

本科毕业论文(设计)

**论文题目：** **多用户小型数据库内核的设计与实现**

学生姓名： 张探探

学 号**：** 1931053726

专 业**：**计算机科学与技术

班 级： 计本1902

指导教师**：** 李薇

完成日期**：2022年5月25日**

**西安财经大学**

**本科毕业论文（设计）独创性及知识产权声明**

本人郑重声明：所呈交的毕业论文是本人在导师的指导下取得的成果，论文写作严格遵循学术规范。对本论文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。因本毕业论文引起的法律结果完全由本人承担。

特此声明

毕业论文签名：

专业：

学号：

年 月 日

**多用户小型数据库内核的设计与实现**

**摘 要**

从古至今重量级高效率系统软件内核都是国外程序员开发，内核关键代码对外不可见，而系统级软件不仅影响系统的稳定，还会影响安全。一旦系统内核存在“后门”，那么无论基于该内核的发行版如何修改,后门接口也不会消失,数据泄露、关键字段修改、甚至实施监听监控,所以急需一个开源稳定简单并且符合国人逻辑思维习惯的数据库内核。

当下主流数据库使用最频繁的是企业级Oracle，内核甚至发行版都是国外开发。并不开源。虽然稳定且高效，但是安全并不能保证。主流开源数据库有MYSQL,但它的缺点很明显，思维逻辑不符合国人，导致学习成本很高，且性能比企业级Oracle低。存在于部分政府使用的数据库SQLServer，一旦国外政府进行技术封锁，进而数据库限制中国使用，那么对于国内经济甚至于政府正常运转都会是一个降维打击。

TMDSQL基于C++，从底层文件入手,因为其前端解释语言设计并实现为中文，符合国人思维逻辑习惯，学习成本不高，并且包含对数据库及表的增删改查，视图层以及多用户权限管理也有涉及，实现多命令批处理。实现数据高效存储与查询。

通过大量测试，TMDSQL功能完善，运行流畅，可以有效的对大量数据进行存储和查询。同时完善日志记录，解释报错以及查询优化实现的较为完善。

**关键词：**内核；安全；C++；查询优化

**Design and Realization of Orientation Management System in Colleges and Universities**

Abstract

As the number of college admissions increases year by year, a large number of new students enter the school in the opening season of each year, and colleges and universities encounter various problems such as query difficulties in management. Therefore, the design and development of a WEB-based college orientation management system can improve the efficiency of management work from the root cause and provide better services for students.

Through the investigation of the registration process and related systems of freshmen in major colleges and universities, this paper designs and develops a college orientation management system. The system is based on B/S architecture, the front end uses JAVA language and JSP dynamic web page development technology, and the back end uses MySQL database to store data. The system is mainly divided into two parts: orientation service and orientation management. The orientation service provides students with relevant information inquiries, including viewing personal dormitory allocation and payment information; the orientation management section sets up different administrators to jointly manage multiple modules such as freshman registration, campus news, and online messages, so as to better build a smart campus, to provide high-quality orientation services for college freshmen.

Through the test, the functions of this system are running well, which can effectively improve the management efficiency of incoming freshmen. At the same time, it also provides convenient services for freshmen, and provides strong information support for the smooth development of the orientation work of colleges and universities.

**Key words:** College Orientation Management, Orientation Service, B/S Architecture,

Smart campus

目 录

[1 绪论 1](#_Toc103785984)

[1.1 研究背景 1](#_Toc103785985)

[1.2 研究目的和意义 1](#_Toc103785986)

[1.3 论文研究的主要内容 2](#_Toc103785987)

[1.4 论文结构 2](#_Toc103785988)

[2 系统实现相关技术介绍 3](#_Toc103785989)

[2.1 系统架构及技术简介 3](#_Toc103785990)

[2.1.1 B/S架构模式 3](#_Toc103785991)

[2.1.2 JSP语言 3](#_Toc103785992)

[2.2 MySQL数据库技术 3](#_Toc103785993)

[3 系统分析 4](#_Toc103785994)

[3.1 系统的可行性分析 4](#_Toc103785995)

[3.1.1 技术可行性 4](#_Toc103785996)

[3.1.2 操作可行性 4](#_Toc103785997)

[3.2 系统功能需求分析 4](#_Toc103785998)

[3.2.1 新生用户需求分析 4](#_Toc103785999)

[3.2.2 管理员需求分析 5](#_Toc103786000)

[3.3 UML用例分析 5](#_Toc103786001)

[3.3.1 活动者分析 5](#_Toc103786002)

[3.3.2 用例描述分析 7](#_Toc103786003)

[3.4 系统非功能需求分析 10](#_Toc103786004)

[3.5 系统业务流程分析 11](#_Toc103786005)

[3.5.1 新生用户业务流程 11](#_Toc103786006)

[3.5.2 管理员业务流程 12](#_Toc103786007)

