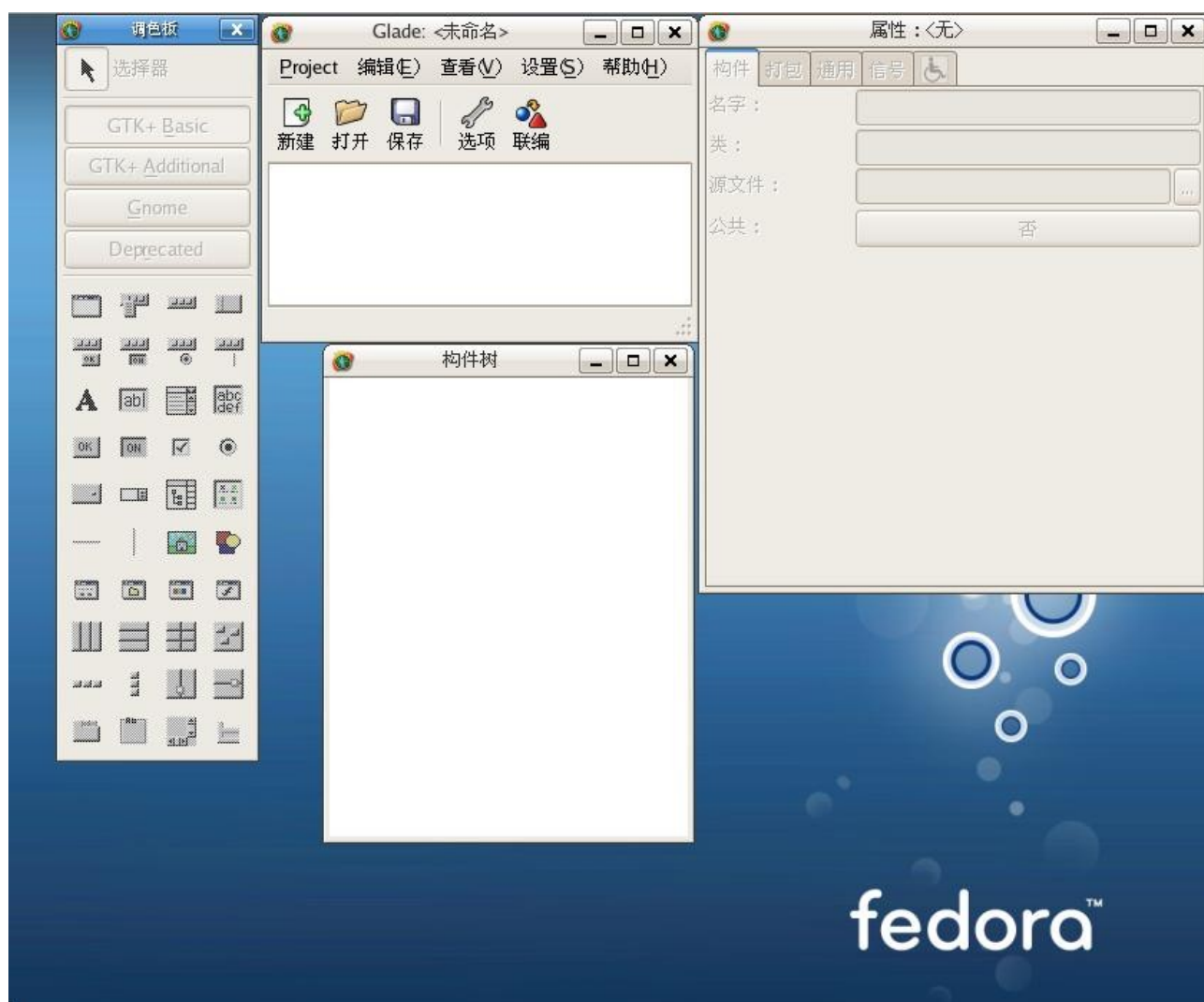


Glade 编程 学以致用



内容目录

序.....	4
Glade 是什么?	7
GTK+是什么?	7
壹、在文本通下建立编译环境.....	8
贰、打开已有 Glade 工程.....	9
一、运行 Galde.....	9
二、打开已有 Glade 工程文件.....	9
三、显示相关的窗口.....	9
四、激活工程项目.....	10
五、激活子构件.....	10
六、工程文件选项设定.....	14
七、新建一个文件夹.....	15
八、改变文件选项.....	16
九、保存工程文件.....	16
十、代码联编.....	16
十一、新建 2 个终端窗口.....	16
十二、生成编译环境.....	16
十三、编译.....	16
十四、运行.....	16
十五、小结.....	16
叁、新建 Glade 工程.....	19
一、运行 Galde.....	19
二、新建一个 Glade 工程.....	19
三、选择工程类型.....	19
四、点 [调色板] 窗口中的 [窗口] 按钮.....	19
五、出现一个界面窗口 window1.....	20
六、将其它容器或部件放入其中.....	20
七、编辑属性以适合特殊需要.....	20
八、其余.....	20
肆、将功能代码加入联编代码中.....	21
一、加入宏定义、头文件和外部变量声明.....	21
1、加入外部变量声明.....	21
2、加入宏定义.....	22
3、加入头文件.....	22
二、加入各种功能模块.....	22
1、加入退出模块.....	22
2、加入文件查找模块.....	22
3、加入文件名保存模块.....	23
4、加入树视图控件的显示控制模块.....	23
5、加入文本视图控件的显示控制模块.....	24
6、显示图片.....	24
7、加入字符串的计数、分解、转换、连接、计算功能.....	25
伍、修改界面布局.....	25

陆、加入过程对话 GUI.....	26
一、用 GLADE 建立一个窗口.....	26
二、修改 main.c.....	26
三、修改 interface.c.....	27
1、对应全局变量进行替换设置.....	27
2、在事件反馈部分增加删除过程 GUI 的语句.....	28
四、在 callback.c 中加入调用初始化和循环语句.....	28
五、小结.....	29
柒、制作 deb 包.....	30
1、确定软件名	30
2、新建打包目录	31
3、新建信息目录	31
4、新建安装目录	31
5、准备文件	32
1) 信息文件 control	32
2) 项目文件	33
6、打包	34
1) 安装 fakeroot	34
2) 打包命令	34
3) 测试安装	34
4) 测试删除	34
5) 错误排除	34
7、简化的方法.....	35
八、附录.....	35
1、main.c.....	35
2、interface.c.....	36
3、callback.h.....	40
4、callback.c.....	43



序

这是“如意”软件的开发笔记，把它拿出来是希望通过一个小软件的开发心得，帮助有心编程的朋友熟悉 Glade 的强大、易用，进而学会使用。

本版的更新主要是加入了怎样安装 ubuntu（个人译为“文本通”，下同）的编译环境和 deb 包的制作，并加入了第陆章加入过程对话 GUI 内容，其余未变。

我不是一个专业的编程人员，也不是一个喜欢钻计算机专业牛角尖的主，但面对已经成熟，蒸蒸日上几臻于完美之境的开源软件集，我不得不动心，去热情地拥抱她，也许软件的“开元盛世”已经来临了！

从不是一个自由软件开发人员，到加入其中，是因为首先从中受益匪浅。面对伸手就要钱的商业软件圈，一个爱好编程却又缺乏编译器的业余人员是不敢或懒得问津的，通过一系列的盗版，获得一个不稳定的编程环境，API 等相关资料残缺不全，这样的环境还有几分创新的冲动呢？

自由软件则不然，免费、高效、稳定、强大、相关资料齐全等都是已经形成的共识，3G 自由软件套件：Gcc 编译器及 MAKE 工具、GTK+/GNOME 图形界面库、Glade 图形界面构造工具，是已经成熟的编程工具集套件；正如著名的互联网服务器套件 LAMP：Linux、Apache、MySQL 和 PHP,Python,Perl 编程语言已闻名遐迩一般；其他如桌面领域的 Openoffice.org 的文档格式已成为国际标准，相应办公套件在 Linux 和 MS Windows 应用越来越广泛；firefox 已成为 Linux 发行版的必选浏览器，也已成为 MS Windows 下避免流氓软件、木马软件、病毒的有效浏览器；GIMP 图像编辑软件也正以它媲美 Photoshop 的强大功能日渐为大家所熟悉；品质上乘的 XINE、Mplayer、VLA、Geexbox 等媒体播放软件日渐流行；Hiweed DeskTop、KNOPPIX、Ubuntu 等集合整套办公、媒体、操作系统的启动光盘成为新的计算机时尚；可定制的 Gentoo Linux 操作系统日渐为计算机技术迷们所追捧等等，自由软件的风光真是一片大好，说“开元”（元与源本来就相通）盛世已经到来实在是不为过。

自由软件的流行当然是因为它免费或便宜，但更重要的却是首先因为它们的开放、稳定、强大，就象本文介绍的 3G 组合——GCC、GTK+、Glade。在自由软件领域，除了兴趣之外，对你别无要求，你是完全自由的，实际上，连兴趣也不能算是一种要求。

但即使在这样的自由环境，学会编程也不是一件一蹴而就的事，基于这样的事实，我开发了一个不用编程、学习计算机语言就能创造程序般功能环境——“如意”软件，集浏览器、摘录器、计算器、向导、自动运行功能于一身，兼容普通纯文本到专门纯文本，兼容手工和自动，可以更方便地让普通人的思路精确地得到交流，而它的要求只是纯文本的记录和分段、断句数行数等日常的识字工作，如果你不只对编程感兴趣，可以参阅该软件和同时发行的手册，使用该软件，相信不会让你失望。

如果你觉得好，不妨介绍给你的朋友——它是自由软件，免费发行，采

用 GNU 通用公共许可证 2007.6 第三版 (GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007)。欢迎访问 <http://www.qvandi.com>, 阅读、下载, 使用手册采用 pdf 格式和软件支持的纯文本 xdf 格式两种形式。

闲言少叙, 如果您是程序员, 或想成为程序员, 那你就开始愉快地阅读下面的内容吧。

本文将向你介绍 GTK+ 的图形界面编程, 它面向如下对象:

- 1) 如果您是一个 C 程序员, 但还不是老手, 那么通过本文, 可以补充一下相关的知识;
- 2) 如果您是一个 C 程序老手, 但还没有在 Linux 下编过图形界面程序, 那么本文的 Glade 知识会很适合你;
- 3) 如果您已经是 Glade 的使用者, 则本文完全从 Glade 编程的思路也许能供你参考;
- 4) 如果您已经是 Glade 的高手, 那么有可能的话您就来帮我一起将示例的软件——“如意”软件 (mindway) 做得更好吧; 因为它是 GNU 通用公共许可证 2007.6 第三版 (GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007) 授权下的自由软件, 或许你本身也会从中受益。

本文介绍的开发环境与前一版本有所不同, 前一版是基于 Fedora core 6 Linux(FC6)环境的, 未使用额外的工具; 随着 Ubuntu (下译为“文本通”) 在桌面领域的崛起, 目前已成为最受欢迎的桌面发行版, 未来的 Linux 桌面应用恐怕将主要居于文本通, 因为它做得更简易、绚烂, 技术上来说, 它采用的 Debian 包管理器确实具有更大的优势, 它改变了软件的安装模式, 使您不用关心软件的精确位置, 只要一条命令, 或在新立得软件包管理器里勾选一下, 就可以将软件安装到电脑里, 当然, 对于软件源里没有的软件, 它也支持传统的双击安装。Redhat 公司已宣布, 未来将采用 Debian 包管理系统, 这样, 作为开发者, 选择文本通平台将是合适的。

不过, 就目前来说, Redhat 公司支持的 Fedora Core 对开发者来说仍有其独特的好处, 因为它的发行版是默认安装 GTK+ 开发环境的, 而文本通由于面向普通大众, 默认不安装开发环境, 您需要有互联网的宽带连接, 按照本文的一些简单命令来安装。

本文的开发环境它包括了图形界面构造器 Glade 2.12.1, GNU 编译器 gcc4.1.1, GTK 图形界面库 GTK+-2.6.0

购买 FC6——60 元左右可以得到一套光盘 (12 张) 和 93 页的安装手册, 建议购买,

它会使你事半功倍。如果你没有购买, 那就从网上免费下载这些工具和库, 它们都自由软件基金会 (The Free Software Foundation) 的项目,

您可以去 <http://www.gtk.org> 去了解有关 GTK+ 的知识, 下载库接口 API 手册, 这很重要。

您可以去 <http://glade.pn.org> 去了解 Glade 的有关知识。

编程不是闭门造车, 您需要一些参考书籍——“如意”软件就是在阅读

这些参考中逐渐编成的，鸣谢下面书籍的相关人员：

- ◆ <http://www.gtk.org> 网站有关 GTK+ 的 API 电子文档
- ◆ Tim-Philipp Müller 的 “GTK+ 2.0 Tree View Tutorial” 电子文档
- ◆ 宋国伟编著，GTK+2.0 编程范例，清华大学出版社，2002 年 11 月第 1 版
- ◆ Peter Wright 著，钟鸣、石永平等译，GTK+ / GNOME 程序设计，机械工业出版社，2002 年 1 月第 1 版
- ◆ Brian W. Kernighan, Dennis M. Ritchie 的 “C 程序设计语言（第 2 版新版）”，机械工业出版社，2006 年 7 月第 2 版；
- ◆ Stephen Prata 的 “C Primer Plus(第五版)中文版”，人民邮电出版社，2005 年 2 月第 1 版
- ◆ Ubuntu 7.10 Linux 建立 C 与 GTK 的开发环境, 发布时间：2008.01.15 06:20
来源：赛迪网 作者：korn

联系：

网站：<http://www.qvandi.com>

<http://sourceforge.net/projects/mindway>

E-mail:s.jg255@shl63d.sta.net.cn

QQ:609074072

遵循 [GNU Free Documentation License](#) 发布，您可以自由复制发布。

本教程由 ksnapshot 在 Fedora Core 5 Linux 下载图，用 OpenOffice.org2.2 在文本通 7.04 下编辑并生成 PDF，鸣谢相关项目人员！

福利

2007 年元月首发

2008 年四月修订



Glade 是什么？

从封面截图可以看到 Glade 运行时的样子，它是专门为构建 GTK+ 库基础上的图形界面而设计的工具。

简单说它是一套可以为你联编代码的图形界面构造器，
有了它，您就只要关心怎样象摆积木一样组装图形界面，
有了它，不会 C 语言也能生成由 C 语言组成的图形界面。
在这个基础上，你只须用 C 语言来完成具体的功能模块就行了，
读完下文，您就能了解它是如何的简单、神奇、激动人心！！！！

有人批评 Glade 的文档做得不好，我不同意这种说法，我觉得是因为 Glade 做得太好，以至于文档都显多余。我只看过 Glade 的介绍，但我却毫不费力地依靠它做出了“如意”软件算得精美的界面，它的所见即所得，调色板、构件树、属性的界面让人一目了然，可谓此时无声胜有声，阅读下面的内容后您也许会发现，这不只是我一个人的感受。



GTK+ 是什么？

它是 GIMP Toolkit 的缩写，是一个开发图形界面的库，它是当前主流桌面管理器 GNOME 的基础，GTK+ 是一个跨平台的库，因此即使对于 MSWindows 的用户，也会很熟悉它。与 GTK+ 并行的就是著名的 Qt 库，它是个 C++ 库，是当前 Linux 桌面下另一个强大的桌面——KDE 的基础。这两个项目目前处于良性竞争中，伯仲之间，难分高下，也有融合的趋势。

不过，个人偏好 GTK+，它有一个非常独特的特点：它是由 C 语言来实现面向对象的编程的，所以用它来进行图形界面编程，您无须学习 C++ 而只要 C 语言基础就够了，本人目前只会 C 编程。



壹、在文本通下建立编译环境

1、配置 GCC

刚装好的系统中已经有 GCC 了，但是这个 GCC 什么文件都不能编译，因为没有一些必须的头文件，所以要安装 build-essential 这个软件包，安装了这个包会自动安装上 g++, libc6-dev, linux-libc-dev, libstdc++6-4.1-dev 等一些必须的软件和头文件的库。

可以图形界面安装：

点选系统菜单[系统]-> [系统管理] -> [新立得软件包管理器]启动 GUI 安装，->找到 build-essential->右键-> [标记以便安装] -> [应用]

或输入下面命令：

```
sudo apt-get install build-essential
这将安装 gcc/g++/gdb/make 等基本编程工具
```

2、GTK 的安装

安装 libgtk2.0-dev libglib2.0-dev 等开发相关的库文件

```
sudo apt-get install gnome-core-devel
```

安装用于在编译 GTK 程序时自动找出头文件及库文件位置

```
sudo apt-get install pkg-config
```

安装基于 GTK 的界面构造程序 Glade，本文的主角。

```
sudo apt-get install glade libglade2-dev
```

下面为可选项：

```
sudo apt-get install devhelp #这将安装 devhelp GTK 文档查看程序
```

```
sudo apt-get install libglib2.0-doc libgtk2.0-doc #这将安装 gtk/glib 的 API 参考手册及其它帮助文档
```

如果使用 vim 来编写代码，当然您可以使用任何自己喜欢的编辑器，比如用系统的文本编辑器 gedit 也不错

```
sudo apt-get install vim
```




