

Специальность 09.02.04 Информационные системы

Преподаватель: Тимофеева С.Н.
Контактные данные преподавателя:
e-mail: timsnikol@mail.ru

**Задания для самостоятельной работы
по дисциплине «Компьютерные сети»**

Дата 1.06.2020
Группа ИС-31.

Задание 3. ПРАКТИЧЕСКАЯ РАБОТА № 19

ПРАКТИЧЕСКАЯ РАБОТА № 19

Тема: Разработка протокола прикладного уровня

Формируемые компетенции

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ПК 1.7 Производить инсталляцию и настройку информационной системы и разрабатывать фрагменты обучения пользователей информационной системы

Цель работы: изучение технологии работы протокола прикладного уровня путем имитации HTTP клиента.

Оборудование: компьютер, локальная сеть

Теория и основные характеристики

TCP должен взаимодействовать не только с протоколами нижележащего уровня, но и с протоколами и приложениями прикладного уровня. Связь с прикладным уровнем осуществляется с помощью набора сервисных примитивов. Сервисные примитивы определены в стандарте протокола, а для прикладных программ они доступны в форме библиотек работы с сокетами.

При установлении соединения каждая из сторон выполняет некоторые операции, называемые открытием соединения. Открытие может быть пассивным или активным. Как правило, одна из сторон производит активное открытие соединения, а другая - пассивное, тогда соединение устанавливается. Оба режима подчиняются четким правилам. Пассивное соединение еще иногда называют серверным, а активное - клиентским.

Порядок выполнения

1. Опишите в отчете: Разработка протокола прикладного уровня
2. Выполните **составление схемы**
3. Составьте сценарий разработки протокола прикладного уровня с помощью команд и кодов.

Задание:

Разработать протокол прикладного уровня Internet (Application level), в соответствии с которым пользователь с помощью клиентской программы устанавливает соединение с сервером и после предварительной регистрации получает от сервера по запросу следующие услуги:

- получение адресов сервера и клиента, участвующих в данном сеансе;
- получение текущей даты и времени;
- передача текстовых файлов с клиентского компьютера на сервер и размещение их в файловой системе сервера; при этом в зависимости от заданных опций все числа, записанные в файлах, переводятся из арабской записи в римскую, из римской в арабскую или остаются без изменения - получение текстовых файлов с сервера по запросу клиента.

Разработка протокола должна быть оформлена письменно в форме отчета, и включать в себя следующие части:

- описание логической характеристики протокола (типы и форматы команд, семантика команд);
- описание процедурной части протокола в виде автоматной модели (указать вероятность для каждого перехода из состояния в состояние);
- описание последовательности выполнения операций установления соединения, передачи и обработки данных, разъединения.

Образец выполнения и оформления

1. Описание логической характеристики протокола.

2. В соответствии с разработанным протоколом пользователь с помощью клиентской программы устанавливает соединение с сервером и получает от сервера по запросу некоторые услуги.

Разработанный протокол осуществляет следующие функции: получение адресов клиента и сервера, получение текущих даты и времени, передача и прием текстовых файлов (с возможным переводом чисел в них из арабской записи в римскую и наоборот).

В протоколе, со стороны клиента, используются следующие команды:

connect <ip-адрес сервера> - подключение к серверу с указанным в качестве параметра ip-адресом.

ip - получение адресов сервера и клиента, участвующих в данном сеансе.

date - получение текущей даты и времени.

send <file_name> - передача текстового файла(file_name) на сервер.

sendAR <file_name> - передача текстового файла(file_name) на сервер, с последующим переводом из арабской записи в римскую.

sendRA <file_name> - передача текстового файла(file_name) на сервер, с последующим переводом из римской записи в арабскую.

recv <file_name> - получение текстового файла(file_name) от сервера.

help — помощь.

detach - отключение, закрытие соединения.

exit — завершение работы клиентской программы с закрытием соединения.

При передаче по сети все команды транслируются во внутреннее представление, т.е. текстовые обозначения команд заменяются числовыми:

8 - подключение к серверу.

1 - получение адресов сервера и клиента, участвующих в данном сеансе.

2 - получение текущей даты и времени.

3 <file_name> - передача текстового файла(file_name) на сервер.

