Progetto Web Server
Lorenzo Maiani

E' qui di seguito riportata il testo della traccia

Traccia 2

Traccia 2: Python Web Server

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.

Informazioni personali dello studente:

Nome: Lorenzo

Cognome: Maiani

Matricola : 915452

Specifiche di progetto

Il progetto ha l'obbiettivo di creare un Web Server in Python, con possibilità di accessi multipli in parallelo da parte di utenti diversi.

Il server gestisce le connessioni attraverso il metodo:

```
socketserver.ThreadingTCPserver(("",port), http.server.SimpleHTTPRequestHandler)
```

tale funzione offre tutte le componenti principali per una connessione TCP ad un HTTP server multi-thread. E' il metodo che controlla e abilita la generazione di thread figli per la gestione di più accessi da parte degli utenti in parallelo

Il parametro port può essere impostasto da tastiera (se l'esecuzione dello script python avvine da console) o è definita con il valore 8080. L'host prestabilito per la connessione è l'host da cui vieni eseguito il codice e perciò tale valore è impostato come 'localhost' o "".

http.server.SimpleHTTPRequestHandler esegue tutte le funzionalità che il client richede al server HTTP, tra le quali: la ricezione del messaggio, la decodifica, la ricerca del file da inviare, la codifica di quest'ultimo ed il suo invio.

La funzione CloseSignal_Handler si occupa di controllare e gestire la chiusura del server attraverso il Ctrl-C (terminazione attraverso l'interrupt da tastiera).

La funzione main() si occupa di gestire le principali funzionalità del codice quali:

- abilitazione del server a ricevere piu connessioni contemporaneamente con le funzioni

```
serverHTTP.daemon_threads = True
serverHTTP.allow_reuse_address = True
```

- gestione di interrupt da tastiera, richiamando la funzione CloseSignal_Handler:

```
signal.signal(signal.SIGINT,CloseSignal_Handler)
```

- infine pone il server in stato di attesa di richieste HTTP con il metodo:

```
serverHTTP.serve_forever()
```

Utilizzo

Per eseguire il codice occorre o lanciarlo da terminale o eseguirlo tramite l'applicazione Spyder.
Una volta eseguito il codice, occorrerà cliccare sul link qui di seguito http://localhost:8080 o aprire un browser/user_agent e copiare la stringa prima citata
e verrà aperta una pagina HTML con la quale poter interagire.

Riguardo al lato HTML:

Il server, come prima pagina, apre il file index.html che mostra i servizi principali di un'azienda ospedaliera, quali:

- un registro prenotazioni per i vaccini
- un registro per prenotazioni per tamponi

Per ognuno di questi servizi è presente un collegamento ad altre pagine html contenute nella directory file_html.

Infine è possibile cliccare sul pulsante presente sul lato sinistro della pagina principale per effettuare il download di un file pdf contente l'esito di un tampone fittizio.