

Il progetto è composto da due componenti, il client e il server. Al client da linea di comando possono essere passate varie opzioni. Se è presente l'opzione -h il client stampa sullo standard output le istruzioni d'uso e termina (indipendentemente dalle altre opzioni eventualmente presenti). Se l'opzione -h non è presente allora è necessaria una ed una sola opzione -f seguita dal nome del socket su cui tentare la connessione, altrimenti il programma termina. Opzionalmente possono essere presenti altre opzioni (se non sono presenti il client tenta la connessione sul socket specificato e termina appena la connessione ha successo o quando scade il tempo di attesa per la connessione). Ogni opzione corrisponde a una richiesta al server, che potrà però essere costituita da più messaggi scambiati secondo il protocollo richiesta-risposta, ad esempio l'opzione -r file1,file2,file3 costituisce una singola richiesta ma col server vengono scambiati più messaggi (ciò assume rilevanza se è specificata l'opzione -p che specifica il tempo che intercorre tra due richieste). Poiché il server memorizza i file con il loro path assoluto il client qualora gli vengano forniti path relativi li traduce in path assoluti (basandosi sulla cwd del client) prima di inviare la richiesta al server. Per comunicare col server il client invia prima un messaggio di lunghezza nota dove specifica il tipo di operazione e la lunghezza del/dei messaggi seguenti, e poi invia al server gli argomenti necessari per svolgere l'operazione (ad esempio il path del file) in accordo con quanto specificato nel primo messaggio. Una volta effettuato l'invio il client si mette in ascolto di una risposta da parte del server che sarà costituita da un messaggio di lunghezza nota (nello stesso formato del primo messaggio inviato) dove sarà riportato l'esito dell'operazione e la lunghezza di eventuali messaggi successivi con i quali il server invierà se necessario ciò che era stato richiesto dal client (ad esempio il file in una richiesta di read). L'unica eccezione è la richiesta -R durante la quale (poiché il numero di file che verranno restituiti non è noto a priori) il server informerà il client, prima di inviare ogni file, della dimensione di quest'ultimo. Qualora fosse specificata l'opzione -d il cliente salverà il/i file ricevuti da server nella cartella specificata utilizzando solo il nome del file (e scartando quindi tutto il path), se quindi vengono restituiti dal server due file con lo stesso nome ma path diverso il primo dei due ad essere salvato verrà sovrascritto dall'altro (vengono sovrascritti anche eventuali file già presenti nella cartella con lo stesso nome di uno dei file restituiti dal server). La comunicazione del client col server è implementata utilizzando l'API fornita che traduce gli argomenti passati alle funzioni in messaggi che vengono poi scritti sul socket (se la connessione è aperta, altrimenti si ritorna un errore che verrà stampato sullo standard error se l'opzione -p è specificata).

Il server è composto da un thread master, n thread workers e un thread per la gestione dei segnali. Il file di configurazione (da passare a linea di comando all'avvio) deve specificare il numero di lavoratori (N_WORKER), la dimensione massima dello storage in bytes (MAX_STORAGE), il numero massimo di file (MAX_N_FILE), la dimensione della tabella di hash (si veda in seguito) per la memorizzazione dei file (N_BUCKETS) e il nome del socket (SOCKET_NAME), nel formato "<NAME> = <value>", uno per riga e nell'ordine specificato. Se si verificano errori nella lettura del file di configurazione o se il file non viene fornito vengono usati dei valori di default. La memorizzazione dei file nello storage avviene in memoria principale utilizzando una tabella hash che garantisce rapidità negli inserimenti e nelle ricerche.