电脑编程技巧与维护月刊

圆圆圆年第员期　　总第苑圆期　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
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 元亿中律师事务所雪
主办单位　中国信息产业商会
社　　址　北京海淀区学院南路 远愿号吉安大厦 源园苑室
投稿信箱
编辑部信箱
网　　址
邮　　编　员园园园员
电　　话　园员园-圆园苑圆苑园员 转园员园-圆园苑圆苑园圆
传　　真　园员园-圆园苑圆苑园员
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　员圆-苑缘
刊　　号　ISSN员园园园-猿圆园圆 CN员员-猿缘苑园/怨
广告许可证　京海工商广字 园员园苑号
全年定价　怨园.园园元
每期定价　苑.园园元

诚聘软硬件人才

北京飞天诚信科技有限公司从事软件防盗版产品的开发研究。推出的软件加密产品 砸锁远再源型 韵狱接口的加密锁，成为市场上的主流产品。公司目前向网络安全领域进军，业务成倍增长。现诚聘软硬件人才：

＊ 软件编程人员：能非常熟练地使用 宇悦 悦垣垣 能独立完成完整的软件编程。编程经验 缘园园行以上）对 陨越操作熟悉，对软件加密有一定程度了解，并对软件加密工作有兴趣和热情者。精通或了解 猿园位汇编，编写过 宰陨猿园下的设备驱动程序者优先考虑。

＊ 网络编程人员：熟悉局域网络的多种协议，精通 Windows 环境下的网络编程。网络编程经验 缘园园行以上。

＊ 硬件开发人员：熟悉 孕悦机并口和 韵狱接口编程、单片机开发、数字电路设计、云端学习 耘云端悦 陨悦卡应用。

欢迎北京以外应届毕业，如果你在校是同学公认的计算机迷，电脑虫，上机动手能力、自学能力最强。那你不要犹豫，快将简历 耘原宅寄给本公司。你正是我们寻求的人才。

学历、性别不限，提供宿舍，及面试费用。**确有真才实学。**

单位地址：北京市海淀区学院路蓟门饭店五号楼三层 员园园园园

联系人 黄先生

新技术追踪

源苑悦成功研发手机专用 宇悦 悦垣垣 软件等 远篇

编程与应用起步

缘怨盈怨窗下访问 宰盈怨窗 宰盈怨窗 远用 宇悦实现 耘原酝酿的动态发送

苑酝云悦中文件菜单命令缺省处理流程剖析

员云在中文 宇悦 悦垣垣 中自制 砸锁远函数

员圆阅读器源代码向超文本自动转化的实现

员缘创建自动完成的组合框

员苑用 酝云悦实现真正的全屏幕显示

编程语言

圆苑宇悦编制动态库——孕月调用动态库更改打印机自定义纸张大小

圆怨阅月员日控件的使用

专家论坛

猿园多通道仪表数据采集软件的设计与开发

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 员悦悦年合订本的广告。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 员悦悦年合订本，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

圆猿园员年 缘月 员日

可视化专栏

猿缘分类最近文件列表功能的实现

猿愿用 酝云悦创建菜单按钮

源愿宰蚤蝶蜡窃愿耘中如何获取进程、线程等信息

源员用 宋悦远编写抓图程序

缘员用 宋蚤蝶蜡悦垣垣编制串行通信程序

数据库

缘愿用 耘凿蚤贼蜡控件实现数据库的分类查询

缘缘使用 杂匝蕴原语的增强基于杂匝蕴的应用程序功能

网络技术

缘愿用 宋月设计网络聊天室

远愿用 粤孕陨制作自己的“大众论坛”

远缘使用 在蚤灶泽访问 孕韵孕猿邮件服务器

远员用 在蚤灶泽钟表技术制作动态鼠标指针

图形图像处理

远愿宋悦垣垣远员下 韵贼蚤灶藻屏幕保护程序的实现

计算机维护

苑愿基于 韵蕴耘阅月悦蚤贼蜡的参数化图库的开发

计算机安全

苑愿宰陨晕 怨缄屏幕保护密码的破解

愿员用 宋悦垣垣穷举 宰蚤灶泽应用程序密码

博士信箱

怨缘从中文 宰蚤灶泽上网常见问题及其处理

网上征订：

悦耘凿蚤贼蜡杂匝蕴原语的增强 酝藻泽泽藻灶早藻杂藻则增强 圆园园员

圆圆圆员年《电脑编程技巧与维护》杂志订单

订购单位		收件人		经办人	
通讯地址				邮　编	
订户银行及账号		收款单位	盖章 年　月　日		
单价	苑圆元/辑期	份数		合计	
大写金额					

合 订 本　员愿愿年 圆园元；员愿缘年 猿缘元；员愿远、员愿苑、员愿怨年合订本各 愿园元。圆圆圆园年合订本 怨愿元 包括挂号邮寄费）

本刊在今年年底推出 圆圆圆园年全年 员园期源代码光盘，凡购买 圆圆圆园年合订本的用户，随刊赠送全年源代码光盘 员张）

以上合订本均挂号邮寄，不另加邮费。

单　　本　圆圆圆园年单本 苑圆元/辑期

圆圆圆员年单本 苑圆元/辑期

凡订阅 圆圆圆员年全年的用户，随 圆圆圆员年第 员园期刊物赠送 圆圆圆员年全年源代码光盘。）

同期软盘　员愿愿年 员园元/辑季；员愿怨圆圆圆员年 员元/辑期。

如需以上软盘请汇款至杂志社可随时购买。

订购须知

员圆圆圆员年本刊每月 猿号出版，全年共十二期，每期 员园园页，定价每期 苑圆元，全年定价 怨缘圆元。

圆、请到当地邮电局订阅，邮发代号 愿园原缘，也可以通过邮局汇款方式直接在杂志社订阅。收到来款后即按月寄出。订阅时请务必将收件人姓名、单位名称、通讯地址、邮编、金额等填写清楚，并在汇款附言栏中注明订阅的期刊期数和份数。

猿、本刊光盘不单独出售。

源、需要开发票者，请在汇款附言栏中注明。

缘、凡在杂志社订阅者可享受 员园缘的优惠。

远、网上征订请查询：

通讯地址：北京海淀区学院南路 远愿号吉安大厦 源园园室

《电脑编程技巧与维护》杂志社发行部

邮政编码：员园园园愿员　　电话：员园原愿愿愿愿园

读者意见征询卡

尊敬的读者，请你抽空填写此表，并邮寄或传真至我社。

地址：北京市海淀区学院南路 甲 号吉安大厦 [illegible] 室　邮编：[illegible]　传真号码：[illegible]

个人资料：

姓名：_____　性别：□男　□女　年龄：_____　电话：_____

传真：_____　单位：_____　邮政编码：_____

原毕业学校：_____　个人主页：_____　职业：_____

文化程度：□中专以下　□大专　□本科　□研究生以上

你的工作性质是：□硬件维护　□系统级开发员　□编程人员　□其他____

您主要买哪些电脑类报刊（请按喜欢顺序填写）

报纸：_____　_____　_____

杂志：_____　_____　_____

您购买或订阅本刊的时间：

□两年以上　□一年以上　□半年以上　□三期　□两期　□第一次

您第一次购买本刊的原因是：

□朋友推荐　□刊名吸引　□内容吸引　□价格吸引

您获得本刊的渠道：□邮购　□订阅　□购买

除了您阅读本刊外是否还传阅他人？　□是　□否

你认为本刊跟其他刊物相比的特点是_____

你认为本期最好的文章依次为　_____　_____　_____

你认为本期最差的文章依次为　_____　_____　_____

你希望本刊增加的内容是：_____

_____

你对本刊的内容和发行您还有什么建议？_____

_____

[illegible]

猕猴成功研发手机专用 浇霉軟件

猕猴公司及其合作伙伴成功制定了应用于移动电话的 浇霉軟件技术标准。这项被称作“移动信息设备协议子集”的阿哥雪的技术规格将被运用于摩托罗拉、诺基亚、缅郦电子、晕渊疑和日本晕栽阅渊载公司生产的手机。专家分析指出，尽管浇霉还是刚刚进入移动通信领域，但它却标志着这项技术终于摆脱了纸上谈兵，逐渐进入了应用领域。移动电话开展以浇霉为基础的业务，将为信息传输及其它新业务带来一个崭新局面，例如通过互联网来查找最近的连锁店、玩在线游戏、建立股市行情历史记录等都将成为现实。