[4 系统设计 13](#_Toc103786008)

[4.1 系统结构设计 13](#_Toc103786009)

[4.2 系统模块设计 13](#_Toc103786010)

[4.2.1 迎新服务模块 14](#_Toc103786011)

[4.2.2 迎新管理模块 17](#_Toc103786012)

[4.3 数据库设计 19](#_Toc103786013)

[4.3.1 数据库概念结构设计 19](#_Toc103786014)

[4.3.2 数据库逻辑结构设计 23](#_Toc103786015)

[5 系统开发与实现 28](#_Toc103786016)

[5.1 系统前端页面 28](#_Toc103786017)

[5.1.1 新生迎新服务页面 28](#_Toc103786018)

[5.1.2 登录页面 29](#_Toc103786019)

[5.1.3 在线留言页面 29](#_Toc103786020)

[5.2 系统管理员端后台 30](#_Toc103786021)

[5.2.1 用户角色管理 31](#_Toc103786022)

[5.2.2 校园新闻管理 32](#_Toc103786023)

[5.2.3 校园风光管理 33](#_Toc103786024)

[5.2.4 宿舍楼信息管理 33](#_Toc103786025)

[5.2.5 留言管理 34](#_Toc103786026)

[5.2.6 班级信息管理 35](#_Toc103786027)

[5.2.7 专业信息管理 35](#_Toc103786028)

[5.2.8 新生报到管理 36](#_Toc103786029)

[5.2.9 宿舍分配管理 36](#_Toc103786030)

[5.2.10 系统管理 37](#_Toc103786031)

[5.3 系统导员端后台展示 37](#_Toc103786032)

[5.4 系统财务端后台展示 38](#_Toc103786033)

[5.5 系统学生端后台 38](#_Toc103786034)

[5.5.1 学生后台切换 38](#_Toc103786035)

[5.5.2 学生后台主页面 39](#_Toc103786036)

[5.5.3 查看宿舍信息 39](#_Toc103786037)

[5.5.4 查看缴费记录 39](#_Toc103786038)

[6 系统测试 40](#_Toc103786039)

[6.1 系统测试 40](#_Toc103786040)

[6.2 测试的必要性 40](#_Toc103786041)

[6.3 测试过程 40](#_Toc103786042)

[6.3.1 主页面的登录测试 40](#_Toc103786043)

[6.3.2 迎新管理功能测试 41](#_Toc103786044)

[6.3.3 迎新服务功能测试 43](#_Toc103786045)

[6.4 测试结论 45](#_Toc103786046)

[7 总结与展望 46](#_Toc103786047)

[7.1 完成的主要工作 46](#_Toc103786048)

[7.2 下一步改进的方向 46](#_Toc103786049)

[参考文献 47](#_Toc103786050)

[致 谢 48](#_Toc103786051)

# 绪论

本章将从研究背景、研究目的和意义、论文研究的主要内容以及论文结构四个角度做系统介绍。

## 研究背景

从古至今重量级高效率系统软件内核都是国外程序员开发，内核关键代码对外不可见，而系统级软件不仅影响系统的稳定，还会影响安全。一旦系统内核存在“后门”，那么无论基于该内核的发行版如何修改,后门接口也不会消失,数据泄露、关键字段修改、甚至实施监听监控,所以急需一个开源稳定简单并且符合国人逻辑思维习惯的数据库内核。

当下主流数据库使用最频繁的是企业级Oracle，内核甚至发行版都是国外开发。并不开源。虽然稳定且高效，但是安全并不能保证。主流开源数据库有MYSQL,但它的缺点很明显，思维逻辑不符合国人习惯，导致学习成本很高，且性能比企业级Oracle低。存在于部分政府使用的数据库SQLServer，一旦国外政府进行技术封锁，进而数据库限制中国使用，那么对于国内经济甚至于政府正常运转将会是降维打击。

## 研究目的和意义

内核是高度精简的接口集，发行版基于内核二次开发，内核相对于发行版相当于心脏一样重要，而高度精简的内核，不仅要涉及不同情况下的接口，也要要求内核运行效率高，同时内核不能太过于庞大。这从技术角度来说并不容易。

如背景所言，安全是一个极为隐私的话题，同时及其容易被忽视，举个简单SQL注入的例子：如果正确的账号为:”admin”，正确的密码为:”passwd”,查询的用户数据库为”USERDATA”,一般登录所执行的语句是：SELECT \* FROM USERDATA WHERE username = ‘admin’ and passwd = ‘passwd’,如果此时用户输入的密码为’ or 1 = 1 #,那么原查询指令就会变成: SELECT \* FROM USERDATA WHERE username = ‘admin’ and passwd = ‘’ or 1 = 1 # ’,那么无论密码是什么，如果账号正确的话，均会返回一行，即后端就会认为登录成功。

