**Отчет по ИДЗ №4**

**По предмету “Операционные системы”**

**Морина Ильи Олеговича**

**БПИ215**

**Вариант 30**

**Тема: «Сетевые взаимодействия с применением транспортного протокола UDP»**

**Условие задачи:**

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.

**Задача на оценку 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.

**Схема работы разработанной программы:**

Взаимодействие между сервером и клиентом в программе осуществляется по протоколу UDP. Ниже приведена схема работы программы на сервере и клиенте.

**Сервер:**

1. Создание сокета сервера (socket()).
2. Привязка сокета к определенному IP-адресу и порту (bind()).
3. Ожидание подключения клиентов.
4. Инициализация информации о каждом игроке (энергия, состояние жизни).
5. Выбор игрока, который будет отдыхать.
6. Прием и обработка инициализационных сообщений от клиентов (recvfrom()).
7. Отправка инициализационных сообщений каждому клиенту (sendto()).
8. Проведение сражений между игроками до тех пор, пока не останется один живой игрок.
9. Определение победителя (findWinner()).
10. Вывод сообщения о победителе и завершение игры.
11. Закрытие сокета сервера (close()).

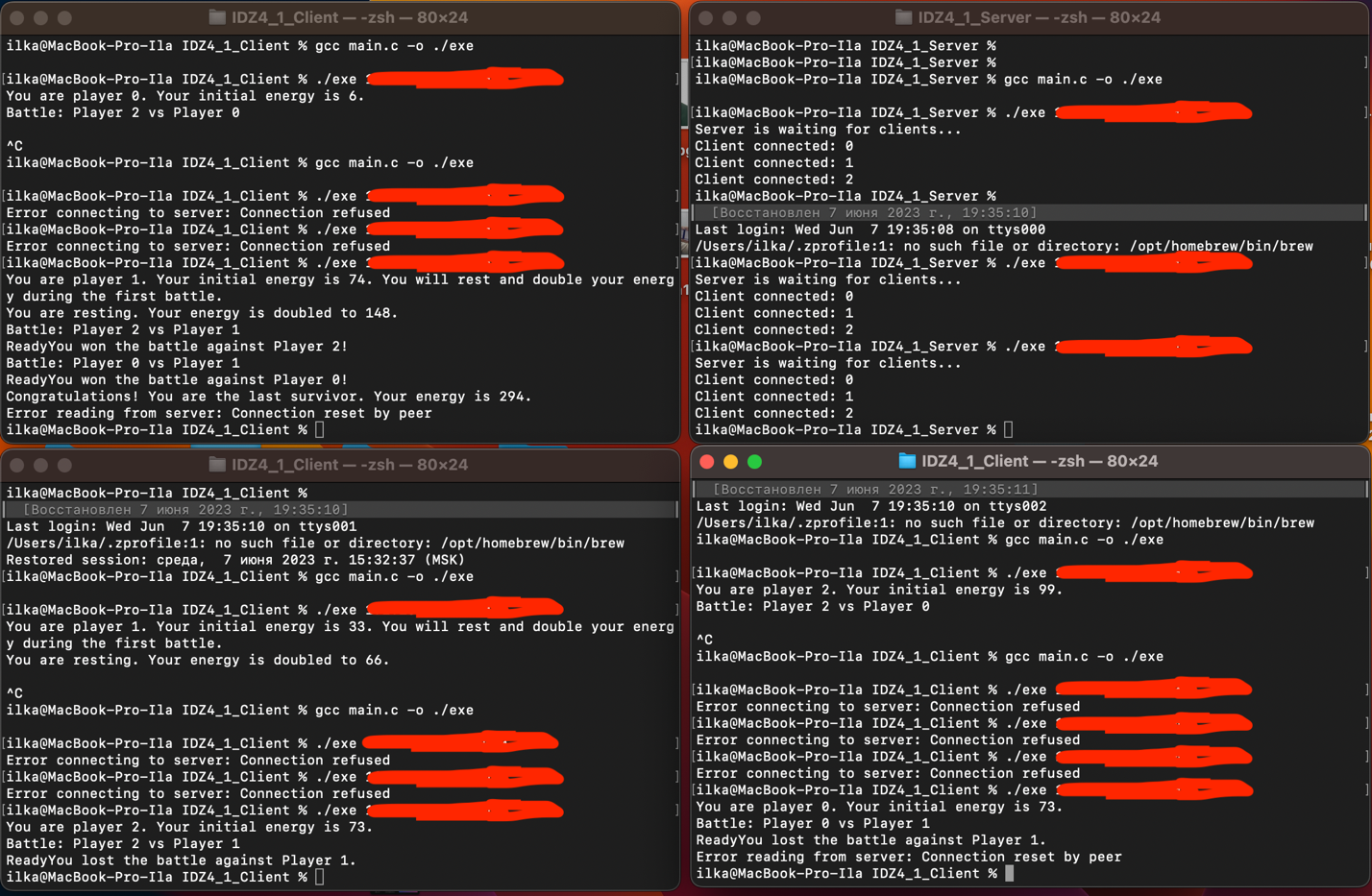
**Клиент:**

1. Создание сокета клиента (socket()).
2. Отправка начального сообщения серверу (sendto()).
3. Прием и обработка сообщений от сервера в бесконечном цикле.
4. Проверка, является ли сообщение сообщением о завершении игры.
5. Вывод принятого сообщения.
6. Закрытие сокета клиента (close()).

**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).



P.S. Автор пытался выполнить ДЗ на оценку 6-7, но будет рад любой оценку свыше или равной 4 ☺