Communication...

加密解密与网络安全技术

Information Technology

张曜 卢涛 张青 编著

冶金工业出版社

加密解密与网络安全技术

张曜　卢涛　张青　编著

北　京

冶 金 工 业 出 版 社

2002

内 容 简 介

软件保护和分析技术是软件开发人员应该掌握的重要技术，网络安全是大家所关心的问题。本书讲述了 Windows 环境下的软件保护和分析技术以及网络安全方面的知识，并通过大量生动的实例向你揭示了黑客和快客的秘密。全书分三个部分。第一部分讲述了软件分析技术的基本常识和主要工具的使用；第二部分讲述了各种软件的保护方法和相应的分析方法；第三部分讲述了网络安全的基本常识以及一般的网络攻击和防守的方法。

本书适用于对软件加密和解密以及网络安全技术感兴趣的读者，也可作为软件开发人员的参考资料或技术人员的进修教材。

图书在版编目（CIP）数据

加密解密与网络安全技术 / 张曜等编著. —北京：冶金工业出版社，2002.6
ISBN 7-5024-3039-3

I. 加… II. 张… III. 计算机网络—安全技术
IV. TP393.08

中国版本图书馆 CIP 数据核字（2002）第 031629 号

出版人　曹胜利（北京沙滩嵩祝院北巷 39 号，邮编 100009）
责任编辑　程志宏
广东出版技校彩印厂印刷；冶金工业出版社发行；各地新华书店经销
2002 年 7 月第 1 版，2002 年 7 月第 1 次印刷
787mm×1092mm　1/16；　25.5 印张；　588 千字；　397 页；　1-2600 册
40.00 元
冶金工业出版社发行部　电话：（010）64044283　**传真：**（010）64027893
冶金书店　地址：北京东四西大街 46 号（100711）　**电话：**（010）65289081
（本社图书如有印装质量问题，本社发行部负责退换）

前　言

1、关于本书

随着电脑应用的普及，如果不会用电脑简直是新世纪的文盲。当想用计算机做某件事时，你肯定会先想到，有没有好的软件帮忙呢？例如，有一天在网上浏览了一大网页后，眼睛很累，突然想，如果有人给自己念一念就好了，就到一般的搜索引擎上搜了一把。还别说，居然真有一个叫网络鹦鹉的好东西能帮你听网！下载下来一试感觉还不错，这下可以休息一下眼睛了。大家可能都会有这种经历，从网上下载这样的一些软件，多数都是共享软件。

这些共享软件当然是别人劳动的成果，在让你免费试用一段时间后会让你交钱。你一定会想，东西已经到我手了怎么让我掏银子？这就涉及到软件的保护技术问题。如果你是一个程序员，你辛辛苦苦开发的软件当然不想被人免费使用或者“借鉴”，这时你可以采用软件保护技术来保护你的成果。

在本书中，我们主要从软件保护和破解两个方面对现在流行的软件保护技术进行了分析，让大家对这两种技术有所了解，不再感到神秘。

随着计算机及网络应用的扩展，电脑信息安全所面临的危险和由此造成的损失也成倍地增长。密码被盗、邮箱遭炸、黑客侵入……各种形式的攻击和侵害让你觉得网络很不安全。媒体有关黑客的报道又让你以为这些入侵者都是计算机天才，他们的技术肯定是高深莫测的。其实并不像你想象的那样，你也可以搞懂他们的技术，从而保护自己的安全。

在本书中，我们主要从网络攻击和防守两个方面对现在流行的操作系统的攻击方法和防守方法进行了分析，让你对网络安全的技术有所了解。

2、本书的结构安排

本书讲述了 Windows 环境下的软件保护和分析技术以及网络安全方面的知识。全书分三个部分。第一部分（第 1 章）讲述了软件分析技术的基本常识和主要工具的使用；第二部分（第 2~5 章）讲述了各种软件的保护方法和相应的分析方法；第三部分（第 6~7 章）讲述了网络安全的基本常识以及一般的网络攻击和防守的方法。

