Задания для самостоятельной работы к л.р. №1-2

Вариант № 1

1. Описание задачи

Необходимо разработать клиентскую и серверную части приложения для обмена файлами между различными узлами сети с использованием блокирующих сокетов, протокола UDP.

2. Требования к реализации

* Приложение должно выполнять как функции передачи файла, так и функции приема.
* Графический интерфейс
* Ввод IP-адреса принимающего узла
* Ввод номера порта принимающего экземпляра приложения
* Выбор передаваемого файла на диске
* Выбор места сохранения полученного файла
* Отображать прогресс передачи/приема файла
* Считать и отображать время передачи файла.
* Отображать сообщения о возникающих ошибках и корректно их обрабатывать.
* Реализовать механизм надежной доставки данных (приложение должно уметь обнаруживать и корректно обрабатывать не только любое завершение клиентской/серверной частей приложения, но и разрывы физического соединения абонентов).

3. Требования к надежности

К приложению предъявляются следующие требования по надежности:

* Не допускается зависание приложения при любых действиях пользователя.
* Не допускается аварийное завершение приложения при любых действиях пользователя.
* Любая ошибочная ситуация должна корректно обрабатываться с выводом соответствующего сообщения.
* Не допускается утечка памяти/дескрипторов в процессе эксплуатации приложения.
* Не допускается полная загрузка процессора приложением в пассивном состоянии.
* Графический интерфейс не должен зависать во время передачи/приема файлов, а также во время ожидания подключения клиента к серверу (многопоточное приложение).

4. Дополнительные требования

Реализовать параллельную передачу/прием нескольких файлов.

Вариант № 2

1. Описание задачи

Необходимо разработать клиентскую и серверную части приложения для обмена файлами между различными узлами сети с использованием блокирующих сокетов, протокола TCP.

2. Требования к реализации

* Приложение должно выполнять как функции передачи файла, так и функции приема.
* Графический интерфейс
* Ввод IP-адреса принимающего/передающего узла
* Ввод номера порта принимающего/передающего экземпляра приложения
* Выбор передаваемого файла на диске
* Выбор места сохранения полученного файла
* Отображать прогресс передачи/приема файла
* Считать и отображать время передачи файла.
* Отображать сообщения о возникающих ошибках и корректно их обрабатывать.
* Приложение должно уметь обнаруживать и корректно обрабатывать не только любое завершение клиентской/серверной частей приложения, но и разрывы физического соединения абонентов.

3. Требования к надежности

К приложению предъявляются следующие требования по надежности:

* Не допускается зависание приложения при любых действиях пользователя.
* Не допускается аварийное завершение приложения при любых действиях пользователя.
* Любая ошибочная ситуация должна корректно обрабатываться с выводом соответствующего сообщения.
* Не допускается утечка памяти/дескрипторов в процессе эксплуатации приложения.
* Не допускается полная загрузка процессора приложением в пассивном состоянии.
* Графический интерфейс не должен зависать во время передачи/приема файлов, а также во время ожидания подключения клиента к серверу (многопоточное приложение).

4. Дополнительные требования

Реализовать параллельную передачу/прием нескольких файлов.

Вариант № 3

1. Описание задачи

Необходимо разработать приложение-чат для обмена сообщениями между несколькими клиентами через сервер с использованием блокирующих сокетов, протокола UDP.

2. Требования к реализации

* Графический интерфейс
* Ввод IP-адреса принимающего узла
* Ввод номера порта принимающего экземпляра приложения
* Отображать время приема и передачи сообщения.
* Отображать сообщения о возникающих ошибках и корректно их обрабатывать.
* Реализовать механизм надежной доставки данных (приложение должно уметь обнаруживать и корректно обрабатывать не только любое завершение клиентской/серверной частей приложения, но и разрывы физического соединения абонентов).

3. Требования к надежности

К приложению предъявляются следующие требования по надежности:

* Не допускается зависание приложения при любых действиях пользователя.
* Не допускается аварийное завершение приложения при любых действиях пользователя.
* Любая ошибочная ситуация должна корректно обрабатываться с выводом соответствующего сообщения.
* Не допускается утечка памяти/дескрипторов в процессе эксплуатации приложения.
* Не допускается полная загрузка процессора приложением в пассивном состоянии.
* Графический интерфейс не должен зависать во время передачи/приема сообщений, а также во время ожидания подключения клиента к серверу (многопоточное приложение).

