Zadanie OSRT

Sortovanie pola integerov

Program na zoradenie cisel od najmensieho po najvacsi, klient uzivatel zada lubovolne cisla v tvare [x,y,z,..] cez socket je pripojeny k serveru, kde sa vykonavaju vsetky applikacie. Server je dimenzovany na komunikaciu s viacerymi klientami. Po pripojeni klienta na port 7777 funkcia akcept priradi na komunikaciu novy socket a program sa rozforkuje kde parent process caka na prijmanie dalsieho klienta a spustenie casovaca, ktory kazdych 10 sekundach necinnosti na porte vypise ze sa nikto novy nepripojil. Child process komunikuje s klientom, triedi jeho spravy. Klient moze dostat 3 druhy sprav.

Prva je vyssie uvedene [x,y,z,..], kde ak pride tento druh spravy zacina sortovanie. Druhy druh je “:e” co znamena, ze klient sa z daneho portu odpojil takze child zavrie socket s danym klientom. Treti druh spravy je defualtny kde som demonstroval pouzivanie mutexu. Naspat k prvemu druhu sprav, vytvori sa nove 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 zoradi od najmensieho po najvacsi. Po zoradeni pole naspat konvertujeme do pozadovaneho tvaru a zasleme naspat klientovi. Pri sortovani su pouzite pipy na vypis pola na obrazovku servera.

V druhej casti po prijati spravy :e od klienta child process zavrie socket a vypise na obrazovku ktory klient sa odpojil.

Tretia cast je defualtna, cize vsetky ostatne varianty sprav. Ak pride tento druh spravy na servery sa vytvoria 2 vlakna, ktore inkrementuju zaroven jednu globalnu premennu o milion, cize po skonceni by sa malo vypisat nieco v tvare 2000000 = 2000000, ale kedze bezia paralelne moze nastat chyba a pri vypise vzdy vypise nieco akolo 1050000. Na plnu spolahlivost funkcnosti incrementacie je dobre pouzit semafory , v nasom pripade specialny druh semaforov mutex, ktory sa globalne initializuje, a pred kazdou inkrementaciu sa lockne a po nej sa unlockne co zamedzi tomu aby nastavali kolizie.

Na zaciatku celeho programu vytvarame jedno vlakno, ktore sluzi cisto len na prijatie spravy na strane servera, kde ked sa zada spravny druh spravy, tak vypne cely server.

V child processe v prvej casti pred funkciou accept spusta timer s casom 10 sekund, a ak sa za ten cas nepripoji ziaden uzivatel vypise na obrazovku pomocou funkcie up, ze sa nikto nepripojil. Funkciu up sme si namapovali pomocou signalov. Ak sa niekto pripoji do 10 sekund timer sa vymaze vytvori sa child process, a opat parent vytvori timer a caka na pripojennie dalisieho klienta.

Nieco malo ku klientskej casti. V nej si vytvorime socket ktory sa pripaja na localhost s portom 7777 a potom caka len na vstup od uzivatela, ktory sa posle cez TCP/IP na server a caka odpoved od servera ktoru vypise na obrazovku. Ak klient zada :e komunikacia medzi serverom a klientom sa ukonci a program sa zavrie.

Na sortovaci algoritmus je pouzity merge sort, ktory funguje nasledovne, pole vzdy rozdeli na 2 polia na polovoci, cyklus sa opakuje pokial kazde pole nebude mat iba jeden prvok. Nasledne potom sa zovola funkcia merge kde sa vzdy susedne prvky porovnavaju a zoradzuju, napr. ak mame prvok 90 a druhy prvok je 18 tak sa vytovori array s prvkami 18, a 90 . V dalsom kroku z dvoch dvoj prvkovych poli vznikne jedno stvorprvkove opat zoradene. Cyklus sa opakuje pokial nenaplnime pole vsetkymi prvkami.

Subory obsahuju aj makefile, pomocou ktoreho ked do terminalu napiseme make nam vytvori executable file zo servera a z klienta.

