第30卷 第18期 计 算 机 工 程 2004年9月
Vol.30 № 18 Computer Engineering September 2004

·开发研究与设计技术· 文章编号：1000—3428(2004)18 —0179—03 文献标识码：A 中图分类号：TP311.52

基于QNX操作系统的EMPRESS数据库图形用户界面软件包

徐灵哲，徐欣圻

(中国科学院国家天文台南京天文光学技术研究所, 南京 210042)

摘 要：介绍了“大天区面积多目标光纤光谱天文望远镜”控制系统中的EMPRESS数据库图形用户界面软件包的设计思想和实用技巧。研制中采用了动态生成控件、动态查询和共享内存等多项先进技术，实现了EMPRESS数据库和QNX操作系统环境下的图形开发工具PHAB的无缝连接，在类Unix操作系统下实现了Windows的界面风格，给用户提供了友好的动态生成数据库表格的强大功能。

关键词：天文望远镜；动态查询；共享内存；动态生成；QNX

EMPRESS Database GUI Software Package Based on QNX OS

XU Lingzhe, XU Xinqi

(Nanjing Institute of Astronomical Optics & Technology, National Astronomical Observatories ,Chinese Academy of Sciences, Nanjing 210042)

【**Abstract**】The methodology and technique tips in designing the EMPRESS database GUI software package are presented. The package is developed for the control system of the large sky area multi-object fiber spectroscopic telescope. A number of advanced techniques are adopted in the design such as dynamic creation of control widgets, dynamic query and share memory. The seamless connection between EMPRESS and the graphical development tool of QNX's PHAB is realized, and so have the Windows look and feel yet under Unix-like operating system. What is more, the package provides the user with a friendly and powerful function of dynamic creation of database tables.

【**Key words**】Astronomical telescope; Dynamic query; Share memory;Dynamic creation; QNX

“大天区面积多目标光纤光谱天文望远镜”(Large sky Area Multi-Object fiber Spectroscopic Telescope,LAMOST)是我国“九五”重大科学工程项目，建成后将成为世界上4m级口径以上的光学望远镜中的视场之最和光谱观测效率最高的望远镜。我们在望远镜总控系统上配置了当前国际上公认的优秀实时操作系统之一QNX，并进而开发实时分布式数据库。该项研制与国际上当代天文望远镜接轨，在国内天文望远镜的使用上尚属首次。数据库采用的是EMPRESS。可是，EMPRESS并没有提供图形化开发工具，而QNX系统下的图形化开发工具PHAB(Photon Application Build)提供的功能又比较简单。另一方面，一个不争的事实是，由于Windows系统的流行，人们已经习惯于使用大量的友好的图形化界面进行数据分析。更重要的是，考虑到该数据库的用户大都是天文学家。一般说来他们只对具体的数据表格和图形感兴趣，不一定十分熟悉如何从数据库中提取出这些表格和图形。为此，我们孜孜以求的是基于QNX(类Unix)系统开发具有Windows操作风格的数据库软件包，尤其是提供表格和图形的动态生成功能，给用户一个十分友好的环境。经过努力，借助一系列编程技巧，我们采用13 000条C语句完成了开发任务。该数据库图形用户界面软件包于2002年7月2日通过了LAMOST工程指挥部组织的北京、南京和合肥三地专家评审会的评审。研制中采用的动态生成控件、动态查询、共享内存等手段，在类Unix操作系统下实现了Windows的界面风格，以及给用户提供了友好的动态生成数据库表格的强大功能等给专家们留下了深刻的印象。

1 系统平台

4台微机，1台COMPAQ AP500和3台NEC PIII500，组成一个局域网模拟系统开发平台。每台机上都装有ONX实时操作系统，版本为QNX4.25 。其中在一台NEC机上安装了EMPRESS v8.60 测试版，作为数据库服务器。

QNX系统是加拿大QSSL公司开发的分布式实时操作系统，是微内核的先驱。EMPRESS数据库开发包是少数几个能支持QNX4.25的软件包中的一种。它支持的是标准的分布式关系型数据库。在QNX下EMPRESS能利用QNX的Fleet协议，而对于其它操作系统则提供ODBC接口。LAMOST 实时分布式数据库平台就是在QNX 环境下再利用EMPRESS软件包(测试版)研制和开发出来的。

