Дмитриев Дмитрий Игоревич, БПИ227

**Вариант 20. Условие задачи:** На клумбе растет 10 цветов, за ними непрерывно следят два садовника и поливают увядшие цветы, при этом оба садовника очень боятся полить один и тот же цветок, который еще не начал вянуть. Иначе он пропадет из–за перелива.

Создать клиент–серверное приложение, моделирующее состояния цветов на клумбе и действия садовников.

Сервер используется для обмена информацией между садовниками и клумбой. Клумба — клиент, отслеживающий состояния всех цветов. Каждый садовник — отдельный клиент.

**Претендую на 10 баллов за выполнение задания**

[**Репозиторий с кодом**](https://github.com/TIN-slayer/OS_IDZ_3)

**Команды для запуска приложения:**

gcc Server.c -o server

gcc GardenerClient.c -o gardener

gcc FlowersClient.c -o flowers

gcc BeholderClient.c -o beholder (от 6 баллов)

./server 2000 (запуск на 1-ом Терминале)

./gardener 0.0.0.0 2000 (запуск на 2-ух Терминалах, чтобы получить 2 Садовника)

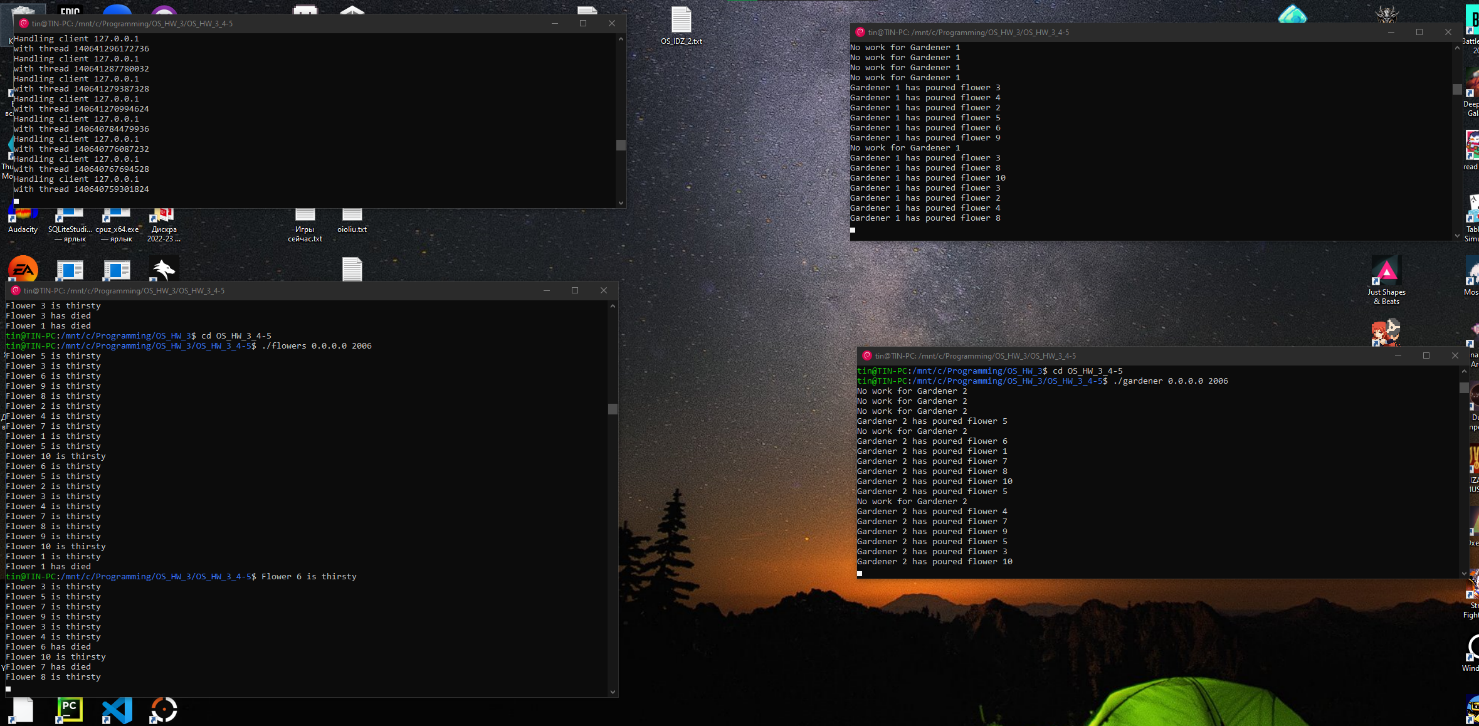
./flowers 0.0.0.0 2000 (запуск на 1-ом Терминале)

./beholder 0.0.0.0 2000 (запуск от 6 баллов на 1 Терминале, от 8 на N Терминалах)