3、本书特点

本书通过生动的语言对枯燥无味的技术进行了生动的讲解，以丰富的实例把难懂的问题变得形象化具体化。丰富的实例是本书的一大特点，常言道：“百闻不如一见，百见不如一做”，很多事情亲自动手比只用眼睛看更容易让人理解。

4、如何使用本书

如果你对本书讲述的内容比较陌生，你可以从第 1 章开始看，然后对照书中的例子一

步步实践，这样可以让你快速入门。你也可以直接在书中找你比较感兴趣的部分而跳过前面的基础知识。书中所有的例子用到的软件以及一些工具都可以在网上找到，你可以用天网 FTP 搜索或其他 FTP 搜索引擎，按照文件名搜索就可以找到。找到后下载下来，你就可以对照书中的步骤来实践了，不过要注意版本问题，有些软件不同的版本差别很大的。

5、本书适用对象

如果你是新手，你可以从本书快速入门；如果你是高手，你也可以从本书中找到你想要的资料，或从书中得到启发。因此，本书不但是软件加密解密与网络安全技术的进阶教程，同时对有一定基础的读者也是一本很好的参考资料。

由于水平有限，书中难免存在不妥之处，望广大读者给予批评指正，以利改进和提高。

编　者

2002 年 5 月

目 录

目 录

目 录

第1章 预备知识

1.1 写在前面的话

软件破解不是学习使用一个什么软件，不是按照说明书来操作，它是一种人和人智力的较量，是一种智慧的战争艺术，是一种知识与知识的较量。从本质上讲，学习破解跟学习任何一门别的知识一样，都是要下苦功夫，要靠灵感，要靠自己思考的。试想谁能有一个软件或者方法能在一天之内学会英语呢？

如果你已经有了心理准备，欢迎进入 Cracker 的世界，这个世界 99%的事情是枯燥无味的，而且从来没有，将来也不会有捷径。

当然，本书并不是现在流行的“傻瓜书”，傻瓜不用也不会弄明白软件的加密解密原理，更不用说想破解别人的软件。所以使用本书需要有一定的计算机常识，谈到常识，想起在网上看到的一个笑话：

一个人给微软的客户服务部打电话：

客户：我在打一篇文章时，什么也看不到了。

服务人员：是不是显示器的接线掉了？

客户：什么是显示器？

服务人员：就是你看到你打的东西的那个桌上的东西。

客户：找到了！

服务人员：看一下它后面出来的线。

客户：等一下，我找不到。

服务人员：什么，找不到？

客户：我去取一根蜡烛。

服务人员：蜡烛？

客户：我们这儿停电了。

服务人员：停电了？我知道毛病出在哪儿了。

客户：你知道了？

服务人员：是的，你的计算机出了很大的毛病，你明天带着它去找销售人员。

客户：我说什么？

服务人员：你就说，我太笨了，不配拥有一台计算机！

以上只是一个笑话，讲这个笑话的意思是说你读本书前要有一点常识，起码知道 CPU、RAM、操作系统等等是什么东西。

计算机硬件中最重要的 CPU 是 Intel 的天下，软件中最重要的操作系统是微软的天下，下面就来介绍软件破解所需要的这两方面的知识。

该章是后面要参考到的基本知识，你可以先跳过不看，在后面要用到或不明白时，再回过来看这一章。

1.2 软件加密解密方法

加密，在计算机领域中早已不是一个陌生的词汇。由于目前我国的软件保护法制还不太健全，人们的法制观念也比较淡薄，并且计算机软件是一种特殊的商品，极易复制，所以加密就成为了保护软件的一种必要手段。现在市场上流行的软件多数都采取了一定的加密方法，其目的就在于保护软件开发者的利益，防止软件被盗版。

目前采用的比较多的商业加密方法可分为两大类：钥匙盘方式和加密狗方式。加密狗是目前流行的一种加密工具，它是插在计算机并行口上的软硬件结合的软件加密产品。加密狗一般都有几十或几百字节的非易失性存储空间可供读写，有的内部还增添了一个单片机。软件运行时通过向并口写入一定数据，判断从并口返回密码数据正确与否来检查加密狗是否存在。此种方式不易被硬解密，因而具有加密可靠等优点。但它也存在一大缺点：成本较高，并且用户使用很不方便。若用户购买了几种带加密狗的软件，在使用不同软件或更换微机时要不断将加密狗插上拔下，给用户增添了很多麻烦。