贰、打开已有 Glade 工程

一、运行 Glade

点桌面上的 [应用程序] —> [编程] —> [Glade Interface Designer]
运行后得到如第 1 页图片所示的界面

二、打开已有 Glade 工程文件

点界面中的 [打开] 按钮，寻找到软件的工程文件——mindway.glade，
成为如图 2 所示的界面



图 2 打开 Glade 工程文件

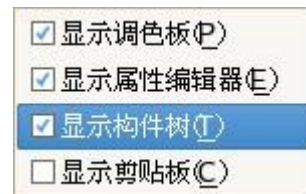


图 3、查看选项

三、显示相关的窗口

如果你运行的界面与图中不相同，可以点菜单的 [查看] 项，然后选中
如图 3 所示的几项。

四、激活工程项目

双击界面的的工程文件名 [mindway], 则相关的属性、构件树、构造的目标界面都会展开如图 4。

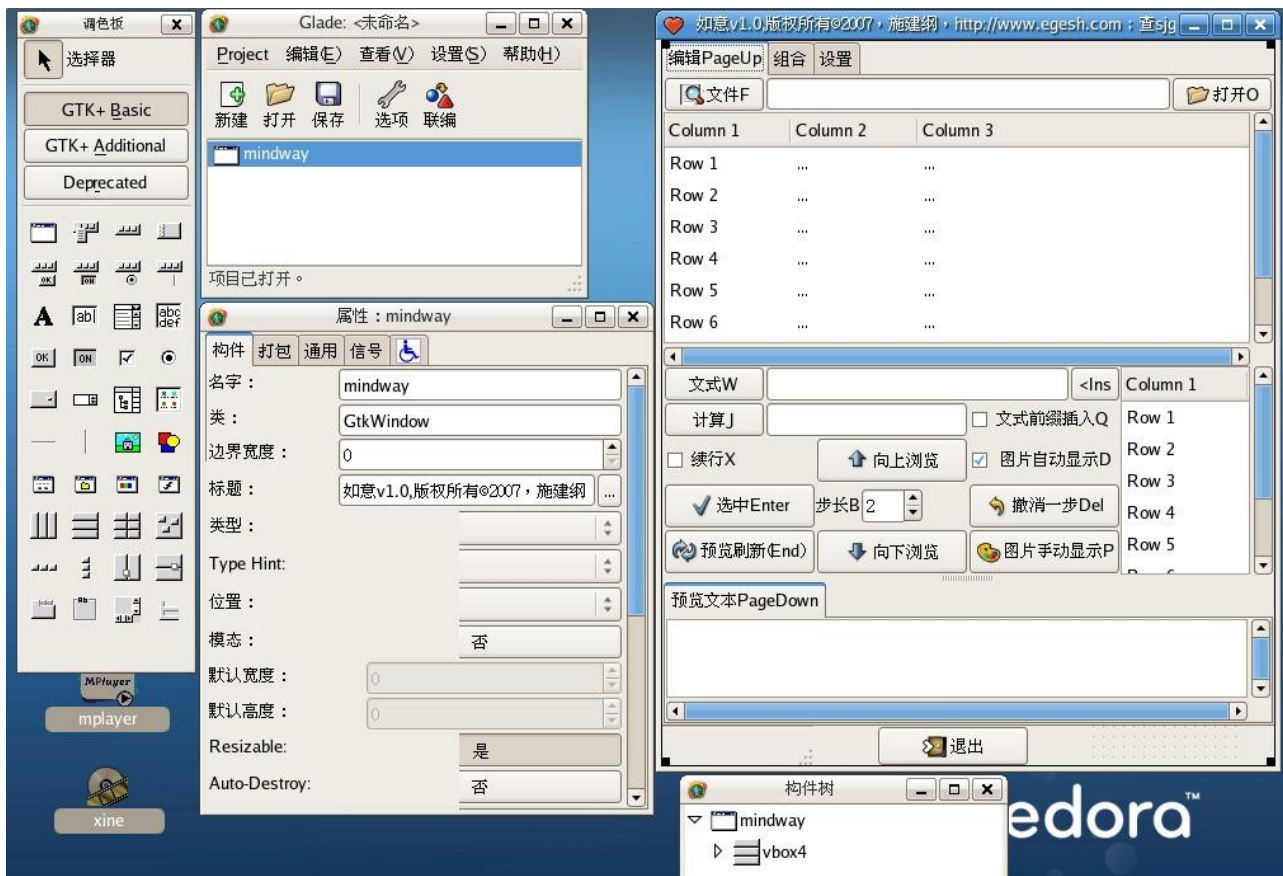


图 4、激活 Glade 工程文件

五、激活子构件

单击子构件，则相关的属性、构件树、构造的目标界面都会相应展开如图 5，此时你可以改变属性窗口的内容来进行定制，属性包括：

1、构件属性

可以更改界面部件的名字，图标、默认值、标签等参数，如图 6。

2、打包属性

可以更改界面部件的扩展、位置属性，如图 7。

3、通用属性

可以更改界面部件的宽度、高度、可见、不可见、工具提示信息、快捷键等属性。工具提示信息编辑见图 8，快捷键属性的修改见图 9。

4、信号

可以更改触发信号及调用子程的名字见图 10，信息列表见图 11、12、13

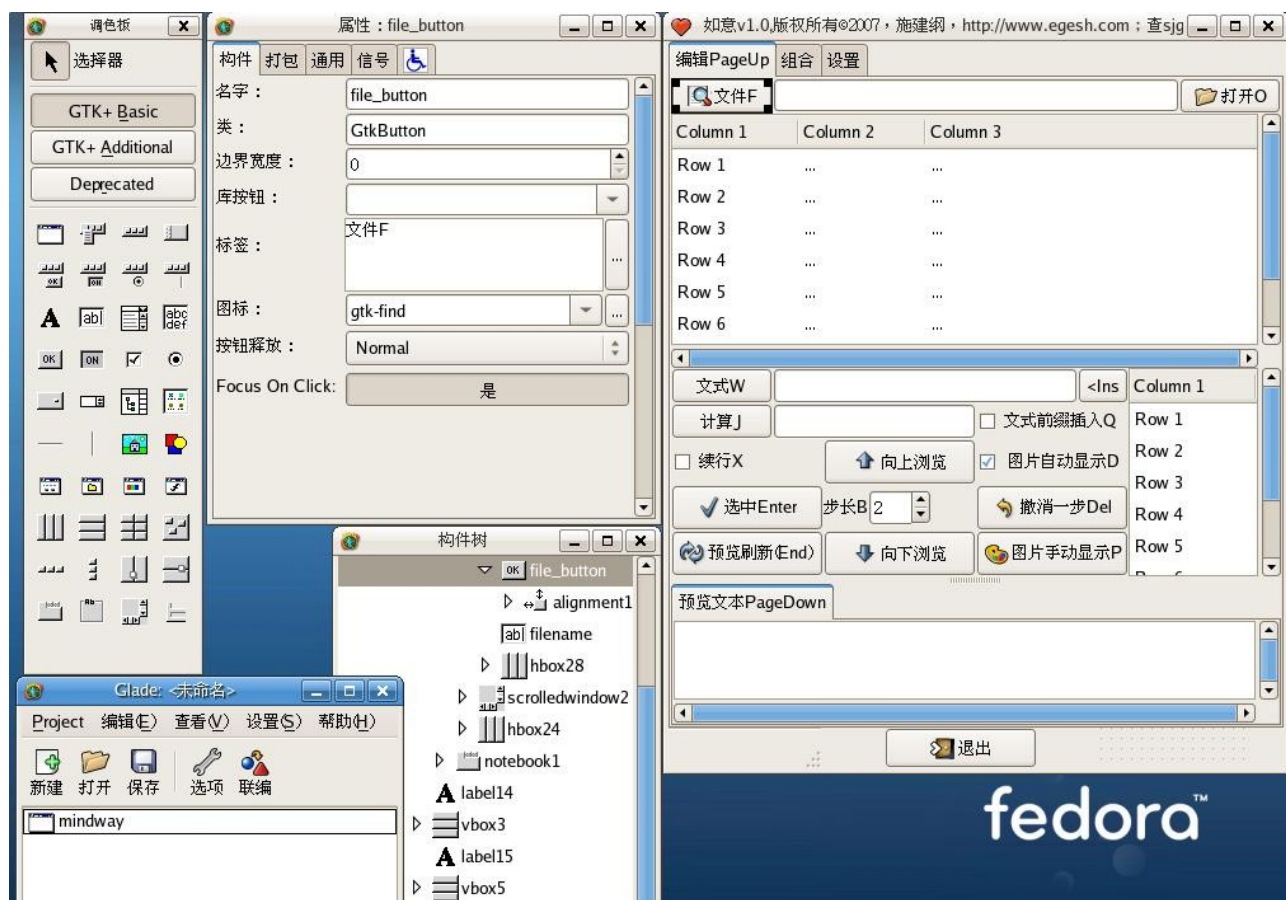


图 5、激活子构件

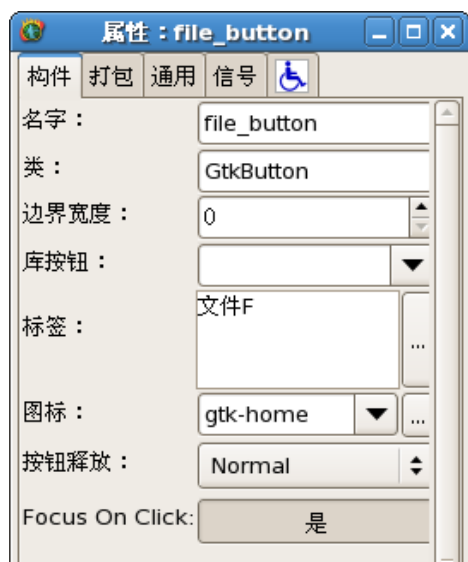


图 6: 构件属性

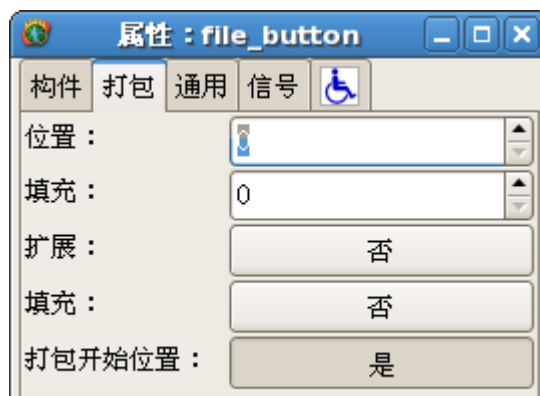


图 7: 打包属性

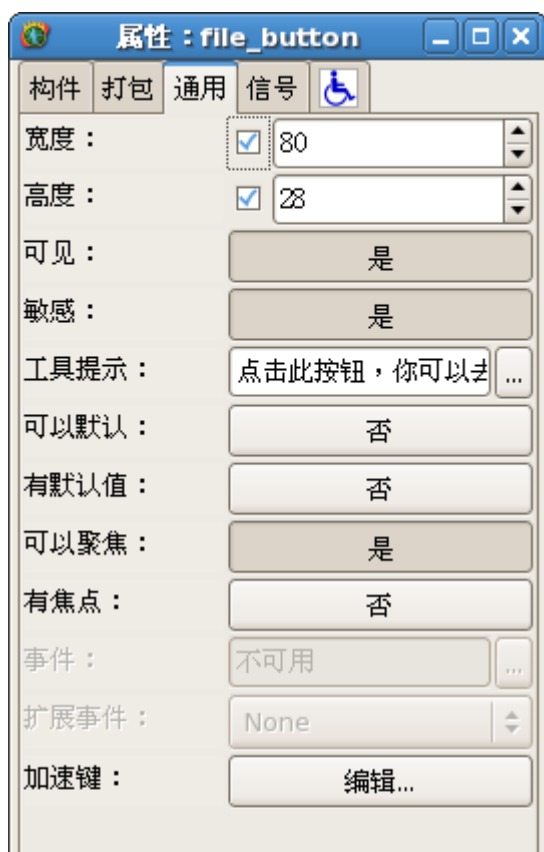


图 8: 通用构件

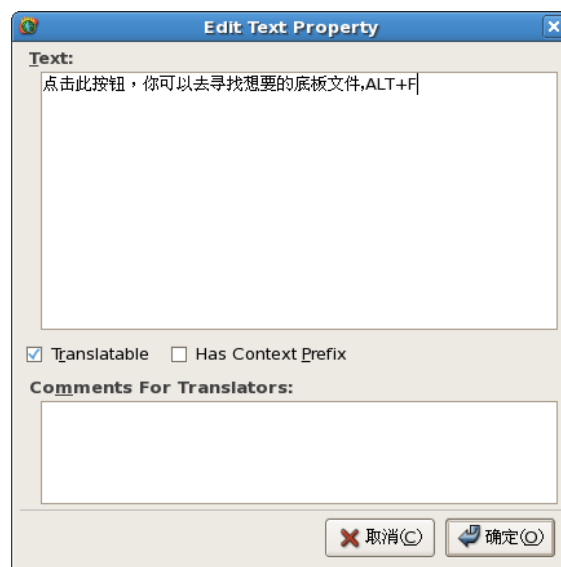


图 9: 编辑工具提示



图 10: 编辑快捷键



图 11: 信号反馈程序编辑

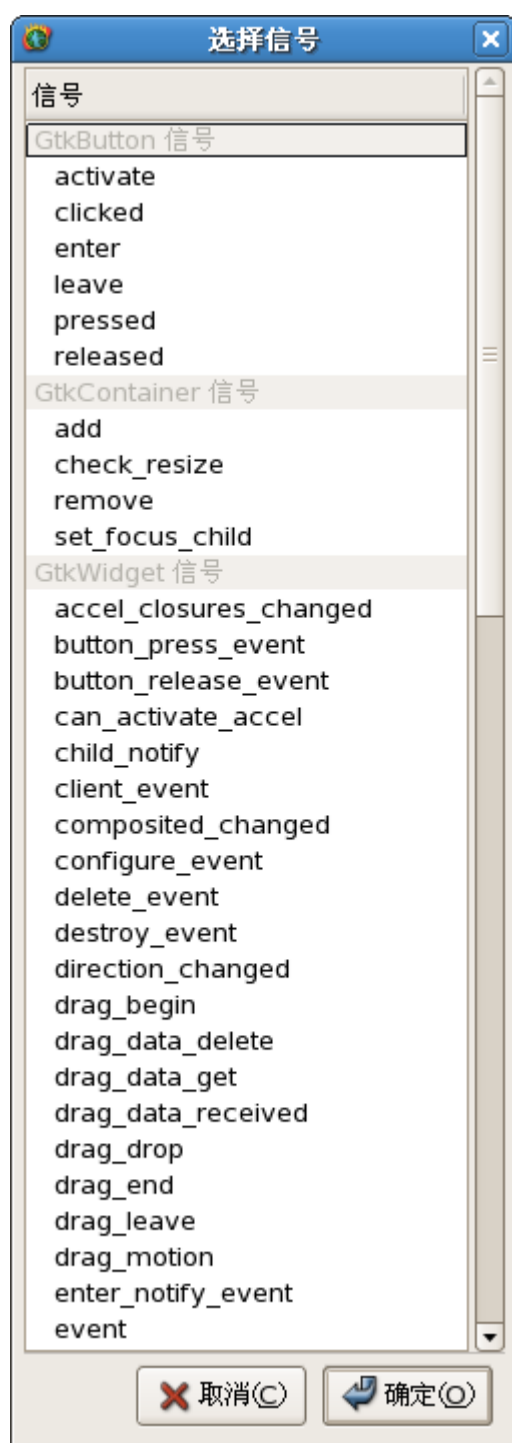


图 12: 信号列表 1

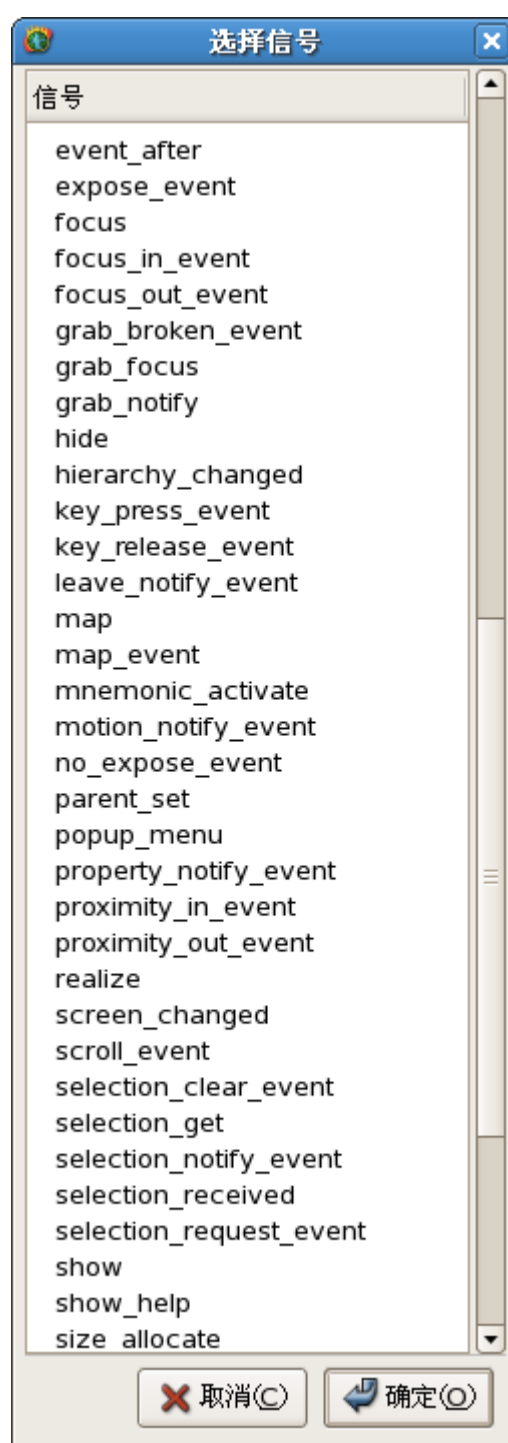


图 13: 信号列表 2

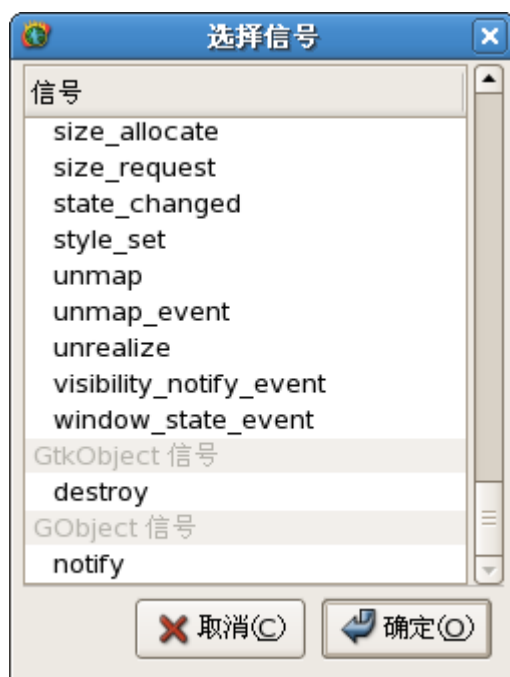


图 14: 信号列表 3

六、工程文件选项设定

当你复制来此工程文件时，不一定复制了相应的工程环境，你需要在此设定，点 [选项] 按钮后如图 6。



图 6: Glade 工程文件选项设定

七、新建一个文件夹

在工程文件的同一级文件目录下，你需要新建一个文件夹，本例中是 mindway。（如图 7 所示，点一下 [创建文件夹] 按钮就可以了）。

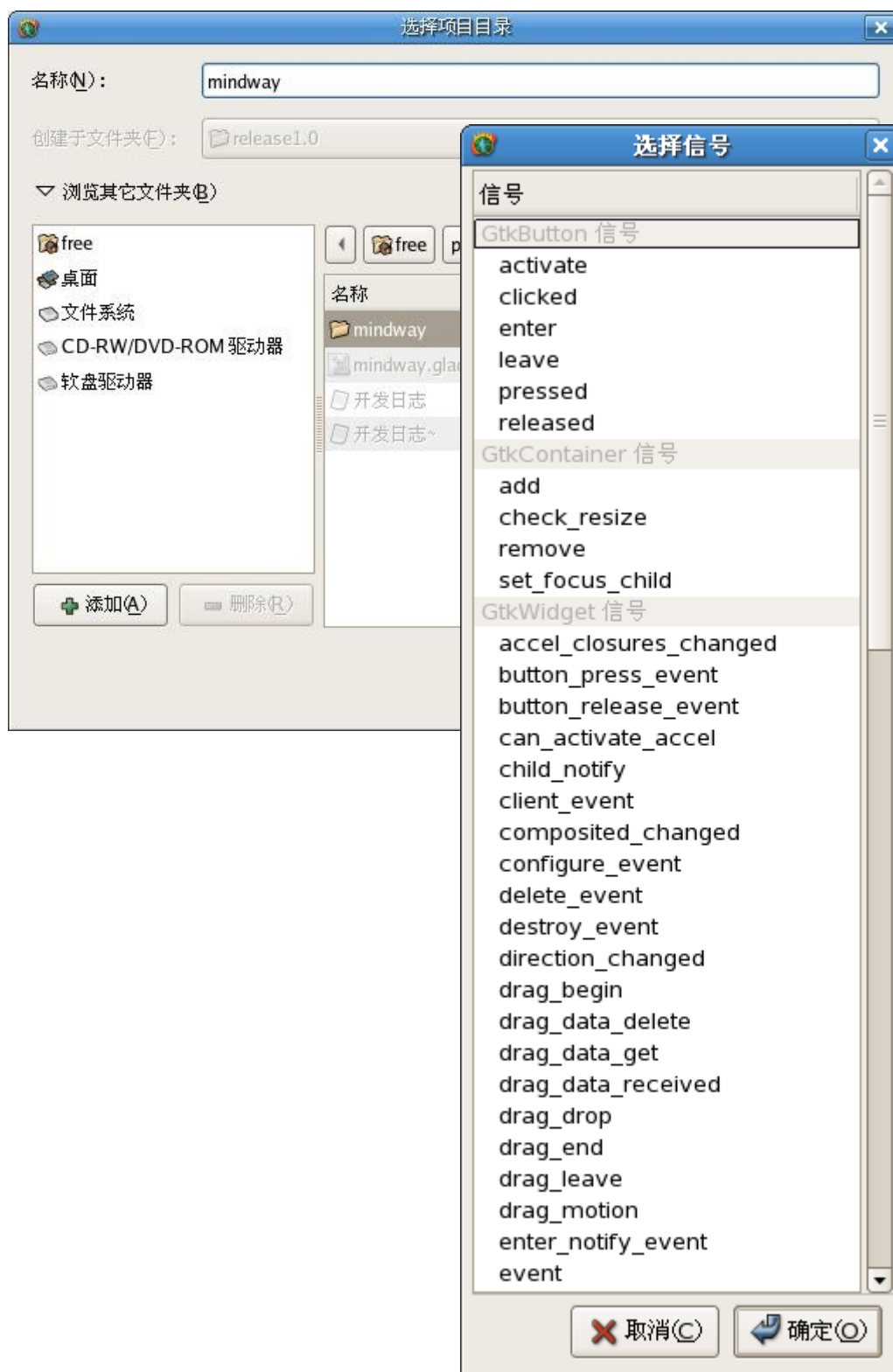


图 7: 工程目录设定

八、改变文件选项

你可以按图 7 所示，改变其中的参数

九、保存工程文件

点一下主窗口中的 [保存] 按钮，保存工程文件

十、代码联编

点一下主窗口中的 [联编] 按钮生成 C 语言代码

十一、新建 2 个终端窗口

点桌面上的 [应用程序] -> [附件] -> [终端]
新建 2 个窗口是便于调试，在编程时很方便高效。

十二、生成编译环境

在终端中进入工程文件夹，本例为 mindway；
输入 `./autogen.sh`

十三、编译

输入 `make`

十四、运行

在另一个终端窗口进入工程文件夹的下一级目录 src
输入 `./mindway`

十五、小结

运行界面如图 8 所示，我们可以看到，通过这十四步，我们的目标界面已经生成，到此时为止，我们还没有用上程序 C 语言的有关知识，所以用 Glade 构造图形界面是不需要编程知识的体力活，也真切反映了 Glade 项目界面和功能完全分开的设计思想。

图 8 界面生成后可具备以下功能：接受各部件的反馈信号，并调用相关的子程序；接受快捷键信号；显示工具提示（即鼠标在相关部件上稍作停留，即显示相关的提示信息）。

生成界面语言可以是 C，也可以是 C++，据说与 Python 语言的绑定（pygtk）更易用，这些就是待学习的内容了。不过，Python（我个人把它译为“派送”语言）锐劲十足，学习十分容易却是真的，google 宣称，它和 C 是 goog

1e 编程的主要语言。

闲话少叙，到此为止，界面虽然生成，但只是一个空壳，就连退出这样一个简单的功能，我们也必须用点右上角的[x]来实现。所以还需要编程能力作为后盾来进行完善，见下面第参章。

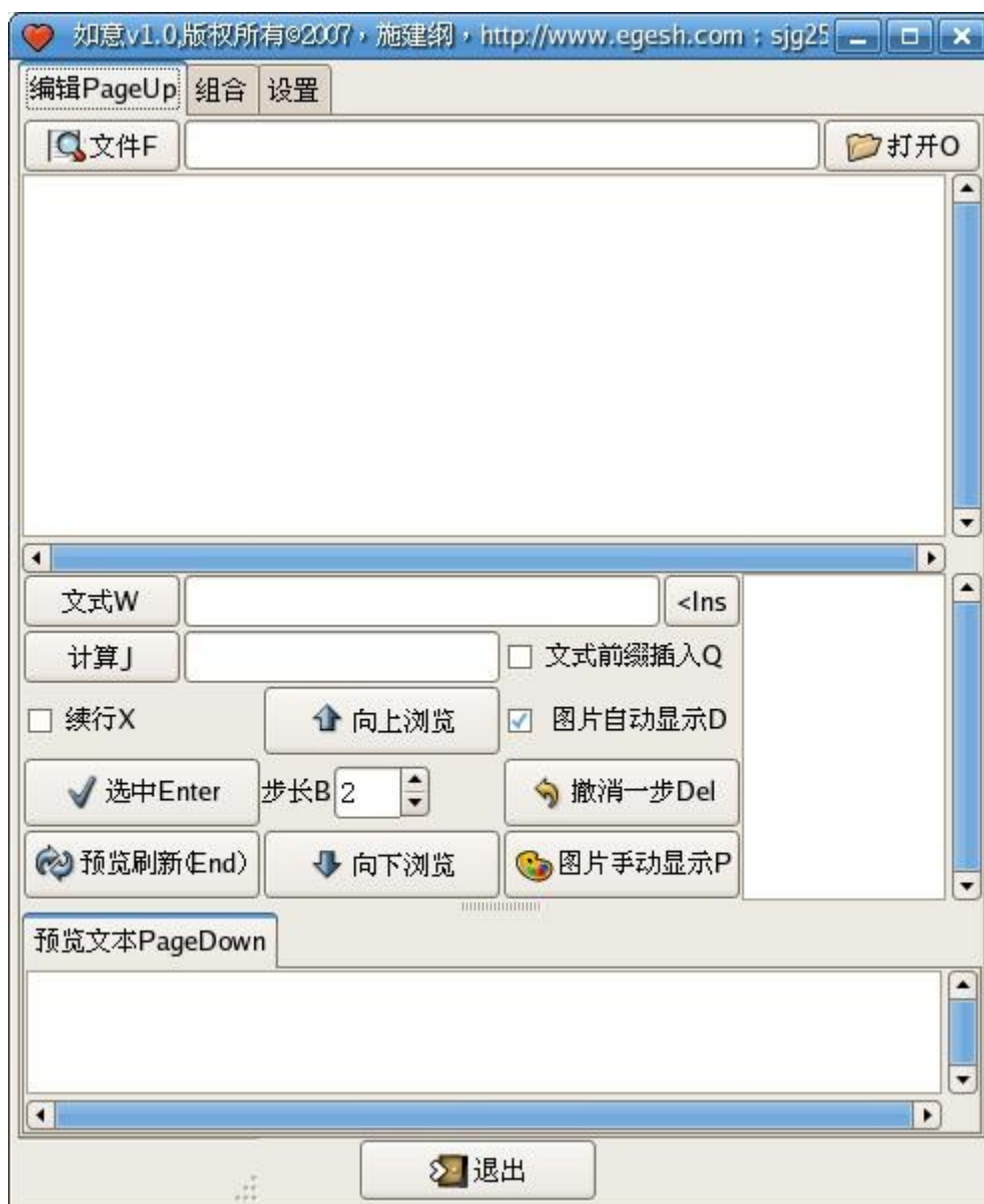


图 8: Glade 自动生成的如意界面

下面是通过功能完善后“如意”软件界面：



图9 功能完善后的如意软件界面



参、新建 Glade 工程

一、运行 Glade

点桌面上的 [应用程序] -> [编程] -> [Glade Interface Designer]

运行后得到如封面图片所示的界面。

二、新建一个 Glade 工程

点主窗口的 [新建] 按钮出现图 10 界面。




图 10、新建 Glade 工程文件界面

三、选择工程类型

点 [New GTK+ Project] ,
GTK+的好处是跨平台；如果你纯粹是 Linux 下的编程，且喜欢 GNOME 桌面，
则选 [New GNOME Project] 更方便。

四、点 [调色板] 窗口中的 [窗口] 按钮

点  可以新建一个窗口。

你也可以点 [对话框] 来创建一个对话框，它们都是最底层的容器，只是窗口的功能是最强的。

在 GTK+ 的编程中，要注意一个“容器”的概念，窗口、对话框、水平框、垂直框、表、帧、水平窗格（分栏）、垂直窗格、笔记簿等是容器，而其它的则是部件，部件可放入容器中，但容器不能放入部件中，容器可以放入容器中，界面的布局就是由容器套容器构成的。

五、出现一个界面窗口 window1



图 11: 加入了水平框和按钮部件后的 window1 窗口

如图 11。

六、将其它容器或部件放入其中

点 [调色板] 窗口的 [水平框] 容器，点已经生成的 window1 窗口；
点 [调色板] 窗口的 [按钮] 部件，放入各水平框容器。

七、编辑属性以适合特殊需要

1、构件属性

更改它的名字，图标、默认值、标签等参数

2、打包属性

更改它的扩展、位置属性

3、通用属性

更改它的宽度、高度、可见、不可见、工具提示信息、快捷键等属性

4、信号

更改触发信号及调用子程的名字

八、其余

其余的过程同打开一个已有 Glade 工程文件相同，见第壹节第五点及后面内容。



肆、将功能代码 加入联编 代码中

Glade 通过自动生成 main.c,interface.c,callback.c,support.c, interface.h,callback.h,support.h 等几个文件及相应的 Makefile 文件来实现图形界面的功能,这几个文件中, callback.c,callback.h, main.c 以追加方式改变,其它文件则以覆盖方式改变。

因此,当你需要进行重新联编时,你应**十分注意**,以避免覆盖了你的劳动成果。其中 interface.c 是十分需要引起注意的一个文件,下面两种方法可供参考:

- 一、用 gedit 或其它编辑器打开 interface.c,
- 二、复制一个 interface.c 的备份

对于 Glade 自动生成的文件,进行功能上的加工和配合后,就能成为完整个性化的程序,用 Glade 编程就是这么简单。

完成功能和配合,您需要作两部分的工作:

- 一、加入宏定义、头文件和外部变量声明
- 二、加入各种功能模块

一、加入宏定义、头文件和外部变量声明

1、加入外部变量声明

外部变量的声明对于多文件编译十分重要,它使变量的值穿过不同且独立的 C 文件而保持不变。

比如,下面的几个 C 文件中,怎样使界面显示程序 interface.c 中文件名变量 *filename 传到 callback.c 中呢?

通过全局变量和外部变量的配合就是一种办法。

你需要在 man.c 中声明一个全局变量

```
char *filename;
```

再在 callback.c 中声明一个外部变量,即说明这个变量是从外部文件 main.c 或 interface.c 传过来的,外部变量声明的关键字为 extern,所以声明格式如下:

```
extern char *filename;
```

如果你比较一下下面的 callback.c 程序,会发现,它的声明为

```
extern gchar *filename;
```

多了一个 g 字,这是 GTK+ 库和它的底层库 Glib 库定义的一个类型,方便于不同类型机器硬件、不同操作系统间移植而设,因此用 gchar 更好一点,当然,用 char 类型也没有错,但移植时就可能有错。

Glib 定义的类型有 gchar ,gint,gdouble,gboolean,使用它有更大的优越性。

Filename 变量需要从 interface.c 传递数值，所以在 interface.c 中也要同样声明一下外部变量。

阅读一下相关的原程序变量声明，附录的原程序中用//注释符提示了修改后的程序与自动联编生成的程序的不同，可作参考。

掌握了全局变量和外部变量的配合，就能灵活掌握 Glade 的强大功能的一半，使图形界面编程与非图形界面一样容易。

2、加入宏定义

宏定义使程序的修改变得十分容易，这不是 GTK+ 的特殊之处，可参考一下 C 语言的书籍。

3、加入头文件

头文件也不是 GTK+ 的特殊处，在你用到相关的库程序时，就调入相关的头文件。

二、加入各种功能模块

功能模块的加入是个性化的东西，但从 GTK+ 程序的角度看也有共性的东西，如正常退出，文件查找，文件名保存，树视图显示，文本视图显示、输入框内容的读入、输出，步长按钮的读入，图片窗口的生成等等，算是一个 GTK+ 界面的介绍吧。

1、加入退出 模块

退出模块是标准化的，你只要抄下来就可以了。
它就是

```
gtk_main_quit();
```

2、加入文件 查找模块

这是 Glade 没有的部分，包括下面内容，值得注意。

```
GtkWidget *open_dialog;//声明这是一个 GTK 部件
