Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования "Новгородский государственный университет имени Ярослава Мудрого"

Кафедра «Информационных технологий и систем»

Дисциплина «Операционные системы»

Курсовая работа

«Морской бой»

Выполнил студент группы 9091

\_\_\_\_\_\_\_\_\_\_\_\_\_/Федотов Дмитрий Борисович/

Подпись ФИО

Принял преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_/Ананьев Владислав Валерьевич/

Подпись ФИО

Великий Новгород

2021

Оглавление

Разработка проекта

Введение

Целью данной работы является создания клона некогда популярной игры «Морской бой». Морской бой — это игра для двух участников, в которой игроки по очереди называют координаты на неизвестной им карте соперника. Если у соперника по этим координатам имеется корабль (координаты заняты кораблем), то корабль или его часть «топится», а попавший получает право сделать ещё один ход. Цель игрока — первым потопить все корабли противника.

Особенность проектируемой игры заключается в том, что она должна работать под операционной системой Linux.

Основные правила и суть задачи. Количества игроков в данной игре — 2 игрока, причем должно быть 3 пары: игрок против игрока, игрок против компьютера, компьютер после компьютера.

В начале игры задается расположение кораблей каждого игрока – чтением координат кораблей из файла или случайным образом. В процессе расстановки кораблей программа должна проверять то, что их количество верное: 1 из 4 клеток (Линейный корабль), 2 из 3 клеток (Крейсера), 3 из 2х (Эсминцы), 4 корабля по 1 клетке (Торпедные катер). Корабли не могут соприкасаться краями и углами. Корабли не могут быть изогнуты или установлены по диагонали, т. е. все клетки корабля располагаются либо горизонтально, либо вертикально.

В начале каждого хода экран очищается и выводится 2 поля 10x10 клеток. На поле игрока, который сейчас должен ходить, отмечены пораженные клетки и корабли. На втором поле отмечены только пораженные клетки. Если в пораженной клетке был корабль, то это должно быть также указано (цветом или особым символом).

Ход игрока состоит в указании координат клетки поля соперника, которая далее будет считаться пораженной. В случае нахождении корабля в пораженной клетке, ход остается у того же игрока, в противном случае ход переходит противнику. За обработку хода каждого игрока отвечает отдельный поток.

Обязательное дополнение для реализации параллельных процессов: компьютерный игрок в процессе ожидания завершения хода другого игрока (неважно, компьютерного или нет) должен составить для себя список последовательности клеток, по которому он собирается поражать клетки другого игрока. Таким образом, к началу своего хода, компьютер уже должен быть готов ходить без дополнительных размышлений. В случае обнаружения корабля противника компьютерный игрок должен начать построение списка с нуля, с учетом выясненной информации. Но построение нового списка должно начаться только когда ход перейдет к противнику, текущий ход будет производиться без использования списка, непосредственным выбором координат стрельбы.

Перечень основных функций, предъявленных к данной игре:

Выбор соперника должен осуществляться через параметры командной строки;

Постановка задачи

Задачей данной курсовой работы является разработка игры «Морской бой», в которой должна быть возможность выбора того, кем будет управлять каждым игроком (Игрок — Игрок, Игрок — Компьютер, Компьютер — Компьютер). Игра должны работать под управлением операционной системы семейства Unix/Linux. Игра должна быть выполнена на языке Си, а сетевая часть с помощью сокетов, представленных системной библиотекой <sys/socket.h>

Выбор инструментальных средств

Сетевая часть программы будет реализована с помощью библиотек сокетов Беркли на Си. В нашем случае мы разрабатываем игру, поэтому нам необходимо, чтобы каждый пакет точно дошел до своего получателя в целости и сохранности, поэтому был использован протокол передачи данных семейства TCP/IP

Графический интерфейс не используется, вместо него используется консольный интерфейс с применением библиотеки <ncurses.h>, предназначенной для управления ввода-выводом на терминал, в числе прочего, библиотека позволяет задавать экранные координаты и цвет выводимых символов. Помимо этого, <ncurses.h> предоставляет нам разные уровни абстракции, позволяющие не беспокоиться об аппаратных различиях терминалов и писать переносимый код.

Так как благодаря <ncurses.h> нам не нужно заботиться об переносимости, мы можем использовать любой дистрибутив линукса и не думать о том, что наша программа где-то не так заработает.

Выбор модели

В сетевой игре могут принимать участие только два игрока. Для этого была выбрана модель взаимодействия компьютеров и программ в сети клиент-сервер. Первая часть игры заключается в расположении игроками кораблей, причем корабли игрок может расставлять сам, задавая координаты вручную, либо игрок может случайно расставить корабли или считать их расположение с текстового файла. Сервер должен принимать координаты от клиентов, выполнять проверку координат, возможность размещение данного корабля на поле. В основной части игры сервер должен принимать координаты, проверять их правильность, координировать ходы клиентов (кто ходит, а кто ждет), отсылать состояния клиентов и результаты ходов.

Выбор протокола транспортной OSI-модели

Перед тем как реализовывать сетевую часть игру, необходимо определиться, а какой протокол нужно использовать UDP или TCP/IP?

Начнем с TCP. Большим плюсом данного протокола — это его высокая надёжность, поскольку данный протокола позволяет не терять данные при передаче, запрашивая подтверждения о получении от принимающей стороны и в случае необходимости отправлять данные повторно. Но у этого протокола есть небольшой минус — это относительная низкая скорость передачи данных, за счет того, что выполнение надежной и упорядоченной передачи занимает больше времени.