2 程序风格

本程序在类Unix系统的QNX下实现了标准的Windows风格。例如，支持鼠标、菜单和快捷键操作、支持拖拽操作、支持多种分辨率以及自动生成统计数据图形功能等。

3 程序结构

为了实现EMPRESS与QNX操作系统下的PHAB 的无缝连接， 本程序采用了如图1所示的结构。图1中界面进程的开发是借助于PHAB用C语言编制的。服务进程是用QNX shell 下的WATCOM C 开发的。在每一个节点上，运行一个服务进程。这个服务进程可以为该节点上多个界面进程服务。界面进程接收用户的指令，服务进程收到界面进程发来的指令后和数据库交互，然后把提取的数据放入共享内存。界面进程从共享内存中取出数据反馈给用户。

基金项目：国家“九五”重大科学工程项目

作者简介：徐灵哲(1975－)，男，高级程序员，主研方向：数据库设计与应用，分布式系统和网络通信；徐欣圻，博导

收稿日期：2003-08-05 E-mail：xqxu@nairc.ac.cn

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

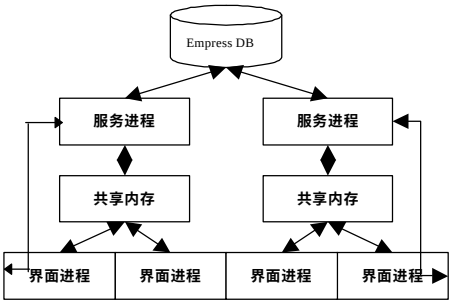

图1 数据库平台的系统结构

4 程序的主要功能

建表、修改表、删除表、建立索引、删除索引、运行SQL语句、选择列条件产生表格、调整表格的列、调整表格的字体、调整表格的颜色、调整表格的大小、调整表格的宽度、选择查询条件、选择查询顺序、选择列条件产生图形、设置图形的横坐标和纵坐标、设置图形的题头、设置图形的缩放、插入数据和删除数据。这些功能基本上实现了常用的DBMS的功能。用户还可以利用程序提供的GUI接口，方便地建立自己的表格(包括动态生成数据库表格)、图形，并加以保存，为数据的分析提供了强有力的手段。

5 技术实现

5.1 数据库服务层的实现

数据库服务层负责接收界面进程发来的消息，对其进行处理，然后同数据库交互。服务程序读取消息中的SQL指令，根据指令中的关键字来判断SQL指令的类型。对于执行类型的SQL语句，例如插入、删除、修改等，由服务程序直接运行并得到状态量返回给界面进程；对于查询类型的SQL指令，则采用SQL语句中的动态SQL技术来实现。由于在查询之前无法确切知道所要查询的列的数量和列的类型，只有通过动态SQL语句来实现。动态SQL语句作了如下一个申明。通过该申明可以实现不同类型数据的读取。

```
EXEC SQL BEGIN DECLARE SECTION;
char sql_str[1024]; /*string to contain SQL    */
  char dname[33][25]; /*for storing item names    */
  int dtype[25]; /* for storing data type of items*/
  int da_num; /* number of active descriptor areas */
  int area;  /* descriptor item area              */
  char *pstr; /* for getting string data           */
  long vlong; /* for getting integer data          */
  double vdouble;/*for getting float data          */
  int    i;
  typedef struct{
     long size;
     char data[1];
  } bulk;
  bulk *pbulk;/*for getting bulk data              */
  short ctrl;/* control variable for null checking */
EXEC SQL END DECLARE SECTION;
```

5.2 进程间的通信

在本系统中牵涉到通信部分的有界面进程和服务进程的通信，服务进程和共享内存的交互，界面进程和共享内存的交互。在开发中，采用了同步通信和异步通信相结合的方式完成了数据的交换。

(1)消息通信

界面进程和服务进程的通信采用消息的方式，通过Send 和 Receive 函数来完成。在QNX中，一个消息是指被打包的若干个字节，它被用来从一个进程同步地传送到另一个进程。图2揭示了进程A向进程B发送一条消息，进程B接收该消息进行处理并给予回答的通信过程。该过程从时间先后关系可分为3步：1)进程A通过向内核发出Send()请求向进程B发送一条消息。此时进程A变成Send阻塞状态，直到进程B发出Receive()接收这一消息。2)进程A转为Reply阻塞状态。3)进程B对接收到的消息进行处理，然后调用Reply()回答。回答消息被拷贝到进程A，使得进程A进入到准备运行状态。此时由进程A和进程B的优先级决定哪个进程将被调入运行。传递过程如图2。

