Dalam pemrograman jaringan, salah satu yang penting adalah bagaiana kemampuan server process dalam melayani request dari client process. Oleh karena itu server di tuntut untuk dapat memilik kemampuan dalama menerima request dari client secara bersama (multitasking). Dalam prakteknya, saat server dan client saling berkomunikasi terdapat aturan yang mengaturnya. Yang biasa di sebut dengan protocol.

1. **Protokol**

Protokol adalah sistem peraturan yang memungkinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua komputer atau lebih. Aturan ini harus dipenuhi oleh pengirim dan penerima agar komunikasi dapat berlangsung dengan baik. Sederhananya, protokol adalah media yang digunakan untuk menghubungkan pengirim dan penerima. Protokol dapat diterapkan pada perangkat keras dan perangkat lunak. Jadi hampir semua komunikasi yang terjadi pada jaringan komputer pasti melibatkan protokol.

1. **Multithreaded Server**

Multithreaded Server adalah sebuah server yang memiliki lebih dari satu thread sehingga punya kemampuan untuk melakukan proses client sacara multitasking (bersama sama). Dalam membuat server yang dapat melakukan client processing secara multi threaded, pada dasarnya terdapat dua thread

* 1. **Main Thread**

Pada saat ada koneksi dari client melalui socket, thread ini akan membuatkan thread baru yang bertujuan khusus untuk client processing. setelah client processing thread terbuat, maka informasi dan variabel socket akan diberikan ke client processing thread tersebut, dan kemudian thread utama akan kembali ke fungsi utamanya yaitu menunggu koneksi baru.

* 1. **Thread Client Processing**

Pada main thread akan di buat thread khusus untuk melayani request dari client. thread ini akan mendapatkan variabel socket dan informasi socket dari main thread. thread ini akan menggunakan socket dengan port yang berbeda dengan main thread. Thread ini dibuat setiap kali ada request khusuus dari client yang akan di proses dan kemudian memberikan result yang akan di kembalikan ke client socket sebagai respon dari request.

1. **Struktur Thread**

Pembahasan ini menggunakan source code berikut

<https://github.com/rm77/progjar/blob/master/progjar4a/file_server.py>.

File ini berisikan contoh bagaimana membuat thread untuk server dalam dua bagian, yaitu main thread dan client processing thread. Dalam kasus ini, thread akan dibuat menggunakan

* + Library threading
  + Implementasi thread sendiri dengan melakukan subclassing pada class Thread, dengan ini, maka kita akan dapat lebih leluasa melakukan kontrol terhadap behaviour dari thread yang kita punyai

Pada dasar thread terdapat init dan run, pada main thread terdapat infinite loop yang dimana loopingnya terdiri dari listen, accept new, new thread, dan dispatch. Untuk client, proses loop akan selesai ketika connection sudah di close. Implementasi protocol terjadi pada saat process request.

1. **Spesifikasi Protokol**

Pendefinisian spesifikasi komunikasi antara client dan server process, yaitu

* Deskripsi dari protoko
* Bagaimana request dilakukan dan parameter apa sajakah yang dibutuhkan
* Bagaimana respon diterima (result)
* Apa sajakah request yang dapat dilayani

Spesifikasi ini akan diterapkan oleh server process untuk melayani request dari client process.

1. **Struktur Program**

Untuk memudahkan alur pembahasan, struktur program terbagi menjadi 3 file yaitu file interface, file protocol, dan file server.

* 1. **File interface**

Fungsi dasar dari operasi file server berupa operasi list dan get isi file. Program adalah fungsional logik dari operasi file server dan dapat di uji secara mandiri sehingga proses debugging dapat dilakukan. Menggunakan classing pada python, agar dapat mengelola lebih tertutup dan lebih jelas. Terdapat list untuk mendapatkan daftar seluruh file, menggunakan filelist dan dict OK atau error.

* 1. **File protocol**

Menjawab cara mengimplementasikan spesifikasi protokol dalam bentuk pemrosesan string. Program ini mempuyai dependensi dengan file interface. String yang diproses didapatkan dari socket di client processing thread lalu diproses dengan memanggil fungsi yang bersesuaian dengan interface. File protocol akan menghasilkan hasil string yang akan dikirimkan client kembali. Protocol menerjemahkan data yang masuk dari socket, kemudian mengirimkan ke dalam bentuk string sesuai spesifikasi protocol yang dibuat.

* 1. **File server**

Sebagai pintu masuk dari client. Terdapat class Server sebagai main thread dan class ProcessTheClient sebagai client processing thread. Kedua class termasuk subclassing dari library thread.

1. **Implementasi client**

Program client (https://github.com/rm77/progjar/blob/master/progjar4a/file\_client\_cli.py) berisikan contoh bagaimana melakukan implementasi untuk client. Dalam contoh ini, diberikan dua fungsi yang berbeda cara dalam menangani respon dari server. Tetapi, kedua fungsi tersebut memiliki fungsi yang sama untuk konek dan request ke server.