

stabilization of semiconductor lasers by resonant optical feedback. *Optics Letters*, 12(11):876-878, 1987.
3. http://www.rn-wissen.de/index.php/Regelungstechnik#Dimensionierung_mit_dem_Bodediagramm.
4. Schünemann U., Engler H., Grimm R., Weidemüller M., Zielonkowski M. Simple scheme for tunable frequency offset locking of two lasers. *Rev. Sci. Instrum*, 1999. 5. SIOS Messtechnik GmbH, Germany, User Manual Stabilized HeNe Laser SL 03-Series, 2010. 6. Sacher Lasertechnik GmbH, Germany, User Manual LYNXTM Series[Datenblatt]. Maburg: Sacher Lasertechnik GmbH, Germany, 2005. 7. http://www.ictradenet.com/EVAL-ADF411X_EB1 8. Best R.E. *Phase-Locked Loops, Design, Simulation & Applications*. McGraw-Hill, 3rd edition, 1997.

УДК 681.3, 621.3

І.І. Пастернак

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

МОДУЛЬНИЙ ІНТЕРФЕЙС КЛІЄНТ-СЕРВЕРНОЇ ВЗАЄМОДІЇ

© Пастернак І.І., 2012

Запропоновано варіант взаємодії клієнта з сервером через модульний інтерфейс в глобальній мережі.

Ключові слова: модульний інтерфейс, клієнт, сервер, клієнт-серверна взаємодія.

The variant of the interaction of the client to the server through a modular interface in the global network.

Key words: modular interface, client, server, client-server interaction.

Вступ

Практично всі сервіси мережі Інтернет побудовані на технології клієнт-сервер. Для роботи цієї технології в мережі необхідно:

- 1) фізично приєднати комп'ютер до одного з вузлів мережі Інтернет;
- 2) одержати IP-адресу на постійній або тимчасовій основі;
- 3) встановити та налаштувати апаратне і програмне забезпечення.

Доступ до мережі користувачі отримують через провайдера [1, 2]. ISP (Internet service provider, постачальник послуг Інтернет) – забезпечує персональний доступ у мережу Інтернет, доставку та збереження електронної пошти (e-mail), а також надання інструкцій щодо налаштування системи. Як правило, користувачі навчальних закладів, великих організацій, фірм, підприємств приєднуються до мережі Інтернет через свою локальну мережу. Один з комп'ютерів локальної мережі виконує функції проксісервера (англ. проху – представник, довірена особа), а саме управління локальною мережею та підтримки зв'язку між комп'ютерами користувачів та мережею Інтернет. У технічному плані для приєднання до комп'ютера провайдера потрібні ПК, відповідне програмне забезпечення і модем – пристрій, що перетворює цифрові сигнали від комп'ютера на сигнали для передачі телефонними каналами зв'язку і навпаки. Комп'ютер провайдера може виконувати функції хосту або звертатися до потужніших хостів, щоб отримати доступ до глобальних ресурсів мережі Інтернет, через канал передачі даних з високою пропускну здатністю – магістраль. Крім функцій мережі, хост може виконувати завдання користувача (програми, обчислення). Деякі хости можуть виконувати функції шлюзів. Роль шлюза між мережею Інтернету і локальними мережами відіграє проксісервер. Доступ до Інтернету буває двох видів:

1. On-line – доступ до будь-яких ресурсів мережі в режимі реального часу.
2. Off-line – доступ відбувається лише з метою приймання або передавання даних, а обробка інформації здійснюється на локальному комп'ютері.

З розвитком глобальної мережі з'являються нові сервіси і відповідні їм протоколи, змінюючи її вигляд і стрімко розширюючи коло користувачів. Для того, щоб використовувати будь-який сервіс мережі Інтернет, необхідно встановити на ПК клієнтську програму, яка здатна працювати за протоколом цього сервісу. Деякі клієнтські програми входять до складу операційних систем, а також до складу програм-переглядачів (браузерів), за допомогою яких можна переглянути сторінку в мережі Інтернет. Для кожного типу ресурсу мережі Інтернет є власний протокол доступу [3, 4], наприклад:

- FTP (File Transfer Protocol) – передача файлів;
- HTTP (Hypertext Transfer Protocol) – передача гіпертексту;
- SMTP (Simple Mail Transfer Protocol) – передача поштових повідомлень;
- SNTP (Simple Network Time Protocol) – синхронізація часу (враховуючи ПК) в мережі або у світовому масштабі.

