

操作系统课程设计 实验报告

小组成员

刘天祺 1320151097

余宇峰 1120151800

牛昕仑 1320151093

刘佳逸 1320151101

崔胜凯 1320151106

学院：计算机学院

专业：物联网工程

日期：2018 年 5 月 10 日

实验六 分组实验

一、实验要求

本实验室是很多仿真系统中的核心部分，在仿真中由很多的模型（Model）组成，每种模型代表一类参与仿真的对象。比如在一场战斗的仿真中，可能有坦克、导弹、雷达等各种对象参与。仿真的过程就是按照时间顺序在各种不同的模型之间进行交互和信息的交换。最核心的要求就是时间顺序不能乱，数据传输正确。

编写一个程序，程序中可以使用多线程或者多进程来实现。每个线程/进程运行一个弹道计算的模型。

程序启动时（初始化阶段），创建所有的进程/线程，然后每个模型随机（或者人为指定）的向模型总数 30% 左右的其他模型建立有向连接关系（例如一共有 100 个模型，那每个模型要把自己的状态发送给至少 30 个其他的模型），此关系建立后不再变化（注意：不要形成图 1 所示的非连通图）。要求每个模型每更新一次自己状态都必须将状态（Result）按照连接的方向发送给下游模型，如图 2 中 M1 更新状态后需要将状态发送给 M2、M3 和 M4，M2、M3 和 M4 接收到 M1 的状态后判断一下时间是否大于等于自己当前时间，如果小于自己当前时间整个程序报错退出。

二、实验环境

操作系统：Ubuntu 18.04 LTS 64bit

编译器：g++ (Ubuntu 7.3.0-16ubuntu3) 7.3.0

三、实验过程

2.1 实验思路

设 A 的下游节点为 B_i ($0 \leq i \leq 30$)，上游节点为 C_j ，当节点 A 请求 `movetoNext()` 之前，进行检查：

对任意一个 B_i 有：

$$A.nexttime \geq B_i.currenttime \quad \dots\dots (1)$$

对任意一个 C_j 有：

$$A.nexttime \leq C_j.nexttime \quad \dots\dots (2)$$

只有满足上述条件时，节点 A 才被允许执行 `movetoNext()` 操作，若不能满足条件(1)则会出现上游节点时间小于下游节点时间的错误，如果不能满足条件(2)则不能保证所有节点都能够安全地推进。

2.2 实验步骤

1. 创建共享主存区，缓冲区中存储每个模型的当前时间、当前 range、下次时间和下一次可抵达的 range

2. 创建 100 个子进程，每个子进程中创建一个模型。

对于每一个模型：

(1) 初始化：初始化对应的共享主存区、随机选择下游节点并建立下游节点表

(2) 推进检测：检测模型的下一次可抵达的 range 是否为所有模型中下一次可抵达的 range 的最小值，若是则转 (3)，否则重复步骤 (2)

(3) 申请 mutex 信号量

- (4) 检错: 检查下游节点当前时间是否小于模型的下次时间, 如是则报错, 退出程序
- (5) 推进: 推进节点, 更新对应共享主存区信息, 输出节点信息
- (6) 释放 mutex 信号量

四、实验结果

```
Model-18 moving... step=25, time: 0.028944, range:25.165137
Model-54 moving... step=05, time: 0.028944, range:25.165137
Model-23 moving... step=25, time: 0.028944, range:25.165137
Model-12 moving... step=25, time: 0.028944, range:25.165137
Model-77 moving... step=25, time: 0.028944, range:25.165137
Model-06 moving... step=05, time: 0.028944, range:25.165137
Model-50 moving... step=25, time: 0.028944, range:25.165137
Model-92 moving... step=05, time: 0.028944, range:25.165137
Model-93 moving... step=05, time: 0.028944, range:25.165137
Model-02 moving... step=01, time: 0.028944, range:25.165137
Model-60 moving... step=25, time: 0.028944, range:25.165137
Model-02 moving... step=01, time: 0.030097, range:26.165076
Model-02 moving... step=01, time: 0.031252, range:27.165014
Model-02 moving... step=01, time: 0.032407, range:28.164953
Model-02 moving... step=01, time: 0.033563, range:29.164892
Model-46 moving... step=10, time: 0.034720, range:30.164831
Model-17 moving... step=10, time: 0.034720, range:30.164831
Model-44 moving... step=10, time: 0.034720, range:30.164831
Model-67 moving... step=10, time: 0.034720, range:30.164831
Model-48 moving... step=05, time: 0.034720, range:30.164831
```

五、实验总结

本次实验应用了前几次实验学到的知识, 比如使用 `fork()` 创建子进程、操作共享主存区、信号量.....实验的关键在于找到一种合适的推进方法, 确保各个模型之间的通讯不会出现时间错误。本次实验中通过预测下一次节点可达到的 **range**, 并选择拥有最小值的节点优先推进, 如此确保了安全, 但是却增加了其他进程的等待时间, 目前没有更好的解决方法。

对于实验中要求的所有节点要形成连通图这一点, 实验中采取固定地向下个节点编号的节点建立连接, 然后随机地向另外 29 个节点建立连接的方法, 确保图的连通。

六、组员分工

- 余宇峰: 算法思路讨论、报告撰写
- 刘天祺: 算法思路讨论、代码实现并调试
- 牛昕仑: 讨论
- 刘佳逸: 讨论
- 崔胜凯: 讨论