

# Design and Implementation of Database Interface in C Language

Ping Wang

Fuzhou University of International Studies and Trade, Fuzhou Fujian  
Email: [pwang@fjnu.edu.cn](mailto:pwang@fjnu.edu.cn)

Received: July 28<sup>th</sup>, 2019; accepted: August 12<sup>th</sup>, 2019; published: August 19<sup>th</sup>, 2019

---

## Abstract

After explaining the characteristics of C language and the importance of database technology, this paper introduces the Union and extensiveness of the two technologies in the application of equipment process control. It focuses on the detailed analysis of the database interface technology encountered in the system development, and shows its application field from one side through an example.

## Keywords

C Language, Database, Interface Technology, Development and Application

---

# C语言中数据库接口的设计与实现

王 平

福州外语外贸学院, 福建 福州  
Email: [pwang@fjnu.edu.cn](mailto:pwang@fjnu.edu.cn)

收稿日期: 2019年7月28日; 录用日期: 2019年8月12日; 发布日期: 2019年8月19日

---

## 摘 要

本文在阐述C语言的特点以及数据库技术的重要性之后, 简述了二者在设备过程控制应用中的结合性和广泛性, 着重就系统开发中碰到的数据库的接口技术进行详细解析, 并通过一个示例从一个侧面展示了其应用领域。

## 关键词

C语言, 数据库, 接口技术, 开发应用

---

Copyright © 2019 by author and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着微电子技术、计算机硬件技术和通信技术的发展,计算机以它的速度快、精度高、又具有逻辑判断能力的特点在过程控制方面可有较多的应用。例如:监测报警系统、医疗图像系统、指纹鉴别系统、电脑记分系统以及生产控制系统。此类系统基本要求是具备较高级别的实时性、可靠性和智能性。

使用 C 语言开发这些系统是很好的选择,其原因是 C 语言具有代码量小、运行速度快、功能强大的特点[1]。例如同样都是办公软件,用 C 语言编写的 WPS 的安装文件只有 50 MB 左右,而 Word 的安装文件超过 500 MB。为了实现运行速度快和功能强大,操作系统也大都是用 C 语言编写的[2]。

数据库系统是为适应数据处理的需要而发展起来的一种较为理想的数据处理的核心机构。计算机的高速处理能力和大容量存储器提供了实现数据管理自动化的条件。数据库因其结构化、共享性、独立性的特点使得开发的系统可以智能分析和全面高效。所以在系统开发中使用 C 语言 + 数据库是一种较为合适的选择[3]。

## 2. 过程控制处理系统的基本结构

传统的过程控制处理是先采集以此为依据的数据,再通过专业人员分析判断做出相应的决策,最后实施对应的处理。这种以人工为主的传统处理模式存在很大的弊端。例如:数据不便保存、因疲劳引起的误判和误动、处理周期长以及不便交流等,尤其是在长时间、快速、反复精确处理的要求环境下,此类问题就更突出。

以计算机为主体的过程控制处理系统可以较完善地解决这些问题。通常,一个完善的系统要有各自功能独立的子系统组成,子系统不仅是软件模块还可以是具有独立处理部分功能能力的计算机。它的大体模式是用先将数据采集进计算机,而后根据需要不断地采集分析判断控制输出。配上局域网又可以实现主机控制分机、主机汇总统计以及主分机间和分机间的交流等功能[4]。

因计算机的处理速度和通讯带宽的提高,实时性的要求一般均能满足。为满足可靠性的要求,针对不同操作系统的特点,在主机上可采用 Windows 平台,以便可视化软件的开发应用。在分机上可采用 Dos 平台,目的是降低成本和提高可靠性(避免断电后 Windows 的维护,复电即可工作)。Dos 平台上, C 语言也是一种较好的选择,它既像高级语言一样面向用户解题过程,又像汇编那样可以对机器硬件进行操作[5]。如,进行端口 I/O 操作,位操作以及地址操作等。许多操作系统都是用 C 语言来写的。所以针对系统的硬件控制、数据采集以及多功能模块的特点可以选择 Borland C++。

本文着重就过程控制类项目的开发过程中遇到问题进行解析和证明,诸如: C 语言与数据库如何结合?在编程时如何用 C 语言写与读带结构的数据库格式文件?并提供了摸索调试后可用的接口代码,最后用一个实际完成的系统加以证实可行性。

