

OSRT dokumentácia

Michal Petkáč

December 2, 2019

1 Funkcia programu

Program je určený na uchovávanie a spracovanie záznamov o veku a plate.

2 Požiadavky

Program je určený pre OS Linux. Testovaný bol na OS Ubuntu 18.04.3 LTS. Zoznam použitých knižníc sa nachádza v súbore header.h.

3 Kompilácia

Kompilácia prebieha cez program make. Stačí do konzoly zadať príkaz make a program sa skompiluje. Na odstránenie súbrov vytvorených pri kompilácii slúži pravidlo clean vykonané príkazom make clean.

4 Spustenie

Program pozostáva z časti servera a časti klienta. Server sa spustí z terminálu príkazom ./server.

Klientská časť sa spúšťa príkazom ./client. Táto časť má viacero možností spustenia. V prípade spustenia bez argumentov, alebo s argumentom 1 sa spustí jeden klient, ktorý je ovládaný z konzoly. V prípade argumentov 2-4 sa spustí klient ovládaný z konzoly plus 1-3 automatický klienti, ktorých správanie je definované v súboroch client1 - client3.

5 Ukončenie

Pre správne ukončenie klientskej časti stačí zadať pri komunikácii -1, čo ukončí komunikáciu so serverom a zavrie sockety. Ošetrené sú aj prípady, keď je klient ukončený signálmi SIGINT, alebo SIGHUP.

Pre správne ukončenie činnosti servera treba poslať signál SIGINT. Keď server

príjme tento signál, najskôr ukončí všetky spustené serverové, aj klientské procesy, odstráni zdieľané pamäte a alokované premenné a nakoniec zatvorí socket a vypne sa.

6 Popis programu

6.1 Pripájanie klientov

Server pri spustení počúva na dopredu stanovenom porte na klientov. Pri pripojení klienta prebehne inicializačná komunikácia. Klient pošle serveru svoj PID. Server náje voľný port, otvorí ho a pošle klientovi číslo tohto portu. Server vytvorí pre klienta obslužný proces, zatvorí socket a čaká na ďalšie pripojenie. Klient následne komunikuje výhradne s novovytvoreným procesom.

6.2 Komunikácia s klientmi

Komunikácia medzi serverom a klientov prebieha na základe požiadaviek. Klient pošle serveru číslo požiadavky, ktoré je stanovené v súbore header.h. Server následne vie, ako sa má zachovať. Po spracovaní požiadavky odpovie server, či bola požiadavka správne spracovaná. Ak áno, server pošle klientovi 1, ak nie, tak 0.

6.3 Vnútná logika servera

6.3.1 Tabuľka spojení

V serveri je neustále spustený jeden proces s dvoma vláknami. Jedno vlákno čaká na prichádzajúce spojenia a druhé slúži na kontrolné výpisy. Pri pripájaní klientov sa pre každého vytvorí osobitný proces.

Každé spojenie server-klient je uložené v databáze, ktorú má k dispozícii hlavný proces servera. V nej je uložený PID klienta a PID serverového procesu, ktorý klienta obsluhuje. Tabuľka slúži najmä pri správnom ukončovaní serveru. Vtedy server ukončí všetky svoje procesy a klientským procesom pošle signál SIGHUP, ktorý im signalizuje, že končí komunikácia so serverom. V databáze je ošetrené aj to, keď sa klient sám odpojí a tým zanikne spojenie. V takomto prípade daný obslužný proces zapíše svoj PID na miesto prístupné hlavnému procesu a ukončí sa. Ukončením dostane hlavný (rodičovský) proces signál SIGCHLD a vie, čo má robiť.

6.3.2 Ukladanie záznamov

Záznamy sú uložené v zdieľanej pamäti prístupnej všetkým procesom servera. Na zdieľanú pamäť je vytvorený mutex semafor, ktorý zabráňuje súčasnému prístupu viacerých procesov. Zdieľaná pamäť má obmedzenú veľkosť definovanú makrom. Ak sa táto veľkosť dosiahne, nebude možné pridávať ďalšie záznamy, kým sa pamäť nevymaže.

6.3.3 Ukončenie spojenia serverom

Ak je klient pripojený na server, avšak nekomunikuje aktívne so serverom, server po stanovenom čase ukončí komunikáciu. Toto je ošetrované časovačom, ktorý má zadefinovanú periódu makrom. Tento časovač každú periódu skontroluje, či prebehla nejaká akcia zo strany klienta. V prípade neaktivity sa obslužný proces ukončí. Samozrejme, dá to vedieť aj klientskému procesu, ktorému zašle signál SIGHUP.

