

基于宽度优先搜索的菜单遍历算法在 C# 中的实现及应用

Implementation and Application of a width - first search based method of visiting menu in C#

娄七明 许海成 王俊梅 杨宽德 (红河学院工学院 云南蒙自 661100)

摘要:在 MIS 系统应用中,通常要对用户权限进行控制。本文首先分析了两种常用的访问控制方法的缺陷;然后提出了一种基于宽度优先搜索遍历菜单,实现用户权限控制的方法;最后结合实例用 C# 实现了该方法。该方法具有很好的通用性和实用性。

关键词:宽度优先搜索 队列 树 MIS

1 引言

在 MIS 系统中,为了保证系统的安全性,必须对用户权限进行控制。目前,最常用的方法是在程序中根据用户的身份进行权限逻辑判定^[1]。然而,这种方法需要预先设计好权限分配逻辑,并在编程时实现,若某用户的权限需要改变,必须修改应用程序并重新编译。另一种改进方法是,将权限信息存储在数据库中,在需要动态改变用户权限时,通过更改存储的权限信息来实现权限控制^[2]。但是,若系统需求发生变化,增加或删除某些应用程序的功能时,需要手工更改数据库中的权限信息,还可能需要对程序中的权限判断进行修改。如果能遍历应用程序的菜单,把遍历结果存入数据表中,然后再根据菜单分配用户权限,就能解决上述方法存在的缺陷,实现真正意义上的用户权限动态控制。本文采用宽度优先搜索算法实现了 C# 菜单的遍历,并把遍历结果存入数据表中,然后根据用户权限设置菜单项的 Enabled 属性,实现用户权限的动态分配。该方法具有三大特点:①以类的形式提供,通用性好;②采用宽度优先搜索算法实现,菜单的层次关系清晰;③不受菜单级数的限制,菜单级数可以达到 N 级。

2 算法分析

遍历菜单就是按一定的顺序访问每一个菜单项,并使每个菜单项只被访问一次。C# 菜单可以看成是一棵 N 叉树,图 1 是图书馆 MIS 的菜单结构。使用宽度优先搜索算法遍历该菜单的步骤如下:

① 定义两个队列 q1 和 q2, q1 用来存放访问过的菜单项, q2 用来存放访问过且包含子菜单的菜单项,以便对这些菜单项进行宽度优先遍历;

② 顺序遍历 MainMenu1 菜单的子菜单项,把基础信息管理、读者管理、图书管理、图书流通管理、统计分析管理、系统管理入队,结果如图 2 所示;

③ q2 队首元素出队,并遍历之,结果如图 3 所示;

④ 重复③,直到队列 q2 为空,遍历结束。这时队列 q1 中元素从队首到队尾依次是:基础信息管理、读者管理、图书管理、图书流通管理、统计分析管理、系统管理、图书馆信息、系统参数设置、书架设置、管理员设置、读者类型设置、读者档案管理、借阅证管理、图书借阅、图书归还、图书续借、图书丢失、图书罚款、图书借阅统计、图书分类统计、用户设置、权限设置、数据备份、数据恢复、办证、挂失与恢复。

3 C#实现链队及其基本操作

由上面的分析知,使用宽度优先搜索算法遍历菜

链式队列的每个节点由两个域构成,其中 data 存放菜单项,而 next 指向下一个节点。类定义如下,其中 Cnode() 是构造函数,用于初始化节点的 next,使其

