

Варианты индивидуального задания по курсу "Сетевые технологии в ИИС"

Общее

Цель работы

Создание клиент-серверного приложения на основе протоколов TCP/IP.

Варианты реализации

Приложение создается в среде LabView. Возможность реализации вне среды LabView, с помощью языков C, C++, Python или др. возможна и должна индивидуально обсуждаться с преподавателем.

Отчет

Отчет должен предваряться титульным листом и содержать все пункты, перечисленные ниже:

п. 1. Задание

п. 2. Клиент–серверное взаимодействие

Описание возможных (не менее двух) подходов к реализации, выбор (обоснование) реализации, описание взаимодействия между клиентом и сервером, спецификация (формат) передаваемых данных,

п. 3. Интерфейс клиентской части

Описываются органы управления/отображения и методика работы пользователя

п. 4. Интерфейс серверной части

Описываются органы управления/отображения и методика работы пользователя.

п. 5. Функционирование клиентской части

Изображение функциональной схемы (диаграммы) виртуального инструмента представляющего клиент, описание функционирования (алгоритм).

п. 6. Функционирование серверной части

Изображение функциональной схемы (диаграммы) виртуального инструмента представляющего сервер, описание функционирования (алгоритм)

п. 7 Заключение

Описание функциональных ограничений накладываемых данной реализацией (что допускает/не допускает полученная система),

Задание

Организовать в среде LabView два виртуальных инструмента один из которых считается клиентом, а другой - сервером. Сервер должен поддерживать работу с произвольным количеством клиентов. Клиент должен предоставлять пользователю возможность указания сервера, с которым будет производиться взаимодействие.

Методические указания

Планирование взаимодействия

В начале работ рекомендуется продумать исходный вариант взаимодействия. В частности, определится что значит "подключение клиента к серверу". Как для протокола TCP, так и UDP конечными точками подключения служат IP-адрес и порт (в совокупности называемые socket). IP-адрес компьютера (а, значит, и программы, на котором она запускается) диктуется конфигурацией сети. Или, если и клиент и сервер запускаются на одном компьютере - можно воспользоваться заранее известным адресом интерфейса обратной петли (local host, 127.0.0.1). Однако, номера портов должны уже определяться разработчиком. Следует ответить на вопрос: на каком порте будет работать серверная часть приложения? Соответственно, при реализации необходимо чтобы клиент отправлял данные именно на тот порт, на котором работает сервер.

Дальше требуется ответить на вопросы:

- какие данные клиент будет передавать серверу?
- какие данные сервер будет передавать клиенту?
- в какой последовательности это должно происходить?
- в каком формате будут передаваться данные? Сетевые протоколы передают данные как набор байт. Но чтобы этот набор данных нёс информацию, нужно чтобы и сервер и клиент их одинаково интерпретировали. Это и есть формат (структура данных).

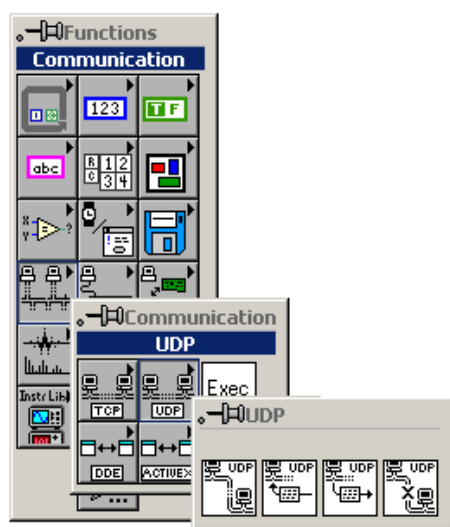
Ответы и размышление над этими вопросами следует изложить в пункте 2 отчёта.

Реализация взаимодействия

Реализация взаимодействия между сервером и клиентом осуществляется либо по протоколу UDP либо TCP - на усмотрение исполнителя.

Виртуальные инструменты (далее – ВИ) LabView для организации обмена данными по протоколу UDP находятся в меню Functions/Communication/UDP.

