

# SQLite 在远程管理 Wince 智能设备中的应用

① 颜昌青 江志农 钟海(北京化工大学 诊断与自愈工程研究中心 北京 100029)

**摘要:** 本文首先介绍了嵌入式数据库 SQLite 的技术特点及内部结构, 之后分析了 SQLite 在 Windows ce.net 系统中的移植和封装, 在此基础上研究了利用 RAPI 远程管理 Windows CE 智能设备。

**关键词:** 智能设备; SQLite; Windows ce.net; RAPI

## Application of SQLite in remote control over Wince intelligent device

YAN Changqing, JIANG Zhinong ZHONG Hai

(Diagnosis and Self-recovery Engineering Research Center, Beijing University of Chemical Technology, Beijing 100029)

**Abstract** This paper firstly introduces the technical features and internal structure of SQLite, a kind of embedded database, then analyses the transplantation and encapsulation of SQLite in Windows ce.net operation system, and finally researches on how to perform the task of RAPI remote control over Windows CE intelligent devices based on this foundation.

**Keywords** Intelligent device; SQLite; Windows ce.net; RAPI

虽然 Oracle、DB2、SQL Server 及 MySQL 等关系数据库系统是当今数据库应用的主流, 但在硬件资源极其匮乏的嵌入式系统中, 大型的关系型数据库未免显得累赘且不可取, 而可嵌入和配置的小型数据库在嵌入式系统的数据管理中优势甚为突出。零配置、功能齐全的开源嵌入式数据库 SQLite 是 SQL 数据库引擎的小型 C 库。本文基于实际项目, 利用 SQLite 及 RAPI 与 Windows CE 智能设备的一个数据采集程序 DataAcq 进行交互, 主要介绍了 SQLite 的特点及内部结构、在 Windows ce.net 系统中的移植与封装, 以及利用 RAPI 和 SQLite 远程管理 Windows CE 智能设备。

## 1 SQLite 的特点及内部结构

### 1.1 SQLite 的特点

开源 SQLite 是一个嵌入式的 SQL 驱动的数据库引擎, 它可以作为数据库引擎或者 C/C++ 库的接口使用。SQLite 完全独立, 不需任何特殊设定就能跨平台编译, 简单而可靠, 且非常高效。它具有以下的特点<sup>[1]</sup>:

- (1) 支持 ACID 事务, 兼容视图, 子查询及触发器;
- (2) 内存占有量小, 编译后低于 250KB, 但能支持容量达 2TB 的数据库;
- (3) 零配置, 无需安装和管理;
- (4) 一个完整的数据库保存在单一磁盘文件中;
- (5) 支持无长度限制的字符串和 BLOB 数据;
- (6) 使用简单, 速度比普通客户/服务器数据库系统快;
- (7) 支持大多数的 SQL92 语句;

(8) 支持 Windows 系列、Unix 系列、Linux 系列、OS2 系列操作系统平台;

(9) 支持多种开发语言, C/C++, Perl, Java 等。

### 1.2 SQLite 的内部结构

SQLite 的体系结构自底而上可分为操作系统接口层、存储管理层、虚拟机层及用户接口层。在内部主要有四个核心组件: 内核 (Core), SQL 编译器 (SQL compiler), 后端 (Backend) 和附件 (Accessories), 结构如图 1 所示。

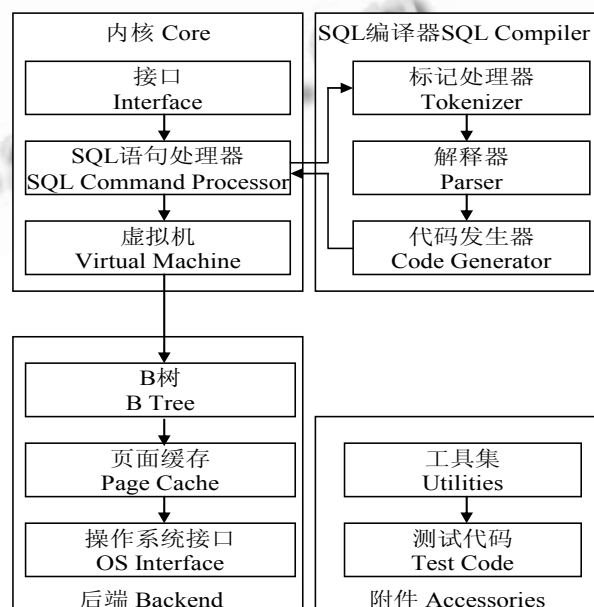


图 1 SQLite 内部结构图

① 基金项目: 国家高技术研究发展计划 (863 计划) 专题课题 (2007AA04Z433); 国家自然科学基金重点资助项目 (50635010)