所谓钥匙盘方式就是通过 BIOS 的 INT13 中断对软盘格式化一些特殊的磁道，有的还在特殊磁道里写入一定信息，软件在运行时要校验这些信息。这种软盘就好像一把钥匙一样，所以被人习惯称为钥匙盘。它也是目前流行的一种加密工具。采用此种加密方式的软加密工具有很多，如 Softlock、Bitlok、Keymaker、Lock95、Lockstar 等。软件商只需一次性投资购买一套加密工具，就可自己制作多张钥匙盘，在软件中读取钥匙盘上的特殊磁道来检查钥匙盘是否存在。此种方式加密简便，成本低，用户使用方便。使用此种加密方式的软件也比较多，如 KILL、KV3000、瑞星杀毒软件等。此种方式存在一大缺陷是易被硬解密（因为别人也可学会通过 BIOS 的 INT13 中断格式化技术），也就是易被非法者生成相同的钥匙盘用来作为正版出售。例如，KV3000 目前的窘况，江民公司采用这种加密技术保护不了自己的利益，只好一再声明用户注意所谓正版激光防伪标志。

目前的解密方法主要可分为两种：软解密和硬解密。

所谓软解密就是针对加密产品，一方面利用软件监测分析软件在运行时向加密点写了什么数据，从加密点返回了什么数据，然后在运行软件前先在内存驻留自编程序监视加密点，当软件向加密点写数据时，软件自动代替加密点并返回相应数据，用软件模拟了加密产品；另一方面是从软件着手，寻找软件调用加密点函数部分，修改判断加密点是否存在的语句，将程序直接跳转到正常执行的部分。

如：打狗棒 Dog、Unlock95、解密之星、Llgz35、Magickey、Msc、密界克星 Sc40、SoftICE、Ulm、Unall 等。但这种解密并不一定很彻底，由于加密者的设计，这种解密后的软件往往设有一定的陷阱，例如，闹得沸沸扬扬的 KV3000 逻辑锁事件；再者软件商在短时间内即做一次软件升级，使解密者难于应付。因而除一些出于学习目的的人购买盗版软件外，一般各种单位公司等软件的正式用户还是比较注重软件的可靠性，而购买正版软件。因而可以说软解密对于软件开发商的利益损害并不算大。

所谓硬解密就是针对加密产品，专门研究加密点结构与数据，而自制具有相同结构及加密点的钥匙盘或加密狗。如目前流行的一种由成都双星软件技术工作室推出的密钥盘硬解密工具 KING_COPY，令广大解密爱好者欣喜若狂，更使软件开发商恨之入骨。这样的硬解密软件除了不是原软件开发商出售的，利益完全被盗版者获得外，其余皆与正版一样，

甚至享有免费升级及售后服务的权利。如著名的杀毒软件 KV3000，市面上不少公司卖的所谓正版，实际上不是江民公司的产品而是盗版者的硬解密产品。愿意购买正版软件的广大用户并不能分清这是盗版。并且购买这种所谓的“正版”后，大多得不到免费升级等售后服务，因而给广大用户带来不便。用户因分辨不清是否是真正的正版，从而不愿再购买此软件，而选择购买别的同类软件！因而这种硬解密对于软件开发商的利益损害极大。明智的软件开发商在选择加密方法时，应把注意力集中到这种产品是否易被硬解密上。