Для протокола TCP есть альтернатива — протокол UDP. Данный уже протокол является противоположность по преимуществам и недостаткам TCP. Если TCP надежный, то UDP уже нет. TCP медленный, а UDP быстрый.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Протокол TCP | | Протокол UDP | | |
| Преимущества | Недостатки | | Преимущества | Недостатки |
| Надежность. Подтверждения получения данных, повторной отправки в случае необходимости | Скорость. | | Скорость. | Надежность. При получении отправленные данные могут приходить не полностью; |
| Упорядоченность. Гарантируется передача данных в том порядке, в котором они были отправлены | - | | - | Упорядоченность. Порядок передачи данных не соблюдается |
| Метод передачи данных — потоковая. Границы фрагментов данных не имеют обозначения. | - | | - | Метод передачи данных — датаграмма. Проверка пакетов на целостность осуществляется принимающей стороной только в случае получения сообщения |

Для своей курсовой работы я решил использовать TCP-протокол, потому что мне необходимо, чтобы все пакеты всегда доходили от сервера до клиентов и от клиентов до сервера. Если до одного из двух игроков не дойдет сообщение, то между одним из полей игрока и сервера будет иметься разница, которая в нашем случае не допустима.

Реализация проекта

Файлы проекта

Реализация работы сервера

Реализация работы клиента

Использованные системные вызовы

Пример работы клиента

Результат работы

В ходе выполнения данной курсовой работы удалось разработать игру «Морской бой» с разными игровыми режимами. Консольный интерфейс удалось реализовать с помощью библиотеки <ncurses.h>. Управлять игроком может как компьютер игрок, так и живой игрок. Причем за обработку каждого игрока отвечает отдельный поток.

Заключение

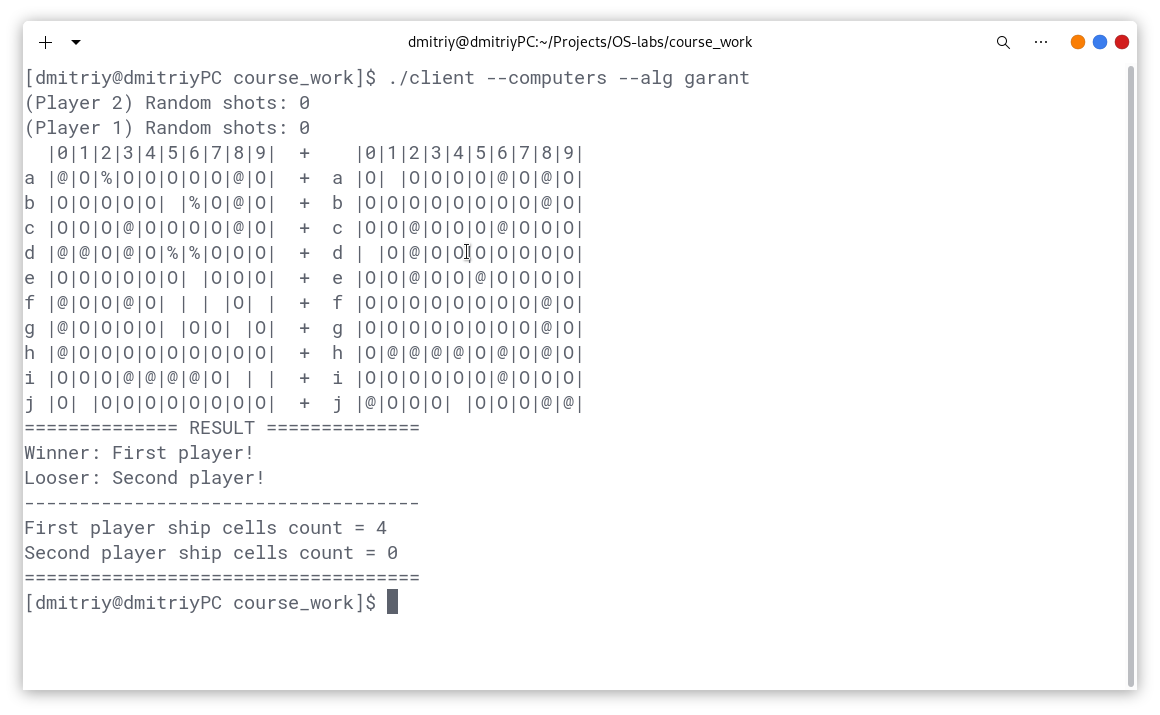
Таким образом, результат курсовой работы полностью соответствует требованиям к реализации, все поставленные задачи были выполены. Программа работает на операционных системах семейста Linux и написана на языке Си (компилятор gcc). Серверная часть программы построена с помощью протокола TCP. Созданная программа логически завершена.

Список использованной литературы

Карпов В. Е, Коньков К. А. «Основы операционных систем. Курс лекций» - Учебное пособие, 2005 — 536 с.

Приложение А. 

Аргументы командной строки



Результат режима “Компьютер против Компьютера”

Приложение Б.

Листинг client.c

Листинг server.c

Листринг define.h

Листринг computer.h

Листинг computer.c

Листинг field.h

Листинг field.c

Листринг uiility.h

Листинг utility.c

Листринг renderer.h

Листринг renderer.c