

УПРАВЛЕНИЕ РЕСУРСАМИ СЕТЕВОЙ ИНФОРМАЦИОННО - ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ

Кипрушкин С.А., Курсков С.Ю., Носович Н.Г.
*Петрозаводский государственный университет,
Петрозаводск*

Целью данной работы является реализация механизма администрирования сетевой информационно-измерительной системы, обеспечивающей удаленный доступ к своим информационным и техническим ресурсам в сетях на базе стека протоколов TCP/IP и предназначенной для поддержки научных исследований и образовательного процесса в области оптической спектроскопии на физико-техническом факультете Петрозаводского государственного университета.

На аппаратном уровне система представляет собой комплекс автоматизированных рабочих мест, объединенных компьютерной сетью [1, 2]. Ключевым звеном системы является коммуникационный сервер, в задачи которого входит поддержка многопользовательского режима, корректное распределение ресурсов между клиентами, мониторинг системы и обеспечение безопасности. Другими компонентами системы являются серверы оборудования и программы-клиенты, осуществляющие сбор, накопление и обработку информации, а также управляющие ходом эксперимента

Доступ администратора к коммуникационному серверу осуществляется с помощью стандартного браузера и Web-сервера. Взаимодействие Web-сервера с коммуникационным сервером реализовано посредством сервлета. Сервлеты – это модули расширения для Web-серверов с поддержкой Java. В данной работе сервлет используется для организации сетевого обмена с коммуникационным сервером в соответствии с протоколом системы и динамической генерации HTML-страниц.

Администратор подключается к коммуникационному серверу как обычный клиент, но с паролем в поле "данные" кадра запроса и указанием длины пароля в поле "ключ", код команды передается в поле "функция". После проверки пароля данному клиенту присваивается идентификационный номер (CID), равный нулю, по которому разрешается выполнение дополнительных функций, таких как просмотр информации о клиентах и используемых ресурсах, удаление клиента из системы, а также освобождение ресурса. Команды администратора выполняются в основном потоке, что позволяет управлять дочерними потоками, которые обслуживают клиентов.

После подключения администратора к коммуникационному серверу и при каждой перезагрузке сервлета коммуникационный сервер посылает администратору информацию, содержащуюся в классах MainClientInfo и MainServerInfo. Данные классы содержат методы для упаковки и передачи информации о клиентах, используемых ресурсах и серверах оборудования. Эта информация, в частности, включает адрес ресурса, IP-адрес клиента-владельца, идентификационный номер клиента-владельца (CID), IP-адрес сервера оборудования, номер порта, на котором сервер оборудования ожидает связи с клиентом, и его

псевдоним. Данные, полученные администратором, выводятся в виде таблицы, в которой имеется форма для работы с клиентами.

Вся информация, включая псевдоним клиента, перед отправкой администратору запаковывается в расширенное поле данных кадра ответа, т. е. в поле "тип данных" передается единица, а в поле "данные" – количество дополнительных байтов данных. Сами данные передаются вслед за основным кадром.

Форма, содержащаяся в HTML-странице, позволяет администратору посылать коммуникационному серверу команды INFO, STOP, SUSPEND, RESUME и RESOURCE FREE (соответственно: передача информации о клиентах и серверах оборудования, отключение клиента от коммуникационного сервера, и освобождение всех его ресурсов, приостановка и последующий запуск работающего клиента, а также освобождение занятого клиентом ресурса). После выполнения команды HTML-страница обновляется. Применение методов класса Thread – stop(), suspend() и resume() – для работы с потоками клиентов возможно, так как коммуникационный сервер не запускает для администратора отдельный поток, а обслуживает его в основном классе коммуникационного сервера StartCServer.

В заключение отметим, что применение технологии сервлетов позволяет просто и эффективно управлять аппаратными ресурсами сетевой информационно-измерительной системы. Преимуществом использованной технологии является то, что она позволяет легко конструировать быстро работающие приложения для серверов, полностью отказавшись от CGI и Perl. Кроме того, сервлеты являются платформенно-независимыми и могут выполняться на любой программно-аппаратной платформе без перекомпиляции или модификации. Они загружаются только один раз, при этом Web-сервер запускает одну виртуальную Java-машину. Сервлет постоянно находится в памяти и не требует повторной загрузки до тех пор, пока его содержание не изменится. В случае модификации сервлет может перезагрузиться без перезагрузки сервера.

Работа выполнена при поддержке Американского фонда гражданских исследований и развития (CRDF) (проект PZ-013-02) и Министерства образования РФ.

СПИСОК ЛИТЕРАТУРЫ

1. Гаврилов С.Е., Жиганов Е.Д., Кипрушкин С.А., Курсков С.Ю. Распределенная информационно-измерительная система для удаленного управления экспериментом в области оптической спектроскопии // Научный сервис в сети Интернет: Тр. Всерос. науч. конф. М.: Изд-во Моск. ун-та, 2002. С. 157–159.
2. Гаврилов С.Е., Кипрушкин С.А., Королев Н.А., Курсков С.Ю. Распределенная информационно-измерительная система для спектроскопического анализа пучковых и плазменных объектов // Материалы семинаров-школ молодых ученых, студентов и аспирантов "Методы и техника экспериментального исследования процессов самоорганизации упорядоченных структур в плазменно-пылевых образованиях" (2002), "Фундаментальные проблемы приложений физики

низкотемпературной плазмы (2003)". Петрозаводск, 2004. С. 279–289.

