**操作系统设计报告**

**——1753603 林寅嘉**

目录

[1设计方案 2](#_Toc7165485)

[1.1生产者设计方案 2](#_Toc7165486)

[1.2消费者设计方案 2](#_Toc7165487)

[1.3缓存区设计方案 3](#_Toc7165488)

[1.4程序界面设计方案 3](#_Toc7165489)

[2所实现的功能 4](#_Toc7165490)

[3项目截图演示 5](#_Toc7165491)

[4实现语言，工具等 6](#_Toc7165492)

## 1设计方案

## 1.1生产者设计方案

通过新建线程来实现增加生产者的功能。每当生产者拿到缓存区的使用权，立即将缓存区加锁锁定，防止其他生产者或者消费者对其进行访问修改，造成数据不一致的情况。之后随机生成一个数值（store\_number）用来指定当前线程想要访问的缓存区。然后判断当前缓存区是否已经超过指定的上限，如果当前缓存区已经满了，那么该线程需立即放弃对缓存区的锁并进入休眠（.wat()）；如果当前缓存区还没有到达上限，那么随机产生一个整数代表要生产的商品数量，更改相应缓存区内的商品数量。为了模拟实际情况，让用户看清楚生产消费信息。生产完成后要让此线程持有缓存区的锁休眠1s（Thread.sleep(1000)）,之后模拟生产完成后，通知其他线程消费（.notifyAll()）。为了不让当前线程持续持有缓存区的锁，在当前线程退出同步代码区块时，让其休眠1s。

## 1.2消费者设计方案

通过新建线程来实现增加消费者的功能。每当消费者拿到缓存区的使用权，立即将缓存区加锁锁定，防止其他生产者或者消费者对其进行访问修改，造成数据不一致的情况。之后随机生成一个数值（store\_number）用来指定当前线程想要访问的缓存区。然后判断当前缓存区中是否有商品，如果当前缓存区为空，那么消费该线程需立即放弃对缓存区的锁并进入休眠（.wat()）；如果当前缓存区有商品可以消费，那么随机产生一个整数代表要消费的商品数量，更改相应缓存区内的商品数量。为了模拟实际情况，让用户看清楚生产消费信息。生产完成后要让此线程持有缓存区的锁休眠1s（Thread.sleep(1000)）,之后模拟消费完成后，通知其他线程生产（.notifyAll()）。为了不让当前线程持续持有缓存区的锁，在当前线程退出同步代码区块时，让其休眠1s。

## 1.3缓存区设计方案

使用Swing中的JProgressBar组件来模拟缓存区的每个单元格，通过声明JProgressBar数组来模拟缓存区，通过其自身的get，set方法来获取，设置缓存区的数组，从而动态时时修改缓存区的商品数量。

## 1.4程序界面设计方案

此程序界面由JLabel，JPrograssBar，JButton，JTextArea等组件构成，程序的界面如下图图所示：



## 2所实现的功能

1. 开始：通过点击开始按钮来启动该程序。
2. 重置：通过点击重置按钮来重置当前生产者，消费者，仓库数量，仓库内的商品数量以及生产消费信息。
3. 暂停：当开始后可通过点击暂停按钮来暂停当前程序。
4. 继续：通过点击继续按钮以继续当前程序。
5. 退出：通过点击退出按钮来退出当前程序。
6. 修改生产者数量
7. 修改消费者数量
8. 修改仓库数量
9. 时时显示生产消费信息以及各仓库内的商品数量

## 3项目截图演示





## 4实现语言，工具等

实现语言：Java

实现工具：Eclipse，exe4j，WindowsBuilder