4 <file_name> - передача текстового файла(file_name) на сервер, с последующим переводом из арабской записи в римскую.

5 <file_name> - передача текстового файла(file_name) на сервер, с последующим переводом из римской записи в арабскую.

6 <file_name> - получение текстового файла(file_name) от сервера.

7 - помощь.

9 - отключение, закрытие соединения.

10 — завершение работы клиентской программы с закрытием соединения.

3. Описание процедурной части протокола в виде автоматной модели

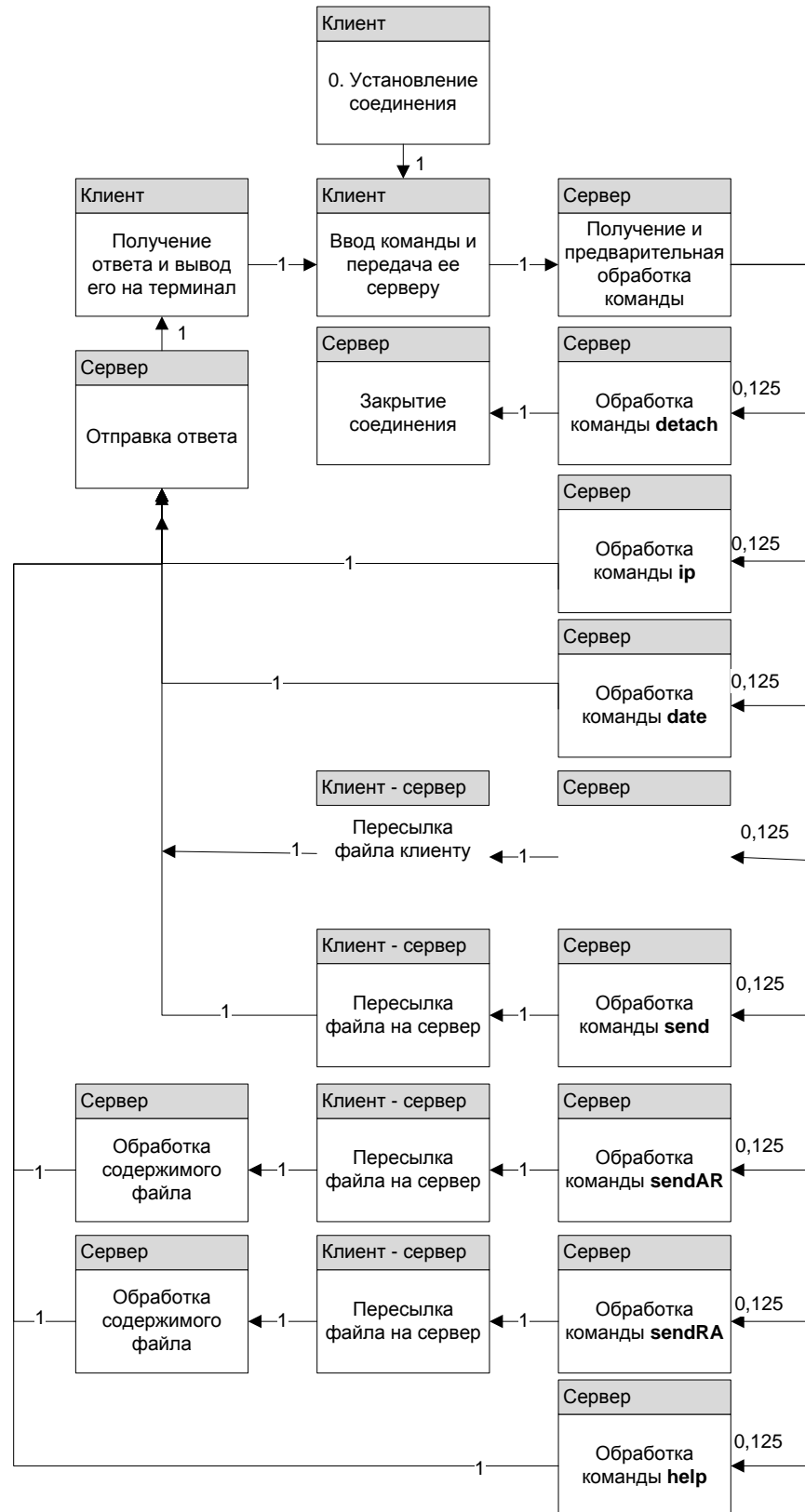


Рисунок 1-Автоматная модель протокола.

