实验 6 分组实验

自由分组，每组不超过 5 名同学。

实验内容：

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。

在本实验中，给大家分发的代码是一个简单的弹道计算模型。各位同学不用关心模型具体的实现算法。

模型的创建方法有如下两种：

1. Model * m = new Model(); //创建默认模型，没记错的话默认的 step 是 10
2. Model * m = new Model(int step); //step 的取值可以为：1，5，10，25 这几种值

创建的模型的初始时间都是 0。

模型的使用方法：

Result * rst=m->moveToNext();

每次调用 moveToNext 函数可以将炮弹的运行时间向前推进并且更新炮弹的状态信息，炮弹的时间和状态等通过 Result 返回。每调用一次就会把模型的当前时间和当前状态进行更新，其中模型的时间不会减小（可能两次相同或者增加），可

以通过 getTime()获得模型当前的时间。每次通过 moveToNext 获得 Result 对象用过之后需要释放内存。在模型中还有一个 getNextTime 的函数，用于预取一下下一次的时间，功能同 moveToNext 的唯一区别就是该函数不会更新模型的当前时间和状态，只是查看一下如果更新状态将获得什么样的时间。

以下是一个简单的模型测试程序，告诉大家可以怎样使用一个模型:

```
#include "ballistics.h"
#include "GBCSolution.h"
#include "Result.h"
#include "stdio.h"
#include "Model.h"
int main()
{

    Model * m = new Model();
    //Model * m = new Model(25);
    Result * rst = NULL;
    while(rst=m->getNextStatus())
    {
        printf("Current Time is %f \t Range is: %f", rst->getTime(), rst->getRange());

        free(rst);
```

}

}

实验要求：

1. 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。
2. 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~~都把各自的~~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~~总数量 50%左右的其他模型（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 50 个其他的模型，具体的模型可以自由选择）。同时保证时间的正确性。时间正确是指，一个模型不能从其他模型收到一个比自己当前时间要早的消息。~~

