

1 Úvod

1.1 Rozbor zadání

Cílem projektu bylo vytvořit chatovacího klienta, který pomocí BSD socketů přijímá a odesílá zprávy na server.

2 Implementace

2.1 Zpracování parametrů

O zpracování vstupních parametrů se stará hlavní funkce `main()`. Funkce kontroluje, zda byl zadán správný počet a typ přepínačů. Jsou vyžadovány dva přepínače `-i ip_addr`, kde `ip_addr` je IP adresa serveru a `-u username`, kde `username` je uživatelské jméno. Pokud nebyly zadány správné vstupní parametry je program ukončen s návratovým kódem 1.

2.2 Nastavení socketu

Hlavní funkce `main()` provede pokus o vytvoření socketu a připojení k serveru. Pokud operace skončí neúspěchem je program ukončen s návratovým kódem 2. Následně je socket přepnut do neblokujícího stavu, v případě chyby je program ukončen s návratovým kódem 3. Po úspěšném nastavení socketu je na server odeslána první zpráva `username logged in`, pokud je prvotní komunikace neúspěšná, končí program s návratovým kódem 4. Následně program spustí další dvě vlákna popsaná níže.

2.3 Vlákno t1

Vlákno `t1` provádí v nekonečné smyčce tělo funkce `sendMessage()`, která se stará o načítání zpráv ze standardního vstupu a jejich odesílání na server. Kvůli korektnímu ukončování vláken, jsem se rozhodl pro neblokující načítání ze `stdin`. Pokud bych implementoval blokující načítání, hlavní vlákno by se při zachycení signálu muselo ukončit pomocí funkce `exit(0)`. Princip mého neblokujícího řešení je založen na funkci `inputAvailable()` zdroj: <http://stackoverflow.com/questions/717572/how-do-you-do-non-blocking-console-i-o-on-linux-in-c>. Funkce kontroluje, zda jsou na standardním vstupu připraveny nějaké znaky a má smysl vyvolat blokující operaci `getline()`. V případě problému s odesláním zprávy, je celý program ukončen s nenulovým návratovým kódem pomocí funkce `terminate()`.

2.4 Vlákno t2

Vlákno `t2` provádí v nekonečné smyčce tělo funkce `recvMessage()`, která se stará o přijímání příchozích zpráv a jejich výpis na standardní výstup. Díky neblokujícímu socketu není problém s korektním ukončením vlákna. V případě problému s přijímáním zpráv, je celý program ukončen s nenulovým návratovým kódem pomocí funkce `terminate()`.

2.5 Ukončení programu

Ukončení programu probíhá pomocí zaslání signálu `SIGINT`, které je odchyceno. Následně je odeslána poslední zpráva `username logged out` a sdílená proměnná `run` je nastavena na hodnotu `false`, což zaručí ukončení cyklů ve vláknech. Hlavní vlákno počká na korektní ukončení vláken `t1` a `t2` a následně je program ukončen s návratovým kódem 0.