Задание на лабораторную работу №6.

Общие моменты для задачи, справедливые для всех вариантов:

При вызове программы клиента задается hostname (или IP) и порт для связи с сервером (можно в самой программе). Передача информации между клиентом и сервером происходит через потоковые сокеты (TCP). Для хранения главной структуры (struct) программы использовать разделяемую память, доступ к которой есть у всех процессов сервера. Для блокирования доступа к разделяемой памяти использовать семафоры. Сервер оформить в виде демона с логированием действий сервера. На сервере всегда существует master-процесс, который создает и уничтожает дочерние процессы-worker’ы. Из конфигурационного файла сервер читает порт, по которому происходит взаимодействие, а также имя файла для лога (к прочитанному имени файла в конец добавляется временная метка, полученная во время чтения (или перечитывания) конфигурационного файла). Сервер всегда принимает SIGTERM для корректного завершения работы (необходимо отдать результат работы клиенту, после этого только завершать работу сервера), при получении SIGHUP сервер перечитывает конфигурационный файл, обработка SIGHUP блокируется на время активных действий сервера. Для генерации любых случайных чисел использовать /dev/urandom.

Для всех вариантов с 2 клиентами алгоритма старта игры следующий: 1-ый клиент будет создавать игру, для этого он пишет команду «start {password}», где {password} – пароль от игры, которая будет создана. Второй клиент подключается к игре, вводя команду «join {password}». Считаем, что сервер может держать только 1 активную игру. Если игр на сервере не существует клиент ввел неправильный пароль или присоединилось максимальное число игроков (2), сервер должен уведомить об этом клиента. Общий алгоритм всех таких игр: в начале возможна генерация сервером некоторых начальных условий (положение кораблей, загаданное слово, число и т.д.), мастер для этого создает воркера, который генерирует то что нужно и записывает в главную структуру, после чего начинается фаза «перестрелки» - клиент 1 делает какое-то действие (вводит команду), затем клиент 2 делает какое-то действие, после этого на сервере происходит обработка текущей ситуации (это цикличная фаза). Предусмотреть, чтобы клиент, который пытается сделать ход в не в свой черед, получил соответствующее уведомление от сервера (такие проверки делает отдельный воркер). Соответственно, когда воркер проверил команду на синтаксическую и логическую корректность, через очередь сообщений он отсылает мастеру сообщение, в зависимости от которого мастер либо шлет клиенту сообщение об ошибке, либо создает воркера, которому через очередь сообщений посылает команду. Воркер не удаляется, пока не обработает обе команды (обрабатывает он их и отдает результат по отдельности, кроме игр «Угадай число» и «Тривиадор», в которых результат хода оба клиента должны получить только после того, как оба сделают ход). Соответственно после обработки либо 1 команды, либо сразу 2 (в зависимости от игры) воркер обновляет необходимые структуры данных, кладет в очередь сообщений сообщения с текстом для клиентов, мастер вычитывает их и посылает клиентам. Если воркер определил конец игры, он удаляет игру и посылает клиентам результат игры.

Можно реализовать команду disconnect (по желанию), которая позволит отключиться от игры, которую сервер автоматически расформирует.

Вариант 7\*\*\*. (Артемьев) Реализация двусторонней игры (2 клиента) «Гомоку» («Крестики-нолики» 5 в ряд).

Размер поля задается двумя константами в программе. Изначально поле пустое, игрок, создавший игру, ходит Х, присоединившийся игрок ходит О. После каждого хода каждого клиента у обоих клиентов выводится игровое поле, оно при этом анализируется на наличие 5 подряд знаков одного типа горизонтально, вертикально или по диагонали для определения конца игры.

Вариант 8\*\*\*. (Гречин) Реализация двусторонней игры (2 клиента) «Морской бой».

Расстановки кораблей генерирует воркер для обоих клиентов и записывает их в главную структуру. После каждого хода поле проверяется на конец игры и выводится у обоих клиентов. Игра заканчивается, когда у кого-то из игроков уничтожены все корабли.

Вариант 9\*\*\*. (Попов) Реализация двусторонней игры (2 клиента) «Серия пенальти»

Считаем, что ворота разделены на 10 зон. Все зоны пронумерованы и указаны в таблице. Зона 0 – зона реагирования вратаря. Если у зоны рядом с номером стоит %, это процент попасть в створ при выборе этой зоны для удара. Считаем, что шанс попадания в зоны 0, 2, 3 при их выборе равен 100%. Но для баланса вводится следующее правило: если вратарь выбрал зону 0 (то есть реагировать на удар), а мяч летит в зону 2 или 3 у него есть 70%-й шанс отбить мяч. Если мяч летит в зоны 1 или 4, а вратарь выбрал зону 0, то шанс отбить мяч 50%. Принцип игры следующий: 1-й игрок выбирает зону для удара, 2-й игрок после этого выбирает зону для отбивания. Считаем, что если вратарь выбрал правильную зону, то он со 95% вероятностью отобьет мяч, если он туда полетит.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 (50%) | 4 (80%) | 0 | 1 (80%) | 5 (50%) |
| 9 (60%) | 0 | 0 | 0 | 6 (60%) |
| 8 (70%) | 3 | 0 | 2 | 7 (70%) |

После каждого удара пишется выбранная пенальтистом зона, выбранная вратарем зона и ситуация, которая произошла (гол, сейв, промах), а также текущий счет. После этого 2-й игрок становится пенальтистом, а 1-й вратарем. Первые 5 серий бьются вне зависимости от счета (так проще для реализации), а если за это время победитель не был выявлен, то бьется по 1 пенальти с каждой стороны, проверяется, выявлен ли победитель и так до победного (стандартные правила серии пенальти).

Вариант 10\*\*\*. (Хисамутдинов) Реализация двусторонней игры (2 клиента) «Поле чудес».

Сервером загадывается слово. Список слов можно загрузить чтением из файла. Для игрока крутится барабан, на котором можно оставить только очки (N значений) и банкрот. Сервер «крутит» барабан, если выпал банкрот, то очки игрока обнуляются и ход переходит к сопернику. Иначе клиент имеет право назвать букву (проверку на повторение букв можно не делать). Если буква отгадана, то игрок продолжает ходить. На любом ходу игрок может попытаться назвать слово командой «answer {слово}», если ответ правильный, игра окончена, иначе игрок удаляется со стола и соперник может ходить, пока не отгадает все. Когда игра окончена, сервер выводит слово, победителя и его количество очков. После каждой отгаданной буквы сервер выводит загадку со всеми отгаданными буквами (как в самом шоу). Считаем, что игрок всегда отказывается от супер-игры (её можно не реализовывать).