open_dialog = gtk_file_selection_new("请选择一个文件:");//打开一个现成的文件查找对话框
g_signal_connect(
    GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->ok_button),"clicked",G_CALLBACK(store_
filename),open_dialog);//如果对话框点中 OK 按钮，就调用 store_filename 子程，保存文件名
g_signal_connect_swapped(
    GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->ok_button),"clicked",G_CALLBACK
(gtk_widget_destroy),open_dialog);//保存文件名后进行第二个反馈操作——退出对话框
g_signal_connect_swapped (GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->cancel_button),
    "clicked",G_CALLBACK(gtk_widget_destroy),(gpointer) open_dialog);//如果按中 Ca
ncel,就销毁对话框退出
gtk_widget_show(open_dialog);//显示对话框
```

3、加入文件 名保存模 块

```
void store_filename(GtkFileSelection *selector,gpointer user_data)
{
    filename = gtk_file_selection_get_filename(GTK_FILE_SELECTION(user_data));//将文件名赋给外部全局变量 filename
    gtk_entry_set_text (GTK_ENTRY(filename),filename);//在 filename 输入框中显示文件名
}
```

4、加入 树视图控 件的显示 控制模块

```
GtkListStore *create_liststorel( ) //显示数据组织程序
{
    GtkTreeIter    iter; //树视图位置变量
    GdkPixbuf      *icon;//图标变量
    GError          *error = NULL;//错误变量
    FILE *fpl;//文件指针
    gchar name[MAXLINE],*item[3];//字符串变量、数组
    gint nt,i,i1,i2;//整数变量
    spechar = gtk_entry_get_text (GTK_ENTRY(specentry));//从输入框 specentry 读入文件名字串
    if(spechar == NULL) spechar="spechar.txt";//如果空就赋值为"spechar.txt"
    fpl = fopen (spechar,"r");//打开文件
    if(fpl){//打开文件成功判断
        storel = gtk_list_store_new(3, G_TYPE_STRING, GDK_TYPE_PIXBUF,G_TYPE_STRING);//创建表数据格式
        nt = LineCntr(fpl);//计数文件共多少行
        for(i=0;i< nt;i++){
            fgets(name,MAXLINE,fpl);//读入文件的每一行赋给字符串指针 name
            for(i2=0;i2<3;i2++){
                item[i2]=nthList (name,i2,MYSPLIT);//将每行分解成三项，赋给字符串数组 item
                if(strlen(item[2])!= 0) icon = gdk_pixbuf_new_from_file(item[2], &error);//如果最后一项不空，就把它作为图标文件名
                else {
                    icon = gdk_pixbuf_new_from_file("icon.png", &error);//最后一项空就用默认图标
                }
                if (error) { //图标文件调入出错，就打印出错信息
                    g_warning ("Could not load icon: %s\n", error->message);
                    g_error_free(error);
                    error = NULL;
                }
                gtk_list_store_append(storel, &iter);//追加一行
                gtk_list_store_set(storel, &iter, //设置追加内容
                    COL_TEXT1, item[0],
                    COL_ICON1, icon,
                    COL_TEXT2, item[1],
                    -1);
            }
        }
        if(fpl)fclose(fpl);
    }
    return storel;//反馈树视图显示数据
}

GtkWidget *create_treeviewl(GtkTreeModel *model)//显示树视图的数据
{
    GtkTreeViewColumn *col;//树视图列变量
    GtkCellRenderer *renderer;//单元格变量
    GtkWidget *view;//树视图变量
    view = gtk_tree_view_new_with_model(model);//新建树视图
```

```

col = gtk_tree_view_column_new();//新建列
gtk_tree_view_column_set_title(col, "内容");//设置列标题

renderer = gtk_cell_renderer_text_new();//非标题行显示文本内容
gtk_tree_view_column_pack_start(col, renderer, TRUE);
gtk_tree_view_column_set_attributes(col, renderer,
                                     "text", COL_TEXT1,
                                     NULL);

renderer = gtk_cell_renderer_pixbuf_new();//文本内容后再显示一个图标
gtk_tree_view_column_pack_start(col, renderer, FALSE);
gtk_tree_view_column_set_attributes(col, renderer,
                                     "pixbuf", COL_ICON1,
                                     NULL);

gtk_tree_view_append_column(GTK_TREE_VIEW(view), col);//第1列显示完成

/* 2nd column */

col = gtk_tree_view_column_new();//新建列
gtk_tree_view_column_set_title(col, "备注");//设置列标题

renderer = gtk_cell_renderer_text_new();//非标题行显示文本内容
gtk_tree_view_column_pack_start(col, renderer, TRUE);
gtk_tree_view_column_set_attributes(col, renderer,
                                     "text", COL_TEXT2,
                                     NULL);

gtk_tree_view_append_column(GTK_TREE_VIEW(view), col);//第2列显示完成
return view;
}

```

5、加入 文本视图 控件的显示控制模块

```

void textview_display_file (GtkWidget *textview ,gchar *filename)//在文本视图内显示一个文件
{
    GtkTextBuffer *buffer;//文本视图缓冲区变量
    GtkTextIter start,end;//文本视图位置变量
    FILE *fp4;//文件指针
    gchar name[MAXLINE];//字符数组
    fp4 = fopen (filename,"r");    //打开文件
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview));//获取文本视图缓冲区
    gtk_text_buffer_get_start_iter (buffer,&start);//获取文本视图起始位置
    gtk_text_buffer_get_end_iter (buffer,&end);//获取文本视图终点位置
    gtk_text_buffer_delete (buffer,&start,&end);//将原内容删除
    gtk_text_buffer_get_end_iter(buffer,&end);//获取文本视图删除终点位置
    if(fp4){
        while(fgets(name,MAXLINE,fp4)){//如果读取文件还有内容
            gtk_text_buffer_insert(buffer,&end,name,-1);//就将它显示出来
        }
    }
    if(fp4)fclose(fp4);//关闭文件
}

