电脑编程技巧与维护月刊

圆圆圆年第 苑期　　总第 苑猿期（员怨怨源年 苑月创刊）　　每月 猿日出版

名誉社长　张　琪
社　　长　孙茹萍
副 社 长　毕研元
总　　编　王路敬
执行主编　叶　永
副 主 编　杨林涛
编　　辑　程　芳　管逸群
　　　　　张　涛　司维清
公关部主任　黄德琏
出版发行部　毕　波
编辑出版　《电脑编程技巧与维护》
　　　　　杂志社
法律顾问　佟秋平（北京亿中律师事务所雪）
主办单位　中国信息产业商会
社　　址　北京海淀区学院南路 远源号
　　　　　吉安大厦 源圆苑室
投稿信箱
[illegible]
编辑部信箱
[illegible]
网　　址
[illegible]
邮　　编　员园园园愿员
电　　话　（园员园）[illegible]
传　　真　（园员园）[illegible]
照　　排　《电脑编程技巧与维护》
　　　　　杂志社电脑排版部
印　　刷　北京巨龙印刷厂
订 阅 处　全国各地邮电局
国内总发行　北京报刊发行局
邮发代号　愿圆-苑缘
刊　　号　[illegible]
广告许可证　京海工商广字 圆园愿苑
全年定价　怨远.园园元
每期定价　愿.园园元



新技术追踪

猿 美国在线、希捷起用新晶片等（员缘篇）

编程与应用起步

缘 汇编语言程序设计系列讲座（一）
苑 [illegible] 控件在 [illegible] 中的应用
怨 [illegible] 应用软件框架
员远 [illegible] 在数字图像处理中的几点技巧与经验
员怨 用 [illegible] 编写界面的几个技巧
圆员 用 [illegible] 和 [illegible] 构建 [illegible] 层客户/服务器系统

编程语言

圆圆 用 [illegible] 开发 [illegible] 的方法
圆缘 [illegible] 中 [illegible] 单元的一个错误及修正
圆远 防止程序开启多个实例的终结篇
圆苑 [illegible] 的硬件端口读写实现与实例
猿园 防止 [illegible] 自动化服务器的多次加载

专家论坛

猿猿 从程序员的角度看 [illegible] 和硬连接
猿怨 使用 [illegible]

需要全本请访问 www.ertongbook.com

《电脑编程技巧与维护》杂志社授权佟秋平律师声明

《电脑编程技巧与维护》杂志社出版的《电脑编程技巧与维护》期刊，以荟萃国内外电子计算机先进信息，以领先性、实用性而享誉电脑报刊市场。但是近来《电脑编程技巧与维护》杂志社接到许多读者询问："重庆科潮电子图书有限公司"在全国多种报刊上刊登邮购本社杂志 [illegible] 年合订本的广告。已有一些读者将定购款汇至"重庆科潮电子图书有限公司"后却未得到本社 [illegible] 年合订本，且与"重庆科潮电子图书有限公司"无法联系。"重庆科潮电子图书有限公司"的行为严重影响了《电脑编程技巧与维护》杂志社的声誉，也损害了读者的权益。为此，本律师郑重声明：

《电脑编程技巧与维护》杂志社的发行和销售渠道是报刊发行局和各地新华书店及部分代理书店。《电脑编程技巧与维护》杂志社并未授权"重庆科潮电子图书有限公司"刊登邮购广告。敬告广大读者如需订阅《电脑编程技巧与维护》杂志可向上述代销商订购，也可直接向本杂志社订购。希望广大读者提高警惕，辨清真伪。

特此声明

《电脑编程技巧与维护》杂志社法律顾问：佟秋平律师

[illegible] 年 [illegible] 月 [illegible] 日

来检测和隔离内存泄漏

可视化专栏

[illegible] 通用多功能计算机监控系统测试软件的设计与开发

[illegible] 在 [illegible] 中实现将应用程序的图标加入到 [illegible] 的系统托盘中

[illegible] 在 [illegible] 中窗口界面的工具条上加各种控制

[illegible] 中 [illegible] 的多种实现方法

[illegible] 利用 [illegible] 接口实现 [illegible] 外壳的上下文菜单操作

数据库

[illegible] 用 [illegible] 实现对 [illegible] 数据库的访问

网络技术

[illegible] 聊天室的四种实现方案及其比较 [illegible]

[illegible] 通过 [illegible] 共享远程数据

[illegible] 用数据报套接字实现控制远程计算机

[illegible] 用 [illegible] 实现 [illegible] 新邮件检测通知程序

图形图像处理

[illegible] 用 [illegible] 处理数据库中图象的总结

[illegible] 为图片加下雪效果 —— 图片处理技术之二

[illegible] 用 [illegible] 处理数字图象

计算机维护

[illegible] 在 [illegible] 中对文本的操作

[illegible] 利用 [illegible] 消息向 [illegible] 发送字符串命令

计算机安全

[illegible] 利用 [illegible] 技术防止 [illegible] 地址盗用

[illegible] 对《再论制作硬盘逻辑锁》的补充

[illegible] 中病毒预警的设想及实现

博士信箱

[illegible] 从中文 [illegible] 上网常见问题及其处理

本期信息窗："北京腾图电子出版社有限公司"等 [illegible] 篇，分别见 [illegible] 页。

公　告

应广大读者的要求，本刊将于苑月推出网站，网站内容主要包括杂志简介、来稿要求、精品文章、代码交流园地、稿件查询、联系信箱、公告栏和读者留言等。在网站刚开始运行时，难免有不尽如人意的地方，衷心希望广大作者和读者能够提出宝贵意见，我们将会不断调整和完善，力争做的更好。在此，我们杂志社的全体同仁也感谢您长期以来对我们的关心、爱护和支持，更希望继续得到您的支持和关爱，我们会努力把杂志和网站做的更好以回报您。网址：澡澡澡援责则赠遭糟援糟燥皂援糟灶

《电脑编程技巧与维护》杂志社

美国在线、希捷起用新晶片

日前，捷威（郧葬贼藻憎葬赠）公司与美国在线将宣布，在今年下半年推出的基于互联网的产品中，他们将使用一种新型的处理器和孕蚤灶怎曾操作系统。这两家公司将打破英特尔和微软在市场上的垄断地位。新型的处理器晶片据说比英特尔的晶片耗能更低、更便宜。据透露，希捷公司将在其圆园园个零售商店中推出该产品，美国在线将在其用户中推销该产品。但两家公司都未透露新产品的最后定价。

汉王公司推出了新款字符识别软件——名片通

日前，北京汉王科技公司针对扫描仪推出了一款新的韵悦砸字符识别软件——名片通。该产品可以借助于任何品牌和型号的扫描仪，将成百上千的名片“一扫而光”，并存储到计算机或掌上电脑中。据介绍，这种名片识别软件对汉字的的识别率已达到怨缘豫以上。

韵则葬糟造藻研制成功新型文件管理系统

韵则葬糟造藻日前在其公司总部发布了一套新软件：互联网文件系统（陨云杂）软件。在会议现场，韵则葬糟造藻老总甚至宣称视窗操作系统用于管理和共享程序文件早已过时了。他说陨云杂将完成取代视窗系统的最后工作。互联网文件系统软件根据设计要求，可以将文件从计算机操作系统中分离出来，这意味着可以利用网络浏览器来查找文件，然后在任何一种电脑上打开。这套软件可以管理多达员缘园种格式的文件，并且不用改变原有文件就能自动生成新版本的文件。在韵则葬糟造藻公司最近发布的韵则葬糟造藻愿蚤数据库软件中就包含有这套软件。

酝蚤糟则燥泽燥枣贼版陨耘存在重大缺陷

酝蚤糟则燥泽燥枣贼公司敦促使用陨耘浏览器的用户最好立即将允葬增葬功能设置成“无效”。这是因为已有事实表明，几乎所有的酝蚤糟则燥泽燥枣贼用户使用的陨耘，由于允葬增葬虚拟机的缺陷，将会使用户处于安全漏洞之下。安全漏洞的内容为，“允葬增葬粤责责造藻贼只能在安装了粤责责造藻贼的主机上进行通信”这一安全限制在某种情况下会遭到破坏。如果这一安全漏洞被恶意利用的话，就有可能导致企业内部网（陨灶贼则葬灶藻贼）网站上的企业秘密被盗。避免该安全漏洞的方法首先是将陨耘的允葬增葬功能设置为无效。

中文语音浏览器诞生

日前，由福建省海峡天音信息技术有限公司开发的“海天易说”中文语音浏览器在京发布，通过该浏览器，用户可以通过口述语音命令的方式无障碍地上网冲浪。该软件以微软开发的网络浏览器陨耘核心组件为基础，同时集成了海峡天音公司一流的语音识别与合成技术以及机器翻译技术。

致力于降低网站成本 韵责藻灶宰葬增藻推出新软件

韵责藻灶宰葬增藻推出一项新产品——宰藻遭杂藻则增藻则软件，该软件可以降低用户制作个人网站的费用。该公司希望此举可以吸引更多的客户。该软件不仅可以降低网站的设备成本，还可以提高网站的运做性能，从而使页面的访问量得到增长。该软件在低配置的电脑上也可以运行，它可以把网页分成一个个小部分，然后储存在电脑的缓存里。它还可以借助服务器建立用户交流系统并保存用户的个人信息。宰藻遭杂藻则增藻则软件将于今年第三季度批量面市。

