

Operating Systems. IHW 4

[Operating Systems](#)

Александр Васюков | БПИ235

Задание

Изучить работу с транспортным протоколом UDP. Научиться разбивать задачу на части, для последующего их выполнения серверами и клиентами. Архитектура «клиент–сервер» широко используется при решении разнообразных прикладных задач. Существуют различные подходы к организации таких приложений с использованием для организации серверов и клиентов как процессов, так и потоков. В ходе выполнения задания необходимо осуществить разделить функции, выполняемые отдельными компонентами и организовать их взаимодействие, необходимое для выполнения заданной прикладной задачи.

Вариант 35

«Камень, ножницы, бумага» 1 – однокруговая система. N студентов, изнывающих от скуки на лекции по операционным системам решили организовать однокруговой турнир в игру «Камень, ножницы, бумага». Каждый с каждым при этом играет только один раз. За победу дается два очка, за ничью – одно, за поражение – ноль. Требуется создать клиент–серверное приложение, моделирующее турнир. Каждый студент – отдельный клиент. Генерация камня, ножниц и бумаги в каждом поединке формируется сервером случайно. Завершение работы программ осуществляется по завершению турнира. Количество участвующих студентов задается сервером.

Решение претендует на 10 баллов.

Программы можно найти на Github по ссылке: https://github.com/vasyukov1/HSE-FCS-SE-2-year/tree/main/Operating%20Systems/Homeworks/IHW_4

Решение

На 4-5 баллов

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.

Сценарий работы:

1. Запуск сервиса и ожидание клиентов.
2. Регистрация клиентов: Каждый клиент отправляет UDP-пакет "REGISTER" на сервер.

3. Начало турнира: После регистрации всех клиентов сервер генерирует матчи и рассылает результаты.
4. Отображение результатов: Каждый клиент получает информацию только о своих матчах и финальный рейтинг.

Реализация:

- **UDP-сокеты:** Используются для обмена данными без установления соединения.
- **Регистрация клиентов:** Сервер собирает адреса клиентов из полученных пакетов.
- **Турнирная логика:** Сервер генерирует все парные матчи и рассылает результаты.
- **Завершение работы:** После отправки всех результатов сервер закрывает соединение.

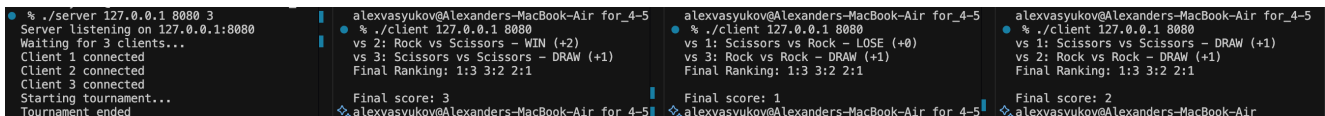
Запуск:

```
cd for_4-5

gcc server.c -o server
gcc client.c -o client

./server <IP> <PORT> <NUM_STUDENTS>
./client <IP> <PORT>
```

Результат:



```
% ./server 127.0.0.1 8080 3
Server listening on 127.0.0.1:8080
Waiting for 3 clients...
Client 1 connected
Client 2 connected
Client 3 connected
Starting tournament...
Tournament ended

alexvasyukov@Alexanders-MacBook-Air for_4-5
% ./client 127.0.0.1 8080
vs 2: Rock vs Scissors - WIN (+2)
vs 3: Scissors vs Scissors - DRAW (+1)
Final Ranking: 1:3 3:2 2:1
Final score: 3

alexvasyukov@Alexanders-MacBook-Air for_4-5
% ./client 127.0.0.1 8080
vs 1: Scissors vs Rock - LOSE (+0)
vs 3: Rock vs Rock - DRAW (+1)
Final Ranking: 1:3 3:2 2:1
Final score: 1

alexvasyukov@Alexanders-MacBook-Air for_4-5
% ./client 127.0.0.1 8080
vs 1: Scissors vs Scissors - DRAW (+1)
vs 2: Rock vs Rock - DRAW (+1)
Final Ranking: 1:3 3:2 2:1
Final score: 2
```

На 6-7 баллов

В дополнение к программе на предыдущую оценку необходимо разработать клиентскую программу, подключаемую к серверу, которая предназначена для отображение комплексной информации о выполнении приложения в целом. То есть, данный программный модуль должен адекватно отображать поведение моделируемой системы, позволяя не пользоваться отдельными видами, предоставляемыми клиентами и серверами по отдельности.