## 3. C 语言中的数据库接口

为提高整个系统的智能性,实现主机对分机的监控、调配和统计,分机应将所处理的数据以数据库的格式传送给主机,所以在 C 语言中要以相应格式读写文件。

广为流行的应用软件系统 dBASE 和 FoxBASE 是带结构的文件,数据库文件由文件结构和数据两部

分组成。文件结构采用的是二进制存储方式而数据部分则是以 ASCII 方式存储。库文件结构部分描述库文件的概况：建立或修改日期、数据记录个数、库文件结构长度、记录长度等。

C 语言具有很好的底层控制能力，也有丰富的函数可以调用，但因为各种文件结构上的差异，故无这些带结构文件的读写函数。如何在 C 语言中编写 dBASE 和 FoxBASE 带结构的文件也是件很繁琐的事，笔者在开发过程中经过反复分析、调试最后圆满地解决了这个问题，很好地实现了带结构的数据库文件的读与写，为日后方便使用现整理如下：

### 3.1. 接口代码

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<dos>
#include<bios.h>
#include<ctype.h>
#include<conio.h>          /*此段为 C 标准库的头文件*/
struct sfox {
    unsigned char mark;      /*标志*/
    unsigned char yy,mm,dd; /*年月日*/
    unsigned long recn;     /*记录个数*/
    unsigned int structl,reccl; /*库结构长度，记录长度*/
    char string[20];        /*保留*/
} sfox;                    /*此段为库文件结构部分*/

struct rec {
    char sign;              /*数据库的逻辑删除记号位*/
    char za[2];             /*字段 1，2 位，类型不限*/
    char zb[2];             /*字段 1，2 位，类型不限*/
    char zc[3];             /*字段 1，3 位，类型不限*/
} record;                  /*此段为库文件数据部分*/

void write()
{
    unsigned int filehead,reccount,reclen;
    int l,start,no;
    char ch,za[2],zb[2],zc[3];
    unsigned char a,b;
    FILE *fpout;
    If((fpout=fopen("fl.dbf","rb+"))==NULL){
        Printf("Can't open fl.dbfn ");
    }                        /*打开文件*/
```

```

fseek(fpout,8,0); /*定位*/
fscanf(fpout," %c%c" ,&a,&b); /*读数据*/
filehead=a+b*0x100; /*头长度*/
fseek(fpout,4,0);
fscanf(fpout," %c%c" ,&a,&b);
reccount=a+b*0x100; /*记录数*/
reclen=sizeof(struct rec); /*一条记录的长度*/
printf("old recount=%d\n",reccount); /*显示旧记录数*/
printf("start new recount=");
scanf("%d",&start); /*从第几条起加新记录*/
printf(" how much new recount=");
scanf("%d",&no); /*加几条新记录*/
rewind(fpout); /*移动文件指针到流的开头*/
fread(&fox,sizeof(struct sfox),1,fout); /*按头结构格式读入*/
fox.recn=start+no-1; /*新记录数*/
for(I=(start-1);I<(start+no-1);I++)
{
    printf("record.za[2]="); /*提示*/
    scanf("%s",za); /*键输*/
    strcpy(record.za,za); /*读入*/
    printf("record.zb[2]=");
    scanf("%s",zb);
    strcpy(record.zb,zb);
    printf("record.zc[3]=");
    scanf("%s",zc);
    strcpy(record.zc,zc);
    fseek(fpout,filehead+I*reclen,0); /*定位*/
    if(fwrite(&record,sizeof(struct rec),1,fpout)==NULL){
        printf("Can't write fl.dbfn");
    } /*写入整条记录*/
}
ch=0x1a;
fwrite(&ch,1,1,fpout); /*写入末尾标记*/
rewind(fpout); /*移动文件指针到流的开头*/
fwrite(&fox,sizeof(struct sfox),1,fout); /*将头结构重新写入*/
fclose(fpout); /*关闭文件*/
}

