

****

《实用操作系统》实验报告

（五）

**姓　　名 ：宋泽涛**

**学　　号 ：25120222201292**

**学 院 ：信息学院**

**专 业 ：软件工程**

**2024年 11 月**

# 实验目的

* 了解死锁与饥饿产生的条件
* 了解死锁的解决方法
* 掌握利用银行家算法进行死锁避免

# 实验环境

* + PC + Linux Red Hat操作系统
  + GCC

# 实验内容

* 给出进程需求矩阵C、资源向量R以及一个进程的申请序列。使用进程启动拒绝和资源分配拒绝（银行家算法）模拟该进程组的执行情况。要求：

初始状态没有进程启动

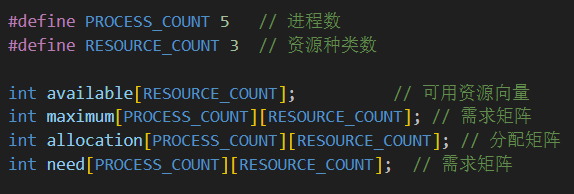
计算每次进程申请是否分配？如：计算出预分配后的状态情况（安全状态、不安全状态），如果是安全状态，输出安全序列。

每次进程申请被允许后，输出资源分配矩阵A和可用资源向量V。

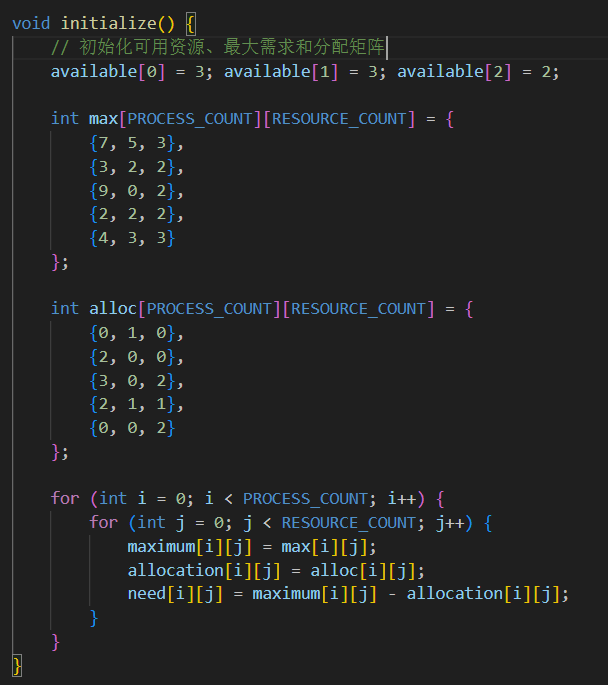
每次申请情况应可单步查看，如：输入一个空格，继续下个申请

# 实验具体实现

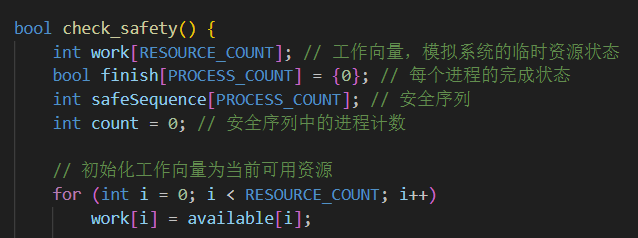
1.实验要求给出进程需求矩阵C、资源向量R以及一个进程的申请序列。使用进程启动拒绝和资源分配拒绝（银行家算法）模拟该进程组的执行情况。



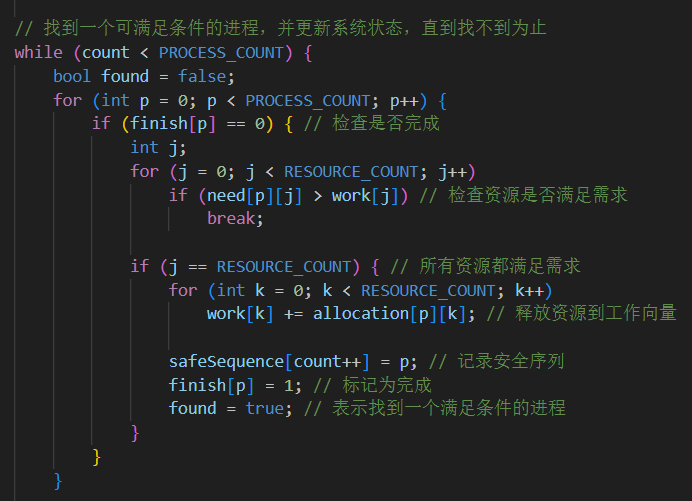
2.首先对可用资源、最大需求和分配矩阵进行初始化操作，其中need数组用于计算剩余需求。



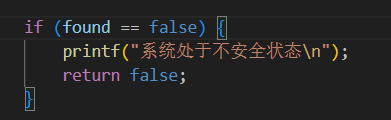
3.在安全性检查函数中模拟系统的临界资源状态，记录每个进程的完成状态，并记录最后的安全序列。



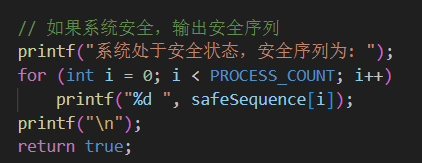
4.尝试寻找一个可满足条件的进程，并更新系统状态，直到找不到为止。



5.未找到满足条件的进程，说明系统不安全。

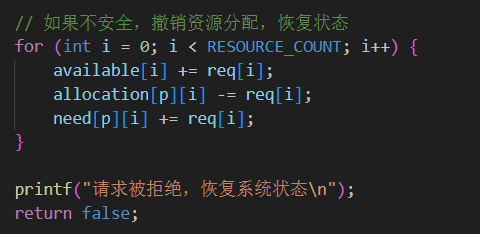


6.如果系统安全，那么就输出之前记录的安全序列。



7.资源请求处理函数，根据请求判断是否分配资源

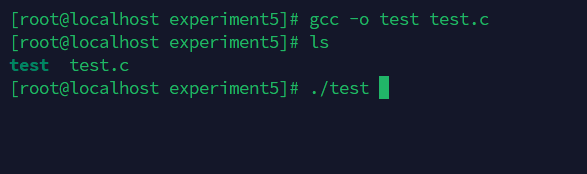




8.模拟多种请求



9.linux中运行结果如下：





# 实验分析与总结

1.通过本实验，深入理解了银行家算法的工作原理及其在死锁避免中的应用。银行家算法基于安全性检查，以保证系统在满足进程请求的前提下不会进入不安全状态，是一种有效的死锁避免方法。

2.资源分配在满足进程需求的同时，也必须保证系统状态安全，才能避免死锁。安全性检查不仅在资源分配时进行，也可以在系统管理资源的其他过程中使用，以防范死锁的发生。

3.通过实验深入理解了死锁和饥饿的概念及其解决方案。银行家算法通过系统性分析资源分配安全性，有效避免死锁问题；而饥饿问题的解决则需要在设计系统时考虑公平性策略，避免进程长期等待资源。