安全是一个很重要的因素，随之而来出现了更严重的情况。因为政府大多使用的是国外付费数据库运维存储政府信息。一旦这些数据库存在后门，那么使用后门的黑客便可以轻而易举修改数据库中的数据。同时如果出现技术封锁，一旦限制使用，短期内找不到替代品，那对于中国将会是巨大的灾难。

## 论文研究的主要内容

本系统基于C++开发，不论是前端的解释语言的设计与实现，还是后端内核代码的实现均采取C++实现。

1. 前端部分： 该部分借鉴当前主流数据库基本功能---解释执行。因为数据库的需要高效的查询，故设计之初从编译型与解释型中选择了执行更为便捷且随输入随执行的解释型语言。支持34条不同的指令。包含设置索引，视图，基本增删改查等操作。
2. 后端部分： 主要是内核代码，将前端部分传回的中文指令提取关键字段，与相对应的接口连接，从文件角度完成数据库操作。

## 论文结构

第一章：绪论。在本章简单的说明高校迎新管理系统的研究背景、选题意义和目的以及论文的框架和主要研究内容。

第二章：介绍系统实现使用的相关技术。

第三章：系统分析。此章主要针对迎新管理系统进行分析，包括：系统的可行性分析，系统功能需求分析、非功能需求分析、业务流程分析。

第四章：系统设计。在基于需求分析上对系统进行设计，主要包括：系统结构设计、系统模块设计和数据库设计。

第五章：系统实现。按照系统的各功能模块进行开发，完善系统功能和界面设计，完成系统编码实现。

第六章：系统测试。根据系统的功能设计测试用例，完成对系统的测试。

第七章：总结与展望。对此系统的开发过程遇到的问题进行总结，提出下一步解决问题的研究方向。

# 2 编译版本与文件逻辑映射表

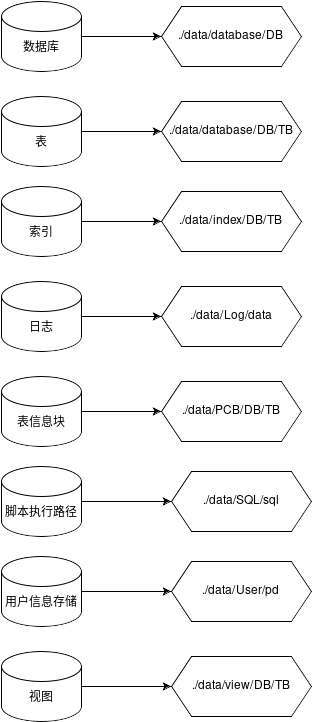
## 2.1 编译版本

本章将对TMDSQL使用的关键思路和高效的查询算法进行简单的介绍。系统所开发环境为Debain11系统，gcc的版本是gcc version 10.2.1 20210110 (Debian 10.2.1-6)。

## 2.2 发行版本与支持系统

所开发的版本基于Linux和Windows10双系统，开发windows10版本所使用的交叉编译链是x86\_64-w64-mingw32-g++。下载网址：<https://gitee.com/ttz_ztt/tmdsql/releases>

## 2.3 文件逻辑映射表

以下介绍逻辑功能：

数据库以文件夹的形式存储，以数据库名命名；数据库中存储的表以数据库文件夹内表的形式体现；

索引存储在固定路径./data/index内，其子目录结构和./data/database目录相同。在创建表的时候，系统内部会index内自动创建同级表以存储索引信息；

日志信息存储在./data/Log文件夹内，日志文件夹内的日志文件以当前日期区分，记录包括登录用户信息，所执行的指令以及成功与否，对错误执行的指令详细指出错误信息；

表信息控制块存储在./data/PCB文件夹内，存储包括独属于表的一些信息；

表的执行脚本被固定在./data/SQL文件夹内，需要将脚本文件放在该目录下；

用户信息包括用户名和密码，被存储在./data/User/pd文件内。没有采取加密，全部采用明文存储；

视图文件被存储在./data/view/文件夹内，与索引类似，其子目录与./data/database/目录类似。存储视图信息。

## 3 前端

### 3.1 体系介绍

系统中存储当前的路径，称为工作路径，设置工作路径后，可以省略参数。工作路径可以使用“显示”指令来显示当前的工作路径。工作路径分为三级：

1. 未设定工作路径

如果没有设置工作路径，那么在执行指令时无法执行省略指令。

2. 数据库级

如果将工作路径设置到数据库级别，那么可以执行省略数据库参数的部分省略指令。

3. 表级

如果将工作路径设置到表级别，那么可以执行省略表参数的部分省略指令。

设置工作路径可以使用“选择”指令来设置，并用“显示”指令来显示当前的工作路径。

视图对于普通用户默认为空，需要管理员或Root添加允许的视图列。

数据库中每条指令前，实现了对应的权限提示符，分为四个等级：

1. 未登录

其权限提示符为: (未登录) -=>

1. 用户  
    其权限提示符为: (用户) -=>
2. 管理员  
    其权限提示符为: (管理员) -=>
3. 超级管理员  
    其权限提示符为: (超级管理员) -=>