SQLite 利用虚拟机和虚拟数据库引擎 (VDBE) 使调试、修改和扩展 SQLite 的内核变得更加方便, 所有 SQL 语句都被编译成能在 SQLite 虚拟机中执行的程序集。

SQLite 类库大部分的公共接口程序是由 main.c、legacy.c 和 vdbeapi.c 源文件中的功能执行的。当执行一个包含 SQL 语句的字符串时, 接口程序要把这个字符串传递给 Tokenizer。Tokenizer 的任务是把原有字符串分成一个个标示符, 并把这些标示符传递给剖析器。在剖析器收集完符号并把之转换成完全的 SQL 语句时, 它调用代码产生器来产生虚拟的机器代码, 这些机器代码将按照 SQL 语句的要求来工作; 在后端, 数据库按照 B 树(B-Tree)的形式存储在磁盘上, 通过可调整的页面缓冲(Pager)得到对数据的快速查找和存储。SQLite 还使用了一个抽象层接口(OS Interface)与不同操作系统进行对接, 附件主要是用于代码测试。

## 2 SQLite 在 Windows CE 系统的移植和封装

目前 Windows CE 系统常用的数据存储管理方法大致有三种, 第一种是直接使用文件系统, 利用文件系统的应用程序接口实现对数据的读写管理操作; 第二种是使用与 Windows CE 系统匹配的数据库 SQL Server CE 对数据进行存储管理; 第三种是使用嵌入式数据库 SQLite 管理数据。SQLite 与文件系统相比在进行数据存储管理时有巨大优势。SQLite 数据库只在写操作时采用独占方式, 读操作时采用共享方式, 允许并发访问; 而 Windows CE 系统对文件的访问采用独占方式, 这样不利于数据的共享。使用 SQLite 能够方便的进行数据的插入、读取、排序, 并能够快捷的获取某一项记录的某一属性; 而使用文件系统要实现上述功能非常麻烦, 需要频繁的使用文件系统 API 函数, 并得筛选出感兴趣的属性。另外 SQLite 支持 ACID 事务, 能够保证即使在系统崩溃或断电时数据的完整性和有效性, 而文件系统却没法实现<sup>[2]</sup>。

SQLite 与 SQL Server CE 某些方面相比也有足够的优势。SQLite 占用很少的存储器空间, 约为 100KB-300KB; 而 SQL Server CE 则大致占用 1MB-3MB, 这对资源极其珍贵的嵌入式系统来说无疑是个挑战。SQLite 与 SQL Server CE 都采用标准的 SQL 语句, SQLite 使用非常简单, 只需要掌握 3 个核心 API 函数就可以实现基本的数据操作, 而 SQL Server CE 相对复杂, 需要充分理解数据库引擎和掌握高级数据库接口组件 ADOCE 的编程。另外 SQLite 完全独立, 有良好的移植性; SQL Server CE 为 Microsoft 开发的

Windows CE 平台上的嵌入式数据库, 没法移植到其他嵌入式平台上。

### 2.1 SQLite在Windows ce.net系统中的移植

首先从 SQLite 官方网站上下载 SQLite 源码, 再在 Embedded visual c++中进行编译。以下是编译流程:

(1) 打开 EVC 新建一个“WCE Dynamic-Link Library”工程, 命名为 sqlite3;

(2) 在接下来的对话框中选择“An empty Windows CE DLL project”;

(3) 将源码中所有的\*.c, \*.h, \*.def 文件复制到工程文件夹下;

(4) 在 Source Files 中添加除 shell.c 和 tclsqlite.c 两个文件外的所有\*.c 文件;

(5) 在 Header Files 中添加所有\*.h 的源文件;

(6) 将 SQLite 源文件中的 sqlite3.def 文件添加到在工程的 Source File 中;

(7) 在 EVC 中选好编译平台, 例如“Win32 (WCE ARMV4) Debug”;

