管理

在这一章，除介绍了保护方式下的内存管理以外，还提到了 DOS-EXTENDER，它实际上是介于 DOS 和应用程序之间的一个界面，用来支持 32 位程序的运行和开发。关于 DOS-EXTENDER 在第三章将集中讨论。

第一节　内存管理模式

80386、80486 通过其分页、分段功能实现内存管理。

2.1.1　分段

保护方式下的寻址空间，同样被分成若干个“段”，只是各段之间相互隔绝，不可随意访问。如果一个段中的程序企图引用其它段的数据，将产生严重故障。

每个段都有自己专用的描述符，定义本段的 32 位线性基地址、边界、属性、和权限。

所有段的段描述符罗列在一起，组成一个描述符序列，称为描述符表。

当然表中的每一项（一个描述符）都被分配一个序号，这个序号被叫做段选择器。

程序如何控制、查阅分段信息呢？

程序先从段寄存器（CS）中取出段选择器，再根据段选择器值从描述符表中取出段描述符，最后从段描述符获得分段的各种信息。

2.1.2　分页

把 32 位的线性地址转换到 32 位的物理地址，即相应的地址总线上，就是通过 80386 的分页机制。

线性地址中有 20 位用来选择内存的物理页面，而另 12 位描述页面内的偏移量。一个物理页面大小是 4KB。

2.1.3　内存访问

80386 通过一个段选择器和一个段内偏移量来索引到内存。保护方式下的内存访问，利用如下方法构造物理地址：

- 从段描述符得到段的线性基地址
- 把段内偏移量与基地址相加，得到一个 4GB 空间内的线性地址空间
- 通过分页机制把线性地址转换成和地址总线相对应的物理地址

第二节　内存分配

2.2.1　分配内存的方法

内存包括常规内存、扩展内存。

常规内存，即 1MB 以内的存贮空间。

扩展内存，即 1MB 以上的所有存贮空间。

应用程序有权使用常规内存，也有权使用扩展内存。但通常是常规内存优先分配，然后才分配扩展内存。

内存分配和重分配总是以页为单位。内存分配没有特殊的顺序，因为不再需要的内存一旦被释放，任何后继的内存请求都可获得其使用权。

关于扩展内存的分配，有三种方法：

· 直接扩展内存，利用 INT15H 的 88 号功能调用，获取扩展内存的大小，接管并保留对内存的控制

· VCPI 虚拟控制程序接口内存，系统引导时，由 EMS 仿真程序截取一块内存，DOS-EXTENDER 利用 VCPI 动态地分配来自这块内存的页面

· XMS 内存，使用 HIMEM. SYS 分配扩展内存

DOS-EXTENDER 支持上述所有三种方法，但注意，同一时刻只能让一种方法有效，如若 VCPI 和 XMS 同时存在，则 DOS-EXTENDER 使用 VCPI，而非 XMS。

2.2.2　常规内存的使用

一般情况下，DOS-EXTENDER 占用大约 80KB（至少 65KB）常规内存归自己使用，并且通过 MS-DOS 调用给应用程序内存缓冲区保留一块内存，因为常规内存是可以从 MS-DOS 获得的内存。常规内存分配情况参见图 2－2－2－1

2.2.3　直接扩展内存的使用

直接扩展内存，指 1MB 以上尚未被分配出去的内存。

初始化期间，DOS-EXTENDER 检查有多少直接扩展内存还没被其它程序所占用，然后再给予分配。一般有两种分配策略。

DOS-EXTENDER 以页为单位、自上而下分配扩展内存，直到用完为止。假设一个保护模式主程序含有若干个保护模式子程序，主程序一旦装入运行其子程序，子程序就可以使用主程序的扩展内存，子程序结束后，把内存自动释放并归还给主程序。

扩展内存页面被释放时，DOS-EXTENDER 将其标记下来，以备后继程序使用。

图2－2－1 常规内存分配表

第三章　DOS-EXTENDER 了解

DOS-EXTENDER 是在 DOS(MS-DOS 或 PC-DOS)下运行或开发 32 位应用程序的保护模式环境,在其中运行的程序能直接访问机器的所有内存。本书将集中探讨的内容—C 语言编写保护方式程序,就将完全依赖这一环境。