以下介绍指令功能时，由于未知权限，则提示符均以 (?) -=> 代替。

### 3.2 支持指令集

1. 退出
2. 选择 数据库 @测试数据库1
3. 选择 表 @测试表1
4. 选择 数据库 表 @测试数据库1 测试表1
5. 显示
6. 显示 数据库
7. 显示 数据库 @DBID(=pwd[1])
8. 显示 表 @showNumber
9. 显示 表 @TBID showNumber
10. 显示 数据库 表 @DBID TBID showNumber
11. 登录 @帐号 密码
12. 执行 @脚本名
13. 创建 用户 @ID Password
14. 创建 管理员 @ID password
15. 创建 数据库 @DBID
16. 创建 数据库 表 @DBID TBID
17. 创建 表 @TBID
18. 删除 数据库 @DBID
19. 删除 表 @TBID
20. 删除 数据库 表 @DBID TBID
21. 删除 用户 @UserName
22. 删除 管理员 @managerID
23. 插入 [value1 value2 ...]
24. 插入 表 @TBID [value1 value2 ...]
25. 插入 数据库 表 @DBID TBID [value1 value2 ...]
26. 查询 @index
27. 查询 表 @TBID index
28. 查询 数据库 表 @DBID TBID index
29. 设置 视图 数据库 表 @DBID TBID UserName [value...]
30. 设置 视图 表 @TBID UserName [value...]
31. 设置 视图 @UserName [value...]
32. 设置 索引 数据库 表 @DBID TBID indexColumn
33. 设置 索引 表 @TBID indexColumn
34. 设置 索引 indexColumn

### 3.3 详细介绍

* 退出  
  使用退出指令，会结束SQL执行。并输出:“ (?) -=> 已退出”。
* 选择 数据库 @测试数据库1
  + 如果测试数据库1不存在，输出“(?) -=> 目标数据库不存在”。
  + 如果数据库存在，则将工作路径设置为目标数据库。
* 选择 表 @测试表1
  + 如果没有选择数据库，则无法越级选择表，提示：“(?) -=> 该指令无法执行，没有选择数据库！”。
  + 如果已经选择数据库，但已经选择的数据库中不存在测试表1，则无法执行，提示：“(?) -=> 无法执行该指令，该数据库中不存在此表”。
  + 如果一切正常，则将工作路径设置为目标表。
* 选择 数据库 表 @测试数据库1 测试表1
  + 如果测试数据库1不存在，输出“(?) -=> 目标数据库不存在”。
  + 如果数据库存在：
    - 如果没有选择数据库，则无法越级选择表，提示：“(?) -=> 该指令无法执行，没有选择数据库！”。
    - 如果已经选择数据库，但已经选择的数据库中不存在测试表1，则无法执行，提示：“(?) -=> 无法执行该指令，该数据库中不存在此表”。
    - 如果一切正常，则将工作路径设置为目标表。
* 显示
  + 使用显示指令，根据当前的工作路径（PWD）分为以下四种情况。
    - PWD=[],则显示:”(未登录) -=> 未登录，请登录后操作!”
    - PWD=[“\”],则显示:”(?) -=> 当前路径为: /”
    - PWD=[“\”][“测试数据库1”]，则显示:”(?) -=> 当前路径为: /测试数据库1”
    - PWD=[“\”][“测试数据库1”][“测试表1”]，则显示:”(?) -=> 当前路径为:/测试数据库1/测试表1”
* 显示 数据库
  + 该指令需要登录后执行，显示当前系统中所有数据库。假设当前数据库内有DB1,DB2,DB3三个数据库，则使用该指令后显示:

(?) -=>

#######

？DB1 ？

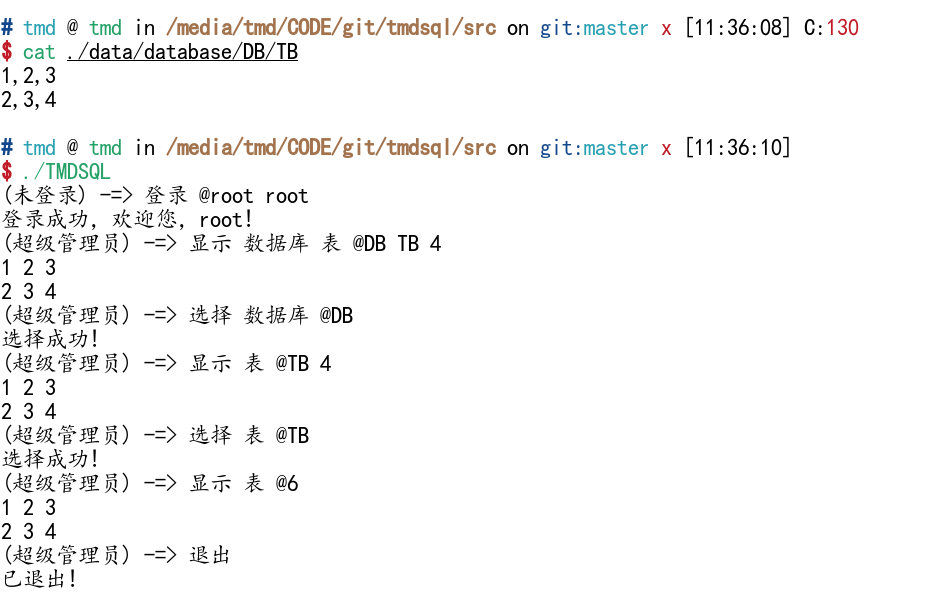
? DB2 ?

