**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。

。。。

## 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。

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。