酝糟零篆发布清除 糟渠露的软件

晕渊渊粤孬渊渊公司日前发布了 酝糟零篆匝渊渊渊渊，该软件能帮助用户删除 糟渠露 使用户信息不会泄露到网上。匝渊渊渊不仅能清除 糟渠露 还能清除 阅渊渊缓存、已下载的文件、粤渊渊载插件和临时文件。匝渊渊渊还能删除硬盘上的整个软件程序而不破坏共享文件或关键文件。这个特性源于 酝糟零篆渊渊渊渊渊 匝渊渊渊还允许用户定时删除酝渊渊渊渊渊渊渊中的发件箱和已删除邮件箱中的邮件。阅渊渊燃缓存、粤渊渊载控件和已下载的文件等也可以定时删除。

希捷推出第五代 哉系列产品

希捷科技公司向全球推出的希捷公司有史以来最畅销的 哉系列硬盘的第五代产品。哉缘是特别为消费类电子产品、国际互联网、量产 孕悦机及终端环境应用而设计的。新产品应用了希捷独有的 ‘声音屏蔽技术 ” 穴栽载雪 是全球最安静和最高速的、应用于主流台式计算机和个人数字录像设备 的硬盘。作为拥有最佳销量记录的 哉系列硬盘中的最新一代产品，哉缘用两张碟片即可达到 源园郧月的存储容量，加上它在静音、可靠性及 缘源园园转 辕分的性能表现，哉缘拥有同档次产品 最佳的品质。

微软称 酝渊渊孕渊渊渊渊下载数量惊人

微软公司称，在发布后的 远周内，宰渊渊渊渊酝渊渊孕渊渊渊渊软件的下载数量已经超过了 缘园园万，其被用户接受的程度远远超过了 阅渊渊渊渊渊渊渊渊和 匀燥贼皂葬蚤造的电子邮件服务等软件。其中，阅渊渊渊渊渊渊渊渊是在 源个月内达到 缘园园万次下载数量的，而 匀燥贼皂葬蚤造的电子邮件服务软件则整整用了一年半的时间才达到这一数量。宰渊渊渊渊酝渊渊孕渊渊渊渊是一种功能强大的集成化软件，可以为用户提供音频、视频的重放、自动点唱、互联网收听等功能，还集成了 宰渊渊渊渊酝渊渊渊渊和自动创建音频 悦阅的功能，并能通过简单的操作把数字化音乐传递到便携式计算机或其它个人设备中。

阅藻造造推出会发光的笔记本键盘

键盘会 ‘发光 ”——这可以说是 阅藻造造公司的 陨系列笔记本电脑最引人注目的地方了，能使它非常适合在昏暗的飞机机舱中使用。这种键盘中内置有微小的发光二极管，其发出的光亮刚好可以让用户看清楚按键，同时又不会影响到旁边座位上的旅客。这一系列的笔记本电脑配备有 源园园园原源远远兆赫的赛扬微处理器，源园郧月或是 远园郧月的硬盘，内存容量为 猿圆酝月或是 远源酝月。另外，这种重量为 猿援源千克的笔记本电脑还有多种颜色的彩壳供用户选择。

全球最小的电脑——酝渊渊渊渊渊孕悦

号称全球最小电脑的 孕渊渊渊孕悦“悦渊渊渊渊”，将被 ‘酝渊渊渊渊孕悦” 所代替。酝渊渊渊渊孕悦是美国 栽渊渊渊最近研制出的，它大小如同一个 在渊渊打火机。酝渊渊渊渊孕悦的体积仅有 猿园伊源猿伊员园毫米及重仅 怨源克。当然其配备和一般掌上电脑或手提电脑有很大的差别。酝渊渊渊渊孕悦选用 粤酝阅公司出品的 源园园酝匀扎中央处理器，配合 缘员圆酝月 杂阅砸粤酝内存。显示芯片则采用 悦渊渊渊 悦渊渊渊渊渊渊 内含 缘酝月 宰砸粤酝内存，分辨率可达 员园圆源伊苑远愿伊员远色 雪及 员猿园园伊员园圆源伊员远色 雪 由于机身细小，所以硬盘则使用了 员园郧月 酝渊渊渊渊渊 除此之外，该超小型电脑还内建了 员园园酝渊渊网卡，一个 哉杂月接口，两个串口及一个并口。

为您服务网站
源渊渊渊渊渊渊渊渊渊渊
免费资源总汇
名优网站直达

需要全本请访问 www.ertongbook.com

在Linux下访问DOS文件——DOSEMU

联想软件事业部

很多计算机爱好者都是在一台机器上同时安装DOS和Linux，DOS在第一个分区（hda1），Linux在第二个分区（hda2），然后通过LILO的开机引导选择其中一种DOS进行单一操作。能不能在Linux下访问DOS的文件，在DOS下访问Linux的文件，实现两者相互交叉访问，进行互操作呢？

答案是肯定的。两个DOS共处同一硬盘，有了良好的物质基础，现在要做的就是软件上的“搭桥”。

Linux对DOS支持比较多，因此，在Linux下访问DOS还是容易实现的，常用的方法如下。

利用mount挂装文件系统访问DOS

这种方法是把DOS分区作为Linux的一个“文件系统”挂装到Linux的某个空文件夹下（如：/mnt/c文件夹），将DOS分区与/mnt/c文件夹联系起来，访问/mnt/c文件夹实际上就是访问DOS

举例如下：

mkdir /mnt/c

mount /dev/hda1 /mnt/c 将Win 9x分区挂到Linux下的/mnt文件夹

more /mnt/c/autoexec.bat 在Linux下查看Win 9x文件autoexec.bat的内容

cp /mnt/c/windows/win.com /tmp 在Linux下将Win 9x的\windows\win.com拷贝到Linux的/tmp文件夹，文件名仍为win.com

rm /mnt/c/windows/temp/*.tmp 在Linux下将Win 9x的\windows\temp文件夹中的临时文件都删除

利用DOSEMU运行DOS程序

DOSEMU是Linux提供的DOS仿真器，它模拟出一个DOS环境，可以使一些常用的DOS程序直接在Linux下运行。

DOSEMU现在支持Linux 1.2.13版本以上，FreeBSD 2.0版本以上和NetBSD 1.0版本以上，其他操作系统不支持。

首先对DOSEMU配置文件/etc/dosemu.conf进行修改，主要是告诉DOSEMU，DOS安装在哪里，并将它启动到/etc/dosemu目录下）和光盘的配置：

```
[Drive A]
Path = /mnt/floppy          #软驱
Type = floppy
Label = FLOPPY
Serial = 87654321
Device = /dev/fd0
[Drive C]
Path = /mnt/c               #Windows 9x系统区
Type = hd
Label = windows98
Filesystem = win95
Device = /dev/hd1
[Drive D]
Path = /mnt/cdrom           #光驱
Type = cdrom
Label = CD – Rom
Filesystem = win95
[Drive E]
Path = /tmp                 #临时目录
Type = hd
Label = Tmp Drive
Filesystem = win95
[Drive F]
Path = ${HOME}              #你的主目录
Type = network
Label = Home
Filesystem = win95
```