Для организации связи необходимо объявить об использовании UDP с

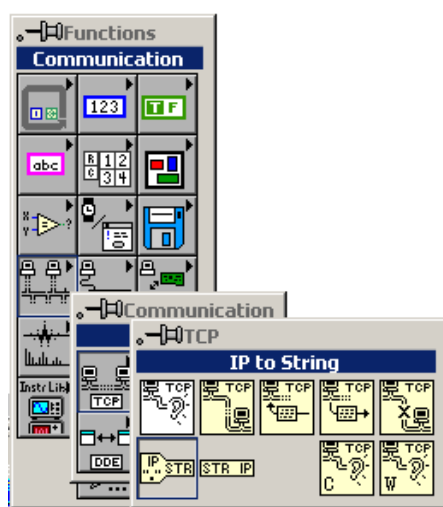


помощью ВИ “UDP open”. Опциональным входом для “UDP open” является “port number”, подачей на который целочисленного значения явно указывается номер порта, с которого или на который будут отправляться/приниматься данные. Если “port number” не указан, номер порта выбирается системой.

Для получения данных необходимо использовать ВИ “UDP Read”, со входом которого “connection id” должен быть соединен выход “connection id” ВИ “UDP open”. Для отправки данных следует использовать ВИ “UDP Write”, для которого также используется выход “connection id” от “UDP Open” вместе с информацией о хосте и порте, куда следует отправлять данные.

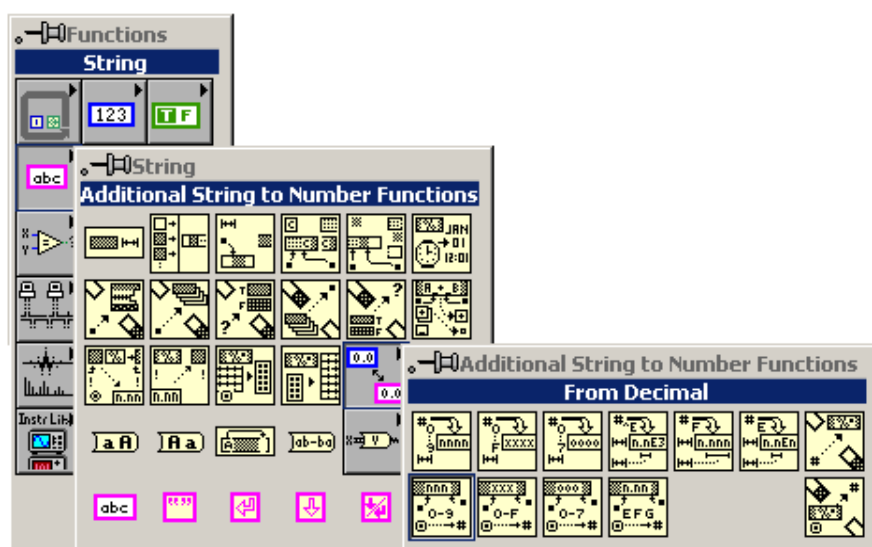
По завершении операций чтения/отправки необходимо закончить использование UDP с помощью ВИ “UDP Close”, единственным входом которого является “connection id”.

В процессе выполнения работы, наряду с указанными могут понадобиться следующие специфические ВИ:



- ВИ для преобразования IP-адреса в строку и строки в IP-адрес

находятся в разделе “Functions/Communication/TCP”



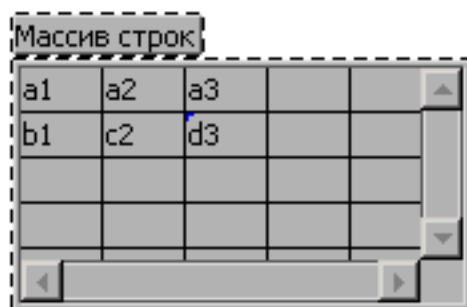
- ВИ для преобразования числа в строку (“From Decimal”) и строки в число находятся в разделе “Functions/String/Additional String to Number Functions”

Варианты индивидуальных заданий

Вариант 1

На сервере пользователь формирует два массива строк размерности $M1 \times N1$ и $M2 \times N2$. Клиент, обращаясь к серверу, синхронизируется с ним таким образом, чтобы отобразить эти два массива строк. В случае, если клиент обращается к серверу, а массив ещё не сформирован, интерфейс клиента сигнализирует о том, что данные ещё не готовы.

