

CommunityBulletinBoard - RAPORT TEHNIC

Smântână Lavinia - grupa X4

Ianuarie 2024

1 Introducere

CommunityBulletinBoard este o aplicație client/server ce simulează un panou de anunțuri comunitare online, unde membri comunității pot posta evenimente, anunțuri sau mesaje de interes local. Serverul va pune la dispoziția clienților un număr minim de comenzi ce permit autentificarea, vizualizarea anunțurilor existente și adăugarea unui anunț nou. Aplicația va avea un mecanism de autorizare, astfel încât doar membri comunității vor putea accesa anunțurile și/sau adăuga un nou anunț.

2 Tehnologii utilizate

Pentru comunicarea dintre clienți și server vom folosi TCP, un protocol orientat pe conexiune care permite ca un flux de octeți trimiși de pe o mașină să ajungă fără erori pe orice altă mașină din rețea. Dacă în transmiterea datelor acestea sunt corupte sau pierdute, atunci TCP începe retransmisia, fără intervenția stratului de aplicare. În acest fel, se garantează că datele ajung la destinatar fără erori - lucru esențial în această aplicație deoarece vor fi transmise mai multe mesaje între server și clienți.

3 Arhitectura aplicației

Arhitectura aplicației **CommunityBulletinBoard** este fundamentată pe o structură client-server, unde serverul gestionează comunicarea și starea centralizată, iar clienții se conectează pentru a accesa și manipula informațiile comune din cadrul comunității. **Serverul** acționează ca punct central de control și stocare a datelor, primind cereri de la clienți și oferind răspunsuri corespunzătoare. **Clienții** se conectează la server pentru a realiza operații precum autentificarea, vizualizarea anunțurilor și adăugarea de noi anunțuri.

Comunicarea dintre server și clienți este realizată prin socket-uri și folosește un protocol simplu de trimitere și primire a datelor. Se utilizează apeluri de sistem precum send, recv pentru a transmite și recepționa informații între entități.

Autentificarea utilizatorilor este realizată prin verificarea credențialelor (username și parolă) într-un fișier de conturi. Pentru securitate, parola este transmisă într-un mod securizat utilizând funcția *getpass* pentru citirea acesteia.

Anunțurile sunt gestionate într-un fișier dedicat ("anunturi.txt"), unde sunt adăugate și/sau citite. Serverul este cel care trimite anunțurile către clienți la cererea acestora, iar noi anunțuri pot fi adăugate oricând de către clienți.

Aplicația oferă utilizatorilor o interfață simplă prin intermediul unui meniu intuitiv. Acesta oferă 3 opțiuni: vizualizarea anunțurilor existente, adăugarea unui anunț nou și posibilitatea de a se deconecta.

4 Detalii de implementare

Dupe ce se realizeaza login-ul, clientul conectat va avea de ales între cele 3 opțiuni: vizualizare anunțuri, adaugă anunț sau ieșire/deconectare.

```
106  int trimite_anunturi(int clientSocket) {
107
108     char c = '\0';
109     FILE *file = fopen("anunturi.txt", "a+");
110     if (file == NULL) {
111         perror("Eroare la deschiderea fisierului");
112         return 0;
113     }
114
115     char buffer[MAX_BUFFER_SIZE];
116     size_t bytesRead;
117
118     memset(buffer, '\0', sizeof(buffer));
119
120     c = fgetc(file);
121     while(c != EOF){
122         strncat(buffer, &c, 1);
123         c = fgetc(file);
124     }
125
126     if (send(clientSocket, buffer, sizeof(buffer), 0) == -1) {
127         perror("[server] Eroare la trimiterea anunturilor.\n");
128         return 0;
129     }
130
131
132     if (fclose(file) == EOF) {
133         perror("Eroare la inchiderea fisierului");
134         return 0;
135     }
136
137     return 1;
138 }
```