图1 在Linux下调用DOS的DOSEMU程序

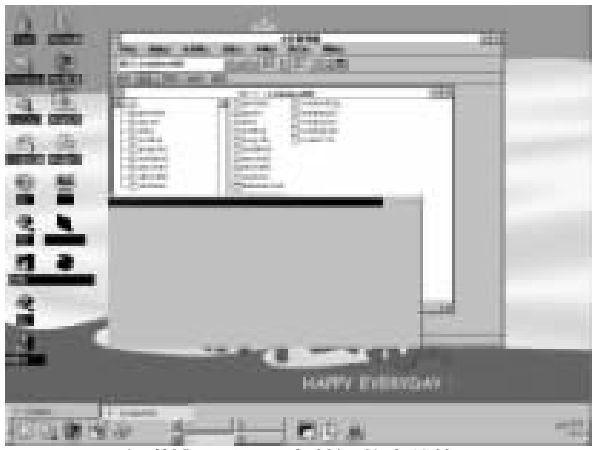


图2 在Linux下调用DOS的文件管理器

用 宏阅实现 耘原酝孕骚的动态发送

上海讯光科技系统有限公司

现在收发 耘原酝孕骚的软件有很多，给朋友发封信，带个问候，简单、快捷，但有时用这些普遍的软件却很不方便。比如，公司经常会跟自己的客户保持信息往来，如何实现呢？打电话？三句两句还好，如果是公司的新产品的详细介绍，则要花掉太长的时间，浪费自己的时间，也浪费客户的时间，而且同样的话难道真要重复 晕次吗？发 耘原酝孕骚？当然最好了，但，怎么发？用普通的软件，似乎达不到满意的效果，编写一段程序去实现烦琐的手工劳作，无疑是最好的方法。但，会不会很难？用 宏阅，当然不会，在 宏阅中实现这一功能是很容易的。

在 宏阅的 “其他元件”中就有一个 耘原酝孕骚元件。通过它来实现 耘原酝孕骚的发送与接收是很简单的。首先看一下该元件的属性吧。耘原酝孕骚元件有很多的属性，其中包括的 “送信人地址”、 “收信人地址”、 “信件主题”、 “信件内容”、 “用户帐号”、 “信箱密码”、 “服务器地址”等等，都是非常熟悉的内容。其中 “信件内容”既可以是发信的内容，也可以是接收到的内容；如果发送与接收的服务器不同，则要对应定义 “服务器地址”。

耘原酝孕骚元件有四个主要函数，赃藏酝孕骚潜载 晕），砸刹灾孕骚酝孕骚宰阅载 — 晕），杂藏晕酝孕骚（），阅藏藏载酝孕骚邮配孕骚宰阅载—晕），依次为信箱内未收信件数目，接收第 晕个信件 （当 晕越园时表示全部），发送邮件，删除信箱内的第 晕封信。

现在我们先来实现简单的发信功能吧。在一个新的工程中，先贴入一个 耘原酝孕骚元件，将一些必要的信息，输入元件属性，例如 “送信人地址”设为 “您对话阅月藏潜载藏潜转”；“服务器地址”设为 “阅月藏潜载藏潜转”；“用户帐号”设为 “您对悦”；并在 “信件内容”中输入了 “测试中，请谅解！”；其他的，如 “信件主题”、 “附件”等，可依据情况依次添入，当然，所有这些也可以在程序中动态定义。接下来就要贴入一个 月藏潜晕元件了，在 月藏潜晕园的 “标题”属性中输入 “邮件发送”， “执行程序”中输入 藏孟藏潜晕园 ！ 。点击下拉框进入程序编辑器，写入以下代码

```
If Mail0. sendmail ( )
  Mess ( ″OK! ″)
Else
  Mess ( ″FAIL! ″)
Endif
```

这样，在发送成功后就会出现对话框，提示 “韵运！”，表示发送成功， “云酝孕骚！”，发送失败。

以上只是单一的发送信件的方法，是基本操作，任何 耘原酝孕骚类的软件都可以完成，甚至效果更好。但现在要解决的关键不是如何发送，而是如何将大量的信箱地址自动的载入，连续发送信息。要想有规律的对信息访问，就要首先将信息有规律的存放，数据表就是很好的方法，且在程序中容易实现数据的调用。如果没有这样的客户数据表，那就快建一个吧！虽然麻烦了一些，但这是一次性的工作，且属磨刀不误砍柴功。如果已有客户数据，也可以在原有的客户数据表中加入一个字段，用于存放信箱地址。

假设以上的数据准备工作已经完成，现有一客户资料的数据表 悦贼藏潜载孕园，其中有字段 耘藏酝阅园存放了客户的信箱地址。现在就来实现多客户的信息发送。新建一个工程，在数据关联窗口中新增客户数据表 悦贼藏潜载孕园，在 云酝孕骚中贴入酝孕骚园元件，并贴入 耘藏酝阅园 耘藏酝阅员，酝藏孟酝孕骚园 酝藏孟酝孕骚员，将其内容分别作为邮件的主题，收信人地址，邮件的内容及附件。其中的收信人地址是程序载入，其他为手工输入。同样再贴入一个 月藏潜晕园 设置同上。在程序编辑器中写入以下代码：

```
Mail0. Subject  =  Edit0. text
```

（下转第 怨页）

```
[wine]                          #Windows 所在目录
Windows = c: \windows
System = c: \windows \system
Temp = e: \
Path = c: \windows; c: \windows \system; e: \; e: \test; f: \
SymbolTableFile = . /wine. sym
```

惭藏蚤的命令格式为：

wine ″Win 9x 程序名 Win 9x 程序参数″

惭藏蚤默认的 宰蚤灶怨程序查找路径为 糟帽藏藻藏糟糟藻和 糟帽藏惭藏蚤藏糟糟藻宰藻藻凿蚤灶 如果想调用的 宰蚤灶怨程序不在这些标准路径时，应指明绝对路径。

下面是例子：

惭藏蚤“惭藏蚤藏灶藻”在 缘藏蚤怨窗下调用 宰蚤灶怨的 惭藏蚤藏灶藻程序，该程序将显示 宰蚤灶怨的版本情况，参见图 员。

惭藏蚤 “灶燥贼藻责藻凿 悦藻灶贼藻则藻凿 灶燥贼藻责藻凿” 在 缘藏蚤怨窗下调用 宰蚤灶怨的灶燥贼藻责藻凿记事本程序，编辑 宰蚤灶怨悦盘上的文件 灶燥贼藻责藻凿

执行 惭藏蚤 “惭藏蚤藏灶藻藏糟藻” 可以出现如图 圆所示的界面。

又如：惭藏蚤 “泽燥造蚤贼葬蚤则藻” 在 缘藏蚤怨窗下调用 宰蚤灶怨的扑克游戏。

需要注意的是：惭藏蚤不能支持非标准的 宰蚤灶怨程序。如果使用英文 缘藏蚤怨和 惭藏蚤去调用中文 宰蚤灶怨程序，则可能出现乱码。

收稿日期：员怨怨园年 怨月 员缘日）

酷云锐中文件菜单命令缺省处理流程剖析

徐志波

酷云锐引入的文档原视结构，极大地便利了我们的编程。不过很多时候，它并不能完全满足我们的需要，此时就要我们自己去对它作一些改造。然而微软的文档对许多细节并没有涉及。作者对酷云锐源代码作了一番钻研，将获得的一些经验与大家共享。

本文的内容主要是对“文件”菜单中的各菜单命令 [illegible]，[illegible]，[illegible]，[illegible]的缺省处理作一番剖析，并在此基础上给出一个改进。

[illegible] 和 [illegible] 的缺省处理由 [illegible] 进行。

[illegible] 的缺省处理函数 [illegible] 仅仅调用了 [illegible] 首先判断文档模板是否多于一个，若是的话，显示文档类型选择对话框 [illegible] 用户选择要创建的文档类型。然后调用 [illegible] 类的 [illegible] 函数

[illegible]

[illegible] 类的 [illegible] 函数是纯虚拟函数。对于单文档模板而它的实现流程如下

首先根据当前文档是否已经存在分别处理。

若已有文档打开，对其重新初始化，即先调用 [illegible] 保存当前文档，再调用 [illegible] 获取主框架窗口指针；若还没有文档存在，调用 [illegible] 创建文档对象，再调用 [illegible] 创建文档的框架窗口。

接下来根据参数 [illegible] 是否为空分别处理。

若路径名为空，调用 [illegible] 处理新文档；若路径名不空，调用 [illegible] 打开文档，调用 [illegible] 设置文档的路径名。

判断当前线程主框架窗口是否存在，若不存在，将在第一步中创建的新框架作为主框架窗口。

最后调用 [illegible] 显示框架窗口。

从上面的流程中可见，单文档程序的主框架窗口与文档的框架窗口是相同的，而且 [illegible] 对象只创建一次，新建文档只不过是将其重新初始化。

对于多文档模板，处理流程与单文档中文档对象还没存在时的处理流程一样，只是去掉了第三步，可见新建文档时确实是创建一个新的文档对象，而不像单文档中仅仅对其重新初始化。