? DB3 ?

#######

* 显示 数据库 @DBID(=pwd[1])
  + 该指令需要登录后使用，可以选择到表后使用该指令，效果可以显示指定数据库中所存的表。
  + 当数据库不存在时,会提示: (?) -=> 数据库不存在
  + 当数据库DBID为空时，会提示: (?) -=> 这个数据库是空的
  + 当这个数据库有TB1,TB2,TB3三张表的时候：  
    (?) -=>  
    \*\*\*\*\*\*\*\*  
    ？TB1 ？  
    ? TB2 ?  
    ? TB3 ?

\*\*\*\*\*\*\*\*

* 显示 表 @showNumber
  + 该指令需要首先选择数据库且选择表，显示showNumber行。
  + 执行如图所示。
* 显示 表 @TBID showNumber
  + 该指令需要选择数据库，将会显示showNumber行数据。如果实际数据不足showNumber，那么将会全部显示。
  + 执行如图所示
* 显示 数据库 表 @DBID TBID showNumber
  + 该指令不需要选择数据库，也不需要选择表，如果数据库不存在，或DBID中TBID不存在，都会使得该指令执行错误。报错同上。
  + 执行如图所示
* 登录 @帐号 密码
  + 如果帐号和密码均正确，则会输出：(?) -=> 登录成功，欢迎您，帐号!
  + 登录失败
    - 如果帐号不存在:则会输出: (?) -=> 帐号不存在，登录失败！
    - 如果密码错误，则会输出: (?) -=> 密码错误，登录失败！
* 执行 @脚本名
  + 脚本必须按照UTF-8编码，因为该数据库的解码方式是按照UTF-8解码，其他格式文件可能解码失败。
  + 注意: 需要将脚本放置于./data/SQL目录下，然后使用如上指令，脚本名需要是放置在SQL的全称，带后缀的那种。如果代码中间存在空行，会跳过空行后的内容，结束执行
  + 如果当前目录不存在该脚本,则会输出: (?) -=> 无法执行该指令，目标脚本不存在!
* 创建 用户 @ID Password
  + 该指令需要OP < 2,也就是需要管理员权限或者超级管理员ROOT权限才能执行。该指令会添加密码为Password的普通用户ID
  + 当用户ID存在时，提示:(?) -=> 该用户已存在，无法添加!
  + 创建成功后提示: （?) -=> 创建成功!
* 创建 管理员 @ID password
  + 该指令需要OP < 1,也就是需要超级管理员ROOT权限才能执行。会添加密码为password的管理员ID。
  + 当用户ID存在时，提示: (超级管理员) -=> 该用户已存在，无法添加!
  + 创建成功后提示: （超级管理员) -=> 创建成功!
* 创建 数据库 @DBID
  + 该指令需要OP < 2,也就是管理员或超级管理员ROOT权限才可以执行。会创建名为DBID的数据库。
  + 如果DBID存在，则提示:（?) -=> 无法执行该指令，目标数据库已存在。
  + -如果DBID不存在，则执行成功，提示:?) -=> 创建成功!
* 创建 数据库 表 @DBID TBID
  + 该指令需要OP < 2,也就是管理员或超级管理员ROOT权限才可以执行。会创建名为TBDI的表。
  + 如果TBID存在，则提示:（?) -=> 无法执行该指令，目标表已存在。
  + -如果TBID不存在，则执行成功，提示: (?) -=> 创建成功!
* 创建 表 @TBID
* 该指令需要OP < 2，也就是管理员或超级管理员权限才可执行。但在此之前需要选定数据库。
* 如果没有选择数据库的话会输出: (?) -=> 该指令无法执行，没有选择数据库!
* 如果已经选择数据库，但表TRID已存在，那会提示: (?) -=> 无法执行该指令，目标表已存在!
* 如果一切顺利，则显示: (?) -=> 创建成功!
* 删除 数据库 @DBID
  + 该指令会删除名为DBID的数据库。执行首先会判断DBID存在与否，如果存在，系统将首先输出该数据库中存在的所有表，然后需要用户键入"确认"才可删除。
  + 如果数据库DBDI不存在，则会提示:目标数据库不存在!
  + 如果DBID是一个空数据库，那么就直接删除，并不需要输入确认:(?) -=> 删除成功!
  + 如果DBID是一个非空数据库，且其中有表TB1，TB2，TB3和TT三张表，如果执行指令，则显示:  
     (?) -=>

