

Nama : Mochamad Fajar Rifqi Priyanto
Kelas : 4IA10
NPM : 50421820
Mata Kuliah : Pemrograman Jaringan

Dokumentasi Alur Kerja Program Transfer File

Program ini mengimplementasikan sistem transfer file client-server. Server memulai dengan membuat dan mendengarkan koneksi pada port tertentu. Klien, setelah dijalankan, meminta nama pengguna dan masuk ke dalam loop utama untuk menerima perintah. Interaksi dimulai ketika klien memasukkan perintah, yang kemudian diproses untuk menentukan operasi yang akan dilakukan. Operasi-operasi ini mencakup menampilkan daftar file di server (ls dan IndexGet), mengunggah file ke server (FileUpload), mengunduh file dari server (FileDownload), dan memverifikasi integritas file (FileHash). Transfer data dapat dilakukan melalui TCP untuk keandalan, di mana panjang pesan dikirimkan sebelum data, atau melalui UDP untuk kecepatan, meskipun dengan potensi kehilangan paket. Klien juga mengelola cache lokal untuk menyimpan file yang diunduh, dengan logika penggantian LRU untuk mengelola ukuran cache. Selama operasi file, server melakukan tugas-tugas seperti menghitung hash MD5, mengirimkan ukuran dan waktu modifikasi file, serta membaca dan menulis data ke dan dari file. Penanganan kesalahan dasar diimplementasikan di kedua sisi, seperti pengecekan keberadaan file dan penanganan kesalahan socket. Loop utama klien berlanjut hingga perintah quit dimasukkan, yang mengakhiri koneksi dan melakukan pembersihan.

Program ini menjelaskan alur kerja program transfer file client-server. Klien berinteraksi dengan server untuk melakukan berbagai operasi file, termasuk menampilkan daftar file, mengunggah, mengunduh, dan memverifikasi integritas file.

1. Inisialisasi

- **Server (server.py):**
 - Membuat *socket* dan menunggu koneksi masuk pada host (HOST) dan port (PORT) yang ditentukan.
 - Mendefinisikan fungsi-fungsi untuk:
 - md5sum(): Menghitung hash MD5 dari sebuah file.
 - send_one_message(): Mengirim data dengan panjang yang ditambahkan di depan.
 - recvall(): Menerima sejumlah byte tertentu.
 - recv_one_message(): Menerima pesan dengan awalan panjang.
- **Klien (client.py):**
 - Meminta pengguna untuk memasukkan nama mereka.
 - Memasuki loop untuk terus menerima dan memproses perintah pengguna.
 - Mendefinisikan fungsi-fungsi untuk:
 - IndexGet(): Menangani perintah IndexGet untuk mengambil daftar file berdasarkan kriteria.
 - listdirServer(): Menangani perintah ls untuk menampilkan daftar file di server.

- upload(): Menangani perintah FileUpload untuk mengirim file ke server.
- cacheshow(): Menampilkan isi cache di sisi klien.
- cacheverify(): Memeriksa dan memperbarui cache di sisi klien.
- download(): Menangani perintah FileDownload untuk menerima file dari server.
- md5sum(): Menghitung hash MD5 dari sebuah file.
- send_one_message(): Mengirim data dengan panjang yang ditambahkan di depan.
- recvall(): Menerima sejumlah byte tertentu.
- recv_one_message(): Menerima pesan dengan awalan panjang.
- Verify(): Menangani perintah FileHash verify untuk memverifikasi satu file.
- Checkall(): Menangani perintah FileHash checkall untuk memverifikasi semua file.
- FileHash(): Memilih antara Verify atau Checkall.
- quit(): Menangani perintah quit untuk mengakhiri koneksi.

2. Loop Pemrosesan Perintah (Klien)

- Klien meminta pengguna untuk memasukkan perintah.
- Klien menyimpan historylist untuk menyimpan riwayat perintah.
- Klien mengurai perintah dan memanggil fungsi yang sesuai.

