Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Кафедра информационных технологий и систем

**Сетевая игра “Покер”**

Курсовая работа

по дисциплине:

«Операционные системы»

Разработал:

Студент группы 0091

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_».\_\_\_\_\_.2022г.

Проверил:

Фуфин М.Н.\_\_\_\_\_\_

«\_\_».\_\_\_\_\_.2022г.

**Великий Новгород**

**2022**

Оглавление

[ВВЕДЕНИЕ 3](#_Toc103672176)

[РАЗРАБОТКА ПРОЕКТА. 4](#_Toc103672177)

[1. Постановка задачи. 4](#_Toc103672178)

[2. Выбор инструментальных средств. 4](#_Toc103672179)

[3. Выбор способа взаимодействия потоков. 4](#_Toc103672180)

[4. Выбор библиотеки для реализации многопоточной модели 4](#_Toc103672181)

[5. Алгоритм решения задачи. 4](#_Toc103672182)

[6. Вывод 5](#_Toc103672183)

# ВВЕДЕНИЕ

Целью данной работы является создание сетевой игры в покер от 2-х до 6 игроков. Разработанные программы под системы Windows.

# РАЗРАБОТКА ПРОЕКТА.

## **1. Постановка задачи.**

Задачей данной курсовой работы является разработка параллельного приложения сервера и приложения клиента, которые будут обмениваться информацией по ходу игры. Приложение сервера использует многопоточность для подключения игроков и синхронизации их действий, а остальные действия выполняются в одном главном потоке. Приложение клиента работает только на главном потоке. Игра должна работать под управлением операционной системы семейства Windows 10. Игра будет иметь графический интерфейс, нарисованный символами ascii. В приложении будет реализовано простое подключение игрока к серверу, анимированная заставка, игровой интерфейс. Приложение сервера будет лишь спрашивать кол-во игроков, участвующих в игре, и выводить строчки информации, которые отправляет клиентам.

## **2. Выбор инструментальных средств.**

Программировать будем в Visual studio 2022 на языке C++.

## **3. Выбор способа взаимодействия потоков.**

Я выбрал мьютексы, так как их поддерживает C++.

**4. Выбор библиотеки для реализации многопоточной модели.**  
Для реализации многопоточности использовалась стандартная библиотека “thread”.

## **5. Алгоритм решения задачи.**

Правила игры следующие. В одной игре может принимать участие от двух до шести игроков и человек, на компьютере которого запущен сервер(конечно, приложение сервера может быть запущенно на компьютере какого-нибудь игрока). Вначале запускается приложение сервера. При запуске оно спрашивает ip, на котором будет работать. Для этого запускаем командную строку(cmd), вводим команду ipconfig, находим "Адаптер Ethernet Ethernet", а в нём поле "IPv4-адрес". Это и есть наш ip, который нужно ввести. Можно конешно ввести символ '!' и нажать 'enter'. Тогда сервер будет использовать ip поумолчанию(но очень сомневаюсь, что он будет схожим с вашим, т.к. в таком случае сервер не заработает). После ip нас попросят ввести кол-во игроков. Вводим число игроков и сервер выдаёт нам сообщение "i am ready", что означает, что можно подключать игроков. При запуске приложения клиента, он попросит ввести ник, а после ip(ip должен быть такой же, как и тот, что мы ввели серверу). Если всё пройдёт успешно, то сервер выдаст сообщение "<ник> connected"(где ник это ник игрока), а у игрока начнётся анимированная заставка, по истечению которой выведется либо "Not enough players to start the game, please wait till more players will join", либо "Are you ready to begin?"(посмотрев код, вы можете подумать, что может вывесться ещё и сообщение "The game is going, please wait till it will over". Скажем так, ранее я планировал сделать так, что люди могут попробовать подключиться прямо во время игры и выведется это сообщение, но я отказался от этой идеи). Как только к серверу подключиться нужное кол-во игроков и они нажмут на любую кнопку после сообщения "Are you ready to begin?", начнётся игра. По началу самой игры у игроков прорисовывается интерфейс. Справа будет написанно кол-во денег, выйграшей и пройгрышей, максимальная ставка на данный момент(по-умолчанию 10$) и 2 карты. Посередине нарисован стол, слева от него крупъе, над ним число раундов, над столом распологаются игроки(их число зависит от кол-ва этих самых игроков). над игроками написан порядок их хода, ник, число побед и поражений, деньги. Слева сверху написанно то, каким ходит сам игрок и выводятся информационные сообщения(по типу: "<ник> made a call"). По мере хода игры, сервер будет выводить команды, которые отправляются игрокам(для упрощения отладки и чтоб держатель сервера видел, что там происходит). По правилам покера первые два игрока делают малый и большой блайд(короче первый игрок ставит 5$, а второй 10$). Потом каждый игрок выбирает, что он хочет сделать в свой ход: fold(скинуть карты и отказаться от игры, но можно посмотреть её до конца), check(согласиться с нынешней ставкой(типа если максимальная ставка 10$, а игрок поставил 5$, то он ставит ещё 5$)), bet(увеличить максимальную ставку). Так продолжается 4 раунда. За это время на стол выкладываются карты(3 карты после 1-го раунда, 1 карта после 2-го и 3-го). В конце приложение клиента ищет комбинации карт для своего игрока составляя его кол-во очков. Комбинации карт взял с этого сайта: https://academypoker.ru/combination.html. Все очки игроков отправляются на сервер и ищется их самое большое кол-во. У кого больше очков, тот и выйгрывает весь банк. Если у 2-х игроков одинаковое число очков, то банк делится между ними поровну. Клиент и сервер выводят ник(-и) победителя(-ей). После определения победителя клиент ставиться на паузу, а при нажатии любой кнопки, приложение клиента закрывается. Сервер же выводит "The end.", и после этого программу можно закрыть вручную. Сама она не закроется, так как(по-моему мнению) ждёт завершения работы потоков. Да, этот нюанс я не доделал. Вообще у меня ещё в планах было сделать сохранение информации о клиентах в отдельный файл, чтоб при запуске приложения клиента, пользователь вводил свой ник и пароль и ему давались данные о его деньгах, победах и пройгрышах. Сейчас у каждого пользователя по 200$ и по 0 побед и пройгрышей(Ну, будем считать, что это открытая бэта, а не я полинился доделать до конца).

## **6. Вывод.**

Таким образом, для реализации сетевой игры “Покер” основной ОС выбрана Windows 10 и языком программирования C++. Для синхронизации потоков были выбраны мьютексы. Графический интерфейс строится из символов ascii. В качестве библиотеки, реализующей работу потоков был выбран thread.