```

6、显示 图片

```

void dispimage()

```

```

{
    GtkWidget * windowimg;
    if(strlen(itemtnxt) != 0){
        windowimg=imagewin(itemtnxt);
        gtk_widget_show(windowimg);
    }
}
GtkWidget *imagewin( gchar *filename)
{
    GtkWidget * windowimg;
    GtkWidget *image;
    windowimg = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title(GTK_WINDOW(windowimg),filename);
    gtk_container_set_border_width(GTK_CONTAINER(windowimg),10);
    image=gtk_image_new_from_file(filename);
    gtk_container_add(GTK_CONTAINER(windowimg),image);
    gtk_widget_show_all(windowimg);
    return windowimg;
}

```

7、加入字符串的计数、分解、转换、连接、计算功能

这部分内容详见附录源程序。



伍、修改界面布局

修改界面布局可以通过点中相关的容器，然后用剪切（**Ctrl+X**）放入剪贴板。

通过删除旧布局，新建新布局后，再粘贴（**Ctrl+V**）入新的容器即可完成。

后面的处理同第貳节第九点之后的内容。



陆、加入过程 对话 GUI

我们有时候会碰到这样的情况，后面的内容依赖于前面的数据，这种有明确时间顺序的对话窗口或对话框我把叫作过程对话 GUI。

摸索后介绍下面一种方法：

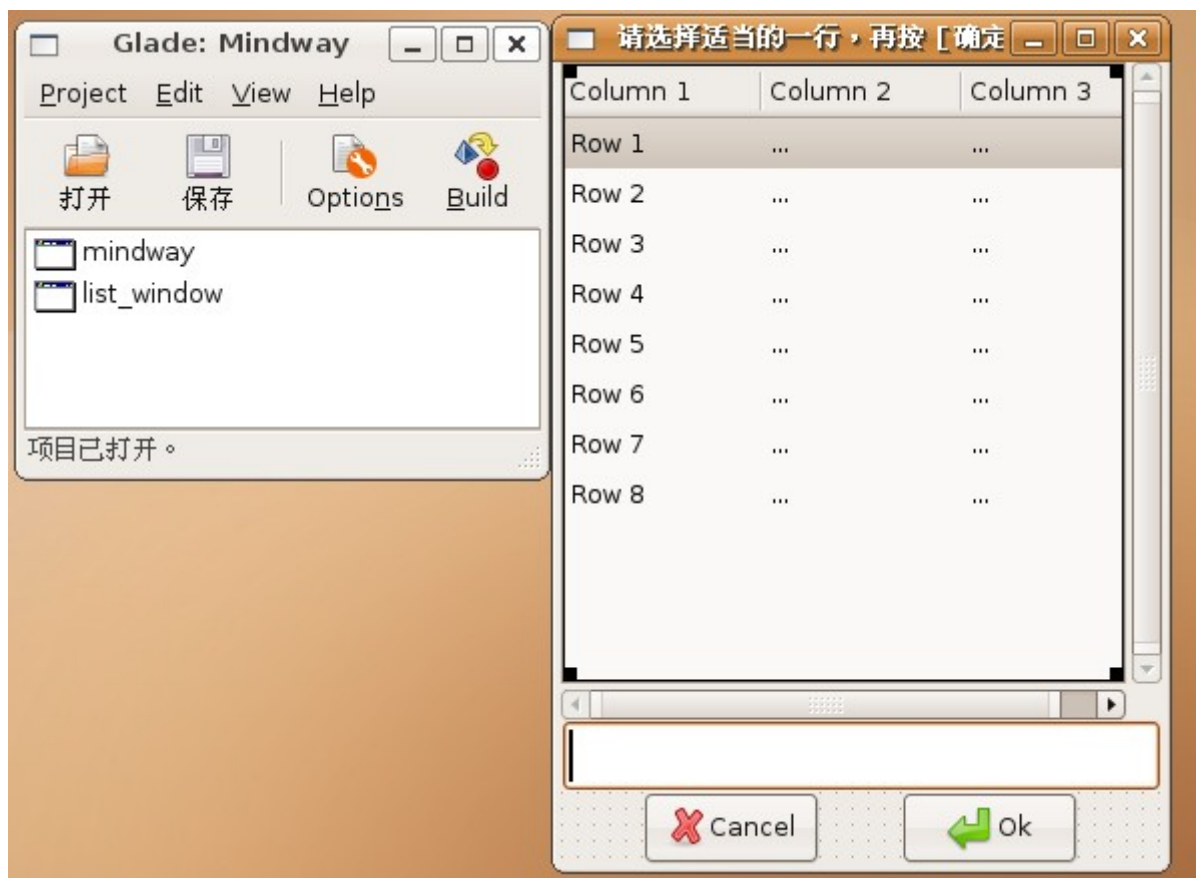


插图 12: 建立过程对话框

一、用 GLADE 建立一个窗口

建立这样一个窗口，如图所示，建立过程参前文；

二、修改 main.c

增加全局变量，本例如下：

```
GtkWidget *list_treeview;
```

```
GtkListStore *list_store;
```



```
GtkTreeModel *list_model;

GtkWidget *scrolledwindow2;

GtkWidget *list_entry;

gchar *step_list[2];
```

三、修改 interface.c

1、对应全局 变量进行 替换设置

本例中找到窗口模块:

```
GtkWidget*

create_list_window (void)

{

    extern GtkWidget *list_treeview;

    extern GtkTreeModel *list_model;

    extern GtkListStore *list_store;

    extern *list_entry;

    GtkWidget *list_window;

    GtkWidget *vbox6;

    GtkWidget *vbox7;

    GtkWidget *scrolledwindow7;

    // GtkWidget *list_treeview;

    // GtkWidget *list_entry;

    .....
```

```
// list_treeview = gtk_tree_view_new ();

list_treeview = create_treeview1(list_model);

.....
```

红色部分为增加或注释掉的内容。

2、在事件反馈部分增加删除过程 GUI 的语句

```
g_signal_connect ((gpointer) list_cancel_button, "clicked",

                  G_CALLBACK (on_list_cancel_button_clicked),

                  NULL);

g_signal_connect_swapped (G_OBJECT( list_cancel_button), "clicked",

                          G_CALLBACK (gtk_widget_destroy),

                          list_window);

g_signal_connect ((gpointer) list_ok_button, "clicked",

                  G_CALLBACK (on_list_ok_button_clicked),

                  NULL);

g_signal_connect_swapped (G_OBJECT(list_ok_button), "clicked",

                          G_CALLBACK (gtk_widget_destroy),

                          list_window);
```

四、在 callback.c 中加入调用初始化和循环语句

```
list_model = GTK_TREE_MODEL(list_store);

gtk_tree_view_set_model(GTK_TREE_VIEW(list_treeview),list_model);

list_dialog=create_list_window ();

gtk_widget_show (list_dialog);
```

```
gtk_main ();
```

上面一段是程序初始化，如果注意一下，是跟 mian.c 实现的做法是一样的，也就是说，过程窗口实现可以象主程一样地去实现。

注意退出窗口的操作语句也是一样的：

```
void  
  
on_list_cancel_button_clicked      (GtkButton      *button,  
  
                                     gpointer         user_data)  
  
{  
  
    step_list[0]="";//附加的反馈判断语句  
  
    step_list[1]="";  
  
    gtk_main_quit();  
  
}
```

五、小结

认识到可以用主程一样的方法和语句去实现过程 GUI，因为界面而增改的语句就显得不多了。

当然，这是 C 语言的实现，也许 GTK+2 与 python 的绑定——PYGTK 更加让人感到这种简洁的出乎意料，这种只要 glade 生成界面，用简单的 python 语句就能实现的 GUI 编程，受到了越来越多人的推荐，除了比小弟开发的“如意”稍微麻烦一点外，好象还没有看到可匹敌的简洁，有机会大家学习后再交流。



柒、制作 deb 包

deb 包是 Debian/Ubuntu 的安装或管理包格式，相比于 tar 包，大同小异。相信打 tar.gz（简称 tgz）包是每一个会使用 Linux 的人都会的，因为在笔者见到的发行版中，图形界面下都是点中->右键->[创建归档文件]即可完成。

deb 包的制作对于一个新手来说也差不多如此简单，笔者从软件 mydeb 的打包过程来介绍操作步骤：

- 1、确定软件名
- 2、新建打包目录
- 3、新建信息目录
- 4、新建安装目录
- 5、准备文件
- 6、生成 SHELL(可省)
- 7、打包

1、确定软件名

英文名：

mindway

中文名：

如意

2、新建打包目录

你准备打包时，请新建一个目录， 本例打包目录：

```
mindwaydeb
```

有三种方法可完成此目标，图形界面最直观，终端命令最直接，SHELL 最省力：

1) 你可以在终端下用命令：

```
mkdir -m 755 (打包目录)
```

```
cd (打包目录)
```

2) ubuntu GUI 下

菜单 [位置] -> [主文件夹] ->右键-> [创建文件夹] ->mindwaydir

3) 使用 SHELL 文本执行

这是最省力的，将一系列命令写入一个文本文件，在第一行加入

```
#!/bin/bash
```

然后保存就可以了。

3、新建信息目录

信息目录 DEBIAN/是 deb 包的必需项，而且目录下必须有 control 文件

```
mkdir DEBIAN
```

4、新建安装目录

安装目录与系统目录一一对应，比如通常我们的运行文件会安装在系统目录的

```
/usr/bin
```

那么在打包目录 mindwaydir 下需对应建立一个目录

usr/bin

它们的名字一样，只是相对位置不一样，系统目录相对于根目录/，

安装目录是相对于打包目录

实现的方式如第 1 步所述

最简单的有用二进制包有 usr/bin 目录就够了，

没有安装目录也可生成 deb 包。

如果需要菜单、图标等则需要更多的目录，本例需要下面几个目录

```
mkdir usr/bin usr/share/(英文名) usr/share/pixmaps usr/share/applications
```

您 可以利用“如意”软件的摘录功能进行选择，点此链接继续，

5、准备文件

您可以将所有需要的文件放于打包目录下，再生成 SHELL 轻松完成。

1) 信息文件 *control*

用如意软件生成 control 文件，点此链接行，再按 [向下浏览] 按钮继续，

control 位于 DEBIAN 目录下，这个文件是必须的，

包管理软件依据该文件来操作，

或者按下面内容修改：

Package: mindway_3.07.zh-cn

Section: main

Version: 3.07

Priority: optional

Maintainer: ShiJiangang <sjg255@shl63d.sta.net.cn>

Architecture: i386

Description: Powerful platform of browsing , extracting , calculating , editing,

It's also a standard examination tool, and provide some interesting games.

2) 项目文件

因各项目而异，本例包括：

(1) 运行文件

usr/bin/mindway

(2) 运行图标文件

usr/share/mindway/click.png icon.png ruyix32.png

(3) 文档文件

usr/share/mindway/doc/readme.xdf 等

(4) 菜单文件

usr/share/applications/mindway.desktop

(5) 菜单图标文件

usr/share/pixmaps/mindway.png

(6) 版权文件

DEBIAN/copyright

(7) 数字印纹文件

DEBIAN/md5sums

可用下面命令实现:

```
find usr -type f -exec md5sum {} \;>>DEBIAN/md5sums
```

6、打包

1) 安装 *fakeroot*

这个命令可以使得软件包的文件在安装后的所有者成为 root(如果你用普通用户来做这项工作的话)

```
sudo apt-get install fakeroot
```

2) 打包 命令

```
fakeroot dpkg-deb -b mindwaydir mindway_3.07_zh_cn.deb
```

3) 测试 安装

用文件浏览器找到 mindway_3.07_zh_cn.deb 文件

->右键->用[GDebi 软件包安装程序]打开。

4) 测试删除

[系统] -> [系统管理] -> [新立得软件包管理器] ->找到 mindway

->右键-> [标记以便彻底删除] -> [应用]

5) 错误排除

如果出现包管错误:

软件包 ***软件 需要重新安装,但是我无法找到相应的安装文件。

打开高速缓存(1)时有内部错误。请报告。

解决如下

```
$ sudo gedit /var/lib/dpkg/status
```

找到相应软件（如 mindway）安装段，删除后事保存退出即可

7、简化的方法

用如意的相关向导生成 SHELL 文件内容是个更方便的方法，如意软件包就是用这种方法发布的，默认的安装包里已包含了“新手制作 deb 包”向导，按它的提示操作就可以了。



八、附录

附录源程序包括 main.c, interface.c 与自动生成不的部分, callback.h, callback.c, 代码中用红色标示了不是自动生成的部分，callback.c 中除程序框架外都是编写的，故不再用红色标示。

有了这些文件，你可以生成整个项目，并且编译成功了，但要运行，你还需要一些图标文件，最新的完整代码下载在 <http://www.qvandi.com>

1、main.c

```
/*
 * Initial main.c file generated by Glade. Edit as required.
 * Glade will not overwrite this file.
 */

#ifdef HAVE_CONFIG_H
# include <config.h>
#endif

#include <gtk/gtk.h>

#include "interface.h"
#include "support.h"
//增加部分
#define MAXLINE 1000
GtkWidget *jgentry;
GtkWidget *filename,*tentry;
GtkWidget *wsentry;
gchar *filname = NULL,*tfilname = NULL;
```

```

    GtkWidget *treeview,*treeview1;
    GtkWidget *textview1,textview2;
    GtkListStore *store,*store1;
    GtkTreeModel *model,*modell;
    GtkWidget *scrolledwindow2;
    gint cnt,textonly=1,doscheck=0,precheck=0,piccheck=1;
    GtkWidget *hentry;
    gchar *linestr[MAXLINE],itemtnxt;
    gchar *first1[MAXLINE];
    guint nhang = 0,stepn=1,stepb=0,stepf=0;xhbutton=0,linesum;
    gchar *spechar;
    GtkWidget *specentry,*schentry;
    GtkWidget *stepspinbutton,*digspinbutton::

//
int
main (int argc, char *argv[])
{
    GtkWidget *mindway;

#ifdef ENABLE_NLS
    bindtextdomain (GETTEXT_PACKAGE, PACKAGE_LOCALE_DIR);
    bind_textdomain_codeset (GETTEXT_PACKAGE, "UTF-8");
    textdomain (GETTEXT_PACKAGE);
#endif

    gtk_set_locale ();
    gtk_init (&argc, &argv);

    add_pixmap_directory (PACKAGE_DATA_DIR "/" PACKAGE "/pixmap");

    /*
     * The following code was added by Glade to create one of each component
     * (except popup menus), just so that you see something after building
     * the project. Delete any components that you don't want shown initially.
     */
    mindway = create_mindway ();
    gtk_widget_show (mindway);

    gtk_main ();
    return 0;
}

```

2、interface.c

```

/*
 * DO NOT EDIT THIS FILE - it is generated by Glade.
 */

#ifdef HAVE_CONFIG_H
# include <config.h>
#endif

#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <string.h>
#include <stdio.h>

```

```

#include <gdk/gdkkeysyms.h>
#include <gtk/gtk.h>

#include "callbacks.h"
#include "interface.h"
#include "support.h"

#define GLADE_HOOKUP_OBJECT(component,widget,name) \
  g_object_set_data_full (G_OBJECT (component), name, \
    gtk_widget_ref (widget), (GDestroyNotify) gtk_widget_unref)

#define GLADE_HOOKUP_OBJECT_NO_REF(component,widget,name) \
  g_object_set_data (G_OBJECT (component), name, widget)

GtkWidget*
create_mindway (void)
{
//增加部分
  extern GtkWidget *jgentry;
  extern GtkWidget *filename,*tentry;
  extern GtkWidget *wsentry;
  extern GtkWidget *treeview,*treeview1;
  extern      GtkTreeModel      *model,*modell;
  extern GtkListStore *store,*storel;
  extern GtkWidget *scrolledwindow2;
  extern gchar *filename,*tfilename;
  extern GtkWidget *stepspinbutton,*digspinbutton;;
  extern GtkWidget *textview1,*textview2;
  extern GtkWidget *specentry;
//
  GtkWidget *mindway;
  GtkWidget *vbox4;
  GtkWidget *notebook2;
  GtkWidget *vpaned1;
  GtkWidget *vbox1;
  GtkWidget *hbox1;
  GtkWidget *file_button;
  GtkWidget *alignment12;
  GtkWidget *hbox38;
  GtkWidget *image12;
  GtkWidget *label18;
//  GtkWidget *filename;
  GtkWidget *hbox28;
  GtkWidget *opbutton;
  GtkWidget *alignment13;
  GtkWidget *hbox39;
  GtkWidget *image13;
  GtkWidget *label19;
//  GtkWidget *scrolledwindow2;
//  GtkWidget *treeview;
  GtkWidget *hbox24;
  GtkWidget *vbox2;
  GtkWidget *hbox25;
  GtkWidget *wsbutton;
  GtkWidget *hbox29;
//  GtkWidget *wsentry;
  GtkWidget *twbutton;
  GtkWidget *hbox26;