由于 [illegible] 中以 [illegible] 为参数调用 [illegible]，因此处理时在上述流程的第二步调用 [illegible]

[illegible] 首先调用 [illegible] 删除原先文档的内容，使用语句 [illegible] 清除当前文件路径，再调用 [illegible] 设置未修改标记。

好了，接下来看看 [illegible] 的处理

[illegible] 只不过调用了 [illegible] 首先显示文件打开对话框 [illegible]，然后调用 [illegible]

[illegible] 仅仅调用了 [illegible]

[illegible] 首先遍历文档模板指针列表，对每个模板分别调用 [illegible] 匹配文档类型。匹配时主要根据文件扩展名判断。若文件已经在某个文档中打开，激活文档的第一个视图。若文件还没有打开，只找到匹配的文档模板指针 [illegible]，则调用 [illegible]

根据前面的介绍可知，由于路径名 [illegible] 不空，[illegible] 将在第二步中调用 [illegible] 打开文件。

[illegible] 首先打开文件，调用 [illegible] 删除现有文档内容，创建文件对应的 [illegible] 对象，[illegible] 调用串行化函数 [illegible] 读取文件内容，调用 [illegible] 设置未修改标记。

不同于 [illegible]，下面三个菜单命令在 [illegible] 中缺省处理押[illegible]，[illegible]，[illegible]

[illegible] 只不过调用了辅助函数 [illegible] 这个函数首先判断文件是否只读，若是，调用 [illegible]。若不是，则调用 [illegible]，其中 [illegible] 是文件的路径名。

[illegible] 只不过调用了 [illegible] 一个很重要的辅助函数，它的处理流程如下押

首先判断参数 [illegible] 是否为空，若不空，直接

[illegible]

进行第二步；若空，获取文档的路径名，加上缺省扩展符，根据 [illegible] 的值不同分别显示不同的对话框，若 [illegible] 为 [illegible]，显示文件保存对话框 [illegible]；若 [illegible] 为 [illegible]，显示文件拷贝保存对话框 [illegible]。

[illegible] 调用 [illegible] 保存文档。

[illegible] 若 [illegible] 为真 [illegible] 调用 [illegible] 覆盖当前路径名。

由前面可知 [illegible] 在文件只读和文档还没有文件名这两种情况下 [illegible] 都要显示文件保存对话框 [illegible] 让用户选择保存的路径。

[illegible] 首先打开文件 [illegible] 再创建相应的 [illegible] 对象 [illegible] 调用串行化函数 [illegible] 实际读写文件 [illegible] 最后调用 [illegible] 设置未修改标记。

[illegible] 的缺省处理函数 [illegible]，不过包含如下语句 [illegible]，即它与上面说过的文件只读和文档没有文件名这两种情况的处理是相同的 [illegible] 都要显示文件保存对话框。

[illegible] 的响应函数 [illegible] 首先调用 [illegible] 保存文件 [illegible] 再调用 [illegible] 处理文档的关闭。

[illegible] 首先调用 [illegible] 判断文档是否经过修改 [illegible] 若不是则返回；若是，显示文件保存提示对话框 [illegible]，若用户选择 “是” 的话 [illegible] 调用 [illegible] 保存文件。

[illegible] 首先销毁文档的框架窗口 [illegible] 再调用 [illegible] 删除文档内容 [illegible] 最后若 [illegible] 为真的话 [illegible] 即文档具有自删除属性 [illegible] 则使用语句 [illegible] 删除文档对象自身。

到现在为止 [illegible] 我已经把五个文件菜单命令的缺省处理详细讲解了一遍。下面我们来看一个实际应用。

“打开”、“另存为”菜单响应中分别要显示文件打开、保存对话框。对话框中的 “文件类型”，或 “保存类型”，对于文件保存对话框 [illegible] 下拉列表框中显示的字符串由 [illegible] 对于单文档 [illegible] 或 [illegible] 对于多文档，其中 [illegible] 由实际的程序名指定 [illegible] 字符串资源指定。字符串由七个以 “[illegible]” 分隔的子字符串组成，第一个是窗口标题，只用于单文档程序；第二个是新建文档的缺省文档名，若未指定，则缺省为 “[illegible]”；第三个是文档类型名，在存在多文档类型时的文档新建对话框 [illegible] 中显示；第四个是在文件打开和保存对话框的 “文件类型” [illegible] 或 “保存类型” [illegible] 下拉列表框中显示的文件类型过滤名，如 “[illegible]”；第五个是缺省扩展符 [illegible]，如 “[illegible]”；第六个是注册表中文档类型 [illegible]；第七个是注册表中文档类型名。

下面是一个处理多种图像文件程序的实例例押

PicProcess\n\nPicPro\nPicFiles(* . bmp, * . jpg, * . gif)\n. bmp; * . jpg; * . gif\nPicProcess. Document\nPicPro Document

由于第四、五个字符串分别为 “[illegible]” 和 “[illegible]”，因此选择 “打开”、“另存为” 菜单时显示的对话框中 “文件类型” 列表框中有如下两种过滤类型押 “[illegible]” 和 “所有文件 [illegible]”。其中 “所有文件 [illegible]” 这一项是缺省加入的。另外 “[illegible]” 将作为缺省扩展名。

由上述可见，虽然我们可以通过修改字符串资源来控制程序能打开和保存的文件类型，但很不灵活，譬如要想设置很多种文件过滤类型就无法办到，因为 [illegible] 的缺省实现最多只允许两种。

下面我对其做出一点改进 [illegible] 以便能自由控制文件过滤类型。

首先在 [illegible] 类中加入如下变量和函数押

```
#define NUM_FILTER 4
CMyApp: Public CWinApp
{
  ......
  private:
    CString szFilter;     //类型过滤字符串;
    CString szExt[NUM_FILTER];  //扩展名数组;
  public:
   //显示文件打开, 保存对话框;
    BOOL DoPromptFileName(CString& fileName, UINT nIDSTitle, DWORD lFlags, BOOL bOpenFileDialog, int & index);
    CString GetFilter();  //返回类型过滤字符串;
    CString GetExtension(int nIndex); //返回指定的扩展名;
    ......
};
```

[illegible] 是过滤类型的数目 [illegible] 根据实际要求指定 [illegible] 在此作为示例 [illegible] 我只定义为 [illegible] 源

```
CMyApp: : CMyApp()
{
 //初始化类型过滤字符串和扩展名数组;
 szFilter = "BMP Files( * . BMP) | * . BMP|GIF Files( * . GIF) | * . GIF|PCX Files( * . PCX) | * . PCX|JPEG Files( * . JPG) | * . JPG||";
 szExt[0] = ". BMP";
 szExt[1] = ". GIF";
 szExt[2] = ". PCX";
 szExt[3] = ". JPG";
}
BOOL CMyApp: : DoPromptFileName(CString & fileName, UINT nIDSTitle, DWORD lFlags, BOOL bOpenFileDialog, int & nIndex)
{
      CFileDialog dlgFile(bOpenFileDialog, NULL, NULL, lFlags, szFilter, NULL);
  CString title;
  VERIFY(title. LoadString(nIDSTitle));
  dlgFile. m_ofn. lpstrTitle = title;
dlgFile. m_ofn. lpstrFile = fileName. GetBuffer(_MAX_PATH);
```


需要全清版请访问 www.ertongbook.com

```
  dlgFile. m_ofn. nFilterIndex = (DWORD)nIndex; // 设置索引值;
        int nResult  = dlgFile. DoModal();
        fileName. ReleaseBuffer();
       nIndex = (int)dlgFile. m_ofn. nFilterIndex; //保存索引值;
        return nResult = =IDOK;
}
CString CMyApp: : GetFilter()
{
        return szFilter;
}
CString CMyApp: : GetExtension(int nIndex)
{
        CString szRes;
       szRes = szExt[nIndex - 1]; //注意索引值与下标相差一;
        return szRes;
}
```