Figure 1: Funcția din server.c care afișează anunțurile existente clientului

```

140  ✓ int adauga_anunt(int client) {
141
142      FILE *file;
143      char title[20];
144      char details[100];
145
146  ✓      if ((file = fopen("anunturi.txt", "a+")) == NULL) {
147          perror("Eroare la deschiderea fisierului de anunturi");
148          return 0;
149      }
150
151      memset(title, '\0', sizeof(title));
152      memset(details, '\0', sizeof(details));
153
154      /* citim titlul si detaliile anuntului primite de la client */
155  ✓      if(read(client, title, sizeof(title)) < 0)
156      {
157          perror("[server] Eroare la citirea titlului.\n");
158          return 0;
159      }
160      title[strlen(title)-1] = '\0';
161
162  ✓      if(read(client, details, sizeof(details)) < 0)
163      {
164          perror("[server] Eroare la citirea detaliilor.\n");
165          return 0;
166      }
167      details[strlen(details)] = '\0';
168
169      fprintf(file, "\nANUNT: %s\nDETAII: %s", title, details);
170
171
172  ✓      if (fclose(file) == EOF) {
173          perror("Eroare la inchiderea fisierului");
174          return 0;
175      }
176
177      return 1;
178  }

```

Figure 2: Funcția din server.c care adaugă anunțul primit de la client

Atunci când clientul alege opțiunea 2, acesta va avea posibilitatea de adăuga un nou anunț prin introducerea unui titlu sugestiv și a unor detalii legate de anunțul menționat.

```
156  else if(option[0] == '2') {
157      // Adauga anunt
158      char title[20];
159      char details[100];
160
161      memset(title, '\0', sizeof(title));
162      memset(details, '\0', sizeof(details));
163
164      //citim si trimitem titlul
165      printf("[client] Introduceți titlul anunțului: ");
166      fflush(stdout);
167
168      if(read(0, title, 20) < 0 )
169      {
170          perror ("[client] Eroare la citirea titlului.\n");
171          return 0;
172      }
173      printf("[client] Titlul a fost introdus cu succes!\n");
174
175      if(send(sd, title, sizeof(title), 0) == -1)
176      {
177          perror ("[client] Eroare.\n");
178          return 0;
179      }
180
181      //citim si trimitem detaliile
182      printf("[client] Introduceți detaliile anunțului: ");
183      fflush(stdout);
184
185      if(read(0, details, 100) < 0 )
186      {
187          perror ("[client] Eroare la citirea detaliilor.\n");
188          return 0;
189      }
190      printf("[client] Detaliile au fost introduse cu succes!\n");
191
192      if(send(sd, details, sizeof(details), 0) == -1)
193      {
194          perror ("[client] Eroare.\n");
195          return 0;
196      }
197  }
```

Figure 3: Clientul trimite către server titlul și detaliile despre anunț

```

osboxes@osboxes:~/Desktop/proiect2023$ ./c3 127.0.0.1 2024
[client] Introduceți username-ul: info
[client] Username-ul a fost introdus cu succes!
[client] Introduceți o parolă:
[client] Ai introdus parolă cu succes!
[client] Te-ai conectat cu succes!

Meniu:
1. Vizualizeaza anunturile
2. Adauga anunt
3. Iesire

[client] Selectati o optiune din Meniu: 2
[client] Introduceți titlul anuntului: Concert de colinde
[client] Titlul a fost introdus cu succes!
[client] Introduceți detaliile anuntului: Va invitam la un concert de colinde pe 24 decembrie la Teatrul National!
[client] Detaliile au fost introduse cu succes!

[client] Selectati o optiune din Meniu: 1

----- ANUNTURI -----s

ANUNT: Concert de colinde
DETALII: Va invitam la un concert de colinde pe 24 decembrie la Teatrul National!

[client] Selectati o optiune din Meniu: 3
[client] Te-ai deconectat.

```

Figure 4: Exemplu folosind toate cele 3 comenzi

5 Bibliografie

- https://profs.info.uaic.ro/~busaco/teach/courses/net/docs/tcpip_net-ro.pdf.
- <https://itigic.com/ro/tcp-vs-udp-features-uses-differences/>.
- <https://ro.wikipedia.org/wiki/TCP/IP>.
- <https://profs.info.uaic.ro/~computernetworks/files/NetEx/S5/servTcpIt.c>.

c