```

```
GtkWidget *jsbutton;
// GtkWidget *jentry;
GtkWidget *precheckbutton;
GtkWidget *hbox57;
GtkWidget *xhcheckbutton;
GtkWidget *xsbutton;
GtkWidget *alignment8;
GtkWidget *hbox21;
GtkWidget *image8;
GtkWidget *label10;
GtkWidget *piccheckbutton;
GtkWidget *hbox58;
GtkWidget *xzbutton;
GtkWidget *alignment3;
GtkWidget *hbox7;
GtkWidget *image3;
GtkWidget *label13;
GtkWidget *hbox10;
GtkWidget *steplabel;
GtkObject *stepspinbutton_adj;
// GtkWidget *stepspinbutton;
GtkWidget *cancelbutton1;
GtkWidget *alignment5;
GtkWidget *hbox17;
GtkWidget *image5;
GtkWidget *label17;
GtkWidget *hbox59;
GtkWidget *zhbutton;
GtkWidget *alignment10;
GtkWidget *hbox23;
GtkWidget *image10;
GtkWidget *label12;
GtkWidget *xxbutton;
GtkWidget *alignment9;
GtkWidget *hbox22;
GtkWidget *image9;
GtkWidget *label11;
GtkWidget *imagebutton;
GtkWidget *alignment11;
GtkWidget *hbox31;
GtkWidget *image11;
GtkWidget *label17;
GtkWidget *scrolledwindow4;
// GtkWidget *treeview1;
GtkWidget *notebook1;
GtkWidget *scrolledwindow3;
// GtkWidget *textview1;
GtkWidget *label4;
GtkWidget *label14;
GtkWidget *vbox3;
GtkWidget *hbox30;
GtkWidget *tbutton;
GtkWidget *alignment14;
GtkWidget *hbox40;
GtkWidget *image14;
GtkWidget *label20;
// GtkWidget *tentry;
GtkWidget *topenbutton;
GtkWidget *alignment15;
```

```
GtkWidget *hbox41;
GtkWidget *image15;
GtkWidget *label121;
GtkWidget *hbox36;
GtkWidget *newbutton;
GtkWidget *alignment20;
GtkWidget *hbox46;
GtkWidget *image20;
GtkWidget *label126;
GtkWidget *bakbutton;
GtkWidget *alignment25;
GtkWidget *hbox51;
GtkWidget *image25;
GtkWidget *label131;
GtkWidget *resbutton;
GtkWidget *alignment24;
GtkWidget *hbox50;
GtkWidget *image24;
GtkWidget *label130;
GtkWidget *savbutton;
GtkWidget *alignment22;
GtkWidget *hbox48;
GtkWidget *image22;
GtkWidget *label128;
GtkWidget *hbox37;
GtkWidget *assbutton;
GtkWidget *alignment19;
GtkWidget *hbox45;
GtkWidget *image19;
GtkWidget *label125;
GtkWidget *drbutton;
GtkWidget *alignment18;
GtkWidget *hbox44;
GtkWidget *image18;
GtkWidget *label124;
GtkWidget *dnbutton;
GtkWidget *alignment17;
GtkWidget *hbox43;
GtkWidget *image17;
GtkWidget *label123;
GtkWidget *xbutton;
GtkWidget *hbox55;
GtkWidget *fixed2;
GtkWidget *fixed3;
GtkWidget *fixed4;
GtkWidget *gjbutton;
GtkWidget *scrolledwindow5;
// GtkWidget *textview2;
GtkWidget *label115;
GtkWidget *vbox5;
GtkWidget *hbox53;
GtkWidget *specbutton;
// GtkWidget *specentry;
GtkWidget *opsbutton;
GtkWidget *alignment26;
GtkWidget *hbox54;
GtkWidget *image26;
GtkWidget *label132;
GtkWidget *hbox56;
```

on_precheckboxbutton_toggled	(GtkToggleButton *togglebutton, gpointer user_data);
void on_xhcheckboxbutton_toggled	(GtkToggleButton *togglebutton, gpointer user_data);
void on_xsbutton_clicked	(GtkButton *button, gpointer user_data);
void on_piccheckboxbutton_toggled	(GtkToggleButton *togglebutton, gpointer user_data);
void on_xzbutton_clicked	(GtkButton *button, gpointer user_data);
void on_cancelbutton1_clicked	(GtkButton *button, gpointer user_data);
void on_zhbutton_clicked	(GtkButton *button, gpointer user_data);
void on_xxbutton_clicked	(GtkButton *button, gpointer user_data);
void on_imagebutton_clicked	(GtkButton *button, gpointer user_data);
void on_treeview1_row_activated	(GtkTreeView *treeview, GtkTreePath *path, GtkTreeViewColumn *column, gpointer user_data);
void on_tfilebutton_clicked	(GtkButton *button, gpointer user_data);
void on_topenbutton_clicked	(GtkButton *button, gpointer user_data);
void on_newbutton_activate	(GtkButton *button, gpointer user_data);
void on_bakbutton_clicked	(GtkButton *button, gpointer user_data);
void on_resbutton_clicked	(GtkButton *button,

```

                                gpointer      user_data);

void
on_savbutton_clicked           (GtkButton    *button,
                                gpointer      user_data);

void
on_assbutton_clicked           (GtkButton    *button,
                                gpointer      user_data);

void
on_drbutton_clicked            (GtkButton    *button,
                                gpointer      user_data);

void
on_dnbutton_clicked            (GtkButton    *button,
                                gpointer      user_data);

void
on_xdbutton_clicked            (GtkButton    *button,
                                gpointer      user_data);

void
on_gjbutton_clicked            (GtkButton    *button,
                                gpointer      user_data);

void
on_textview2_move_cursor      (GtkTextView *textview,
                                GtkMovementStep step,
                                gint          count,
                                gboolean      extend_selection,
                                gpointer      user_data);

void
on_specbutton_clicked          (GtkButton    *button,
                                gpointer      user_data);

void
on_opsbutton_clicked           (GtkButton    *button,
                                gpointer      user_data);

void
on_okbutton1_clicked           (GtkButton    *button,
                                gpointer      user_data);

void
on_piccheckbutton_toggled      (GtkToggleButton *togglebutton,
                                gpointer      user_data);

//增加部分
void xzwork();
void xzlwork();
void zh ();
GtkWidget *imagewin( gchar *filename);
void dispimage();
void store_sfilename(GtkFileSelection *selector,gpointer user_data);
void textview_display_file (GtkWidget *textview ,gchar *filename);
gchar * nth (FILE *fp , int line , int column , int id);
char * substr (char * str,int nb,int ne);
int subchar (char * str,int nb);

```

```

char *strcacu ( char *str );
gint found (char *symb,char *lstr);
void *arrangstr (char *lstr,char *symb);
char *strlink(char * str1,char *str2,int max);
gboolean numberp(char c);
gchar * altstr(gchar *str1);
int LineCntr(FILE *fp);
int ItemCntr(FILE *fp,int LineNo);
int WrnCntr(FILE *fp, int line);
char * WordBody(char *str) ;
gchar *LastItem (FILE *fp,gint LineNo);
gboolean Listp (gchar *str);
gchar * nthList (gchar *str , int column , gchar ch);
static void store_filename(GtkFileSelection *selector,gpointer user_data);
static void store_tfilename(GtkFileSelection *selector,gpointer user_data);
GtkWidget * create_treeview(GtkTreeModel *model);
GtkListStore *create_liststore(gint nhang);
GtkWidget * create_treeviewl(GtkTreeModel *model);
GtkListStore *create_liststorel();

```

4、callback.c

```

#ifdef HAVE_CONFIG_H
# include <config.h>
#endif

#include <gtk/gtk.h>

#include "callbacks.h"
#include "interface.h"
#include "support.h"
//增加
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <stdio.h>
#define WCIN 1
#define WCOU 0
#define LISTIN 1
#define LISTOUT 0
#define STRIN 1
#define STROUT 0
#define MAXCHAR 255
#define MAXLINE 1000
#define MYSPLIT ' '
#define MAXITEM 50
enum
{
    COL_ID, //整数
    COL_ICON, //图标
    COL_TEXT, //字符
    NUM_COLS //最后一行
};
enum
{
    COL_TEXT1, //字符
    COL_ICON1, //图标
    COL_TEXT2, //字符

```

```

NUM_COLS1 //最后一行
};
extern gchar *filename,*tfilename;
extern GtkWidget *treeview,*treeview1;
extern GtkListStore *store,*store1;
extern GtkWidget *scrolledwindow2;
extern gchar *linestr[MAXLINE];
extern gchar *first1[MAXLINE];
extern GtkWidget *jgentry;
extern GtkWidget *filename,*tentry;
extern GtkWidget *wsentry;
extern GtkTreeModel *model,*modell;
extern guint nhang;
extern guint stepn;
extern guint stepp;
extern GtkWidget *stepspinbutton,*digspinbutton;
extern GtkWidget *textview1,*textview2;
extern xhbutton;
extern GtkWidget *hhentry;
extern gint textonly,linesum,stepf,stepb,doscheck,precheck,piccheck;
extern gchar *itemtnxt;
char *numt1[5];
int numt1id[5];
extern gchar *spechar;
extern GtkWidget *specentry,*schentry;

void
on_file_button_clicked(GtkButton *button,
                       gpointer user_data)
{
    GtkWidget *open_dialog;
    open_dialog = gtk_file_selection_new("请选择一个文件: ");
    g_signal_connect(
        GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->ok_button),"clicked",G_CALLBACK(store_
filename),open_dialog);
    g_signal_connect_swapped(
        GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->ok_button),"clicked",G_CALLBACK(
gtk_widget_destroy),open_dialog);//退出
    g_signal_connect_swapped(GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->cancel_button),
        "clicked",G_CALLBACK(gtk_widget_destroy),(gpointer) open_dialog);
    gtk_widget_show(open_dialog);
}

void
on_opobutton_clicked(GtkButton *button,
                    gpointer user_data)
{
    nhang = 0;
    model = GTK_TREE_MODEL(create_liststore(nhang));
    gtk_tree_view_set_model(GTK_TREE_VIEW(treeview),model);
}

void
on_treeview_row_activated(GtkTreeView *treeview,
                          GtkTreePath *path,
                          GtkTreeViewColumn *column,

```

```

                                gpointer      user_data)
{
    GtkTreeSelection* selection;
    GtkTreeIter      iter;
    selection = gtk_tree_view_get_selection(GTK_TREE_VIEW(treeview));
    stepn = gtk_tree_selection_count_selected_rows(selection);
    if (gtk_tree_selection_get_selected(selection, &model, &iter)){
        gtk_tree_model_get (model, &iter, COL_ID, &stepn, -1);
    }
    xzwork();
    if(stepn == 0){
        gtk_spin_button_set_value      (stepspinbutton,1);
    }
    else {  gtk_spin_button_set_value      (stepspinbutton,stepn);
    }
}

void
on_wsbutton_clicked              (GtkButton      *button,
                                gpointer      user_data)
{
    GtkTreeSelection* selection;
    GtkTreeIter      iter;
    gint hh,il,i2;
    gchar *hhstr,*hhstrl;
    selection = gtk_tree_view_get_selection(GTK_TREE_VIEW(treeview));
    stepn = gtk_tree_selection_count_selected_rows(selection);
    if (gtk_tree_selection_get_selected(selection, &model, &iter)){
        gtk_tree_model_get (model, &iter, COL_TEXT, &hhstr, -1);
    }
    hh= strlen(hhstr);
    if(doscheck==0)i2=1;
    else i2=2;
    for(il=0;il<hh-i2;il++)
        hhstrl[il]=hhstr[il];
    hhstrl[il]='\0';
    gtk_entry_set_text (GTK_ENTRY(wsentry),hhstrl);
}

void
on_twsbutton_clicked              (GtkButton      *button,
                                gpointer      user_data)
{
    xzlwork();
}

void
on_jsbutton_clicked              (GtkButton      *button,
                                gpointer      user_data)
{
    gchar *cacustrl,*strl;
    cacustrl = gtk_entry_get_text (GTK_ENTRY(wsentry));
    strl=altstr(cacustrl);
    gtk_entry_set_text (GTK_ENTRY(jgentry),strl);
}

```

```

void
on_precheckboxbutton_toggled                (GtkToggleButton *togglebutton,
                                           gpointer          user_data)
{
    if(precheck == 0){
        precheck = 1;
    }
    else {
        precheck = 0;
    }
}

void
on_xhcheckboxbutton_toggled                 (GtkToggleButton *togglebutton,
                                           gpointer          user_data)
{
    if(xhbutton == 0){
        xhbutton = 1;
    }
    else {
        xhbutton = 0;
        gtk_entry_set_text (GTK_ENTRY(wsentry),"");
    }
}

void
on_xsbutton_clicked                      (GtkButton          *button,
                                           gpointer          user_data)
{
    gint stept,ip,isum,itemdata,il=0,i2,i3;
    gchar *hhstr,*lastiteml,*itemn,*itemt,getline[MAXLINE]**items;
    FILE *fpl;
    stept = gtk_spin_button_get_value_as_int (stepspinbutton);
    if(textonly==0){
        ip = nhang-stepb-1;
        if(!(fpl = fopen (filename,"r")))
            fpl = fopen ("welcome.txt","r");
        if(fpl){
            lastiteml=LastItem(fpl,nhang);
            if(Listp(lastiteml)){
                itemt = WordBody(lastiteml);
                itemn = nthList(itemt,0,MYSPLIT);
                if(itemn != NULL){
                    itemdata = atoi(itemn);
                    if(itemdata !=0){
                        ip = nhang -stepb-itemdata - 1;
                    }
                }
            }
            fclose(fpl);
        }
        if(ip < 0)nhang=0;
        else nhang=ip;
    }
    else {

```



```

        stept = gtk_spin_button_get_value_as_int (stepspinbutton);
        ip = nhang - stept;
        if(ip<0)nhang=0;
        else nhang=ip;
    }
    if(stepn==0)stepn=1;
    model = GTK_TREE_MODEL(create_liststore(nhang));
    gtk_tree_view_set_model(GTK_TREE_VIEW(treeview),model);
}

```

```

void
on_piccheckbutton_toggled                (GtkToggleButton *togglebutton,
                                           gpointer          user_data)
{
    if(piccheck == 0){
        piccheck = 1;
    }
    else {
        piccheck = 0;
    }
}

```

```

void
on_xzbutton_clicked                      (GtkButton          *button,
                                           gpointer          user_data)
{
    xzwork();
}

```

```

void
on_cancelbutton1_clicked                (GtkButton          *button,
                                           gpointer          user_data)
{
    FILE *fd1,*fd2;
    gchar tline[MAXLINE],*tline2;
    int il,i2;
    fd1 = fopen ("mindwaytemp.txt","r");
    fd2 = fopen ("previewtemp","w");
    il=LineCntr(fd1);
    if(fd1&&fd2)for(i2=0;i2<il-1;i2++){
        fgets(tline,MAXLINE,fd1);
        fputs(tline,stdout);
        fputs(tline,fd2);
    }
    if (fd1) fclose (fd1);
    if (fd2) fclose (fd2);
    fd1 = fopen ("previewtemp", "r")      ;
    fd2 = fopen ("mindwaytemp.txt", "w");
    if (fd1 && fd2){
        while (fgets(tline,MAXLINE,fd1)) {
            fputs(tline,fd2);
            fputs(tline,stdout);
        }
    }
    if (fd1) fclose (fd1);
    if (fd2) fclose (fd2);
}

```

```
}

```

```
void
on_zhbutton_clicked          (GtkButton       *button,
                              gpointer         user_data)
{
    FILE *fp4;
    gchar *ffilename="welcome.txt",*tfilename;
    gchar name[MAXLINE];
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    tfilename = "mindwaypreview.txt";
    zh();
    fp4 = fopen (tfilename,"r");
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview1));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter   (buffer,&end);
    gtk_text_buffer_delete         (buffer,&start,&end);
    gtk_text_buffer_get_end_iter(buffer,&end);
    if(fp4){
        while(fgets(name,MAXLINE,fp4)){
            gtk_text_buffer_insert(buffer,&end,name,-1);
        }
    }
    if(fp4)fclose(fp4);
}
```

```
void
on_xxbutton_clicked          (GtkButton       *button,
                              gpointer         user_data)
{
    gint stept,ip,isum,itemdata,il=0,i2,i3;
    gchar *hhstr,*lastiteml,*itemt,*itemn,getline[MAXLINE],**items;
    FILE *fpl;
    GtkTreeSelection* selection;
    GtkTreeIter      iter;
    selection = gtk_tree_view_get_selection(GTK_TREE_VIEW(treeview));
    stepn = gtk_tree_selection_count_selected_rows(selection);
    if (gtk_tree_selection_get_selected(selection, &model, &iter)){
        gtk_tree_model_get (model, &iter, COL_ID, &stepn, -1);
    }
    if(stepn==0)stepn=1;
    if(textonly==0){
        if(!(fpl = fopen (filename,"r"))){
            fpl = fopen ("welcome.txt","r");
        }
        if(fpl){
            lastiteml=LastItem(fpl,nhang);
            if(Listp(lastiteml)){
                itemt = WordBody(lastiteml);
                itemn = nthList(itemt,stepn,MYSPLIT);
                if(itemn != NULL){
                    itemdata = atoi(itemn);
                    nhang = nhang + stepf + itemdata + 1;
                }
            }
            else {
                nhang = nhang+stepf+1;
            }
        }
    }
}
```

```

        }
        else {
            nhang = nhang+stepf+1;
        }
        fclose(fp1);
    }
    else {
        nhang = nhang+stepf+1;
    }
    if(nhang>=linesum)nhang=linesum-1;
}
else {
    stept = gtk_spin_button_get_value_as_int (stepspinbutton);
    if(stept==0)stept=1;
    ip = nhang + stept;
    if(ip>=linesum)nhang=linesum;
    else nhang=ip;
}
model = GTK_TREE_MODEL(create_liststore(nhang));
gtk_tree_view_set_model(GTK_TREE_VIEW(treeview),model);
}

void
on_imagebutton_clicked          (GtkButton      *button,
                                gpointer         user_data)
{
    dispimage();
}

void
on_treeview1_row_activated      (GtkTreeView    *treeview,
                                GtkTreePath     *path,
                                GtkTreeViewColumn *column,
                                gpointer         user_data)
{
    xz1work();
}

void
on_tfilebutton_clicked          (GtkButton      *button,
                                gpointer         user_data)
{
    GtkWidget *open_dialog;
    open_dialog = gtk_file_selection_new("请选择一个文件: ");
    g_signal_connect(
        GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->ok_button),"clicked",G_CALLBACK(store_
tfilename),open_dialog);
    g_signal_connect_swapped(
        GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->ok_button),"clicked",G_CALLBACK
(gtk_widget_destroy),open_dialog);//退出
    g_signal_connect_swapped (GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->cancel_butt
on),
        "clicked",G_CALLBACK(gtk_widget_destroy),(gpointer) open_dialog);
    gtk_widget_show(open_dialog);
}

```

```

void
on_topenbutton_clicked          (GtkButton      *button,
                                gpointer         user_data)
{
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    FILE *fp4;
    gchar name[MAXLINE];
    fp4 = fopen (tfilename,"r");
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter (buffer,&end);
    gtk_text_buffer_delete (buffer,&start,&end);
    gtk_text_buffer_get_end_iter(buffer,&end);
    if(fp4){
        while(fgets(name,MAXLINE,fp4)){
            gtk_text_buffer_insert(buffer,&end,name,-1);
        }
    }
    if(fp4)fclose(fp4);
}

```

```

void
on_newbutton_activate          (GtkButton      *button,
                                gpointer         user_data)
{
    FILE *fp4;
    gchar *tempname;
    tempname = gtk_entry_get_text (GTK_ENTRY(tentry));
    fp4 = fopen (tempname,"a");
    if(fp4){
        fclose(fp4);
    }
    else
        g_printf("新建文件: %s 未成功! \n",tempname);
}

```

```

void
on_bakbutton_clicked          (GtkButton      *button,
                                gpointer         user_data)
{
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    FILE *fp4;
    gchar *name,*tempfile,ch;
    gint i,il;
    tempfile = "baktemp.txt";
    fp4 = fopen (tempfile,"w");
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter (buffer,&end);
    name = gtk_text_buffer_get_text(buffer,&start,&end,TRUE);
    if(fp4){
        il=strlen(name);
        for(i=0;i<il;i++){