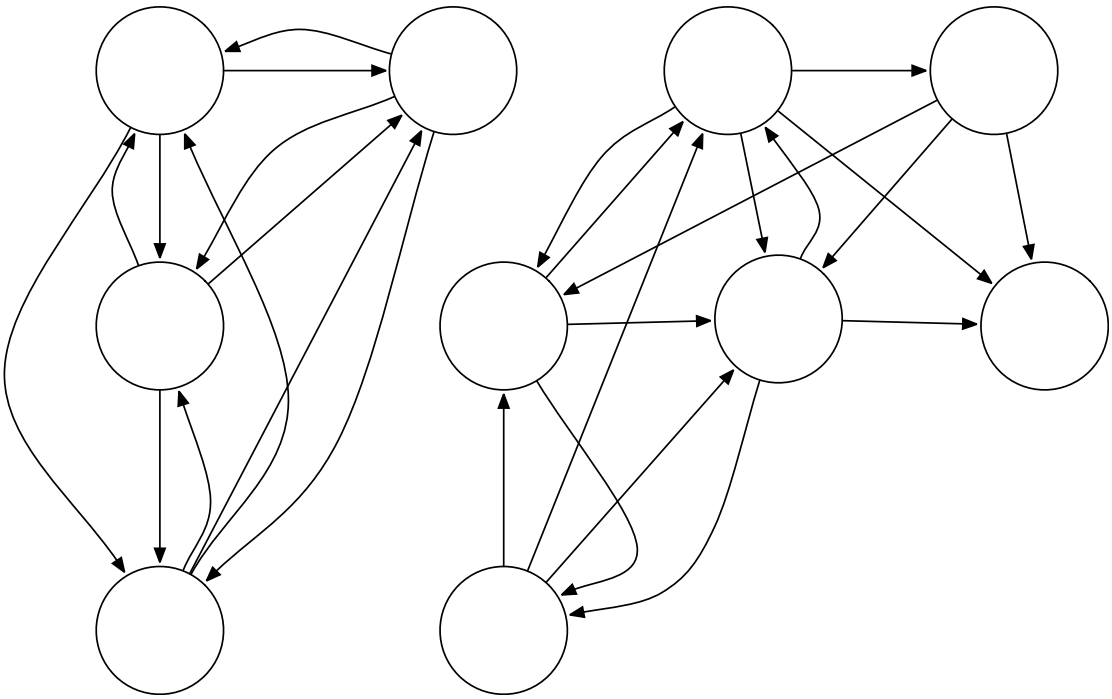


图 1　不正确的连接关系

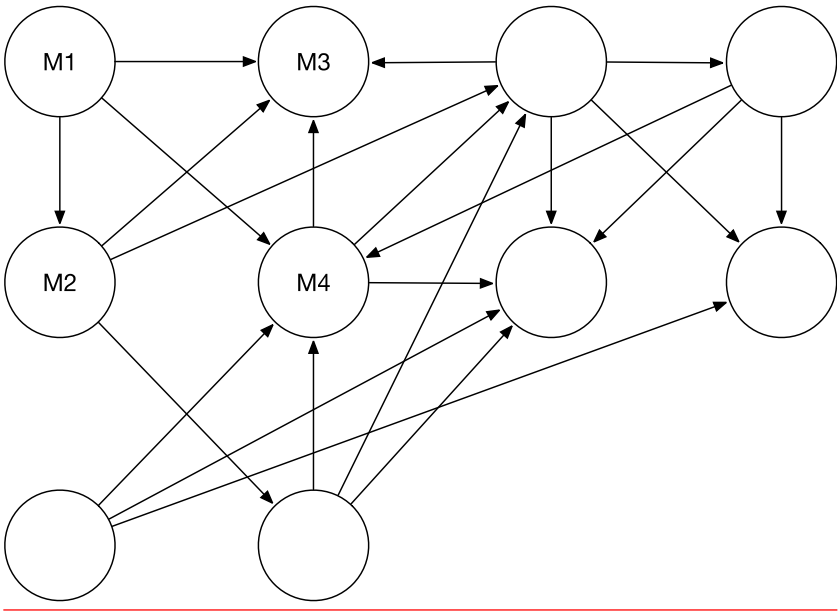
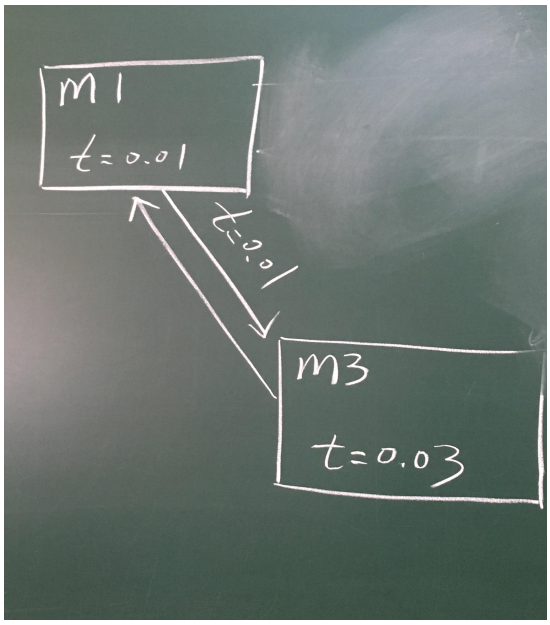


图 2 正确的连接关系

3. 除了保证每次更新后将状态发送给下游模型外，还需要保证时间的正确性。时间正确是指，一个模型不能从其他模型收到一个比自己当前时间要早的消息。如下图所示，模型 1（m1）更新自己的状态后时间为 0.01，它如果将自

己的状态发送给 m3，但是 m3 接到消息后发现自己的时间已经是 0.03 了，这就就会造成了时间错误~~，因为 m3 的当前时间已经是 0.03 了~~。因此此状态不可能出现。对于下图所示的两个模型来说，在 m1 的时间更新到 0.03 之前，m3 是不能更新到 0.03 的。



4. 创建的 Model 中需要至少包含一个 step 值为 1 和一个 step 值为 25 的，不能所有模型的 step 都一样。（step 值越大计算越快）

测试环境：

Ubuntu

实验评判标准：

1. 支持尽量多的模型，越多越好
2. 计算性能佳，主要就是整体的计算时间，越快越好。
3. 性能测试基准初步选定 100 个模型来测，后面根据实际情况可以调整。
4. 每组提交一个实验报告。验收时每组需要通过 ppt 讲解一下实验情况，包括算

法思路等，还要写明组员分工情况。

补充~~提示~~：

目前我也不知道最佳的解决方案是什么，其中一种可能的解决方法就是找到全局安全的一个时间点，然后所有模型在该点同步。然后寻找下一个安全的全局时间点。这样的方法估计性能比较差。各位同学可以思考~~那种~~设计一种算法能够有比较高的效率。