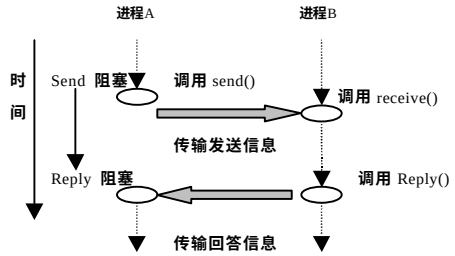

图2 进程A和进程B之间的消息通信

(2)共享内存

服务进程从数据库中得到的数据是通过共享内存实现传输的。以下对这种通信方式做进一步的分析。

共享内存区是最快的可用进程间通信形式。一旦这样的内存区映射到共享它的进程空间，这些进程间数据的传递就不再涉及内核。然而往该共享内存区存放信息或从中取走信息的进程间通常需要某种形式的同步。例如互斥锁、条件变量、读写锁、记录锁、信号灯等。在本系统开发过程中，我们采用的是信号灯的方式。利用这种方式对共享内存的操作一般如下：服务器使用一个信号灯取得访问某个共享内存区对象的权力。服务器映射一个地址指向该共享内存，然后读入数据到该地址。服务器读入完毕时，使用一个信号灯通知客户。客户从该共享内存对象写出这些数据到输出文件中。

本程序中采用的办法是把一大块共享内存分割成N个小块，然后设置两个信号量，即read信号量和write信号量来控制对每个小块的读写。read信号量初始化为0，而write信号量初始化为N。服务进程每写一小块内存，write = write-1，read = read + 1，指针指向下一小块内存。界面进程每读一小块内存 write = write + 1， read = read - 1。指针指向下一小块内存。当服务进程写完第N小块后，若write > 0 则指针又指向第一块内存。当界面进程读完第N小块后，若read>0则指针又指向第一块内存。当write ＝ 0时服务进程阻塞，当read＝0时界面进程阻塞。通过这种方式实现了内存的互斥，提高了系统的吞吐率，避免了内存的溢出。

5.3 界面程序的编写

(1)动态控件和组合控件的使用

应用界面是利用QNX环境下的PHAB来完成的。由于它的功能简单，对于编写比较复杂的图形化程序必须借助一系列编程技巧来实现。

在开发过程中，由于程序灵活性的需要和开发工具本身限制，我们大量使用了动态控件来组合成较复杂的控件。例如，动态生成的表格就是由PtList、 PtDivider、 PtButton 3种控件组合生成的。其中每一个列头都是一个PtButton控件。由于用户语法生成的不确定性,因此整个表格都是由动态控

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

件生成的。

在程序开发中，采用的是创建控件仓库的方式来动态调用控件。对一个要多次动态调用的控件，最好的方式无疑是采用控件仓库。控件仓库提供一次生成、多次调用的功能，同时还能事先生成控件的回调函数。调用之后，立即生效。

以下介绍其中的主要的函数：

ApCreateWidget函数用于控件仓库中选一个控件进行调用

PtWidget_t ApCreateWidget(ApDBase_t const db,char const wgt_name,int x,int y,int nargs,ptArg_t const *agrs)

其中，db为控件仓库名，wgt_name为要调用的控件名，x、y为控件的初始X、Y坐标；nargs为初始化参数的数量，args为初始化参数。

(2)动态控件的管理

当动态控件被创建出来后，它就要占用系统资源。而当调用它的模板被关闭后也相应地需要程序去注销这些控件，以释放资源。本程序通过链表的方式来管理动态控件。

链表是一种可以实现动态分配的存储结构，它不需要一组地址连续的存储单元，而是用一组任意的，甚至是在存储空间中零散分布的存储单元来存放线形表数据。同时在进行元素插入和删除时，链表也避免了像向量那样进行约一半的元素移动，从而克服了向量存储结构的缺点。

