

```

STD = stc
ELSE      &&既不是左也不是右括号时
IF stc > STD  &&输入项优先级较高时
    DO PUSH
    STD = stc
ELSE      &&输入项优先级较低时
    sth = stb  &&存当前输入项
    stg = stc
    DO WHILE stg < = STD .AND. .NOT. stkempty (top2,
top1)
        DO TOP
        DO POP
        logical = .T.
        DO PUSH
        logical = .F.
        IF .NOT. stkempty(top2,top1)
            DO TOP
            stb = stelm
            DO yxj
            STD = stc
            ENDDIF
        ENDDO
        stelm = sth  &&恢复当前输入项
        DO PUSH
        ENDDIF
        ENDDIF
        ENDDIF
        ENDDO
        logical = .F.  &&栈1中运算符后进先出压入栈2
        DO WHILE .NOT. stkempty(top2,top1)
            DO TOP
            DO POP
            logical = .T.
            DO PUSH
            logical = .F.
        ENDDO
        RETURN nbljs()  &&转换结束
        PROCEDURE yxj  &&确定优先级过程
        stl = .F.  &&输入项为运算符
        DO CASE
        CASE stb = '('
            stc = 0
        CASE stb = ')'
            stl = .F.
        CASE stb = '+' .OR. stb = '-'
            stc = 1
        CASE stb = '*' .OR. stb = '/'
            stc = 2
        CASE stb = '&' &&记录连接符
            stc = 3
        OTHERWISE

```

```

        stl = .T.  &&输入项为操作数或记录号
    ENDCASE
    RETURN

```

提高 FoxPRO 软件运行速度的若干措施

杜学绘 张 斌 (郑州电子技术学院)

摘要:在信息管理系统(MIS)应用开发中,提高应用软件运行速度至关重要,本文介绍在 NOVELL 网络环境下提高 FoxPRO 应用软件运行速度的几种措施。

FoxPRO 网络数据库在过去几年有了长足的发展,用其编制的应用软件越来越多,也越来越趋大型化。由客户、服务器方式构成的数据库网络在当前十分流行,客户、服务器结构把 DBMS 和应用分开,使网络中某些计算机专门用于执行 DBMS 的功能,这些计算机称作数据库服务器,而其他结点上的计算机安装 DBMS 的外用开发工具(如 FoxPRO 等),支持用户的应用,这些计算机称为客户机。从而构成了客户/服务器方式的数据库网络。

客户/服务器结构的优点在于将微机工作站和小型机、大型机上的 DBMS 的优势有机结合起来,把文件服务器的观点向前推进了一大步,客户机承担应用开发工具和支持应用程序运行的任务,同时它还可以通过网络获得服务器的服务,使用服务器的共享资源,这种把处理任务一分为二的做法,克服了文件服务器的弱点,使用户接口、事务处理、数据管理等项工作统一规划,保障了应用的相对独立性,提高了用户对基础硬件、软件更新换代的自由度。

因此,在客户/服务器方式下开发 FoxPRO 网络数据库必然成为大势所趋,为了满足 FoxPRO 网络数据库大型化趋势对软件的运行速度的要求,需要采用一系列措施,现介绍如下。

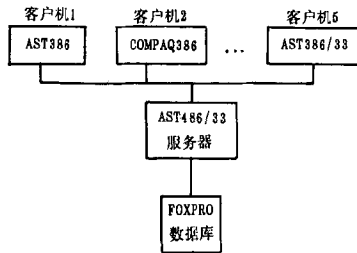
一、系统配置

1. 硬件环境

一台 486/33 微机作服务器,5 台 386 微机作客户机,NOVELL 网络(NETWARE3.31),DOS6.0。其结构如图所示:

2. 内存管理

在 MIS 应用开发中,应用软件的大型化趋势要求使用更多的内存空间,所以应该最优化地划分不同类型内存的区域,以达到提供给用户尽可能的内存空间。



(1)使用 HIMEM.SYS 扩充内存管理程序。它负责存取扩充内存,并确保不同的程序不会同时使用相同的扩充内存空间。使用 HIMEM.SYS 具有如下优点:避免不同程序要求内存而发生冲突时所引起的错误,允许其他按 XMS(扩充内存规范)编制的扩充内存管理程序使用未被 HIMEM.SYS 占用的扩充内存,使用户在扩充内存的高内存区(HMA)中运行 MS-DOS6.0,以节省常规内存。在 80386 与 80486 计算机中,它与 EMM.386 合用,可使用户能够使用高端内存区(UMB),从而节省常规内存,它在处理 640K 以上的内存方面起着关键作用。

(2)安装 EMM386.EXE。EMM386.EXE 的功能在于将 386 或 486 的扩充内存(XMS)模拟成数据内存(EMS),并支持保留内存仿真转换为 EMS,将扩展内存映射到上位内存,使 600K 至 1MB 的空白区域转变为 UMB。参数 NOEMS 指为 EMM386.EXE 提供 UMB 支持,但不为用户程序提供仿真 EMS 扩展内存服务,参数 RAM 表示提供 UMB 支持,同进为需要 EMS 的应用程序提供 EMS 内存。

(3)在扩充内存中安装 MS-DOS6.0。使 MS-DOS6.0 在常规内存和 UMB 之间保证一种连接。据此,在工作站(客户机)本地硬盘中,CONFIG.SYS 内容如下:

```

DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
DOS=HIGH,UMB
  
```

(4)用 LOADHIGH 命令将网络驱动程序载入 UMB。为将网络驱动程序载入 UMB,在工作站建立如下的自动批处理文件 AUTOEXEC.BAT。

```

@ECHO OFF
PROMPT $P$G
LH IPX
LH NETX
F:
LOGIN
  