Для того, щоб комп'ютер-приймач міг визначити, для якої програми призначена інформація в пакеті, до адреси пакета додається ще одне число (порт), що викликає потрібну програму.

Огляд літературних джерел

Під час взаємодії клієнта з сервером клієнт і сервер – активні абстрактні об'єкти; клієнт запитує, а сервер формує відповідь за допомогою вказаного модульного інтерфейсу. Запит клієнта використовується для звернення до програми, яка надає деякі послуги. Користувацькі запити використовуються для умовного позначення людини в структурі мережі.

Клієнтські та серверні процеси взаємодіють, посилаючи один одному повідомлення, у вигляді запитів і відповідей [5, 6]. Повідомлення – найменша одиниця даних, які можуть бути відправлені й отримані під час взаємодії клієнта з сервером через модульний інтерфейс.

Постановка задачі

Полягає у розробленні модульного інтерфейсу для клієнт-серверної взаємодії в глобальній мережі Інтернет.

Основні результати досліджень.

Модульний інтерфейс клієнт-серверної взаємодії складається з модулів, безпосередньо не пов'язаних з будь-яким конкретним обладнанням або програмним забезпеченням. Ці модулі можуть бути як активними, так і пасивними. Розглянемо активні модулі, які активує клієнт, коли формує запит до сервера. Основними частинами при роботі модульного інтерфейсу під час взаємодії клієнта з сервером, тобто модулями, є:

- файл сервер, який обробляє логічні аспекти файла;
- сервер зберігання даних, який обробляє фізичні аспекти файла;
- фізичне сховище даних, яке забезпечує накопичування ЗМІ.

Коли клієнт формує запит до сервера через модульний інтерфейс, то під час проходження цього запиту по мережі клієнт активує модулі. На рис. 1 наведена структура модульного інтерфейсу на основі модулів. Як можна побачити на рис. 1, модульний інтерфейс складається з таких основних частин:

- 1) файл клієнта, у якому записані: фізична адреса конкретного комп'ютера в мережі та інформація щодо запитів;
- 2) файл сервера, у якому зберігається IP адреса сервера, а також відповідь на запити клієнта;
- 3) сервер зберігання даних, що використовується для зберігання запитів від клієнта, щоб у разі виникнення схожого запиту відповідь вже була сформована;
- 4) фізичне сховище даних – апаратна частина модульного інтерфейсу, яку можна замінювати у разі необхідності.

Файл клієнта, що ідентифікує користувача в мережі, пов'язаний з управлінням файлами системи та їх конвертуванням, якщо потрібно користувачу, а також з файлами серверів. Ім'я сервера, який забезпечує збереження ідентифікатора (ID).

