**Отчет по ИДЗ №3**

**По предмету “Операционные системы”**

**Морина Ильи Олеговича**

**БПИ215**

**Вариант 30**

**Тема «Сетевые взаимодействия с применением транспортного протокола TCP»**

**Задание (4 – 5):** 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.

**Задание (6 – 7):** 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

Задача о Пути Кулака. На седых склонах Гималаев стоит древний буддистский монастырь: Гуань-Инь-Янь. Каждый год в день сошествия на землю боддисатвы Монахи монастыря собираются на совместное празднество и показывают свое совершенствование на 11 Пути Кулака. Всех соревнующихся монахов разбивают на пары, победители пар бьются затем между собой и так далее, до финального поединка. Монах который победил в финальном бою, забирает себе на хранение статую боддисатвы. Реализовать приложение, определяющего победителя. В качестве входных данных используется массив, в котором хранится количество энергии Ци каждого монаха. При победе монах забирает энергию Ци своего противника. Каждый бой длится некоторое случайное время. Разбивка на пары перед каждым сражением осуществляется случайным образом и асинхронно. То есть, очередную пару образуют первые освободившиеся от боя монахи. Но бои следующего круга начинаются одновременно. Монах, оставшийся без пары, удваивает свою энергию, отдохнув от поединка и является первым для последующего выбора. Каждый монах — отдельный клиент. Сервер используется для распределения пар и формирования результатов поединка. Проигравший монах–клиент отключается. Победивший — продвигается дальше с новой энергией.

(Далее слово монах будет заменено на игрок или клиент)

**В качестве победителя каждого боя и игры в целом выбирается игрок с большим количеством энергии.**

**Схема работы программы**

Программы представляют собой серверную и клиентскую части игры, основанной на сокетах. В игре принимают участие три игрока, которые борются друг с другом.

1. **Сервер**: Серверная программа начинает с открытия сокета и привязки его к определенному IP-адресу и порту. Это делается с использованием команды bind(). Затем сервер начинает прослушивание порта на наличие входящих подключений с помощью listen(). Сервер принимает подключения от трех клиентов с помощью accept(), и каждый клиентский сокет сохраняется в массиве client\_sockets. Когда все клиенты подключены, игра начинается. Игроки получают случайное начальное количество энергии и одному из игроков разрешается отдыхать, удваивая его энергию. В каждом раунде случайно выбираются два игрока, и их энергии сравниваются. Игрок с большим количеством энергии побеждает, и его энергия увеличивается на количество энергии проигравшего игрока. Проигравший игрок исключается из игры. Это продолжается, пока в игре не останется только один игрок. В этом случае игра заканчивается, и оставшийся игрок объявляется победителем. Наконец, сервер закрывает все клиентские сокеты и свой собственный сокет.
2. **Клиент**: Клиентская программа начинает с открытия сокета и подключения его к серверу. Это делается с использованием команды connect(). После успешного подключения к серверу, клиент начинает читать сообщения от сервера. Это сообщения могут содержать информацию о текущем статусе игры, такую как текущее количество энергии игрока, результаты боев и т.д. Клиент продолжает читать сообщения от сервера, пока не получит сообщение о том, что он проиграл. После этого клиент закрывает свой сокет и завершает работу.

Важно отметить, что в текущей реализации игроки не могут влиять на процесс игры, а все события (выбор игроков для боя, результаты боев и т.д.) определяются случайным образом на сервере.

**Инструкция по работе с программой**

Для упрощения тестирования и отчасти логики программы, было принято решение сделать программу, которая осуществляет игру между тремя клиентами (меньшее или большее кол-во клиентов не поддерживается).

Для работы программы необходимо скомпилировать код сервера и запустить с указанием IP и порта как указано на скриншоте (ip замазан для конфиденциальности).



После того как сервер запущен приступаем к подключению клиентов. Код так же компилируется и запускается с указанием ip и порта. После того как клиент подключается эта информация отображается в терминале сервера. Бои начнутся только после того как подключатся 3 клиента.

P.S.

Автор понимает, что работа выполнена далеко не супер качественно и скорее всего отвечает не всем требованиям, но все же смеет указать, что желаемая оценка 5+.