

Национальный исследовательский университет «Высшая школа экономики»
Факультет компьютерных наук
Программная инженерия

Операционные системы

Отчёт по индивидуальному домашнему заданию №3
Вариант 10

Работу выполнил:

А. В. Андронов

Группа: БПИ-213

Преподаватель:

А. И. Легалов

Москва
2023

Содержание

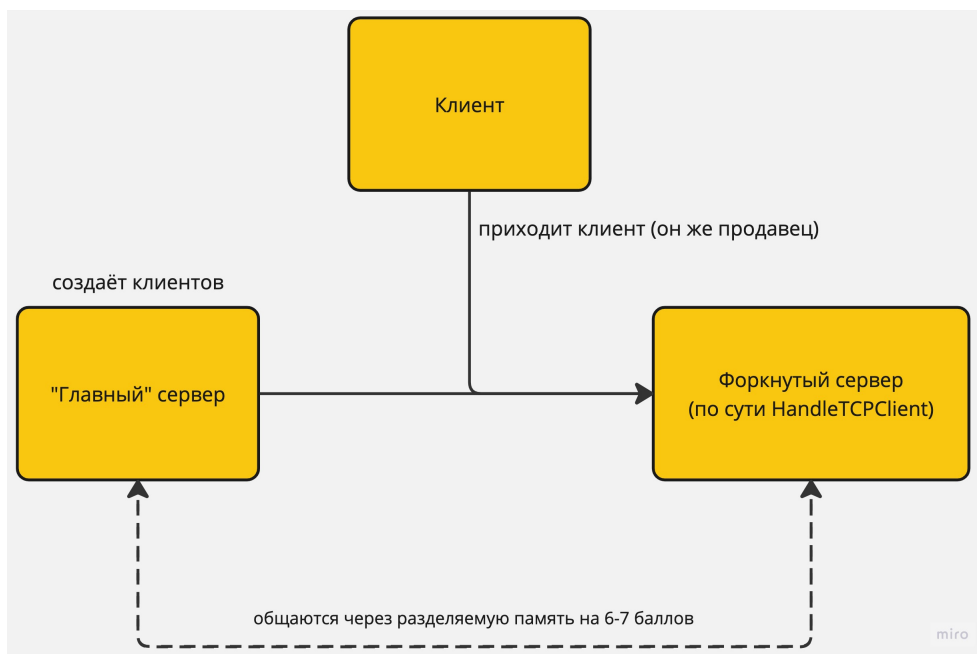
Постановка задачи	3
Общая схема решения	3
4-5 баллов	4
6-7 баллов	5

Постановка задачи

В магазине работают три отдела, каждый отдел обслуживает один продавец. Покупатель, зайдя в магазин, делает покупки в одном или нескольких произвольных отделах, обходя их в произвольном порядке. Если в выбранном отделе продавец не свободен, покупатель становится в очередь и засыпает, пока продавец не освободится. Создать приложение, моделирующее рабочий день магазина. Каждый продавец — клиент, обслуживающий покупателя случайное время, Сервер создает покупателей, «сопровождает» их случайным образом по отделам и обеспечивает завершение обхода магазина.

Программы должны быть написаны на языке программирования C и выполняться в среде ОС Linux

Общая схема решения



4-5 баллов

Разработать клиент–серверное приложение, в котором сервер (или серверы) и клиенты независимо друг от друга отображают только ту информацию, которая поступает им во время обмена. То есть, отсутствует какой-либо общий вывод интегрированной информации, отображающий поведение системы в целом.

Желаемое количество покупателей задаётся с помощью аргумента командой строки при запуске сервера.

Каждый покупатель являет собой по сути набор из трёх чисел - порядок продавцов, к которым он пойдёт, например, клиент с параметрами 1 0 3 - в первую очередь такой клиент пойдёт к первому продавцу, затем не пойдёт никуда (смысл кода 0), то есть пропустит поход, и затем пойдёт к третьему продавцу.

При работе с программами на 4-5 нужно сначала запустить сервер, он создаст заданное количество покупателей, затем, при запуске клиента (продавца) нужно указать его номер (параметр <Number of Seller>). Клиент пройдёт по всем созданным покупателям и, если кто-то из них сейчас стоит у него в очереди, он потратит на них какое-то время и отправит к следующему продавцу в их списке. Таким образом можно детально просматривать ход работы программы в целом.

Стоит сразу обозначить, что из-за ограниченной точности функции `time`, `srand(time(NULL))` для многих, казалось бы, разных моментов времени, генерируется один и тот же сид, поэтому было принято решение делать `sleep(1)` после генерации каждого клиента, а также было добавлено поведение, при котором один покупатель не может несколько раз пойти к одному продавцу.

Сервер закрывается по нажатию `Ctrl+C`.

6-7 баллов

В дополнение к программе на предыдущую оценку необходимо разработать клиентскую программу, подключаемую к серверу, которая предназначена для отображение комплексной информации о выполнении приложения в целом. То есть, данный программный модуль должен адекватно отображать поведение моделируемой системы, позволяя не пользоваться отдельными видами, предоставляемыми клиентами и серверами по отдельности.

Был добавлен файл `observer.c`. Изменился запуск: запускаем сервер как раньше, затем, до запуска (!) первого клиента (продавца), запускаем `observer`, передаём IP-адрес и то же количество клиентов, что и серверу, затем дальше работаем с клиентами как раньше. В терминале с наблюдателем будут отображаться все изменения в очередях, для каждого клиента зелёным подсвечен номер продавца, в чьей очереди он сейчас находится. Завершение наблюдателя происходит когда все покупатели прошли все свои очереди, при этом же посылается сигнал серверу завершить работу (можно реализовать прямо в сервере, но так менее интересно, да и наблюдательно-модерирующая роль `observer` уменьшилась бы).