

Progetto per il Laboratorio di Programmazione di Reti
A.A. 2023-2024

Marco Buda
Mat. 0001071464
marco.buda3@studio.unibo.it

Traccia 2: Web Server Semplice

HTT Pasta

Funzionamento del server:

Il server utilizza il modulo `socket` per implementare una socket TCP e il modulo `threading` per creare più thread che gestiscano concorrentemente le varie connessioni ai client.

Le funzioni principali del programma sono le seguenti:

- `main()`
Inizializza la server socket e la associa alla porta indicata dall'utente, oppure a quella di default (8080), verificando in entrambi i casi che essa non sia già in uso.
Resta poi in attesa di richieste di connessione da parte dei client, creando un thread per ciascuna di esse.
- `handle_request(...)`
Riceve la request dalla connessione con un client e invia una response appropriata.
In particolare esamina la prima riga della request (request-line) e trasforma l'URL specificato in un percorso valido per il file system tramite manipolazione di stringhe e funzioni del modulo `os.path` (ad esempio in ambiente Windows l'URL `/a/b/./c` verrà convertito nel percorso `a\c`).
Se tale percorso fa riferimento ad un file esistente, oppure ad una cartella contenente un file `index.html`, il server ne invierà il contenuto al client, restituendo status code 200 (OK).
In caso di errore, il server invierà una semplice pagina HTML che informi il client dell'accaduto.

Elenco dei possibili errori:

- **400 (Bad Request)**: La request-line non rispetta la sintassi Metodo URL-Richiesto Versione-HTTP.
- **505 (HTTP Version Not Supported)**: La request non utilizza il protocollo HTTP/1.1.
- **405 (Method Not Allowed)**: La request non utilizza il metodo GET.
- **403 (Forbidden)**: La request tenta di accedere a file al di fuori della cartella in cui è contenuto lo script. Per motivi di sicurezza questo viene impedito.
- **404 (Not Found)**: Non è stato trovato un file associato all'URL specificato.

Esecuzione del programma:

Per avviare il server è necessario avere installato Python 3.10+ ed eseguire il comando `python httpasta.py [port]`.

L'argomento `port` deve essere un numero intero compreso tra 0 e 65535 e permette di specificare la porta su cui aprire il server. Nel caso venga omissso verrà utilizzata la porta di default (8080).

La repository contiene una pagina di esempio che consente di verificare rapidamente il funzionamento del programma. Sarà sufficiente avviare il server sulla porta di default ed aprire sulla stessa macchina un browser da cui accedere all'URL `http://localhost:8080/example`.

Per interrompere l'esecuzione del server è sufficiente premere `Ctrl` + `Pausa Interr` (`Ctrl` + `Break` su tastiere non italiane).

Considerazioni aggiuntive:

Il programma è in grado di gestire anche URL che contengono una query o un fragment (ad esempio l'URL `/file?query#fragment` è considerato equivalente a `/file`).

Non è stata implementata, invece, la funzionalità di traduzione delle sequenze speciali come `%20` e simili poiché ritenuta secondaria.