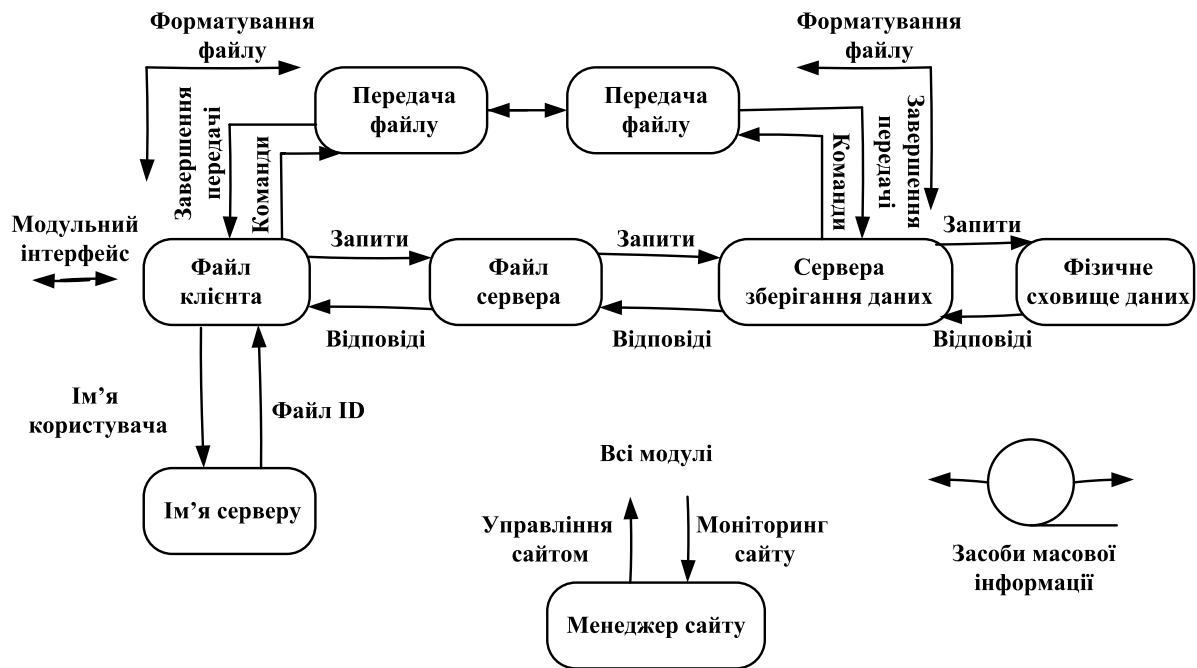


Рис. 1. Структура модульного інтерфейсу

Маючи основні частини модульного інтерфейсу, можемо тепер розглянути його приблизну роботу. Під час клієнт-серверної взаємодії клієнт посилає запит до сервера, і саме їхня взаємодія відбувається через модульний інтерфейс. Модульний інтерфейс з вигляду – це інтернет-сайт, за яким веде нагляд менеджер сайту. Своєю чергою, менеджер сайту виконує функції управління і моніторингу сайту, тобто безпосередньо може проглядати, активні чи не активні модулі й коли їх потрібно активувати для коректної взаємодії клієнта з сервером. Також менеджер сайту контролює операції, збір статистичних даних і формує запити до бази даних та здійснює контроль за роботою сайту. Модульний інтерфейс необхідний, щоб краще контролювати і зрозуміти взаємодію клієнта з сервером, в деяких випадках можна її прискорити. З'ясувавши структуру модульного інтерфейсу, розглянемо його роботу.

Тепер ми можемо безпосередньо описати роботу та вигляд модульного інтерфейсу з активними модулями (рис. 2).