数据库并不为空，以下为当前存在的表项

\*\*\*\*\*\*\*

| TB1 |

| TB2 |

| TB3 |

| TT |

\*\*\*\*\*\*\*

如果您仍然执意删除，请键入 "确定" :

此时你需要在':'后键入确定:

* + - 如果你没有输入确定而输出别的(包括只输入回车)，则会提示:(?) -=> 该指令已取消!
    - 如果你输入确定,则会提示:(?) -=> 删除成功!
* 删除 表 @TBID
  + 该指令会删除名为TBID的表，当然执行该指令的前提是你首先得选择到数据库，系统会判断当前pwd的路径，如果pwd.size() > 2,且pwd[1]中有该表，系统会输出该表当前的前五行，然后同样需要用户键入"确认"才可删除。
  + 如果你没有选择数据库，则会提示：(?) -=> 该指令无法执行，没有选择数据库!
  + 如果已选数据库，但pwd[1]中没有表TBID,则会提示：(?) -=> 目标表不存在!
  + 如果存在，但表为空，执行会提示:(?) -=> 删除成功!
  + 如果存在，且不为空的情况下，执行后提示:  
    (?) -=>

表不为空，以下是该表的一部分内容:

测试1 测试2 测试3

测试1 测试2 测试3

如果您仍然执意删除，请键入 "确定" :

