Zadanie OSRT

Sortovanie poľa integerov

Program na zoradenie čísel od najmenšieho po najväčšie, klient zada ľubovoľné čísla v tvare [x,y,z,..] cez socket je pripojený k serveru, kde sa vykonavajú všetky operracie. Server je dimenzovaný na komunikaciu s viacerými klientami. Po pripojení klienta na port 7777 funkcia accept priradí na komunikaciu nový socket a program sa rozforkuje, kde parent process čaka na prijmanie ďalšieho klienta a spustenie časovača, ktorý každých 10 sekúndach nečinnosti na porte vypíše, že sa nikto nový nepripojil. Child process komunikuje s klientom, triedi jeho spravy. Klient môže dostať 3 druhy sprav.

Prvá je vyššie uvedene [x,y,z,..], kde ak pride tento druh správy začina sortovanie. Druhý druh je “:e” čo znamena, že klient sa z daneho portu odpojil takže child zavrie socket s daným klientom. Tretí druh spravy je defualtny, kde som demonstroval použivanie mutexu. Naspäť k prvému druhu sprav, vytvori sa nové vlakno kde prebieha sortovanie nasledovne, program z pola znakov vytvori pole integerov, to sa neskor posle do funkcie merge\_sort(), kde sa nasledne pomocou algoritmu merge sort zoradí od najmenšieho po najväčši. Po zoradení pole naspät konvertujeme do požadovaného tvaru a zašleme naspäť klientovi. Pri sortovani su použité pipy na vypis pola na obrazovku servera.

V druhej časti po prijatí spravy :e od klienta child process zavrie socket a vypíše na obrazovku ktorý klient sa odpojil.

Tretia časť je defualtna, čiže všetky ostatné varianty správ. Ak pride tento druh spravy na servery sa vytvoria 2 vlákna, ktoré inkrementuju zároveň jednú globálnu premennu o milion, čiže po skončeni by sa malo vypisať niečo v tvare 2000000 = 2000000, ale keďže bežia paralelne môže nastať chyba a pri vypíse vždy vypiše niečo akolo 1050000. Na plnu spoľahlivosť funkčnosti incrementácie je dobre pouziť semafory , v našom pripade špecialný druh semaforov mutex, ktorý sa globalne initializuje, a pred každou inkrementaciu sa lockne a po nej sa unlockne čo zamedzi tomu aby nastavali kolizie.

Na začiatku celého programu vytvarame jedno vlakno, ktoré slúži čisto len na prijatie spravy na strane servera, kde keď sa zada spravny druh spravy, tak vypne celý server.

V child processe v prvej časti pred funkciou accept spúšťa timer s časom 10 sekúnd, a ak sa za ten čas nepripoji žiaden uživateľ vypiše na obrazovku pomocou funkcie up, že sa nikto nepripojil. Funkciu up sme si namapovali pomocou signalov. Ak sa niekto pripojí do 10 sekúnd timer sa vymaže vytvori sa child process, a opät parent vytvorí timer a čaka na pripojennie ďalšieho klienta.

Niečo malo ku klientskej časti. V nej si vytvorime socket ktorý sa pripaja na localhost s portom 7777 a potom čaka len na vstup od uživatela, ktorý sa pošle cez TCP/IP na server a čaka odpoveď od servera ktorú vypiše na obrazovku. Ak klient zada :e komunikacia medzi serverom a klientom sa ukonči a program sa zavrie.

Na sortovaci algoritmus je použity merge sort, ktorý funguje nasledovne, pole vždy rozdeli na 2 polia na polovoci, cyklus sa opakuje pokiaľ každe pole nebude mať iba jeden prvok. Nasledne potom sa zovola funkcia merge, kde sa vždy susedné prvky porovnavaju a zoradzuju, napr. ak mame prvok 90 a druhy prvok je 18 tak sa vytovori array s prvkami 18, a 90 . V ďalšom kroku z dvoch dvoj prvkovych poli vznikne jedno štvorprvkové opäť zoradené. Cyklus sa opakuje pokial nenaplnime pole všetkymi prvkami.

Subory obsahuju aj makefile, pomocou ktorého keď do terminalu napíšeme make nám vytvori executable file zo servera a z klienta.