陨月酝原阅月圆通用数据库捆绑 砸藻凿匀葬贼

日前，陨月酝公司宣布将免费提供可运行于砸藻凿匀葬贼网络服务器上的阅月圆通用数据库（远援缘）版本的开发许可，以便使更多的杂怎责藻则公司享受阅月圆通用数据库强大的功能所创造的经济效益。该软件将随一个预定的粤责造糟蚤凿杂燥枣贼更新版本一起提供，其中包括阅月圆杂怎曾燥灶版中的……和……

微软为其安全漏洞发布补丁程序

日前，微软公司发布了一个安全补丁程序，希望以此改善使用微软视窗操作系统的电脑易受拒绝服务攻击的现状。攻击者通过在电脑或网站服务器上植入一段代码，并通过这段代码让这些计算机同时向一个网站发出数以千计的电子邮件，使该网站的电脑运行速度大大下降。此前微软公司曾承认视窗操作系统容易受到拒绝服务的攻击。

砸[illegible]推出针对微软新软件

日前，砸[illegible]公司的网站上出现了一项新的免费下载软件，它可以更灵活地处理多种不同形式的音乐和视频游戏。砸[illegible]网站上的娱乐中心包括一个[illegible]娱乐咨讯站，能将各种[illegible]音乐转换成计算机文件，如[illegible]和[illegible]媒体等格式。同时，新软件在流媒体技术方面也有长足的进步，使音乐和视频文件下载的时间缩短了许多，清晰度也有所提高。砸[illegible]发布的新软件主要针对其竞争对手微软。

微软更新[illegible]开发版工具

微软公司发布了新的开发工具包，用于构建基于微软[illegible]的工作流，以及整合[illegible]消息平台和[illegible]的协作应用软件。[illegible]版本包括[illegible]和[illegible]的工作流设计者[illegible]以及一个[illegible]拷贝。该产品中除了有[illegible]的开发版外，还包括服务器软件，这样开发者在展开使用前就可以构建并测试他们的应用软件。[illegible]将于[illegible]月底前上市，价格为[illegible]美元。

微软公布[illegible]的错误的源码

此前的一段时间里，这些代码一直被秘密地保管在[illegible]园区。自上周开始就有传言说微软可能会公布这些代码并将其作为与司法部达成协议的一部分。这次提前公布代码被看作是微软为了防止被分解而采取的抢先行动。之后微软还会公布[illegible]中由编码机器人所产生的源代码，这也等于微软承认他们使用编码机器人产生代码，而过去很长时间以微软一直否认这个事实。

[illegible]测试版已经在网上出现

美国在线公司发布消息称，该公司发现其最新版本的客户端软件[illegible]已在互联网上出现，虽然该公司并未准备立刻为用户提供这套软件。该公司同时称，目前在网上发布的[illegible]软件还只是一个测试版本，它据说是由一位名叫肯顿[illegible]的人首先在网上发布的，用户可以在[illegible]网站上下载。肯顿称自己这样做的目的是希望美国在线公司加强安全防卫措施。

[illegible]运公司开发[illegible]用垂直磁记录型阵列媒体

在“[illegible]运科技展”上，[illegible]运公司向外界透露说，他们正在开发用于硬盘装置[illegible]的新型记录媒体——垂直磁记录用阵列媒体。它通过在[illegible]中埋入[illegible]合金粒子形成排列，使[illegible]个粒子记忆[illegible]。此次展出的是在[illegible]见方的区域内，按[illegible]的间隔埋入直径为[illegible]的[illegible]合金粒子。其记录密度相当于[illegible]寸。媒体阵列的形成，采用了电子束的直接描画法。但是，该记录媒体还未能实现记录和再生。

以色列高密度数据存储技术研究有新突破

以色列魏茨曼科学研究院日前公布，该院材料部沙戈乌教授领导的研究小组应用独特的分子处理技术为高密度、长期数据存储开辟了一条新路，同时为纳米存储器的诞生提供了技术依据。科学家认为，利用这一技术可开发存储更多音乐的[illegible]盘。魏茨曼科学院称，他们所采用的结构和化学复合方法不仅为未来纳米器件获得精确控制打下了基础，也为超强存储能力的纳米器件的诞生铺设了一条潜在的道路。

安森美半导体推出[illegible]风扇转速控制器

安森美半导体[illegible]最新推出的[illegible]脉宽调制[illegible]风扇转速控制器，不仅具备故障检测功能，而且能配合温度转速控制设计，将任何标准双线无电刷直流风扇，提升成为智能型风扇。产品数据资料与其它技术信息可从以下网址获取[illegible]

硬盘厂商将推出新型[illegible]硬盘

日前，昆腾公司宣布推出[illegible]硬盘，采了通用的[illegible]接口，[illegible]硬盘每秒最快可以传输[illegible]的数据。昆腾公司持有该技术的专利，早期的[illegible]和[illegible]规格也是昆腾公司提出的。不同的厂家对于[illegible]规格的称呼有些不同，包括[illegible]或者[illegible]业界希望[illegible]规格的产品能够迅速被广泛接受，[illegible]年昆腾公司推出[illegible]规格，但由于兼容性问题，直到今年成为通用规格。英特尔的新的[illegible]芯片组将支持[illegible]规格，一些厂家计划在[illegible]主板上配置[illegible]规格的接口。其它硬盘厂家也即将推出支持[illegible]规格的硬盘。

[illegible]等公司推出基于[illegible]的统一消息接发

[illegible]和[illegible]公司日前宣布推出首个开放的基于[illegible]的统一通信[illegible]系统，该系统是为服务供应商向最终用户提供增强的消息接发和呼叫管理服务设计的。[illegible]是统一消息接发的扩展，给入站通信和出站通信增加了实时呼叫管理。[illegible]还是基于[illegible]的、提供集中服务[illegible]的体系结构。而统一消息接发目前还是传统专用的，不是基于[illegible]的。

汇编语言程序设计系列讲座（一）

李玉珏 徐 旭

编者按

本系列讲座的目的是帮助读者学习汇编语言程序设计，达到入门和提高。讲座共分四讲：第一讲 介绍一个完整的源程序及其运行；第二讲 分支、循环及子程序设计举例；第三讲 汇编语言程序的调试；第四讲 汇编语言程序举例。每讲前，介绍该讲的主要内容，突出一个重点。每讲后给出要求，指出读者应掌握的知识。讲座汇集了经验，解答了学习汇编语言经常遇到的一些问题。讲座通过展示实例，通俗讲解汇编语言程序设计的方法和技巧。

第一讲 介绍一个完整的源程序及其运行

为了对汇编语言程序有个总体的了解，本讲给出一个完整的汇编语言程序样例并说明上机运行操作的方法。

一个完整的源程序

下面通过一个简单的源程序样例来说明一个完整的汇编语言程序的结构和格式。

例：将ASCII码转换为十六进制码

在内存缓冲区BCDBUF存放着9个字节的用ASCII码表示的十进制数。每一个字节中存放两位ASCII码，要求把它们分别转换为十六进制码存放在ASCBUF缓冲区。高位的ASCII码转换完后顺放在地址较高的单元。并将ASCII码转换为十六进制码的结果显示出来。

```
DATA    SEGMENT                          ;数据段开始
BCDBUF  DB  25H, 41H, 30H, 16H, 38H, 79H, 15H, 31H, 73H
                                         ; 9个字节10进制数
COUNT   EQU  $-BCDBUF
; $是当前地址计数器的值, 这里 COUNT=9
ASCBUF DB 18 DUP(?) ; 用于存放转换为 ASCII 码的结果 DB '$'
DATA    ENDS                             ;数据段结束
STACK   SEGMENT  PARA  STACK  'STACK'    ;堆栈段开始
        DW  64  DUP(?)                   ;为堆栈段保留64个字空间
STACK   ENDS                             ;堆栈段结束
CODE    SEGMENT                          ;代码段开始
        ASSUME  CS: CODE, DS: DATA, ES: DATA, SS: STACK
; ASSUME 语句用来通知汇编程序, 各段寄存器与各程序段的对应关系
START : MOV  AX, DATA
        MOV  DS, AX                      ;设置数据段寄存器
        MOV  ES,  AX                     ;设置附加段寄存器
        MOV  SI, OFFSET  BCDBUF          ;设置数据指针
        MOV  DI, OFFSET  ASCBUF
        MOV  CX , COUNT                  ;设置计数器
        CLD                              ;串操作地址按增量方向
CYCLE : LODSB                            ;取一个字节(两位)10进制数
        MOV  BL, AL
        AND  AL, 0FH
        OR   AL , 30H                    ;低位10进制数→ASCII码
        STOSB                            ;存低位10进制数的ASCII码
        MOV  AL, BL
        PUSH  CX
        MOV  CL, 4
        SHR   AL, CL
        POP  CX
        OR    AL, 30H                    ;高位10进制数→ASCII码
        STOSB                            ;存高位10进制数的ASCII码
        LOOP  CYCLE                      ;循环
        MOV  DX, OFFSET  ASCBUF
        MOV  AH, 9
        INT  21H;                        ;将转换结果输出显示
        MOV  AX, 4C00H
        INT  21H                         ;返回DOS
CODE    ENDS                             ;代码段结束
        END  START                       ;程序结束
```

由此可见：

●源程序由若干个段组成。数据段含有程序要使用的数据，代码段含有程序的代码，堆栈段作为堆栈使用，附加段作为辅助的数据段来使用，若程序用到串操作指令，则目的操作数存放在附加段。