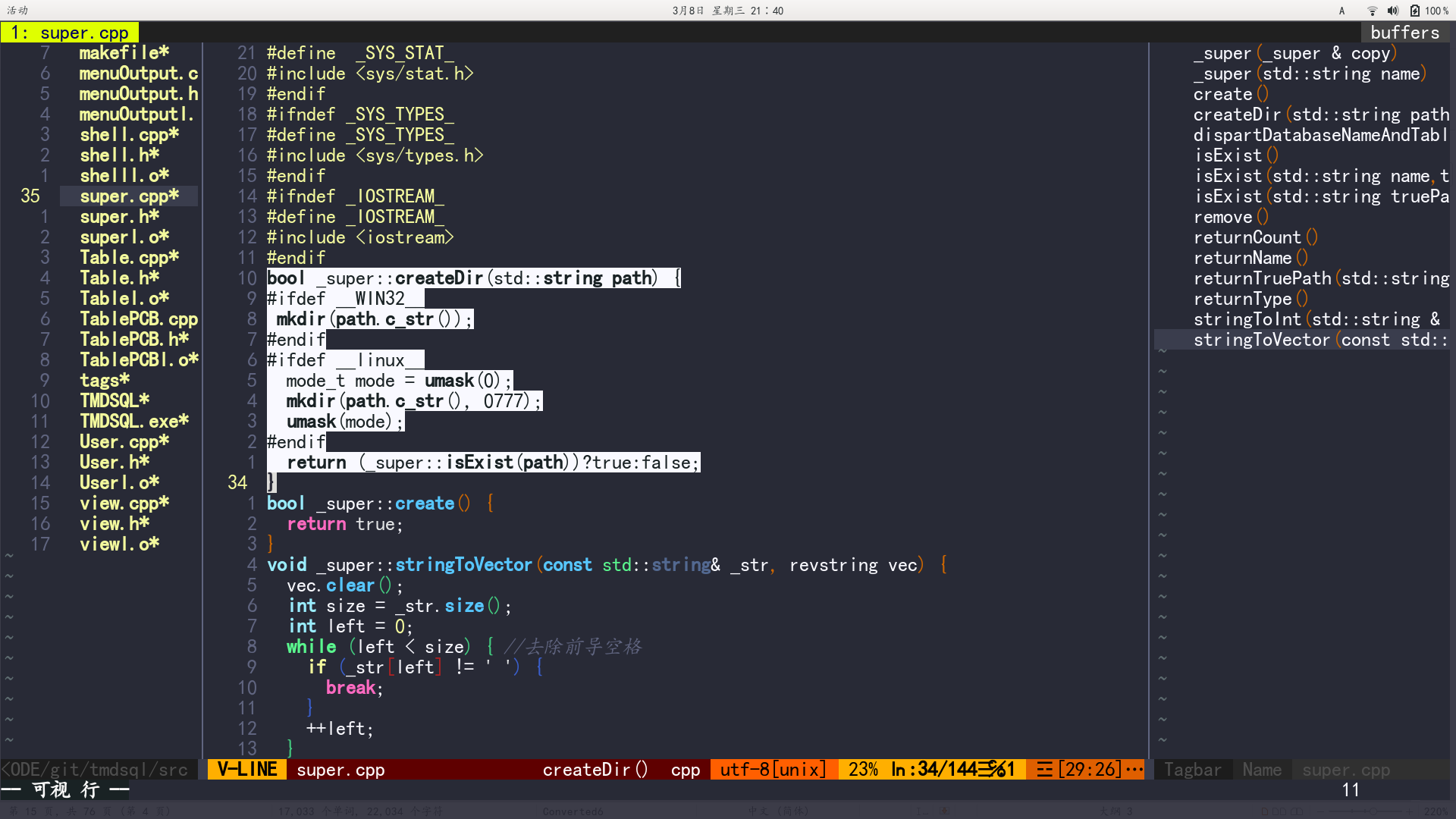
* + - 如果你没有输入确定而输出别的(包括只输入回车)，则会提示:该指令已取消!
  + 如果你输入确定,则会提示:删除成功!此时你需要在':'后键入确定:
    - 如果你没有输入确定而输出别的(包括只输入回车)，则会提示:该指令已取消!
    - 如果你输入确定,则会提示:删除成功!
* 删除 数据库 表 @DBID TBID
  + 该指令会删除DBID下的TBID，首先判断DBID是否存在，且是否有TBID这张表，其实和上一个指令差不多，区别是将上一个指令中默认的pwd[1]修改为DBID。
* 删除 用户 @UserName
  + 该指令会删除名字为UserName的普通用户。
  + 当UserName不存在时，提示:(?) -=> 该用户不存在!
  + 当与用户存在的时候，提示:(?) -=> 删除成功!
* 删除 管理员 @managerID
  + 该指令需要超级管理员ROOT权限才能执行。
  + 当managerID不存在时，提示:(?) -=> 该管理员不存在!
  + 当与用户存在的时候，提示:(?) -=> 删除成功!
* 插入 [value1 value2 …]
  + 该条指令需要用户选择到表，一切顺利的情况下，尾插value1，value2等数据。
* 插入 表 @TBID [value1 value2 …]
  + 该条指令会在选择数据库，且该数据库内有TBID后可执行，尾插value1，value2等数据。
  + 当未选择数据库的时候，会提示:(?) -=> 该指令无法执行，没有选择数据库!
  + 当已选择数据库，但pwd[1]中没有该表，则会显示:(?) -=> 目标表不存在!
  + 当一切顺利，则会显示:(?) -=> 插入成功!
* 插入 数据库 表 @DBID TBID [value1 value2 …]
  + 该指令是上一条的扩展，无需先选择数据库，只要参数正确，指定数据库下的表尾插入指定集合即可。
  + 当数据库DBID不存在时，会提示:(?) -=> 目标数据库不存在!
  + 当表不存在的时候,会提示:(?) -=> 无法执行该指令，该数据库中不存在此表!
  + 当一切顺利，则会显示:(?) -=> 插入成功!
* 查询 @index
  + 需要选择数据库，选择表，即pwd.size() > 2
* 查询 表 @TBID index
  + 需要选择数据库，即pwd.size() > 1,pwd[1]中有表TBID，查询索引为index的内容。
  + 如果执行前没有选择数据库，则会显示:(?) -=> 该指令无法执行，没有选择数据库!
  + 如果选择数据库后，但目标数据库内没有表TBID，则会显示：(?) -=> 目标表不存在!
  + 如果TBID存在，但index并不在表TBID中，则会显示:(?) -=> 搜索结束，没有搜到目标值!
  + 如果index存在，则会返回搜索到的值。
* 查询 数据库 表 @DBID TBID index
  + 不需要选择数据库，属于权限最小的一系列指令，只需登录即可执行。查询DBID下的TBID中索引为index的内容。
  + 数据库DBID不存在,则会输出:(?) -=> 目标数据库不存在!
  + 数据库DBID存在，但是表TBID不存在，则会输出:(?) -=> 目标表不存在!
  + 数据库DBID存在且表TBID存在，但是没有搜到Index，则会输出：(?) -=> 搜索结束，没有搜到目标值!
  + 如果搜到，则会正常显示搜索内容。
* 设置 视图 数据库 表 @DBID TBID UserName [value...]
* 设置 视图 表 @TBID UserName [value...]
* 设置 视图 @UserName [value...]
* 设置 索引 数据库 表 @DBID TBID indexColumn
* 设置 索引 表 @TBID indexColumn
* 设置 索引 indexColumn

