Понятие мультиплексирования. Мультиплексирование сообщений. Модель взаимодействия процессов клиент — сервер. Неравноправность клиента и сервера

Используя технику из предыдущего примера, вы можете организовать получение сообщений одним процессом от множества других процессов через одну очередь сообщений и отправку им ответов через ту же очередь сообщений, т.е. осуществить *мультиплексирование* сообщений. Вообще под мультиплексированием информации понимают возможность одновременного обмена информацией с несколькими партнёрами. Метод мультиплексирования широко применяется в модели взаимодействия процессов *клиент–сервер*. В этой модели один из процессов является сервером. Сервер получает запросы от других процессов – клиентов – на выполнение некоторых действий и отправляет им результаты обработки запросов. Чаще всего модель клиент–сервер используется при разработке сетевых приложений, с которыми вы столкнётесь в материалах завершающих семинаров курса. Она изначально предполагает, что взаимодействующие процессы неравноправны:

1. Сервер, как правило, работает постоянно, на всем протяжении жизни приложения, а клиенты могут работать эпизодически.
2. Сервер ждёт запроса от клиентов, инициатором же взаимодействия является клиент.
3. Как правило, клиент обращается к одному серверу за раз, в то время как к серверу могут одновременно поступать запросы от нескольких клиентов.
4. Клиент должен знать, как обратиться к серверу (например, какого типа сообщения он воспринимает) перед началом организации запроса к серверу, в то время как сервер может получить недостающую информацию о клиенте из пришедшего запроса.

Рассмотрим следующую схему мультиплексирования сообщений через одну очередь сообщений для модели клиент–сервер. Пусть сервер получает из очереди сообщений только сообщения с типом 1. В состав сообщений с типом 1, посылаемых серверу, процессы-клиенты включают значения своих идентификаторов процесса. Приняв сообщение с типом 1, сервер анализирует его содержание, выявляет идентификатор процесса, пославшего запрос, и отвечает клиенту, посылая сообщение с типом, равным идентификатору запрашивавшего процесса. Процесс-клиент после отправления запроса ожидает ответа в виде сообщения с типом, равным своему идентификатору. Поскольку идентификаторы процессов в системе различны, и ни один пользовательский процесс не может иметь PID равный 1, все сообщения могут быть прочитаны только теми процессами, которым они адресованы. Если обработка запроса занимает продолжительное время, сервер может организовывать параллельную обработку запросов, порождая для каждого запроса новый процесс-ребенок или новую нить исполнения.

Задачи на семинар

***Задача 1 (10 баллов)**:*

Напишите 2 программы для предложенной схемы мультиплексирования сообщений: клиент и сервер. Клиент вводит с клавиатуры число с плавающей точкой и отправляет серверу. Дожидается от сервера ответа на запрос – это будет число с плавающей точкой – и печатает его на экране. Сервер принимает от клиентов запросы. Полученное от клиента число возводится в квадрат, и результат отправляется клиенту.

***Подзадача 1 (+2 бонусных балла)**:*

Написать сервер так, чтобы одновременно не могли работать 2 и более экземпляров сервера.

***Подзадача 2 (+5 бонусных баллов)**:*

*Написать отдельную программу – киллер – после получения сообщения от которой сервер аккуратно бы завершал свою работу. **Для этого в сервере необходимо использовать политику выборки с порогом**. Киллер должен посылать сообщение серверу с типом отличным от типа сообщений клиентов. Аккуратное завершение подразумевает возможность запуска нового экземпляра сервера сразу после завершения работавшего.*

***Примечание**:*

Бонусные баллы не учитываются в максимально возможном количестве баллов за семестр.

Домашнее задание

Дедлайн для вашей группы прописывается в таблице с результатами. Сдать задание нужно до наступления указанного дня, сдавать в указанную дату уже поздно. Проверка производится один раз, штрафы за недочеты выставляются сразу.

***Задача 1 (10 баллов)**:*

*Решите задачу из домашнего задания 3, используя для синхронизации **одну** очередь сообщений. Формулировка задачи:*

Рассмотрим катер, совершающий кольцевые туристические рейсы по реке Москве. Катер с утра приплывает на пристань и совершает ровно K поездок. В плавание он не отправляется, пока не будут заполнены все места, вместимость катера равна N (число N относительно невелико). Считаем, что пассажиры совершают поездку по одиночке, а не в компании. Пассажир приходит на пристань, ждет прибытия катера и выхода прибывших пассажиров, и после этого в порядке очереди заходит на катер. Трап узкий – поэтому все пассажиры заходят друг за другом. Когда все места заполнены — выполняется путешествие, после прибытия обратно пассажиры выгружаются строго по одному. Повторные поездки пассажиры не совершают.

*Смоделируйте поведение катера и каждого из пассажиров отдельными процессами, синхронизировав их действия с помощью **одной** очереди сообщений System V IPC. Для этого напишите две программы – одна из них моделирует поведение катера, совершающего K поездок, другая генерирует N*K идентичных процессов, моделирующих поведение пассажиров. И катер, и пассажиры должны детально докладывать о своих действиях. Считать при тестировании, что N=5, K=3.*