4. Дополнительные требования

* Реализовать вывод на экран клиента списка текущих участников чата.
* Реализовать «приватный» режим передачи сообщений.

Вариант № 4

1. Описание задачи

Необходимо разработать приложение-чат для обмена сообщениями между несколькими клиентами через сервер с использованием блокирующих сокетов, протокола TCP.

2. Требования к реализации

* Графический интерфейс
* Ввод IP-адреса принимающего/передающего узла
* Ввод номера порта принимающего/передающего экземпляра приложения
* Отображать время приема и передачи сообщения.
* Отображать сообщения о возникающих ошибках и корректно их обрабатывать.
* Приложение должно уметь обнаруживать и корректно обрабатывать не только любое завершение клиентской/серверной частей приложения, но и разрывы физического соединения абонентов.

3. Требования к надежности

К приложению предъявляются следующие требования по надежности:

* Не допускается зависание приложения при любых действиях пользователя.
* Не допускается аварийное завершение приложения при любых действиях пользователя.
* Любая ошибочная ситуация должна корректно обрабатываться с выводом соответствующего сообщения.
* Не допускается утечка памяти/дескрипторов в процессе эксплуатации приложения.
* Не допускается полная загрузка процессора приложением в пассивном состоянии.
* Графический интерфейс не должен зависать во время передачи/приема сообщений, а также во время ожидания подключения клиента к серверу (многопоточное приложение).

4. Дополнительные требования

* Реализовать вывод на экран клиента списка текущих участников чата.
* Реализовать «приватный» режим передачи сообщений.

Вариант № 5

1. Описание задачи

Необходимо разработать клиентскую и серверную части приложения для обмена файлами между различными узлами сети с использованием блокирующих сокетов, протокола UDP.

2. Требования к реализации

* Приложение должно выполнять как функции передачи файла, так и функции приема.
* Графический интерфейс
* Ввод IP-адреса принимающего узла
* Ввод номера порта принимающего экземпляра приложения
* Выбор передаваемого файла на диске
* Выбор места сохранения полученного файла
* Отображать прогресс передачи/приема файла
* Считать и отображать время передачи файла.
* Отображать сообщения о возникающих ошибках и корректно их обрабатывать.
* Реализовать механизм надежной доставки данных (приложение должно уметь обнаруживать и корректно обрабатывать не только любое завершение клиентской/серверной частей приложения, но и разрывы физического соединения абонентов).

3. Требования к надежности

К приложению предъявляются следующие требования по надежности:

* Не допускается зависание приложения при любых действиях пользователя.
* Не допускается аварийное завершение приложения при любых действиях пользователя.
* Любая ошибочная ситуация должна корректно обрабатываться с выводом соответствующего сообщения.
* Не допускается утечка памяти/дескрипторов в процессе эксплуатации приложения.
* Не допускается полная загрузка процессора приложением в пассивном состоянии.
* Графический интерфейс не должен зависать во время передачи/приема файлов, а также во время ожидания подключения клиента к серверу (многопоточное приложение).

4. Дополнительные требования

Реализовать возможность «докачки» файлов, передача которых была прервана, например из-за разрыва соединения.

Вариант № 6

1. Описание задачи

Необходимо разработать клиентскую и серверную части приложения для обмена файлами между различными узлами сети с использованием блокирующих сокетов, протокола TCP.

2. Требования к реализации

* Приложение должно выполнять как функции передачи файла, так и функции приема.
* Графический интерфейс
* Ввод IP-адреса принимающего/передающего узла
* Ввод номера порта принимающего/передающего экземпляра приложения
* Выбор передаваемого файла на диске
* Выбор места сохранения полученного файла
* Отображать прогресс передачи/приема файла
* Считать и отображать время передачи файла.
* Отображать сообщения о возникающих ошибках и корректно их обрабатывать.
* Приложение должно уметь обнаруживать и корректно обрабатывать не только любое завершение клиентской/серверной частей приложения, но и разрывы физического соединения абонентов.

