

A.A. 2020/2021
Relazione di Progetto
Programmazione di Reti

Traccia 2

Author : Alessandro Pioggia

Matricola :0000941343

INDICE

1.Introduzione :

- 1.1 Enunciato traccia.
- 1.2 Descrizione introduttiva del progetto.

2.Descrizione tecnica e tecnologie utilizzate :

- 2.1 Moduli python utilizzati
- 2.2 Struttura applicazione

3. Guida all'uso

- 3.1 Avviare e testare l'applicazione

1.1 – Enunciato traccia

Si immagini di dover realizzare un Web Server in Python per una azienda ospedaliera.

I requisiti del Web Server sono i seguenti:

- Il web server deve consentire l'accesso a più utenti in contemporanea*
- La pagina iniziale deve consentire di visualizzare la lista dei servizi erogati dall'azienda ospedaliera e per ogni servizio avere un link di riferimento ad una pagina dedicata.*
- L'interruzione da tastiera (o da console) dell'esecuzione del web server deve essere opportunamente gestita in modo da liberare la risorsa socket.*
- Nella pagina principale dovrà anche essere presente un link per il download di un file pdf da parte del browser*
- Come requisito facoltativo si chiede di autenticare gli utenti nella fase iniziale della connessione.*

1.2 - KB 24 Hospital

Il Kobe Bryant 24 Hospital è un ospedale con sito a Los Angeles, istituito dalla famiglia Bryant, il quale scopo è di aiutare tutte le persone che hanno bisogno di curarsi nonostante non abbiano una buona assicurazione (molto importante negli US).

Il progetto è molto ambizioso, dunque è necessario un sito web in grado di ricevere opportune richieste da più utenti in contemporanea in modo da fornire le informazioni necessarie per usufruire a pieno del servizio.

A questo proposito si è deciso di implementare una applicazione TCP per la realizzazione di un HTTP server multithread.

2.1 - Moduli python utilizzati

- *sys*
- *signal*
- *http.server*
- *socketserver*

sys

Il modulo sys permette di accettare, modellare e gestire le informazioni passate utilizzando la linea di comando.

signal

Il modulo signal invece, consente di definire handlers personalizzati una volta recepito un segnale.(es. Interruzione da tastiera)

http.server

Probabilmente il modulo più significativo del progetto, ci permette di ricevere ed inviare richieste. Una classe, HTTPserver è una sottoclasse socketserver.TCPserver.

socketserver

Il modulo socketserver semplifica la realizzazione di web server. Si compone di 4 sottomoduli :

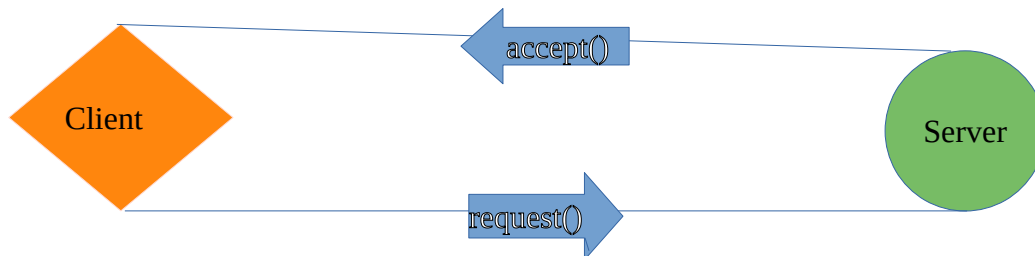
- *TCPserver*
- *UDPserver*

- *UnixStreamServer*
- *UnixDatagramServer*

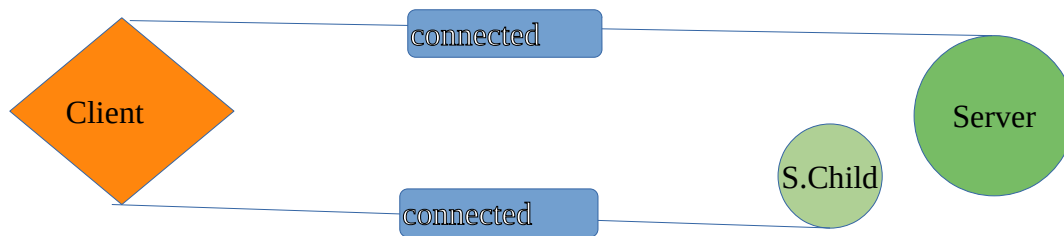
2.2 – Applicazione

fig1

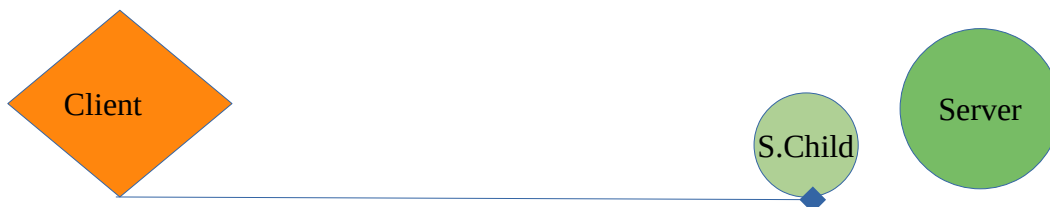
» richiesta di connessione ed accettazione da parte del server



» fork



» connessione col processo figlio attraverso porte effimere.



Per la parte dell'applicativo è fondamentale essere in grado di gestire molteplici richieste http(anche contemporanee) e per fare ciò è necessario comprendere ed implementare ciò che è rappresentato in **fig 1**.

1. Inizialmente il client avanza una richiesta di connessione attraverso il metodo : `connection()`, la quale verrà poi accettata dal server(`accept()`).
2. Una volta accettata la richiesta di connessione, il server, genera un processo figlio, il quale sarà anch'esso connesso con il client. Successivamente il Server padre(`parent`) effettuerà la chiusura delle porte effimere.
3. Nello step finale verranno chiuse le porte note da parte del child, rimarrà la connessione sulle porte effimere.

Con questa strategia, applicata n volte permetterà di gestire n connessioni contemporanee.

Nota esplicativa :

In una applicazione client-server i computer in comunicazione non hanno ruoli identici, ma sono definite, per così dire, due personalità, in cui uno (il client) svolge la funzione di richiedente, mentre l'altro (il server) risponde. Nel caso in cui anche il computer server si trovi nella necessità di effettuare delle richieste, allora esisterà al suo interno un diverso processo, avente funzione di client.

Avviare/testare l'applicazione

Clonazione in locale:

- Eseguire il download dello script, oppure la clonazione con git del progetto, presente al link :
<https://github.com/AlePioggia/HospitalWebServer-networking-project>.

Attivazione del web server:

- Navigare nella directory src/main
- Lanciare lo script con “python3 web_server_main.py”

Apertura della pagina web(lato client):

- Aprire il browser
- Digitare <http://localhost:8080/resources/>

Fine lettura.

Grazie

Alessandro Pioggia

