

**2019年春季学期  
计算机学院大二软件构造课程**

**Lab 6实验报告**

|  |  |
| --- | --- |
| 姓名 | 张钟升 |
| 学号 | 1170300431 |
| 班号 | 1703004 |
| 电子邮件 | [1170300431@stu.hit.edu.cn](mailto:1170300431@stu.hit.edu.cn) |
| 手机号码 | 13029801568 |

**目录**

[1 实验目标概述 1](#_Toc11532473)

[2 实验环境配置 1](#_Toc11532474)

[3 实验过程 1](#_Toc11532475)

[3.1 ADT设计方案 1](#_Toc11532476)

[3.2 Monkey线程的run()的执行流程图 2](#_Toc11532477)

[3.3 至少两种“梯子选择”策略的设计与实现方案 2](#_Toc11532478)

[3.3.1 策略1 2](#_Toc11532479)

[3.3.2 策略2 2](#_Toc11532480)

[3.3.3 策略3（可选） 2](#_Toc11532481)

[3.4 “猴子生成器”MonkeyGenerator 2](#_Toc11532482)

[3.5 如何确保threadsafe？ 3](#_Toc11532483)

[3.6 系统吞吐率和公平性的度量方案 3](#_Toc11532484)

[3.7 输出方案设计 3](#_Toc11532485)

[3.8 猴子过河模拟器v1 5](#_Toc11532486)

[3.8.1 参数如何初始化 5](#_Toc11532487)

[3.8.2 使用Strategy模式为每只猴子随机选择决策策略 5](#_Toc11532488)

[3.9 猴子过河模拟器v2 5](#_Toc11532489)

[3.9.1 对比分析：固定其他参数，选择不同的决策策略 5](#_Toc11532490)

[3.9.2 对比分析：变化某个参数，固定其他参数 6](#_Toc11532491)

[3.9.3 分析：吞吐率是否与各参数/决策策略有相关性？ 8](#_Toc11532492)

[3.9.4 压力测试结果与分析 8](#_Toc11532493)

[3.10 猴子过河模拟器v3 9](#_Toc11532494)

[4 实验进度记录 9](#_Toc11532495)

[5 实验过程中遇到的困难与解决途径 10](#_Toc11532496)

[6 实验过程中收获的经验、教训、感想 10](#_Toc11532497)

# 实验目标概述

本次实验训练学生的并行编程的基本能力，特别是 Java 多线程编程的能力。

根据一个具体需求，开发两个版本的模拟器，仔细选择保证线程安全（threadsafe）的构造策略并在代码中加以实现，通过实际数据模拟，测试程序是否是线程安全的。另外，训练学生如何在 threadsafe 和性能之间寻求较优的折中，为此计算吞吐率和公平性等性能指标，并做仿真实验。

* Java 多线程编程
* 面向线程安全的 ADT 设计策略选择、文档化
* 模拟仿真实验与对比分析

# 实验环境配置

repo: <https://github.com/ComputerScienceHIT/Lab6-1170300431>

# 实验过程

请仔细对照实验手册，针对三个问题中的每一项任务，在下面各节中记录你的实验过程、阐述你的设计思路和问题求解思路，可辅之以示意图或关键源代码加以说明（但千万不要把你的源代码全部粘贴过来！）。

## ADT设计方案

设计了哪些ADT、各自的作用、属性、方法；

给出每个ADT的specification；

* Monkey

表示一个猴子. 属性包括ID, 最大速度, 方向, 策略等. ; AF(ID)表示一个猴子, AF(MV)表示猴子的最大速度, AF(dir)表示猴子的方向, AF(lNum, rNum)表示猴子在的位置.

* Ladder

Ladder表示一个梯子. 属性包括ID, 方向, 猴子数量等. 方法包括移动梯子上的猴子, 返回梯子上猴子数量, 返回梯子头部是否有猴子, 向梯子上增加一个猴子, 取得离梯子头最近的一个猴子的速度等. RI: . AF(dir)表示梯子上猴子的方向, AF(ID)表示一个梯子, AF(monkeyNum)表示桥上猴子数; AF(finish)表示下桥的猴子数, AF(monkeys)表示梯子上的空间;

## Monkey线程的run()的执行流程图

这里无需考虑具体采用的梯子选择策略。

执行策略

上桥?

开始

结束

否

是

## 至少两种“梯子选择”策略的设计与实现方案

### 策略1

优先选择没有猴子的梯子，若所有梯子上都有猴子，则在岸边等待，直到某个梯子空闲出来；

### 策略2

优先选择没有猴子的梯子，若所有梯子上都有猴子，则优先选择没有与我对向而行的猴子的梯子；若满足该条件的梯子有很多，则随机选择.

### 策略3（可选）

优先选择没有猴子的梯子，若所有梯子上都有猴子，则优先选择没有与我对向而行的猴子的梯子；若满足该条件的梯子有很多，选择末尾速度差距与我小于一个阈值的梯子. 如果还是有很多这样的梯子, 选择速度差最小的. 如果没有这样的梯子, 则等待.

## “猴子生成器”MonkeyGenerator

实例化MonkeyGenerator时用N, k, t, MV初始化; MonkeyGenerator持有一个Monkey集合. 当调用Monkey.create()时, 创建一个线程每隔t秒调用birth(int). birth创建给定参数个猴子, 加入到Monkey集合. 创建Monkey时就会启动Monkey线程. create线程创建后, 启用线程调用Ladder.move. 阻塞主线程(或者在主线程里调用也可以), 直到所有猴子都下桥. 最后MonkeyGenerator计算Mbps和fairness.

## 如何确保threadsafe？

在各策略中, 根据策略锁MonkeyGenerator的Monkey池; 锁Ladder.ladders以防止在获取数据时这些数据被其他线程更改; 在写log, 更新GUI的时候锁MonkeyGenerator, 防止这些操作的过程中Monkey的属性发生变化.

## 系统吞吐率和公平性的度量方案

* “吞吐率”是指：假如只猴子过河的总耗时为秒，那么每只猴子的 平均耗时为秒，则吞吐率表征每秒钟可过河的猴子数目;
* “公平性”是指：如果对象比出生得早，那么 应该不晚于抵达对岸，则为“公平”; 若比晚到对岸，则为“不公平”。设和的产生时间分别为和，抵达对岸的时间分别为和, 那么公平性 。对只猴子两两计算其之间的公平性并综合到一起，得到本次模拟的整体公平性, 其取值范围为

## 输出方案设计

* 日志

1. 2019-06-14T15:15:14.208207300Z  INFO
2. Monkey(0) on Ladder(0), Rung(17). R->L, 1s from birth.
3. 2019-06-14T15:15:14.249978800Z  INFO
4. Monkey(1) waiting at right shell, 1s from birth.
5. 2019-06-14T15:15:14.250977900Z  INFO
6. Monkey(2) waiting at right shell, 1s from birth.
7. 2019-06-14T15:15:14.250977900Z  INFO
8. Monkey(3) waiting at right shell, 1s from birth.
9. 2019-06-14T15:15:14.250977900Z  INFO
10. Monkey(4) waiting at right shell, 1s from birth.
11. 2019-06-14T15:15:14.250977900Z  INFO
12. Monkey(5) waiting at left shell, 1s from birth.
13. 2019-06-14T15:15:14.250977900Z  INFO
14. Monkey(6) waiting at right shell, 1s from birth.
15. 2019-06-14T15:15:14.250977900Z  INFO
16. Monkey(7) waiting at right shell, 1s from birth.
17. 2019-06-14T15:15:14.250977900Z  INFO
18. Monkey(8) waiting at left shell, 1s from birth.
19. 2019-06-14T15:15:14.254101700Z  INFO
20. Monkey(9) on Ladder(1), Rung(15). R->L, 1s from birth.
21. 2019-06-14T15:15:14.254631100Z  INFO
22. Monkey(10) on Ladder(2), Rung(5). L->R, 1s from birth.
23. 2019-06-14T15:15:14.255631300Z  INFO
24. Monkey(11) waiting at right shell, 1s from birth.
25. 2019-06-14T15:15:14.255631300Z  INFO
26. Monkey(12) on Ladder(3), Rung(1). L->R, 1s from birth.
27. 2019-06-14T15:15:14.255631300Z  INFO
28. Monkey(13) waiting at right shell, 1s from birth.
29. 2019-06-14T15:15:14.255631300Z  INFO
30. Monkey(14) on Ladder(4), Rung(2). L->R, 1s from birth.
31. 2019-06-14T15:15:14.255631300Z  INFO
32. Monkey(15) waiting at right shell, 1s from birth.

