

FoxBASE 文件数据切分及数据转换方法

沈 鸣 曹福元 赵 华 (南京大学图书馆)

摘要:本文着重介绍 C 语言与 FoxBASE 数据文件的接口及 FoxBASE 中 DBF 文件数据的切取方法,并介绍实现 FoxBASE 与其它数据库管理系统间的数据转换程序的设计方法。这种方法较好地解决了不同数据库环境下的数据转换问题。

C 语言具有灵活性强、移植性能好、数据运算和处理能力强、运行效率高等特点,作为多种关系数据库系统的宿主语言,它得到了越来越广泛的应用。

国内现有的应用系统中,采用 FoxBASE、dBASE 作为开发工具的占相当大的比例。但它存在系统本身占用内存空间大、数据运算和处理能力弱、运行效率低等弱点,由于 C 语言具有上述特点,程序设计人员常采用 C 语言和 FoxBASE 混合编程的方法,提高运行速度和数据处理能力;另一方面,随着以 C 语言为宿主语言的关系数据库管理系统(DBMS),如 Informix、Oracle、Unify 在我国的普及,采用这些数据库管理系统开发的应用系统相继出现,C 程序设计人员常需要调用 dBASE、FoxBASE 中的 DBF 文件数据或对其实施数据转换。因此使用 C 语言直接对 FoxBASE 建立的 DBF 文件进行操作则成为关键。本文就使用 C 语言实现 FoxBASE 与其它 DBMS 间的数据转换为例,介绍 C 语言与 FoxBASE 数据文件的接口、FoxBASE 的 DBF 文件数据的切取及数据转换的程序设计方法。

数据转换工作虽然可通过将 DBF 文件首先转换成正文文件,而后通过读取正文文件、按字段切分的方法加以实现。但这种方法使用不便,不够灵活。

一、FoxBASE 数据库文件的结构描述

FoxBASE 关系数据库最基本的数据库文件是一种二维表格文件。由库结构和数据两部分组成。数据库文件结构见表 1。其中库结构是数据库文件的描述部分,它包括库文件说明和字段说明。

表 1 FoxBASE 数据库文件存储结构

库说明	字段 1 说明	..	字段 N 说明	库描述 结束符	记录 1 .. N	..	记录 N	库结构 结束符
32	32		32	1	1.. 4000 字符			1

1.库文件说明部分

它由 DBF 库文件的最后修改日期、记录个数、库结构长度和记录长度等信息组成。见表 2。

表 2 FoxBASE 数据库文件说明

字节序号	占字节数	说 明
1	1 字节	数据库文件标识
2-4	3 字节	最后修改日期(年、月、日)
5-8	4 字节	库文件记录个数
9-10	2 字节	库结构长度
11-12	2 字节	记录长度
13-32	20 字节	保留未用

库结构长度 = (字段数 + 1) × 32 + 1(字节)。

字段数 = 库结构长度 / 32 - 1。(整除)。

FoxBASE 库文件以 0DH 作为库结构描述结束符;以 1AH 为数据库文件的结束符。

2.字段说明部分

包括数据库文件中各字段的说明。它定义了库文件中各字段的名称、字段类型、字段数据相对于当前记录的位移量、字段长度及小数位数等属性。见表 3。

其中每条记录的首字符表示当前记录的状态: '*'(2AH)表示记录删除;空格(20H)表示记录有效。

记录的实际长度等于 DBF 文件中所定义的各字段长度之和加 1。

表 3 FoxBASE 数据库文件字段说明

字节序号	占字节数	说 明
1-11	11 字节	库文件字段名, 不足补 OOH
12	1 字节	字段类型
13-16	4 字节	字段数据相对地址
17	1 字节	字段长度
18	1 字节	小数位数
19-32	14 字节	保留未用

二、FoxBASE 与其它 DBMS 数据转换的步骤

1. 在 C 程序中建立 DBF 文件与其它 DBMS 文件字段对照表

建立字段对照表的最大好处是保持程序与 DBF 文件字段属性的相对独立, 使用户在实施数据转换时无需考虑 DBF 文件中各字段的名称、字段的长度、小数位数, 只需考虑 DBF 文件与转换的目标文件中的字段对应顺序及数据类型。当 DBF 文件中的字段名称、字段长度发生变化, 也无需修改源程序, 大大简化了操作。这里指的目标文件结构并不要求与 DBF 文件结构相同。

2. 以二进制读取文件方式, 读取 DBF 库文件描述信息。计算字段数

3. 根据库结构长度、字段数, 获取各字段的属性

4. 通过文件指针定位, 以正文文件方式读取各个字段的数据, 并根据各个字段的数据类型实施转换、文件间字段数据的传送和转换目标文件中的数据拼接。

在 DBF 文件中, 字符型、数值型、逻辑型及日期型数据均以字符方式存储, 逻辑型数据用字符 T、F 表示真值和假值; 日期型数据的存储与输入形式稍有不同, 如日期型数据“01/31/93(MM/DD/YY)”在 DBF 文件中是以“19930131”的形式存储。因此用户在转换数据时, 应根据不同的数据类型执行相应的操作。有关转换目标文件中数据的拼接处理, 用户可根据各自应用系统中的具体要求编写。

