

# **System Programming**

## **BIL244**

**FINAL PROJECT**

**GÖZDE DOĞAN**  
**131044019**

Course Teacher: Erkan ZERGEROĞLU

## İçindekiler

1. System Requirements .....	3
1. Server.....	3
2. Client.....	3
2. System Solution Approach .....	4
1.Server .....	4
2.Client .....	5
3. Running and Results .....	6
1. Log dosyalarının oluşumu, CLIENT: .....	6
2. Log Dosyalarının oluşumu, SERVER: .....	6
3. Log Dosyalarının İçeriği, CLIENT: .....	7
4. Log Dosyalarının İçeriği, SERVER: .....	8
5. ÇALIŞTIRILMA ŞEKLİ: .....	9
6. ÇALIŞTIRILMA SONRASI .....	9
7. CTRL-C, CLIENT: .....	10
8. CTRL-C, SERVER: .....	11
9. BIRDEN FAZLA CLIENT: .....	12
4. ÖNEMLİ:.....	12

# 1. System Requirements

## 1. Server

server.c

compile: ./server <PORT ID> <Thread Pool Size>

PORTID: Haberleşmeyi sağlayacak kanal id si

Thread Pool Size: Bir anda clienttan gelen threadlerden cevap verileceklerin sayısı.

Server thread per request ve worker pool şekilleri ile iki farklı tarzda implement edilmeli.

Bağlanan clienttan matrix için gerekli verileri alır ve cevap verir.

Bağlanan clienttan yollanan her thread için aynı boyutlara sahip farklı matrixler üretir.

Bu matrixlerden yola çıkarak x matrixi bulur.

Bulunan matrixlerdeki hatayı hesaplar.

Matrix üretme, solve etme ve hata bulma işlemleri paralel şekilde gerçekleştirilir.

## 2. Client

clients.c

compile: ./clients <PORT ID> <COLUMN> <ROW> <NUM OF THREAD>

PORTID: server ile haberleşmeyi sağlayacak kanal

COLUMN: oluşturulacak A matrix'inin sütun sayısı

ROW: oluşturulacak A matrix'inin satır sayısı

NUM OF THREAD: client içinde semaphore ile oluşturulacak ve serverden istek yapacak threadlerin sayısı.

Client belirtilen sayıda thread oluşturarak server'a saldırır ve isteklerinin gerçekleşmesini beklerler.

## 2. System Solution Approach

### 1.Server

Server ilk olarak client dan pid ve row, col, numOfThread sayılarını alır.

Ardından client tan oluşturduğu thread in pid sini de ister.

Paralel olarak peş peşe 3 fork yapar ve her fork ta gerekli işlemi gerçekleştirir.

Forklar da paralellik sırasında ilk oluşturulan fork son işlemi yaptığı için ilk fork da P3, 2. Fork da P2 ve 3. Fork da P1 işlemleri gerçekleştirilir.

P1 işlemi, A ve B matrixlerini oluşturur.

P2 işlemi paralel olarak mutex ile thread oluşturup x matrix'ini 3 şekilde çözer.

Solve işlemini x matrix'ine farklı değerler atarak gerçekleştirdim.

x lerin her birini hesaplama süresi bulundu ve bu sürelerden bir ortalama x hesaplama süresi bulundu.

P3 işlemi bulunan x değerlerinden hareketle 3 farklı hata bulur.

P1 de bulunan A ve B matrix'i 2. Process'e, P2'ye, shared memory ile gönderildi.

Aynı şekilde P2 de bulunan x matrixleri , A ve B de aynı şekilde farklı bir shared memory vasıtasıyla P3'e gönderildi.

Server da gelen her bağlantı için

s\_c\_BağlananClientPidsi\_ClienttanBağlananThreadPidsi.log

şeklinde bir log dosyası oluşturup, A, B, X1, X2, X3, Xlerin hesaplama zamanları(t1, t2, t3), Xlerin ortalama hesaplama zamanı (tAverage), Xler bulunurken yapılan hatalar (e1, e2, e3) değerlerini yazar.

Bulduğu bütün bu verileri client'a gönderir.

Server da paralellik nedeniyle bütün veriler istenilen sırada yazılamaz ve karmaşıklık oluşabilir. Bu nedenle bir fifo yardımı ile bütün verileri fifoya yazıp P1, P2 ve P3 processlerinin bitiminde bu verileri fifodan okuyup client'a yolladım.

