山东大学 软件 学院

**操作系统** 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202200300092 | 姓名：郑睿杰 | | 班级：大数据2班 |
| 实验编号：实验5 | | | |
| 实验题目：进程互斥实验 | | | |
| 实验学时：4 | | 实验日期：2024.5.27 | |
| 实验目的：  进一步研究和实践操作系统中关于并发进程同步与互斥操作的一些经典问题的解法，加深对于非对称性互斥问题有关概念的理解。观察和体验非对称性互斥问题的并发控制方法。进一步了解 Linux 系统中 IPC 进程同步工具的用法，训练解决对该类问题的实际编程、调试和分析问题的能力。 | | | |
| 硬件环境：  机型：ThinkBook-13x-ITG  CPU：11th Gen Intel(R) Core(TM) i7-1160G7 @ 1.20GHz  内存：16150988 kB | | | |
| 软件环境：  Ubuntu 9.4.0-1 ubuntu1~20.04.2 + vscode2022 | | | |
| 实验步骤与内容：  【实验模型描述】  这个理发师实验模型是一个基于进程间通信（IPC）的程序，模拟了一个理发店的运营过程。在这个模型中，理发店的场景被抽象为沙发和等候室两个区域，顾客到来时会先坐在沙发上等待理发，如果沙发满了则会进入等候室等待。  这个模型中包含了以下关键元素：  1. 沙发（Sofa）：  - 沙发是理发店的主要座位区域，每个沙发可以容纳最多 4 个顾客。  - 当有新顾客到来时，会首先检查沙发是否还有空位，如果有空位，则新顾客可以直接坐在沙发上等待理发。  - 如果沙发已满，则新顾客需要进入等候室等待。  2. 等候室（Waiting Room）：  - 等候室是沙发满了时的备用座位区域，可以容纳最多 13 个顾客。  - 当沙发已满时，新顾客会被安排到等候室中等待理发。  - 如果等候室也满了，则新顾客无法进入理发店，需要等待其他顾客离开等候室后才能进入。  3. 消息队列（Message Queues）：  - 使用消息队列进行沙发和等候室之间的通信，以及顾客和理发师之间的通信。  - 沙发和等候室分别有自己的消息队列，用于存储等待理发的顾客信息。  - 当有新顾客到来时，会将其信息发送到相应的消息队列中。  4. 信号量（Semaphores）：  - 使用信号量进行进程同步和互斥操作，保证多个进程之间的正确执行顺序和资源的安全访问。  - 在这个模型中，使用了一个信号量来控制等候室的容量，当等候室已满时，阻塞新顾客的到来，直到有其他顾客离开等候室。  【主要算法代码分析】  Makefile:    Ipc.h:    ipc.c:    barber.c:    customer.c:    【运行结果】    【实验结果的分析综合】  理发师模型的IPC进程实验结果呈现了一个充满活力且高效运作的理发店场景。通过该模型，我们可以观察到以下关键的分析和综合结果：   1. 进程间通信的有效性：   - IPC机制在理发师模型中被成功地应用，通过消息队列实现了沙发和等候室之间的顺畅通信。消息队列作为一个缓冲区，有效地管理了不同进程之间的信息交换，保证了信息的安全传递和及时响应。  2. 资源利用效率：  - 模型中合理利用了沙发和等候室的座位资源，通过动态分配座位，保证了顾客的有序等待和理发。沙发和等候室的座位容量限制，有效地控制了顾客数量，防止了理发店过度拥挤和资源浪费。  3. 顾客体验与服务质量：  - 模型中的顾客可以根据座位的情况自主选择等待的位置，这种灵活性提高了顾客的满意度和体验。同时，通过及时地响应顾客到来和有效地管理座位资源，保证了顾客能够尽快得到理发服务，提高了服务的效率和质量。  4. 程序的实时性和响应能力：  - 采用非阻塞的消息接收方式，使得程序能够及时地响应顾客到来的情况，保持了程序的实时性。这种及时响应的能力，有助于应对顾客数量的突然增加或减少，保证了理发店的运营流畅性和稳定性。  5. 多进程协作的有效性：  - 理发师模型中多个进程之间的协作十分有效，通过消息队列和信号量的配合，实现了顾客的有序等待和座位资源的动态管理。不同进程之间的良好协作，保证了理发店的正常运营和服务质量。  综合来看，理发师模型的IPC进程实验结果展示了一个高效、有序、灵活的理发店运营场景。通过该模型的分析和综合，我们可以更深入地理解进程间通信的重要性和有效性，以及如何通过合理的资源管理和协作机制提升服务质量和顾客满意度。 | | | |
| 结论分析与体会：  【实验中遇到的问题和解决方法】   1. 多个进程同时竞争访问共享资源，可能导致数据不一致或不可预测的行为。解决方法：使用信号量等同步机制来保证对共享资源的互斥访问，确保每次只有一个进程可以访问共享资源，避免竞态条件的发生。   消息队列满溢：   1. 消息队列可能会因为消息积压而导致满溢，无法再接收新的消息，从而影响系统的正常运行。解决方法：定期监测消息队列的长度，及时处理积压的消息，例如增加消息队列的大小或者定期清理消息队列中的旧消息。 2. 多个进程之间因为互相等待对方释放资源而无法继续执行，形成死锁。解决方法：避免循环等待资源、使用资源的顺序加锁等方法来预防死锁的发生，或者设计合理的死锁检测和解除机制来应对死锁情况的发生。   通过以上解决方法，可以有效地解决在理发师模型的IPC进程实验中可能遇到的问题，确保程序能够正常运行并且具有良好的性能和稳定性。 | | | |