

# 基于 QT 的跨平台数据管理接口设计

杜影, 郑义, 石家勇, 刘磊

(北京航天测控技术有限公司, 北京 100041)

**摘要:** 针对数据库跨平台适配复杂高和改造工作量大的问题, 基于 QT 设计了跨平台数据管理接口; 阐述了数据管理接口的开发背景和设计思路, 详细说明了关键数据结构定义和接口函数, 该接口由 C++ 语言开发, 不需要数据库连接, 不依赖任何数据库 API, 可替代数据库常用功能, 执行数据增加、删除、更新、查询和备份, 并可根据平台软件功能需要扩展和优化; 通过在 Intel+Windows、龙芯+中标麒麟系统环境均可编译运行的实例程序验证了该接口的复用性、实用性和替代性, 为使用数据库基本特性的应用情景, 特别是地面装备测试平台软件的数据库国产化替代提供了新的解决思路。

**关键词:** Qt; 跨平台; 数据管理接口; 中标麒麟操作系统

## Design of Cross Platform Data Management Interface Based on QT

DU Ying, ZHENG Yi, SHI Jiayong, LIU Lei

(Beijing Aerospace Measurement & Control Technology Co., Ltd., Beijing 100041, China)

**Abstract:** In response to the complexity and heavy workload in database cross platform adaption, a cross platform data management interface based on Qt is designed. Describe the development background and design ideas, discuss the key data structure definitions and interface functions in detail, develop this interface in C++ language without database connection and relying on any database API, replace common functions of databases, and perform the functions of data addition, deletion, update, query, and backup, which can be extended and optimized according to the functional needs of the platform software. In Intel+Windows and LOONGSON+NeoKylin operating system environments, practical example program that can be compiled and run verifies the reusability, practicality and substitutability of the interface, providing new solution for application scenarios in the basic characteristics of databases, especially for the localization substitution of database in ground equipment testing platform software.

**Keywords:** Qt; cross platform; data management interface; NeoKylin operating system

## 0 引言

“自主可控”背景下, 国产核心基础软硬件在性能、稳定性、扩展性等方面逐步发展与完善, 从而使国产化替代成为军工领域信息化装备发展的必经之路。在国产化替代进程中, 许多现有的地面装备测试系统软件无法直接在国产化平台下运行, 需要对软件功能模块、依赖软件及库函数进行不同程度的移植修改, 甚至重新开发<sup>[1-3]</sup>。特别是基于 Intel+Windows 环境开发的测试系统早已经成为军用信息化应用的主流。在推进国产化时, 需妥善处理原系统的重构与新系统的兼容问题, 既要实现完全国产化替换, 又要确保新系统稳定可靠运行。

地面装备测试平台软件迁移的核心工作在于代码移植和数据库组件更新。尽管国产数据库厂商普遍提供了相关

工具, 但由于不同数据库产品的语法特性和底层机制各异, 实际操作中可能遇到大量复杂的适配和改造需求。

在某装备测试平台软件国产化过程中, 我们设计了一种基于 QT 的跨平台数据管理接口, 通过 C++ 代码开发, 不使用 QSql 模块, 不依赖任何数据库 API, 实现了特定场景数据的增加、更新、删除、查询和备份要求, 同时可以根据需要优化和扩展, 随意在 Intel+Windows 和龙芯+中标麒麟系统跨平台使用。至今为止, 该数据管理接口已在多个地面装备测试平台软件的国产化过程中替代了原有的数据库组件, 减轻了软件开发人员跨平台迁移数据库的复杂度和工作量。

## 1 数据库管理

数据库是“按照数据结构来组织、存储和管理数据

收稿日期:2024-08-06; 修回日期:2024-11-18。

作者简介:杜影(1981),女,高级工程师。

引用格式:杜影,郑义,石家勇,等.基于 QT 的跨平台数据管理接口设计[J].计算机测量与控制,2025,33(3):213-218.