下面给出了具体实现的程序段。从程序中可以看

出, 其重点在于 DBF 文件的切分。只需调用转换目标数据库管理系统(Unify, Oracle 等), 所提供的数据库增加或插入语句(包括 SQL 语句), 即可完成不同数据库管理系统环境下的数据转换。

```

/*****
FoxBASE 与其它 DBMS 数据转换程序示例
*****/
#define MAXFLD (sizeof(f_arr) / sizeof(struct F_ARR))
#define DEL '*' /* 定义删除标记 */
#include "stdio.h"
#include "malloc.h"
#include "file_db.strv" /* file_db 文件视图与结构(省略) */

struct F_ARR { /* dbf 数据与 file_db 文件结构对应表 */
    int fld_no; /* dbf 数据字段序号 */
    char type; /* dbf 数据字段类型 */
    char * flchame; /* 对应转换目标文件 file_db 字段标识符 */
    char * ptr; /* 存放 Foxbase dbf 数据字符串指针 */
} f_arr[] = { /* 因篇幅有限简化 */
    {1, 'C', "no"}, /* 字符型 */
    {2, 'C', "title"},
    {3, 'C', "author"},
    {4, 'C', "publisher"},
    {5, 'N', "page"}, /* 数值型 */
    {6, 'D', "date"} /* 日期型 */
};

struct DBSTRU { /* dbf 库结构各字段说明见表 2 */
    unsigned char c, yy, mm, dd;
    long recnum;
    int dblen;
    int reclen;
    char bak[20];
} dbstru;

struct FLDDIR { /* 各字段说明见表 3 */
    char name[11], type;
    long address;
    unsigned char width, dec;
    char bak[14];
} flddir[MAXFLD];

FILE * fp;
int flchum;
    
```

```

/*****
  主函数 main( )
***** /
main(argc,argv) /* 主函数简化如篇幅有限可省略 */
int argc;
char * argv[];
{
if((fp = fopen(argv[1],"rb+")) = NULL){
errdisp("不能打开指定 dbf 文件!",22,0);
exit(1); /* errdisp()—定位显示信息函数(省略) */
}
init__head();
Get__dbf();
exit(0);
}
/*****
取 Foxbas dbf 文件头信息
***** /
int init__head()
{
int i;

fseek(fp,OL,0);
fread(&dbstru,sizeof(dbstru),1,fp); /* 取库结构信息 */
flchum = dbstru.dblen / 32-1; /* 计算字段数 */

fseek(fp,32L,0);
for(i=0;i<flchum;i++) /* 取 dbf 文件各字段定义 */
fread(&flddir[i],sizeof(flddir[i]),1,fp);
for(i=0;i<flohun;i++){ /* 根据各字段长度分配存储空间 */
f__arr[i].ptr = (char *) malloc(flddir[i].width+1);
if (!f__arr[i].ptr){
errdisp("分配内存空间失败!",22,0);
return(-1);
}
}
return(0);
}
/*****
取 Foxbas dbf 文件数据并进行转换
***** /
int Get__dbf()
{
char c,ch;
int j,k=0;
long ps;

```

```

ps = dbstru.dblen;
fseek(fp,ps,0);

for(k=0;k<dbstru.recnum;k++){ /* 取 dbf 文件记录直到文件结束 */
if((c = fgetc(fp)) = DEL){ /* 判记录删除标记 */
ps += dbstru.reclen; /* 根据记录长度移动文件指针 */
fseek(fp,ps,0);
continue;
}
for (j=0;j<flchum;j++){ /* 取各字段数据存入相应 f__arr[j].ptr 中 */
fread(f__arr[j].ptr,flddir[j].width,1,fp);
*(f__arr[j].ptr+flddir[j].width) = '\0';
ps += flddir[j].width; /* 根据各字段长度移动文件指针 */
}
AM__rec(); /* 取 f__arr[j].ptr 内容拼接,存入转换目标文件结构中 */
if(dbadd("file_db",&fk)) /* 写库操作(fk 为文件 file_db 的库结构)(省略) */
errdisp("8 增加数据操作失败!",22,0);
}
return(0);
}
/

```

注:AM__rec():函数(省略)—取 f__arr[j].ptr 中的内容,即字段内容拼接,存入数据转换目标文件结构中
用户可根据各自应用系统文件结构编写

以上对使用 C 语言转换 FoxBASE 文件数据的程序设计方法作了详细描述。这种方法不受 DBF 文件字段名、字段长度等属性的制约,可有选择地进行数据转换,十分灵活。此方法稍加修改还可直接对 DBF 文件进行数据的增、删、改等操作。在录前使用 C 语言、Informix、Unify、Oracle 和 FoxBASE 所开发的应用系统并存的情况下,掌握和灵活应用这种方法有广泛的意义。

参考文献:

- [1] 郑莆京编著:“FOXBASE+关系数据库系统”,清华大学出版社,1991.9
- [2] [美] Herbert Schildt 著:“C 语言大全”,电子工业出版社,1990.10
- [3] 金茂忠主编:“C 程序设计高级教程”,北京航空航天大学出版社,1990.6 社,1989.7