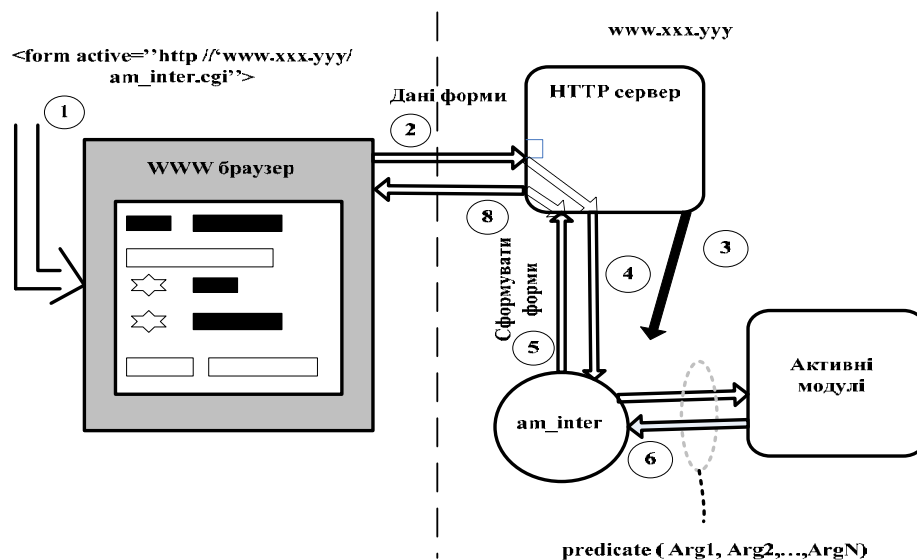


Рис. 2. Вигляд модульного інтерфейсу на основі активних модулів

На стороні користувача модулі матимуть вигляд браузера з шаблонами, у відповідність на стороні сервера ставиться написана певною мовою програмування програма з функціями і аргументами. Ці модулі форми модулів активуються відразу, коли надходить запит від клієнта до сервера.

Модульний інтерфейс просто передає вигляд матеріалу, отриманого з сервера (4) запущеного додатка (5), а потім скеровує вихід з програми (6) на сервер перед завершенням, а сам додаток продовжує працювати. Модульний інтерфейс може бути просто сценарієм, а сама програма буде складена, як зазвичай.

Коли клієнт взаємодіє з сервером через модульний інтерфейс, активуються модулі. Активним модулем може бути активний об'єкт, якщо модульність здійснюється через об'єкти [7–9]. Компіляція активного модуля створює виконуваний файл, який, працюючи, відіграє роль сервера для ряду взаємодій між клієнтів з одним сервером, які є предикатами, що експортуються модулем. Взаємодію клієнта з сервером через експортовані активні модулі можна підтримати інший клієнт в глобальній мережі, просто «завантаженням» модуля, через модульний інтерфейс, і відповідно це буде «віддаленою взаємодією». Ідея полягає в тому, що процес завантаження активного модуля не вимагає передавання коду, а просто використання його модульним інтерфейсом під час взаємодії клієнта з сервером через глобальну мережу. Крім того, модульний інтерфейс з використанням активного імпортованого модуля використовує його так само, як і будь-який інший модуль, крім того, що він використовує «use_active_module» замість «use_module». Крім того, активний модуль має адресу (мережеву адресу), яка має бути відома для того, щоб використовувати його. Адреса може бути оголошена активним модулем при запуску за допомогою файла або ім'я сервера (що містить в собі інший активний модуль з фіксованим адресою) [10].

Висновки. Отже, розроблений модульний інтерфейс, який працює під час клієнт-серверної взаємодії, дає змогу контролювати цю взаємодію. Зроблено також спроби використати конфігурацію активних модулів для підвищення ефективності і швидшої роботи клієнта з сервером через глобальну мережу, що дало змогу суттєво коректувати клієнт-серверну взаємодію.

1. Липаев В.В. *Обеспечение качества программных средств. Методы и стандарты.* – М.: Синтез, 2001. – 246 с. 2. Тамре Л. *Введение в тестирование программного обеспечения.* – М.: Вильямс, 2003. – 368 с. 3. Chris Giametta *Pro Flex on Spring.* 2009. – 445 p. 4. Бормотов С.В. *Системное администрирование на 100%.* – СПб.: Питер, 2006. – 256 с.: ил. 5. Роберт Дж. Оберг *Технология СОМ + Основы и программирование = Understanding and Programming СОМ+: A Practical Guide to Windows 2000 First Edition.* – М.: Вильямс, 2000. – 480 с. 6. Поляк-Брагинский А.В. *Администрирование сети на примерах.* – СПб.: БХВ-Петербург, 2005. – 320 с.: ил. 7. Макгрегор Дж., Сайкс Д. *Тестирование объектно-ориентированного программного обеспечения.* – К.: Диасофт, 2002. – 432 с. 8. Alex WebKpacKer *Быстро и легко. Хакинг и антихакинг: защита и нападение: учеб. пособие.* – М.: Лучшие книги, 2004 – 400 с.: ил. 9. Татарчук М. І. *Корпоративні інформаційні системи: навч. посіб.* – 2005. – 245 с. 10. Кеннет Г. *Основы сетей Windows.* – К.: Диалектика.