

Gebze Technical University
Computer Engineering

CSE 244
2017 Spring

FINAL REPORT

DEVRİM AKINCI
141044052

COURSE TEACHER: ERKAN ZERGEROĞLU

PROJEDE İZLEDİĞİM YOL

- Server-Client bağlantısını oluşturmak için her iki programda da socket yapısını oluşturdum.
- Server-Client yapısının Multiclient bir şekilde çalışabilmesi için clients programından girilen değer kadar thread oluşturdum ve her bir thread için socket oluşturdum.
- Server programında sırası ile socket-bind-listen fonksiyonlarını çağırarak socketi dinleme moduna aldım.
- Server'da client'dan gelen her bir request için bir tane thread oluşturdum.
- Request'leri accept fonksiyonu ile kabul ederken accept fonksiyonun arasına sem_wait ve sem_post fonksiyonlarını koyarak bir kritik alan oluşturdum.
- Aynı şekilde clients programındaki threadlerin kendine ait olmayan socketi kullanmalarına izin vermemek için connect fonksiyonun arasına semafor koydum. Böylece threadlerin kendine ait olmayan bölgeyi işgal etmesini engelledim.
- Daha sonra write ve read fonksiyonları ile socket üzerinde veri alışverişini sağladım.
- Son olarak her programın(server ve clients) log dosyalarını oluşturdum.

KARŞILAŞTIĞIM SORUNLAR

- Server programındaki port parametresi yetersizdi. Çünkü programların sokette haberleşme sağlayabilmeleri için aynı port üzerinde olmaları gerekmektedir. Bundan dolayı clients programına bir parametre eklemek zorunda kaldım.(Usage: ./clients port row col threadsize)
- P1,P2,P3 processlerinin aynı anda çalışması istenmiştir. Bunu sağlayabilmek için P1,P2 ve P2,P3 arasındaki shared memory'e bir şart koydum.Yazdı haberi için shared memory'e -22, okudu haberi için ise shared memroy'e -11 yazdım.
- P2'nin içerisinde üç thread olması ve bu üç threadin farklı yollar ile(SVD,QR,Pseudo-Inverse) $AX=B$ denklemini çözmesi istenmiştir. Fakat projenin bu kısmında zamanım yetmediği için x matrisini rastgele sayılar(1-10) ile oluşturdum ve hata kısmını bu rastgele oluşturulan matris ile hesapladım. Bundan dolayı programın log dosyasındaki hata çıktısının değeri beklenenden büyüktür.

PROGRAMIN COMPILE EDİLMESİ VE ÇALIŞTIRILMASI

Programı make komutu ile compile edebilirsiniz ve make clean komutu ile binary ve executable uzantılı dosyaları silebilirsiniz.

Çalıştırılma şekli;

./clients 8080 2 3 5

./server 8080 50

TEST

Test-1

./clients 9090 2 3 5
clients.log

```
THREAD ID:140091588011776
MATRIX-A
9.00 1.00
1.00 6.00
10.00 8.00
MATRIX-B
4.00
3.00
4.00
MATRIX-X
6.00
8.00
ERROR:97.7036
THREAD ID:140091604797184
MATRIX-A
8.00 10.00
2.00 3.00
8.00 1.00
MATRIX-B
10.00
9.00
1.00
MATRIX-X
10.00
3.00
ERROR:130.8587
THREAD ID:140091596404480
MATRIX-A
8.00 1.00
9.00 1.00
4.00 6.00
MATRIX-B
8.00
10.00
5.00
```

./server 9090 0
server.log

```
THREAD ID:140091596404480
MATRIX-A
8.00 1.00
9.00 1.00
4.00 6.00
MATRIX-B
8.00
10.00
5.00
MATRIX-X
THREAD ID:140091588011776
MATRIX-A
1.00 6.00
10.00 8.00
4.00 3.00
MATRIX-B
4.00
6.00
8.00
9.00
1.00
THREAD ID:140091604797184
MATRIX-A
8.00 10.00
2.00 3.00
8.00 1.00
MATRIX-B
10.00
9.00
1.00
MATRIX-X
10.00
3.00
```

Test-2

./clients 9090 4 6 15

./server 9090 0 ile çalıştırıldığında clients tarafından ctrl-c sinyali geldiğinde server hala çalışmaya devam etmektedir.

```
devo35@devrim:~/Masaüstü/Server-Client$ ./client 9090 4 6 15
^Cdevo35@devrim:~/Masaüstü/Server-Client$
```

```
devo35@devrim:~/Masaüstü/Server-Client$ ./server 9090 0
```