Жалилов Актан БПИ 223

Вариант 13

Условие:

Первая задача о магазине. В магазине работают три отдела, каждый отдел обслуживает один продавец. Покупатель, зайдя в магазин, делает покупки в одном или нескольких произвольных 13 отделах, обходя их в произвольном (случайном) порядке. Если в выбранном отделе продавец не свободен, покупатель становится в очередь и ожидает, пока продавец не освободится. Создать многопроцессное приложение, моделирующее рабочий день магазина. Каждого покупателя и продавцов моделировать отдельными процессами. Размер очереди не оговаривается. Считается, что для данной задачи она не ограничена (но моделирование должно быть в разумных пределах).

4-5:

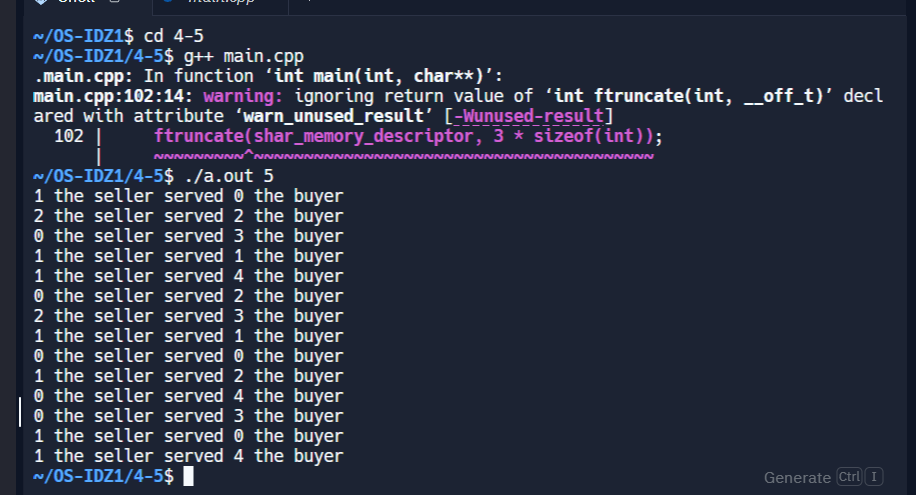
Главный процесс (родитель) инициализирует семафоры и разделяемую память.

Запускаются процессы для каждого продавца.

Каждый процесс продавца в бесконечном цикле ждёт вызова от покупателя (через семафор needed), обслуживает покупателя и сигнализирует о своей доступности (через семафор available).

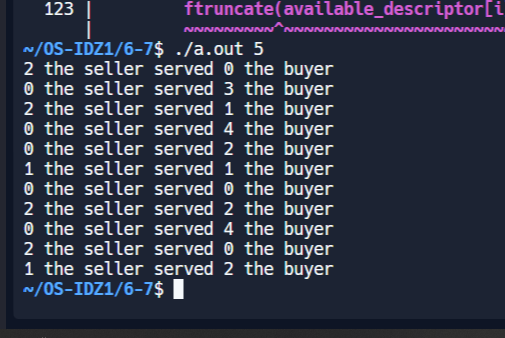
Покупатели запускаются в своих процессах. Каждый покупатель решает, сколько отделов он посетит, и для каждого отдела ожидает доступность продавца, передает информацию через разделяемую память и вызывает продавца.

После завершения работы всех покупателей, процессы продавцов завершаются, и происходит очистка всех инициализированных ресурсов.



6-7:

Аналогично 4-5



8:

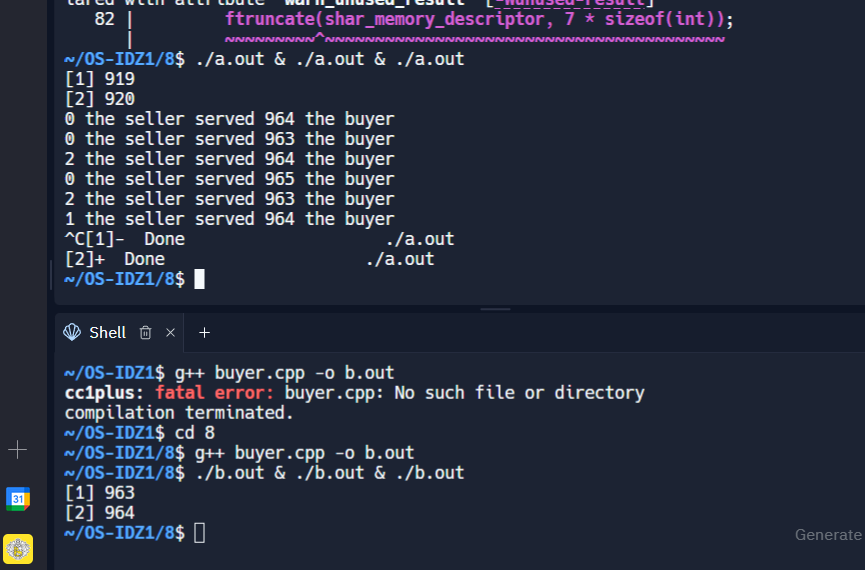
При запуске, система инициализирует разделяемую память и семафоры. В разделяемой памяти выделяются блоки для информации о продавцах и их состоянии.

Каждый продавец работает в бесконечном цикле, ожидая вызова от покупателя. После получения вызова, продавец обрабатывает запрос покупателя, используя данные из разделяемой памяти. После обработки запроса, продавец снова становится доступным для новых запросов.

Каждый покупатель при его запуске случайным образом выбирает количество отделов, которые он хочет посетить, и начинает посещение этих отделов. При посещении каждого отдела покупатель ждет, пока соответствующий продавец станет доступен, передает ему свой запрос и ожидает обработки.

Система контролирует количество продавцов с помощью семафора create\_mutex. Если места для новых продавцов закончились, новые процессы не могут быть добавлены.

При получении сигнала прерывания (например, SIGINT), система активирует функцию exit\_func, которая завершает все процессы и освобождает все ресурсы, включая разделяемую память и семафоры.



9:

В начале работы программы инициализируются все необходимые ресурсы: разделяемая память, семафоры и именованные каналы.

Создаются файлы FIFO для каждого продавца, чтобы покупатели могли отправлять данные через эти каналы.

Продавцы ждут, когда покупатель отправит запрос через соответствующий именованный канал.

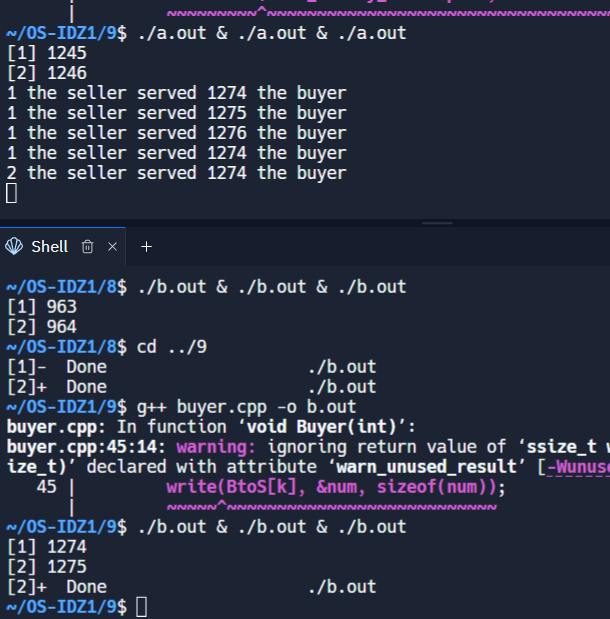
Каждый продавец ожидает семафор needed, сигнализирующий о том, что покупатель ожидает обслуживания.

После получения данных от покупателя продавец обрабатывает запрос, после чего сигнализирует через семафор available, что он снова доступен для следующего покупателя.

Покупатель случайным образом выбирает, сколько и какие отделы посетить.

Для каждого отдела покупатель ждет, пока соответствующий продавец станет доступен (семафор available), затем отправляет свой идентификатор через FIFO и сигнализирует через семафор needed.

Когда программа получает сигнал (например, SIGINT), активируется функция exit\_func, которая завершает работу всех продавцов, освобождает ресурсы, закрывает и удаляет каналы FIFO, а также уничтожает разделяемую память и семафоры.



10:

Аналогично 9

