**《操作系统》课程实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | 操作系统 | 班级 | B220111 | 实验日期 | 2023/03/31 |
| 姓名 | 陈龙 | 学号 | B22011108 | 实验成绩 |  |
| 实验名称 | 线程同步与互斥 | | | | |
| 实验目的及要求 | 1. 掌握Java线程的创建 2. 理解共享资源使用的互斥机制 3. 理解生产者与消费者问题中同步机制 4. 掌握Java中线程同步的实现方法 | | | | |
| 实验环境 | Java Eclipse环境 | | | | |
| 实验要求 | 1. 运行“模拟订票系统程序”，观察运行结果是否正确，试着用操作系统原理来解释。 2. 修改“模拟订票系统程序”，在需要同步的代码前加上同步关键词，使得其正确运行。   通过synchronized() notify() wait()三个方法完成线程同步。   1. 将两个线程通信的（即生产者和消费者）程序补充完整，使其运行，观察运行结果，试着用操作系统原理来解释。 2. 修改“消费者和生产者”程序，使两个线程同步，观察运行结果，用操作系统原理来解释。 | | | | |
| 算法描述及实验结果记录 | 1. 答：不正确，各个线程在并发执行的时候，没有互斥执行，票是临界资源，如果不加以限制就会出现同一张票重复售出，或者系统已经没有票了，但是因为没有互斥执行，导致最后超售票。 2. 通过Java中synchronized，wait()，notify()完成      1. 当写入数据没有完成时，整个临界资源Stroge是被Input对象线程所占用的；当写入线程结束，通过notify将读取线程激活，从而读出数据，修改代码展示如下：        1. 生产者和消费者两个线程同步的话，只要Storge中存在数据，那么消费者就会一直读数据，同时生产者也会一直写数据进去。这时候Storge就不再是临界资源了，更像是管道通信。修改代码如下： | | | | |
| 总结 | 通过Java编程更加深刻学习了线程相关的知识内容。也在实验过程中遇到了各种各样的问题。  好比：在调用notify()和wait()方法必须在synchronized块里面调用；每次调用wait()方法等待的线程被唤醒之后，是接着wait()方法后面的代码继续执行，并非重新进入同步代码块；而且调用的这两个方法的对象一定是当前synchronized同步的锁对象，不然程序会报错。  相对于之前C语言中的进程操作而言，java的线程相对简单些，但是也有很多逻辑上需要注意的地方。 | | | | |
| 附录 |  | | | | |