1.3 软件破解入门知识

学习软件加密解密知识需要一些基本知识，如汇编语言。如果从来没有用过也不要紧，只要清楚下面几个指令就可以：

1. 几条常用的汇编命令

- 跳转命令

根据条件作出是否跳转的决定，通常前面会有一个判断语句，例如：

CMP AX，BX

JZ XX

上面两条命令意为用 AX 减 BX，它的值如果为 0 则跳转到 XX 的标号行。常用的跳转命令有：

JZ/JE：相等或为零则跳转。

JNZ/JNE：不相等或不为零则跳转。

JL/JLE：小于/小于或等于则跳转。

JG/JGE：大于/大于或等于则跳转。

JMP：无条件跳转。

- 比较语句

CMP AX，BX AX：寄存器减去 BX 寄存器的内容。

AND AX，BX：AX 与 BX 做“与”运算。

OR AX，BX：AX 与 BX 做“或”运算。

TEST AX，BX：与 AND AX，BX 命令有相同效果。

XOR AX，AX：使 AX 的内容清零，每个寄存器与自己作异或运算等于清零动作。

- 子程序

一个子程序如下所示：

CALL 15F:334422

子程序是个很重要的概念，它是主程序的一个分支，用来做特定动作。打个比方：你要上班，你先是走路到车站，然后上车，然后下车，然后走到自己的办公室。这里如果要把上班编为一段程序的话，那么就可以把“走路”，“搭车”，“走到办公室”作为分支程序来处理。说得再通俗一点就是：要破解的程序不可能就是一条主程序到底，肯定会呼叫下面的子程序，由子程序来处理发送的注册信息，然后比较，然后标记是否注册正确，这些都是靠它来完成的。所以说，破解的关键在于，找准程序在哪儿将会作注册判断，并进入那个注册子程序，仔细观察，就成功了。子程序的返回码是 RET。

- 算术运算

ADD AX，BX：加法运算 AX=AX+BX。

SUB AX，BX：减法运算 AX=AX-BX。

INC AX：寄存器加一 AX=AX+1。

DEC AX：寄存器减一 AX=AX-1。

MUL：乘法运算。

DIV：除法运算。

- 数据操作

MOV AX，BX：数据传送指令，将 BX 的值移送到 AX 中。

XCHG AX，BX：将 AX 与 BX 的值互换。

有了上面的汇编语言知识，还需要一些软件工具，下面介绍几种可以进行软件分析和破解的工具。

2. 破解的工具介绍

修改档案的工具：PCTOOLS，HVIEW，PSE 等，这些都是 DOS 下的修改程序，现在使用的主要操作系统是 Windows，所以应该选用 ULTRAEDIT，它的功能非常强大。

除错程序的工具：DEBUG，SYMDEB，GAMETOOLS，CM386，但是这些都只能调试实模式下的程序，所以如果要破 WINDOWS 下的软件，必须备上：SoftICE、TR 等。

编码相关工具：UNP，PKLITE，GT，WWPACK，UNWWPACK 等是一些压缩及解压缩的工具，一个程序被压缩的目的是防止有人破解它，它通过一定的运算法编辑，执行的时候用自己的压缩算法在内存中开辟一块区域，解码后再执行。所以如果你要破解一个被压缩了的软件时，在内存中看到的机械码会与软件本身不一样，从而使你不能修改它。因此另一类常驻式的修改工具相应而生，如 TSRCRACK。

档案回写工具：EXEWRITE，EXESHAPE 等，这些工具的目的也是为了解压缩程序，它通过在内存中追踪到的一些寄存器的位址，加上一些运算来重新产生一个新的没有编码的软件。

SoftICE 功能非常强大，防死机能力也是一流，几乎没有什么程序是它不能够追踪下去的，本书后面对它有详细讲解。

共享软件，英文是“Shareware”，一般是指“先尝后买”的软件。在共享软件中，某些产品其实只是商业软件的测试版，共享是为了方便传播。共享软件有时加入时间限制，试用期一过，再想继续使用就不太容易了。还有一些是不用注册的，也没有时间限制，但对软件功能上有限制。

