

Raport tehnic al proiectului IoT Simulator

Alina Popa - A3

Ianuarie 2019

1 Introducere

Internet of Things (abreviat și IoT; denumire din engleză, în traducere liberă Internetul Lucrurilor) este un concept ce presupune folosirea Internetului pentru a conecta între ele diferite dispozitive, servicii și sisteme automate, formând astfel o rețea de obiecte.

Proiectul IoT Simulator reprezintă o comunicare între un client și trei servere. Clientul va avea o interfață grafică și va juca rol de telecomandă pentru mai multe dispozitive, fiecare dispozitiv fiind un server. În cele din urmă, am ales ca dispozitive un bec, o telecomandă TV și un timer.

2 Tehnologii utilizate

Se va folosi TCP (Transmission Control Protocol), protocol ce efectuează o conectare virtuală între server și client. Acesta este optimizat pentru livrarea exactă a datelor și garantează livrarea unui flux de date fără duplicarea sau pierderea acestora.

Modul de operare al protocolului TCP implică existența a trei faze. În prima fază, conexiunea trebuie stabilită între componentele aplicației. Imediat ce conexiunea a fost realizată, urmează transferul datelor, iar conexiunea trebuie terminată în ideea de a închide calea virtuală și de a elibera resursele implicate în proces.

Pentru verificarea existenței datelor de intrare de la descriptori de citire multipli se va folosi `select()` care permite utilizarea apelurilor blocante pentru un set de descriptori (fișiere, pipe-uri, socket-uri,...) și suspendă programul până când descriptori din liste sunt pregătiți de operații de I/O.

3 Arhitectura aplicației

Aplicația va folosi o arhitectură de tipul client-server TCP concurent.

De asemenea, se va folosi modularizarea ce este un principiu de baza al programării în C și care constă în utilizarea pe scară largă a funcțiilor definite de utilizator, permițând astfel conectarea mai multor clienți.

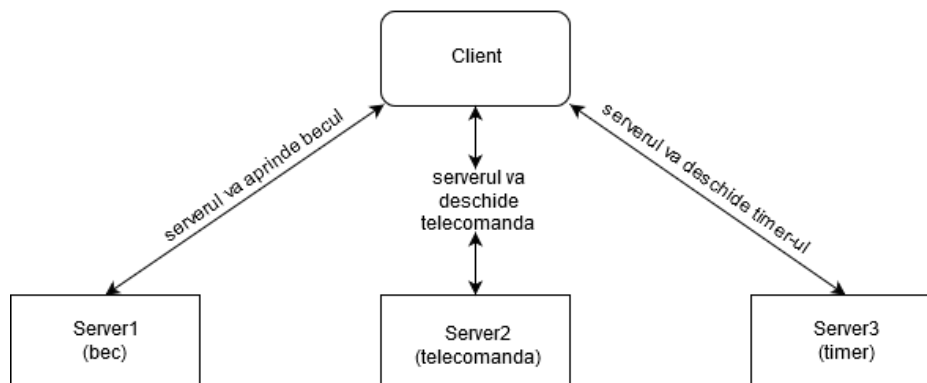


fig. 1

În fig. 1 s-a arătat cum se realizează comunicarea între client și cele 3 servere. Astfel, primul server, va fi capabil să stingă și să schimbe luminozitatea becului. Cel de al doilea va putea alege canalul și să închidă telecomanda, pe când cel de-al treilea va putea porni, opri și reseta timer-ul.

4 Detalii de implementare

Implementarea va constă în conectarea de către client la servere și transmiterea datelor printr-un TCP concurrent.

Pentru a se putea realiza comunicarea dintre server și client se va folosi canalul bidirecțional de comunicare `socket()`, astfel server-ul va putea să-i transmită clientului/clientilor datele necesare și respectiv, invers.

```

/* crearea unui socket */

if ((sd = socket (AF_INET, SOCK_STREAM, 0)) == -1)
{
    perror ("[server]Eroare la socket().\n");
    return errno;
}

```

La crearea interfeței s-a folosit Glade 3 ca mediu de dezvoltare în programarea bazată pe GTK+.

```

/*crearea unui window*/

gtk_builder_connect_signals (builder, NULL);
window = GTK_WIDGET (gtk_builder_get_object (builder, file2));
if (NULL == window)
{
fprintf(stderr, "Unable to file object with id interfata");
return window;
}

```

5 Concluzii

O îmbunătățire ar putea să o reprezinte un buton de întoarcere la meniul principal în client.

References

- [1] https://ro.wikipedia.org/wiki/Internet_of_Things
- [2] https://ro.wikipedia.org/wiki/Transmission_Control_Protocol
- [3] <http://www.scrigroup.com/calculatoare/c/Modularizarea-programelor75192.php>
- [4] https://stackoverflow.com/questions/8352027/gtk-timer-how-to-make-a-timer-within-a-frame?fbclid=IwAR3qcBXc1pkAs3V_NKwKtBk3ekgmSQRdTZ2XCxqR0wzwlBfQy8QnsbkJj4
- [5] https://stackoverflow.com/questions/34694383/gtk-glade-another-window-c?fbclid=IwAR0HVri642S2TX3GoBU60Y_j_Ustja_IY9dzz-tNg1gMRAtKzs5I03YpSps
- [6] https://stackoverflow.com/questions/26429173/getting-a-gtk-label-to-dislay-an-int-in-c?fbclid=IwAR2qYbmDBthwlo0-opco5vJlrUOmTfIEa_2DloPsvyBvSPlo8v9YQ-30k-Q