

Задание:

Дочерний процесс (клиент) должен ожидать нажатия клавиши для завершения работы. Обмен данными между родительским и дочерними процессами должна осуществляться через механизмы непосредственной передачи данных.

Функционал с управляющим процессом из 2-ой лабораторной работы должен присутствовать в полной мере (реализованы операции добавления/удаления). Родительский процесс (сервер) осуществляет вывод уникальной строки активного дочернего процесса (строки должны быть разными). При этом операция вывода строки должна быть атомарной, т.е. процесс вывода должен быть синхронизирован таким образом, чтобы строки на экране не перемешивались. В качестве метода синхронизации использовать семафоры. Реализация механизма непосредственной передачи данных должна быть разной под разные ОС (каналы под одну ОС, разделяемая память/отображаемая память под другую). Данное задание является развитием 2-ой лабораторной работы.

Требования следующие:

1. Реализация непосредственно основной части задания должна быть максимально платформонезависимой.
2. Лабораторные должны быть выполнены в двух вариантах: Unix/Windows.
3. Код должен быть читабельным.

Варианты:

1. Реализовать взаимодействие кофейного аппарата и человека. Человек может опустить в аппарат определенную сумму денег и сделать выбор. Кофейный аппарат может как выдавать различные типы напитков (минимум 2), так и проинформировать пользователя об отсутствии какого-либо напитка либо нехватке средств. Состояние кофейного аппарата сохраняется, т.е. следующий пользователь может не успеть взять определенный напиток, т.к. последнюю порцию забрал предыдущий человек.

Клиент - человек.

Сервер - кофейный аппарат.

2. Реализовать взаимодействие банкомата и человека. Человек может запросить определенную сумму денег в одной из валют (максимум оставляю на Ваше усмотрение, минимум — 2 валюты). Банкомат может либо выдать деньги, либо проинформировать пользователя о нехватке денег. Состояние банкомата после взаимодействия должно сохраниться.

Клиент - человек.

Сервер - банкомат

3. Реализовать взаимодействие заправки и автомобиля. Автомобиль может запросить определенное количество одного типа (минимум — 2). Заправка в свою очередь может удовлетворить потребность полностью либо, в случае нехватки запрошенного количества, залить все, что есть, и проинформировать автомобиль об этом, т.е. вернуть лишние деньги и выдать сообщение об ошибке. Состояние заправки должно сохраняться.

Клиент - автомобиль.

Сервер - заправка

4. Реализовать взаимодействие человека и таксофона. Человек может опустить в аппарат определенную сумму и денег и запросить нужное количество минут разговора определенного типа (локальный/межгород). Таксофон может выдать сообщение об ошибке (количество денег не соответствует количеству запрошенных минут нужного типа) или допустить человека к разговору. Состояние таксофона сохраняется.

Клиент - человек.

Сервер - таксофон.

5. Реализовать взаимодействие человека и архивной системы. Архивная система

владеет большим текстовым файлом. Человек передает в систему определенные правила, согласно которым нужно искать строчки из файла. Система работает минимум с двумя правилами, в противном случае - выдает пользователю ошибку. Если правила заданы правильно, система сигнализирует об успешном нахождении данных. Состояние архивной системы сохраняется.

Клиент - человек.

Сервер - архивная система.

6. Реализовать взаимодействие покупателя и аптекаря. Покупатель может отдать определенную сумму денег и попросить определенное лекарство (минимум два типа — с рецептом и без рецепта). Аптекарь может как выдать лекарство, так и проинформировать покупателя об отсутствии какого-либо лекарства либо нехватке средств. Состояние аптеки сохраняется.

Клиент - покупатель.

Сервер - аптекарь.

7. Реализовать взаимодействие пользователя и телефона. Пользователь может попробовать совершить звонок либо отправить смс, телефон, в свою очередь, может выполнить операцию, либо проинформировать пользователя о том, что средств на счете не хватает. Состояние счета должно сохраняться.

Клиент - пользователь.

Сервер - телефон.

8. Реализовать взаимодействие человека и почты. Человек может отдать определенную сумму денег и запросить операцию (отправка посылки/письма). На почте могут выполнить операцию, либо проинформировать о нехватке средств. Почта отправляет только 5 посылок и 10 писем в день. Состояние должно сохраняться.

Клиент - человек.

Сервер - почта.

9. Реализовать взаимодействие человека и ресторана. Человек может отдать определенную сумму денег и сделать выбор. Ресторан может как выдавать различные типы продуктов (минимум 2), так и проинформировать человека об отсутствии какого-либо продукта либо нехватке средств. Состояние ресторана сохраняется.

Клиент - человек.

Сервер - ресторан.

10. Реализовать взаимодействие покупателя и овощного магазина. Покупатель может отдать определенную сумму денег и попросить определенное количество овощей (минимум два типа). Продавец может как отдать овощи, так и проинформировать покупателя об отсутствии какого-либо типа овощей либо нехватке средств. Состояние магазина сохраняется.

Клиент - покупатель.

Сервер - овощной магазин.

11. Реализовать взаимодействие покупателя и сервиса продажи музыки. Покупатель может платить определенную сумму денег и запросить определенное количество альбомов различных жанров (минимум два типа). Сервис может как выполнить продажу, так и проинформировать покупателя об отсутствии альбома какого-либо жанра либо нехватке средств. Состояние магазина должно сохраняться.

Клиент - покупатель.

Сервер - сервис продажи музыки.

12. Реализовать взаимодействие студента и преподавателя. Студент может сдавать определенное количество лабораторных по какому-либо предмету (минимум два). Преподаватель может как принять лабораторные, так и проинформировать студента об отсутствии времени на прием лабораторных определенного предмета либо об отсутствии этого предмета в нагрузке. Состояние занятости преподавателя должно сохраняться.

Клиент - студент.

Сервер - преподаватель.

13. Реализовать взаимодействие вендинг машины и человека. Человек может опустить в машину определенную сумму денег и сделать выбор. Машина может как выдавать различные типы продуктов (минимум 2), так и проинформировать пользователя об отсутствии какого-либо продукта либо нехватке средств. Состояние машины сохраняется.

Клиент - человек.

Сервер - вендинг машина.

14. Реализовать взаимодействие программиста и компилятора. Программист может запросить у компилятора собрать программу определенного типа. Компилятор может как собрать ее и проинформировать пользователя, так и выдать сообщение о том, что он не поддерживает определенный тип файлов. Придумать дополнительный тип файлов, на которых компилятор будет находить опасный момент и сигнализировать об этом пользователю. Состояние компилятора сохраняется.

Клиент - программист.

Сервер - компилятор.

15. Реализовать взаимодействие пользователя и сервиса скачивания файлов. Пользователь может запросить у сервиса скачивания файла по определенной URL (просто какая-то строка). Если URL не соответствует определенным правилам (придумайте сами), процесс скачивания рапортует об ошибке. Иначе сигнализирует о успешном скачивании. Состояние сервиса сохраняется.

Клиент - пользователь.

Сервер - сервис скачивания файлов.