●每个段有个段名，以伪指令SEGMENT开始，以伪指令ENDS结束。

●各个段在源程序中的顺序可以任意，段的数目可依需要而定。

●而一个段又是由若干条语句组成。一条语句占一行，超过一行必须用续行符号驭。

●语句有两类：指令语句（如MOV、ADD等）和伪指令语句（如DW、END等），一个完整的源程序必须含有这两类语句。

●整个源程序以语句END作为结束。

上面，由例程序我们建立了对程序总体的认识。下面，结合该例程序，我们讲解汇编语言中的符号、表达式和伪指令的概念。

汇编语言中的符号

一个完整的汇编语言语句由下列几部分组成：标识符、保留字、界符、常数和注释，所有这些都称为符号。

● 标识符：是由程序员建立起来的，有特定意义的字符序列。如例程序中的BCDBUF、ASCBUF、CYCLE、START、DATA、STACK、CODE等。

● 保留字：是在汇编语言的中预先保留下来的，具有固定意义的符号。凡是指令助记符、伪操作命令、寄存器名和运算符都是保留字。保留字不能作标识符用。如例程序中的MOV、DW、DB、SEGMENT、ENDS、ASSUME、END等。

● 界符：界符作为一个程序中或者一条指令中两个部分的分

需要全本请访问 www.ertongbook.com

割符号用。如例程序中的‘,’、‘;’、‘:’、‘?’、‘:’等。

● 常数：出现在源程序中的固定值。如例程序中的 10和 0，分别指令中的立即数 0。

● 注释：为了便于程序的理解和阅读，常常加上注释。注释要用分号‘;’打头。在汇编过程中，注释被略去而不做处理。

(2)表达式

表达式是由操作数和操作符组成的序列。每一个表达式在汇编时都产生一个值。如例程序中的 $-$ 是操作符，$-$ 后面的 $-$ 是操作数，$-$ 和 $-$ 组成的表达式的值为 $-$。

(3)伪指令

伪指令没有对应的机器指令，只是在汇编时为汇编程序提供汇编信息。如例程序中 $-$、$-$、$-$、$-$、$-$、$-$、$-$、$-$等。

2.汇编语言源程序的运行

源程序经过汇编和连接处理后，生成一个可执行文件。运行这可执行文件时，操作系统便把这个文件装入并给各逻辑段分配相应的物理段。在 DOS把控制权转给该程序时，便设置了代码段寄存器和指令指针寄存器。

汇编语言源程序的运行步骤如下：

(1)编辑源程序。

利用文本编辑工具（如 EDIT）来编辑源程序，生成一个汇编语言源程序的纯文本文件（汇编语言源程序的扩展名应为 ASM）。例如：将上述例源程序建立成一个名为 BCDTOASC.ASM 的纯文本文件。

(2)汇编源程序。

在 DOS下，利用汇编器（如 MASM或 ALASM）来汇编源程序生成扩展名为 OBJ的目标代码程序。汇编源程序主要完成如下工作：

● 检测源程序中各语句是否有语法错误。若有错误，就给出错误信息。程序设计人员依据给出的错误信息进行改正，否则不能正确汇编。如果没有错误，便完成汇编，就能生成目标代码文件。

● 实现宏功能。若源程序中有宏定义和宏调用，汇编过程中便完成宏展开，并把宏展开的各语句嵌入源程序的相应位置。

● 生成目标代码文件。汇编器把源程序中各指令语句汇编成相应的目标代码。

其汇编操作步骤如下：

```
C>MASM
Microsoft (R) Macro Assemble Version 5.00
Copyright (c) maicrosoft Corp 1981 – 1985, 1987
Source filename [.ASM]
: BCDTOASC   指定源程序名 BCDTOASC.ASM
Object filename [BCDTOASC.OBJ]
: 按回车键，表示生成的目标代码文件名为默认值
Source listing  [NUL.LST]:        按回车键，表示不生成列表文件
Cross – reference[NUL.CRT]: 按回车键，表示不生成交叉参考文件
0  Warning  Errors
0 Severe  Errors
```

源程序 BCDTOASC.ASM经过汇编，生成名为 BCDTOASC.OBJ目标代码文件。

在汇编过程中，可产生三个文件：

● 目标文件（OBJ文件）：是二进制代码的目标程序文件。

● 列表文件（LST文件）：是把源程序的各语句与对应机器语言目标代码一一对应列出的程序清单文件。

● 交叉参考文件（CRF文件）：列出源程序中自定义的各个符号（如段名、过程名、常量名、变量名和标号等），并指明这些符号在源程序中定义位置和引用位置。

(3)连接目标程序。

利用连接器（如 LINK或 TLINK）实现目标代码程序和库函数代码的连接，生成可执行程序文件，其扩展名为 EXE。其连接操作步骤如下：

```
C> LINK
Microsoft (R) 8086 Object Linker  Version 3.05
Copyright (C) Microsoft Corp 1983, 1984, 1985.
Object Modules [.OBJ]: BCDTOASC  ;指定目标代码程序
Run File [BCDTOASC.EXE]:
;按回车键，表示生成的可执行程序名为默认值
List File[NUL.MAP]:        ;按回车键，表示不生成定位图文件
Libraries[.LIB]:           ;按回车键，表示不使用库文件
```

目标代码文件 BCDTOASC.OBJ经过连接器，生成可执行程序 BCDTOASC.EXE。

如果一个程序由若干个程序模块构成，那么连接程序可把各模块连接一起，形成一个完整的可执行文件。如：

```
C>LINK ABC1 +ABC2 +ABC3 , ABC
```

这是将 ABC1.OBJ、ABC2.OBJ、ABC3.OBJ三个目标代码文件连接后，生成一个可执行文件，存放在 ABC.EXE中。

(4)运行可执行程序。

汇编、连接后生成的 EXE文件，可在 DOS下键入文件名直接运行。

例如，运行例程序 BCDTOASC.EXE的操作如下：

```
C> BCDTOASC
```

显示器的屏幕上显示出 BCDTOASC.EXE的运行结果：

```
5214036183975113 37
```

3.本讲要求：

(1)了解汇编语句的特点和应用场合

(2)掌握汇编语言的寻址方式和指令系统

(3)了解一个完整的汇编语言源程序结构。

(4)了解一个汇编语言语句的组成。掌握汇编语言中的符号

(5)掌握汇编语言中的表达式和伪指令

(6)掌握汇编语言程序运行的步骤

4.本讲座推荐的参考书如下：

王路敬主编 徐旭、李玉珏等编著 《高档微机硬件实用技术基础》中国水利水电出版社，1995

杨季文等编著 《面向21世纪汇编语言程序设计教程》清华大学出版社，1995

（收稿日期：1997年5月25日）

控件在 中的应用

唐苑　吴凡　于辉

是美国 公司的办公自动化软件 的 家族中专门用于制作演示文稿的优秀工具软件。它以简洁明快的风格成为用户制作在学术交流、工作汇报、产品展示等场合使用的演示文稿的常用软件之一。用户使用其中的一般功能（如编辑幻灯演示文稿的基本方法、多媒体对象的添加、动画制作、超级链接等）就能制作出具有多媒体对象、操作方便的演示文稿，但美中不足的是演示内容与过程过于呆板，缺乏良好的人机交互界面。针对这一问题，使用 自身提供的 控件，就可以以一种精细的方式在幻灯片放映过程中与用户交换信息，建立良好的交互关系。

中的 控件与在 、 之类的编程语言中的控件很相似，它包括用来创建自定义程序、开关按钮、对话框、文本框和其他控件等。它们存放于“控件工具箱”中，以下是几种主要的控件：

控件类型	说明
复选框	可以通过选择或者清除来打开或者关闭的选项
文本框	可以键入文本的框
命令按钮	单击时可激活某个动作的按钮
选项按钮	用来从一组选项中选择其中某个选项的按钮
列表框	包含项目列表的框
组合框	具有组合框文本框
切换按钮	单击这类按钮时，它会保持按下状态，再次单击时还原
数值调节钮	可附加到文本框的按钮，如果要增加或者减少数值，单击相应的箭头
滚动条	当单击滚动肩头或者拖动滚动框是，可以滚动数值列表的控件
标签	另外添加的文本，用于提供有关幻灯片或幻灯片上控件的信息
图像	嵌入图片或图形的控件
其他控件	附加的 控件列表

