**7.8. Практическая работа № 7. Работа с интерфейсом Mailslots**

**7.8.1. Цель и задачи работы**

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

Результатом практической работы являются разработанное распределенное приложение, применяющее интерфейс Mailslots.

**7.8.2. Теоретические сведения**

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

**7.8.3. Разработка серверной части распределенного приложения**

**Задание 1.** Ознакомьтесь со схемой сервера, изображенной на рисунке 5.2.1. Создайте с помощью Visual Studio новое консольное приложение **ServerMS** (наименование проекта), которое будет использовано для построения серверной части распределенного приложения (сервера). Реализуйте блоки 1 и 3 сервера. В блоке 1 используется функция **CreateMaislot**, описание которой приводится в разделе 5.3 пособия. Следует обратить внимание на формат имени канала, используемый функции **CreateMaislot –** ондолжен быть локальным. Пусть имя создаваемого именного канала будет ***Box***. Кроме того, установите в параметрах функции **CreateMaislot** бесконечный интервал ожидания для функции **ReadFile** и максимальную длину сообщения равную **300** байт. Разработайте функции обработки ошибок интерфейса Mailslot, работающие по тому же принципу, что и функции обработки ошибок Winsock2, используемые в предыдущих практических работах. Запустите приложение **ServerMS** на выполнение и убедитесь, что поток не приостановился для ожидания и программа **ServerMS**  завершилась без ошибок.

**Задание 2.** Реализуйте в программе  **ServerMS** блок 2 схемы сервера (рисунок 5.2.1) и запустите на выполнение. В блоке 2 применяется универсальная функция **ReadFile**, описание которой приводится в разделе 4.5 пособия. Запустите программу на выполнение. Убедитесь, что главный поток программы **ServerMS** приостановился для ожиданиясообщения от клиента. Завершите программу с помощью отладчика.

**Задание 3.** Измените в параметрах вызова функции **CreateMaislot** программы **ServerMS** интервал времени ожидание на значение соответствующее трем минутам. Запустите программу **ServerMS** на выполнение и убедитесь, что главный поток приостановился на три минуты для ожидания сообщения клиента. Обработайте этот случай: выведите соответствующее сообщение на консоль сервера.

**7.8.4. Разработка клиентской части распределенного приложения**

**Задание 4.** Ознакомьтесь со схемой клиента, изображенной на рисунке 5.2.1. Создайте с помощью Visual Studio новое консольное приложение **ClientMS**, которое будет использовано для построения клиентской части распределенного приложения (клиента). Реализуйте схему клиента полностью. Для реализации первого блока необходимо использовать, функцию **CreateFile**,описание которой приводится в разделе 5.4. Установите локальный формат имени почтового сервера в параметрах функции. Для отправки сообщения используется универсальная функция **WriteFile**, описание которой можно найти в разделе 4.5. Разработайте функции обработки ошибок интерфейса Mailslot. Программа **ClientNS** должна пересылать серверу сообщение ***Hello from Maislot-client***. Запустите программу **ClientMS**  без наличия на компьютере сервера. Убедитесь, что обработка ошибок работает корректно.

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

**Задание 5.** Запустите на выполнение программу **ServerMS.**  После того, как программа **ServerMS** перейдет в состояние ожидания запустите на этом же компьютере программу **ClientMS**. С помощью отладчика убедитесь, что сообщение от клиента к серверу передается корректно.

**Задание 6.** Внесите такие изменения в программу **ClientMS** , чтобы она могла пересылать сообщение серверу **ServerMS**, который расположен на другом компьютере. Добейтесь работоспособности распределенного приложения **ClientMS- ServerMS**.

**Задание 7.** Внесите такие изменения в программу **ClientMS**, чтобы она могла пересылать сообщение нескольким экземплярам сервера **ServerMS**, расположенным на других компьютерах локальной сети. Добейтесь работоспособности одной клиентской программы с тремя экземплярами сервера.

**Задание 8.**  Увеличьте размер максимальный размер пересылаемого сообщения до 500 (параметры функции **CreateMaislot**) и проверьте возможность работы одного клиента с тремя серверами.

**Задание 9.**  Оцените скорость пересылки 1000 сообщений от одного клиента одному серверу, по тому же принципу, как это было сделано в задании 14 практической работы № 2.