在 惴酬髻装中响应 陽呃羻瓣藓菜单命令押

```
void CMyApp: : OnFileOpen()
{
        static int nIndex = 1; //过滤类型索引;
        CString PathName;
          if(DoPromptFileName(PathName, AFX_IDS_OPENFILE, OFN_HIDEREADONLY |  OFN_FILEMUSTEXIST, TRUE, nIndex))
       OpenDocumentFile(PathName);
}
```

然后在 惴酬罽黠类中重载辅助函数 阅瑟藓押

```
 CMyDoc: public CDocument
{
  ......
 public:
    virtual  BOOL  DoSave(LPCTSTR  lpszPathName,  BOOL bReplace  =  TRUE);
  ......
};
BOOL  CMyDoc: :  DoSave(LPCTSTR  lpszPathName,  BOOL bReplace)
{
        static int nIndex = 1;
        CString  newName  =  lpszPathName;
        if (newName. IsEmpty())
        {
            CDocTemplate *  pTemplate  =  GetDocTemplate();
               ASSERT(pTemplate ! = NULL);
               CMyApp  *  pApp = (CMyApp * )AfxGetApp();
                 ASSERT(pApp! = NULL);
    newName = m_strPathName; //m_strPathName 是路径名;
               if (bReplace & &  newName. IsEmpty())
               {
               newName  =  m_strTitle; //m_strTitle 是文档标题;
                      // 检查非法文件名;
            int iBad  =  newName. FindOneOf(_T(" #%; /\\"));
                     if (iBad ! =  -1)
                            newName. ReleaseBuffer(iBad);
                      // 在文件名中加入扩展名;
                      CString strExt;
                      strExt = pApp ->GetExtension(nIndex);
                      if(! strExt. IsEmpty())
                      {
                             ASSERT(strExt[0]  = =  '. ');
                             newName  + =  strExt;
                      }
               }
                 //显示文件保存对话框;
if(! pApp ->DoPromptFileName(newName, bReplace ? AFX_IDS_SAVEFILE: AFX_IDS_SAVEFILECOPY, OFN_HIDEREADONLY | OFN_PATHMUSTEXIST,  FALSE, nIndex))
           return FALSE;
    }
    CWaitCursor wait;
    if (! OnSaveDocument(newName))
    {
           return FALSE;
    }
    if (bReplace)  //覆盖文档的路径名
           SetPathName(newName);
    return TRUE;
}
```

实际的应用程序还应该重载 韵韵羻谢黠皂藓蟪韵犹蕨贮 遗粥豫藏藓蘑 韵韵羻谢黠皂藓蟪韵犹蕨贮 遗粥豫藏藓蘑和 阅藓黠犹谋嫌蘑函数，根据文件的类型处理文档数据。然后在 惴酬粼猎帏帏杖钢漕憎杭恍恍* 羡恍喣中显示文档。

本文程序的运行环境是 宊悦透嬲企业版、中文 宰蜜愿

收稿日期：圆圆圆年 远月 虎园日）

（上接第 远页）

```
Mail0. lines  =  Memo0. lines
M_attach  =  atom (Memo1. lines)
Mtoa (M_attach , Mail0. attachfiles)
Customer. gotop ()
Do while ! customer. eof ()
  If ! empty (customer ->e_add)
Edit1. text  =  alltrim (customer ->e_add)
Mail0. toaddress. lines[1]  =  edit1. text
If Mail0. sendmail ()  =  0
  Mess ("地址为  " + customer ->e_add  +  "  的客户, 发送失败! ")
Endif
Customer. skip (1)
  Endif
Enddo
```

写到这里，其实已经完全实现了所要的功能，执行程序，输入相应的主题，内容或附件，点击 鬥蹤镌鬗 你的 耘原酝藓赃就会到乖乖地跑到每个客户的信箱里了。

怎么样？很简单吧，如果想要做信件的接收，也不难，只是在接收时，要先算出信件的数量，然后逐一的查看。在 宊阅中还有很多的功能，能让你用简单的程序代码，轻松实现，有兴趣不妨试试看。

收稿日期：圆圆圆年 怨月 员怨日）

在中文 VFP 中自制 RTRIM 函数

周建军

摘 要 VFP 中的 RTRIM 函数不能有效的剪去中文字符串的尾空。本文分析了起因并自制了一个名为 MYRTRIM 的函数，以取代 VFP 中的 RTRIM 函数。

VFP 中的 RTRIM 函数是剪掉一个字符串的尾空，该函数在西文 VFP 中是可用的。而在中文 VFP 中却不能有效的利用（笔者在实际应用中发现该问题，从本文所给的示例可证明）。究其原因是由于 VFP 使用 DBCS 编码集存储和操作字符串。DBCS 编码集是一种用两个字节表示一个字符的字符集。另外一些程序，如 SQL Server 2000 使用 Unicode 编码集存储和操作字符串。因此在实际应用中，会遇到 DBCS 和 Unicode 之间的兼容性问题。下表列出了不同环境下 Unicode 和 DBCS 之间的兼容性问题。下表列出了不同环境下 Unicode 和 DBCS 字符集。

各种环境下使用的字符集：

VFP	DBCS
独立的数据对象库	DBCS
远程的数据对象库	Unicode 和 DBCS
SQL Server 2000	DBCS
Visual FoxPro 的 SQL Server 视图	DBCS
SQL Server 2000	Unicode 和 DBCS
Visual FoxPro 的 SQL Server 视图	DBCS

表 1 给出了 Unicode DBCS 和 DBCS 中 “粤” 的字符编码示例。

表 1 Unicode DBCS 和 DBCS 中 “粤” 的字符代码

A	ANSI 字符 "A"	&H41
A	Unicode 字符 "A"	&H41 &H00
A	DBCS 字符，表示一个日本的全角"A"	&H82 &H60
A	Unicode 全角 "A"	&H21 &HFF

从表 1 中可以看出：同一个 “粤” 字在 DBCS 字符集和 Unicode 字符集中其编码是不同的。VFP 中的 RTRIM （“字符串”）函数的工作原理是从字串尾开始逐个字节扫描 “字符串” 字串，一旦遇到编码不等于 32（空格符的编码）的字节，则停止扫描并去掉该字节以后的全部字节。在中文 VFP 中，由于使用 DBCS 编码集存储和操作字符串，即按双字节对字符串进行操作，此时的 RTRIM 函数实际上是以 DBCS（双字节）为单位对字符串进行扫描。由于 “RTRIM” 函数的扫描方法以及采用不同字符集的问题，导致 RTRIM 函数在中文 VFP 中不能正常使用。

本文笔者利用 VFP 中的 SUBSTR 函数自制了一个名 MYRTRIM 函数以取代中文 VFP 的 RTRIM 函数。SUBSTR 函数的用法于下：

SUBSTR（“字符串”，1，LEN（“字符串”））

其中：字串 “字符串” 既可以是一个字母，又可以是一个字串。不管是字节（SBCS）字符串还是字符串，它返回的是第一个字符的 ASCII 码值。藉此可把双字节的空格提取一个出来，以作为判断的依据。这样就可正确的把尾空剪掉了。