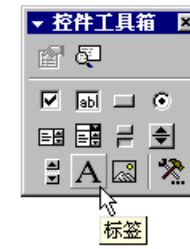
在 中应用 控件的一般过程如下：

①打开要添加 控件的演示文稿；

②将鼠标指向“视图”菜单中的“工具栏”，再单击“控件工具箱”；



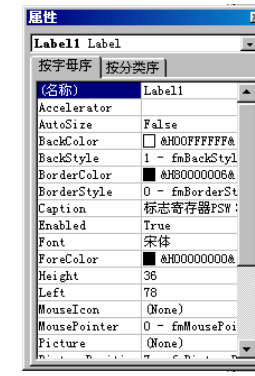
③在“控件工具箱”中选择要添加的控件按钮；



④在幻灯片上将控件拖动成所需的大小；

⑤如果要设置控件的属性，则在控件上单击鼠标右键选择“属性”，并进行各选项的设置；





⑥右键单击快捷菜单中的“查看代码”，然后在“ 编辑器”中写入自定义 控件行为的宏代码，以便对刚才设定的各 控件进行激活；

⑦放映幻灯片，可发现在同一张幻灯片所呈现的内容更加丰富，用户通过操作各控件，方便自如地实现了人机交互。

下面举例说明 控件在 中的应用：

针对汇编语言教学中讲解标志寄存器各标志时比较抽象，制作了介绍标志寄存器的幻灯片，如图：



其中使用了 个命令按钮控件、 个文本框控件和 个标签控件，通过选择不同的标志位对应的按钮，在文本框中将显示相应的介绍。添加的自定义宏代码如下：

```
Dim CHOICE As Integer
```

```
Private Sub CommandButton1_Click(  )  )  )  )  )  )  )  )  )
TextBox1. Text  =  ″″
CHOICE  =  0
TextBox1_GotFocus
End Sub
Private Sub CommandButton2_Click(
TextBox1. Text  =  ″″
CHOICE  =  1
TextBox1_GotFocus
End Sub
Private Sub CommandButton3_Click(
TextBox1. Text  =  ″″
CHOICE  =  2
TextBox1_GotFocus
End Sub
Private Sub CommandButton4_Click(
TextBox1. Text  =  ″″
CHOICE  =  3
TextBox1_GotFocus
End Sub
Private Sub CommandButton5_Click(
TextBox1. Text  =  ″″
CHOICE  =  4
TextBox1_GotFocus
End Sub
Private Sub CommandButton6_Click(
TextBox1. Text  =  ″″
CHOICE  =  5
TextBox1_GotFocus
End Sub
Private Sub CommandButton7_Click(
TextBox1. Text  =  ″″
CHOICE  =  6
TextBox1_GotFocus
End Sub
Private Sub CommandButton8_Click(
TextBox1. Text  =  ″″
CHOICE  =  7
TextBox1_GotFocus
End Sub
Private Sub CommandButton9_Click(
TextBox1. Text  =  ″″
CHOICE  =  8
TextBox1_GotFocus
End Sub
Private Sub TextBox1_DblClick(ByVal Cancel As MSForms. Return-
Boolean)
TextBox1. Text  =  ″″
End Sub
Private Sub TextBox1_GotFocus(  )
TextBox1. Visible  =  True
Select Case CHOICE%
  Case 0
   TextBox1. Text  =  ″溢出标志, 超出目的操作数的范围溢出为 1, 否则为 0″
   TextBox1. Text  =  TextBox1. Text  +  Chr(10)  +  Chr(13)  +  ″作用: 表示带符号数的溢出″
  Case 1
   TextBox1. Text  =  ″符号标志, 负时为 1, 否则为 0″
   TextBox1. Text  =  TextBox1. Text  +  Chr(10)  +  Chr(13)  +  ″作用: 根据运算结果的正负来改变程序的流程″
  Case 2
   TextBox1. Text  =  ″零标志, 结果是零为 1, 否则为 0″
   TextBox1. Text  =  TextBox1. Text  +  Chr(10)  +  Chr(13)  +  ″作用: 根据运算结果为 0 否来改变程序流程″
  Case 3
   TextBox1. Text  =  ″辅助进位标志, 低 4 位向高 4 位产生进位为 1, 否则为 0″
   TextBox1. Text  =  TextBox1. Text  +  Chr(10)  +  Chr(13)  +  ″作用: 只用于十进制调整指令, 即压缩 BCD 码的″
   TextBox1. Text  =  TextBox1. Text  +  Chr(10)  +  Chr(13)  +  ″     加减调整指令″
  Case 8
   TextBox1. Text  =  ″奇偶标志, 1 的个数是偶数为 1, 否则为 0″
   TextBox1. Text  =  TextBox1. Text  +  Chr(10)  +  Chr(13)  +  ″作用: 用来检查 ASCII 符号的奇偶性是否正确″
  Case 4
   TextBox1. Text  =  ″进位标志, 最高有效位产生进位/ 借位为 1, 否则为 0″
   TextBox1. Text  =  TextBox1. Text  +  Chr(10)  +  Chr(13)  +  ″最高有效位: 8 位数为第 7 位; 16 位数为第 15 位″
   TextBox1. Text  =  TextBox1. Text  +  Chr(10)  +  Chr(13)  +  ″作用: (1) 表示无符号数的溢出″
   TextBox1. Text  =  TextBox1. Text  +  Chr(10)  +  Chr(13)  +  ″    (2) 实现多字节的加、减运算″
  Case 5
   TextBox1. Text  =  ″方向标志, 为 1 时 SI、DI 减量, 否则增量″
  Case 6
   TextBox1. Text  =  ″中断标志, 为 1 时允许中断, 否则关闭中断″
  Case 7
   TextBox1. Text  =  ″陷阱标志, 为 1 时单步方式操作, 否则正常″
    End Select
End Sub
```

以上程序在 宰[illegible]和 孕[illegible]环境下通过。

参考文献

员援东箭工作室编著 援[illegible]中文版程序设计 援北京：清华大学出版社，员怨怨苑

圆援何弘、刘伟华编著 援[illegible]中文版即学即会 援北京：中国城市出版社，员怨怨怨

（收稿日期：圆园园园年 源月 员缘日）

EXCEL 应用软件框架

缪国钧 葛晓霞

摘 要 文通过一个完整的利用 Excel 开发的应用软件的框架作为范例，介绍及演示了利用 Excel 作为应用软件开发平台的方法。

关键词 Excel 定制 Excel 应用软件开发

一、 引言

提起 Excel 许多人都知道是一个办公软件，是 Office 系列软件中的表格制作软件；提起实用软件开发，许多人想到的是 VB、VF、Delphi 等。实质上这两件事在一些情况下是统一的。因为，Excel 除了一般人所共知的功能以外，它还是一个非常优秀的软件开发平台。特别是在考虑一些要对大量数据进行统计分析而且应用比较专业、人机交互较简洁的应用软件时，采用 Excel 作为开发平台来开发应用软件将会是最明智的选择。当你选择了 Excel 作为你的应用软件开发平台时，你就可以操纵 Office 为 Excel 提供的一组面多种预建立、预测试的对象和强大的常用公式函数库；可以获得许多公司或个人为 Excel 开发的增强对象；还可以利用其它开发工具开发的动态链接库。在这些优秀特性的支持下，你可以实现方便、快捷、高效、可靠的应用软件开发。总之，可以毫不夸张地讲，Excel 属于 Office 最优秀的软件。

二、 Excel 的编程简介

VBA 是 Office 的应用软件编程环境，它是从 VB 中演化而来的。是 Excel 编程的主要工具。

VB 最初出现在 Excel 中，然后出现在其它 Office 应用程序中，作为一种编程控制环境的工具，可以通过 OLE 自动化与其它应用程序一起工作。1993 年，Office 允许其他厂家购买 VB 语言工具和环境版权，在自己的产品中使用，从而使 VB 世界出现了一个飞跃。因而你在 VB 中付出的心血将会在更多的软件应用和开发中得到回报。

VB 使用 VB 标准，对于熟悉 VB 语言的编程人员，将会在很短的时间内成为一个熟练的 VB 编程人员；而对于熟悉 VB 的编程人员，进入 VB 将更加容易。

VB 编程的最大特点和它的最大优点是它的面向对象编程（OOP）的思想和方法。它使用一个标准的“核心语言”和由应用程序提供的对象。因而在 Excel 的编程中你相当多的时间是在操纵各种对象。另一方面，VB 在新版本中不断改进与加强与基础（OLE）对象模型的交互作用。

VB 编程的另一强大生命力来源于它的可扩展性。VB 可以方便地调用系统的 API 和你自己开发的 API；可以方便地扩展你自己的类和对象。当然也可以方便地使用其他公司或个人的 API 和对象。事实上专业的 VB 开发离不开开发自己的 API 和类。

VB 编程的另一特点是它的代码能很方便地实现跨平台开发。Excel 本身就能在所有 Macintosh 平台上运行，同时又支持 Windows 开发环境和 Macintosh 平台。而且用 VB 开发的 Excel 应用工程可以方便地被其他开发工具使用，如 VB、Delphi、C++Builder 等。所以 Excel 既可以开发独立完整的应用软件，也可以开发一个大工程中的某个功能模块。

三、 一个演示应用软件框架的编程方法

Excel 编程的核心思想是面向对象编程（OOP）思想。在用 Excel 编程时您相当多的时间是在与各种对象打交道。下面我们来实现一个完整的应用软件框架。该框架有三张表：Sheet1、Sheet2 和一张隐藏表：Sheet3。在三张可见表中只做了保护操作，在 Sheet3 表中做了设置的存取修改操作。范例中有一个模块 Module1 实现启动初始化功能、程序退出时的 Excel 恢复功能及菜单设置功能。另外范例中还有多个对话框，实现人机交互。该框架已经演示和实现了许多常规的应用软件所具备的界面面要求。如：

1. 自己的主窗口标题——定制标题；
2. 自己的菜单——定制菜单；
3. 程序启动、初始化的可靠控制——启动初始化期间合理控制人机交互；
4. 系统功能的屏蔽和恢复（系统状态条、Ctrl+Break 键的屏蔽）——演示 API 函数及 API 的调用；
5. Excel 界面的定制（菜单栏、工具条、公式栏、状态条、工作簿、工作表的使用和控制）——演示 Excel 对象的使用和界面定制方法；
6. Excel 中数据的存取——演示 Range 对象的操纵；
7. 人机交互——通过索取密码和修改密码来演示；
8. 关于对话框——演示版权对话框；
9. 联机帮助——演示 Help 功能。

实现该软件框架可按下下列步骤开始：

1. 启动 Excel 生成一个新的 Excel 文档，命名为：Module1；
2. 在该文档中插入一个新表，并将其命名为 Sheet3；
3. 在 Sheet3 表中命名三个单元格：A1 命名为 Password，B1 命名为 MainTitle，C1 命名为 MainMenu。这三个单元格及以下的一个区域用于保存您对 Excel 所做的界面设置，以便我们的 Module1 运行

需要全本请访问 www.ertongbook.com

结束后恢复这些设置。

源程序在 ThisWorkbook 中定义一个区域，命名为 toolbar_range；在 Sheet1 中定义一个区域，命名为 toolbar_num；在 Sheet2 中定义一个区域，命名为 tool_bar。我们的 VBA 应用程序运行后，您将只能在这三个区域中输入数据。

