

**PRINCÍPY OPERAČNÝCH SYSTÉMOV**

# Semestrálna práca

## Server-client aplikácia: Chat v C

**Marián Hudec**

**PROGRAMATORSKÁ DOKUMENTÁCIA**

**Štruktúra**

Aplikácia je rozdelená dve hlavné časti klient a server ktoré medzi sebou komunikujú pomocou protokolu TCP. Je naprogramovaná v jazyku c. Ma 4 hlavné štruktúry: Message, User, ClientData a ThreadData.

Štruktúra Message si uchováva odosielateľa správy a text prijatej správy.

Štruktúra User uchováva meno a heslo užívateľa, priateľov (definovaných ako pole reťazcov), žiadosti o prijatie a zrušenie priateľstva. Takisto si uchováva konkrétny počet žiadostí o priateľstvo a zrušenie priateľstva, počet priateľov a číslo socketu, prostredníctvom ktorého komunikuje so serverom.

Štruktúra ClientData si uchováva meno a heslo užívateľa pripojeného na server, číslo socketu, prostredníctvom ktorého aplikácia komunikuje, poradové číslo klienta a stav registrácie/prihlásenia. Taktiež uchováva zoznam registrovaných a prihlásených užívateľov.

Štruktúra ThreadData si uchováva číslo vlákna, ktoré je pridelené na začiatku pripojenia klienta, stav vlákna, smerník na konkrétne klientské dáta a mutexy.

**Sockety, Vlákna, Synchronizačný problém**

Na komunikáciu medzi serverom a klientom sa používajú sockety. Server pri každom pripojení klienta vytvorí nové vlákno. Tam sú dve kritické oblasti:

1. **Inicializácia užívateľa** - pri pripojení užívateľ dostáva zdieľanú štruktúru ThreadData, ktorá obsahuje všetky potrebné dáta ako id vlákna a v rámci klientských dát aj ostatné dáta potrebné pre klienta ako id socketu, id klienta, zoznam registrovaných užívateľov atď. Keďže počet užívateľov je obmedzený, každý užívateľ potrebuje unikátny identifikátor a upravuje sa zdieľaná štruktúra. Inicializačná časť je synchronizovaná mutexom.
2. **Komunikácia klienta so serverom** – Keďže sú niektoré premenné ako napríklad počet registrovaných alebo prihlásených užívateľov zdieľané medzi viacerými vláknami, je potrebné v čase modifikácie daných operácii použiť mutex.

Každému pripojenému klientovi bez ohľadu na to, či je prihlásený alebo nie je pridelené osobitné vlákno aplikácie, v ktorom prebieha vzájomná komunikácia medzi klientom a serverom nezávisle od ostatných pripojených klientov.

**POUŽIVATEĽSKÁ DOKUMENTÁCIA**

**Spustenie servera**

Užívateľ spustí server, na ktorý sa následne môžu pripojiť klienti. Pri spúšťaní servera je potrebné ako argument serverovej aplikácie zadať číslo portu, na ktorý sa budú následne konkrétni klienti pripájať. Po úspešnom spustení servera aplikácia vypíše na konzolu „\*\*\* Socket and Bind OK“. V prípade neúspešného spustenia servera aplikácia vypíše na konzolu „\*\*\* Error binding socket address“ s konkrétnym popisom chyby, kvôli ktorej neprebehlo spustenie servera úspešne.

**Pripojenie klienta**

Klient sa pripojí na už existujúci server zadaním dvoch argumentov ako argumenty klientskej aplikácie: hostname a číslo portu servera. Po úspešnom spustení servera aplikácia vypíše na konzolu „\*\*\* Socket and Bind OK“, zobrazí úvodné menu a čaká na operáciu. V prípade neúspešného spustenia servera aplikácia vypíše na konzolu „\*\*\* Error connecting to socket“ s konkrétnym popisom chyby, kvôli ktorej neprebehlo spustenie servera úspešne.

**Ovládanie aplikácie**

Po úspešnom pripojení klienta sa zobrazí úvodné menu, ktoré čaká na výber operácie, ktorú chce klient vykonať. Operácia sa volí zadaním čísla operácie.

Obrázok, na ktorom je text

Automaticky generovaný popis

Ak užívateľ zvolí možnosť „Registrácia“, aplikácia ho následne vyzve na zadanie prihlasovacieho mena a hesla, ktoré bude následne používať pre prihlásenie sa do aplikácie pomocou možnosti „Prihlásenie“. Stav registrácie je vypísaný na konzolu.

Ak užívateľ zvolí možnosť „Ukončenie programu“, aplikácia odošle na server príkaz pre ukončenie komunikácie a program sa ukončí.

Ak užívateľ zvolí možnosť „Prihlásenie“, aplikácia vyzve užívateľa na zadanie prihlasovacieho mena a hesla a v prípade úspešného prihlásenia zobrazí hlavné menu užívateľa.

Obrázok, na ktorom je text

Automaticky generovaný popis

Užívateľ si následne môže zvoliť konkrétnu z daných operácií, ktorá sa má vykonať.

Ak užívateľ zvolí možnosť „Pridať priateľa“, aplikácia si vyžiada zo servera počet žiadostí a v prípade, že nejaké existujú, zobrazí ich užívateľovi a čaká na potvrdenie. V prípade potvrdenia žiadosti odošle na server potvrdenie a užívateľa pridá do zoznamu priateľov. Následne je klientovi zaslaný a zobrazený zoznam registrovaných užívateľov a užívateľ je vyzvaný na zadanie mena užívateľa, ktorého si chce pridať do priateľov. Stav zaslania žiadosti je vypísaný na konzolu.

Ak užívateľ zvolí možnosť „Odstrániť priateľa“, aplikácia si vyžiada zo servera počet žiadostí a v prípade, že nejaké existujú, zobrazí ich užívateľovi a čaká na potvrdenie. V prípade potvrdenia žiadosti odošle na server potvrdenie a užívateľa odstráni do zoznamu priateľov. Následne je klientovi zaslaný a zobrazený zoznam priateľov a užívateľ je vyzvaný na zadanie mena užívateľa, ktorého si chce odstrániť z priateľov. Stav zaslania žiadosti je vypísaný na konzolu.

Ak užívateľ zvolí možnosť „Zobraziť správy/Poslať správu priateľovi“, aplikácia si vyžiada zo servera počet správ a v prípade, že nejaké existujú, postupne ich vypíše na konzolu vrátane odosielateľa. Následne aplikácia vyzve užívateľa na zadanie mena užívateľa, ktorému chce užívateľ poslať správu. Následne je užívateľ vyzvaný na zadanie tela správy, ktorá je obmedzená na maximálne 160 znakov (po vzore SMS). Po potvrdení tela správy je celá správa odoslaná na server a stav odoslania správy je vypísaný na konzolu.

Ak užívateľ zvolí možnosť „Zrušiť účet“, aplikácia odošle na server žiadosť o zrušenie účtu a účet sa odstráni.

Ak užívateľ zvolí možnosť „Odhlásiť sa“, aplikácia odošle na server žiadosť o odhlásenie užívateľa a vráti sa späť do úvodného menu.