

Глебов Павел Алексеевич. БПИ228. Вариант 22.

1. Условие задачи:

Задача о картинной галерее. Вахтер следит за тем, чтобы в картинной галерее одновременно было не более 25 посетителей. Для обозрения представлены 5 картин. Каждый посетитель случайно переходит от картины к картине, но если на желаемую картину любуются более пяти посетителей, он стоит в стороне и ждет, пока число желающих увидеть эту картину не станет меньше. Посетитель покидает галерею по завершении осмотра всех картин. Каждый посетитель уникальный, то есть имеет свой номер (например, PID). В галерею также пытаются постоянно зайти новые посетители, которые ожидают своей очереди и разрешения от вахтера, если та заполнена. Создать многопроцессное приложение, моделирующее однодневную работу картинной галереи (например, можно 19 ограничить числом посетителей от 100 до 300). Вахтер и посетители — отдельные процессы.

2. Схема решения (4-5 баллов):

1. Инициализация:

- Создаются два неименованных POSIX семафора: mutex для синхронизации доступа к общим данным и cond для сигнализации ожидающим процессам.
- Создается разделяемая память, в которой хранится информация о количестве зрителей у каждой картины и общее количество посетителей.

2. Процесс сторожа (watchman):

- Сторож следит за текущим количеством посетителей в галерее.
- Если в галерее есть место для новых посетителей, сторож создает новый процесс-посетитель.
- Каждый новый процесс-посетитель отправляет сигнал о своем прибытии, увеличивая счетчик посетителей.

3. Процесс посетителя (visitor):

- Посетитель входит в галерею и начинает просмотр картин.
- Если на какую-то картину собирается уже много посетителей, посетитель ждет, пока не освободится место.
- После просмотра всех картин, посетитель покидает галерею.

4. Взаимодействие процессов:

- Процессы посетителей создаются процессом-сторожем при необходимости.
- Семафоры mutex и cond используются для синхронизации доступа к общим данным и для управления ожиданием процессов.
- Разделяемая память используется для хранения информации о количестве зрителей у каждой картины и общем количестве посетителей.

3. Схема решения (6-7 баллов):

1. Инициализация ресурсов:

- Создание общей памяти для структуры SharedData, где будут храниться данные о посетителях и зрителях у каждой картины.
- Создание трех именованных семафоров: для мьютекса (/gallery_mutex), условной переменной (/gallery_cond), и для семафора выхода (/exit_mutex).

2. Создание процессов:

- Главный процесс создает сторожа (watchman), который отслеживает количество посетителей в галерее и разрешает новым посетителям входить.

- Сторож в своем бесконечном цикле проверяет количество посетителей и, если есть место, создает новый процесс посетителя.
- Посетитель начинает свою работу, заходит в галерею, просматривает все картины, а затем покидает галерею.

3. Синхронизация:

- Использование семафоров для синхронизации доступа к общим данным, таким как SharedData.
- Мьютекс используется для защиты критических секций кода, где происходит изменение общих данных.
- Условная переменная используется для уведомления сторожа о том, что посетитель завершил просмотр картины.
- Семафор выхода используется для обеспечения того, чтобы только один посетитель мог выходить из галереи в определенный момент времени.

4. Завершение работы:

- По завершению работы программы освобождаются все ресурсы: семафоры, общая память.