编程打开 Excel 的 Visual Basic 设计窗口，在其中添加模块 VBA 应用程序，并输入代码。

还要在 Excel 的 Visual Basic 设计窗口中插入用户自定义窗体，窗体名称为 About，窗体标题为 About，窗体标签为 Label1，并作相应的设计。

存盘保存文件，然后关闭。

重新打开 VBA 应用程序，如果 Excel 询问是否运行宏时，请选择“启用宏”。如果没有意外将会出现一个被您控制了的软件界面。



详细内容请参见所附 Excel 文件：VBA 应用程序框架。用该应用软件框架为基础开发的应用软件，已经成功地应用于某电厂的化学车间 DCS 系统和报表制作系统的软件中。在 DCS 系统中，Excel 开发的软件与 DCS 系统协同工作，实现实时数据的采集和实时数据库的建立维护工作；同时与我们开发的实用语音报警软件通过 DDE 通信实现语音报警功能。在报表制作系统中，主程序由 VB 开发，Excel 开发的软件与主程序协同工作，实现实时数据和历史数据的报表制作、打印和人工采集数据数据库的建库、维护和报表制作打印工作。

当你要采用 Excel 开发一个应用软件时，可以直接在该框架下开发应用功能模块并对界面做一些简单的修改即可。假如要开发一个数据库应用软件，要做的主要是开发数据库操作的模块；而要开发一个财务应用软件就要开发相应的财务计算模块，如此等等。当然在这些模块中都可以使用 Excel 本身和其他公司或个人开发的函数、模块或 ActiveX 对象；也可以方便地使用 Excel 本身的各种对象和其他 DCS 的 ActiveX 对象，使你的软件轻轻松松地实现 DCS 的许多强大功能，当然也就轻轻松松地使你的软件具备了更专业、更可靠的特点。

四、 实用软件的框架范例

下面给出一个完整的应用程序框架 VBA 应用程序的主模块 VBA 应用程序的源代码如下（所有代码，包括用户自定义 VBA 的代码在 VBA 应用程序都可在 VB 窗口浏览修改）：

```
Option Explicit
Public bIsExit As Boolean, bIsMenu As Boolean
Public iTryCount As Integer
'以下常量供 API 函数调用
Public Const FWP_STARTSWITH = 0
Public Const FWP_CONTAINS = 1
Public Const SWP_HIDEWINDOW = &H80
Public Const SWP_SHOWWINDOW = &H40
Private Const GW_HWNDNEXT = 2
Private Const SW_RESTORE = 9
Private Const SPI_SCREENSAVERRUNNING = 97
Private Declare Function EnumWindows Lib "user32" _
  (ByVal lpEnumFunc As Long, ByVal lParam As Long) As Long
Private Declare Function GetWindowText Lib "user32" _
  Alias "GetWindowTextA" (ByVal hWnd As Long, _
  ByVal lpString As String, ByVal cch As Long) As Long '
Declare Function FindWindow Lib "user32" _
  Alias "FindWindowA" (ByVal lpClassName As String, _
  ByVal lpWindowName As String) As Long  '系统任务条控制
Declare Function SetWindowPos Lib "user32" (ByVal hWnd As
Long, _
  ByVal hWndInsertAfter As Long, ByVal x As Long, _
  ByVal y As Long, ByVal cx As Long, ByVal cy As Long, _
  ByVal wFlags As Long) As Long  '系统任务条控制
Private Declare Function SystemParametersInfo Lib "user32" _
  Alias "SystemParametersInfoA" (ByVal uAction As Long, _
  ByVal uParam As Long, lpvParam As Any, _
  ByVal fuWinIni As Long) As Long  '屏蔽 Ctrl+Alt+Del 键
Sub AUTO_OPEN()
  Dim toolbar_range As Object  '工具条的状态保存区域
  Dim toolbar_num As Integer
  Dim tool_bar As Variant
  iTryCount = 0  '密码输入计数
  bIsExit = False
  bIsMenu = False
  About.Show
  NotEdit
  Application.WindowState = xlMaximized
  RelaseWorkbook  '撤消工作簿保护
  '界面初始化开始
  Application.ScreenUpdating = False
  Application.Caption = "南京电力高等专科学校"
  ActiveWindow.Caption = ""
  If Application.DisplayFormulaBar = True Then
    Worksheets("SetUp").Range("formula_bar") = 1
  Else
    Worksheets("SetUp").Range("formula_bar") = 0
  End If
  If Application.DisplayStatusBar = True Then
    Worksheets("SetUp").Range("status_bar") = 1
```

```
    Else
        Worksheets("SetUp").Range("status_bar") = 0
    End If
    Application.DisplayFormulaBar = False
    Application.DisplayStatusBar = False
    toolbar_num = 1
Set toolbar_range = Worksheets("SetUp").Range("tool_bars")
    For Each tool_bar In Application.Toolbars
        If tool_bar.Visible = True Then
            toolbar_range.Offset(toolbar_num, 0) = 1
        Else
            toolbar_range.Offset(toolbar_num, 0) = 0
        End If
        If (toolbar_num <> 14 And toolbar_num <> 15 _
            And toolbar_num <> 16 _
            And toolbar_num <> 17 And toolbar_num <> 18) Then
            tool_bar.Visible = False
        Else
            toolbar_range.Offset(toolbar_num, 0) = 0
        End If
        toolbar_num = toolbar_num + 1
    Next tool_bar
    Application.ScreenUpdating = True
    DelExcelM
    AddMenuIn
    '界面初始化结束
    ReEdit
    Password.Show    '要启动密码
    ProtectWorkbook  '启动表的保护功能
    SysUnable
End Sub
Sub AUTO_CLOSE()
    Dim toolbar_range As Object
    Dim toolbar_num As Integer
    Dim tool_bar As Variant
    '界面恢复开始
    Application.DisplayAlerts = False
    If Worksheets("SetUp").Range("formula_bar") = 1 Then
        Application.DisplayFormulaBar = True
    End If
    If Worksheets("SetUp").Range("status_bar") = 1 Then
        Application.DisplayStatusBar = True
    End If
    toolbar_num = 1
Set toolbar_range = Worksheets("SetUp").Range("tool_bars")
    For Each tool_bar In Application.Toolbars
        If toolbar_range.Offset(toolbar_num, 0) = 1 Then
            tool_bar.Visible = True
        End If
        toolbar_range.Offset(toolbar_num, 0) = 0
        toolbar_num = toolbar_num + 1
    Next tool_bar
    Application.Caption = "Microsoft Excel"
    BackExcelM
    SysEnanle
    '界面恢复结束
    Application.DisplayAlerts = False
    ProtectWorkbook          '恢复工作簿保护
    ActiveWorkbook.Close savechanges:=False
    Application.Quit
End Sub
Sub NotEdit()        '控制界面,使用户不能交互
    Application.Interactive = False
    Application.ScreenUpdating = False
    Application.DisplayStatusBar = True
    Application.StatusBar = "正在刷新表格内容,请稍侯..."
End Sub
Sub ReEdit()         '恢复正常的用户界面交互
    Application.DisplayStatusBar = False
    Application.Interactive = True
    Application.ScreenUpdating = True
End Sub
Sub ProtectWorkbook()    '保护工作簿中的工作表
    On Error Resume Next
    Application.Goto Reference:="MyPrct1"
    Selection.Locked = True
    Selection.FormulaHidden = True
    Cells(1, 1).Select
    ActiveSheet.Protect
    Application.Goto Reference:="MyPrct2"
    Selection.Locked = True
    Selection.FormulaHidden = True
    Cells(1, 1).Select
    ActiveSheet.Protect
    Application.Goto Reference:="MyPrct"
    Selection.Locked = True
    Selection.FormulaHidden = True
    Cells(1, 1).Select
    ActiveSheet.Protect
End Sub
Sub RelaseWorkbook()     '撤消工作表保护
    Sheets("Sheet1").Select
    ActiveSheet.Unprotect
    Sheets("Sheet2").Select
    ActiveSheet.Unprotect
    Sheets("Sheet3").Select
    ActiveSheet.Unprotect
End Sub
Sub SysUnable()      '取消有关系统功能
    Dim ret As Integer
    Dim pOld As Boolean
    Dim rtn As Long
    rtn = FindWindow("Shell_traywnd", "")  '获得窗口
    Call SetWindowPos(rtn, 0, 0, 0, 0, 0, SWP_HIDEWINDOW)
'隐藏任务条
    ret = SystemParametersInfo(SPI_SCREENSAVERRUNNING, _
        True, pOld, 0)    '使 Ctrl+Alt+Del 无效
End Sub
Sub SysEnanle()      '恢复系统
    Dim ret As Integer
    Dim pOld As Boolean
    Dim rtn As Long
    rtn = FindWindow("Shell_traywnd", "")  '获得窗口
        Call SetWindowPos(rtn, 0, 0, 0, 0, 0,
```

```
SWP_SHOWWINDOW) '显示任务条
    ret = SystemParametersInfo(SPI_SCREENSAVERRUNNING, _
        False, pOld, 0)    '使 Ctrl + Alt + Del 有效
End Sub
Sub AddMenuIn()            '添加自定义菜单
    On Error GoTo del
    MenuBars("工作表菜单栏"). Menus. Add _
        Caption: = "Test 帮助 . . . ", Before: = 3
    Application. MenuBars("工作表菜单栏"). Activate
    MenuBars("工作表菜单栏"). Menus("Test 帮助 . . . "). MenuItems. Add _
        Caption: = "帮助", _
        OnAction: = "MyHelp", Before: = 3
    MenuBars("工作表菜单栏"). Menus("Test 帮助 . . . "). MenuItems. Add _
        Caption: = "关于 . . . ", _
        OnAction: = "AboutM", Before: = 3
    MenuBars("工作表菜单栏"). Menus. Add _
        Caption: = "Test 维护", Before: = 2
    Application. MenuBars("工作表菜单栏"). Activate
    MenuBars("工作表菜单栏"). Menus("Test 维护"). MenuItems. Add _
        Caption: = "密码设置", _
        OnAction: = "FrameCtr. SetPassword", Before: = 2
    Exit Sub
del:
    DeleteIn
End Sub
Sub DeleteIn()             '删除自定义菜单
    On Error Resume Next
    MenuBars("工作表菜单栏"). Menus("Test 帮助 . . . "). Delete
    MenuBars("工作表菜单栏"). Menus("Test 维护"). Delete
End Sub
Sub SetPassword()    '菜单函数
Password. Show
End Sub
Sub AboutM()               '菜单函数
About. About
End Sub
Sub MyHelp()               '菜单函数
    Application. Help "Windows. hlp", 0    '调用 WINDOWS 帮助
End Sub
Sub DelExcelM()            '删除不需要的 EXCEL 菜单
Dim i As Integer
    On Error Resume Next
'删除除 '文件 '之外的所有菜单
    MenuBars("工作表菜单栏"). Menus("帮助"). Delete
    MenuBars("工作表菜单栏"). Menus("窗口"). Delete
    MenuBars("工作表菜单栏"). Menus("数据"). Delete
    MenuBars("工作表菜单栏"). Menus("工具"). Delete
    MenuBars("工作表菜单栏"). Menus("格式"). Delete
    MenuBars("工作表菜单栏"). Menus("视图"). Delete
    MenuBars("工作表菜单栏"). Menus("编辑"). Delete
    MenuBars("工作表菜单栏"). Menus("插入"). Delete
'删除 '文件 '菜单中除 '打印预览 '和 '打印 '菜单项之外的所有项
    MenuBars("工作表菜单栏"). Menus("文件"). MenuItems("新建(& N). . . "). Delete
    MenuBars("工作表菜单栏"). Menus("文件"). MenuItems("打开(& O). . . "). Delete
    MenuBars("工作表菜单栏"). Menus("文件"). MenuItems("保存(& S)"). Delete
    MenuBars("工作表菜单栏"). Menus("文件"). MenuItems("另存为(& A). . . "). Delete
    MenuBars("工作表菜单栏"). Menus("文件"). MenuItems("保存工作区(& W). . . "). Delete
    MenuBars("工作表菜单栏"). Menus("文件"). MenuItems("页面设置(& U). . . "). Delete
    MenuBars("工作表菜单栏"). Menus("文件"). MenuItems("打印区域(& T)"). Delete
    MenuBars("工作表菜单栏"). Menus("文件"). MenuItems("发送(& D)"). Delete
    MenuBars("工作表菜单栏"). Menus("文件"). MenuItems("属性(& I)"). Delete
    For i = 5 To 15
        MenuBars("工作表菜单栏"). Menus("文件"). MenuItems(i). Delete
    Next i
End Sub
Sub BackExcelM()     '恢复 EXCEL 缺省菜单设置
    MenuBars("工作表菜单栏"). Reset
End Sub
```