Один массив(таблица) строк, выглядит следующим образом:



Ориентировочный интерфейс сервера

Два массива строк, переключатель "Готов".

Сервер запускается с переключателем "Готов" в положение "выкл". Пользователь заполняет два массива строк, после чего ставит переключатель "Готов" в положение вкл.

Ориентировочный интерфейс клиента

Поле ввода адреса сервера.

Два массива строк, не доступных для редактирования.

Индикатор "готовность сервера" (лампочка).

Итоговая демонстрация

1. Запускается сервер. Запускается клиент (клиент №1). На клиенте индикатор "готовность сервера" должен загореться красным.

2. На сервере заполняется два массива строк. Переключатель "готов" переводится в положение "вкл". После этого на клиенте "само по себе" (без дополнительных действий со стороны пользователя) должны появиться два массива строк, такие же, как на сервере. При этом индикатор "готовность сервера" должен загореться зелёным.

3. Запускается второй клиент (клиент №2). На нём должно появиться то же, что и на клиенте №1.

4. На сервере переключатель "Готов" переводится в положение "выкл". На обоих клиентах индикатор "готовность сервера" должен загореться красным.

5. На сервере изменяют два массива строк. Затем переключатель готов переводится в положение "вкл". На обоих клиентах должны появиться новые два массива строк, такие же, как на сервере, и индикатор "готовность сервера" должен загореться зелёным.

Вариант 2

Клиент позволяет пользователю тем или иным образом ввести два или три числа и указать какие действия из возможных (сложение, вычитание, умножение) с ними следует произвести. Затем данные отправляются на сервер, где производятся вычисления, и результат отсылается обратно клиенту. Сервер производит вычисления в диапазоне $[0, 255]$. Если результат вычислений меньше 0 или больше 255, сервер возвращает, а клиент отображает, ошибку. Соответственно, возможны два типа ошибок: "результат меньше 0" и "результат больше 255". Клиент отображает ответ, полученный от сервера.

Ориентировочный интерфейс сервера

Не требуется какого-либо интерфейса сервера. Можно на сервере установить элементы, облегчающие отладку приложения.

Ориентировочный интерфейс клиента

Поле ввода адреса сервера.

Поле "результат".

Поле/индикатор ошибки (может отсутствовать. Плюсом будет если ошибка будет отображаться в поле "результат").

Возможно, кнопка "Отправить" (Посчитать).

Ввод арифметического выражение может быть сделан как в виде одной строки, так и в виде предопределённых элементов. Допускается любой интерфейс, который позволит так или иначе вводить выражения:

2+2

2*3
2*3+1
1-5+3
300-10*5
и пр.

Итоговая демонстрация

1. Запускается сервер.
2. Запускается клиент (клиент №1). На клиенте вводится 2+2. В поле "Результат" должно появиться 4.
3. Запускается ещё один клиент (клиент №2). На клиенте вводится 10+2*5. В поле "Результат" должно появиться 10.
4. Запускается ещё один клиент (клиент №3). На клиенте вводится 100*3-50. В поле "Результат" должно появиться 250.
5. На клиенте №1 вводится 50+200+6. В поле "Результат"/на индикаторе должно появиться "результат больше 255".
6. На клиенте №2 вводится 12-6*3. В поле "Результат"/на индикаторе должно появиться "результат меньше 0".

Вариант 3

Сервер позволяет устанавливать два значения температуры: для "Канала 1" и для "Канала 2". Эти значения он должен предоставлять по протоколу http, внутри html страниц. Клиентом(ами) являются стандартные интернет-браузеры.

Для этого реализуется функционально ограниченный HTTP-сервер. HTTP-сервер должен обрабатывать запросы типа "GET" и динамически формировать HTTP-ответ, содержащий html-страницы. В html-страницах "Канала 1" и "Канала 2" должны фигурировать значения, установленный пользователем для "Канала 1" и "Канала 2" соответственно.