的仓库”，是一个长期存储在计算机内的、有组织的、可共享的、统一管理的大量数据的集合<sup>[4]</sup>。数据库管理的目的是科学有效地组织和存储数据，从大量的数据中快速地获得所需的数据以及对数据进行维护。实际项目中通常采用数据库技术实现数据的存储、检索等功能。

### 1.1 国产数据库现状

在过去的几十年里，数据库系统在不同的平台上都有实现，如 Windows、Linux、MacOS 等。但在国产 CPU（龙芯、申威等）平台上，国产操作系统、数据库等底层基础软硬件以及上层应用软件仍在进行硬件适配、软件移植、性能优化，以更好地支撑全国产化项目开发工作。

如今，国产数据库作为底层基础软件已经实现了从无到有，并向着“好用”阶段迈进。常见的国产数据库有 OceanBase、PolarDB、OpenGauss、TiDB、人大金仓、武汉达梦、GBase、GaussDB、TDSQL 等<sup>[5-14]</sup>。它们都高度兼容 MySQL 的特性，广泛应用于信创产业领域。

在航天、军工（装备制造）域，军工科研院所等机构通过逆向设计、自主研发等方式建立“自主可控”的软硬件国产化生态，打造基于开源软件的通用测试平台，推动国产化替代进程。基于技术发展和综合性能上的考虑，国产龙芯 CPU 搭载中标麒麟操作系统<sup>[15-21]</sup>成为了装备测试仪器国产化硬件平台的首选。而“龙芯”平台上支持且市场占有率高的国产数据库是武汉达梦、神州通用和人大金仓等<sup>[22-23]</sup>。

虽然信息设备国产化替代进程已经展开，鉴于国产数据库仍存在兼容性、安全性、可靠性等问题，装备测试软件中的国产数据库还只是在试点运行并未大规模应用。

### 1.2 数据库跨平台移植

软件程序跨平台移植主要面临的问题是跨平台兼容性和数据库兼容性，跨平台兼容性包括：不同的操作系统、不同的数据库和不同的开发环境。数据库兼容性又包括：不同的数据库类型、不同的数据库版本和不同的数据库语言。因此，针对数据库跨平台迁移，其兼容性往往成为一项挑战。

在实现数据库跨平台移植时，我们需要考虑以下几个方面：

- 1) 使用跨平台编程语言；
- 2) 使用跨平台库；
- 3) 使用跨平台框架。

这样才能在不同的平台上运行和访问数据。

### 1.3 QT 中的数据库操作

Qt 是跨平台 C++ 图形用户界面应用程序开发框架<sup>[24]</sup>，其集成开发环境（IDE）名为 Qt Creator，既可

以开发 GUI 程序，也可用于开发非 GUI 程序，被广泛应用于嵌入式、电力系统、军工系统等要与硬件交互的界面系统中。

选择 Qt 作为开发工具不仅因为其开发功能强大，更主要是因为它有开源版本且支持跨平台运行。Qt 中提供了不同的数据库接口和丰富的数据库操作类。QtSql 模块实现了对数据库的访问，同时提供了一套与平台和具体所用数据库无关的调用接口。

Qt 中的数据库操作有两种方法：一种是不单独配置其他数据库，直接在 QT 开发平台上创建和操作数据库表；另一种是在 QT 平台之外安装数据库，然后通过命令的方式创建数据表结构及添加表，该方法操作更直观。

## 2 数据管理接口设计

### 2.1 设计背景

现有军用地面装备测试系统的国产化替代涉及操作系统、软件及终端设备的全面国产化升级，技术难度高、投入成本大、改造周期较长。

随着国产化仪器生态环境的日趋完善，中标麒麟 5.0 操作系统集成安装 QT5.6.2 开发工具成为测试装备国产化软件的主流开发平台。在该环境中，界面设计和应用程序几乎不需要改动就能够跨平台编译，而使用国产数据库替代传统数据库往往复杂度高且工作量大。

然而，基于多个地面装备测试系统软件的开发经验可以得知，地面测试系统软件的数据管理使用的是关系型数据库，主要实现数据查询、增加、更新、删除、备份等操作，没有海量存储的需求，没有繁多的测试项目和关联关系，也不存在远程数据同步和共享问题。因此，我们只需根据数据库特性和内部机制封装一个数据管理接口执行上述数据操作，保证能快速高效地获取可靠的应用数据，为数据分析和可视化提供有效支撑。