下面就从程序编写角度对共享软件的注册问题进行分析。

3. 共享软件注册问题

- 程序怎么判断注册与否

1）在程序码的某一处，藏有“注册印记”。

2）安装时在“安装信息文件”，如*.INI 中存入资料档。

- 程序怎么生成注册码

1）一般是通过用你的 ID 作 KEY，经过一定的算法，牵引出 CODE 码。

2）但是也有通过依照随机产生的数字，显示出来，如果你要注册，需交钱给软件设计者，并告诉他随机数字，他会给你一个经过按照随机数字产生出的注册码。

其实上面两种情况都是一个道理，即程序本身在判断你输入的注册码时都会呼叫算注册码的子程序，只要将这段程序正确找到，就可以破解它。

- 程序怎么做注册处理

1）当输入 ID，CODE 等，立即进行对比，正确的话就做注册处理，以后它也不会再麻烦你，也就是说你已经是注册用户了。不正确的话，当然是输入到正确为止了。

2）同上，但以后一执行程序还是会判断。对于这种情况，想只改机器码是不行的，必须把注册码揪出来。

3）输入时不比对，只写入资料到“*.ini”文件中，程序第二次启动时再来比对。例如 UltraEdit，特点是太麻烦，想破解它们，必须要有足够的耐心。

1.4 保护模式简介

80386 有三种工作方式：实模式，保护模式和虚拟 8086 模式。本章介绍保护方式下的 80386 及相关的程序设计内容。实模式下的 80386 寄存器，寻址方式和指令等基本概念，除特别说明外在保护方式下仍然保持。

尽管实方式下 80386 的功能要大大超过其先前的处理器（8086/8088,80186,80286），但只有在保护方式下，80386 才能真正发挥更大的作用。在保护方式下，全部 32 条地址线有效，可寻址高达 4G 字节的物理地址空间；扩充的存储器分段管理机制和可选的存储器分页管理机制，不仅为存储器共享和保护提供了硬件支持，而且为实现虚拟存储器提供了硬件支持；支持多任务，能够快速地进行任务切换和保护任务环境；4 个特权级和完善的特权检查机制，既能实现资源共享又能保证代码和数据的安全和保密及任务的隔离；支持虚拟 8086 方式，便于执行 8086 程序。

1.4.1 存储管理机制

为了对存储器中的程序及数据实现保护和共享提供硬件支持，为了对实现虚拟存储器提供硬件支持，在保护方式下，80386 不仅采用扩充的存储器分段管理机制，而且提供可选的存储器分页管理机制。这些存储管理机制由 80386 存储管理部件 MMU 实现。

1. 目标

80386 有 32 根地址线，在保护方式下，它们都能发挥作用，所以可寻址的物理地址空间高达 4G 字节。在以 80386 及其以上处理器为 CPU 的 PC 兼容机系统中，把地址在 1M 以下的内存称为常规内存，把地址在 1M 以上的内存称为扩展内存。

80386 还要对实现虚拟存储器提供支持。虽然与 8086 可寻址的 1M 字节物理地址空间相比，80386 可寻址的物理地址空间可谓很大，但实际的微机系统不可能安装如此达的物理内存。所以，为了运行大型程序和真正实现多任务，必须采用虚拟存储器。虚拟存储器是一种软硬件结合的技术，用于提供比在计算机系统中实际可以使用的物理主存储器大得多的存储空间。这样，程序员在编写程序时不用考虑计算机中物理存储器的实际容量。

80386 还要对存放在存储器中的代码及数据的共享和保护提供支持。任务甲和任务乙

并存，任务甲和任务乙必须隔离，以免相互影响。但它们又可能要共享部分代码和数据。所以，80386 既要支持任务隔离，又要支持可共享代码和数据的共享，还要支持特权保护。

2. 地址空间和地址转换

保护方式下的虚拟存储器由大小可变的存储块构成，这样的存储块称为段。80386 采用称为描述符的数据来描述段的位置、大小和使用情况。虚拟存储器的地址（逻辑地址）由指示描述符的选择子和段内偏移两部分构成，这样的地址集合称为虚拟地址空间。80386 支持的虚拟地址空间可达 64T 字节。程序员编写程序时使用的存储地址空间是虚拟地址空间，所以，他们可认为有足够大的存储空间可供使用。

显然，只有在物理存储器中的程序才能运行，只有在物理存储器中的数据才能访问。因此，虚拟地址空间必须映射到物理地址空间，二维的虚拟地址必须转化成一维的物理地址。由于物理地址空间远小于虚拟地址空间，所以只有虚拟地址空间中的部分可以映射到物理地址空间。由于物理存储器的大小要远小于物理地址空间，所以只有上述部分中的部分才能真正映射到物理存储器。

