

Лабораторная работа №8. Использование механизма сокетов в ОС Windows

Ход работы:

1. Ознакомиться с теоретическими сведениями по организации межпроцессного взаимодействия на основе механизма сокетов.
2. Получить у преподавателя индивидуальный вариант задания, который должен предусматривать разработку серверной и клиентской частей программы и взаимодействие между ними посредством механизма сокетов.
3. Разработать и отладить обе программы – сервер и клиент - в соответствии с полученным заданием.
4. Написать отчет и представить его к защите вместе с исполняемыми модулями программ и их исходными текстами.

Содержание отчета:

1. Цель работы с постановкой задачи.
2. Описание использованных функций WinSock API работы с сокетами.
3. Примеры использования (вызова) этих функций как в серверной, так и в клиентской частях программы.
4. Примеры работы программы (не менее трех примеров работы клиентской и не менее трех - серверной).

Во время защиты лабораторной работы необходимо иметь при себе исполняемые модули программы, ее исходные тексты, а также твердую копию отчета.

Дополнительные источники:

1. Windows Sockets 2 - [http://msdn.microsoft.com/en-us/library/windows/desktop/ms740673\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740673(v=vs.85).aspx);
2. Winsock Functions - [http://msdn.microsoft.com/en-us/library/windows/desktop/ms741394\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741394(v=vs.85).aspx).

Требуется разработать серверную и клиентскую части программы в виде двух консольных Windows-приложений, которые обмениваются сообщениями посредством заданного механизма. Предполагается, что клиент и сервер должны позволять взаимодействовать, будучи запущенными на разных сетевых узлах. На стороне клиента необходимо запрашивать у пользователя IPv4-адрес сервера.

При реализации обязательно использование функций Winsock2 API. При наличии альтернатив предпочтение должно всегда отдаваться асинхронным версиям коммуникационных и вспомогательных процедур, то есть должны использоваться функции, чьи имена начинаются с префикса WSA).

Обе части программы должны быть устойчивыми к некорректному пользовательскому вводу, а сообщения об ошибках должны быть информативными.

Исходный код пользовательского интерфейса должен быть логически и физически отделен от кода, реализующего операции по обмену сообщениями.

В нечетных вариантах заданий используются потоковые сокет и протокол TCP, а четных — дейтаграммные сокет и протокол UDP.

Справочный материал, а также варианты заданий приводятся в приложенном документе.