### 2.2 设计思路

考虑到 1.2 节所述的三方面因素，自定义小型“数据库”类操作接口，使用 C++ 代码开发，不使用 QtSql 模块，不依赖于任何数据库连接，常规数据库管理的增、查、删、改、备份都可以通过该接口实现。

首先，数据文件存储成二进制格式，文件大小不超过 500 M，当数据文件大于 500 M 时进行数据备份并重新建立同名数据文件。将数据存成二进制文件可以节省存储空间，内存中参加计算的数据都是用二进制无格式储存起来的，因此使用二进制储存到文件就更快捷；另外，对于一些比较精确的数据，使用二进制储存不会造成有效位的丢失。同时，限制数据文件大小可以降低数据访问复杂度、提高存取效率。

第二，依据 2.1 节所述，我们设计的数据管理接

口不涉及参数绑定、多个数据表格、复杂的数据关联和其他高级特性。在没有数据库引擎和组件支持的情况下, 新建数据表、更新修改表格数据、查询关联数据等操作需要编写复杂的逻辑程序。对于特定环境下的地面装备测试平台软件来说, 数据管理的主要工作是增加、查询和备份数据, 可以不安装和配置数据库组件以及操作数据库接口, 数据文件可以按照自定义的数据结构存储。

第三, 结构体中尽量不包含 QString 类型成员, 因为 QString 类型数据长度不固定, 会使搜索定位特定数据的逻辑运算变得复杂。

### 2.3 关键数据结构定义

综上所述, 数据文件<sup>[25]</sup>保存格式定义如下, 每一段数据记录由四部分组成: 数据包类型、数据包在文件中的起始位置、数据包总长度和数据正文。如图 1 所示, 图中数据文件标识即文件头主要用于区别数据文件和其他类文件, 以确定文件的有效性。

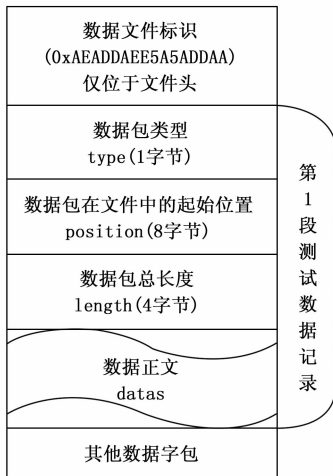


图 1 数据文件保存格式

考虑到“数据正文”部分是数据保存主体, 数据长度不固定, 所以将数据包类型 (4 字节)、数据包在文件中的起始位置 (8 字节)、数据包总长度 (1 字节) 三部分组成数据包索引结构体, 其定义见图 2, 字节数固定为 13 字节, 使用该数据结构可以搜索不同类型数据包的正文数据, 在此枚举 4 种数据包类型, 分别为: 测试信息 (INFO)、测试报告 (REPORT)、电源数据 (POWER) 和通信数据 (COMMUNICATION), 如图 3 所示。

### 2.4 关键数据处理方法

为了使数据查询工作简洁可靠, “数据正文”部分为不包含 QString 类型成员的长度固定的结构体数据或纯字符串 (QString), 如果字符串携带多个信息则以“,” 隔开。

以电源数据为例 (参见图 4), 其数据结构可定义

```
typedef struct RecordIndex{
    qint64 position;//数据包在文件中的位置
    uint length; // 数据包长度(字节数)
    uchar type; // 数据包类型
}RECORD_INDEX
```

图 2 数据段索引结构定义

```
typedefenum RecordType{
    INFO=0x0, //测试信息
    REPORT, //测试报告
    POWER, //电源数据
    COMMUNICATION//通信数据
}RECORD_TYPE
```

图 3 数据段类型枚举

为: 64 位整型时间戳 (time) 和单精度数组 (data)。其中时间戳用于记录电源数据采集的时间, 可精确到毫秒; 单精度数组则用于保存实时采集的电压或电流数据。

```
typedef struct PowerData{
    qint64 time;
    float data[6];
}POWER_DATA
```

图 4 电源数据结构

通信数据的数据结构 (图 5) 可定义为: 64 位整型时间戳 (time)、整型数据: channel、received、datalength。其中时间戳用于记录发送或接收通信数据的时间; 整型数据依次用于保存通信通道号、发送/接收和数据长度。

```
typedef struct Comm Data{
    qint64 time;
    int channel;
    int received;
    int datalength;
}COMM_DATA
```