每一个任务有一个虚拟地址空间。为了避免多个并行任务的多个虚拟地址空间直接映射到同一个物理地址空间，采用线性地址空间隔离虚拟地址空间和物理地址空间。线性地址空间由一维的线性地址构成，线性地址空间和物理地址空间对等。线性地址 32 位长，线性地址空间容量为 4G 字节。

80386 分两步实现虚拟地址空间到物理地址空间的映射，也就是分两步实现虚拟地址到物理地址的转换，但第二步是可选的。如图 1-1 所示是地址映射转换的示意图。

图 1-1

通过描述符表和描述符，分段管理机制实现虚拟地址空间到线性地址空间的映射，实现把二维的虚拟地址转换为一维的线性地址。这一步总是存在的。

分页管理机制把线性地址空间和物理地址空间分别划分为大小相同的块，这样的块称为页。通过在线性地址空间的页与物理地址空间的页建立之间建立的映射表，分页管理机制实现线性地址空间到物理地址空间的映射，实现线性地址到物理地址的转换。分页管理机制是可选的，在不采用分页管理机制时，线性地址空间就等同于物理地址空间，线性地址就等于物理地址。

分段管理机制所使用的可变大小的块，比较适宜处理复杂系统的逻辑分段。存储块的大小可以根据适当的逻辑含义进行定义，而不用考虑固定大小的页所强加的人为限制。每个段可作为独立的单位处理，以简化段的保护及共享。分页机制使用的固定大小的块最适合于管理物理存储器，无论是管理内存还是外存都同样有效。分页管理机制能够有效地支持实现虚拟存储器。

段及分页这两种机制是两种不同的转换机制，是整个地址转换函数的不同的转换级。虽然两种机制都利用存储在主存储器中的转换表，但这些表具有独立的结构。事实上，段表存储在线性地址空间，而页表存储在物理地址空间。因此，段转换表可由分页机制重新进行定位而不需段机制的参与。段转换机制把虚拟地址转换为线性地址，并在线性地址中

访问段转换机制的表格，而不会觉察分页机制已把线性地址转换为物理地址。类似地，分页机制对于程序产生的地址所使用的虚拟地址空间一无所知。分页机制只是直接地把线性地址转换为物理地址，并且在物理地址中访问转换表格，并不知道虚拟地址空间的存在，甚至不知道段转换机制的存在。

3. 虚拟存储器概念

虚拟存储器是一种设计技术，用于提供比在计算机系统中实际可以使用的物理主存储器大得多的存储空间。使用者会产生一种错觉，好像在程序中可以使用非常大的物理存储空间。使用虚拟存储器的好处是：一个程序可以很容易地在物理存储器容量大不一样的、配置范围很广的计算机上运行；编程人员使用虚拟存储器可以写出比任何实际配置的物理存储器都大得多的程序。虚拟存储器由存储管理机制及一个大容量的快速硬盘存储器支持。在程序运行的任何时刻，只把虚拟地址空间的一小部分映射到主存储器，其余部分则存储在磁盘上。因为只有存储在主存储器中的部分虚拟存储器可由处理器使用，这种虚拟存储技术将依赖程序内部访问存储器的局部化特性，在程序执行中只需整个虚拟存储器中的少量存储内容在主存储器中驻留。而当访问存储器的范围发生变化时，有必要把虚拟存储器的某些部分从磁盘调入主存储器，虚拟存储器的另外的部分，也能从主存储器传送回磁盘上。

地址转换机制以以下两种方式支持虚拟存储器：

1）把实际驻留在主存储器中的那部分虚拟存储器标记为无效，并建立起虚拟存储器驻留部分的虚拟——物理映射关系，把驻留部分的相应虚拟存储器地址，转换为对应物理存储器的地址。如果程序访问的虚拟地址对应于虚拟存储器未驻留的部分，将由于无效映射信息而引起异常。操作系统通过把未驻留部分从磁盘上读入到主存储器中，来处理这种异常，并根据需要更新地址转换表。在引起异常的原因排除以后，异常处理程序完成异常事件的处理，并返回原来的程序恢复执行。在后面的文章中将会看到，从异常处理程序返回后，这时要重新执行一次原来引起异常的指令，而该指令在后一次执行时自然会成功地完成。