Ориентировочный интерфейс сервера

Два поля ввода: "Температура канала 1", "Температура канала 2".

Ориентировочный интерфейс клиента

Клиентом является интернет-браузер.

Итоговая демонстрация

1. Запускается сервер. На сервере устанавливаются значения для "Температура канала 1" - 23, для "Температура канала 2" - 18.
2. Запускается интернет-браузер. В адресную строку вводится адрес сервера. В результате, отображается html-страница следующего, ориентировочного, вида:

Индивидуальное задание

Нажмите на ссылки, чтобы узнать параметры каналов:

[Параметры канала 1](#)

[Параметры канала 2](#)

Выполнили студенты XXX

3. В браузере нажимается ссылка "Параметры канала 1". В результате, отображается html-страница следующего, ориентировочного, вида:

Индивидуальное задание

Параметры канала 1:
температура 23 град.

[вернуться в начало](#)

4. В браузере нажимается ссылка "вернуться в начало", затем "Параметры канала 2". В результате, отображается html-страница следующего, ориентировочного, вида:

Индивидуальное задание

Параметры канала 2:
температура 18 град.

[вернуться в начало](#)

Вариант 4

Сервер позволяет пользователю ввести два числа: "Операнд 1" и "Операнд 2". Клиент, в свою очередь, позволяет пользователю тем или иным образом указать какое действие из возможных (сложение, вычитание, умножение) и при каком условии с ними должен произвести сервер. Условия задаются как:

операндN больше или меньше числа Y
где N - 1 или 2; Y - задаётся пользователем.

Т.е. пользователь на клиенте выбирает, например, действие "сложить", а в условии определяет "операнд 1" < 10. Это команда передаётся на сервер. Сервер, если, действительно, "операнд 1" < 10, складывает "операнд1" и "операнд2" и возвращает результат клиенту. Клиент его отображает. Если "операнд 1" больше или равен 10, сервер не производит вычислений, но возвращает клиенту статус, означающий "условие не выполнено".

Ориентировочный интерфейс сервера

Два поля ввода для чисел: "Операнд 1" и "Операнд 2".

Ориентировочный интерфейс клиента

Поле ввода адреса сервера.

Поле "Действие": позволяет выбрать одно из +, -, *

Группа "Условие" состоит из поля выбора Операнд1 или Операнд2, поля выбора "больше" или "меньше" и поля ввода "Параметр условия" (число Y).

Поле "результат".

Поле/индикатор не применимости условия (может отсутствовать. Плюс будет если в поле "результат" будет писаться "условие не выполнено"). Возможно, кнопка "Отправить" (Посчитать).

Итоговая демонстрация

1. Запускается сервер. В поле "Операнд1" вводится число 40, в поле "Операнд2" вводится число 100.
2. Запускается клиент (клиент №1). Действие "сложить", условие "операнд 2 > 50". В поле "Результат" должно появиться 140.
3. На клиенте (№1) задаётся действие "вычесть", условие "операнд 1 < 50". В поле "Результат" должно появиться -60.
4. На клиенте (№1) задаётся действие "вычесть", условие "операнд 1 < 20". На клиенте должен появиться статус "условие не выполнено".
5. Запускается ещё один клиент (клиент №2). Действие "сложить", условие "операнд 2 > 5000". На клиенте должен появиться статус "условие не выполнено".
6. На сервере в поле "Операнд 2" вводится 7250.
7. На клиенте (№2) задаётся действие "сложить", условие "операнд 2 > 5000". В поле "Результат" должно появиться 7290.

Вариант 5

Интерфейс сервера представляет собой панель управления, состоящую из выключателем "Готов", трёх выключателей "Канал 1", "Канал 2", "Канал 3" и трёх задаваемых пользователем числовых значений "Параметр 1", "Параметр 2", "Параметр 3". Клиент, должен, отобразить индикаторами действующие Каналы и соответствующие им значения, в случае, если переключателем "Готов" на сервере находится в положении "вкл". Если переключателем "Готов" на сервере находится в положении "выкл", клиент должен сигнализировать, что сервер не готов и не отображать данные.

Ориентировочный интерфейс сервера

Выключатель "Готов", по умолчанию установленный в положение "выкл".