Сценарий работы:

1. **Сервер:**
 - Принимает два типа сообщений: REGISTER (участники) и MONITOR (наблюдатели)
 - Рассылает участникам результаты их матчей
 - Рассылает мониторам полную информацию о каждом матче
 - Отправляет рейтинг всем подключенным клиентам
2. **Клиент-участник:**
 - Отправляет REGISTER при старте

- Получает только свою статистику

3. Клиент-монитор:

- Отправляет MONITOR при старте
- Получает полную информацию о всех матчах
- Видит общий рейтинг участников

Запуск:

```
cd for_6-7

gcc server.c -o server
gcc client.c -o client
gcc monitor.c -o monitor

./server <IP> <PORT> <NUM_STUDENTS>
./monitor <IP> <PORT>
./client <IP> <PORT>
```

Результат:

На 8 баллов

В дополнение к предыдущей программе реализовать возможность, подключения множества клиентов, обеспечивающих отображение информации о работе приложения. Это должно позволить осуществлять наблюдение за поведением программы с многих независимых компьютеров. При этом клиентов—наблюдателей можно отключать и подключать снова в динамическом режиме без нарушения работы всего приложения. Отчет расширить информацией о добавленной реализации и привести соответствующие результаты работы программы.

Изменения:

1. Поддержка множества мониторов:

- Сервер регистрирует мониторы через сообщение MONITOR .
- Максимальное количество мониторов ограничено MAX_MONITORS (10).
- Добавлена проверка num_monitors < MAX_MONITORS при регистрации.

- Могут работать одновременно на нескольких устройствах.

2. Рассылка событий:

- Все ключевые события (матчи, рейтинг, завершение) рассылаются мониторам через `send_to_monitors()`.

3. Корректное завершение:

- Мониторы получают специальное сообщение `END` без данных.

4. Таймауты в мониторе:

- Установлен 300-секундный таймаут на операцию `recvfrom`
- При обнаружении таймаута монитор завершает работу

Запуск:

```
cd for_8

gcc server.c -o server
gcc client.c -o client
gcc monitor.c -o monitor

./server <IP> <PORT> <NUM_STUDENTS>
./monitor <IP> <PORT>
./client <IP> <PORT>
```

Результат:

Компьютер 1 с запущенным сервером, монитором и клиентами.

Также демонстрируется корректное отключение монитора.

```
alexvasyukov@Alexanders-MacBook-Air for_8 % ./server 192.168.1.35 8080 3
.168.1.35 8080 3
Server listening on 192.168.1.35:8080
Waiting for 3 students and monitors...
Monitor 1 connected
Monitor 2 connected
Monitor 3 connected
Student 1 registered
Student 2 registered
Student 3 registered
Starting tournament...
Tournament ended. Total monitors: 3
alexvasyukov@Alexanders-MacBook-Air for_8 %

alexvasyukov@Alexanders-MacBook-Air for_8 % ./monitor 192.168.1.35 8080
.168.1.35 8080
Monitor connected. Waiting for data...
Match: 1 (Rock) vs 2 (Paper) -> WIN2
Match: 1 (Rock) vs 3 (Scissors) -> WIN1
Match: 2 (Paper) vs 3 (Rock) -> WIN1
Current Ranking: 2:4 1:2 3:0
Tournament ended. Exiting...
alexvasyukov@Alexanders-MacBook-Air for_8 %

alexvasyukov@Alexanders-MacBook-Air for_8 % ./client 192.168.1.35 8080
.168.1.35 8080
vs 2: Rock vs Paper - LOSE (+0)
vs 3: Rock vs Scissors - WIN (+2)
Final Ranking: 2:4 1:2 3:0
Final score: 2
alexvasyukov@Alexanders-MacBook-Air for_8 %

alexvasyukov@Alexanders-MacBook-Air for_8 % ./client 192.168.1.35 8080
.168.1.35 8080
vs 1: Paper vs Rock - WIN (+2)
vs 3: Paper vs Rock - WIN (+2)
Final Ranking: 2:4 1:2 3:0
Final score: 4
alexvasyukov@Alexanders-MacBook-Air for_8 %

alexvasyukov@Alexanders-MacBook-Air for_8 % ./client 192.168.1.35 8080
.168.1.35 8080
vs 1: Scissors vs Rock - LOSE (+0)
vs 2: Rock vs Paper - LOSE (+0)
Final Ranking: 2:4 1:2 3:0
Final score: 0
alexvasyukov@Alexanders-MacBook-Air for_8 %
```

Компьютер 2 с монитором.

```
● andrey@MacBook-Air--Andrey for_8 % ./monitor 192.168.1.35 8080
Monitor connected. Waiting for data...
Match: 1 (Rock) vs 2 (Paper) -> WIN2
Match: 1 (Rock) vs 3 (Scissors) -> WIN1
Match: 2 (Paper) vs 3 (Rock) -> WIN1
Current Ranking: 2:4 1:2 3:0
Tournament ended. Exiting...
○ andrey@MacBook-Air--Andrey for_8 % █
```

На 9-10 баллов

Изменения:

1. Динамическое управление подключениями:

- Реализован механизм проверки активности клиентов через периодическую отправку PING.
- Неактивные клиенты автоматически удаляются из списка.
- Добавлена обработка сетевых ошибок (ECONNREFUSED, EHOSTUNREACH, ENETUNREACH).
- Турнир запускается только при наличии минимум 2 активных клиентов

2. Командный интерфейс сервера:

- Добавлены команды:
 - `start` - запуск турнира с текущими активными клиентами
 - `status` - отображение количества активных клиентов и мониторов
 - `exit` - корректное завершение работы сервера
- Турнир можно запускать многократно без перезапуска сервера. Но необходимо подождать отключения прошлых клиентов-игроков.

3. Корректное завершение работы:

- При получении команды `exit` сервер рассылает сообщение SHUTDOWN всем активным клиентам и мониторам.
- Клиенты и мониторы при получении SHUTDOWN корректно завершают работу.
- Добавлена задержка перед завершением сервера для гарантированной доставки сообщений.

4. Обработка ошибок:

- Все операции отправки проверяются на ошибки.
- При обнаружении проблем с доставкой клиент/монитор помечается как неактивный.
- Реализован механизм автоматического удаления неактивных клиентов.

Запуск:

```
cd for_9-10
```

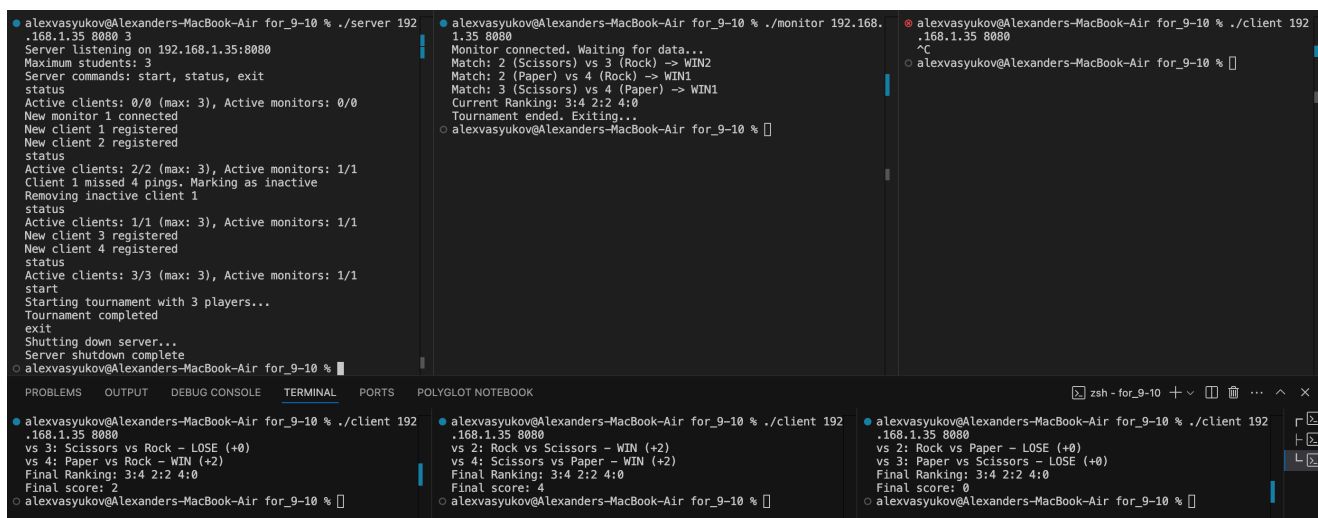
```
gcc server.c -o server
```

```
gcc client.c -o client
gcc monitor.c -o monitor

./server <IP> <PORT> <MAX_NUM_STUDENTS>
./monitor <IP> <PORT>
./client <IP> <PORT>
```

Результат:

1. Запуск сервера
2. status - вывод информации
3. Регистрация монитора и 2 клиентов
4. status - вывод информации
5. Отключение клиента
6. status - вывод информации
7. Регистрация 2 клиентов
8. status - запуск турнира
9. exit - завершение



```
alexvasyukov@Alexanders-MacBook-Air for_9-10 % ./server 192
.168.1.35 8080 3
Server listening on 192.168.1.35:8080
Maximum students: 3
Server commands: start, status, exit
status
Active clients: 0/0 (max: 3), Active monitors: 0/0
New monitor 1 connected
New client 1 registered
New client 2 registered
status
Active clients: 2/2 (max: 3), Active monitors: 1/1
Client 1 missed 4 pings. Marking as inactive
Removing inactive client 1
status
Active clients: 1/1 (max: 3), Active monitors: 1/1
New client 3 registered
New client 4 registered
status
Active clients: 3/3 (max: 3), Active monitors: 1/1
start
Starting tournament with 3 players...
Tournament completed
exit
Shutting down server...
Server shutdown complete
alexvasyukov@Alexanders-MacBook-Air for_9-10 %

alexvasyukov@Alexanders-MacBook-Air for_9-10 % ./monitor 192.168.
1.35 8080
Monitor connected. Waiting for data...
Match: 2 (Scissors) vs 3 (Rock) -> WIN2
Match: 2 (Paper) vs 4 (Rock) -> WIN1
Match: 3 (Scissors) vs 4 (Paper) -> WIN1
Current Ranking: 3:4 2:2 4:0
Tournament ended. Exiting...
alexvasyukov@Alexanders-MacBook-Air for_9-10 %

alexvasyukov@Alexanders-MacBook-Air for_9-10 % ./client 192
.168.1.35 8080
alexvasyukov@Alexanders-MacBook-Air for_9-10 %

alexvasyukov@Alexanders-MacBook-Air for_9-10 % ./client 192
.168.1.35 8080
vs 3: Scissors vs Rock - LOSE (+0)
vs 4: Paper vs Rock - WIN (+2)
Final Ranking: 3:4 2:2 4:0
Final score: 2
alexvasyukov@Alexanders-MacBook-Air for_9-10 %

alexvasyukov@Alexanders-MacBook-Air for_9-10 % ./client 192
.168.1.35 8080
vs 2: Rock vs Scissors - WIN (+2)
vs 4: Scissors vs Paper - WIN (+2)
Final Ranking: 3:4 2:2 4:0
Final score: 4
alexvasyukov@Alexanders-MacBook-Air for_9-10 %

alexvasyukov@Alexanders-MacBook-Air for_9-10 % ./client 192
.168.1.35 8080
vs 2: Rock vs Paper - LOSE (+0)
vs 3: Paper vs Scissors - LOSE (+0)
Final Ranking: 3:4 2:2 4:0
Final score: 0
alexvasyukov@Alexanders-MacBook-Air for_9-10 %
```