Gebze Technical University
Computer Engineering

CSE 344 Spring
System Programming

Final Proje Raporu

FERHAT ŞİRİN
161044080

server.c

Çalıştırılması için gerekenler :
server.c **gcc -c server.c** ile derledikten sonra gcc **server.o -pthread -lm -o server** ile link yapılıp çalıştırılabilen server dosyası elde edilir. Program multithread olduğu için -pthread ve pow fonksiyonu kullanıldığı için -lm link sırasından eklenmelidir. Cos değeri taylor serisi ile hesaplanmaktadır. Dosyada makefile dosyası olduğu için **make** komutu ile program derlenip çalıştırılabilir hale getirilebilir. Server'ı çalıştırmak için port değeri, providerların olduğu dosya ve log dosya isimlerinin sırasıyla girilmesi gerekir. Örnek : **./server 5555 final.dat log.dat**
port değeri 5555 , final.dat provider değerilerini içerir ve çıktılar log.dat dosyasına kaydedilir.
make runServer ile server çalıştırılabilir.

Programın tasarımı :
Provider değerleri için verilen dosya '\n' newline karekteri baz alınarak parçalara ayrılıp isim, performance, time ve price değerleri alınıp struct Provider yapısına kaydedilir. **struct Provider** ile her bir provider yapısı temsil edilir. Provider yapısı oluşturulduktan sonra her biri için yeni bir thread oluşturulur ve **providerF** fonksiyonu çalıştırılır.

Her bir providerın çalışma süresi olduğu için bu süreleri tutan ve zamanı dolan providerları kapanmasını sağlayan ayrı bir thread oluşturulur ve **wakerF** fonksiyonu çalıştırılır. Bu thread süresi dolan proverların isOpen değerini değiştirir. IsOpen değeri 0 olan providerlar ellerindeki iş bittikten sonra kapanıcaktır.

Provider threadleri çalıştıktan sonra main thread'te soket açılıp bind ve listen yapıldıktan sonra client isteklerinin gelmesi beklenir. Gelen her bir istek için yeni bir thread oluşturulur ve **serverF** fonksiyonu çalıştırılır. ServerF ile client ve provider arasındaki iletişim sağlanır. Client'tan gelen **struct Client** yapısı okunur ve priority değerine göre açık ve boş olan provider'a gönderilir. **Cost (C) için en düşük price değeri quality (Q) için en yüksek performance değeri ve time (T) için en düşük çalışma süresine sahip olan provider seçilir eğer doluysa o değerlere en yakın diğer provider'a bakılır.**Provider cevabı hesapladıktan sonra **struct Answer** içine değerler kaydedilir ve client'a gönderilir. İletişim tamamlandıktan sonra o iletişimi sağlayan thread kapanır. Arka planda main thread sürekli yeni gelen istekleri toplar ve gelen her bir istek için serverF fonksiyonunu çalıştıran thread'i oluşturur.

Süresi dolan providerlar ellerindeki iş bitmeden kapanmazlar ve kendilerine yeni iş verilemez.

Sinyal gelmesi durumu:
providerF, serverF ve wakerF fonksiyonları ve dolayısıyla onları çalıştıran threadler pthread_sigmask ile SIGINT sinyaline karşı korumaya alınmıştır. Bu fonksiyonları çalıştıran threadler bu sinyalin gelmesi durumunda etkilenmeyecek ve ellerindeki işe devam edeceklerdir. Main thread ise bu sinyali aldığında artık clienttan gelen istekleri kabul etmeyi bırakacak ve programın düzgün bir şekilde kapanmasını sağlayacaktır. Main thread öncelikle wakerF fonksiyonuna SIGUSR1 sinyali göndererek artık elindeki işi bitirtikten sonra kapanmasını ister ve pthread_join ile kapanmasını bekler bu durumda açık kalan diğer provider threadlari main thread kapatacaktır. WakerF kapandıktan sonra mainThread serverThreadlerin ellerindeki işi bitirmesini bekler ve her biri için pthread_join'i çağırarak biten serverThread lerin kapanmasını garantiler böylelikle providerlar artık ellerinde iş olmuyacak ve kapanmaya hazır olucaklar. ServerThreadler kapatıldıktan sonra main thread açık kalan providerların isOpen değerini 0 olarak atar ve kapanmalarını pthread_join ile garantiler. Daha sonra her bir provider'ın istatistik değeri yazdırılır ve program kapatılır.

Tüm providerların süresi bittiği halde sinyal gelmemesi durumunda wakerF fonksiyonu main thread'a SIGINT sinyalini göndererek programın düzgün bir şekilde kapanması sağlanır.

client.c

Çalıştırılması için gerekenler :
client.c **gcc -c client.c** ile derledikten sonra **gcc client.o -o client** ile link yapılıp çalıştırılabilen client dosyası elde edilir. Dosyada makefile dosyası olduğu için **make** komutu ile program derlenip çalıştırılabilir hale getirilebilir. Client'ı çalıştırmak için isim, öncelik, hesaplanması istenen değer, adres ve port değerinin sırasıyla girilmesi gerekir.
Örnek : **./client ferhat C 45 127.0.0.1 5555**
isim ferhat, öncelik C (cost), hesaplanıcak değer 45, adres 127.0.0.1, ve port değeri 5555
make runClient ile client, command dosyasında yazan değerler ile çalıştırılabilir.

Programın tasarımı :
Komut satırından alınan değerler ile struct Client oluşturulur ve socket açılıp verilen adres ve port değerine bağlantı sağlanır. Struct Client sokete yazıldıktan sonra read() ile struct Answer değerinin gelmesi beklenir. Sonuç yazdırıldıktan sonra program kapatılır.