void read();

```

```

{
    unsigned int filehead,reccount,reclen;
    int I;
    char ch,za[2],zb[2],zc[3];
    unsigned char a,b;
    FILE *fpout;
    If((fpout=fopen("fl.dbf","rb+"))==NULL){
        Printf("Can't open fl.dbf\n");
    }
    fseek(fpout,8,0);
    fscanf(fpout,"%c%c",&a,&b);
    filehead=a+b*0x100;
    fseek(fpout,4,0);
    fscanf(fpout,"%c%c",&a,&b);
    reccount=a+b*0x100;
    reclen=sizeof(struct rec);
        for(I=0;I<reccount;I++)
            {
                fseek(fpout,filehead+I*reclen,0);
                if(fread(&record,sizeof(struct rec),1,fpout)==NULL){
                    printf("Can't read fl.dbf\n");
                }
                strcpy(za,record.za);
                strcpy(zb,record.zb);
                strcpy(zc,record.zc);
                printf("za=%s",za);
                printf("zb=%s",zb);
                printf("zc=%s",zc);
            }
            fclose(fpout);
        }
        void main()
        {
            write();
            read();
        }

```

### 3.2. 使用说明

上述代码虽然只是针对 dBASE 和 FoxBASE 这样文件，但对于其他另外结构格式的文件，开发原理相同，代码大同小异，为更好地使用和扩充，特归纳提示如下：

① 计算机中的文件存储都是二进制形式，关系数据库展开来体现的是张表格，其中的列就是字段，列有不同的宽度，所以在编写程序时：虽然数据库字段类型可以不同，但字段长度应与定义的库文件数据的长度一致。

② 计算机中的文字是以 ASCII 方式表示，字符的间隔、字符串的结束都有约定的代码表示，就是因为字符串结束符的限定，在调用“读函数”或“写函数”时会则有区别，虽然“写函数”输入时只要不超过定义的位数即可，但调用“读函数”时则会出错。所以要注意定义的位数的匹配。

③ 在计算机的输入输出底层接口，为提高通信速度，如同其他文件的传输，此接口程序也是以记录

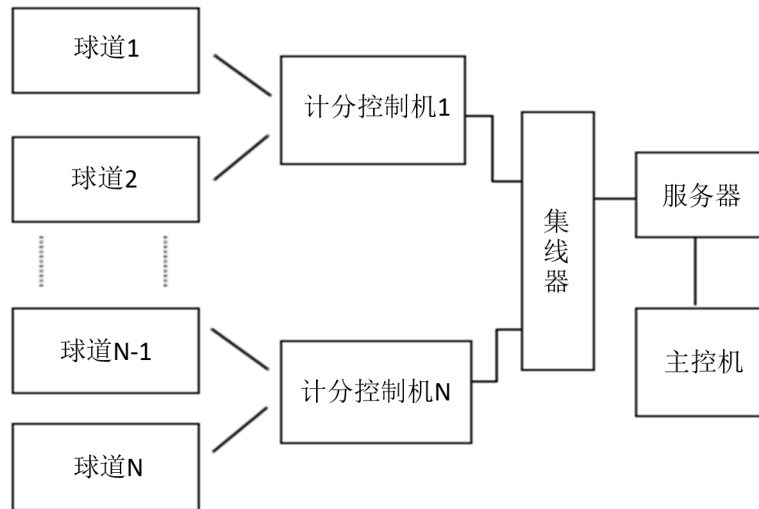


Figure 1. The hardware block diagram  
图 1. 硬件框图

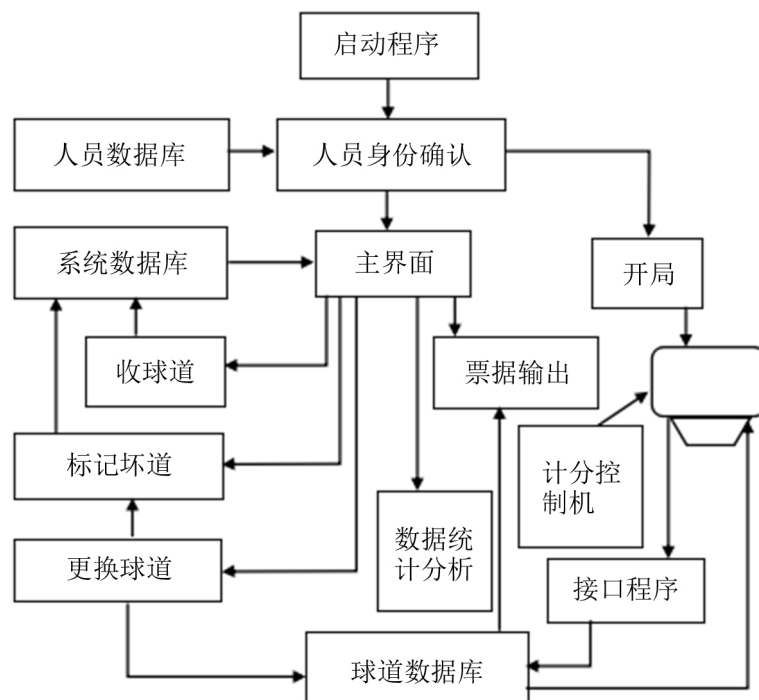


Figure 2. The software block diagram  
图 2. 软件框图

为单位进行读写的。

④ 关系数据库这张表格，其中的行就是记录。为了提高可用性和完善功能，因其“写函数”中记录末尾标记的引入和新记录数的重写，在使用过程中数据库的记录数会很方便地随之改变。

