

ОСОБЛИВОСТІ РОЗРОБКИ СЕРВЕРНИХ ДОДАТКІВ КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ

Стаття присвячена спрощенню та пришвидшенню процесу розробки програм-серверів клієнт-серверної архітектури. Розроблений універсальний інтерфейс оптимізує процес створення серверів за рахунок необхідності розробки лише частини програми з безпосереднім описом протоколу прикладного рівня.

Стаття посвящена упрощению и ускорению процесса разработки программ-серверов клиент-серверной архитектуры. Разработанный универсальный интерфейс оптимизирует процесс создания серверов за счет необходимости разработки только части программы с непосредственным описанием протокола прикладного уровня.

Вступ

Клієнт-серверна архітектура постає домінуючою технологією при створенні розподілених мережеских [програм](#), забезпечує взаємодію та обмін даними між розробленими програмними продуктами. Сервери і клієнти такої архітектури є незалежними один від одного і функціонують паралельно [1]. Відсутність жорсткої прив'язки між клієнтами та серверами обумовлює можливість ситуації, коли один сервер одночасно обробляє запити від різних клієнтів чи клієнт звертається по чергові до різних серверів.

Сервер у клієнт-серверній архітектурі, крім простого забезпечення одночасного доступу до даних, здатний ще виконувати низку спеціалізованих програм. До переваг клієнт-серверної архітектури відносять зменшення навантаження на клієнтські місця, що, в свою чергу, призводить до зменшення вимог до апаратно-програмного забезпечення

клієнтів та зниження вартісних показників системи в цілому. Тому клієнт-серверну архітектуру ще називають архітектурою з тонким клієнтом [2].

Популярність клієнт-серверної архітектури обумовлює необхідність оптимізації процесу створення серверів, що є актуальним завданням у програмуванні мережевих ресурсів.

Метою роботи є спрощення та пришвидшення процесу розробки програм-серверів клієнт-серверної архітектури за рахунок розробки спеціального інтерфейсу, що забезпечить автоматизацію процесу створення серверів. Об'єктом дослідження постають процеси клієнт-серверної архітектури. Предметом дослідження є засоби створення програм-серверів. Головним завданням вбачаємо розробку універсального інтерфейсу, який дозволить оптимізувати процес створення серверів, та дослідження його можливостей в ієрархії рівнів моделі OSI.

Розробка інтерфейсу для оптимізації процесу створення серверів

Дослідження коду відомих HTTP серверів Apache HTTP server та Nginx показали, що часто для розробки серверних програм клієнт-серверної архітектури (далі серверів) програмний код щоразу пишеться з початку або ж використовуються лише окремі частини існуючого коду в форматі бібліотек (.lib, .a) чи динамічних бібліотек (.dll, .so). Такий підхід до вирішення задачі написання серверів виправданий лише за умови розробки спеціалізованих серверів, які використовують складні налаштування на різних рівнях OSI моделі, або у випадку, коли серверу потрібний повний контроль над передачею даних на рівнях OSI, що можна контролювати програмно.

Розглянемо найпростішу схему роботи додатків клієнт-серверної архітектури (рис. 1) [3].



Рисунок 1 – Узагальнена схема роботи додатків клієнт-серверної архітектури

Тут блоки передачі/прийому і обробки даних є спільними для клієнта і сервера. Проте при розробці програм унікальним залишається переважно лише блок обробки даних. Тому можна зробити висновок, що для уніфікації, спрощення та пришвидшення процесу розробки серверів можливим є відокремлення всіх базових блоків. Для імплементації залишається тільки блок обробки даних.

З метою забезпечення високої ефективності процесу створення серверів було розроблено універсальний інтерфейс, що значно спрощує та пришвидшує процес розробки серверних додатків клієнт-серверної архітектури. Розроблений інтерфейс дозволяє здійснити безпосередній опис протоколу прикладного рівня, а саме перетворення та обробку прийнятих даних. Таким чином, програма виключає необхідність імплементації двох додаткових рівнів моделі OSI (рис. 2). Зліва на рисунку 2 наведено рівні OSI моделі для реалізації без розробленого інтерфейсу (позначені червоним квадратом), справа – з використанням інтерфейсу.

Модель OSI		Модель OSI	
Дані	Рівень	Дані	Рівень
Дані	Прикладний доступ до мережевих служб	Дані	Прикладний доступ до мережевих служб
Дані	Представлення представлення і кодування даних	Дані	Представлення представлення і кодування даних
Дані	Сеансовий керування сеансом зв'язку	Дані	Сеансовий керування сеансом зв'язку
Блоки	Транспортний безпечне та надійне з'єднання «точка - точка»	Блоки	Транспортний безпечне та надійне з'єднання «точка - точка»
Пакети	Мережевий визначення маршруту та IP (логічна адресація)	Пакети	Мережевий визначення маршруту та IP (логічна адресація)
Кадри	Канальний MAC та LLC (фізична адресація)	Кадри	Канальний MAC та LLC (фізична адресація)
Біти	Фізичний кабель, сигнали, бінарна передача	Біти	Фізичний кабель, сигнали, бінарна передача

Рисунок 2 – Порівняння процесів створення серверів клієнт-серверної архітектури: зліва – рівні OSI моделі для простої реалізації сервера (позначені червоним квадратом), справа – з використанням розробленого інтерфейсу

Створений інтерфейс дозволяє налаштування сервера на більш низькому рівні, проте дає змогу використовувати базові налаштування за замовчуванням.

Запропонований інтерфейс можна використовувати для розробки різноманітних інтернет-протоколів прикладного рівня. Для прикладу на базі розробленого інтерфейсу було створено спеціальний HTTP сервер та сервер для віддаленого ввімкнення комп'ютера – WoL (Wake on LAN – виведення комп'ютера з режиму сну за спеціальним сигналом з мережі).

Висновки

Використання розробленого інтерфейсу оптимізує, пришвидшує та спрощує процес розробки програм-серверів клієнт-серверної архітектури за рахунок необхідності розробки лише частини програми, що відповідає безпосередньо за обробку отриманих даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Пасічник В.В. Глобальні інформаційні системи та технології: моделі ефективного аналізу, опрацювання та захисту даних. Монографія / В.В.Пасічник, П. І. Жежнич, Р. Б. Кравець, А. М. Пелешишин, Д. О. Тарасов – Львів: Видавництво Львівської політехніки, 2006. – 348 с. ISBN: 966-553-578-1.
2. Маклаков С.В. ВРwin ERwin CASE-средства разработки информационных систем. – М.: Диалог-МИФИ, 2001. – 304 с.
3. Эммерих В. Конструирование распределенных объектов. Методы и средства программирования интероперабельных объектов в архитектурах OMG/CORBA, Microsoft/COM и Java/RMI / Пер. с англ. – М.: Мир, 2002. – 510 с.