### 

# 4. 后端

4.1 文件管理层

Windows系统与linux系统略有不同。

 在windows系统中，由于windows是属于单用户多线程系统，所以C++为其提供mkdir函数，便可以直接创建文件夹。所创建的文件夹的权限与上一级权限相同。

但linux系统是属于多用户多线程系统，OS对权限以及用户组要求及其严格，某一用户创建的文件夹存在属主，属组和其他组概念，如果设置错误，不仅可能A用户创建的文件夹B用户无法打开。甚至有可能A用户创建的文件夹连自己也无法打开。所以C语言对于mkdir增添一个mode参数，以数字方式设定文件夹权限。同时新增一个概念，权限掩码，与子网掩码类似，由umask设定。其最终设定的文件夹权限值为：

mode&~mask

4.2 权限管理层

4.3 文件控制块PCB

4.4 代码解释层

4.5 脚本层

5. 日志层

6. 视图层

7. 索引查询

8. Cache查询

# 总结与展望

## 完成的主要工作

本文旨在通过建立高校迎新管理系统，简化新生报到流程并提高校管理工作的效率，更好的服务于高校的学生。在一定程度上为建设智慧校园进行了探索和研究，主要完成了以下工作：

1）对高校迎新系统进行调查和研究，确定实现系统的技术方案；

2）明确需求。基于学校迎新管理工作的痛点，对系统进行需求分析，明确系统实现的功能，设置不同的管理员角色共同管理迎新工作，对新生提供多元化的查询服务；

3）实现系统。根据需求分析完成对系统模块的设计与实现，编码过程中提高代码的复用性和可读性，逐一实现各模块功能；

4）对系统进行测试。针对系统功能设计测试用例，检验系统存在的问题，是否符合开发的目标，完成对系统的调试，保障系统良好的运行与使用。

同时，本系统注重迎新服务，为学生提供了很多模块进行查询，操作简单、实用性较强；后期的维护成本较低；也具有较强的可移植性。

## 下一步改进的方向

由于自己技术水平的有限，对于项目开发的实战经验还存在不足，开发的系统还存在一些漏洞，例如不同学院的要求制度可以设立不同学院的迎新报到入口；对于用户管理员权限和管理模式的设置，与实际应用还具有一定的距离；由于新生信息较多，目前只针对新生的信息做以筛选的标签，希望之后改进通过筛选标签完成图表可视化，这样可以更直观的反映高校新生的一些数据。

在之后的学习和研究中，不断完善高校迎新系统，更加深入了解新生报到和学校管理的困难，能够从多维度解决问题，优化系统同时达到改进目标。希望系统能够实际投入高校迎新工作的使用，实现新生入校快捷的流程，提高学校管理的工作效率。

# 参考文献

1. 高洋.经济背景下我国高校迎新系统的分析与设计[J]. 营销界, 2020(35):96-98
2. 黄浩扬.智慧校园下的迎新微服务的开发与应用[J]. 计算机产品与流通 ,2019(05):90
3. 滕华, 侯镇锋, 高哲. 高校智慧迎新系统的建设与探索[J]. 现代信息科技, 2020.
4. 王乾坤.基于BS架构管理系统的设计问题与对策初探[J]. 数码设计, 2018, (5):10-11.
5. Liu, Youjie,Shabaz, Mohammad. Design and research of computer network micro-course management system based on JSP technology[J]. International Journal of System Assurance Engineering and Management,2021(prepublish).
6. 林殷才.迎新管理与服务系统设计与实现[D]. 南京邮电大学, 2020.
7. 黄旭,圣文顺,李会.数据库设计的重要性及原则[J].网络安全技术与应用,2019(08):74-75.
8. Honghui Li,Aihua Zhao,Dalin Zhang,Junwen Zhang. Research on building software usage model based on UML model[J]. International Journal of System Assurance Engineering and Management,2018,9(3).
9. 陈宏媛.基于UML用例图的三维车体钢结构设计系统需求分析[J].自动化应用, 2021(09):154-157.
10. 张敏.新时代Web前端页面设计的实践[J].无线互联科技, 2021, 第18卷(15):35-36.
11. 吴家皋,吴敏,周剑,刘林峰.基于E-R图的工程教育专业认证课程达成度评价方法研究[J].计算机教育,2020(11):151-156.
12. 邢晓伟.软件测试方法研究[J]. 缔客世界, 2020, 第6卷(5):50.
13. 张雅东,张清淘.软件工程中软件有关测试技术的思考[J].石河子科技,2022(01):17-19.

# 致 谢

时间在不知不觉中流逝，转眼间我也将迎来毕业，心中有不舍、有遗憾、有感激。这四年是值得怀念的，也见证了我一步步的成长。

首先要感谢父母对我的抚养与支持，让我有信心做出每个决定。其次要感谢在大学里遇到的老师和同学们，老师在授业的同时为我指点迷津，明晰方向，特别是我的导师李薇老师，在我就业迷茫时、学习无措时，给予了我信心和勇气。在科技联盟组织结识了良师益友，组织承办不同的竞赛和活动的同时，也让我积累经验参加各类竞赛并取得了优秀的成绩，让我逐渐成长收获颇丰。

我将不惧困难，朝着既定的方向努力奋斗。未来的路还很长，我将永远怀有热爱坚定的走下去!