* GUI

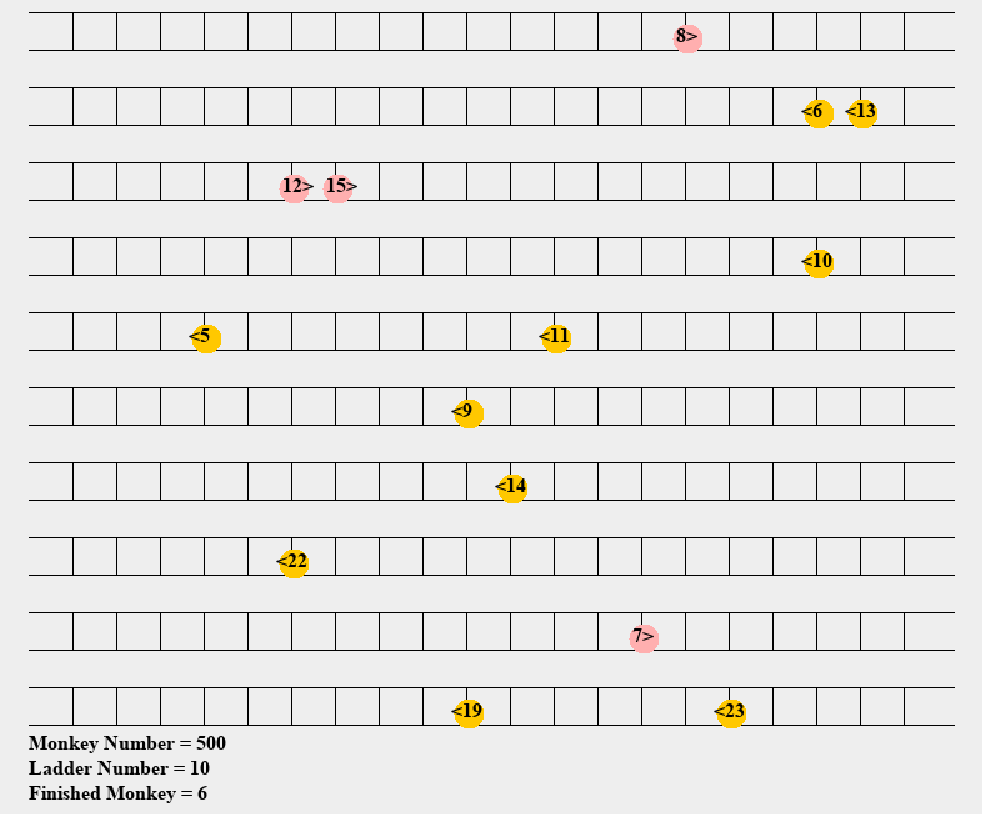


图 1 GUI示例

## 猴子过河模拟器v1

### 参数如何初始化

MonkeyGenerator(**int** n, **int** h, **int** t, **int** N, **int** k, **int** MV);

其中n和h传给Ladder,

**static** **void** init(**int** n, **int** h);

### 使用Strategy模式为每只猴子随机选择决策策略

猴子持有int型变量表示策略, 用Strategy.choose(Monkey, int)调用各策略; 用户可以通过指定该变量以更改策略, 也可以传入随机数以随即决定策略.

## 猴子过河模拟器v2

在不同参数设置和不同“梯子选择”模式下的“吞吐率”和“公平性”实验结果及其对比分析。

### 对比分析：固定其他参数，选择不同的决策策略

图 2

### 对比分析：变化某个参数，固定其他参数

图 3

图 4

图 5

图 6

图 7

### 分析：吞吐率是否与各参数/决策策略有相关性？

有相关性. 比如n比较小的时候, 吞吐率一定小, n更大吞吐率也会增加. 在一定范围内, N增大也可以增加吞吐率, 但吞吐率的上限显然受策略影响. 当k比较小的时候, 吞吐率会比较小.

### 压力测试结果与分析

图 8

图 9

当猴子比较少, 而且追求公平性的时候, 策略1适用; 在猴子比较多的时候, 策略2和策略3的吞吐率比较接近.

## 猴子过河模拟器v3

图 10 测试平均值

# 实验进度记录

请尽可能详细的记录你的进度情况。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 计划任务 | 实际完成情况 |
| 6-13 |  | v1 | 基本完成 |
| 6-14 |  | v1 | 完成 |
| 6-14 |  | v2 | 完成 |
| 6-14 |  | v3 | Competition1 |
| 6-15 |  | v3 | 完成 |

# 实验过程中遇到的困难与解决途径

猴子的移动很难控制-由Ladder控制猴子在桥上的移动;

GDI绘图闪烁-双缓冲绘图

# 实验过程中收获的经验、教训、感想

本节除了总结你在实验过程中收获的经验和教训，也可就以下方面谈谈你的感受（非必须）：

1. 多线程程序比单线程程序复杂在哪里？你是否能体验到多线程程序在性能方面的改善？
2. 你采用了什么设计决策来保证threadsafe？如何做到在threadsafe和性能之间很好的折中？
3. 你在完成本实验过程中是否遇到过线程不安全的情况？你是如何改进的？
4. 关于本实验的工作量、难度、deadline。
5. 到此为止你对《软件构造》课程的意见和建议。