**选课系统的实现与总结**

**摘要：**本学期『网络编程』的期末作业是在实现基本要求的基础上，完成一个选课系统。本篇内容，我会以个人完成的作业项目为基础，围绕如何实现课程要求、实现过程中遇到的难点和解决办法、项目总结等方面，做一个简单概述。并且还会对线程同步、防死锁策略、数据库安全三大重要内容着重展开讨论。

本学期『网络编程』的期末作业是在实现基本要求的基础上，完成一个选课系统。本篇内容，我会以个人完成的作业项目为基础，围绕如何实现课程要求、实现过程中遇到的难点和解决办法、项目总结等方面，做一个简单概述。并且还会对线程同步、防死锁策略、数据库安全三大重要内容着重展开讨论。

**关键字：**选课系统；多线程；死锁预防；数据库安全；

# 作品介绍

本次『网络编程』的期末作业项目，需要在实现基本要求的起初上完成一个选课系统。需要实现的基本功能有如下：本次『网络编程』的期末作业项目，需要在实现基本要求的起初上完成一个选课系统。需要实现的基本功能有如下：

* 项目的实现**必须是Java程序**。因此本次作业呈现采用的是 JavaWeb 的形式，页面显示的内容交由 html 完成，核心的通信、多线程、防死锁诸多问题以及选课部分交由 Java 完成。项目的实现**必须是Java程序**。因此本次作业呈现采用的是 JavaWeb 的形式，页面显示的内容交由 html 完成，核心的通信、多线程、防死锁诸多问题以及选课部分交由 Java 完成。
* 程序必须进行**身份认证**。此处理解的身份认证即使用系统必须进行账号登录。系统分设有客户端和服务器端，无论进入哪个部分，都会先对账号密码进行校验，再判断是否拥有进入对应系统的权限。
* 程序需要使用**多线程**。我在选课部分使用到了多线程，为每一个学生的每一次选课任务创建一个线程，交由线程进行选课。而课程余量相关的问题则涉及到了线程同步，通过synchronized关键字实现线程同步，确保选课人数不会超出课程容量。
* 程序需要**使用 URL URLConnection**，主要完成对目标资源的连接。因此在客户端即学生选课端获取选课信息处，使用了 URLConnection 与目标资源的连接，使用 POST 的方式获取信息并显示。
* 程序需要**使用 Socket**，完成的两个进程之间的双向通信。因此在客户端即学生选课系统端获取个人完整信息处，使用了 Socket 与服务器端建立连接进行双向通信，从而获取学生个人的完整信息。
* 程序需要考虑**数据库安全**，程序中涉及到对学生和课程信息的管理，因此需要对数据库的安全采取措施以免受到污染破坏。

## 服务器端

* 课程数据库：内含课程编码、课程名称、课程简介、课程适用专业、开设学期、人数限定、选课资格限定。数据库中创建了表 `course` 来记录课程数据，字段间的关系如下表：

|  |
| --- |
|  |
| **字段名称** | **字段类型** | **字段含义** | **是否主键** | **是否为空** | **备注** |
| code | varchar(20) | 课程编码 | 是 |  |  |
| name | varchar(20) | 课程名称 |  |  |  |
| summary | varchar(20) | 课程简介 |  | 是 |  |
| profession | varchar(20) | 适用专业 |  | 是 |  |
| semester | varchar(20) | 开设学期 |  |  |  |
| number | int | 人数限定 |  |  |  |
| qualifications | varhcar(20) | 选课资格 |  | 是 |  |

* 学生数据库：内含学号、姓名、所在学院、专业、联系方式、已修课程、新选课程、认证信息。数据库中创建了表 `account` 来记录学生的账号信息，对应关系如下表:

| **字段名称** | **字段类型** | **字段含义** | **是否主键** | **是否为空** | **备注** |
| --- | --- | --- | --- | --- | --- |
| id | varchar(20) | 学号 | 是 |  |  |
| password | varchar(20) | 密码 |  |  |  |
| name | varchar(20) | 姓名 |  |  |  |
| college | varchar(20) | 学院 |  |  |  |
| profession | varchar(20) | 专业 |  |  |  |
| contact | varchar(20) | 联系方式 |  | 是 |  |
| certification | varchar(20) | 认证信息 |  | 是 |  |

* 接收远程客户端的接入申请、用户验证。客户端想要登录，需要先于服务器端建立连接，连接成功以后，通过对账号密码的验证，才能成功进入客户端。
* 客户的获取数据进行显示是需要先与服务器端进行连接，服务器端会允许远程客户端的连接申请，再进行对应的数据传输。
* 管理人员对课程数据库的修改，即拥有新设课程、管理课程的权力。管理人员即教室，可以登录教师端，对课程进行添加、修改、删除，对学生进行添加、编辑信息，对学生的课程选择进行通过等处理。
* 对每位选课的学生建立线程，并且考虑人数上限的处理。，每一位学生的每一次选课都会创建一个线程，同时进行计数，当同一时间内线程数量到达一个阈值，就会暂时停用选课功能。