Выключатель "Канал 1", по умолчанию установленный в положение "выкл", и соответствующее ему поле числового ввода (или вращающаяся ручка, или иное) "Параметр 1".

Выключатель "Канал 2", по умолчанию установленный в положение "выкл", и соответствующее ему поле числового ввода (или вращающаяся ручка, или иное) "Параметр 2".

Выключатель "Канал 3", по умолчанию установленный в положение "выкл", и соответствующее ему поле числового ввода (или вращающаяся ручка, или иное) "Параметр 3".

Ориентировочный интерфейс клиента

Поле ввода адреса сервера;

Индикатор "готовность сервера" (лампочка);
Индикатор "Канал 1" и поле вывода "Параметр 1";
Индикатор "Канал 2" и поле вывода "Параметр 2";
Индикатор "Канал 3" и поле вывода "Параметр 3";

Итоговая демонстрация

1. Запускается сервер.
2. Запускается клиент №1. Запускается клиент №2. На клиентах индикатор "готовность сервера" должен загореться красным.
3. На сервере вводится Параметр 2 равный 66 и включается выключатель "Канал 2". На обоих клиента должен загореться индикатор "Канал 2" и в поле вывода "Параметр 2" должно появиться число 66.
4. На сервере вводится Параметр 3 равный 590 и включается выключатель "Канал 3". На обоих клиента должен дополнительно загореться индикатор "Канал 3" и в поле вывода "Параметр 3" должно появиться число 590.
5. На сервере вводится Параметр 1 равный 2 и включается выключатель "Канал 1". На обоих клиента должен дополнительно загореться индикатор "Канал 1" и в поле вывода "Параметр 1" должно появиться число 2.
6. На сервере вводится Параметр 2 равный 10. На обоих клиента в поле вывода "Параметр 2" должно появиться число 10.
7. На сервере переключатель "Готов" переводится в положение "выкл". На обоих клиента индикатор "готовность сервера" должен загореться красным. Индикаторы каналов должны выключиться.

Вариант 6

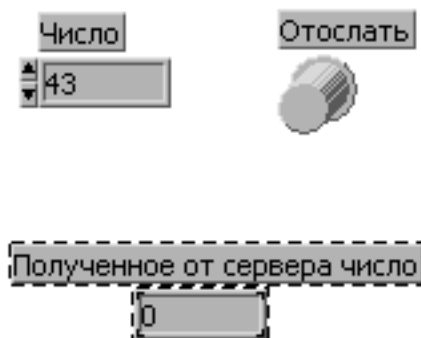
Интерфейс клиента позволяет задать число и инициировать отправку, например, нажатием кнопки. Клиент отправляет заданное число на сервер, а сервер, в свою очередь, отправляет это число всем клиентам. Клиенты отображают полученное от сервера число. Таким образом, отправка числа с одного (произвольного) клиента приводит к отображению этого числа всеми клиентами.

Ориентировочный интерфейс сервера

Не требуется какого-либо интерфейса сервера. Можно на сервере установить элементы, облегчающие отладку приложения.

Ориентировочный интерфейс клиента

Поле ввода адреса сервера;
Элементы для ввода, отправки и отображения числа.



Итоговая демонстрация

1. Запускается сервер.
2. Запускается клиент №1. Запускается клиент №2. Запускается клиент №2.
3. На клиенте №2 вводится число 43 и нажимается кнопка "Отослать". На всех клиентах в поле "Полученное от сервера число" должно появиться 43.
4. На клиенте №1 вводится число 999 и нажимается кнопка "Отослать". На всех клиентах в поле "Полученное от сервера число" должно появиться 999.
5. На клиенте №3 вводится число 8000 и нажимается кнопка "Отослать". На всех клиентах в поле "Полученное от сервера число" должно появиться 8000.

Вариант 7

Сервер генерирует и отображает на графике сигнал синусоидального вида (ВИ sine pattern). Клиент должен обеспечивать удаленное управление параметрами генерации сигнала на сервере. В частности, клиент управляет следующими параметрами: включением/выключением генерации сигнала, амплитудой, частотой, количеством генерируемых значений (sampling frequency).