(收稿日期：[illegible])

北京腾图电子出版社有限公司

腾图文教电子发展有限公司自[illegible]年[illegible]月成立以来，经过三年多的努力，从全国申请成立电子出版机构的上百家单位中脱颖而出，经中宣部和国家新闻出版署批准，与另外七家大型国有企业（包括北大方正、清华同方、万方数据、中国大恒等）一起，成为首批获得电子出版权的八家企业。

腾图自成立以来，已经累计开发、出版了[illegible]多种教育、文化、科学和娱乐类出版物。其中，与世界著名出版公司英国麦可米伦公司合作，出版的《经济学前沿问题》系列丛书在国内外出版界产生了巨大反响。腾图自主开发、出版、发行的《腾图[illegible]风暴》系列产品一经发布就受到国内软件行业和电脑爱好者的瞩目。腾图还是中国最早涉足电脑游戏业的公司之一，所属的“大唐”、“八爪鱼”、“火狐狸”、“魅力”等工作室，自主开发、出版、发行的《烽火三国》、《水浒英雄传》、《火狐狸》、《西游记之美猴王》等游戏软件更是远销到东南亚及台湾地区，取得相当大的成功。

《腾图[illegible]风暴》系列产品简介

一、[illegible]》（双[illegible]配套手册）

二、《[illegible]游戏宝典》（单[illegible]配套手册）

三、《[illegible]多媒体及模拟器工具大全》（单[illegible]配套手册）

四、《[illegible]网络安全卫士》（单[illegible]配套手册）

五、《[illegible]数据库及应用工具大全》（单[illegible]配套手册）

北京腾图电子出版社出版发行　电话：[illegible]

VC++在数字图像处理中的几点技巧与经验

陆宏伟

Visual C++作为一个功能非常强大的可视化应用程序开发工具，是计算机界公认的最优秀的应用开发工具之一。Visual C++提供的基本类库MFC使得开发Windows应用程序比以往任何时候都要容易。Visual C++提供的各种函数、指针操作和直接对硬件操作使得图像处理速度较快，专门为Visual C++设计的DirectDraw和DirectShow技术可以使开发人员在Windows环境下比较容易地完成图像图形的高性能处理和显示。用Visual C++编制的应用软件无论在处理速度和图像的显示等方面都表现出较高的效率，因此在编制基于Windows的各类数字图像处理程序时，Visual C++无疑是程序员进行图像处理的最佳选择。

本文主要介绍了本人在用Visual C++进行图像基本处理时获得一些技巧与经验，希望与广大爱好图像处理的编程者分享。

一、模板Template的使用

对于图像处理应用软件来说，图像数据的管理是至关重要的，在《电脑编程技巧与维护》杂志2000年第1期上，本人介绍了一个封装的通用图像基类CImageBase，可完成256色格式图像数据的管理和一些基本的图像处理功能。但是该CImageBase类只是对位图进行处理，即操作的数据类型为无符号8位字符型BYTE，但是在进行边缘检测、相关跟踪、快速傅立叶变换等图像处理中经常需要定义一些16位或32位类型的二维数组来保存处理的中间结果或最终结果。另外如果用DirectDraw编程来进行图像处理和显示，为了保证跨平台特性，DirectDraw定义了一套数据类型：有无符号8位字符型BYTE，16位短整型WORD，无符号16位短整型USHORT，无符号16位字符型UCHAR，32位字符型LONG，无符号32位字符型ULONG，单精度浮点数FLOAT等类型。如果对应于每一种数据类型都定义一个类，这样虽然能解决问题，但并不是一个明智的办法。例如要修改一个变量或函数，那对应于每一种数据类型定义的类多要做同样的修改，毫无疑问，这种方法操作繁琐，容易发生错误，需要做大量的拷贝修改工作，因此是一种低效的方法。

Visual C++引入的模板Template概念，这一个关键字会告诉编译器下面的定义将操作一个或更多的非特定的类型。只有当对象被定义时，这些类型才必须被指定以使编译器能够替代它们。因此可以用一个非特定的参数来作类型名称，用该非特定的参数来定义一个通用的数据类或一个函数，诸如于MFC类库中的CArray类那样。下面给出了支持不同数据类型的一个图像二维数据模板类和一个快速排序类Sort。

1.定义不同数据类型的数据模板类

数据模板类的定义如下：

```
// - - - 不同数据类型图像二维数组模板的定义 - - -
template <class T>
class ImageTemplate
{
public:
    T **          lp_AddRow;  // 二维数组的数据指针
    unsigned int Width;       // 数组的宽度(列数)
    unsigned int Height;      // 数组的高度(行数)
    unsigned int ImageSize;   // 数组的大小
public:
    ImageTemplate(void);      // 构造函数
    ~ImageTemplate(void);     // 析构函数
    // 带参数的析构函数
    ImageTemplate(unsigned int w, unsigned int h);
    // 分配数据内存的函数
    void Construct(unsigned int w, unsigned int h);
    // 释放数据内存的函数
    void DeleteData();
};
// 构造函数
template <class T>
ImageTemplate< T >::ImageTemplate(void)
{
    lp_AddRow = NULL;
    Width = Height = 0;
}
// 析构函数
template <class T>
ImageTemplate< T >::~ImageTemplate(void)
{
    DeleteData();
}
// 带参数的析构函数
template <class T>
ImageTemplate< T >::ImageTemplate(unsigned int w, unsigned int h)
{
    lp_AddRow = NULL;
    Width = w;
    Height = h;
    ImageSize = w * h;
    Construct(w, h);
}
// 分配数据内存的函数
template <class T>
void ImageTemplate< T >::Construct(unsigned int w, unsigned int h)
{
    DeleteData();
    if( Height > 0 ) lp_AddRow = new T*[Height];
    if( ImageSize > 0 )
    {
        lp_AddRow[0] = new T[ImageSize];
```

```
        memset(lp_AddRow[0], 0, ImageSize * sizeof(T) );
        for(unsigned int i = 1; i < Height; i++)
            lp_AddRow[i] = lp_AddRow[i - 1] + Width;
    }
}
// 释放数据内存的函数
template <class T>
void ImageTemplate < T >::DeleteData()
{
    if(lp_AddRow != NULL)
    {
    if( lp_AddRow[0] != NULL ) delete []lp_AddRow[0];
        delete []lp_AddRow;
        lp_AddRow = NULL;
    }
}
// 定义不同数据的类
typedef ImageTemplate <     int     > IntImage;
typedef ImageTemplate <    float    > FloatImage;
typedef ImageTemplate <  short int  > SintImage;
typedef ImageTemplate < unsigned char > BYTEImage;
```