DOS 一直都只是一个实方式的操作系统,而借助于 DOS-EXTENDER,我们直接感受到的将是 386 以上微处理器完善的保护方式性能。

第一节　入门者须知

DOS-EXTENDER 有哪些版本?哪些相关产品?适用于什么环境?兼容性如何?下面逐一介绍。

3.1.1　DOS-EXTENDER 的版本

DOS-EXTENDER 有开发版本、运行版本两种。

开发版是一个名为 RUN386.EXE 的文件,是 Phar Lap 公司的 DOS-EXTENDER 开发工具。

运行版是一个名为 RUN386B.EXE 的文件,这是用户进入实用阶段需要的版本。

3.1.2　DOS-EXTENDER 的相关产品

386NMM,使用 DOS-EXTENDER 的虚拟存贮系统,若希望在小内存的 PC 机上运行大的应用程序,可使用它。

3.1.3　DOS-EXTENDER 的适用环境

它可以运行于 80386、80486 系列的所有 PC 机之上。

3.1.4　DOS-EXTENDER 的兼容性

它与以下系统软件兼容:实模式和标准模式下运行的 Microsoft Windows 3.0、Quarterdeck DESQ View 386、和所有支持 VCPI 虚拟控制程序接口的 EMS 扩展内存系统仿真程序。

Phar Lap 公司正着手新的版本以支持 OS/2、UNIX、增强模式下运行的 Windows 3.0。

第二节　DOS-EXTENDER 使用简述

这里只简述其使用方法，后面 C 语言编程部分还会涉及一些具体操作。如果您不想在此花费太多时间，可跳过本节。

3.2.1　生成并运行保护方式的应用程序

3.2.1.1　如何生成 80386 目标代码

源程序如果是用高级语言编写，则必须用可生成 80386 目标代码的编译器进行编译。在后面 C 语言编写保护方式程序部分会告诉您如何操作。

源程序如果是汇编语言，则必须用 386 汇编工具，如 phar lap assembler，386ASM，生成 80386 目标代码。

例如 汇编程序源文件 HELLO. ASM，执行如下命令：

386asm hello ↙

生成 80386 目标代码文件 HELLO. OBJ。

3.2.1.2　连接目标码生成保护方式的可执行程序

对于各种方法生成的 80386 目标文件，要连接生成保护方式的可执行文件，还需要专门的连接程序，如 phar lap 的 386|LINK。

例如连接 HELLO. OBJ 和库文件 CLIB. LIB 以生成保护方式可执行文件 HELLO. EXP，操作如下：

386link hello -lib clib ↙

3.2.1.3　运行保护方式的可执行程序

DOS-EXTENDER 有两种版本，其用法也各不相同。

开发版的 DOS-EXTENDER 是一个名为 RUN386. EXE 的文件，命令行语法如下：

run386　　命令行开关　　执行文件名　　命令行参数

例如

run386 hello ↙

运行版本的 DOS-EXTENDER，除一个名为 RUN386B. EXE 的文件，还有一个连接实用程序 BIND386。

用 BIND386 把您的程序和 RUN386B. EXE 连接到保护方式应用程序中，生成 . EXE 文件，能在实方式的 MS-DOS 命令行方式运行。

例如 您有一个 HELLO. EXP 文件，执行下面操作：

bind386 run386b hello ↙

则生成的 HELLO. EXE，在 MS-DOS 下运行，只需键入：

hello ↙

3.2.2 DOS-EXTENDER 命令行开关

利用命令行开关能改变 DOS-EXTENDER 的缺省操作。在一个命令行中，如果有若干个开关发生冲突，则最后一个有效。

DOS-EXTENDER 的所有命令行开关都在下面列出。

3.2.2.1 开关名称及其语法

开关类别	全称及语法	缩写
空闲内存管理	-MINREAL nparagraphs	-MINR
	-MAXREAL nparagraphs	-MAXR
DOS 缓冲	-MINIBUF nkilobytes	-MINI
	-MAXIBUF nkilobytes	-MAXI
物理内存使用控制	-MAXBLKXMS nbytes	-MAXBL
	-MAXXMSMEM nbytes	-MAXX
	-MAXEXTMEM nbytes	-MAXE
	-MAXVCPTMEM nbytes	-MAXV
扩展内存使用数量控制	-EXTLOW address	-EXTL
	-EXTHIGH address	-EXTH
调整特殊内存	-NOSPCLMEM	-NOSPCL
	-NOBIM	-NOB
线性内存使用限制	-MAXPGMMEM nbytes	-MAXP
特权级别开关	-PRIVILEGED	-PRIV
	-UNPRIVIEGED	-UNPRIV
混合模式程序开关	-REALBREAK nbytes	-REALB
	-CALLBUFS nkilobytes	-CALLB
描述符长度开关	-GDTENT nentries	-GDTE
	-LDTENT nentries	-LGDE
协处理器开关	-WEITEK AUTO	-WEITEK AUTO
	-WEITEK ON	-WEITEK ON
	-WEITEK OFF	-WEITEK OFF
	-CYRIX AUTO	-CYRIX AUTO
	-CYRIX ON	-CYRIX ON
	-CYRIX OFF	-CYRIX OFF
栈分配开关	-NISTACK nbuffers	-NI
	-ISTKSIZE nkilobytes	-IS
分页禁止开关	-NOPAGE	-NOP
80386 错误 17 工作区	-ERRATE 17	-ERR
PharLap 虚拟内存管理程序开关	-CODESIZE nbytes	