Ориентировочный интерфейс сервера

График, на котором отображается генерируемый сигнал.

Ориентировочный интерфейс клиента

Поле ввода адреса сервера.

Выключатель. В положении "вкл" сервер запускает генерацию сигнала. В положении "выкл" сервер останавливает генерацию сигнала.

Ручки управления амплитудой, частотой, количеством генерируемых значений.

Итоговая демонстрация

1. Запускается сервер.
2. Запускается клиент. На сервере график остается пустым.

3. На клиенте выключатель ставится в положение "вкл". Сервер начинает генерировать сигнал, что видно на его графике.

4. На клиенте вращают ручки управления амплитудой, частотой, количеством генерируемых значений. На сервере соответственно меняются параметры генерации сигнала, что видно на графике.

Вариант 8

На сервере формируется список записей с полями "наименование" и "количество" (ориентировочный вид представлен на рисунке).

| | |
|------------|----|
| муфта | 15 |
| переходник | 10 |
| разъём | 3 |
| кабель ВВГ | 21 |

Поле "количество" не может быть меньше нуля. Общее количество записей специально не ограничивается, в случае необходимости может установлено максимальное количество 100.

Интерфейс клиента позволяет пользователю ввести "наименование"(строка S) и "изменение" (число X). После отправки этих данных на сервер, сервер должен выполнить указанную операцию: произвести перерасчёт записи с "наименованием" S прибавив к "количеству" число X и отобразить изменения.

Например, если на клиенте ввести в качестве "наименования"(S) "переходник", в качестве "изменения"(X) -4, то вышеприведенный список на сервере должен измениться следующим образом:

| | |
|------------|----|
| муфта | 15 |
| переходник | 6 |
| разъём | 3 |
| кабель ВВГ | 21 |

Помимо этого, сервер формирует и отправляет, а клиент отображает результат операции. В случае успеха, в интерфейсе клиента должна отобразиться строка "операция выполнена успешно". В случае:

- если "наименование" не содержится в списке сервера, на клиенте должен загореться красный индикатор и отобразиться строка "наименование неверно";

- если "количество" после выполнения операции (прибавления X) станет меньше 0, операция не производится и на клиенте должен загореться красный индикатор и отобразиться строка "количество меньше нуля";

Ориентировочный интерфейс сервера

Таблица на два столбца: "наименование" и "количество".

Ориентировочный интерфейс клиента

Поле ввода адреса сервера;

Поле строкового ввода "Наименование".

Поле числового ввода "Количество".

Возможно, кнопка "Отправить".

Отображения результата отправки: индикатор красная лампочка, строка состояния.

Итоговая демонстрация

1. Запускается сервер, на котором сформирован следующий список:

| | |
|------------|----|
| муфта | 15 |
| переходник | 10 |
| разъём | 3 |
| кабель ВВГ | 21 |

2. Запускается клиент №1. Запускается клиент №2.

3. На клиенте 2 вводится "наименование": "разъём", "изменение": 3, (нажимается кнопка отправить). На сервере список должен принять следующий вид:

| | |
|------------|----|
| муфта | 15 |
| переходник | 10 |
| разъём | 6 |
| кабель ВВГ | 21 |

На клиенте строка состояния должна отобразить: "операция выполнена успешно".

4. На клиенте 1 вводится "наименование": "кабель ВВГ", "изменение": -6, (нажимается кнопка отправить). На сервере список должен принять следующий вид:

| | |
|------------|----|
| муфта | 15 |
| переходник | 10 |
| разъём | 6 |
| кабель ВВГ | 15 |

На клиенте строка состояния должна отобразить: "операция выполнена успешно".

5. На клиенте 2 вводится "наименование": "абвг", "изменение": 2, (нажимается кнопка отправить). На сервере список не должен измениться. На клиенте должен загореться красный индикатор и строка состояния должна отобразить: "наименование неверно".

6. На клиенте 1 вводится "наименование": "переходник", "изменение": -11, (нажимается кнопка отправить). На сервере список не должен измениться. На

клиенте должен загореться красный индикатор и строка состояния должна отобразить: "количество меньше нуля".