3. Требования к надежности

К приложению предъявляются следующие требования по надежности:

* Не допускается зависание приложения при любых действиях пользователя.
* Не допускается аварийное завершение приложения при любых действиях пользователя.
* Любая ошибочная ситуация должна корректно обрабатываться с выводом соответствующего сообщения.
* Не допускается утечка памяти/дескрипторов в процессе эксплуатации приложения.
* Не допускается полная загрузка процессора приложением в пассивном состоянии.
* Графический интерфейс не должен зависать во время передачи/приема файлов, а также во время ожидания подключения клиента к серверу (многопоточное приложение).

4. Дополнительные требования

Реализовать возможность «докачки» файлов, передача которых была прервана, например из-за разрыва соединения.

Вариант № 7

1. Описание задачи

Необходимо разработать приложение-чат для обмена сообщениями между несколькими клиентами через сервер с использованием блокирующих сокетов, протокола UDP.

2. Требования к реализации

* Графический интерфейс
* Ввод IP-адреса принимающего узла
* Ввод номера порта принимающего экземпляра приложения
* Отображать время приема и передачи сообщения.
* Отображать сообщения о возникающих ошибках и корректно их обрабатывать.
* Реализовать механизм надежной доставки данных (приложение должно уметь обнаруживать и корректно обрабатывать не только любое завершение клиентской/серверной частей приложения, но и разрывы физического соединения абонентов).

3. Требования к надежности

К приложению предъявляются следующие требования по надежности:

* Не допускается зависание приложения при любых действиях пользователя.
* Не допускается аварийное завершение приложения при любых действиях пользователя.
* Любая ошибочная ситуация должна корректно обрабатываться с выводом соответствующего сообщения.
* Не допускается утечка памяти/дескрипторов в процессе эксплуатации приложения.
* Не допускается полная загрузка процессора приложением в пассивном состоянии.
* Графический интерфейс не должен зависать во время передачи/приема сообщений, а также во время ожидания подключения клиента к серверу (многопоточное приложение).

4. Дополнительные требования

* Реализовать вывод на экран клиента списка текущих участников чата.
* Реализовать механизм, осуществляющий не допуск клиента к чату на основе анализа его логина и/или IP-адреса («бан» по логину, «бан» по IP-адресу).

Вариант № 8

1. Описание задачи

Необходимо разработать приложение-чат для обмена сообщениями между несколькими клиентами через сервер с использованием блокирующих сокетов, протокола TCP.

2. Требования к реализации

* Графический интерфейс
* Ввод IP-адреса принимающего/передающего узла
* Ввод номера порта принимающего/передающего экземпляра приложения
* Отображать время приема и передачи сообщения.
* Отображать сообщения о возникающих ошибках и корректно их обрабатывать.
* Приложение должно уметь обнаруживать и корректно обрабатывать не только любое завершение клиентской/серверной частей приложения, но и разрывы физического соединения абонентов.

3. Требования к надежности

К приложению предъявляются следующие требования по надежности:

* Не допускается зависание приложения при любых действиях пользователя.
* Не допускается аварийное завершение приложения при любых действиях пользователя.
* Любая ошибочная ситуация должна корректно обрабатываться с выводом соответствующего сообщения.
* Не допускается утечка памяти/дескрипторов в процессе эксплуатации приложения.
* Не допускается полная загрузка процессора приложением в пассивном состоянии.
* Графический интерфейс не должен зависать во время передачи/приема сообщений, а также во время ожидания подключения клиента к серверу (многопоточное приложение).

4. Дополнительные требования

* Реализовать вывод на экран клиента списка текущих участников чата.
* Реализовать механизм, осуществляющий не допуск клиента к чату на основе анализа его логина и/или IP-адреса («бан» по логину, «бан» по IP-адресу).

Контрольные вопросы

1. Что такое Winsock? (Если была использована другая библиотека – уметь рассказать про нее).

2. Что такое сокет?

3. В чем различие протоколов TCP и UDP?

4. Какие методы используются для приема/передачи сообщений/файлов по протоколу UDP?

5. Какая функция используется для перехода в режим ожидания подключений? Как она работает?

6. Какая функция используется при установке соединения? Как она работает?

7. В чем отличие UPD сервера от TCP сервера?

8. Что такое очередь подключений?