本文所给例中除了一个窗体外，还用了一个名为 Command1 的命令按钮控件和一个名为 Command2 的命令按钮控件，首先单击 Command1 命令按钮程序自动建立一个名为 Test.dat 的文件并往文件写入 5 条记录；然后单击 Command2 按钮程序打开 Test.dat 文件并顺序按记录读取该文件到内存，同时分别以自制的 MYRTRIM 函数和 VFP 中的 RTRIM 函数剪各个字串变量的尾空，同时各自把剪尾空后的字串变量值打印在窗体中。从输出结果可看到用自制的 MYRTRIM 函数可以有效地剪掉变量的尾空，而 VFP 的 RTRIM 函数却不能剪掉变量的尾空。实现程序于下：

```
Private Type arriy
 Name As String * 16
 Bh As String * 10
End Type
Dim iCount As Integer
' 自制剪尾空函数
Function RTrimW(strw As String) As String
  Dim i As Integer
  For i = Len(strw) To 1 Step -1
    ch = Mid(strw, i, 1)          ' 取子串
    If Asc(ch) <> & H20 And Asc(ch) <> 0 Then
     RTrimW = Mid(strw, 1, i)    ' 剪掉尾空后的字串
     Exit Function
    End If
   Next i
End Function
' 建立文件子程序
Private Sub CreatFile()
  Dim Myarriy(1 To 5) As arriy
  Dim i As Integer
   '用随机的方式打开/建立"test.dat"文件.
Open "test.dat" For Random As #1 Len = Len(Myarriy(1))
   For i = 1 To 5
    Myarriy(i).Name = "姓 名" & Format(i)   ' 给变量赋值
    Myarriy(i).Bh = " " & Format(i) & "#"
    Put #1, i, Myarriy(i)
   Next i
   iCount = i - 1          '写入文件的记录个数
   Close #1                ' 关闭文件.
End Sub
'显示用自制 RTrimW 函数和 RTrim 函数剪尾空后的结果
Private Sub ShowResult()
 Dim ShowArriy(1 To 6) As arriy
 Dim JiNum As Long
```

```
 Dim fileLength As Long
Open "test.dat" For Random As#1 Len = Len(ShowArriy(1))
 fileLength = LOF(1)  ' 取得文件长度.
 i = 1
 While i < = fileLength / Len(ShowArriy(1))
   Get #1, i, ShowArriy(i)    ' 从文件读数据到内存
   i = i + 1
 Wend
 iCount = i  '读了多少条记录
 Close #1    ' 关闭文件.
Print "以下是用自制 RTrimW 函数剪尾空"
 For i = 1 To iCount - 1
   Print RTrimW(ShowArriy(i).Name);
   Print RTrimW((ShowArriy(i).Bh))
 Next i
 Print "以下是用 VB 的 RTrim 函数剪尾空"
 For i = 1 To iCount - 1
   Print RTrim(ShowArriy(i).Name);     '
   Print RTrim(ShowArriy(i).Bh)        '
 Next i
End Sub
Private Sub cmdCreatFile_Click()    '按窗体中的 cmdCreat-
File 命令按钮
 CreatFile
End Sub
Private Sub cmdShowData_Click()    '按窗体中的 cmdShow-
Data 命令按钮
 ShowResult
End Sub
```

上述程序在 [illegible] 以及 [illegible] 下均调试通过。

名词解释

[illegible] 是个人计算机使用得最普遍的字符集。由于 [illegible] 标准使用单一字节表示每个字符，因此最多只能有 [illegible] 个字符和标点符号代码。虽然对英语来说已经足够了，但不能完全支持其它语言。

[illegible] 发行在亚洲大部分地区的 [illegible] 系统使用 [illegible] 它支持很多不同的东亚语言字母，如汉语、日语和朝鲜语。[illegible] 使用数字 [illegible] 表示 [illegible] 字符集。其它大于 [illegible] 的数字作为前导字节字符，它并不是真正的字符，只是简单的表明下一个字符属于非拉丁字符集。在 [illegible] 中，[illegible] 字符的长度是一个字节，而日语、朝鲜语和其它东亚字符的长度是 [illegible] 个字节。

[illegible] 是用两个字节表示每个字符的字符编码方案。国际标准组织 [illegible] 几乎为每种语言的每个字符和符号在 [illegible] 范围内定义了一个数字（再加上为将来发展保留的一些空余空间）。在所有 [illegible] 位版本的 [illegible] 中，部件对象模型 [illegible] 都使用 [illegible] 它是 [illegible] 和 [illegible] 技术的基础。[illegible] 全部支持 [illegible] 虽然 [illegible] 和 [illegible] 都是双字节字符，但它们的编码方案完全不同。

（收稿日期：[illegible] 年 [illegible] 月 [illegible] 日）

中国计算机学会会刊　全国科技核心期刊　中国计算机学会会刊　全国科技核心期刊

过去的 [illegible] 年里有你的相伴，我们一起经历了风风雨雨

未来的 [illegible] 年里我们将陪伴你，走向共同的成功和辉煌

欢迎订阅

《计算机应用与软件》（月刊）

邮发代号：[illegible]　[illegible] 年全年订阅价：[illegible] 元

国内计算机专业权威技术刊物

了解流行软件、网络和应用的窗口

交流、研讨先进技术和观念的阵地

中国计算机学会会刊　全国科技核心期刊　中国计算机学会会刊　全国科技核心期刊

阅读器源代码向超文本自动转化的实现

随着因特网的广泛应用，许多程序代码需要在因特网上发布。对习惯于集成开发环境的开发人员来说，在浏览器中看到类似开发环境中一样的程序代码显示风格可以比较快速地理解代码。比如在许多语言的集成开发环境中用蓝色斜体显示注释、用加粗黑体显示开发语言的保留字等等。如何在浏览器上达到同样的显示效果呢？问题的解决需要在源代码向超文本的转化上下功夫。本文将介绍 阅读器的程序代码向超文本自动转化的实现方法，并且提供自动转化的全部程序代码。

一、基础知识

1.阅读器程序代码基本组成

在 阅读器程序代码中包括注释部分和正文部分即非注释部分，在其集成开发环境中代码的显示风格是用蓝色斜体显示注释、用加粗黑体显示语言的保留字。为了要实现上述两个目的必须分析程序代码从哪里注释开始到哪里注释结束，挨个分析程序代码正文部分中的单词并且判断该单词是否为语言的保留字。

在 阅读器的程序代码中有三种注释方法：邀 妖对称注释方法、 （* *）对称注释方法、 辖到行结束注释方法。阅读器语言的保留字都是由 葬原字母组成，不区分大小写。

2.相关的超文本元素

在超文本中要实现一段文字的斜体显示是用元素组 约跃和 约辖跃 一段文字的加粗显示是用元素组 约月和 约辖跃 一段文字采用蓝色显示是用 约云操制浏览器时与蓝和 约辖操跃 另外要注意的是在浏览器中对超文本的解释会自动忽略空格和换行符，并且在显示时会自动折行，为了解决这一个问题可以选用 约孕和 约辖和 约辖跃元素组。元素符组 约孕和 约辖和 约辖跃之间的文字将不自动换行并且空格保留，而且对其中出现的超文本有效元素符作出解释执行。

3.字符串中可能引起的歧义分析

在程序代码字符串中可以使用任何字符。如在程序代码中有一个字符串为 ‘邀 ，字符串中有注释符号 ‘邀’，此时不能把它当成程序代码的注释部分而应当当成正文部分。而这样的字符串在 阅读器中是常见的，如 则 的表示就是一个用花括号对 邀括起的一串数字。自动转换程序实现对字符和字符串用红色显示。

4.注意超文本中的 “约” 符号

在超文本中用可以用 “约” 和 ‘跃’ 两个符号来进行对超文本注释。在浏览器解释过程中对这部分内容判断是否为有效的超文本元素，如果是做出相应的解释动作，否则全部当成注释部分直到 ‘跃’ 为止。所以在转化过程中对程序代码中的 “约” 符号转化为特殊超文本元素符 ‘驭邀’。

二、自动转化程序需要完成的处理

自动转化程序按文本的方式打开所要转换的 阅读器代码文件，然后依次读出一个个字符进行判断处理。概括起来转化程序需要做以下四个方面的处理工作。

1.注释部分的处理

当读到字符 ‘邀’ 时转化程序对后面所有读出的字符都当成注释部分处理，直到字符 ‘妖’ 为止。当读到字符 “辕’ 或 ‘穴’ 时转化程序根据下一个字符是否为 “辕’ 或 “* ” 判断是否为注释部分开始。如果是 “辕’ 注释，那么直到本行结束为止。如果是 “穴 ” 注释，当读到字符 “* ” 时转化程序根据下一个字符是否为 “雪’ 判断是否为注释部分结束。在注释部分里不做保留字的判断和字符串的判断处理。

2.正文部分的字符串处理

在 阅读器中字符和字符串的界定符号是单引号。一个编译通过的 阅读器代码中正文部分的单引号总是成对出现的。从这点可以理解成从代码头开始对正文部分的单引号累计，如果累计值是单数表示字符串开始，如果累计值是双数表示字符串结尾。在字符串里不做保留字的判断和注释部分的判断处理。

