**7.7. Практическая работа № 6. Работа с интерфейсом Named Pipe**

**7.7.1. Цель и задачи работы**

Основнойцелью практической работы является приобретение навыков использования интерфейса Named Pipe для разработки распределенного приложения.

Результатом практической работы являются разработанное распределенное приложение, применяющее интерфейс Named Pipe.

**7.7.2. Теоретические сведения**

Теоретические сведения необходимые для выполнения практической работы изложены в четвертой главе пособия.

**7.7.3. Разработка серверной части распределенного приложения**

**Задание 1.** Ознакомьтесь со схемой сервера, изображенной на рисунке 4.2.1. Создайте с помощью Visual Studio новое консольное приложение **ServerNP** (наименование проекта), которое будет использовано для построения серверной части распределенного приложения (сервера). Реализуйте блоки 1 и 4 сервера. В блоке 1 используются функции **CreateNamedPipe** и **ConnectNamedPipe**, описание которых приводится в разделе 4.3 пособия. Там же приведены примеры использования этих функций. Следует обратить внимание на формат имени канала, используемый функции **CreateNamedPipe –** ондолжен быть локальным. Пусть имя создаваемого именного канала будет ***Tube***. В блоке 4 используется функция **DisconnectNamedPipe**  для разрыва соединения. Описание функции приводится в разделе 4.3 пособия. Разработайте функции обработки ошибок интерфейса Named Pipe, работающие по тому же принципу, что и функции обработки ошибок Winsock2, используемые в предыдущих практических работах. Запустите приложение **ServerNP** на выполнение и убедитесь, что поток приостановился для ожидания соединения после вызова функции **ConnectNamedPipe**.

**7.7.4. Разработка клиентской части распределенного приложения**

**Задание 2.** Ознакомьтесь со схемой клиента, изображенной на рисунке 4.2.1. Создайте с помощью Visual Studio новое консольное приложение **ClientNP**, которое будет использовано для построения клиентской части распределенного приложения (клиента). Реализуйте блоки 1 и 4 клиента. В блоке 1 применяется функция **CreateFile**, описание которой приводится в разделе 4.4 пособия. Установите в параметрах вызова функции **CreateFile**, такое же имя именованного канала, что и для сервера (задание 1). Запустите программу **ClientNP** отдельно (без сервера) и проверьте работоспособность функций обработки ошибок.

**7.7.5. Обмен данными между клиентом и сервером**

**Задание 3.** Запустите на выполнение программу **ServerNP**. После того, как программа  **ServerNP** перейдет в состояние ожидания, запустите на этом же компьютере программу **ClientNP**. Убедитесь, что программа   **ServerNP** вышла из состояния ожидания и успешно завершилась. Программа **ClientNP** тоже должна завершиться без ошибок.

**Задание 4**. Реализуйте блоки 2 и 3 программ **ServerNP** и **ClientNP**. Добейтесь, чтобы программы обменялись сообщениями с помощью функций **WriteFile** и **ReadFile**, описание которых приводится в разделе 4.5 пособия.

**Задание 5**. Внесите изменения в программы **ServerNP** и **ClientNP** таким образом, чтобы программы взаимодействовали также, как и программы **ServerT** и **ClientT** в заданиях 14 и 15 практической работы № 2.

**Задание 6**. Внесите изменения в программу **ClientNP**  и добейтесь взаимодействие программ клиента и сервера в случае расположения на разных компьютерах локальной сети. Следует иметь в виду, что при вызове функции **CreateFile**,теперьследует использоватьсетевой формат имени именованного канала, описанный в разделе 4.3 пособия. В сетевом формате используется символическое имя серверного компьютера. Напомним, что это имя можно получить с помощью утилиты **hostname**, описанной в разделе 2.9.

**Задание 7**. Разработайте новую программу **ClientNPt**, использующую функцию **TransactNamedPipe** (раздел 4.6 пособия) вместо пары функций **WriteFile** и **ReadFile**. Добейтесь взаимодействия программы **ClientNPt**  с сервером **ServerNP**.

**Задание 8**. Разработайте еще одну новую программу **ClientNPсt**, использующую функцию **CallNamedPipe** (раздел 4.6 пособия). Добейтесь взаимодействия программы **ClientNPct**  с сервером **ServerNP**.