

1 Zadanie projektu

Úlohou tohoto projektu je vytvoriť implementáciu jednoduchého chatovacieho klienta, ktorý po pripojení na server prijíma a odosiela správy. Prijímané správy sú vypisované na štandardný výstup.

2 Spracovanie prepínačov

Implementácia klienta má prijímať dva povinné prepínače. Užívateľsky definovaná funkcia `parse_args()` načítava vstupné argumenty a podrobujú ich rade kontrolných podmienok. V prípade nesprávneho počtu argumentov alebo neznámeho argumentu, je program ukončený chybovým kódom `OPTS_ERR`. Pokiaľ načítanie prebehne bez komplikácií, `parse_args()` naplní príslušné premenné hodnotami parametrov a úspešne končí.

3 Nadviazanie spojenia so serverom

Nadviazanie spojenia so serverom je vykonávané pomocou vlastnej funkcie `set_connection()`. Táto funkcia pracuje s implicitne zadaným portom 21011 a IP adresou danou parametrom, ktorej platnosť overujeme pomocou vstavanej funkcie `gethostbyname()`. Okrem `gethostbyname()` pracuje `set_connection()` s radou ďalších funkcií definovaných v sieťových knižniciach, ktorých úlohou je správne naplniť potrebnú štruktúru, vytvoriť soket a spojiť sa so serverom. Pri akejkoľvek komplikácii je program ukončený s návratovým kódom `CONNECTION_ERR`.

4 Implementácia prijímania a odosielania správ

4.1 Odosielanie správ

Pre odosielanie neobmedzeného počtu správ bolo potrebné implementovať funkciu `send_to_server()`, ktorá pomocou vstavanej funkcie `getline()` a nekonečného cyklu číta užívateľský obsah vpísaný na štandardný vstup. Po príchode znaku nového riadku je kontext prepnutý do funkcie `send_minion()`. Táto užívateľsky definovaná funkcia zabezpečuje odoslanie jednej správy, pričom forma takto odosielanej správy závisí od hodnoty parametru `flag`. V prípade, že odosielame jednu z dvojice automaticky generovaných správ pre oznámenie o prihlásení alebo odhlásení, má parameter `flag` hodnotu `false`. Takáto hodnota spôsobí odlišné vyhodnotenie kontrolnej podmienky, text správy vo vnútri funkcie už nie je nijak upravovaný a na server je odoslaná jedna z dvojice správ `[username] logged in/out`. Pri hodnote `true` je text správy upravený do formy `[username]: [message]\r\n`. Po úspešnom ukončení `send_minion()` sa kontext vracia naspäť do funkcie `send_to_server()`, ktorá pokračuje ďalšou iteráciou nekonečného cyklu pre čítanie užívateľského vstupu.

4.2 Prijímanie správ

Prijímanie správ zo strany servera zabezpečuje funkcia `receive_from_server()`, ktorá v nekonečnom cykle prijíma pomocou vstavanej funkcie `recv()` jednotlivé časti správy a rovno ich vypisuje na štandardný výstup. Po vypísaní sú pred ďalším prechodom cyklu resetované obsahy všetkých potrebných premenných.

4.3 Funkcionalita

V prípade úspešného nadviazania spojenia so serverom program volá funkciu `send_minion()`, pomocou ktorej odosiela oznamovaciu správu `[username] logged in`. Následne je ako nová instancia triedy `thread` vytvorené vlákno, ktoré prechádza do kontextu funkcie `send_to_server()`, pričom rodičovské vlákno vchádza do funkcie `receive_from_server()`.

5 Ukončenie programu

Program sa ukončí po zachytení signálu `SIGINT` a odoslaní odhlasovacej správy `[username] logged out`. Za týmto účelom bola implementovaná funkcia `sigint_handler()`, ktorá odošle odhlasovaciu správu pomocou funkcie `send_minion()`, správne uzavrie soket a ukončí program s úspechom `ALL_OK`.