**Прогрессирующий отчёт на баллы от 4 до 10:**

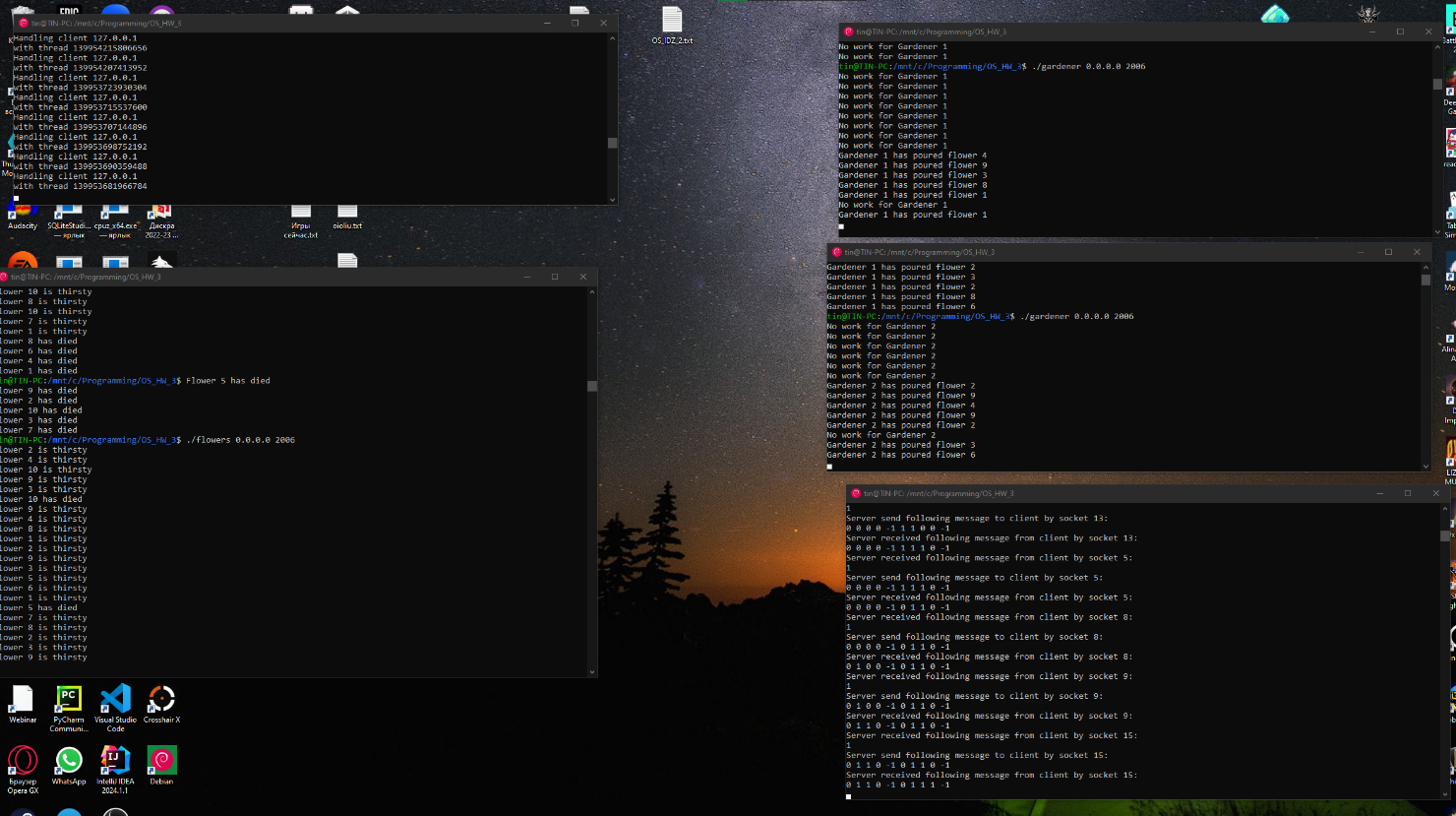
**4-5 баллов:**

1. Есть сервер Server.c, который передаёт информацию между клиентами FlowersClient.c и 2-мя клиентами GardenerClient.c.
2. В FlowersClient.c я делаю 10 процессов под каждый цветок, который будет управлять его циклом жизни.
3. Работа процесса цветка: в рандомное время от 1 до 10 секунд цветок начинает хотеть пить и уведомляет об этом садовников с помощью сообщения на сервер, если его не успеют полить до того момента, когда он снова захочет пить, он умрёт.
4. Работа садовника: садовники следят за цветами, запрашивая их состояние у сервера. Они по очереди начинают проходиться по цветкам (чтобы 2 садовника одновременно не начинали поливать цветы, я блокирую доступ к информации о цветах с помощью mutex сервере) в рандомном порядке и если цветок хочет пить, то он его поливает, после чего садовник уходит на отдых (1-2 секунды).
5. Немного подробнее об общении сервера и клиентов. Сервер у меня разделен на потоки: главный поток, который принимает подключения, а потом делает отдельный поток сервера под каждого клиента. Эти потоки ждут сообщения от клиента, что ему нужна информация о состоянии цветов, после чего он им её отправляет. Так как у меня есть общая информация о состоянии цветов для всех потоков сервера, я блокирую с помощью мьютексов при обращении к ней.
6. IP адрес и порты задаю через командную строку.
7. Сервера и клиенты могут работать удалённо благодаря TCP.
8. Результаты работы приложения:



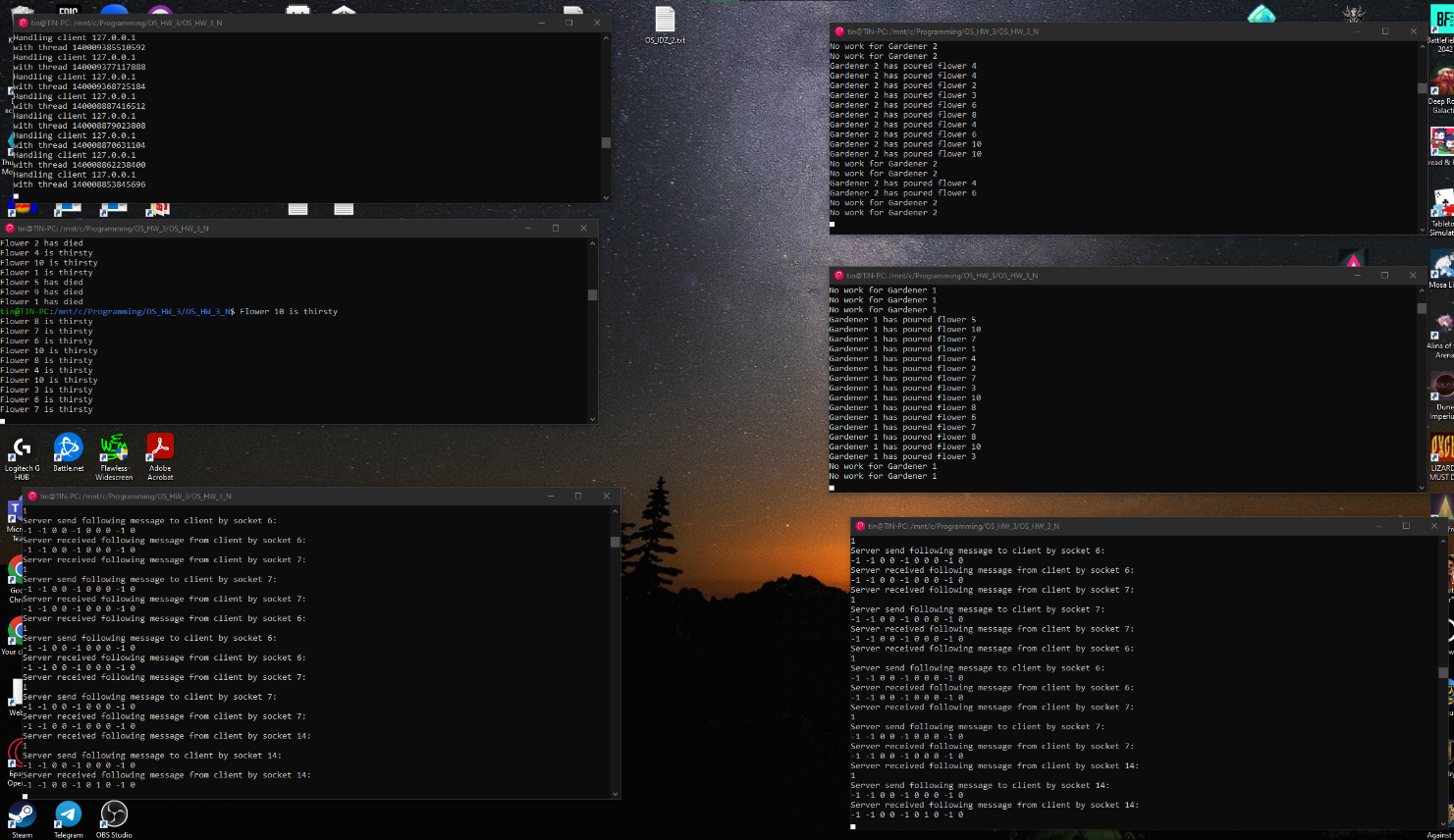
* 1. **баллов:**

1. Добавил клиента BeholderClient.c, который запрашивает у сервера, что он отправлял и получал, и выводит это.
2. Результаты работы приложения:



**8 баллов:**

1. Добавил возможность подключения нескольких клиентов смотрителей BeholderClient.c.
2. Результаты работы приложения:



**9-10 баллов:**

1. Добавил возможность безопасного отключения и подключения клиентов без потери работоспособности сервера. Также при отключении сервера происходит корректное отключение всех его клиентов.
2. Результаты работы приложения:

