

Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Сети и телекоммуникации

Отчет по лабораторной работе
Программирование сокетов протоколов TCP

Работу
выполнил:
Волкова М.Д.
Группа: 43501/3
Преподаватель:
Алексюк А.О.

Санкт-Петербург
2018

1. Цель работы

Изучение принципов программирования сокетов протоколов TCP.

2. Программа работы

- разработать простейший клиент и сервер на основе протоколов TCP
- разработать прикладной протокол в соответствии с индивидуальным заданием, реализовать протокол в виде клиент-серверного приложения на основе протоколов TCP

3. Ход выполнения работы

3.1. Простейшие клиент и сервер

Простейшие клиент и сервер были выполнены на основе протоколов TCP, а также адаптированы под ОС Windows и Linux. Сервер выполняет функции эхо-сервера, т.е. принимает сообщения от клиентов и посылает копии обратно. Клиент посылает сообщение, после чего завершается.

3.2. Индивидуальное задание

Разработать приложение клиент и приложение сервер системы поиска/публикации новостей. Новости сгруппированы по темам. Каждая новость имеет уникальный идентификатор, название и текст новости.

Серверное приложение реализует следующие функции:

- Прослушивание определенного порта
- Обработка запросов по подключению
- Поддержка одновременной работы нескольких клиентов системы
- Прием запросов от клиента на передачу списка тем, списка новостей по теме, текста новости, добавление новости по теме
- Осуществление добавления тем, новостей по темам
- Передача списков тем, списков новостей и текстов новостей
- Обработка запроса на отключение клиента
- Принудительное отключение клиента

Клиентское приложение реализует следующие функции:

- Установление соединения с сервером
- Передача запросов на передачу списка тем, списка новостей по теме, текста новости, добавление новости по теме
- Получение результатов от сервера
- Разрыв соединения
- Обработка ситуации отключения клиента сервером

3.2.1. Описание протокола

Сообщение клиента всегда содержит команду, определяющее тип сообщения. Некоторые типы сообщений содержат также поле опций.

Поле команды имеет размер до 5 байт. Поле опций может (в зависимости от команды) отсутствовать. Опции и команды разделяются пробелом.

Сообщение клиента содержит команду, определяющее тип сообщения.

- Команда для получения списка тем - 1
- Команда для добавления новости - 2
- Команда для выхода клиента - 3

В любом другом случае, появится сообщение об ошибке.

Сообщение сервера содержит команду, определяющее тип сообщения.

- Команда отключения клиента - kill. Опции - номер клиента.
- Просмотр подключенных клиентов - online
- Закрытие системы - shutdown

3.2.2. Описание структуры приложения

Сервер:

- Создание сокета
- Создание потока для прослушивания сокета
- Читеник команд от сервера
- Читеник команд от клиента
- Получение сообщения от клиентов
- Анализ сообщения
- Ответ на сообщение

Клиент:

- Чтение IP адреса сервера
- Подключение к серверу
- Создание потока
- Чтение данных от сервера и реакция
- Отправка команд серверу
- Поток для получения данных от сервера

4. Тестирование

```
-----  
1. VIEW  
2. ADD  
3. EXIT  
-----  
  
1  
  
-----  
TOPICS:  
1. war  
2. sport  
3. cars  
4. technologies  
5. new  
6. cats  
7. waf  
-----
```

Рисунок 4.1. Просмотр списка тем

```
-----  
TOPICS:  
1. war  
2. sport  
3. cars  
4. technologies  
5. new  
6. cats  
7. waf  
-----  
  
war  
  
CAPTION:  
1. first  
2. second  
3. testwoeij  
4. wep
```

Рисунок 4.2. Просмотр название новостей по теме war

```
CAPTION:  
1. first  
2. second  
3. testwoeij  
4. wep  
  
2  
  
TEXT:  
some texttext test
```

Рисунок 4.3. Просмотр текста второй новости

```

-----
1. VIEW
2. ADD
3. EXIT
-----

2

ENTER TOPIC:
top

ENTER CAPTION:
cap

ENTER TEXT:
text

```

Рисунок 4.4. Добавление новой новости

```

-----
TOPICS:
1. war
2. sport
3. cars
4. technologies
5. new
6. cats
7. waf
8. top
-----

top

CAPTION:
1. cap

1

TEXT:
text

```

Рисунок 4.5. Просмотр добавленной новости на втором клиенте

```

-----
1. VIEW
2. ADD
3. EXIT
-----

3
closing connection
stakenschneider:client Maria$ █

```

Рисунок 4.6. Выход клиента

```
-----  
TOPICS:  
1. war  
2. sport  
3. cars  
4. technologies  
5. new  
6. cats  
7. waf  
8. top  
-----  
  
dogs  
  
!ERROR! NOT EXIST !ERROR!
```

Рисунок 4.7. Пример вывода ошибки

5. Выводы

В ходе работы были изучены принципы программирования сокетов с использованием протокола TCP.

Во время выполнения индивидуального задания была реализована клиент-серверная программа системы поиска/публикации новостей, с написанием собственного протокола на основе TCP. В результате этого были изучены принципы программирования сокетов TCP. Основной проблемой при реализации приложения на TCP была необходимость контроля длины послыки. Ее решением стало добавление символа окончания послыки. TCP требует установления соединения, поэтому на сервере выделяется поток, в котором происходит прием запросов на соединение от клиентов через выделенный для этого сокет. После подключения очередного клиента порождается отдельный поток, осуществляющий обмен пакетами с этим клиентом через отдельный сокет.