山东大学 计算机科学与技术 学院

操作系统 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201705130113 | 姓名：黄瑞哲 | | 班级：计科17.3 |
| 实验题目：死锁问题实验 | | | |
| 实验学时：4 | | 实验日期：2019.12.2 | |
| 实验目的：  通过本实验观察死锁产生的现象，考虑解决死锁问题的方法。从而进一步加深对 于死锁问题的理解。掌握解决死锁问题的几种算法的编程和调试技术。练习怎样构 造管程和条件变量，利用管程机制来避免死锁和饥俄问题的发生。 | | | |
| 硬件环境：  Intel Core i5-8300H | | | |
| 软件环境：  Windows 10 Pro 1903  Ubuntu 18.04  VS Code | | | |
| 实验内容与设计：  在两个城市南北方向之间存在一条铁路，多列火车可以分别从两个城市的车站 排队等待进入车道向对方城市行驶，该铁路在同一时间，只能允许在同一方向上行 车，如果同时有相向的火车行驶将会撞车。请模拟实现两个方向行车，而不会出现 撞车或长时间等待的情况。  对于这个实验，我设计的方案是两个方向的列车一次通行，当一个方向的列车通过后会优先唤醒另外一个方向的列车，如果另一个方向没有等待的列车，则唤醒这个方向上的列车，使其通行。由此可以解决饥饿的问题。  这个实验要求使用管程和条件变量实现。下图是管程的声明。    主进程会fork出10个子进程来模拟列车，其中列车的到达分两批，每批次5辆。    为了模拟随机性的场景，我对每辆列车的行驶方向做了随机化的处理。每次取一个随机数，然后这个随机数在模2意义下便是它的行驶方向：0是往左，1是往右。但是实验过程中发现如果方向在每个子进程中执行rand()会导致结果是一样的，我只好事先在主进程中准备一个数组，利用这个数组来获取每个子进程的方向。    最后我的实现介于一些奇怪的原因，只能实现同一时间只能有同一方向的一辆列车通行，具体的问题我会在下一块具体描述。  下图是运行结果。一共有10辆车，5辆车首先到达，然后又有5辆车到达。同时铁轨上只能跑一辆车。 | | | |
| 结论分析与体会：  在写独立实验的时候，遇到一个非常麻烦的问题。现在有两个信号量，和一个互斥锁。两个信号量0、1分别控制往左走的列车和往右走的列车。现在有两个进程（列车）都是往右走的，其中一个列车往右通过铁路以后，会优先唤醒往左走的列车，但是另一个往右走在等待的列车却被唤醒了。这个问题到现在还没有解决，原因猜测是互斥锁的问题，但是不知道相关的内部实现，因此也没有一个确切的结论。  在研究这个问题的过程中，我查看了linux的源码，发现在信号量这一部分，linux内核和gcc是两种描述，分别在/usr/include/linux/sem.h和/usr/include/x86\_86-linux-gnu/sys/sem.h。其中有个结构体类型semid\_ds在这两个里面就是不一样的。在linux内核中它有一个类型为sem\_queue的成员，记录了在这个信号量上等待的进程，这个信息在gcc中却没有记录（具体结构见下图）。我本来想查看一下等待的进程，可无奈不知道怎么修改。  不过还是要感谢助教帮我调了好久的bug！  最后完成这个实验，只能更换了写法，仅仅使用两个信号量完成，这种写法只能保证在同一时刻只能有一辆列车经过铁路驶向对面。 | | | |