3. Operasi File

- **ls (Daftar File):**
 - Klien mengirim perintah ls ke server.
 - Server mengambil daftar file di direktori saat ini.
 - Server mengirimkan daftar tersebut kembali ke klien.
 - Klien mencetak daftar file.
- **IndexGet (Dapatkan Index File):**
 - Klien mengirim perintah IndexGet dengan opsi (shortlist, longlist) dan kriteria (misalnya, rentang tanggal, string pencarian).
 - Server memproses perintah IndexGet, berpotensi menggunakan perintah shell find dan lainnya untuk memfilter file.
 - Server mengirimkan daftar file yang dihasilkan (dengan detail seperti ukuran, waktu modifikasi, dll.) kembali ke klien.
 - Klien menampilkan informasi file yang diterima.
- **FileUpload (Unggah File):**
 - Klien mengirim perintah FileUpload, termasuk nama file dan mode transfer (TCP atau UDP).
 - **Mode TCP:**
 - Klien membuka file dan mengirimkan isinya ke server menggunakan send_one_message().
 - Server membuat file baru dengan nama yang diberikan dan menulis data yang diterima.
 - **Mode UDP:**
 - Klien membuka file dan mengirimkan isinya dalam bentuk potongan menggunakan paket UDP.
 - Server menerima paket UDP dan menulis data ke file baru.

- **FileDownload (Unduh File):**
 - Klien mengirim perintah FileDownload, termasuk nama file dan mode transfer (TCP atau UDP).
 - Server memeriksa apakah file tersebut ada. Jika tidak, server dapat mengirimkan kesalahan.
 - **Mode TCP:**
 - Server mengirimkan ukuran file, waktu modifikasi, dan hash MD5 ke klien.
 - Server membuka file dan mengirimkan isinya ke klien.
 - Klien membuat file baru dengan nama yang diberikan dan menulis data yang diterima.
 - **Mode UDP:**
 - Server mengirimkan ukuran file, waktu modifikasi, dan hash MD5 ke klien menggunakan paket UDP.
 - Server membuka file dan mengirimkan isinya dalam bentuk potongan menggunakan paket UDP.
 - Klien menerima paket UDP dan menulis data ke file baru.
- **FileHash (Verifikasi Integritas File):**
 - Klien mengirim perintah FileHash dengan verify (untuk satu file) atau checkall (untuk semua file).
 - **verify:**
 - Server menghitung hash MD5 dari file yang ditentukan dan mengirimkannya, bersama dengan waktu modifikasi terakhir, ke klien.
 - Klien menghitung hash MD5 dari salinan lokal file tersebut.
 - Klien membandingkan hash server dan hashnya sendiri. Klien mencetak apakah keduanya cocok dan juga mencetak waktu modifikasi terakhir.
 - **checkall:**
 - Server melakukan iterasi melalui semua file di direktori.
 - Untuk setiap file, server menghitung hash MD5 dan mengirimkan nama file, hash, dan waktu modifikasi terakhir ke klien.
 - Klien menghitung hash MD5 dari salinan lokal setiap file (jika ada).
 - Klien membandingkan hash server dan hashnya sendiri untuk setiap file dan mencetak hasilnya.
- **Cache (Caching di Sisi Klien):**
 - Klien mengimplementasikan cache file sederhana.
 - **cacheverify:** Memeriksa apakah file yang diunduh ada di cache. Jika tidak, klien mengunduh file dan menambahkannya ke cache (penggantian LRU jika cache penuh).
 - **cacheshow:** Menampilkan file yang saat ini ada di cache.
- **read (Baca file):**
 - Klien membaca file di sisi klien dan mencetak konten ke terminal
- **quit (Keluar):**
 - Klien mengirimkan perintah quit ke server.
 - Server menutup koneksi dan, secara opsional, keluar.
 - Klien menutup koneksi dan dapat melakukan pembersihan (misalnya, menghapus direktori cache).