3.保留字的处理

在源代码的正文部分需要做保留字的处理。首先把 阅读器语言的所有保留字都用小写保存到 运 文件中，一行一个保留字。转化程序从该文件把把所有保留字读到一个字符串表 中，用 的 函数判断一个字符串是否在保留字表中，如果在保留字表中表示该字符串是 阅读器语言的保留字。在代码的正文部分如果读出的字符是 ‘葬’ 或 ‘扎’ 或 ‘粤’ 或 ‘在’ 并且在不是字符串的模式下就把字符累加到变量 运 中，否则用 运 的值同 运 表对比判断是否为保留字。

4.“约” 符号的处理

无论是在程序代码的正文部分，还是在程序代码的注释部分对 “约” 符号都必须把它转化成特殊的超文本元素符 ‘驭邀’。

三、自动转化程序实现方法

自动转化程序根据实际，把转化过程分成以下四种状

态：

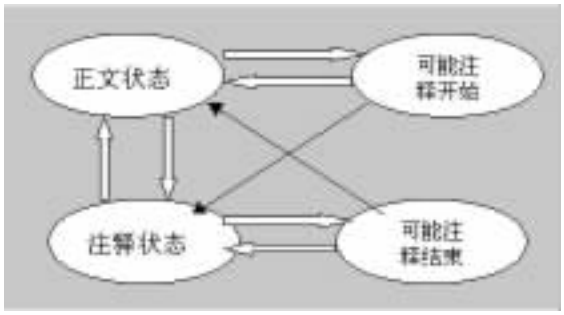
正文状态：自动转化程序分析到当前处在程序代码的非注释部分时的状态，它又分为字符串模式和非字符串模式。字符串模式自动转化程序分析到当前处在正文状态的字符串里，此时不做注释符号和保留字的判断处理。在非字符串模式要做注释符号和保留字的判断处理，所以在这时可能发生状态变化。在正文状态内部可能会发生字符串模式到非字符串模式之间的相互变化；

可能注释开始状态：在正文状态的非字符串模式时如果读到字符 “穴’ 或 “辕’ 时转入此状态；

注释状态：在 “邀’、 “ (* ”、 “辕袁’ 的注释过程中；

可能注释结束状态：在注释状态时如果读到字符 “* ”转入此状态。

这四种状态的转化可以见下图所示：



在正文状态读到 “邀’ 进入注释状态，读到 “ (” 或 “辕’ 进入可能注释开始状态；

在可能注释开始状态读到字符 “* ”或 “辕’ 进入注释状态，否则回到正文状态；

在注释状态读到字符 “妖’ 进入正文状态，读到 “* ”进入可能注释结束状态；

在可能注释结束状态读到字符 “) ”进入正文状态，否则回到注释状态；

程序中用一个枚举变量 [illegible] 来表示这四种状态。

用 [illegible] 表示正文状态、用 [illegible] 表示可能注释开始状态；

用 [illegible] 表示注释状态、用 [illegible] 表示可能注释结束状态。

四、转化程序运行结果

如图 员所示。

五、程序源码及保留字文本样式

程序是写成一个过程。入口参数为 阅[illegible] 的程序代码源文件名 穴[illegible] 转化成的超文本存放到目标文件的文件名 穴[illegible] 代码如下： 你可以不做任何修改粘贴到 阅[illegible] 程序中使用）

```
procedure pasToHtml(pascalFilenameame, htmlFilename: string);
  var
```



图 员

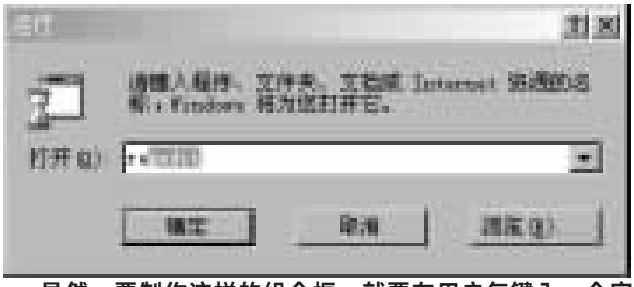
```
    leftch, C    : char;  //leftch 用于从正文状态到可能注释开始状态转变中保存 “{ ”和 “/ ”
    state        : (Copying, begincomment, incomment, endcomment);  //四种分析过程状态
  startstring: boolean; //正文状态下的字符串和非字符串模式
    Source       : TextFile;
    dest         : TextFile;
    keywordlist: TstringList;   //保留字字符串表
    keyword      : string;
  begin
    keywordlist: =TStringList. create;
    keywordlist. LoadFromFile(´keyword. txt´);
//从保留字文件中读入所有保留字到字符串表 KeywordList
    assignFile(Source, pascalFilename);
    reset(Source);
    assignFile(dest, htmlFilename);
    rewrite(Dest);
    writeln(Dest, ´ <´HTML>´);
    writeln(Dest, ´ <Head>´, ´ <Title>´ +HTMLFilename +
´ <´/Title>´, ´ <´/head>´);
    writeln(Dest, ´ <Body>´);
    writeln(Dest, ´ <Pre>´);
//把超文本设置成不自动换行并且保留空格位置
    //初始值设置
    state: =copying;    //正文状态
    startstring: =false;   //正文状态下非字符串模式
    keyword: =´´;       //保留字
    leftch: =#0;
    while not(eof(Source)) do
      begin
       //当一行结束时需要的工作
        if eoln(Source) then
          begin
            readln(Source);
```

```
//如果是在正文状态非字符串模式必须判断保留字
    if (length(keyword)>0) and (state=copying)
and (not(startstring)) then
    begin
  if keywordlist.IndexOf(lowercase(keyword))<>-1
    then write(dest,'<B>',keyword,'</B>')
      else write(Dest,keyword);
    keyword:='';
  end;
  writeln(Dest);
//如果是"//"的注释状态换行后状态必须转换到正文状态
  if (state=incomment) and (leftch='/') then
    begin
     state:=copying;
     write(Dest,'</I>','</Font>');
    end;
  end
  else
  begin
    read(Source,C);
    case state of
      copying: begin
        if startstring then    //如果是字符串模式
          begin
            write(Dest,C);
            if c='''' then
              begin
                startstring:=false;
                write(Dest,'</Font>');
              end;
          end
//在正文状态非字符串模式下如果遇上的非字母字符需要做保留字判断
        else if not(C in ['a'..'z','A'..'Z']) then
          begin
            if length(keyword)>0 then
              begin
  if keywordlist.IndexOf(lowercase(keyword))<>-1
    then  write(dest,'<B>',keyword,'</B>')
      else write(Dest,keyword);
        keyword:='';
      end;
  case c of   //不是字母字符情况下可能状态变化的字符
    '(','/': state:=begincomment;
      '{'    : begin
          state:=incomment;
    write(Dest,'<Font Color=blue>','<I>','{');
          end;
     ''''    : begin
          startstring:=true;
        write(Dest,'<Font Color=Red>');
        write(Dest,C);
          end;
    else
        write(Dest,C);
    end;
  end
  else keyword:=keyword+c;
    leftch:=C;
  end;
//如果是可能注释开始状态判断是"(*"注释还是"//"注释
  begincomment: if (c='*') and (leftch='(') then
    begin
  write(Dest,'<Font Color=blue>','<I>','(*');
    state:=incomment;
    end
  else
    if (c='/') and (leftch='/') then
      begin
  write(Dest,'<Font Color=blue>','<I>','//');
      state:=incomment;
    end
  else
    begin
      write(dest,leftch);
      state:=copying;
    case c of
      'a'..'z','A'..'Z': keyword:=C;
    '''' : begin   //如果非注释状态对单引号判断
     startstring:=not(startstring);
      if startstring
    then write(Dest,'<Font Color=Red>',C)
      else write(Dest,C,'</Font>');
      end;
    else
      write(Dest,C);
      end;
    end;
  incomment: begin
      if (c='*') and (leftch='(')
        then state:=endcomment
      else if (C='}') and (leftch='{')
        then begin
          state:=copying;
    write(Dest,'}</I>','</Font>');
        end;
      if C<>'}' then write(Dest,c);
    end;
  endcomment: begin
      if c=')' then begin
        state:=copying;
    write(Dest,')</I>','</Font>');
        end
      else
      begin
        write(dest,'*');
        write(Dest,C);
      end;
    end;
  end;
 end;
writeln(Dest,'</pre>');
writeln(Dest,'</body>');
```

