

Мережеве програмування в середовищі Unix

Лабораторна робота №4

TCP клієнт-сервер з мультиплексуванням введення-виведення

Для реалізації одночасної комунікації з кількома клієнтами в одному сервері можна створювати додаткові процеси та/або потоки, але це є перекладання реалізації цієї одночасної комунікації на ядро. Ядро планує виконання процесів або потоків (залежить від реалізації підтримки багатопотокових програм, наприклад, планувальник у режимі N:1 мультиплексує виконання потоків користувача в контексті одного процесу). Зрозуміло, що створення додаткових процесів та/або потоків дозволяє використовувати кілька процесорів у системі, але наявних процесорів може бути значно менше кількості клієнтів, з якими необхідно комунікувати. Також система може мати невелику кількість мережових інтерфейсів у порівнянні з кількістю клієнтів, а один мережовий інтерфейс одночасно не може приймати або відправляти дані необмеженої кількості мережових з'єднань, часто мережовий інтерфейс дозволяє відправляти та приймати тільки один пакет за раз. Можна сказати, що «одночасна комунікація з кількома клієнтами» є умовною, у комп'ютері одночасно або паралельно працюють процесори, більшість пристроїв, які виконують різноманітні введення-виведення, можуть виконувати свої завдання асинхронно (самостійно) по відношенню до процесора, але розмір їх буферів обмежений. Нескінченно розпаралелювати введення-виведення в сервері шляхом створення процесів та/або потоків не має сенсу, таке масштабування може призвести до зниження продуктивності всієї системи.

Мультиплексування введення-виведення – це спосіб пасивного очікування можливості виконання потрібних дій (введення та/або виведення) відразу для кількох дескрипторів файлів. Якщо хоча б для одного із вказаних дескрипторів файлів можна виконати потрібну дію, тоді програма отримує цю інформацію. Хоча мультиплексування визначається для введення-виведення, цей спосіб також використовується для отримання інформації про статус виконання раніше розпочатих дій з дескрипторами файлів. Мультиплексування введення-виведення можна комбінувати зі створенням додаткових процесів та/або потоків, оскільки мультиплексування введення-виведення фізично не розпаралелює введення-виведення в сервері.

Ця лабораторна робота присвячена практичному використанню функцій POSIX для мультиплексування введення-виведення. Завдання полягає в розробці однопотокового сервера, який дозволяє одночасно працювати з кількома клієнтами за допомогою мультиплексування введення-виведення, використовуючи користувальницький протокол, визначений в лабораторній роботі №3.

Завдання на роботу

Розробити однопотоковий сервер, який виконує наступне:

1. Сервер підтримує аргументи командного рядка, визначені в лабораторній роботі №3. Також сервер підтримує аргумент командного рядка, який визначає максимальну кількість клієнтів, з якими сервер може одночасно працювати. Сервер не приймає нові TCP з'єднання після досягнення цього значення.
2. Сервер працює з клієнтами відповідно до користувальницького протоколу, визначеного в лабораторній роботі №3.
3. Сервер дозволяє одночасно працювати з кількома клієнтами за допомогою мультиплексування введення-виведення. Сервер послуговується системними викликами `select()` або `poll()` для мультиплексування введення-виведення.

4. Кількість даних, які сервер зчитує або відправляє одному клієнту під час виконання введення-виведення з ним, треба обмежити. Ця кількість задається в коді сервера константою, яка може мати значення 1 байт та більше. Тобто, якщо сервер отримав інформацію від ядра про можливість виконати введення або виведення для якогось дескриптора файлу сокета, тоді серверу дозволено відправити або отримати даних розміром не більше вказаної константи. Це обмеження дає змогу майже порівну розподіляти час роботи сервера для кожного клієнта, який потребує комунікації. Також невеликі значення цієї константи дозволяють імітувати проблеми з мережею та частково імітувати різну поведінку клієнтів.

Сервер не має завершувати своє виконання у випадку виникнення несистемної помилки.

Рекомендації для сервера такі самі, які були дані в лабораторній роботі №3.

Звіт

- Опис розроблених програм (за бажанням).
- Лістинг розроблених програм.
- Приклади використання розроблених програм.

Література

Література надана у вступі до лекцій.

Лекції 1, 2, 3, 4, 6.