МНОГОПОТОЧНЫЙ СЕРВЕР МИКРОКОНТРОЛЛЕРОВ INTEL MCS-196

Кипрушкин С.А., Курсков С.Ю., Мурсалимов О.А.
*Петрозаводский государственный университет,
Петрозаводск*

Данная работа посвящена созданию многопоточного сервера микроконтроллеров MCS-196 распределенной информационно-измерительной системы, используемой для спектроскопического анализа пучковых и плазменных объектов [1–3]. Разработанный сервер предназначен для подключения к системе до четырех указанных микроконтроллеров и управления их ресурсами. Микроконтроллеры обеспечивают интеграцию в распределенную систему цифровых счетчиков и аналоговых вакуумметров.

Прикладной протокол распределенной системы базируется на стеке протоколов TCP/IP, что обеспечивает доступ к ее ресурсам в сетях Интранет/Интернет. Прикладной протокол регламентирует порядок обмена информацией между клиентом и сервером, форматы кадров запроса и ответа, определяет коды команд системы и коды ошибок, которые могут возникнуть в ходе сеанса связи.

Микроконтроллеры подключаются к ПК через последовательные порты COM1-COM4 по стандарту RS-232.

Сервер микроконтроллеров представляет собой программу на языке Java, которая создает до 4-х отдельно работающих потоков, каждый из которых управляет своим микроконтроллером, подключенным к соответствующему COM-порту. В его задачу входит определение допустимости запрошенной функции и указанного адреса, передача запроса микроконтроллеру, а также пересылка клиенту ответа или номера ошибки при возникновении исключительной ситуации. В задачу сервера не входит обеспечение защиты ресурсов микроконтроллеров от разных клиентов распределенной информационно-измерительной системы. Эта функция возложена на коммуникационный сервер системы, поскольку только через него клиенты получают доступ к серверу микроконтроллеров. Коммуникационный сервер системы различает микроконтроллеры по адресу COM-порта, который включается в поле адреса ресурса кадра запроса.

В состав сервера входят следующие классы:

- MCS_Server – основной класс сервера – реализует "прослушивание сети", подключение коммуникационного сервера, реализует алгоритм обслуживания клиента, в частности, обеспечивает проверку корректности параметров запроса клиента и выполнение команды управления микроконтроллером;
- CserverProtocol – в данном интерфейсе определены коды операций, ошибок и другие константы коммуникационных протоколов (общий для системы);
- QueryToEServer – в данном классе определен объект "кадр запроса" к серверу и методы для работы с этим объектом (общий для системы);

- ReplyFromEServer – в классе определен объект "кадр ответа" и методы для работы с этим объектом (общий для системы);

- MCS96Lib – данный класс содержит библиотеку методов для работы с микроконтроллерами, а также ряд дополнительных методов выполняющих вспомогательные функции. Основные методы класса: Reset – сброс микроконтроллера в исходное состояние, Init – инициализация микроконтроллера, Load_Hex – загрузка откомпилированной программы пользователя в микроконтроллер, TestLine – проверка связи с микроконтроллером и его инициализация, Run – выполнение загруженной программы пользователя, ReceiveData – прием данных от микроконтроллера и их последующая обработка. Также в этом классе объявлены три внешних метода, реализованных на языке Си: InitCom – установка параметров COM-порта, InCom – чтение байта из порта ввода/вывода, OutCom – запись байта в порт ввода/вывода.

В случае изменения алгоритма получения данных с устройства, подключенного к определенному микроконтроллеру, необходимо изменить программу в ПЗУ микроконтроллера. Это делается программой-клиентом, управляющей данным микроконтроллером. Программа, написанная на языке Assembler-96 с учетом изменений в эксперименте, компилируется, и полученный HEX-файл прошивки пересылается программой-клиентом на сервер. Если данный микроконтроллер не занят, то сервер загружает прошивку в память микроконтроллера.

При создании сервера микроконтроллера использовался пакет разработчика Java – J2SE Development Kit v.5.0. Применение технологии Java обеспечивает возможность использования разработанного сервера на любых программно-аппаратных платформах, для которых реализована виртуальная Java-машина.

Работа выполнена при поддержке Американского фонда гражданских исследований и развития (CRDF) (проект PZ-013-02) и Министерства образования РФ.

СПИСОК ЛИТЕРАТУРЫ

1. Гаврилов С.Е., Жиганов Е.Д., Кипрушкин С.А., Курсков С.Ю. Распределенная информационно-измерительная система для удаленного управления экспериментом в области оптической спектроскопии // Научный сервис в сети Интернет: Тр. Всерос. науч. конф. М.: Изд-во Моск. ун-та, 2002. С. 157–159.
2. Gavrilov S.E., Kiprushkin S.A., Kurskov S.Yu., Khakhaev A.D. Distributed information system with remote access to physical equipment // Proceedings of the International Conference on Computer, Communication and Control Technologies: CCCT '03 and The 9th International Conference on Information Systems Analysis and Synthesis: ISAS '03 (July 31, August 1–2, 2003, Orlando, Florida, USA). Orlando, 2003.
3. Kiprushkin S.A., Korolev N.A., Kurskov S.Yu., Khakhaev A.D. Data security in the distributed information measurement system // Proceedings of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004) (July 18 – 21, 2004, Orlando, Florida, USA). Orlando, 2004. V. 1. P. 13–16.