Fakulta riadenia a informatiky, Žilinská univerzita v Žiline

Semestrálna práca

Programovacie jazyky pre vstavané systémy

Diskusné fórum

**5ZK011**

Peter Janták

Tomáš Zuzčák

Šk.rok 2017/2018

Obsah

[1. Úvod 3](#_Toc501295599)

[2. Programátorská príručka 4](#_Toc501295600)

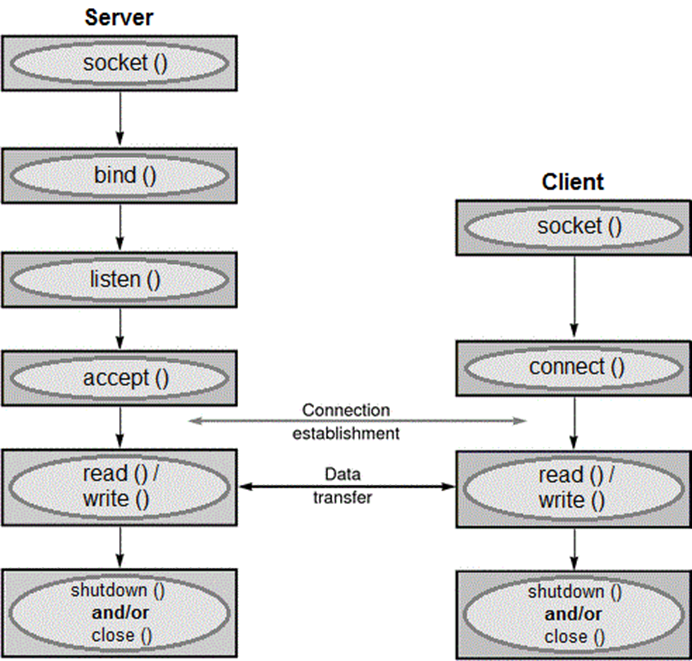
[3. Používateľská príručka 5](#_Toc501295601)

[4. Záver 6](#_Toc501295602)

# Úvod

Úlohou našej semestrálnej práce bolo naprogramovať diskusné fórum. Aplikácia pracuje na princípe klient – server ,kde klientov predstavujú používatelia ,ktorí sa pomocou svojej klientskej aplikácie pripájajú na server. Po pripojení klienta na server môže používateľ „četovať“ s ostatnými aktívnymi účastníkmi. Používateľ môže jednoducho z „četu“ odísť a to tak ,že sa odpojí zo servera.

V semestrálnej práci sme použili polia, štruktúry ,smerníky na funkcie ,sockety a vlákna.



Obrázok 0‑1 Komunikácia medzi klientom a serverom

# Programátorská príručka

Zdrojové kódy aplikácií server a klient sú rozdelené do .h a .c súborov.

Aplikácie sú rozdelené nasledovne:

* Server:
  + server.h
  + server.c
  + main.c
* Klient:
  + klient.h
  + klient.c
  + main2.c

V server.c sú nasledovné funkcie:

* void \*nacitaj - Metoda bežiaca vo vlákne, pre prípad ukončenia servera
* void vypisChybu – metóda na vypisovanie chybových hlášok
* bool pridajKlienta – slúži na pridanie nového klienta do listu klientov
* bool vymazKlienta – slúži na vymazanie klienta
* void posliVsetkym – pošle správu všetkým okrem seba
* void posliSebe – pošle správu len sebe a nikomu inému
* void ukazAktivnychKlientov – vypise zoznam aktívnych klientov (okrem seba)
* bool \*obsluhaKlienta – slúži na obsluhovanie klienta

V klient.c sú nasledovné funkcie:

* void vypisChybu – slúži na vypisovanie chybových hlášok
* void vypis – vypíše uvítací nápis po pripojení klienta na server
* void \*prichadzajuceSpravy – vypisuje správy ,ktoré prichádzajú klientovi a zároveň ukončuje klienta v prípade splnenia požiadaviek na odhlásenie.

Pre komunikáciu medzi klientom a serverom využívame sockety a vlákna ,ktoré sú použité v main.c (server) a main2.c (klient). Pre každého klienta je vytvorené jedno vlákno. Na vytváranie klientov používame štruktúru klient ,ktorú po vytvorení vkladáme do listu klientov (štruktúra Klienti). Tieto štruktúry sú v súbore server.h .

Štruktúra klient ma nasledovné premenné:

* struct sockaddr\_in addr
* int idSoket
* char meno[32]

Štruktúra Klienti má nasledovné premenné:

* KLIENT klienti[MAX\_POCET]
* int pocetKlientov

# Používateľská príručka

Pre správne fungovanie je potrebné najskôr spustiť aplikáciu server a následne aplikáciu klient. Keby boli aplikácie spustené v opačnom poradí ,tak by sa aplikácia klient ukončila s chybovou hláškou. Aplikácií server je potrebné definovať port na ktorom bude počúvať a komunikovať s klientom. Pri spustení klienta je potrebné zadať parametre: meno klienta, IP adresu (localhost) a port na ktorom bude klient komunikovať so serverom.

Spustenie servera:

***./server <port>***

Príklad: .*/server 10000*

Spustenie klienta:

***./klient <IP\_addr (localhost)> <port> <meno>***

Príklad: *./klient localhost 10000 User1*

Po pripojení klienta na server je klientovi vypísaná uvítacia správa na terminál. Následne môže klient prijímať správy od ostatných klientov. Keď klient odošle správu ,tak takáto správa je poslaná automaticky všetkým aktívnym klientom. Klient si taktiež môže nechať vypísať pomocnú nápovedu a to príkazom „!pomoc“. Po zadaní tohto príkazu sa mu na terminál vypíše nápoveda s nasledovnými možnosťami:

\*\*\*\*\*\*\*\*\*\*\*POMOCNIK\*\*\*\*\*\*\*\*\*\*\*

!ukazmeno - vase meno na chate

!aktivni - aktivni pouzivatelia

!pomoc - vyvolanie tohto pomocnika

!koniec - odhlasenie sa

Po zadaní jednotlivých príkazov sa vykonajú nasledovné akcie:

* !ukazmeno – vypíše meno klienta ,ktorý zadal tento príkaz
* !aktivni – vypíše zoznam aktívnych klientov ,ktorí sú pripojený na server
* !pomoc – vypíše nápovedu ,ktorú môžete vidieť vyššie
* !koniec – týmto príkazom sa klient odpojí od servera a už viac nie je aktívny

# Záver

Pri tvorení semestrálnej práce sme sa bližšie oboznámili s jazykom C a to hlavne s prácou so socketmi. Pri postupnej tvorbe semestrálne práce sme sa nezaobišli bez problémov ,ktoré sme museli následne odstraňovať ,respektíve odlaďovať aplikáciu tak ,aby fungovala korektne.