```

```

        putc(name[i],fp4);
    }
    fclose(fp4);
}

void
on_resbutton_clicked          (GtkButton       *button,
                               gpointer         user_data)
{
    textview_display_file (textview2,"baktemp.txt");
}

void
on_savbutton_clicked          (GtkButton       *button,
                               gpointer         user_data)
{
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    FILE *fp4;
    gchar *name,*tempfile,ch;
    gint i,il;
    tempfile = gtk_entry_get_text (GTK_ENTRY(tentry));
    fp4 = fopen (tempfile,"w");
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter (buffer,&end);
    name = gtk_text_buffer_get_text(buffer,&start,&end,TRUE);
    if(fp4){
        il=strlen(name);
        for(i=0;i<il;i++){
            putc(name[i],fp4);
        }
        fclose(fp4);
    }
}

void
on_assbutton_clicked          (GtkButton       *button,
                               gpointer         user_data)
{
    FILE *fp4;
    gchar *ffilename="welcome.txt",*tfilename;
    gchar name[MAXLINE];
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    tfilename = "mindwaypreview.txt";
    zh();
    fp4 = fopen (tfilename,"r");
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
    if(fp4){
        while(fgets(name,MAXLINE,fp4)){
            gtk_text_buffer_insert_at_cursor(buffer,name,-1);
        }
    }
    if(fp4)fclose(fp4);
}

```

```

}

void
on_drbutton_clicked                      (GtkButton      *button,
                                           gpointer      user_data)
{
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    FILE *fp4;
    gchar *name,*tempfile,ch;
    gint i,il;
    tempfile = "previewtemp";
    fp4 = fopen (tempfile,"w");
    // g_printf("进入保存操作文件:%s! \n",tempfile);
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter (buffer,&end);
    name = gtk_text_buffer_get_text(buffer,&start,&end,TRUE);
    // g_printf("缓冲这符:%s! \n",name);
    if(fp4){
        il=strlen(name);
        for(i=0;i<il;i++){
            if(name[i]!='r')putc(name[i],fp4);
        }
        putc(name[i],stdout);
    }
    fclose(fp4);
    textview_display_file (textview2,"previewtemp");
}

```

```

void
on_dnbutton_clicked                      (GtkButton      *button,
                                           gpointer      user_data)
{
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    FILE *fp4;
    gchar *name,*tempfile,ch;
    gint i,il;
    tempfile = "previewtemp";
    fp4 = fopen (tempfile,"w");
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter (buffer,&end);
    name = gtk_text_buffer_get_text(buffer,&start,&end,TRUE);
    if(fp4){
        il=strlen(name);
        for(i=0;i<il;i++){
            if(name[i]!='n')putc(name[i],fp4);
        }
        fclose(fp4);
    }
    textview_display_file (textview2,"previewtemp");
}

```

```

void

```

```

on_xdbutton_clicked                                (GtkButton      *button,
                                                    gpointer         user_data)
{
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    FILE *fp4;
    gchar *name,*tempfile,ch;
    gint i,i1,i2,i3,in1,in2,it;
    tempfile = "previewtemp";
    fp4 = fopen (tempfile,"w");
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter (buffer,&end);
    name = gtk_text_buffer_get_text(buffer,&start,&end,TRUE);
    i2=0;
    i3 = 0;
    in1 =0;
    in2=0;
    if(fp4){
        il=strlen(name);
        for(i=0;i<il;i++){
            if(name[i]!='\n'){
                if (name[i]=='\n') in2++;
            }
            else {
                fprintf(fp4,"%d %d\n",in1,in2);
                for(it= i2;it<i;it++)
                    if(name[it]!='\n'){
                        putc(name[it],fp4);
                    }
                in1=in2;
                in2=0;
                i2 = i;
            }
        }
        fprintf(fp4,"%d %d\n",in1,in2+1);
        for(it= i2;it<i;it++)
            if(name[it]!='\n'){
                putc(name[it],fp4);
            }
        fprintf(fp4,"\n%d 0\n",in2+1);
        fclose(fp4);
    }
    textview_display_file (textview2,"previewtemp");
}

void
on_gjbutton_clicked                                (GtkButton      *button,
                                                    gpointer         user_data)
{
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    FILE *fp3,*fp4;
    gchar *name,*tempfile,*tempfile1,ch,ch1,ch2,chlt,ch2t,*linch1,*linch2;
    gint i,i1,i2,i3,in1,in2,it,lin1,lin2,nxtlin,prelin,printl,print2;
    tempfile = "previewtemp";
    tempfile1 = "mindwaypreview.txt";
    fp3 = fopen (tempfile,"w");

```



```

buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
gtk_text_buffer_get_start_iter (buffer,&start);
gtk_text_buffer_get_end_iter (buffer,&end);
name = gtk_text_buffer_get_text(buffer,&start,&end,TRUE);
if(fp3){
    il=strlen(name);
    for(i=0;i<il;i++){
        putc(name[i],fp3);
    }
    fclose(fp3);
}
while(strchr(name,"")){
    fp3 = fopen (tempfile,"r");
    fp4 = fopen (tempfile,"w");
    i2=0;
    i3 = 0;
    inl =0;
    in2=0;
    if(fp4&&fp3){
        for(i=0;i<il;i++){
            if(name[i]!='\n'){
                if (name[i]=='\n') {
                    in2++;
                }
            }
            else {
                ch1=name[i+1];
                ch2=name[i+2];
                linch2=nth(fp3,in2,1,0);
                linl = in2;
                i2=i;
                break;
            }
        }
        for(i=i2+1;i<il;i++){
            if(name[i]!='\n'){
                if (name[i]=='\n'){
                    in2++;
                }
            }
            else {
                ch1t=name[i+1];
                ch2t=name[i+2];
                if((ch1==ch1t) && (ch2==ch2t)){
                    i3=i;
                    linch1=nth(fp3,in2,0,0);
                    lin2 = in2;
                    nextlin = lin2-(linl+atoi(linch2)+1);
                    prelin = lin2-atoi(linch1)-(linl+1);
                    break;
                }
            }
        }
        printl=0;
        print2=0;
        for(i=0;i<il;i++){
            if ((i>=i2)&&(i<=i2+2)){
                if(printl==0){
                    fprintf(fp4," %d ",nextlin);

```

```

        print1=1;
    }
}
else if ((i>=i3)&&(i<=i3+2)){
    if(print2==0){
        fprintf(fp4," %d ",prelin);
        print2=1;
    }
}
else{
    putc(name[i],fp4);
}
}
fclose(fp3);
fclose(fp4);
}
textview_display_file (textview2,"mindwaypreview.txt");
buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
gtk_text_buffer_get_start_iter (buffer,&start);
gtk_text_buffer_get_end_iter (buffer,&end);
name = gtk_text_buffer_get_text(buffer,&start,&end,TRUE);
fp3 = fopen (tempfile,"w");
if(fp3){
    il=strlen(name);
    for(i=0;i<il;i++){
        putc(name[i],fp3);
    }
    fclose(fp3);
}
}
}

void
on_textview2_move_cursor (GtkTextView *textview,
                           GtkMovementStep step,
                           gint count,
                           gboolean extend_selection,
                           gpointer user_data)
{

}

void
on_specbutton_clicked (GtkButton *button,
                       gpointer user_data)
{
    GtkWidget *open_dialog;
    open_dialog = gtk_file_selection_new("请选择一个特殊字符文件: ");
    g_signal_connect(
        GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->ok_button),"clicked",G_CALLBACK(store_
sfilename),open_dialog);
    g_signal_connect_swapped(
        GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->ok_button),"clicked",G_CALLBACK
(gtk_widget_destroy),open_dialog);//退出
    g_signal_connect_swapped (GTK_OBJECT(GTK_FILE_SELECTION(open_dialog)->cancelbutt
on),
        "clicked",G_CALLBACK(gtk_widget_destroy),(gpointer) open_dialog);
    gtk_widget_show(open_dialog);
}

```

```

}

void
on_opobutton_clicked          (GtkButton      *button,
                               gpointer        user_data)
{
    model1 = GTK_TREE_MODEL(create_liststore1());
    gtk_tree_view_set_model(GTK_TREE_VIEW(treeview1),model1);
}

void
on_okbutton1_clicked         (GtkButton      *button,
                               gpointer        user_data)
{
    gtk_main_quit();
}

void store_filename(GtkFileSelection *selector,gpointer user_data)
{
    filename = gtk_file_selection_get_filename(GTK_FILE_SELECTION(user_data));
    gtk_entry_set_text (GTK_ENTRY(filename),filename);
}
void store_tfilename(GtkFileSelection *selector,gpointer user_data)
{
    tfilename = gtk_file_selection_get_filename(GTK_FILE_SELECTION(user_data));
    gtk_entry_set_text (GTK_ENTRY(tentry),tfilename);
}
void xzwork()
{
    FILE *fpl;
    gchar *jlin_e,*jgstr;
    GtkTreeSelection* selection;
    GtkTreeIter      iter;
    gint hh;
    gchar *hhstr;
    selection = gtk_tree_view_get_selection(GTK_TREE_VIEW(treeview));
    stepn = gtk_tree_selection_count_selected_rows(selection);
    if (gtk_tree_selection_get_selected(selection, &model, &iter)){
        gtk_tree_model_get (model, &iter, COL_ID, &stepn, -1);
    }

    if(nhang == 0){
        fpl = fopen ("mindwaytemp.txt","w");
    }
    else {
        fpl = fopen ("mindwaytemp.txt","a");
    }
    if (fpl){
        fprintf (fpl,"%d ",nhang);
        fprintf (fpl,"%d ",stepn);
        fprintf (fpl,"%d ",0);
        jlin_e = gtk_entry_get_text (GTK_ENTRY(wsentry));
        jgstr = gtk_entry_get_text (GTK_ENTRY(jgentry));
        if (jlin_e != NULL) {
            fprintf (fpl,"%s",jlin_e);
            if (strlen(jgstr) != 0){
                fprintf (fpl,"\\n  =%s",jgstr);//v31
            }
        }
    }
}

```

```

        }
        if(xhbutton == 0)fprintf (fp1,"\\n\\n");
        else fprintf (fp1,"\\n");
    }
}
if(fp1)fclose(fp1);
}
void xzlwork()
{
    GtkTreeSelection* selection;
    GtkTreeIter      iter;
    gint isyb,il,i2,lin;
    gchar *sybstr,*sybstr1;
    selection = gtk_tree_view_get_selection(GTK_TREE_VIEW(treeview1));
    lin = gtk_tree_selection_count_selected_rows(selection);
    if (gtk_tree_selection_get_selected(selection, &model, &iter)){
        gtk_tree_model_get (model, &iter, COL_TEXT1, &sybstr, -1);
    }
    isyb= strlen(sybstr);
    sybstr1 = strndup(sybstr,isyb);
    if(precheck==0){
        gtk_entry_append_text(wsentry,sybstr1);
    }
    else {
        gtk_entry_prepend_text(wsentry,sybstr1);
    }
}
void zh ()
{
    FILE *fp1,*fp2,*fp3,*fp4;
    gchar *ffilename="welcome.txt",*tfilename="a.";
    gchar name[MAXLINE];
    gchar *texts[MAXCHAR],*strpre,*strnxt;
    gint il,i2,i3,it,itl,ic,jt,jc,ptl[3],j,lineid,lineid1,lineidn[3];
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    long filpos;
    int c,retn=0;
    fp2 = fopen ("previewtemp","w");
    if(filename == NULL){
        tfilename = "mindwaypreview.txt";
        fp4 = fopen (tfilename,"w");
    }
    else {
        tfilename = "mindwaypreview.txt";
        ffilename = filename;
        fp4 = fopen (tfilename,"w");
    }
    fp3 = fopen ("mindwaytemp.txt","r");
    fp1 = fopen (ffilename,"r");
    if (fp1){
    }
    if (fp2){
        ic = LineCntr(fp3);
        for (it=0;it<ic;it++){
            jc = ItemCntr(fp3,ic);
            lineid = 0;
            for (jt=0;jt<jc;jt++) {
                texts[jt] = nth (fp3,it,jt,0);
            }
        }
    }
}

```

```

        if(jt<3){
            lineidn[it] = atoi (texts[jt]);
            if(jt==1)lineidl=lineidn[it];
            lineid = lineid + lineidn[it];
        }
    }
    rewind(fp1);
    for (itl=0;itl<=lineid;itl++){
        if(itl == lineid){
            il=WrdCntr(fp1,itl);
            if(doscheck==0)fgets(name,il,fp1);
            else fgets(name,il-1,fp1);
            if(textonly==1){
                if(lineidl!=0) {
                    fputs(name, fp2);
                }
            }
            else {
                if(lineidl!=0) {
                    fputs(name, fp2);
                }
            }
        }
        else {
            fgets(name,MAXLINE,fp1);
        }
    }
    if(jc == 4){
        strnxt = nth (fp3,it,3,0);
        strnxt = WordBody(strnxt);
        if(strnxt != NULL){
            il = strlen(strnxt);
            fprintf (fp2,strnxt);
        }
    }
}

if(fp1)fclose(fp1);
if(fp2)fclose(fp2);
if(fp3)fclose(fp3);
fp2 = fopen ("previewtemp","r");
if(fp2){
    retn =0;
    while ((c = getc(fp2)) != EOF) {
        if(retn == '\\') {
            if(c = 'n'){
                putc('\n',fp4);
                retn = 0;
                continue;
            }
            else {
                putc('\\',fp4);
                putc(c,fp4);
                retn = 0;
                continue;
            }
        }
        else if(c == '\\'){
            retn = c;

```

```

        }
        else retn =0;
        if(retn == 0)putc(c,fp4);
    }
}
if(fp2)fclose(fp2);
if(fp4)fclose(fp4);
}
void dispimage()
{
    GtkWidget * windowimg;
    if(strlen(itemtnxt) != 0){
        windowimg=imagewin(itemtnxt);
        gtk_widget_show(windowimg);
    }
}
GtkWidget *imagewin( gchar *filename)
{
    GtkWidget * windowimg;
    GtkWidget *image;
    windowimg = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title(GTK_WINDOW(windowimg),filename);
    gtk_container_set_border_width(GTK_CONTAINER(windowimg),10);
    image=gtk_image_new_from_file(filename);
    gtk_container_add(GTK_CONTAINER(windowimg),image);
    gtk_widget_show_all(windowimg);
    return windowimg;
}
void textview_display_file (GtkWidget *textview ,gchar *filename)
{
    GtkTextBuffer *buffer;
    GtkTextIter start,end;
    FILE *fp4;
    gchar name[MAXLINE];
    fp4 = fopen (filename,"r");
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter (buffer,&end);
    gtk_text_buffer_delete (buffer,&start,&end);
    gtk_text_buffer_get_end_iter(buffer,&end);
    if(fp4){
        while(fgets(name,MAXLINE,fp4)){
            gtk_text_buffer_insert(buffer,&end,name,-1);
        }
    }
    if(fp4)fclose(fp4);
}
void store_sfilename(GtkFileSelection *selector,gpointer user_data)
{
    gchar * spechar = gtk_file_selection_get_filename(GTK_FILE_SELECTION(user_data));
    gtk_entry_set_text (GTK_ENTRY(specentry),spechar);
}

gchar * nth (FILE *fp , int line , int column , int id )
{
    //id=0,only text;id=1,text with '('&')'
    gint c,nl,nw,nc,state,list_state,kn;
    gint i;
    gint retn=0, str_state, alist = 0; //是括号

```

```

gchar str[MAXCHAR],*strl;
long filpos;
filpos = ftell(fp);
if(fp){
    rewind(fp);
}
state =      WCOUT;
list_state = LISTOUT;
str_state = STROUT;
nl = nw = nc = kn = i = 0;
while (((c = getc(fp)) != EOF) && (nl <= line) ) {
    if(retn== 1) {
        retn = 0;
        nw = 0;
    }
    if (c == '\n'){
        nl++;
        retn =1;
        state =      WCOUT;
    }
    if(c == '('){
        list_state = LISTIN;
        kn++;
    }
    else if ((list_state == LISTIN) && (c ==  ')){
        kn--;
        if(kn == 0){
            list_state = LISTOUT;
        }
    }
    if (c==""){
        if(str_state == STRIN){
            str_state = STROUT;
        }
        else {
            str_state =STRIN;
        }
    }
    if((list_state == LISTOUT)&&(str_state == STROUT)){
        if(c == MYSPLIT || c == '\n' || c == '\t'){
            if(state == WCIN){
                nw++;
                state = WCOUT;
            }
        }
        else if (state == WCOUT) {
            state = WCIN;
        }
    }
    if(nl == line && nw ==  column ){
        if((list_state == LISTOUT)&&(str_state == STROUT)){
            if((c!='\n')&& (c!='\t')&&(c!='\r')&&(c!=MYSPLIT)){
                str[i] = c ;
                i++;
            }
            else continue;
        }
        else {
            str[i] = c ;

```



```

        i++;
    }
}
str[i]='\0';
str1 = strdup(str,i);
fseek(fp,filpos,SEEK_SET);
return str1;
}
gchar *LastItem (FILE *fp,gint LineNo)
{
    gint c,nl,il;
    gchar *str,*str1;
    long filpos;
    filpos = ftell(fp);
    if(fp)rewind(fp);
    nl=ItemCntr(fp,LineNo);
    str=nth(fp,LineNo,nl-1,0);
    il=strlen(str);
    str1=strdup(str,il);
    fseek(fp,filpos,SEEK_SET);
    return str1;
}

gboolean Listp (gchar *str)
{
    gint il,i2=0,nl;
    nl=strlen(str);
    for(il=0;il<nl;il++)
        if(str[il]=='(')i2=1;
        if(str[il]==')')i2+=1;
    if(i2>0){
        return TRUE;
    }
    else return FALSE;
}

int LineCntr(FILE *fp)
{
    int c,nl;
    nl = 0;
    long filpos;
    filpos = ftell(fp);
    if(fp)rewind(fp);
    while ((c = getc(fp)) != EOF)
        if (c == '\n')
            ++nl;
    fseek(fp,filpos,SEEK_SET);
    return nl;
}
int ItemCntr(FILE *fp,int LineNo)
{
    int c,nl,nw,nc,state,list_state,kn;
    int i;
    int retn=0, str_state= 0, alist = 0;
    char str[MAXCHAR];
    long filpos;
    filpos = ftell(fp);