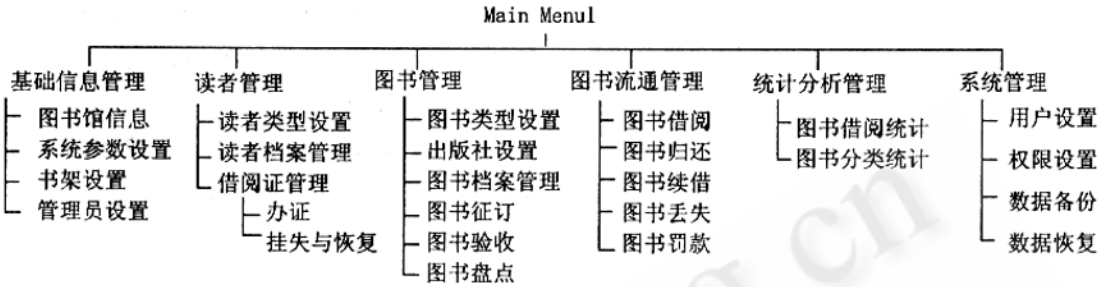


图 1 图书管理系统菜单结构

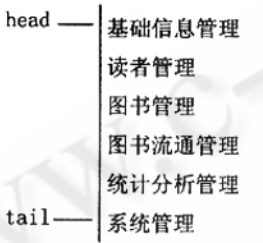


图 2 第②步后 q1、q2 的状态

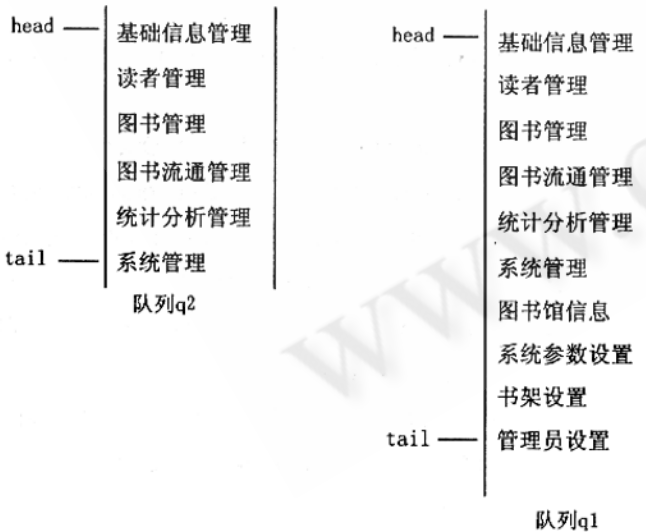


图 3 第③步后 q1、q2 的状态

单,必须使用队列,下面给出 C#实现的链式队列。

3.1 定义节点类

值为 null。

```

public class CNode
{
    public MenuItem data; public CNode next;
    public CNode() { next = null; }
}
    
```

3.2 定义链式队列及其基本操作

链式队列有两个域,其中 head 是队首指针,tail 是队尾指针。链式队列类定义如下,其中 QueueCnode() 是构造函数,实现队列的初始化;方法 Insert 实现入队操作,把 data 入队,并作为队首元素;方法 Remove 实现出队操作,把队首元素出队,并把下一个元素作为队首元素;属性 IsEmpty 判断队列是否为空。

```

public class QueueCnode
{
    private CNode head, tail;
    public QueueCnode() { head = tail = null; }
    public void Insert(MenuItem data) //入队
    {
        if( head == null)
            { head = new CNode(); head. data = data;
              tail = head; }
        else
            { tail. next = new CNode(); tail. next. data =
              data; tail = tail. next; }
    }
}
    
```

```

}
public void Remove(ref CNode ch)//出队
{ ch = head;
  if( head == null)
  { }
  else if( head == tail)
  { head = tail = null; }
  else
  { head = head. next; }
}
public bool IsEmpty//判断队列是否为空
{ get { if ( head == null ) return true; else return
false; } }
}

```

4 基于宽度优先搜索的 C#菜单遍历算法

4.1 创建表存放菜单项^[3]

为了实现 MIS 系统中用户权限的动态控制,需要建立三个数据表 tabmenu、tabuser、tabqx,分别存放菜单信息、用户信息、用户权限信息。因篇幅有限,这里不对表的设计作介绍,只给出创建表的语句,如下:

```

create table tabmenu( menunum int primary key,
menutext varchar(40) not null)
create table tabuser( userid int primary key, user-
name char(8) not null,pwd char(10) not null)
create table tabqx ( userid int references tabuser
(userid), menunum int references tabmenu ( menu-
num), qx bit not null,constraint c1 primary key ( userid,
menunum) )