⑤ C 语言的一大特点就是函数，已有的函数都存放在相应的头文件中，编程时可以方便地调用。本程序也可根据需要修改程序后再放到头文件中，仅在主函数中调用即可。

#### 4. 应用示例

以开发的保龄球计分通讯系统为例，如图 1、图 2 所示，每台分机(电脑)可独立地处理两个球道的开局、计分、动画以及机器的控制，采用 Dos 平台、C 语言编程以提高其可靠性，各台分机将执行的数据以数据库的格式记录留底。主机则采用 Windows 平台、可视化编程，通过局域网传送的数据对全部分机进行监控、打印、分析、汇总以及开局、换道等。形成了一套功能完善、性能可靠的系统[6] [7] [8]。

#### 5. 总结

众所周知，智能控制系统能够极大地提高工作效率，对于计算机应用而言，智能控制就是过程控制。在过程控制系统中，计算机中运行的软件程序如同人类的大脑，周边配置的硬件如同人类的鼻、耳、手、脚，只有软、硬件完美配合才能完成复杂的任务[9]。所谓系统工程强调的是整体效能，故在规划系统时要考虑可靠性与性价比[10]。在项目规划时，C 语言与数据库的结合是一很好的技术路线，在具体实施中，带结构的数据库文件读与写的编程是一绕不过的关口，本文 C 语言写的数据库接口程序可以直接调用，方便过程控制系统的开发。

#### 基金项目

福建省电子信息工程特色专业建设点(L200802008)。

#### 参考文献

- [1] 王肖飞. 基于 C 语言的计算机软件编程技术探究[J]. 无线互联科技, 2018(5): 47-48.
- [2] 冀泓宇. 基于 C 语言的计算机软件编程技术研究[J]. 电脑迷, 2018(5): 47-48.
- [3] 刘振兴. 软件编程技术水平的标志性关键点分析与应用[J]. 软件, 2017, 38(11): 191-195.
- [4] 郭畅. 基于单片机的篮球计分器系统的设计[J]. 电子测试, 2017(16): 31-33, 17.
- [5] 牛军, 王萍. 无线电子打分计分系统的设计与实现[J]. 南阳理工学院学报, 2015, 7(4): 1-4.
- [6] 李维强. 体育用电子记分器的设计及制作研究[J]. 电子设计工程, 2015, 23(22): 191-193.
- [7] 张丽. 基于 C 语言访问的 MySQL 数据库[J]. 电子技术与软件工程, 2016(22): 165-166.
- [8] 罗慧兰. C 语言处理文件的关键技术[J]. 电子技术与软件工程, 2017(6): 242.
- [9] 王晓燕. MySQL 安全问题探讨[J]. 办公自动化, 2018(23): 62-64.
- [10] 刘建宏. MySQL 数据库优化与集群[J]. 数字通信世界, 2017(7): 47.

### 知网检索的两种方式:

1. 打开知网首页: <http://cnki.net/>, 点击页面中“外文资源总库 CNKI SCHOLAR”, 跳转至: <http://scholar.cnki.net/new>, 搜索框内直接输入文章标题, 即可查询;  
或点击“高级检索”, 下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询。
2. 通过知网首页 <http://cnki.net/>顶部“旧版入口”进入知网旧版: <http://www.cnki.net/old/>, 左侧选择“国际文献总库”进入, 搜索框直接输入文章标题, 即可查询。

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: [csa@hanspub.org](mailto:csa@hanspub.org)