2）地址转换机制通过收集驻留在主存储器中的虚拟存储器部分的使用统计信息来支持虚拟存储器，这些使用统计信息，在主存储器空间紧缺时，帮助操作系统决定可以将哪些部分传送回磁盘。

1.4.2 保护机制

为了支持多任务，对各任务实施保护是必需的。从80286开始，处理器就具备了保护机制。保护机制能有效地实现不同任务之间的保护和同一任务内的保护。

1. 不同任务之间的保护

保护的一个重要方面是应用程序之间的保护。通过把每个任务放置在不同的虚拟地址空间的方法来实现任务与任务的隔离，达到应用程序之间保护的目的。虚拟地址到物理地址的映射函数在每个任务中进行定义，随着任务切换，映射函数也切换。任务A的虚拟地址空间映射到物理地址空间的某个区域，而任务B的虚拟地址空间映射到物理地址空间的另外区域，彼此独立，互不相干。因此，两个不同的任务，尽管虚拟存储单元地址相同，

但实际的物理存储单元地址可以不同。

每个任务各有一组独立的映射表，即具有不同的地址转换函数。在 80386 上，每个任务都有自己的段表及页表。当处理器进行切换并执行新的任务时，这种任务切换的一个重要部分，就是为新任务切换任务的转换表。为了使操作系统与所有的应用程序相隔离，可以把操作系统存储在一个单一的任务中。然而，下面即将看到，在一个任务内操作的保护机制，更适合于保护操作系统，使其不被应用程序破坏。这种机制，使操作系统由所有任务共享，并且可在每一任务中对其进行访问，而且仍然保护了操作系统，使其不被应用程序破坏。这种保护操作系统的方法，是把操作系统存储在虚拟地址空间的一个公共区域，然后，再使每一任务按此区域分配一个同样的虚拟地址空间，并进行同样的虚拟——物理地址映射。各个任务公用的这部分虚拟地址空间，被称为全局地址空间。

仅由一个任务占有的虚拟地址空间部分，即不被任何其他任务共享的虚拟地址部分，称为局部地址空间。局部地址空间包含的代码和数据，是任务私有的，需要与系统中的其他任务相隔离。

每个任务中有不同的局部地址空间。因此，两个不同的任务中，对同一虚拟地址的访问，实际上转换为不同的物理地址。这就使操作系统对每个任务的存储器，可以赋予相同的虚拟地址，仍然保证任务的隔离。另一方面，对全局地址空间中同一虚拟地址的访问，在所有任务中都转换为同样的物理地址，从而支持公共的代码及数据的共享，例如对操作系统的共享。

2. 同一任务内的保护

在一个任务之内，定义有四种执行特权级别，用于限制对任务中的段进行访问。按照包含在段中的数据的重要性和代码的可信程度，给段指定特权级别。把最高的特权级别分配给最重要的数据段和最可信任的代码段。具有最高特权级别的数据，只能由最可信任的代码访问。给不重要的数据段和一般代码段分配较低的特权级别。具有最低特权级别的数据，可被具有任何特权级别的代码访问。

特权级别用数字 0、1、2 和 3 表示，数字 0 表示最高特权级别，而数字 3 表示最低特权级别，即数字较大的级别具有较低的特权。为了避免模糊和混淆，在比较特权级别时，不使用“大于”或“小于”这样的术语，而使用“里面”或“内层”这样的术语表示较高特权级，级别的数字较小；使用“外面”或“外层”这样的术语表示较低特权级别，级别的数字较大。0 级为最内层的特权级别，3 级为最外层的特权级别，按这样的表示方法，四种特权级的层次关系如图 1-2 所示（图中右边的数字为特权级）。

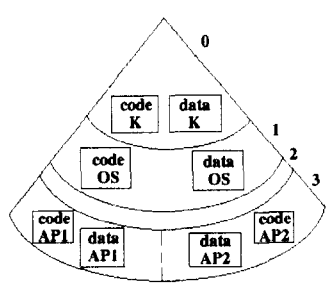

图 1-2