在程序中使用方法如下：

```
FloatImage fImage(640, 480);
SintImage  sImage(640, 480);
```

修改或获取数据可以通过公共的数据指针变量 [illegible] 如

```
for(int j = 0; j < 480; j++)
   for(int i = 0; i < 640 ;i++)
    {
      fImage.lp_AddRow[j][i] = i*j/ 100.0f;
      iImage.lp_AddRow[j][i] = 20* (i + j);
    }
```

由于图像处理有着快速的要求，因此一般将数据指针作为公共变量直接使用，如果考虑到程序的安全性和稳定性，可以将该数据指针定义为私有变量，通过重载的操作符号 [illegible] 来进行数据的修改和获取。

支持不同数据类型的快速排序的类

在图像处理中，对于提取出来的各种特征量要进行分析和识别，快速数据排序是其中常用的一种算法。虽然可以用函数重载的方法对不同的类型编制该类型数据的排序程序，但是引入的代码较多，因此函数重载的方法是一种低效的方法。本快速排序的类的排序功能由 [illegible] 标准库 [illegible] 中定义的

```
qsort( void * base, size_t num, size_t width, int (__cdecl *
compare )(const void *elem1, const void *elem2 ) );
```

函数来完成，其中参数含义如下：

```
        void   *base  数据指针
        size_t  num   数据个数
        size_t  width 单个数据的字节宽度
        int (__cdecl *compare ) 比较函数
```

对应于不同类型的数据使用该函数，按一般的方法需要编制对应不同数据类型的比较函数，而且还要进行数据类型强制转换，因此比较麻烦。利用下面定义的类就可以很方便的实现数据的快速排序。

```
//基于模板的快速排序类
template <class T>
class LQuickSort
{
public:
    // 升序排列的比较函数
    static int cmpT1(const void *ap, const void *bp);
    // 降序排列的比较函数
    static int cmpT2(const void *ap, const void *bp);
    // 排序的函数
    static void rank(T* lp,            // 数据的指针
                     int length, // 数据的个数
                     BOOL IsAscend = true ); //升序
};
// 升序排列的比较函数
template <class T>
int LQuickSort <T>::cmpT1(const void * ap, const void * bp)
{
    T a = *(T *)ap;
    T b = *(T *)bp;
    if( a > b )
        return 1;
    if( a < b )
        return -1;
    return 0;
}
// 降序排列的比较函数
template <class T>
int LQuickSort <T>::cmpT2(const void * ap, const void * bp)
{
    T a = *(T *)ap;
    T b = *(T *)bp;
    if( a > b )
        return -1;
    if( a < b )
        return 1;
    return 0;
}
// --- 排序的函数 ---
// IsAscend = true
// T*   lp,          数据的指针
// int  length,    数据的个数
// BOOL IsAscend   true 升序排列, false 降序排列
template <class T>
void LQuickSort <T>::rank(T* lp, int length, BOOL IsAscend)
{
    if( IsAscend )
        qsort( (void *)lp, length, sizeof(T), cmpT1 );
    else
        qsort( (void *)lp, length, sizeof(T), cmpT2 );
}
```

由于定义的类的成员函数为静态 [illegible] 类型，因此也可以不用声明该类的对象而调用排序函数。待排序的数据格式如下：

```
int   ix[10] = {10, -10, 2, ...};
float fx[10] = {10.0, -10.4, 2.3, ...};
```

具体的排序用法有下面两种。

用法一：定义类的对象

```
LQuickSort <int>  iSort;
LQuickSort <float> fSort;
iSort. rank(ix, 10, true);  // 升序排列
iSort. rank(ix, 10, false); // 降序排列
fSort. rank(fx, 10, true);  // 升序排列
fSort. rank(fx, 10, false); // 降序排列
```

用法二：不定义类的对象

```
LQuickSort <int>: : rank(ix, 10, true);   // 升序排列
LQuickSort <int>: : rank(ix, 10, false);  // 降序排列
LQuickSort <float>: : rank(fx, 10, true); // 升序排列
LQuickSort <float>: : rank(fx, 10, false); // 降序排列
```

由于排序函数的最后一个参数缺省为 [illegible] 因此升序排列也可以写为：

```
LQuickSort <int>: : rank(ix, 10);   // 升序排列
```

用上面定义的类就可以很方便地实现对不同类型的数据快速排序。

使用模板来定义类时，可以看出除了 [illegible] 外，该类看上去像一个通常的类，这里的 [illegible] 是替换参数，它表示一个类型名称。如果使用非内联成员函数定义，则要在函数定义前加入 [illegible] 来告诉编译器察看模板声明。在成员函数的定义中，类名称被限制为模板参数类型，如 [illegible] 而且更为特殊的是，即使定义非内联成员函数时，函数的实现代码也要放在类的头文件中，否则会发生连接错误。这似乎违背了通常的头文件定义准则：“不要在分配存储空间前放置任何东西”，这条准则是为了防止连接时的多重定义错误。但模板定义比较特殊，由 [illegible] 处理的任何东西都意味着编译器在当时不为它分配存储空间，它一直处于等待状态直到被一个模板实例告知。在编译器和连接器的某一处，有一机制能去掉指定模板的多重定义。所以为了容易使用，几乎总是在头文件中放置全部的模板声明和定义。

可以看出使用了模板后，大大简化了代码，提高了图像处理编程的效率。

二、高精度获得处理所用的时间

对于一种算法，其处理占用的时间越少则表明该算法效率越高，因此获得算法占用的时间是衡量算法的一个指标。

普通的获得处理占用时间的方法有：[illegible] 函数或 [illegible] 获得秒量级的时间；[illegible] 函数 [illegible] 可以获得毫秒量级的时间。下面给出一个可以达到微秒量级的时间测量类，主要通过 [illegible] 函数 [illegible] 两个函数来实现。用 [illegible] 函数获得计算机高精度的性能计数的频率，该频率值在多台机器上测量均为 [illegible] 这样其分辨率为 [illegible] 秒，只要获得处理前后 [illegible] 返回当前高精度的性能计数值的差值，就可以获得处理占用的时间。具体程序如下：

```
// 精确获得算法处理时间的类(毫秒量级)
class LTimeCount
{
private:
    // 算法处理时间(单位: 秒)
    double          UseTime;
    // 计数值
    LARGE_INTEGER Time, Frequency, old;
public:
    // 计时开始
    void Start()
    {
        QueryPerformanceFrequency(& Frequency);
        QueryPerformanceCounter(& old);
        UseTime = 0. 0;
    }
    // 计时结束
    void End()
    {
        QueryPerformanceCounter(& Time);
    UseTime = (double) ( Time. QuadPart - old. QuadPart) /
(double)Frequency. QuadPart;
    }
    // 获得算法处理时间(单位: 秒)
    double GetUseTime()
    {
        return UseTime;
    }
};
```

使用的方法为：

```
// 定义一个时间测量类的对象
LTimeCount m_CountUseTime;
// 开始计时
m_CountUseTime. Start();
// 下面进行相应的处理
// --图像处理算法--
// 计时结束
m_CountUseTime. End();
// 获得并显示处理占用的时间
CString msg;
msg. Format( "处理的时间是: %12. 8f 秒\n",
m_CountUseTime. GetUseTime() );
AfxMessageBox(msg);
```

通过实际的测试，该方法获得的时间可以达到微妙量级，同时发现用 [illegible] 函数获得的计数很不准确。以时间间隔为 [illegible] 毫秒，用 [illegible] 函数测量的时间间隔误差在 [illegible] 毫秒。

三、图像处理算法程序优化的方法

[illegible] 减少乘除运算，将乘除运算变为加减运算。

如将 [illegible] 写为 [illegible] 将多项式计算 [illegible]

[illegible] 写为 [illegible]

[illegible] 使用指针加快速度

由于指针直接指向操作的内存地址，因此用指针可以加快程序执行的速度。一般在进行图像的卷积运算时，摸板的数据是连续的内存，则可以用 *遭蹋责垣垣语句对数据进行操作，该语句的意思是在进行操作后，指针 责蹋责加 员从而指向下一个数据。由于 悦垣垣语言对 “垣垣”进行了优化，因此可以较大幅度的提高速度。

猿援建立查找表或变量

如果在一个循环中要多次用到 泽藻灶雪 糟燥泽雪 藻曾责雪等函数时，这些函数的参数是不变的，则可以通过在外定义查找表或变量的方法来减少计算量。如普通的获得图像数据的方法如下：

```
Sin1 = sin(Theta);
Cos1 = cos(Theta);
Exp1 = exp( -Theta * Theta);
for(RowAddress = 0, j = 0; j < ImgH; j++)
{
   for(i = 0; i< ImgW; i++)
image[RowAddress + i] = i * Sin1 + j * Cos1 + i*j* Exp1;
    RowAddress += imgW;
 }
```

用上面的处理方法可以减少 陨藻灶 * 次陨藻灶与原员次乘法运算和猿* 陨藻灶 * 陨藻灶原员次调用 泽藻灶雪 糟燥泽雪 藻曾责等函数运算。如果进行高斯滤波等运算，则可以建立数据查找表，事先将模板的数据算好存在查找表中，不须计算可以直接使用，可以节省大量的计算时间。如果一次运算需要涉及多行的数据，则可以通过建立指向图像数据行首地址的指针查找表来加快速度，如下押

```
BYTE * *RowAddress = new BYTE *[ImgH];
for(RowAddress[0] = InputImage ,i= 1 ; i< ImgH ; i++)
{
    RowAddress[i] = RowAddress[i-1] + ImgW;
}
```

