OS final project report

Hamidreza Mafi
610399209

January 2024

1 how it works :

First, each client selects a unique name for itself. Then, it requests the server to check if the chosen name is available. In order for two clients to engage in a game with each other, one of them must wait for the other to invite them to play. Additionally, clients can discover other available players by the list of available players provided by the server.

2 server implementation :

Initially, the server begins by accepting new connections through sockets to accommodate new clients. When a new client is accepted, a new thread is created for that client, which initializes a player class. This class continuously listens to the player's socket in a loop and handles all the client's requests.

At the start of each game round, the party leader selects the type of game to play and creates an instance of the game class, which monitors moves and game completion. When the game concludes, the status of both players changes back to available.

The server contains data that multiple threads need to access, so semaphores are employed to prevent race conditions.

3 client implementation :

On the client side, the current game state is stored to be displayed on the window. Every action that players make is sent to the server, and the client handles the server's responses, including game state updates, game endings, and game starts.no computation is done on the client side.

4 challenges :

The issue of players from different threads on the server side needing to find other players' class instances was resolved by using a shared list of players.