图 5 通信数据结构

纯字符串数据主要用于保存说明性信息, 如图 6 所定义的 strInfo 即为此类数据。在此指出, QString 类型数据需要转换成 QByteArray 类型以获取其数据长度。

```
QString str Info=QString(“环境测试,
    产品1, 温度, 湿度, 地点, 张三”);
```

图 6 字符串数据

### 2.5 关键接口函数设计

1) 文件打开:

文件打开函数 open (), 可替代数据库访问时“创

建并打开数据库连接”操作。主要完成文件读取、索引信息搜索并保存成列表，以待后续数据查询和存储使用。为防止文件类型差错，在文件开始位置存有 8 字节特定标识：0xAEADDAEE5A5ADDA。没有该标识的文件为无效文件。从文件唯一标识开始搜索至文件尾，解析并保存索引信息至内存，为了节约内存占用，数据正文部分暂不获取和保存。

#### 2) 数据增加：

数据增加函数（等同于数据库 INSERT）即新增一条测试记录，可根据数据包类型分别封装成几个函数，2.3 节共枚举 4 种类型数据包，因此要定义 4 个数据增加函数 AddTestRecord（）、AddTestReport（）、AddPowerData（）、AddCommData（）。数据填充格式按 2.3 节所述，经计算给 RECORD\_INDEX 索引结构的 position、type、和 length 赋值，然后将填充好的数据直接写入文件末尾。QString 类型数据要转换成 QByteArray 类型并获取其数据长度保存在 RECORD\_INDEX 中的 length，然后再按格式打包数据添加至文件尾。

#### 3) 数据查询：

数据查询（等同于数据库 SELECT）需要根据数据包类型封装成几个函数，数据包有 4 种枚举类型，则定义 4 个函数：QueryRecord（）、QueryReport（）、QueryPower（）、QueryComm（）。在查询数据之前，必须先打开文件（调用 open 函数）获取索引信息列表，数据查询的过程是：

首先，根据信息类型（数据结构 RECORD\_INDEX 中的 type）在索引信息列表中查找索引信息。

然后，在数据文件中通过索引信息中的数据位置 position 和长度 length 查找并获取数据正文。

最后，解析数据正文保存至相应的结构体数组，提供给软件应用层使用。

#### 4) 数据删除：

数据删除（等同于数据库 DELETE），该操作根据 RECORD\_INDEX 结构体计算数据段在文件中的位置并从数据文件中删除一整段数据记录，同时遍历和重置后续数据段的 position，该操作是接口函数中比较复杂的程序，不适用于数据量大的软件项目。

#### 5) 数据更新：

数据更新（等同于数据库 REPLACE），该操作根据 RECORD\_INDEX 结构体的 position 计算并替换某一段数据记录的数据正文，替换的前提是两段数据的类型（type）一致。

#### 6) 数据备份：

当数据文件大于 500 M 时，重命名文件进行数据备份，同时新建原名数据文件，在文件开始位置保存 8 字节特定标识。

#### 7) 关闭文件：

关闭文件 close（）与打开文件是相对应的，实现起来相对比较简单，主要是调用函数释放文件和内存。

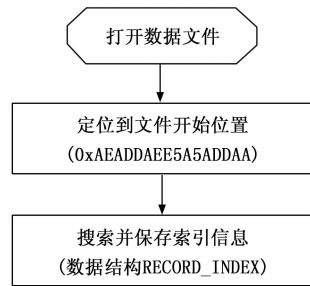


图 7 文件打开操作内容

## 2.6 编译

为了实现程序间共享、节省空间、方便调用、进而简化应用层软件设计，可以将数据管理接口类进一步生成动态链接库文件。使用 CMake 创建动态链接库，在 Windows 下会生成两个文件：“lib\*.dll”和“lib\*.a”，“.a”文件用于项目构建过程中添加动态库依赖，“.dll”是程序运行时加载的动态库；在 Linux 下创建动态链接库时，只会生成一个“lib\*.so”（shared object）文件，该文件用于程序运行时加载。生成动态链接库文件需要“.h”和“.c/.cpp”文件，编译库文件前先编写 Makefile 文件，然后在终端执行 make 命令<sup>[2]</sup>。在跨平台使用该接口类时，需要在相应的工程中添加库链接，程序运行时会在链接路径加载各自的动态链接库文件。