```

这样优化的结果,使基本内存占用量减少,剩余可用量增加,而其高端内存块与之相反,占用量增加,剩余量减少,为应用软件节省了尽可能多的基本内存,同时也提高了用 FoxPRO 编写的网络数据库软件的运行速度。

二、全屏幕编辑程序

FoxPRO 已经提供了 BROWSE 全屏幕编辑命令,我们为什么还要编写全屏幕编辑程序呢?主要基于两点理由:一是全屏幕编辑不仅仅只是查阅,经常还要做插入、删除等操作,这时 BROWSE 中必须与 ON KEY LABEL 合用才能实现这两种操作,当数据库中记录达到一定数量时,BROWSE 的速度特别慢,而用全屏幕编辑程序就快捷得多。二是 BROWSE 能显示库中各项字段和内容。如果字段名是西文字符,则显示结果用户不容易看出来是什么意思,这时就需要用一个函数来作一个西文字符到汉字的转换,而这样一来其查询速度非常慢,对库中比较多的记录,显示等待的时间比较长。相反,若用全屏幕编辑程序查询,不仅屏幕形式可以设计得比较灵活、美观,而且查询速度有明显提高,几乎不需要有等待时间,其方法如下。

先建立一个模式库 ZBK.DBF,存放着屏幕显示信息。用 INKEY(O)给用户设置一个陷阱,等待用户按下一个键(如→←),即可通过 ZBK.DBF 对逻辑数据库进行全屏幕编辑。

ZBK.DBF 的库结构如下:

LINE 显示字段的行坐标

COL 显示字段的列坐标

NAME 需要处理的数据库的字段名

NAME_C 存储具体数据库字段的中文或西文说明

LEN 为数据库该字段作为横向列表中的一栏的宽度,其目的是为了列表画线时控制&NAME 字段栏目横向的最大宽度。LEN1 = MAX(IIF

(&NAME 为字典代码,LEN(字典代码所对应词组),LEN(&NAME))
,ALLTRIM(NAME))

其程序的框架如下:

```

USE XBK
GO TOP
C_NO=LINE
C_LN=COL
INK=0
DO WHILE.T.
  DO CASE
    CASE INK = 19
      DO LEFT WITH C_NO,C_LN &&左移处理
    CASE INK = 4
      DO RIGHT WITH C_NO,C_LN &&右移处理
    CASE INK = 5
      DO UP WITH C_NO,C_LN &&上移处理
    CASE INK = 24
      DO DOWN WITH C_NO,C_LN &&下移处理
    CASE INK = 27
      EXIT
  ENDCASE
  IF INK > 48 AND INK < 58
    KEYBOARD INK
    IF RLOCK()
      @C_NO,C_NO GET &NAME
      READ
      UNLOCK
    ENDIF
  ENDDO

```

这种通用的全屏幕编辑程序我们称为抽象程序,它是可以重复使用的,在具体应用中不必改变,而坐标库(ZBK.DBF)是可变的,坐标库的内容不同,程序的运行结果也就不同,它的优点就是:生成的系统运行速度快,适应性强,易于修改、扩充,软件维护方便。

三、如何提高查询、统计速度

在一般的数据库管理中,经常需要软件能自动统计出结果,如年报表、季报表,或分社向总社汇总,希望得到汇总表,然而当库中记录较多时,速度方面往往不如人意。笔者采用拷贝临时库的办法,效果甚好,现介绍如下。

一般,分社向总社汇总表,就需把多个分社的报表合计以后生成总社报表,而每期报表含有若干科目的数据。如果以科目为索引循环,先用 SUM 语句计算出几个分社此科目之和,然后再用 LOCATE 或 SEEK 语句查找到总社中这个科目的记录,并写进去。每一次都在原大库中用 LOCATE 或 SEEK 定位查找,而 LOCATE 对每个数据库记录进行“顺序”的搜索,在原库记录非常多的情况下要化很多的时间。相反,如果先用 SYS(3) 函数生成一个临

时库,把需要统计的所有分社的数据记录拷贝到这个临时库中,原库中通过 SET FILTER TO 筛选出总社的模式库。此时可以按总社模式库中的科目循环,选择临时库所在工作区进行 SUM 求和运算,要节省很多时间。

FOXPRO 通用程序如下:

```

SELE 1
USE &库文件名
SET FILTER TO &FILTEXP &&FILTEXP 是筛选条件
LSK=SYS(3)
COPY TO &LSK
SELE 2
USE &LSK
SELE 1
SET FILTER TO &FILTEXP1 &&FILTEXP1 用来筛选总库模式
DO WHILE!EOF(1)
  SELE 2
  SUM &FIELD_NAME TO TJZ FOR &FILTEXP2
  * FIELD_NAME 为要求和字=的字段名,FILTEXP2 为求和条件
  SELE 1
  IF RLOCK()
    REPL &JILED_NAME WITH TJZ
  ENDI
  UNLOCK
  SKIP IN 1
ENDDO
SELE 2
USE
DELETE FILE LSK+'&.DBF'
SELE 1
SET FILTER TO
RETURN

```

实验证明,当数据库记录较多时,这种方法,效果非常好。它节省了在大库中用 LOCATE 或 SEEK 搜索记录的时间,并且 SUM 语句按条件求和,在小的临时库中求和也要比在大库中求和运算节省时间。显而易见,这种拷贝临时库的方法大大节约了时间,速度增快不少。

以上是在利用 FoxPRO 编写应用软件中就如何提高运行速度方面的一些体会,在 NOVELL 网络环境下运行通过。

参考文献:

[1]王振诚、董云,《FoxPRO2.0 应用技术开发》,北京希望电脑公司

[2]王珏,《NOVELL 网组原理与设计》,清华大学出版社