

## Arch. di Reti / Reti di Calcolatori – 10 gennaio 2023

Si progetti un'applicazione Client/Server che, utilizzando le socket, aiuti un maker appassionato di domotica (e con un principio di disturbo ossessivo-compulsivo) a consultare le informazioni raccolte dai vari sensori installati presso la propria abitazione. L'applicazione deve presentare la seguente interfaccia:

***leggi\_dati\_sensori server porta***

dove **server** rappresenta il nome logico del Server e **porta** rappresenta il numero di porta del servizio.

Per prima cosa, il Client si deve interfacciare con il maker, da cui riceve (via terminale) il *tipo di sensore* (es., “pressione”, “temperatura”, “umidità”, ecc.), il *nome della stanza* in cui esso si trova (es. “cantina”, “soggiorno”, ecc.), e il numero *N* di letture di interesse. Il Client deve quindi trasmettere le informazioni al Server, che a sua volta dovrà reperire le *N* letture più recenti di dati ricevute dal tipo di sensore di interesse installato nella camera di interesse e restituirle al Client.

A questo proposito, si supponga che sul server sia in esecuzione un servizio automatico di raccolta dati che periodicamente (es. ogni 30 secondi) contatta tutti i vari sensori installati nella casa del maker per ottenere una nuova lettura da ciascuno di essi e salva le informazioni in una serie di file di testo all'interno del percorso `/var/letture_sensori`<sup>1</sup>, ciascuno dei quali conterrà le informazioni per una specifica stanza. Quindi, per esempio, le informazioni raccolte dai sensori installati in cucina saranno salvate nel file `/var/letture_sensori/cucina.txt`, elencate in ordine cronologico crescente (ovverosia dalla più vecchia alla più recente). Ciascuna riga di tale file conterrà le informazioni per una specifica lettura, ovverosia il timestamp di lettura, il tipo del sensore, il valore letto, ecc.

Una volta ricevute le informazioni dal Server, il Client le stampa a video e si mette in attesa della richiesta successiva. Il Client deve terminare quando l'utente digita “fine”.

**ATTENZIONE:** Si realizzino il Client e il Server in C, ma il Client deve essere realizzato anche in Java.

---

<sup>1</sup> Ovviamente nei PC del laboratorio non avrete permessi di accesso al percorso `/var`. Ai fini dell'esame, potete utilizzare un percorso all'interno della vostra home e lasciare un opportuno commento nella soluzione dell'esercizio (es. “uso il percorso `./letture_sensori` al posto di `/var/letture_sensori`”).

## Arch. di Reti / Reti di Calcolatori – 10 gennaio 2023

Design a Client/Server application which, using Sockets, helps a maker who is passionate about home automation (and has an obsessive-compulsive disorder principle) to consult the information collected by the various sensors installed in his home. The application must have the following interface:

***read\_sensor\_data server porta***

where *server* represents the logical name of the Server and *port* represents the port number of the service.

First of all, the Client must interface with the maker, from whom it receives (via terminal) the *type of sensor* (e.g., "pressure", "temperature", "humidity", etc.), the *name of the room* in which it is located (e.g. "cellar", "living room", etc.), and the number *N* of readings of interest. The Client must then transmit the information to the Server, which in turn must retrieve the most recent *N* readings of data received from the type of sensor of interest installed in the room of interest and return them to the Client.

In this regard, suppose that an automatic data collection service is running on the Server which periodically (e.g. every 30 seconds) contacts all the various sensors installed in the maker's home to obtain a new reading from each of them, and saves the information in a series of text files within the path */var/sensor\_readings*<sup>2</sup>, each of which will contain the information for a specific room. So, for example, the information collected by the sensors installed in the kitchen will be saved in the file */var/sensor\_readings/kitchen.txt*, listed in ascending chronological order (i.e. from oldest to most recent). Each line of this file will contain the information for a specific reading, i.e. the reading timestamp, the type of sensor, the value read, etc.

Once the information has been received from the Server, the Client prints it on the screen and waits for the next request. The Client must terminate when the user types "end".

**ATTENTION:** The Client and the Server are implemented in C, but the Client must also be implemented in Java.

---

<sup>2</sup> Obviously, you will not have access permissions to the */var* path on the laboratory PCs. For the purposes of the exam, you can use a path within your home directory and leave a suitable comment in the solution of the exercise (e.g. "I am using the *./sensors\_readings* path instead of */var/sensor\_readings*").