## 3 测试验证

本节使用 Qt 通过数据管理接口编写一个实例小程序，主要实现数据的读取、存储、查询和备份。该程序分别在 Intel+Windows、龙芯+中标麒麟系统环境跨平台编译运行。

### 3.1 测试背景

建立一个测试数据管理程序，实现测试信息的读取、保存、查询和备份功能。要求从可视化界面输入测试记录信息、电压电流信息、通信数据和测试报告信息供保存使用，通过关键字查询，可在界面表格显示符合查询条件的测试信息、电压电流数据和测试报告。其中测试记录信息包括：产品代号、编号、地点、温度、湿度、一岗、二岗；电压电流信息包括 6 路电压值、通信数据有 5 路值；测试报告信息包括测试项目、要求值、实测值、结论共 4 项内容。

### 3.2 开发过程

Qt 能够实现跨平台，主要依赖于 qmake 工具，该工具可以生成不同平台下的 MakeFile 文件，然后在不同 Makefile 下进行编译，而得到不同的可执行程序。目前，Qt5 是主流，也是最成熟的一代产品，Windows 系统中

Qt5 子版本有 Qt5.1 到 Qt5.15, 而 Qt6 是最新发布版本, 技术最新; 中标麒麟系统自带安装的应用开发环境是 Qt5.6.2 版。为了保证跨平台编译的一致性 (高版本 Qt 会增加新插件), 选择 Qt5.6.2 版完成案例开发。

软件编程主要包括两个部分, 可视化界面和后台程序处理。根据 3.1 节要求, 界面设计如图 8 所示。后台程序主要实现信息数据的读取、增加、查询及显示 4 种功能, 都可以利用数据管理接口实现。

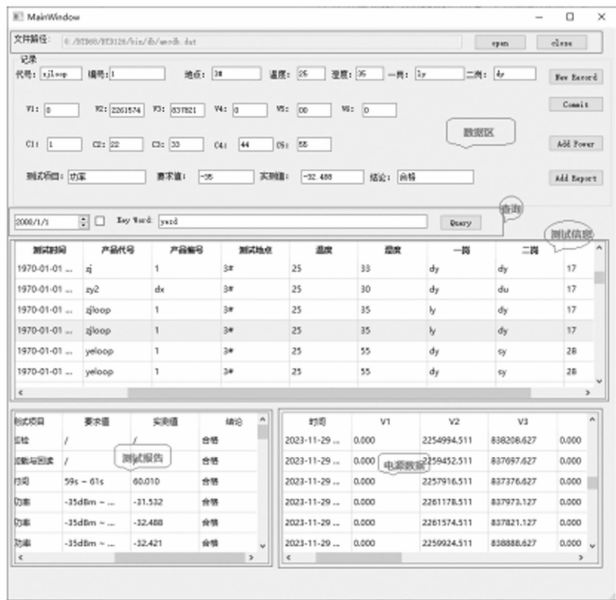


图 8 测试数据管理程序界面

数据管理接口可以引用源文件 (.cpp), 也可以导入库文件。利用 QT 的槽机制, 可将界面文件与用户逻辑代码文件有效隔离, 并将界面与函数功能关联起来<sup>[24]</sup>。这样, 在 Windows 和中标麒麟 (Linux) 下的界面文件和逻辑程序都能够跨平台使用, 不需要更改源码。在两种平台下我们只需配置工程环境重新编译运行就能轻松实现测试程序的移植。

### 3.3 验证过程

在 Intel+Windows 和龙芯+中标麒麟系统平台分别编译运行实例程序。

如图 7 所示, 界面点击“open”将弹出选择文件路径对话框, 选择接口文件 (如: xxx/amcdb.dat) 打开。输入“代号”“编号”等信息后, 点击按钮“New Record”新建一条测试记录; 界面输入“V1~V6”值, 点击“Add Power”保存电压电流信息; 界面输入“C1~C5”值点击“Commit”保存通信数据; 界面输入“测试项目”“要求值”等信息后, 点击“Add Report”保存测试报告信息。这些数据保存在文件“amcdb.dat”中。