在本程序的开发中，大量使用了链表这种数据结构。以下就是用来存储报表中一列的数据结构。

```
struct ST_DESIGN{
PtWidget_t widget ;/*the handle of widget*/
Char item_name[NAMESIZE];/*the attrib of the table*/
char text[NAMESIZE];/*the name of the attrib*/
PhArea_t area;/*the size of the widget*/
PgColor_t font_color;/*the color of the widget*/
PgColor_t fill_color;/*the fill color of the widget*/
char font[NAMESIZE];/*the font of the widget*/
struct ST_DESIGN next;/*the point of the list*/
};
```

(3)数据结构的存储

当用户完成对报表或图表的设计、对其进行保存时，就涉及到采用何种方式进行保存。在本程序的开发中，我们采用了流式文件的存储方式来进行存储。这种方式的优点在于，它能直接对数据结构进行存取，不需要用户再做其它的转换。下面介绍用到的几个主要函数。

1)Fopen 函数打开一个流

FILE *fopen(char *filename, char *type);

2)fread函数从一个流中读数据

int fread(void *ptr, int size, int nitems, FILE *stream);

3)fwrite函数写内容到流中

int fwrite(void *ptr, int size, int nitems, FILE *stream);

4)fclose函数关闭一个流

int fclose(FILE *stream);

其中，filename为文件名，type为文件操作方式，ptr为缓存指针，size为缓存大小，nitems 为读或写记录的个数，stream为流式文件指针。

6 结束语

这项工作在我国天文望远镜控制系统上尚属首次。当然，该软件包除了可以满足LAMOST 控制系统数据在线(或离线)处理、存储、程序生成表格和图形等实际需要外，也可推广应用到其它相关领域。

充分利用该软件包自动生成数据库表格的功能，针对LAMOST控制系统在天体观测应用中的实际需要，我们已经生成了包括"机架指向和跟踪表"、"主动光学控制表"、"圆顶控制表"、"网络状态表"、"联机帮助表"和"诊断表"等表格并保存起来。对于这些表格用户可以拿来就用，当然根据需要作适当的修正或者利用其直观的图形界面完全创建新表也不是一件难事。

参考文献

1 徐灵哲，徐欣圻.LAMOST望远镜总控系统上采用分布式数据库前景评估. LAMOST 工程部内部技术报告，2001-09

2 徐欣圻.LAMOST望远镜控制系统.天体物理学报,2000,20(增刊)

3 徐欣圻.QNX实时操作系统及其在大型天文望远镜上应用前景初探.天文仪器与技术，1999

4 侯业勤,张 菁.分布式嵌入式实时操作系统QNX.北京:宇航出版社,1999-01

5 Stevens W R. Unix网络编程.北京:电子工业出版社,2000-03

☆☆

(上接第175页)

```
Public Property Set DataSource(ByVal New_DataSource As Recordset)
Set m_DataSource = New_DataSource
PropertyChanged "DataSource" '通知容器DataSource属性值已变
'为与表中字段内容同步，调用RefreshText方法来更新竖向编辑
'控件中的文本
  If Not m_DataSource.EOF And Not m_DataSource.BOF Then
    If Trim(m_DataField) <> "" Then RefreshText m_DataSource(m_DataField) & ""
  End If
End Property
'在记录集对象的MoveComplete事件过程中更新竖向编辑控件
'的文本
Private Sub m_DataSource_MoveComplete(ByVal adReason As ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  If Not pRecordset.EOF And Not pRecordset.BOF Then
    If Trim(m_DataField) <> "" Then RefreshText pRecordset(m_DataField) & ""
  Else
    If Trim(m_DataField) <> "" Then RefreshText ""
  End If
End Sub
```

3 结束语

本文详细探讨了实现竖向编辑ActiveX控件的关键技术，解决了Web模式软件研发中蒙古文的垂直输入和显示等问题。目前，作者开发完成的多种竖向编辑控件已经应用到蒙古文多媒体课件、蒙古文档案管理系统等应用项目中，取得了较好的效果，还可以进一步推广到其他蒙古文应用软件项目中，促进和提高蒙古文应用软件的研发技术和水平，更好地为内蒙古地区的经济发展服务。

参考文献

1 赵良叶,齐剑锋.Visuall C++ 5 使用大全.北京:电子工业出版社,1998

2 Slier B,Spotts J.Visual Basic 6 开发使用手册.北京:机械工业出版社,1999

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net