## 客户端

* 通过 HTTP 协议登录选课系统。
* 提供选课人员的身份信息。学生在登录系统以后，会有单独的页面与服务器端建立连接，获取该学生的完整信息。
* 浏览可选课程、拟选课程。

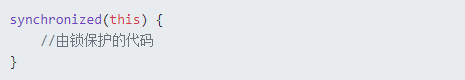
# 难点解决

## 线程同步

线程同步：即当有一个线程在对内存进行操作时，其他线程都不可以对这个内存地址进行操作，直到该线程完成操作， 其他线程才能对该内存地址进行操作，而其他线程又处于等待状态，实现线程同步的方法有很多，临界区对象就是其中一种。

线程同步的作用：线程有可能和其他线程共享一些资源，比如，内存，文件，数据库等。当多个线程同时读写同一份共享资源的时候，可能会引起冲突。这时候，我们需要引入线程“同步”机制，即各位线程之间要有个先来后到，不能一窝蜂挤上去抢作一团。

多线程的并发执行可以提高程序的执行效率,提高系统资源的利用率。然而在某些情况下,当多个线程同时访问一个变量时,如果处理不当,很可能发生混乱。而多线程同步机制的引入则可以保证多个线程可以独立运行、互不干扰[1]。



在本次作业中，我采用的同步代码块的方式来实现线程同步，从而保证课程的正常选取。

* synchronized关键字通过一把锁锁住一段代码, 使得线程只有在持有锁的时候才能访问这段代码。
* 只有强盗锁的线程才可以执行同步代码块，其余的线程即使抢到了CPU执行去哪，也只能等待，等待锁的释放。
* 代码执行完毕或者程序抛出异常都会释放锁，然后还未执行同步代码块的线程争抢锁，抢到的线程就能运行同步代码块。

## 死锁预防

死锁是指两个或两个以上的进程在执行过程中，由于竞争资源或者由于彼此通信而造成的一种阻塞的现象，若无外力作用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁，这些永远在互相等待的进程称为死锁进程。

1971年, Coffman提出了死锁发生的四个条件, 即互斥访问条件, 请求和保持条件, 不可抢占条件, 环路等待条件。只有当这四个条件同时成立时, 才有可能会出现死锁[2]。

死锁产生的条件：

1）互斥条件：指进程对所分配到的资源进行排它性使用，即在一段时间内某资源只由一个进程占用。如果此时还有其它进程请求资源，则请求者只能等待，直至占有资源的进程用毕释放。

2）请求和保持条件：指进程已经保持至少一个资源，但又提出了新的资源请求，而该资源已被其它进程占有，此时请求进程阻塞，但又对自己已获得的其它资源保持不放。

3）不剥夺条件：指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。

4）环路等待条件：指在发生死锁时，必然存在一个进程——资源的环形链，即进程集合{P0，P1，P2，···，Pn}中的P0正在等待一个P1占用的资源；P1正在等待P2占用的资源，……，Pn正在等待已被P0占用的资源。

死锁的预防：

只要打破四个必要条件之一就能有效预防死锁的发生：

* 打破互斥条件：改造独占性资源为虚拟资源，大部分资源已无法改造。
* 打破不可抢占条件：当一进程占有一独占性资源后又申请一独占性资源而无法满足，则退出原占有的资源。
* 打破占有且申请条件：采用资源预先分配策略，即进程运行前申请全部资源，满足则运行，不然就等待，这样就不会占有且申请。
* 打破循环等待条件：实现资源有序分配策略，对所有设备实现分类编号，所有进程只能采用按序号递增的形式申请资源。

## 数据库安全

目前计算机网络数据库中包含着大量的信息，其中涉及各行各业的信息，同时也会对许多利益相关者产生影响，那么，就需要不断强化计算机网络数据库的安全管理，保证互联网安全运行，还能够让计算机的运行更加高速。互联网数据信息技术的飞速发展，为社会的发展科技的进步，提供了必要的科技支持，也成为人们赖以生存的技术手段。新时代下物联网在不断融入人们的生活生产之中成为各种数据信息处理的工具，利用计算机信息技术让我国未来的科技发展有更强大的生命力。新时代下物联网在不断融入人们的生活生产之中成为各种数据信息处理的工具，利用计算机信息技术让我国未来的科技发展有更强大的生命力[3]。

数据库系统的安全特性主要是针对数据而言的，包括数据独立性、数据安全性、数据完整性、并发控制、故障恢复等几个方面。

可以使用如下几方面的安全策略：

* 使用安全的密码策略
* 使用安全的账号策略
* 加强数据库日志的记录
* 管理扩展存储过程
* 使用协议加密
* 不要让人随便探测到您的TCP/IP端口
* 修改TCP/IP使用的端口
* 拒绝来自1434端口的探测
* 对网络连接进行IP限制

# 参考文献

1. 彭娇.Java多线程同步机制在网络售票系统中的应用[J].科技资讯,2016,14(29):18-19.
2. 任彦冰,顾健.一种操作系统的死锁预防算法[J].电子技术与软件工程,2015,(10):175.
3. 唐德浩.网络安全要求下的数据库安全技术分析[J].网络安全技术与应用,2021,(12):54-55.