

## Лабораторная работа № 2

### Клиент-серверное приложение для передачи файлов с использованием протоколов TCP и UDP

#### 2.1 Цель работы

Изучение принципов работы клиент-серверных приложений на основе протоколов TCP и UDP, реализующих функцию передачи файлов.

#### 2.2 Задание на лабораторную работу

Так же, как и в первой работе, необходимо разработать 2 клиент-серверных приложения на основе протоколов TCP и UDP, для передачи файлов.

#### 2.3 Методические указания

Выполнение второй лабораторной работы по структуре программы мало отличается от первой, но в ней имеются некоторые тонкости.

Во-первых, передача файла, подразумевает передачу сведений о нем получателю, следовательно, передача данных идет двунаправленно, то есть на каждой стороне процессы переходят из состояния передачи в состояние приема, и наоборот.

Во-вторых, передача файлов подразумевает, что данные будут подвержены ошибкам и потерям. Но если для TCP эта проблема решена на уровне протокола, так как TCP является протоколом гарантированной доставки сообщений, то для успешной передачи файла по протоколу UDP, на его основе необходимо разработать пользовательский протокол. В приведенной ниже таблице и схеме рассмотрен пример реализации такого протокола.

**Таблица 2.1**

**Таблица команд**

Запрос	Параметры запроса	Ответ	Параметры ответа
Запрос файла	Имя файла	Код ошибки	1-файл существует 0-файла нет
Запрос размера файла	-	Размер файла	Целое число
Размер пакета	Целое число	Подтверждение	Целое число
Запрос пакета	№ Пакета	Пакет № N	Данные

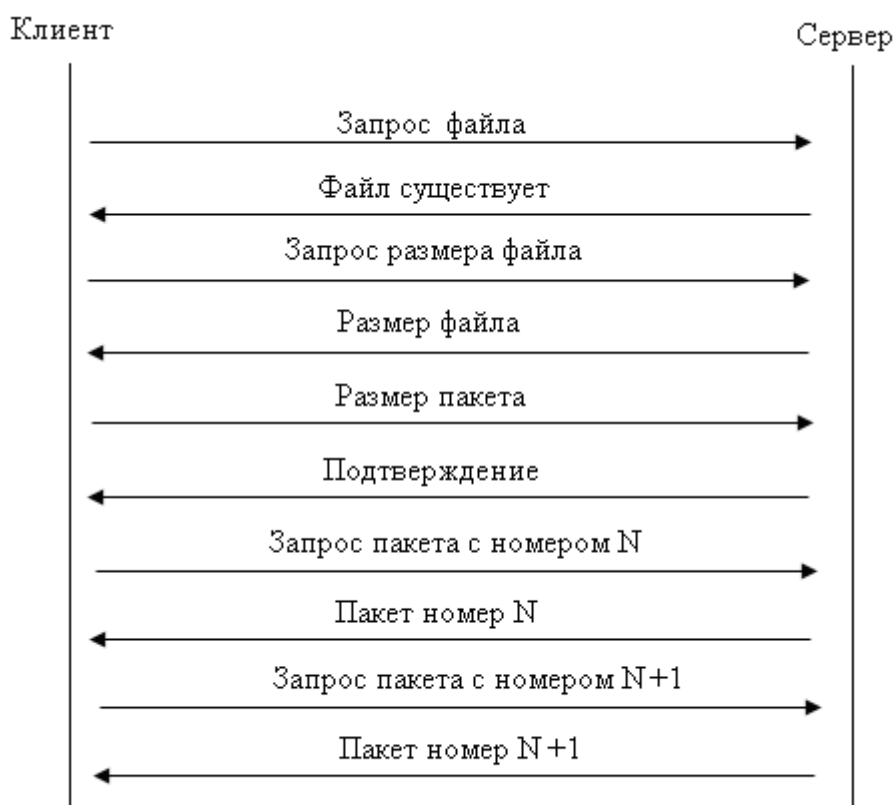


Рисунок 2.1 Схема обмена сообщениями

Клиент посылает запрос, в котором указано имя необходимого файла, сервер в своей директории ищет этот файл и в случае успеха отправляет в ответе подтверждение. Далее клиент запрашивает размер этого файла.

Получить размер файла можно, воспользовавшись функцией:

Таблица 2.2

Описание функции в *nix	Описание функции в Windows
<code>#include &lt;io.h&gt;</code>	<code>#include &lt;windows.h&gt;</code> <code>#include &lt;winbase.h&gt;</code>
<pre>long filesize(char *filename);</pre> <p>Возвращает длину файла в байтах в случае успеха, в противном случае возвращает -1 и устанавливает errno.</p>	<pre>DWORD GetFileSize(     HANDLE hFile, // Указатель     LPDWORD lpFileSizeHigh //     );</pre> <p>В случае возникновения исключения будет возвращено значение типа INVALID_FILE_SIZE.</p>

Размер файла получен, теперь необходимо задать серверу размер блоков, которые будем принимать. Зная размер файла и размер пакета, на клиенте нужно вычислить количество пакетов, это позволит запрашивать блоки с нужным номером.

#### **2.4 Контрольные вопросы и задания**

1. Измерить скорость передачи файла для протокола TCP
2. Измерить скорость передачи файла для протокола UDP
3. Нарисовать диаграмму зависимости скорости передачи от размера пакета для протокола TCP
4. Нарисовать диаграмму зависимости скорости передачи от размера пакета для протокола UDP