保存多条测试数据后, 选择日期或在“Key Word”中输入关键字, 点击“Query”按钮后, 通过数据管理接口将符合条件的测试信息、测试报告和电源数据读出

并显示在相应表格中。

最后, 点击“close”关闭文件。当文件超过 500 M 后, 打开文件时备份原文件并更名为“amcdb1.dat”保存, 新建“amcdb.dat”文件以保存新数据。

### 3.4 验证结果

经验证, 选用跨平台软件 Qt5.6.2 开发的数据管理程序不需要更改界面源码和逻辑代码, 在 Intel+Windows、龙芯+中标麒麟系统环境下均可编译运行, 通过使用数据管理接口函数完全实现了数据的读取、增加、查询、备份等功能。

## 4 应用案例

在某装备地面测试平台软件国产化过程中, 使用我们设计的数据管理接口替代了传统数据库模块, 具体设计方案如图 9 所示。数据处理包括测试参数、测试信息、状态数据、过程数据和测试结果的记录与查询。

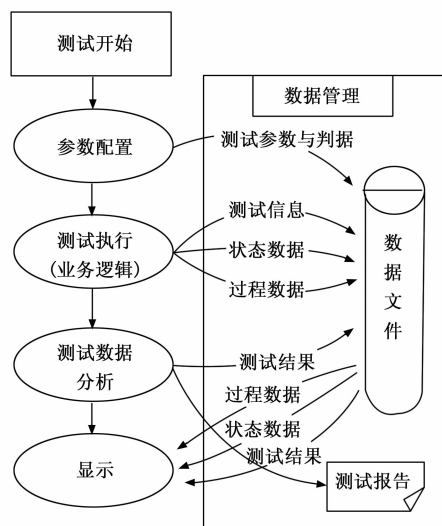


图 9 数据管理接口应用方案

根据功能需求, Windows 平台下该测试软件使用 Sqlite 数据库创建了 18 个数据表, 下面以“测试记录表” (表 1) 为例作数据库移植分析。

表 1 测试记录表

字段名称	字段类型	约束	说明
ID	INTEGER	是	唯一标识
STARTTIME	DATETIME	否	测试开始时间
ENDTIME	DATETIME	否	测试结束时间
TARGET_ID	VARCHAR	否	产品编号
PRODUCT_ID	VARCHAR	否	产品名称
LOCATION	VARCHAR	否	测试地点
EMPERATURE	DOUBLE	否	环境温度
HUMIDITY	DOUBLE	否	环境湿度
STATION_1ST	VARCHAR	否	测试一岗
STATION_2ND	VARCHAR	否	测试二岗

在龙芯平台的 QT 环境下只需要执行以下步骤即可完全实现数据库功能移植和替代:

1) 将数据管理接口的动态链接库添加到软件工程专业 .pro 中, 代码如下:

```
unix : LIBS += -LPWD/lib - lamcdb
```

2) 用数据管理接口函数替换原数据库模块操作, 如: 用函数 open ()、AddTestRecord () 替换建立数据库链接和表格插入数据 (INSERT) 操作; 用函数 QueryRecord () 替换数据查询 (SELECT) 操作;

3) 在龙芯+中标麒麟系统编译运行工程。

从硬件到软件全部完成国产化替代的装备地面测试平台经过多次拷机测试, 系统运行稳定, 平台软件在数据保存和查询过程中没有出现丢失数据和数据不精确现象, 测试结果能够准确完整的记录相关数据与相关过程。实际应用案例表明该数据管理接口有以下优点:

1) 独立封装的数据访问接口可替代传统数据库模块, 实现特定应用场景的数据增加、查询、删除、更新和备份工作;

2) 不使用数据库驱动和组件, 在 QT 环境下基于 C++ 设计开发, 可以在 Intel+Windows、龙芯+中标麒麟系统环境跨平台编译使用, 不需要考虑硬件平台、软件环境适配和数据库适配改造的问题;