	-DEMANDLOAD	
	-FLUSHSWAP	
	-LFU	
	-LOCKSTACK nbytes	
	-MAXSWPSIZE nbytes	
	-MINSWFSIZE nbytes	
	-NOPGEXP	
	-NOSWFGROWIST	
	-NOVM	
	-NUR	
	-PAGELOG filename	
	-SWAPCHK ON|OFF|FORCE|MAX	
	-SWAPDEFDISK	
	-SWAPDIR dimame	
	SWAPNAME filename	
	SWFGROWIST	
	VMFILE filename	
	-VSCAN nmilliseconds	
	-VSLEN nbytes	
硬件结构类别开关	-XT	-XT
	-AT	-AT
	-EISA	-EISA
20 行地址线开关	-A20	-A20
VDISK 兼容性开关	-VDISK	-VDISK
保留 32 位寄存器开关	-SAVEREGS	-SAVER
禁止乘法检查开关	-NOMUL	-NOMUL
在 VCPI 下不运行开关	-NOVCPI	-NOVCPI
启动调试打印开关	DEBUG level	-DEBUG
I/O 重定向开关	-COM1	-COM1
	-COM2	-COM2
	-BAUD speed	-BAUD
功能调用指针转换	-DATATHRESHOLD nbytes	-DATAT
保存实模式中断向量	-SAVEINTS	-SAVEI
重定位 BIOS 打印屏幕	-PRIVEC vector	-PRI
重定位硬中断 IRQ0～7	-INTMAP vector	-INTM
	-HWIVEC vector	-HWI
BIOS 块传送	-NOBMCHK	-NOBMCHK
设置与模式打		

开.EXP 文件　　-OPENDENY　　-OPEND

3.2.2.2　开关使用范例

开关名称	范例
-MINREAL	run386 -minreal 100h hello
-MAXREAL	run386 -minr 128 -maxr 512 hello
-MINIBUF	run386 -minibuf 64 -maxi 64 filecopy
-MAXIBUF	run386 -maxibuf 2 hello
-MAXBLKXMS	run386 -maxbl 80000h hello
-MAXXMSMEM	run386 -maxx 100000h hello
-MAXEXTMEM	run386 -600000h hello
-MAXVCPTMEM	run386 -600000h hello
-EXTLOW	run386 -ext(on) 200000h hello
-EXTHIGH	run386 -ext 180000h -exth 400000h hello
-NOSPCLMEM	
-NOBIM	run386 -nobim hello
-MAXPGMMEM	run386 -maxp 500000h hello
-PRIVILEGED	run386 -priv myprog
-UNPRIVIEGED	run386 -upriv hello
-REALBREAK	run386 -realbreak 2010h -callb 2 switch
-CALLBUFS	run386 -realbreak 2010h -callb 2 switch
-GDTENT	run386 -gdte 600 -ldte 1200 hello
-LDTENT	run386 -gdte 600 -ldte 1200 hello
-WEITEK AUTO	
-WEITEK ON	run386 -weitek on float.exp
-WEITEK OFF	
-CYRIX AUTO	
-CYRIX ON	run386 -cyrix on float.exp
-CYRIX OFF	
-NISTACK	run386 -ni 8 -istk 2 switch.exp
-ISIKSIZE	run386 -ni 8 -istk 2 switch.exp
-NOPAGE	run386 -nopage numcrunch
-ERRATE 17	run386 -errata numcrunch
-CODESIZE	
-DEMANDLOAD	
-FLUSHSWAP	
-LFU	
-LOCKSTACK	