(8) 编译工程, 则会在工程目录的 ARMV4Dbg 目录生成了两个重要文件: 动态链接库文件 sqlite3.dll 和引入库文件 sqlite3.lib。

### 2.2 SQLite API的封装

SQLite 提供了很多的 API 函数供用户进行数据库编程, 如打开数据库函数 int sqlite3\_open(char \*,sqlite3 \*\*); 数据库语句执行函数 int sqlite3\_exec(sqlite3\*,const char \*sql,sqlite3\_callback,void \*,char \*\*errmsg); 数据库关闭函数 int sqlite3\_close(sqlite3 \*)等。若直接使用原始的 SQLite API 函数, 不仅代码的可读性不强, 维护起来也相对困难<sup>[3]</sup>。本文对原始的 API 函数进行了简单的封装, 并且导出动态链接库 DLL, 供上位机 (PC 端) 远程管理 Windows CE 智能设备时使用。以下是 MySQLite 类的声明:

```
class MySQLite
{
private:
    sqlite3 *db;//数据库句柄
    char **azResult;//二维数组存放结果
    char *zErrMsg;//保存错误信息
    int row;//行
    int column;//列
public:
    //连接数据库
```

```
bool DBConnect(char *filename);
bool ExecSql(char *sql); //执行 SQL 命令
void DBSearch(char *search_sql); //查询
//查询某行某列的值
char *GetTableData(int x,int y);
bool DBDisconnect(); //断开数据库连接
char *GetErrorMsg(); //取得当前错误提示
};
```

利用 SQLite.dll 和类 MySQLite 生成动态链接库 TaskMag.dll, 导出外部接口 SetParams(设定程序 DataAcq 的采集参数)、StartDataAcq(启动 DataAcq 程序, 根据所设参数进行采集, 并把数据写进数据库)及 ReadData(从数据库中读取采集数据并解析), 其中每个函数的声明需满足如下格式: STDAPI INT FuncName(DWORD cbInput, BYTE \*pInput, DWORD \*pcbOutput, BYTE \*\*ppOutput, IRAPIStr \*pIRAPIStr eam)。

### 3 远程应用编程接口 RAPI

PC 端应用程序利用 RAPI 就能远程管理 Windows CE 设备, 处理 Windows CE 设备的文件系统、注册表和数据库以及查询系统配置信息。调用 RAPI 函数之前, 必须调用函数 CeRapiInit 或 CeRapiInitEx 来初始化 RAPI 库。结束所有必要的 RAPI 调用时, 需调用 CeRapiUninit 函数来进行清除。RAPI 服务包括很多预定义的 RAPI 函数, 如系统信息函数、文件目录管理函数、数据库函数、注册表管理函数和外壳管理函数等。无论 RAPI 支持多少预定义的接口函数, 总会出

现一些应用程序需要但 RAPI 接口没有提供的函数。因此 RAPI 提供了一种通用方法允许 PC 端的应用程序去调用 Windows CE 设备上的用户定义函数。RAPI 函数 CeRapiInvoke 允许我们调用远程设备中任何动态链接库中的 API 函数。有了上述的准备, 我们就可以通过调用 TaskMag.dll 中的 SetParams、StartDataAcq、ReadData 进行参数设定、数据采集和数据解析, 并利用 CeCreateFile、CeReadFile 等函数将采集的数据文件写到 PC 端, 供 PC 端强大丰富的数据处理软件做进一步的分析和研究。

### 4 总结

嵌入式数据库的广泛应用已成为嵌入式系统的一个重要分支。SQLite 以其小巧的体积、强大的功能、开源免费及零配置等优势迅速成为各嵌入式操作系统优选数据库。本文基于实际项目, 证明了 SQLite 在 Windows ce.net 系统中具有占用资源少、性能可靠及操作简单等优点。相信 SQLite 一定会有更深入的研究和更广泛的应用。

#### 参考文献

- 1 Unavailable. Features Of SQLite, 2008-10-20. <http://www.sqlite.org/features.html>.
- 2 史震宇. 基于嵌入式数据库 SQLite 的交通信息采集单元[硕士学位论文]. 天津: 天津大学, 2007.
- 3 岑冬梅, 陈和平. 基于 SQLite 的二次封装方法在车载导航系统中的应用[J]. 计算机系统应用, 2008(10): 24-27.