4. Описание последовательности выполнения операций установления соединения, передачи и обработки данных, разъединения

При запуске клиентской программы сначала она запрашивает с консоли адрес сервера (имя хоста или IP-адрес), предоставляющего услуги согласно разработанному протоколу. Затем происходит установка соединения клиента с сервером. После этого сервер передает клиенту строку с начальным приветствием, и начинается цикл запросов команд для получения услуг сервера.

На клиентской стороне происходит получение команды с консоли и его отправка в виде текстовой строки. Если введена команда `exit`, то происходит завершение работы клиентской программы. Если введены команды `send`, `sendAR` или `sendRA`, то клиент подготавливает файл и отправляет его вместе с командой (№команды+имя файла+данные). После этого клиент получает ответ от сервера в виде текстовой строки и выводит его на экран. Но если была введена команда `гесv`, то сервер в качестве ответа пришлет данные файла, или индикатор ошибки. В таком случае клиент запускает процедуру приема файла.

Сервер после получения команды от клиента, проводит ее интерпретацию и начинает обработку в зависимости от вида команды. Команда `detach` - закрывает соединение с клиентом, и пользователю передаются прощальные слова. В случае команд `ip` и `date` формируется соответствующий ответ и отправляется клиенту. Если сервер увидел команду `send`, то он начинает ее обработку, осуществляя процедуру приема файла. Для команд `sendAR`, и `sendRA`, помимо приема сервером организуется модификация файла (перевод чисел в файле из арабской записи в римскую или обратно). В случае успешного приема файла сервер отправляет клиенту слова благодарности за высланный файл. Если была получена команда `гесv`, и нужный файл имеется у сервера, то в качестве ответа он отправляет данные этого файла, или сообщение «`error`».

В случае получения неизвестной команды сервер отправляет клиенту сообщение об этом.

Ход работы

Создание примера диалога

Для того, чтобы создать пример диалога между клиентом и сервером с использованием разработанного протокола, используйте пустой текстовый файл «`session.txt`». В нем приведите пример рабочей сессии, в рамках которой должны быть произведены следующие действия:

- Загрузка файла на сервер.
- Получение содержимого файла.
- Получение текущей директории.
- Получение списка файлов в директории.

- Удаление файла.

При составлении текста примера используйте следующие специальные обозначения:

- <NL> — новая строка.
- <SERVER > CLIENT> — далее приводится текст ответа сервера.
- <CLIENT > SERVER> — далее приводится текст клиентского запроса.

После каждого специального обозначения начинайте новую строку.

Разработка протокола прикладного уровня модели OSI: спецификация протокола для работы с файлами на удаленном компьютере

Сообщения протокола

Как запросы клиента так и ответы сервера имеют следующий формат:

```
{заголовок сообщения}
{тело сообщения}
{пустая строка}
```

причем, элемент ? не является обязательным.

Тело сообщения, если оно есть, кодируется следующим образом: ?

Сообщения клиента

В сообщениях клиента заголовок имеет формат: ?

Ниже приводится список кодов действия для заголовка клиентского сообщения:

Код	Описание
?	?

...

Сообщения сервера

В сообщениях сервера заголовок имеет формат: ?

Ниже приводится список кодов ответа для заголовка сообщения сервера:

Код	Описание
?	?

...

Сценарий работы

После установки соединения с сервером клиент ?. После ответа сервера, ?.

После того, как клиент установил соединение с сервером, либо после того, как был отправлен ответ клиенту, сервер ожидает в течение ? запроса от клиента. Если в течение этого времени клиент не отправил запрос, ?.

Со стороны клиента соединение разрывается путем ?.

Студент ??? группы ???

Вопросы для самоконтроля

- Почему в данном протоколе не требуется предусматривать механизмы для корректной передачи данных по сети?
- Какому из распространенных протоколов семейства TCP функционально соответствует разработанный протокол?
- Какова роль IP для данного протокола?
- Возможно ли реализовать описанный протокол?
- Перечислите прикладные протоколы

<https://ws.studylib.ru/doc/3766225/razrabotka-protokola-prikladnogo-urovnnya-modeli-osi>