## 2.Client

Client ilk olarak server'a pidsini, path'ini yollar.

Semaphore vasıtasıyla parallel thread yapılar oluşturur. İstenilen sayıda.

Oluşturulan threadlerin pid si server'a yollanır.

Sonra da server'dan A, B, X1, X2, X3, t1, t2, t3, tAverage, e1, e2, e3 değerlerinin kendisine gönderilmesini bekler.

Client oluşturduğu her thread için

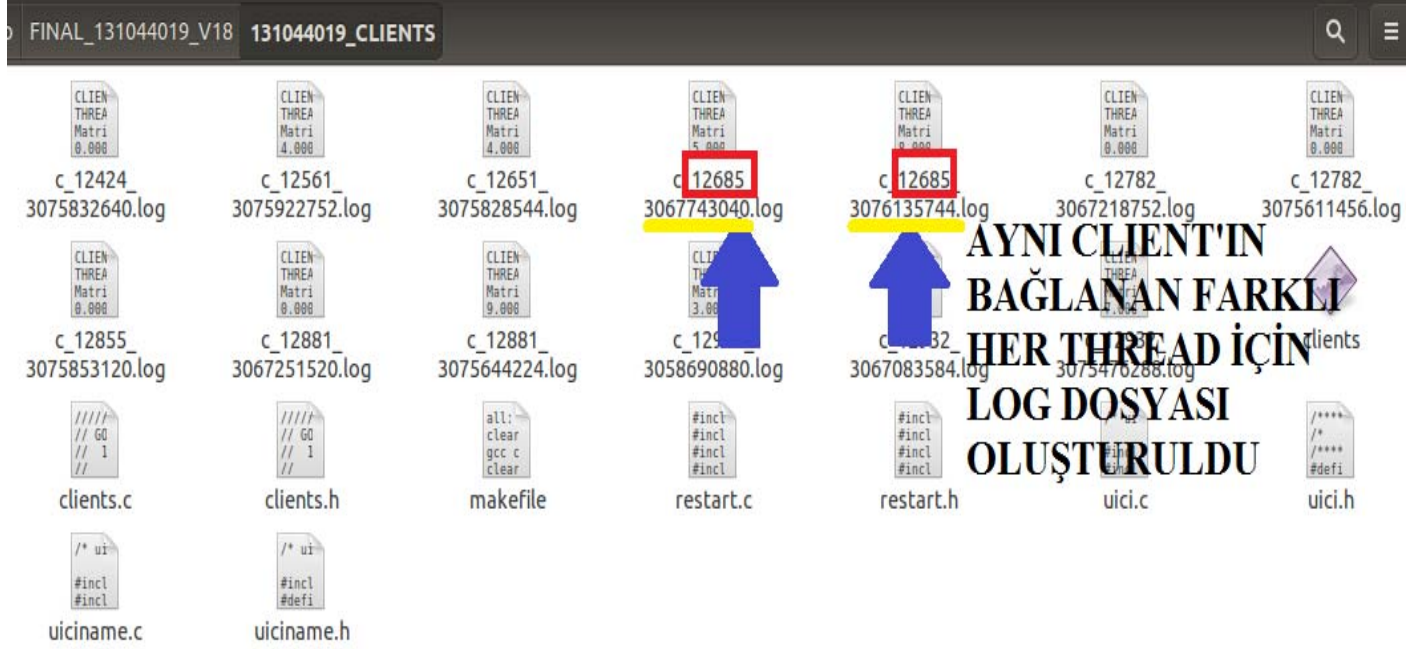
`c_ClientPidsi_ThreadPidsi.log`

şeklinde log dosyası oluşturur.