```

```

        if(fp)rewind(fp);
        state =      WCOUT;
        list_state =  LISTOUT;
        nl = nw = nc = kn = i = 0;
        while ((c = getc(fp)) != EOF ) {
            if(retn== 1) {
                retn = 0;
                nw = 0;
            }
            if (c == '\n'){
                nw++;
                nl++;
                if(nl > LineNo )      {
                    break;
                }
                else {
                    retn =1;
                    state =      WCOUT;
                }
            }
            if(c == '('){
                list_state = LISTIN;
                kn++;
            }
            if (c==""){
                if(str_state == STRIN)str_state=STROUT;
                else str_state = STRIN;
            }
            if((list_state == LISTOUT)&&(str_state == STROUT)){
                if(c == MYSPLIT){
                    if(state == WCIN){
                        nw++;
                        state = WCOUT;
                    }
                }
                else if (state == WCOUT) {
                    state = WCIN;
                }
            }
            else if ((list_state == LISTIN) && (c == ' ')) {
                kn--;
                if(kn == 0){
                    list_state = LISTOUT;
                }
            }
        }

        fseek(fp,filpos,SEEK_SET);
        return nw;
    }

int WrdCntr(FILE *fp, int line)
{
    int c,nl,nw;
    long filpos;
    filpos = ftell(fp);
    nl = nw = 0;
    if(fp)rewind(fp);
    while ((c = getc(fp)) != EOF) {
        nw++;

```

```

        if (c == '\n'){
            ++nl;
            if(line < nl){
                fseek(fp,filpos,SEEK_SET);
                return nw;
            }
            nw = 0;
        }
    }
}

char * WordBody(char *str)
{
    int i1,i2;
    char *str1,*str2;
    i1 = strlen(str);
    for (i2=0;i2<(i1-2);i2++)
        str1[i2] = str[i2+1];
    str1[i2] = '\0';
    str2=strndup(str1,i2);
    return str2;
}

gchar * nthList (gchar *str , int column , gchar ch)
{
    gint nl,nw,nc,state,list_state,kn;
    gint i,it,i1;
    gint retn=0, khstate= 0, alist = 0;
    gchar c,str1[MAXCHAR],*str2;
    state =      WCOUNT;
    list_state = LISTOUT;
    nl = nw = nc = kn = i = 0;
    kn = strlen(str);
    for(it=0;it<kn;it++){
        c=str[it];
        if(retn== 1) {
            retn = 0;
            nw = 0;
        }
        if (c == '\n'){
            nl++;
            retn =1;
            state =      WCOUNT;
        }
        if(c == ' '){
            list_state = LISTIN;
            kn++;
            khstate = 1;
        }
        else if ((list_state == LISTIN) && (c == ' ')) {
            kn--;
            if(kn == 0){
                list_state = LISTOUT;
                khstate = 0;
            }
        }
        if(list_state == LISTOUT){
            if(c == ch || c == '\n' || c == '\t'){
                if(state == WCIN){

```

```

        nw++;
        state = WCOUT;
    }
}
else if (state == WCOUT) {
    state = WCIN;
}
}
if( nw == column ){
    if(list_state == LISTOUT){
        if((c!='\n')&& (c!='\t')&&(c!='\r')&&(c!=ch)){
            str1[i] = c ;
            i++;
        }
    }
    else {
        if((c!='\n')&& (c!='\t')&&(c!='\r')){
            str1[i] = c ;
            i++;
        }
    }
}
str1[i]='\0';
il=strlen(str1);
str2=strndup(str1,il);
return str2;
}
GtkWidget *
create_treeview(GtkTreeModel *model)
{
    GtkTreeViewColumn *col;
    GtkCellRenderer *renderer;
    GtkWidget *view;
    view = gtk_tree_view_new_with_model(model);

    col = gtk_tree_view_column_new();
    gtk_tree_view_column_set_title(col, "行号");

    renderer = gtk_cell_renderer_text_new();
    gtk_tree_view_column_pack_start(col, renderer, TRUE);
    gtk_tree_view_column_set_attributes(col, renderer,
                                        "text", COL_ID,
                                        NULL);

    renderer = gtk_cell_renderer_pixbuf_new();
    gtk_tree_view_column_pack_start(col, renderer, FALSE);
    gtk_tree_view_column_set_attributes(col, renderer,
                                        "pixbuf", COL_ICON,
                                        NULL);

    gtk_tree_view_append_column(GTK_TREE_VIEW(view), col);

    /* 2nd column */

    col = gtk_tree_view_column_new();
    gtk_tree_view_column_set_title(col, "内容");

    renderer = gtk_cell_renderer_text_new();

```

```

    gtk_tree_view_column_pack_start(col, renderer, TRUE);
    gtk_tree_view_column_set_attributes(col, renderer,
                                        "text", COL_TEXT,
                                        NULL);

    gtk_tree_view_append_column(GTK_TREE_VIEW(view), col);

    return view;
}

GtkWidget *
create_treeview1(GtkTreeModel *model)
{
    GtkTreeViewColumn *col;
    GtkCellRenderer *renderer;
    GtkWidget *view;
    view = gtk_tree_view_new_with_model(model);

    col = gtk_tree_view_column_new();
    gtk_tree_view_column_set_title(col, "内容");

    renderer = gtk_cell_renderer_text_new();
    gtk_tree_view_column_pack_start(col, renderer, TRUE);
    gtk_tree_view_column_set_attributes(col, renderer,
                                        "text", COL_TEXT1,
                                        NULL);

    renderer = gtk_cell_renderer_pixbuf_new();
    gtk_tree_view_column_pack_start(col, renderer, FALSE);
    gtk_tree_view_column_set_attributes(col, renderer,
                                        "pixbuf", COL_ICON1,
                                        NULL);

    gtk_tree_view_append_column(GTK_TREE_VIEW(view), col);

    /* 2nd column */

    col = gtk_tree_view_column_new();
    gtk_tree_view_column_set_title(col, "备注");

    renderer = gtk_cell_renderer_text_new();
    gtk_tree_view_column_pack_start(col, renderer, TRUE);
    gtk_tree_view_column_set_attributes(col, renderer,
                                        "text", COL_TEXT2,
                                        NULL);

    gtk_tree_view_append_column(GTK_TREE_VIEW(view), col);
    return view;
}

GtkListStore *
create_liststore(gint nh )
{
    GtkTreeIter iter;
    GdkPixbuf *icon,*iconc,*iconr;
    GError *error = NULL;
    gint ch,i;
    gint nh0 = 0;
    gint ptl[3];
    FILE *fpl;
    gchar name[MAXLINE],*linestr[MAXLINE],*item[4],*item3;

```

```

gchar *itemt,*itemtpre,*itemtl,*itemlast1,*itemlast2,*itemn;
gint itemdata1,itemdata2,nw,nt,il,i2,i3,itemdata[2],equalp=0,branchp=1;
if(!fpl = fopen (filename,"r"))
    fpl = fopen ("welcome.txt","r");
if(nh==0){
    linesum = LineCntr (fpl);
    nw = ItemCntr(fpl,0);
    if(nw > 1){
        for(il=0;il<nw;il++){
            item[il] = nth(fpl,0,il,0);
            if(il<2)itemdata[il]= atoi(item[il]);
        }
        for(il=0;il<nw;il++){
        }
        if((itemdata[0] == 0) && (itemdata[1] != 0)){
            textonly = 0;
            stepb = itemdata[0];
            stepf = itemdata[1];
        }
    }
    else {
        textonly = 1;
    }
    if(textonly == 0){
        nt = nh+stepf+1;
    }
    else {
        nt = linesum;
        if(linesum > MAXLINE) nt=MAXLINE;
    }

    fgets(name,MAXLINE,fpl);
    i2 = strlen(name);
    doscheck =0;
    for(il=0;il<i2;il++){
        ch=name[il];
        if(ch == '\r'){
            doscheck =1;
        }
    }
}
else {
    if(textonly == 0){
        nw = ItemCntr(fpl,nh);
        for(il=0;il<nw;il++){
            item[il] = nth(fpl,nh,il,0);
            if(il<2)itemdata[il]= atoi(item[il]);
        }
        stepb = itemdata[0];
        stepf = itemdata[1];
        nt = nh+stepf+1;
    }
    else {
        nt = linesum;
        if(linesum > MAXLINE) nt=MAXLINE;
    }
}
rewind(fpl);
store = gtk_list_store_new(3, G_TYPE_INT, GDK_TYPE_PIXBUF,G_TYPE_STRING);
icon = gdk_pixbuf_new_from_file("icon.png", &error);

```

```

    if (error){
        g_warning ("Could not load icon: %s\n", error->message);
        g_error_free(error);
        error = NULL;
    }
    iconc = gdk_pixbuf_new_from_file("click.png", &error);
    if (error) {
        g_warning ("Could not load icon: %s\n", error->message);
        g_error_free(error);
        error = NULL;
    }
    iconr = gdk_pixbuf_new_from_file("ruiyix32.png", &error);
    if (error)
    {
        g_warning ("Could not load icon: %s\n", error->message);
        g_error_free(error);
        error = NULL;
    }

    itemlast1 = "";
    itemlast2 = "";
    for(i=0;i< nt;i++){
        if(i >= nh){
            gtk_list_store_append(store, &iter);
            if(nh0==0){
                if(textonly == 0){
                    nw = ItemCntr(fpl,nh);
                    item3 = nth(fpl,i,2,0);
                    il= strlen(item3);
                    itemt1 = strdup(item3,il);
                    itemt = WordBody(itemt1);
                    for(i2=0;i2<i1-2;i2++)if(itemt[i2]=='=')equalp=1;
                    if(equalp==1){
                        itemtpre=nthList(itemt,0,'=');
                        if( strcmp(itemtpre,"src") == 0 ){
                            for(i2=0;i2<i1-2;i2++)if(itemt[i2]==';')bran
                                if(branchp==1){
                                    itemtpre=nthList(itemt,1,';');
                                    itemt1=nthList(itemt,0,';');
                                    itemtnxt = nthList(itemt1,1,'=');
                                    if(piccheck == 1)dispimage();
                                }
                                else    itemtnxt = nthList(itemt,1,'=');
                            }
                        }
                    }
                    else    itemtnxt = "";

                    itemlast2 = LastItem(fpl,i);
                    il= strlen(itemlast2);
                    itemlast1 = strdup(itemlast2,il);
                    itemt1 = WordBody(itemlast1);

                    if(Listp(item3)){
                        gtk_list_store_set(store, &iter,
                            COL_ID, nh0,
                            COL_ICON, icon,
                            COL_TEXT, NULL,
                            -1);
                    }
                }
            }
        }
    }
    chp=1;

```

```

        else if(equalp == 1){
            if(branchp==1){
                gtk_list_store_set(store, &iter,
                                   COL_ID, nh0,
COL_ICON, iconr,
COL_TEXT,itemtpre,
                                   -1);
            }
            else {
                gtk_list_store_set(store, &iter,
                                   COL_ID, nh0,
COL_ICON, iconr,
COL_TEXT,NULL,
                                   -1);
            }
        }
        else {
            gtk_list_store_set(store, &iter,
                               COL_ID, nh0,
COL_ICON, icon,
COL_TEXT, itemt,
                               -1);
        }

        fgets(name,MAXLINE,fpl);
    }
    else {
        gtk_list_store_set(store, &iter,
                           COL_ID, nh0,
COL_ICON, icon,
COL_TEXT, fgets(name,MAXLINE,fpl),
                           -1);
    }
}
else {
    if(textonly == 0){
        if(Listp(itemlast2)){
            itemn = nthList(itemt1,nh0,MYSPLIT);
            if(strlen(itemn) > 0){
                if(itemn[0] != '0' ) {
                    gtk_list_store_set(store, &iter,
                                       COL_ID, nh0,
COL_ICON, iconc,
COL_TEXT, fgets(name,MAXLINE,fpl),
                                       -1);
                }
                else{
                    gtk_list_store_set(store, &iter,
                                       COL_ID, nh0,
COL_ICON, NULL,
COL_TEXT, fgets(name,MAXLINE,fpl),
                                       -1);
                }
            }
        }
        else{
            gtk_list_store_set(store, &iter,
                               COL_ID, nh0,
COL_ICON, NULL,
COL_TEXT, fgets(name,MAXLINE,fpl),
                               -1);
        }
    }
}

```



```

        -1);
    }
}
else {
    gtk_list_store_set(store, &iter,
        COL_ID, nh0,
        COL_ICON, NULL,
        COL_TEXT, fgets(name,MAXLINE,fpl),
        -1);
}
}
else {
    gtk_list_store_set(store, &iter,
        COL_ID, nh0,
        COL_ICON, NULL,
        COL_TEXT, fgets(name,MAXLINE,fpl),
        -1);
}
}
nh0++;
}
else {
    fgets(name,MAXLINE,fpl);
}
}
if(fpl)fclose(fpl);
return store;
}
GtkWidget *
create_liststorel( )
{
    GtkTreeIter    iter;
    GdkPixbuf      *icon;
    GError          *error = NULL;
    FILE *fpl;
    gchar name[MAXLINE],*item[3];
    gint nt,i,il,i2;
    spechar = gtk_entry_get_text (GTK_ENTRY(specentry));
    if(spechar == NULL) spechar="spechar.txt";
    fpl = fopen (spechar,"r");
    if(fpl){
        storel = gtk_list_store_new(3,  G_TYPE_STRING, GDK_TYPE_PIXBUF,G_TYPE_STRING);
        nt = LineCntr(fpl);
        for(i=0;i< nt;i++){
            fgets(name,MAXLINE,fpl);
            for(i2=0;i2<3;i2++){
                item[i2]=nthList (name,i2,MYSPLIT);
                if(strlen(item[2])!= 0) icon = gdk_pixbuf_new_from_file(item[2], &error);
                else {
                    icon = gdk_pixbuf_new_from_file("icon.png", &error);
                }
                if (error) {
                    g_warning ("Could not load icon: %s\n", error->message);
                    g_error_free(error);
                    error = NULL;
                }
                gtk_list_store_append(storel, &iter);
                gtk_list_store_set(storel, &iter,
                    COL_TEXT1, item[0],

```

```

        COL_ICON1, icon,
        COL_TEXT2, item[1],
                                -1);
    }
    if(fp1)fclose(fp1);
}
return storel;
}

gchar * altstr(gchar *str1)
{
    int strn,digit_num;
    int bracketn,slen,nct,kl,il,i2,bracketnt,numn,bracketb,brackete;
    char *prestr,*midstr,*endstr,prestr1[MAXCHAR],midstr1[MAXCHAR],endstr1[MAXCHAR],sta,*values;
    double value;
    bracketn = 0;
    slen = strlen( str1);
    nct =0 ;
    while(nct < slen){
        strn = subchar (str1, nct);
        nct++;
        if(strn == '(')bracketn = bracketn+1;
    }
    kl=1;
    for(kl=0;kl<bracketn;kl++){
        bracketnt = 0;
        numn = 0;
        bracketb = 0;
        brackete = 0;
        nct = 0;
        sta = 1;
        while (nct < slen){
            strn = subchar(str1,nct);
            if(strn == '(') {
                bracketnt = nct;
            }
            else if(strn==')') {
                bracketb = bracketnt;
                brackete = nct;
                break;
            }
            nct ++;
        }
        prestr = substr(str1,0,bracketb );
        midstr = substr(str1,(bracketb+1),(brackete-bracketb-1));
        endstr = substr(str1,(brackete+1),(slen-brackete));
        if (midstr != NULL){
            il = strlen(midstr);
            str1 = strdup(midstr,il);
            str1 = strcasu(str1);
        }
        str1=strlink(prestr,str1,MAXCHAR);
        printf("连接 1,2 后字符串: %s !\n",str1);
        str1=strlink(str1,endstr,MAXCHAR);
        printf("连接 2,3 后字符串: %s !\n",str1);
    }
}

```

```

    str1=strcacu ( str1);
    value = atof (str1);
    digit_num = gtk_spin_button_get_value_as_int (digspinbutton);
    if(value>1e-5 &&value<1e5){
        if(digit_num == 0)sprintf(str1,"%0f",value);
        else if(digit_num == 1)sprintf(str1,"%1f",value);
        else if(digit_num == 2)sprintf(str1,"%2f",value);
        else if(digit_num == 3)sprintf(str1,"%3f",value);
        else if(digit_num == 4)sprintf(str1,"%4f",value);
        else if(digit_num == 5)sprintf(str1,"%5f",value);
        else if(digit_num == 6)sprintf(str1,"%6f",value);
    }
    else sprintf(str1,"%5e",value);
    return str1;
}
int subchar (char * str,int nb)
{
    char *str1;
    int il,i2,ch;
    i2 = strlen(str);
    for(il=0;il<i2;il++){
        if(il==nb){
            ch = str[il];
            break;
        }
    }
    return ch;
}
char * substr (char * str,int nb,int ne)
{
    char str1[MAXCHAR],*str2;
    int il,i2,ch;
    i2 = strlen(str);
    for(il=0;il<ne;il++){
        if(il<i2){
            str1[il]= str[il+nb];
        }
        else break;
    }
    str1[il]='\0';
    str2=strndup(str1,il);
    return str2;
}

gint found (char *symb,char *lstr)
{
    int bracketn,slen,xlen,ncs,nct,nct1,pd0,pd1,pd3,founds,il,i2,i3;
    char prestr,prenum,endnum,endstr,numt[MAXCHAR],*numtt,strt,strn1,strns;
    int state = 0,statet = 0 ;
    int n5[5];
    char *numt1[5];
    bracketn = 0;
    slen = strlen (symb);
    xlen = strlen (lstr);
    ncs = 0;
    nct = 0;
    nct1 = 0;
    pd0 = 0;
    pd1 = 0;

```

```

founds=0;
pd3 = 0;
strt='\0';
while (nct < xlen) {
    strn1 = subchar(1str, nct);
    strns = subchar(symb, ncs);
    if ((numberp( strn1)) ||
        (((strn1 == '-')||(strn1==''))&&
         ((nct==0)||(strt == 'e')||(strt == 'E')||(strt==' ')||(strt=='*')||(strt=='/')))){
        state = 1;
        if(statet == state){
            numt[nctl]=strn1;
            nctl++;
        }
        else {
            if(nct == 0){
                numt[nctl]=strn1;
                nctl++;
            }
            else {
                numt[nctl]='\0';
                numtt=strndup(numt,nctl+1);
                if(strcmp(numtt,symb) == 0){
                    return nct;
                }
                nctl = 0;
                numt[nctl]=strn1;
                strt=strn1;
                nctl++;
            }
        }
    }
    else {
        state = 2;
        if(statet == state){
            numt[nctl]=strn1;
            nctl++;
        }
        else {
            statet =state;
            if(nct == 0){
                numt[nctl]=strn1;
                nctl++;
            }
            else {
                numt[nctl]='\0';
                numtt=strndup(numt,nctl+1);
                nctl = 0;
                numt[nctl]=strn1;
                nctl++;
            }
        }
    }
    statet = state;
    strt = strn1;
    nct++;
}
return 0;
}