```

4.2 算法实现

为提高算法的通用性,定义如下的菜单宽度优先搜索类 searchmenu。其中,mainmenu 是要遍历的菜单,constr 存放数据库的连接参数,q1 存放遍历结果的队列,q2 存放具有子菜单的菜单项的队列,通过构造函数 searchmenu 初始化;方法 Enter 实现遍历过的菜单项入队;方法 Search 实现菜单的宽度优先搜索遍历;方法 WriteData 实现遍历结果存入表 tabmenu;方法 Setqx 根据用户的操作权限设置菜单项的 Enabled 属性。

```
public class searchmenu
```

```

{
  private MainMenu mainmenu;private string constr;pri-
  vate QueueCnode q1,q2;
  public searchmenu( MainMenu menu,string str)
  { mainmenu = menu; constr = str; q1 = new QueueC-
  node(); q2 = new QueueCnode(); }
  private void Enter( MenuItem m)
  { if( m. MenuItem. Count > 0 ) { q2. Insert( m); q1. In-
  sert( m); } else { q1. Insert( m); } }
  public void Search( )
  {
    CNode node1 = new CNode();
    foreach ( MenuItem m in mainmenu. MenuItem)
    { Enter( m); }
    while( q2. IsEmpty == false)
    {
      q2. Remove( ref node1);
      foreach ( MenuItem n in node1. data. MenuItem)
      { Enter( n); }
    }
    public void WriteData ( )//把遍历结果写入表 tab-
    menu
    {
      int num = 0; SqlConnection con = new ( SqlConnec-
      tion);
      con. ConnectionString = constr; con. Open();
      SqlCommand insertcmd = con. CreateCommand();
      insertcmd. CommandText = " insert into tabmenu
      values( @ menunum, @ menutext)";
      insertcmd. Parameters. Add( "@ menunum", SqlDbType. Int,4," menunum" );
      insertcmd. Parameters. Add( "@ menutext", SqlDbType. VarChar, 40," menutext" ); CNode node1 =
      new CNode();
      while( q1. IsEmpty == false)
      {
        q1. Remove( ref node1); num = num + 1; insertc-
        md. Parameters[0]. Value = num;
        insertcmd. Parameters[1]. Value = node1. data.

```

```

Text;
    insertcmd. ExecuteNonQuery();
}
con. Close();
}
public void Setqx(int user)//读 tabqx 表,并根据用户权限设置菜单的 Enabled 属性
{
    int num = 1; bool enableit = false; SqlDataReader datareader;
    SqlConnection con = new SqlConnection();
    con. ConnectionString = constr; con. Open();
    SqlCommand sltcmd = con. CreateCommand();
    sltcmd. CommandText = " select qx from tabqx where userid = @ user and menunum = @ num";
    sltcmd. Parameters. Add( " @ user", SqlDbType. Int, 4, "userid" );
    sltcmd. Parameters. Add( " @ num", SqlDbType. Int, 4, "menunum" );
    CNode node1 = new CNode();
    while( q1. IsEmpty == false)
    {
        sltcmd. Parameters[0]. Value = user; sltcmd. Parameters[1]. Value = num;
        datareader = sltcmd. ExecuteReader( CommandBehavior. SingleRow );
        datareader. Read(); enableit = ( bool ) datareader["qx"];
        datareader. Close(); q1. Remove( ref node1 );
        node1. data. Enabled = enableit; num = num + 1;
    }
    con. Close();
}
}

```

5 在权限设置中的应用

设图书馆管理系统的主窗体是 FormMain,该窗体

的菜单 MainMenu1 如图 1 所示,系统数据库的连接参数存放在 constring 中,当前用户的编号存放在 user 中。要在窗体的 Load 事件中使用宽度优先搜索算法遍历菜单 MainMenu1,并根据用户权限设置菜单的 Enabled 属性,实现权限控制,只需要在 FormMain 的 Load 事件中定义 searchmenu 类的对象 smenu,然后根据需要调用对象 smenu 的 search 方法、WriteData 方法和 Setqx 方法即可,代码如下:

```

private void FormMain_Load( object sender, System. EventArgs e)
{
    searchmenu smenu = new searchmenu( mainMenu1 ); smenu. Search();
    if( isfirst == true ) smenu. WriteData(); else Smenu. Setqx( user );
}

```

其中, isfirst 标识系统是否是第一次运行,若是,把遍历结果存入 tabmenu 中,以便进入系统进行用户权限设置;否则,读取用户 user 的权限信息,并根据用户权限设置菜单项的 Enabled 属性。该参数可以存放在数据库中,也可以存放在系统配置文件中。

6 结束语

本文针对 MIS 系统开发过程中用户权限的设置问题,采用面向对象的方法,使用 C# 实现了菜单的宽度优先搜索算法,该算法通用性和实用性极佳,有推广应用价值。

参考文献

- 1 张奇、李律松、卫建伟等, Visual C# 数据库项目案例导航[M], 北京:清华大学出版社, 2005.
- 2 张建英、王秀坤, 一种面向应用的基于访问权限的动态控制方法[J], 计算机工程与应用, 2003(07): 120.
- 3 熊小敏、周燕玲, PowerBuilder 下实现菜单级别的权限控制[J], 计算机与现代化, 2003(12): 90.