3) 接口程序代码简单, 占用资源少, 且可以根据平台软件功能需要扩展和优化, 保证快速获取实时可靠的应用数据。

## 5 结束语

国产化数据库替代与移植研究对于减少应用软件开发人员工作复杂度、提高数据访问效率、节约成本、提升关键设备可靠性等方面具有重要的现实意义。以开源 Qt 为开发工具设计的数据管理接口, 初步解决了军用地面装备测试平台软件数据库跨平台移植所面临的问题, 实现了特定应用场景下的数据库基本特性, 能够在龙芯平台上替代传统的数据库组件, 为跨平台软件的数据库国产化替代提供了新的解决思路。

### 参考文献:

- [1] 刘康丽, 谷 静, 杜 影. 中标麒麟系统下基于 QT 的 PXI 仪器软件开发 [J]. 计算机测量与控制, 2019, 27 (10): 159-162.
- [2] 杜 影, 朱元元, 刘康丽. 基于龙芯 3A3000 与中标麒麟操作系统的 PXI 设备驱动设计 [J]. 计算机测量与控制, 2019, 27 (11): 163-166.
- [3] 高翔宇, 马林华. Linux 下 PCI 设备驱动开发方法及应用实例 [J]. 微计算机信息, 2007 (10): 310-312.
- [4] 萨师焯, 王 珊. 数据库系统概论 [M]. 4 版. 北京:

高等教育出版社, 2006.

- [5] 墨天轮中国数据库流行度排行 [EB/OL]. [2024-01-04]. <https://www.modb.pro/dbRank? 1213>.
- [6] OceanBase [DB/OL]. [2024-01-04]. <https://www.modb.pro/wiki/34>.
- [7] PolarDB [DB/OL]. [2024-01-04]. <https://www.modb.pro/wiki/37>.
- [8] openGauss [DB/OL]. [2024-01-04]. <https://www.modb.pro/wiki/601>.
- [9] TiDB [DB/OL]. [2024-01-04]. <https://www.modb.pro/wiki/20>.
- [10] 人大金仓 [DB/OL]. [2024-01-04]. <https://www.modb.pro/wiki/13>.
- [11] 达梦数据库 [DB/OL]. [2024-01-04]. <https://www.modb.pro/wiki/10>.
- [12] GBASE [DB/OL]. [2024-01-04]. <https://www.modb.pro/wiki/26>.
- [13] GaussDB [DB/OL]. [2024-01-04]. <https://www.modb.pro/wiki/46>.
- [14] TDSQL [DB/OL]. [2024-01-04]. <https://www.modb.pro/wiki/50>.
- [15] 龙芯中科技术有限公司, 龙芯 3A3000/3B3000 处理器用户手册 [Z]. 北京: 龙芯中科技术有限公司, 2017.
- [16] 芮 雪, 王亮亮, 杨 琴. 国产处理器研究与发展现状综述 [J]. 现代计算机, 2014 (3): 15-19.
- [17] 马 威, 姚静波, 常永胜, 等. 国产 CPU 发展的现状与展望 [J]. 集成电路应用, 2019, 36 (4): 5-8.
- [18] 中标软件: 操作系统国产化将迎来产业变革机遇 [J]. 软件产业与工程, 2014, (4): 24-8.
- [19] 吴金才, 张 辛, 吴勇军. 国产操作系统的技术特点及应用范围研究 [J]. 电脑知识与技术, 2015, 11 (4): 94-5.
- [20] LUO Y, LU Z. Feasibility analysis for application of domestic neokylin operating system in radar measurement and control system [C] //2017 2nd International Conference on Computer Engineering, Information Science and Internet Technology, 2017: 526-529.
- [21] 中标软件. 中标麒麟桌面操作系统 [EB/OL]. [2024-01-04]. <http://www.cs2c.com.cn/product /2024-01-04>.
- [22] 陈 光. 基于中标麒麟平台对达梦数据库访问技术研究 [J]. 计算机技术与发展, 2017 (12): 201-204.
- [23] 姜家文, 许荣胜, 胡振宇. 中标麒麟环境下基于 Qt 的神通数据库编程浅析 [J]. 软件工程, 2017, 201 (3): 18-20.
- [24] 陆文周. Qt5 开发及实例 [M]. 北京: 电子工业出版社, 2014.
- [25] 严蔚敏, 吴伟民. 数据结构 (C 语言版) [M]. 北京: 清华大学出版社, 2007.