这样用 砸杂载蚤藻数组元素即可以对坐标 为载蚤藻的灰度值进行操作。

源援内存的快速初始化和拷贝

用 皂藻皂泽藻贼函数可以对一块连续内存进行初始化，皂藻皂糟责赠雪函数可以将一块连续内存的数据拷贝到另一块连续内存中。如皂藻皂泽藻贼皂藻皂糟责赠(陨藻灶 * 陨藻灶)* 泽蚤扎藻燥枣(遭赠贼藻)语句将 糟澡葬则指针指向的一块连续 糟澡葬类型的内存初始化为 园 而 皂藻皂糟责赠(陨藻灶 * 陨藻灶 * 泽蚤扎藻燥枣(糟澡葬)语句可以将 陨藻灶 * 陨藻灶指针指向的一块连续内存的数据快速拷贝到 陨藻灶 * 陨藻灶指针指向的一块连续内存中。如果拷贝目标或源内存是不连续，由于在一行上面是连续的，因此可以变成数据高次数行拷贝也可完成。

缘援数据指针转换

由于计算机的数据总线的位数一般是 猿圆 因此数据类型的占用字节位数为 猿圆或 远源的整数倍，则操作速度将会加快，以用按位取反操作符 (~) 来对整数型数据取反为例，将数据指针强制由 月再载燥* 类型转换为 陨灶贼* 类型，速度比用 月再载燥* 类型指针速度提高 源倍。

远援寄存器优化

传统的编译器对寄存器的利用率往往不高，因为传统的编译器对寄存器的分配是以表达式为单位，对于不同的表达式寄存器被重新分配。通常这种方法有利于克服寄存器的溢出，但这种方法在一个循环中会造成一个数据反复从内存中装入寄存器，如下面的语句。

```
for(i = 0; i< m; i++){ c1[i] =a * d1[i];c2[i] =b * d2[i]; }
```

变量 葬和 遭公用了一个寄存器，因此每次循环时变量 葬和 遭都要重新装入。但是这样的方法受到寄存器数量的限制，而且要求程序员对寄存器非常熟悉，因此一般并不常用。实际上许多图像的处理形式都类似于一个模板与图像进行卷积运算。因此针对模板的大小进行了不同的优化。对小模板处理以 杂蚤扎藻算子为例，一般的程序如下

```
for(j=1; j<ImgH-1; j++)
{
    for(i=1; i<ImgW-1; i++)
    {
      tempx = tempy =0;
      for( y= -1; y<=1; y++)
      {
          for(x= -1; x<=1; x++)
         {
      tempx += Image[j+y][I+x] * xMask[y+1][x+1]
      tempy += Image[j+y][I+x] * yMask[y+1][x+1]
         }
      }
           nG = int(sqrt(dx*dx+dy*dy) + 0.5);
      NewImage[j][i] = (nG>255) ? 255 : nG;
    }
}
```

将内部循环展开，这样首先循环开销如循环控制变量的修改、循环条件的判断被删除；其次，内部循环展开使得原先指令数较少的循环体指令序列变成指令数众多的非循环指令序列。这样许多指令调度和经典优化可以被使用。将模板的元素都各自分配一个寄存器，如 皂糟园 越 皂糟责赠(陨藻灶 * 陨藻灶 * 泽蚤扎藻燥枣(糟澡葬)语句在循环体中用寄存器变量 皂糟园代替 皂糟责赠(陨藻灶 * 陨藻灶 * 泽蚤扎藻燥枣(糟澡葬)可以减少大量的寄存器装载时间。另外如果模板的系数已知，则直接用常数计算，如下面的代码，速度比未优化前提高了许多。

```
for(j=1; j<ImgH-1; j++)
{
      for(i=1; i<ImgW-1; i++)
      {
 dx = RowAddress[j-1][i+1] - RowAddress[j-1][i-1];
 dx += (RowAddress[ j ][i+1] + RowAddress[ j ][i+1] -
RowAddress[ j ][i-1] - RowAddress[ j ][i-1]);
dx += RowAddress[j+1][i+1] - RowAddress[j+1][i-1];
dy = RowAddress[j+1][i-1] - RowAddress[j-1][i-1];
dy += (RowAddress[j+1][ i ] + RowAddress[j+1][ i ] -
RowAddress[j-1][ i ] - RowAddress[j-1][ i ]);
dy += RowAddress[j+1][i+1] - RowAddress[j-1][i+1];
            nG = int( sqrt(dx* dx+dy* dy) + 0.5 );
            Image[i] = (nG>255) ? 255: nG;
      }
```

```
        Image + = ImgW;
    }
```

而对于大模板处理，由于寄存器的数量有限，可以将大模板运算变成若干个子模板的计算来完成。由于循环体语句众多，这样处理增加的绝对计算量与节省的计算量相比可以忽略不计。

四、其它一些杂项

员援用 悦陨藻憎糟运行时类获得操作区域

有时仅需要对某一个区域进行处理，如果用常规的方法响应鼠标左键按下，放开和鼠标移动消息，编程则比较复杂，但如果用 酝云悦类库中的 悦陨藻憎糟运行时类就可以很方便地用鼠标获得要处理的区域。

用法如下：

首先在 宇藻憎糟类头文件中声明一个对象

```
CRectTracker RectTracker;
```

然后在 鼠运陨运运行函数中加入下面的语句

```
RectTracker. Draw(pDC);
```

最后在响应鼠标左键按下消息处理函数 鼠运陨运运行函数中加入语句

```
if (RectTracker. TrackRubberBand(this, point))
{
    Invalidate();
}
```

这样便可用 悦陨藻憎糟运行时的成员变量 运陨晕运获得所选的区域。

圆援运藻糟造操作符的使用

该操作符的用法有 运藻糟造一元表达式 或 运藻糟造类型名 返回的整型值给出操作数在内存空间占用的字节数。如果操作数是一个数组类型，则可以得到该数组数据指针指向的内存空间的大小，如下例。

```
int hist[100][10];
int size1 = sizeof(hist);       // = 4000;
int size2 = sizeof(hist[0]);    // = 40;
int size3 = sizeof(hist[0][0]); // = 4;
```

对于一维数据指针 运藻糟造运藻糟造运藻糟造运藻糟造就可以获得数组的大小。

猿援使用 “越”操作符时的注意事项

如下例：

```
int a = 10, b = 5;
int c = a - - b;
```

由于 糟垣垣语言是从右到左执行，因此上式执行完后 葬越糟越缘鸦

源援在循环中使用 “垣垣”或 “原原”操作符时的注意事项

在循环语句 藻憎造初始化表达式中无论使用 垣垣蚤还是 蚤垣垣，或 原原蚤和 蚤原原， 在循环体中的变量 蚤的值是一样熏应为 垣垣蚤或 蚤垣垣相当于在循环体当次循环的最后一句执行，如下例。

```
int i, k = 0, l = 0;
for(i = 0; i < 5; i + +)
{
   TRACE(″k = %d l = %d i = %d\n″, k + +, l + +, i);
}
k = l = 0;
for(i = 0; i < 5; + +i)
{
   TRACE(″k = %d l = %d i = %d\n″, k + +, + +l, i);
}
```

结果押

```
k = 0 l = 0 i = 0
k = 1 l = 1 i = 1
k = 2 l = 2 i = 2
k = 3 l = 3 i = 3
k = 4 l = 4 i = 4
k = 0 l = 1 i = 0
k = 1 l = 2 i = 1
k = 2 l = 3 i = 2
k = 3 l = 4 i = 3
k = 4 l = 5 i = 4
```

以上是本人使用 宋悦垣编制图像处理程序中得到的一点技巧和经验，希望对大家有所帮助。

（收稿日期：圆园园源年 源月 员园日）

山特‘热’泉城

近日，圆园园源年济南国际信息技术博览会暨项目发展会隆重召开，山特济南分公司作为国内 耘孕泽行业的知名企业携山东总代理济南先导科技有限公司共同参加了此次展会。

山特公司是世界前五大专业从事不间断电源 耘孕泽的开发、生产及经营的厂商之一。在展会上，山特公司重点推出了新产品 酝灶员园园园、耘酝员园园园及线上互动 耘孕泽——晕灶员园园园。其中 耘酝系列产品属于后备式 耘孕泽，是山特公司专为家庭及办公环境设计，价格低廉、外型轻巧，对于 孕悦用户而言，是最为理想的安全卫士；线上互动式 晕灶员园园园 耘孕泽提供了轻松上网智能管理网络电源功能，是网络时代的最佳选择。

智神嵌入式操作系统　独放异彩

悦陨杂运行时类在圆园园源中国国际信息通信展览会上，上海高通电脑有限公司研发的‘智神’嵌入式操作系统异彩独放，倍受瞩目。

智神嵌入式操作系统为信息家电提供全面解决方案，此次展出的应用机型为‘东方 圆号’机顶盒，功能完善，可满足广大无 孕悦用户浏览因特网、收发电子邮件、家庭办公 包括文字处理、传真收发等功能）、家庭教育、休闲游戏、股票分析交易的需求。美国国家半导体亚太区总经理对此表示极大的关注，国家信息产业部有关领导人对此项技术给予了充分的肯定。

智神嵌入式操作系统功能强大，性能稳定、快速，受到广大消费者和商家的好评。并且，有多家投资商和硬件生产商纷纷表示要与高通公司进一步合作。

需要全本请访问 www.ertongbook.com