```

```

void *arrangstr (char *symb,char *lstr)
{
    int bracketn,slen,xlen,ncs,nct,nctl,pd0,pd1,pd3,founds,il,i2,i3;
    char prestr,prenum,endnum,endstr,numt[MAXCHAR],*numtt,strt,strn1,strns;
    int state = 0,statet = 0 ;
    int n5[5];
    for(il=0;il<5;il++){
        numtlid[il]=0;
    }
    bracketn = 0;
    slen = strlen (symb);
    xlen = strlen (lstr);
    ncs = 0;
    nct = 0;
    nctl = 0;
    pd0 = 0;
    pd1 = 0;
    founds=0;
    pd3 = 0;
    strt = '\0';
    while (nct < xlen) {
        strn1 = subchar(lstr, nct);
        strns = subchar(symb, ncs);
        if ((numberp( strn1)) ||
            (((strn1 == '-')||(strn1=='+'))&&
             ((nct==0)|| (strt == 'e')||(strt == 'E')||(strt==' ')||(strt=='*')||(strt=='/'))))
            state = 1;
        if(statet == state){
            numt[nctl]=strn1;
            if(nct == (xlen-1)){
                numt[nctl]=strn1;
                numt[nctl+1]='\0';
                numtt=strndup(numt,nctl+1);
                numt1[3]=strndup(numtt,nctl+1);
                numtlid[3]=1;
                break;
            }
            nctl++;
        }
        else {
            if(nct == 0){
                numt[nctl]=strn1;
                nctl++;
            }
            else {
                numt[nctl]='\0';
                numtt=strndup(numt,nctl+1);
                if(strcmp(numtt,symb) == 0){
                    founds=1;
                }
                if(founds == 1){
                    numt1[2]=strndup(numtt,nctl);
                    numtlid[2]=1;
                    if(nct == (xlen-1)){
                        nctl = 0;
                        numt[nctl]=strn1;
                        numt[nctl+1]='\0';
                        numtt=strndup(numt,nctl+1);
                        numt1[3]=strndup(numtt,nctl+1);
                    }
                }
            }
        }
    }
}

```

```

                                numtlid[3]=1;
                                break;
                                }
                                }
                                else {
                                    if(pd0 == 0){
                                        numt1[0]=strndup(numtt,nct1);
                                        numtlid[0]=1;
                                    }
                                    else if (pd0 == 1){
                                        numt1[0]=strlink(numt1[0],numtt,MAXCHAR);
                                        numtlid[0]=1;
                                    }
                                }
                                nct1 = 0;
                                numt[nct1]=strn1;
                                strt=strn1;
                                nct1++;
                            }
                        }
                    }
                else {
                    state = 2;
                    if(statet == state){
                        numt[nct1]=strn1;
                        nct1++;
                    }
                    else {
                        statet =state;
                        if(nct == 0){
                            numt[nct1]=strn1;
                            nct1++;
                        }
                        else {
                            numt[nct1]='\0';
                            numtt=strndup(numt,nct1+1);
                            if(founds == 1){
                                numt1[3]=strndup(numtt,nct1);
                                numtlid[3]=1;
                                numt1[4]=substr(1str,nct,xlen-nct+1);
                                numtlid[4]=1;
                                break;
                            }
                        }
                        else {
                            if(pd1 == 0){
                                numt1[1]=strndup(numtt,nct1);
                                numtlid[1]=1;
                                pd1 = 1;
                            }
                            else {
                                numt1[0]=strlink(numt1[1],numt1[0],MAXCHAR);
                                numt1[1]=strndup(numtt,nct1);
                                numtlid[0]=1;
                                numtlid[1]=1;
                                pd0 =1 ;
                            }
                        }
                    }
                    nct1 = 0;
                    numt[nct1]=strn1;

```

```

        nctl++;
    }
}
    }
    nct++;
    strt = strn1;
    statet =state;
}
for(il=0;il<5;il++){
}
}
char *strlink(char * str1,char *str2,int max)
{
    int il,i2,i3;
    char str3[max],*str4;
    il= strlen(str1);
    i2 = strlen(str2);
    for(i3=0;i3<(il+i2);i3++){
        if(i3<il)str3[i3]=str1[i3];
        else if (i3 < max){
            str3[i3]=str2[i3-il];
        }
        else break;
    }
    str3[i3]='\0';
    str4= strdup(str3,i3);
    return str4;
}
gboolean numberp(char c)
{
    if ((( c >= '0') && ( c <= '9')) || (c == '.') || (c == 'e') || (c == 'E'))
        return TRUE;
    else return FALSE;
}

char *strcacu ( char *str )
{
    char *caculist1[10]={"sin","cos","tg","ctg","arcsin","arccos","arctg","sinh","cosh","tgh"};
    char *caculist2[3]={"^","ln", "lg"};
    char *caculist3[2]={"*","/"};
    char *caculist4[2]={"+", "-"};
    int listn,nc,foundt,foundl,rank,ntcl,nct2,pass,fndp,il,i2,i3,i4,i5,i6,fndid;
    char *values,*cacustrt,*cacustrt1,*cacustrt2,*strl;
    float value,paral,para2;
    int numbers;
    listn = 10;
    pass = 0;
    nc = 0;
    foundt = 0;
    rank =listn;

    for(il=0;il<listn;il++){
        foundt = found(caculist1[il],str);
        if(foundt>0){
            if(il<rank){
                rank = il;
                foundl =foundt;
            }
        }
        nc++;
    }
}

```

```

        pass = 1;
    }
}
while(pass ==1){
    for(i5=0;i5<5;i5++){
        numt1[i5]='\0';
        arrangstr(caculist1[rank],str);
        if( strcmp(numt1[2],"sin") == 0 ){
            i4=strlen(numt1[3]);
            cacustrt = strndup(numt1[3],i4);
            paral= atof (cacustrt);
            value = sin (paral);
        }
        if( strcmp(numt1[2],"cos") == 0 ){
            i4=strlen(numt1[3]);
            cacustrt = strndup(numt1[3],i4);
            paral= atof (cacustrt);
            value = cos (paral);
        }
        if( strcmp(numt1[2],"tg") == 0 ){
            i4=strlen(numt1[3]);
            cacustrt = strndup(numt1[3],i4);
            paral= atof (cacustrt);
            value = tan (paral);
        }
        if( strcmp(numt1[2],"ctg") == 0 ){
            i4=strlen(numt1[3]);
            cacustrt = strndup(numt1[3],i4);
            paral= atof (cacustrt);
            value = 1.0/tan (paral);
        }
        if( strcmp(numt1[2],"arcsin") == 0 ){
            i4=strlen(numt1[3]);
            cacustrt = strndup(numt1[3],i4);
            paral= atof (cacustrt);
            value = asin (paral);
        }
        if( strcmp(numt1[2],"arccos") == 0 ){
            i4=strlen(numt1[3]);
            cacustrt = strndup(numt1[3],i4);
            paral= atof (cacustrt);
            value = acos (paral);
        }
        if( strcmp(numt1[2],"arctg") == 0 ){
            i4=strlen(numt1[3]);
            cacustrt = strndup(numt1[3],i4);
            paral= atof (cacustrt);
            value = atan (paral);
        }
        if( strcmp(numt1[2],"sinh") == 0 ){
            i4=strlen(numt1[3]);
            cacustrt = strndup(numt1[3],i4);
            paral= atof (cacustrt);
            value = sinh (paral);
        }
        if( strcmp(numt1[2],"cosh") == 0 ){
            i4=strlen(numt1[3]);
            cacustrt = strndup(numt1[3],i4);
            paral= atof (cacustrt);
            value = cosh (paral);
        }
    }
}

```



```

    }
    if( strcmp(numt1[2],"tgh") == 0 ){
        i4=strlen(numt1[3]);
        cacustrt = strndup(numt1[3],i4);
        paral= atof (cacustrt);
        value = tanh (paral);
    }

    sprintf(values,"%10.6e",value);
    if(numtlid[0]!= 0){
        str1=numt1[0];
    }
    if(numtlid[1]!= 0){
        str1 = strlink(str1,numt1[1],MAXCHAR);
        str1 = strlink(str1,values,MAXCHAR);
    }
    else {
        str1 = values;
    }
    if(numtlid[4]!= 0){
        str1 = strlink(str1,numt1[4],MAXCHAR);
    }
    i4=strlen(str1);
    str = strndup(str1,i4);

    pass =0;
    foundt = 0;
    rank =listn;
    for(il=0;il<listn;il++){
        foundt = found(caculist1[il],str);
        if(foundt>0){
            if(il<rank){
                rank = il;
                foundl =foundt;
            }
            nc++;
            pass = 1;
        }
    }

    listn = 3;
    pass = 0;
    nc = 0;
    foundt = 0;
    rank =listn;
    for(il=0;il<listn;il++){
        foundt = found(caculist2[il],str);
        if(foundt>0){
            if(il<rank){
                rank = il;
                foundl =foundt;
            }
            nc++;
            pass = 1;
        }
    }
    while(pass ==1){
        numbers = 0;

```

```

i3=rank;
for(i5=0;i5<5;i5++)numt1[i5]='\0';
arrangestr(caculist2[i3],str);
if( strcmp(numt1[2],"^") == 0 ){
    if(numtlid[1]==1){
        i4=strlen(numt1[1]);
        cacustrt1 = strndup(numt1[1],i4);
    }
    if(numtlid[3]==1){
        i4=strlen(numt1[3]);
        cacustrt2 = strndup(numt1[3],i4);
        para2= atof(cacustrt2);
    }
    if((cacustrt1[0] == 'e')||(cacustrt1[0] == 'E')){
        if(numtlid[3]==1){
            value = exp (para2);
            numbers = 1;
        }
    }
    else {
        if(numtlid[1]==1){
            paral= atof (cacustrt1);
        }
        if((numtlid[1]==1)&&(numtlid[3]==1)){
            value = pow (paral,para2);
            numbers = 2;
        }
    }
}
if( strcmp(numt1[2],"ln") == 0 ){
    if(numtlid[3]==1){
        i4=strlen(numt1[3]);
        cacustrt = strndup(numt1[3],i4);
        paral= atof (cacustrt);
        value = log (paral);
        numbers = 1;
    }
}
if( strcmp(numt1[2],"lg") == 0 ){
    if(numtlid[3]==1){
        i4=strlen(numt1[3]);
        cacustrt = strndup(numt1[3],i4);
        paral= atof (cacustrt);
        value = log10 (paral);
        numbers = 1;
    }
}
if(numbers != 0){
    sprintf(values,"%10.6e",value);
    str1 = values;
    if(numtlid[0]!= 0){
        if(numbers == 1){
            str1 = strlink(numt1[0],str1,MAXCHAR);
        }
        str1 = strlink(numt1[0],str1,MAXCHAR);
    }
    if(numtlid[4]!= 0){
        str1 = strlink(str1,numt1[4],MAXCHAR);
    }
}

```

```

        i4=strlen(str1);
        str = strndup(str1,i4);
    }

    pass =0;
    foundt = 0;
    rank =listn;
    for(il=0;il<listn;il++){
        foundt = found(caculist2[i1],str);
        if(foundt>0){
            if(il<rank){
                rank = il;
                foundl =foundt;
            }
            nc++;
            pass = 1;
        }
    }
}

listn = 2;
pass = 0;
nc = 0;
foundt = 0;
rank =listn;
for(il=0;il<listn;il++){
    foundt = found(caculist3[i1],str);
    if(foundt>0){
        if(il<rank){
            rank = il;
            foundl =foundt;
        }
        nc++;
        pass = 1;
    }
}
while(pass ==1){
    numbers = 0;
    i3=rank;
    for(i5=0;i5<5;i5++)numt1[i5]='\0';
    arrangstr(caculist3[i3],str);
    if( strcmp(numt1[2],"*") == 0 ){
        if(numt1id[1]==1){
            i4=strlen(numt1[1]);
            cacustrt1 = strndup(numt1[1],i4);
            paral= atof (cacustrt1);
        }
        if(numt1id[3]==1){
            i4=strlen(numt1[3]);
            cacustrt2 = strndup(numt1[3],i4);
            para2= atof(cacustrt2);
        }
        if((numt1id[1]==1)&&(numt1id[3]==1)){
            value = paral * para2;
            numbers = 2;
        }
    }
}
if( strcmp(numt1[2],"/") == 0 ){
    if(numt1id[1]==1){

```

```

        i4=strlen(numt1[1]);
        cacustrt1 = strndup(numt1[1],i4);
        para1= atof (cacustrt1);
    }
    if(numt1id[3]==1){
        i4=strlen(numt1[3]);
        cacustrt2 = strndup(numt1[3],i4);
        para2= atof(cacustrt2);
    }
    if((numt1id[1]==1)&&(numt1id[3]==1)){
        if(fabs(para2)<1e-6)printf("除数不能为零! \n");
        else value = para1 / para2;
        numbers = 2;
    }
}
if(numbers != 0){
    sprintf(values,"%10.6e",value);
    str1 = values;
    if(numt1id[0]!= 0){
        str1 = strlink(numt1[0],str1,MAXCHAR);
    }
    if(numt1id[4]!= 0){
        str1 = strlink(str1,numt1[4],MAXCHAR);
    }
    i4=strlen(str1);
    str = strndup(str1,i4);
}

pass =0;
foundt = 0;
rank =listn;
for(il=0;il<listn;il++){
    foundt = found(caculist3[il],str);
    if(foundt>0){
        if(il<rank){
            rank = il;
            foundl =foundt;
        }
        nc++;
        pass = 1;
    }
}

listn = 2;
pass = 0;
nc = 0;
foundt = 0;
rank =listn;
for(il=0;il<listn;il++){
    foundt = found(caculist4[il],str);
    if(foundt>0){
        if(il<rank){
            rank = il;
            foundl =foundt;
        }
        nc++;
        pass = 1;
    }
}

```

```

}
while(pass ==1){
    numbers = 0;
    i3=rank;
    for(i5=0;i5<5;i5++)numt1[i5]='\0';
    arrangestr(caculist4[i3],str);
    if( strcmp(numt1[2],"+") == 0 ){
        if(numtlid[1]=1){
            i4=strlen(numt1[1]);
            cacustrt1 = strndup(numt1[1],i4);
            paral= atof (cacustrt1);
        }
        if(numtlid[3]=1){
            i4=strlen(numt1[3]);
            cacustrt2 = strndup(numt1[3],i4);
            para2= atof(cacustrt2);
        }
        if((numtlid[1]=1)&&(numtlid[3]=1)){
            value = paral + para2;
            numbers = 2;
        }
    }
    if( strcmp(numt1[2],"-") == 0 ){
        if(numtlid[1]=1){
            i4=strlen(numt1[1]);
            cacustrt1 = strndup(numt1[1],i4);
            paral= atof (cacustrt1);
        }
        if(numtlid[3]=1){
            i4=strlen(numt1[3]);
            cacustrt2 = strndup(numt1[3],i4);
            para2= atof(cacustrt2);
        }
        if((numtlid[1]=1)&&(numtlid[3]=1)){
            value = paral - para2;
            numbers = 2;
        }
    }
}
if(numbers != 0){
    sprintf(values,"%10.6e",value);
    if(numtlid[0]!= 0){
        str1=numt1[0];
        str1 = strlink(str1,values,MAXCHAR);
    }
    else str1=values;
    if(numtlid[4]!= 0){
        str1 = strlink(str1,numt1[4],MAXCHAR);
    }
    i4=strlen(str1);
    str = strndup(str1,i4);
}

pass =0;
foundt = 0;
rank =listn;
for(il=0;il<listn;il++){
    foundt = found(caculist4[il],str);
    if(foundt>0){
        if(il<rank){

```

```

        rank = il;
        foundl =foundt;
    }
    nc++;
    pass = 1;
}
}
}
return str;
}

void
on_schbutton_clicked(GtkButton *button,
                    gpointer user_data)
{
    gint linen=0;
    gchar *schchar;
    GtkTextBuffer *buffer;
    GtkTextIter iterl;
    GtkTextIter start,end;
    gchar *name,*tempfile,ch;
    gint i,il,i2,i3,in,finds;
    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter (buffer,&end);
    name = gtk_text_buffer_get_text(buffer,&start,&end,TRUE);
    schchar = gtk_entry_get_text (GTK_ENTRY(schentry));
    i=i2=i3=in=finds=0;
    il=strlen(schchar);
    while(ch=name[i]){
        i3++;
        if(ch==schchar[i2]){
            if(i2==il-1){
                finds=1;
                if(schloc!= NULL){
                    if(i>schloc)break;
                }
                else break;
            }
            i2++;
        }
        else {
            i2=0;
            if(name[i]=='\n'){
                in++;
                i3=0;
            }
        }
        i++;
    }
    linen=in;
    if(finds==1){
        gtk_text_buffer_get_iter_at_line_index(buffer,&iterl,linen,i3);
        gtk_text_buffer_place_cursor(buffer,&iterl);
    }
    schloc = i;
    if(schloc==strlen(name))schloc=0;
}
void

```

```

on_dwbutton_clicked                                (GtkButton      *button,
                                                    gpointer        user_data)
{
    gint linen=0;
    gchar *schchar;
    GtkTextBuffer *buffer;
    GtkTextIter iterl;
    GtkTextMark *markl;

    schchar = gtk_entry_get_text (GTK_ENTRY(schentry));
    linen=atoi(schchar);
    if(linen!=0){
        buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
        gtk_text_buffer_get_iter_at_line(buffer,&iterl,linen);
        gtk_text_buffer_place_cursor(buffer,&iterl);
        markl=gtk_text_buffer_create_mark(buffer,"jack",&iterl,FALSE);//定义光标
        gtk_text_view_scroll_to_mark(GTK_TEXT_VIEW(textview2),markl,0.0,FALSE,0.0,0.0);//文本
视图窗口滚动到光标处
    }
}

void
on_schbutton_clicked                                (GtkButton      *button,
                                                    gpointer        user_data)
{
    gint linen=0;
    gchar *schchar;
    GtkTextBuffer *buffer;
    GtkTextIter iterl;
    GtkTextIter start,end;
    gchar *name,*tempfile,ch;
    gint i,il,i2,i3,in,finds;
    GtkTextMark *markl;

    buffer = gtk_text_view_get_buffer(GTK_TEXT_VIEW(textview2));
    gtk_text_buffer_get_start_iter (buffer,&start);
    gtk_text_buffer_get_end_iter   (buffer,&end);
    name = gtk_text_buffer_get_text(buffer,&start,&end,TRUE);
    schchar = gtk_entry_get_text (GTK_ENTRY(schentry));
    i=i2=i3=in=finds=0;
    il=strlen(schchar);
    while(ch=name[i]){
        i3++;
        if(ch==schchar[i2]){
            if(i2==il-1){
                finds=l;
                if(schloc!= NULL){
                    if(i>schloc)break;
                }
                else break;
            }
            i2++;
        }
        else {
            i2=0;
            if(name[i]=='\n'){
                in++;
            }
        }
    }
}

```

```
        i3=0;
    }
    }
    i++;
}
linen=in;
if(finds==1){
    gtk_text_buffer_get_iter_at_line_index(buffer,&iter1,linen,i3);
    gtk_text_buffer_place_cursor(buffer,&iter1);
    mark1=gtk_text_buffer_create_mark(buffer,"jack",&iter1,FALSE);
    gtk_text_view_scroll_to_mark(GTK_TEXT_VIEW(textview2),mark1,0.0,FALSE,0.0,0.0);
}
schloc = i;
if(schloc==strlen(name))schloc=0;
}
```