7 Zoznam súborov

7.1 Súbory s kódom (.c)

Súbor server.c obsahuje kód serveru.

Súbor client.c obsahuje kód klienta.

V súbore functions.c sa nachádzajú funkcie využívané aj serverom, ak klientom.

Sú to najmä funkcie na prácu so socketmi, zdieľanými pamäťami a časovačmi.

V súbore statistic.c sa nachádzajú funkcie používané serverom na prácu s databázou.

V súbore requests.c sa nachádzajú funkcie používané klientom na prijímanie a zobrazovanie informácií prijatých od servera.

7.2 Hlavičkové súbory (.h)

V hlavičkových súboroch, ktoré sú rovnomenné, ako súbory s kódom sa nachádzajú hlavičky funkcií. Je to z toho dôvodu, aby kompilácia každého súboru mohla prebiehať samostatne. Výstupné súbory sa potom zlinkujú a vytvorí sa výsledný súbor.

Špeciálnym súborom je súbor header.h, kde sa nachádzajú definície makier, include súbory a štruktúry používané v celom programe.

7.3 Iné súbory

Okrem vyššie spomenutých súborov sa používajú aj súbory client1 - client3. Tieto súbory definujú správanie klientov. Použité môžu byť súbory s indexami do 9 (client9), avšak za predpokladu, že sa v súbore header.h zmení konštanta MAX_BOT_CLIENTS na požadovanú veľkosť.

8 Používané funkcie

8.1 Súbor functions

Funkcia open_socket slúži na otvorenie socketu.

Funkcia connect_socket slúži na pripojenie s socketu.

Funkcie send_string a receive_string slúžia na posielanie, respektíve prijímanie stringov cez sockety

Funkcie send_int a receive_int slúžia na posielanie, respektíve prijímanie intov

cez sockety

Funkcia `new_socket` slúži na nájdenie voľného portu a následne vytvorenia socketu na danom porte.

Funkcia `create_shm` slúži na vytvorenie zdieľanej pamäte.

Funkcia `create_my_timer` slúži na vytvorenie timeru

Funkcia `set_my_timer` slúži na nastavenie timeru.

Funkcia `remove_connection` slúži na odstránenie pripojenia zo zoznamu pripojení

8.2 Súbor requests

Funkcia `add_record` slúži na pridanie záznamu

Funkcia `add_record_console` slúži na pridanie záznamu, ktorý bude zadáný na štandardný vstup

Funkcia `show_all` zobrazí všetky záznamy zo servera

Funkcia `show_single_record` zobrazí jeden záznam zo servera

8.3 Súbor statistics

Funkcia `mean` zobrazí priemerný plat

Funkcia `max_value` vráti najvyšší plat a ID záznamu

Funkcia `min_value` vráti najnižší plat a ID záznamu

8.4 Súbor server a client

V týchto súboroch sa nachádzajú najmä funkcie, ktoré reagujú na signály. Signály `sigint`, `sigquit` a `sighup` slúžia na správne ukončenie programov (zatvorenie socketov, odstránenie alokovaných premenných a podobne). Špeciálne funkcie sú `client` a `processing`. Funkcia `client` sa spustí v klientskom procese a komunikuje so serverom. Funkcia `processing` sa spustí ako proces servera a obsluhuje klientský proces.

Osobitná funkcia je tiež `status_thread` nachádzajúca sa v súbore `server`. Táto funkcia sa spustí v samostatnom vlákne a slúži na kontrolné a informačné výpisy počas behu servera.

9 Rozšírenia

Program bol písaný tak, aby sa nové funkcionality implementovali čo najjednoduchšie. Dodefinovať akúkoľvek matematickú, respektíve štatistickú operáciu s množinou záznamov nie je žiaden problém. Stačí zadefinovať jeho hodnotu cez makro (nie nutné, ale vrelo odporúčané), vytvoriť požiadavku klienta, spracovanie servera a dané funkcie zavolať v serveri a klientovi.

Počet klientov v podstate nie je limitovaný (je limitovaný makrom, ktoré sa však dá bez problémov zmeniť). Testovaný bol však maximálne pre 10 klientov. Pri vyšších počtoch, rozsiahlej databáze a náročných požiadavkách môže nastať spomalenie z dôvodu čakania na prístup k zdieľanej pamäti.

10 Stiahnutie

Kompletný program si môžete stiahnuť z github-u cez nasledovný link:
<https://github.com/pekacc/OSRT-zadanie>