创建自动完成的组合框

元凯宁

在 宰蛮燃糟彩辫赖惫的开始菜单中，有一菜单项名叫 ‘运行”，界面如下图所示。当我们在其中键入一些文字时，它会将它们记录下来。等到我们下次再键入同样的内容时，会有一种很 ‘智能”的功能出现：组合框中的文字会自动匹配原来曾经键入过的内容，基本上只要键入前几个字母就可以得到完整的内容了。这就是组合框的 ‘自动完成”功能。下面，我们就来尝试制做这种自动完成的组合框。



显然，要制作这样的组合框，就要在用户每键入一个字符之后都进行一系列的处理：即判断输入的内容是否是某一列表项的从头开始的若干个字符，然后将整个字串设为编辑框的内容，再将用户还未输入的字符 剩下的）设置为选择状态。

寻找特定的列表项的工作可由 恍恍毫遭到墨类的成员函数 杂藏糟糟据辱雪来完成，这个函数在找到列表项之后还将它设置为组合框的编辑框的内容。如果有多个列表项的前若干个字符相同而又同时符合条件，则 杂藏糟糟据辱雪只找到序号较低的 在列表中靠上的）那一个。

将用户还未输入的字符设置为选择状态的工作过程可描述成以下步骤：第一步，保存编辑框中的文字；第二步，先查找匹配的列表项 同时也就设置了编辑框中的文字），如果不成功，则恢复原来的文字；第三步，将自动添加的文字设置为选择状态。

所有的这些工作应当在用户的键入动作完成之后进行，粗略看来，有两种选择： 恍手最嘘河戴说沟弯颖转和 恍手最嘘河戴说郧阅弯颖转两种控制通知消息可供处理，但 恍手最嘘河戴说沟弯颖转是在用户对编辑框中的文字进行了改变之后，即屏幕上的文字已经改变了之后才发送，而 恍手最嘘河戴说郧阅弯颖转是在编辑框即将显示改变的文字 还未改变）时发送。所以，应当在 恍手最嘘河戴说郧阅弯颖转的消息处理器中进行上述的操作。

还有一个问题，在用户使用 月箭标赛箭键和 阅翼膝键的时候，不应当进行上述的过程，但当用户使用 月箭标赛箭键和 阅翼膝键后，恍手最嘘河戴说郧阅弯颖转消息同样会被发送。为此，我们加入一个作为标志的成员变量 皂则整恍戴说标志 并重载 孕则郧耕阻止颖转显示篇辱雪函数，当用户按下 月箭标赛箭键和 阅翼膝键时，在 孕则阻止颖转显示篇辱雪函数中设此标志为 云整篇转，否则设置此标志为 耘颖转。在 酚杜显戴说郧阅弯颖转函数中判断此标志，若为 云整篇转，则直接返回；若为 耘颖转，则进行上述的处理过程。

下面是程序代码：

头文件：

```
#if !defined(_EWAY_AUTOCOMPLETE_COMBOBOX_H_
INCLUDED_)
#define _EWAY_AUTOCOMPLETE_COMBOBOX_H_INCLUDED_
#if _MSC_VER >= 1000
#pragma once
#endif // _MSC_VER >= 1000
// AutoCompleteComboBox.h : header file
class CAutoCompleteComboBox : public CComboBox
{
public:
  CAutoCompleteComboBox();
```

```
  writeln(Dest, ´</html>´);
  closefile(Dest);
  closefile(Source);
  keywordlist.free;
end;
```

保留字文本是每个保留字占一行用小写字母组成。如下所示：

```
assignfile
begin
class
end
```

...

六、转化程序运行的要求

该自动转化程序的运行要求所转化的程序代码通过了 阅翼膝的编译，这实际也是在网上发布最起码的要求。

自动转化程序是在 阅翼膝郧阅园中开发通过，对 责择糟造阅翼膝其他版本的程序代码同样使用。本人根据上述原理，还完成实现 灾云字源代码向超文本自动转化、龙辑源代码向超文本自动转化的程序。有兴趣的可以向本人免费索要。

收稿日期：圆斑斑年 苑月 圆斑日）

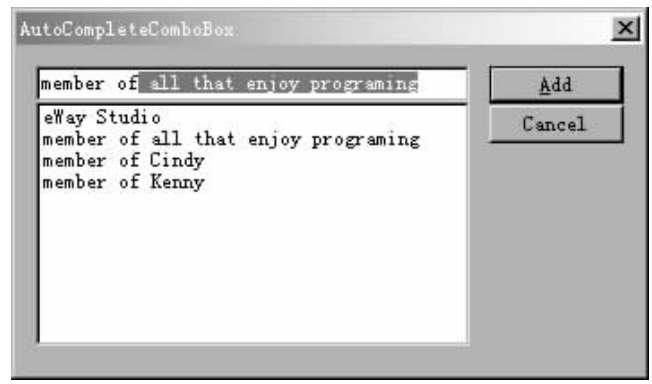
```
public:
    virtual ~CAutoCompleteComboBox();
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CAutoCompleteComboBox)
    public:
    virtual BOOL PreTranslateMessage(MSG* pMsg);
    //}}AFX_VIRTUAL
protected:
    BOOL m_bWorkOnText;
    //{{AFX_MSG(CAutoCompleteComboBox)
    afx_msg void OnEditUpdate();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
//{{AFX_INSERT_LOCATION}}
// Microsoft Developer Studio will insert additional declarations immediately before the previous line.
#endif // !defined(_EWAY_AUTOCOMPLETE_COMBOBOX_H_INCLUDED_)
```

下面是 糟责文件：

```
// AutoCompleteComboBox.cpp : implementation file
#include "stdafx.h"
#include "AutoCompleteComboBox.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////
// CAutoCompleteComboBox
CAutoCompleteComboBox::CAutoCompleteComboBox()
{
    m_bWorkOnText = TRUE;
}
CAutoCompleteComboBox::~CAutoCompleteComboBox()
{
}
BEGIN_MESSAGE_MAP(CAutoCompleteComboBox, CComboBox)
    //{{AFX_MSG_MAP(CAutoCompleteComboBox)
    ON_CONTROL_REFLECT(CBN_EDITUPDATE, OnEditUpdate)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
/////////////////////////////////////////
// CAutoCompleteComboBox message handlers
BOOL CAutoCompleteComboBox::PreTranslateMessage(MSG* pMsg)
{
    if (pMsg->message == WM_KEYDOWN)
    {
        int nVKey = (int) pMsg->wParam;
        m_bWorkOnText = (nVKey == VK_DELETE || nVKey == VK_BACK)?FALSE:TRUE;
    }
    return CComboBox::PreTranslateMessage(pMsg);
}
void CAutoCompleteComboBox::OnEditUpdate()
{
    //如果按下的是 VK_BACK 和 VK_DELETE, 不进行处理;
if (!m_bWorkOnText)
return;
//取得编辑框中的文字;
CString strWindowText;
GetWindowText(strWindowText);
int nLength = strWindowText.GetLength();
//取得当前的选择范围
DWORD dwCurSel = GetEditSel();
WORD dStart = LOWORD(dwCurSel);
WORD dEnd = HIWORD(dwCurSel);
//实际做查找和选择的语句
if (SelectString(-1, strWindowText) == CB_ERR)
{
SetWindowText(strWindowText);        //恢复文字
        if (dwCurSel != CB_ERR)
            SetEditSel(dStart, dEnd); //恢复原先的位置
}
//把添加的文字做为选择的文字
if (dEnd < nLength && dwCurSel != CB_ERR)
SetEditSel(dStart, dEnd); //避免过多地选择文字
else
    SetEditSel(nLength, -1);
}
```

要使用这种组合框，只需在对话框模板中画出一个组合框 但其风格不能为 [illegible] 赞赞），在工程中添加上面的这两个文件，在 [illegible] 的 [illegible] 页签中为组合框添加成员时，选择 [illegible] 再选 [illegible] 即可。最后，别忘了在对话框的 糟责文件中添上一句：[illegible]。

下图是一个例子运行时的状态。



收稿日期：[illegible]

需要全本请访问 www.ertongbook.com