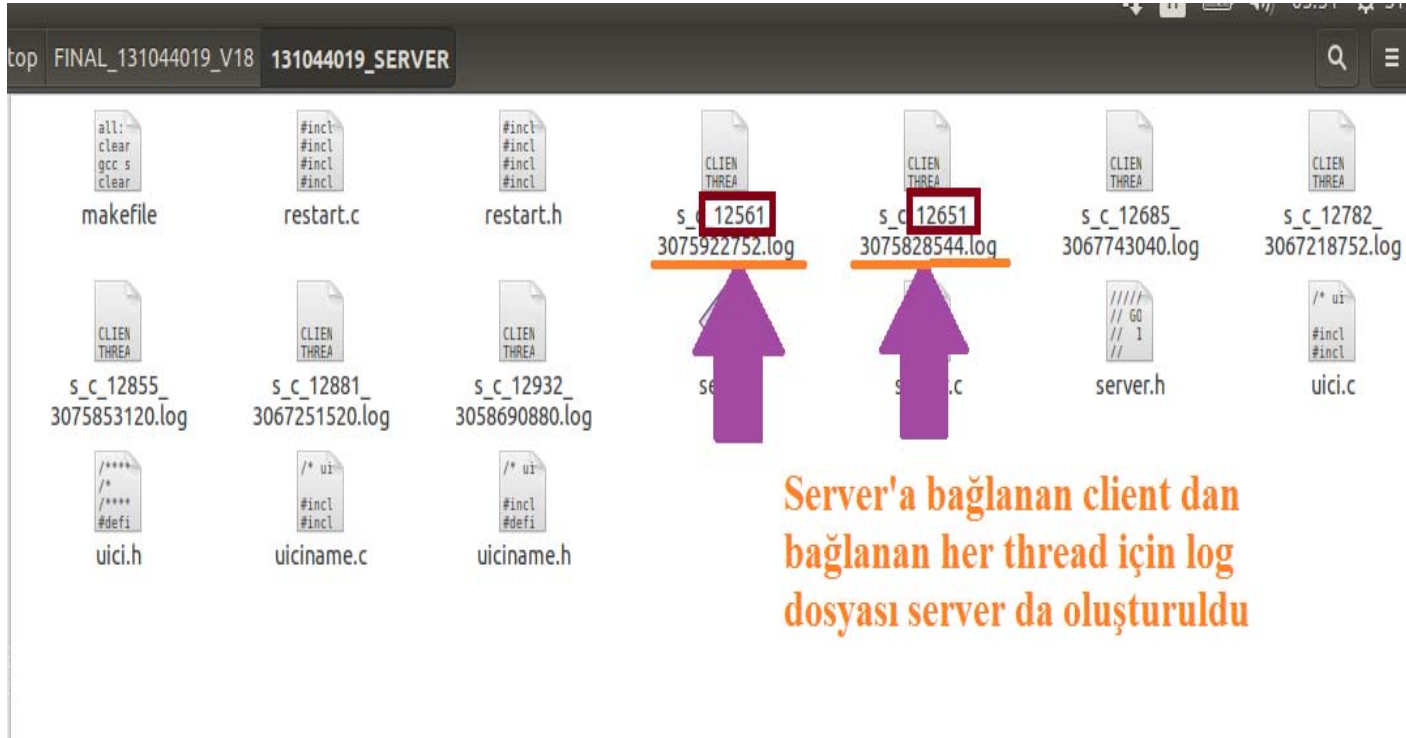
Log dosyasına A, B, X1, X2, X3, t1, t2, t3, tAverage, e1, e2, e3 değerlerini sırayla yazar ve dosyayı kapatır.

### 3. Running and Results

#### 1. Log dosyalarının oluşumu, CLIENT:



#### 2. Log Dosyalarının oluşumu, SERVER:



### 3. Log Dosyalarının İçeriği, CLIENT:

```
clients.c x c_13215_3076246336.log x
1 CLIENTID: 13215
2 THREADID: 3076246336>>>>>>>>>>
3 Matrix A:
4 0.0000 0.0000 0.0000 1.0000
5 0.0000 0.0000 0.0000 7.0000
6 0.0000 0.0000 0.0000 9.0000
7
8
9 Matrix B:
10 0.0000
11 0.0000
12 0.0000
13
14
15 Matrix X1:
16 0.0000 0.0000 0.0000 0.0000
17
18 time: 0.0000
19
20
21 Matrix X2:
22 0.0000 0.0000 0.0000 0.0000
23
24 time: 3.0000
25
26
27 Matrix X3:
28 3.0000 3.0000 3.0000 0.0000
29
30 time: 0.0000
31
32
33 timeAverage: 1.0000
```

#### 4. Log Dosyalarının İçeriği, SERVER:

```
s_c_5155_3059166016.log x server.c x
1
2
3 CLIENTID: 5155
4 THREADID: 3059166016>>>>>>>>>>
5 Matrix A:
6 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
7 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
8 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
9 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
10 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
11 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
12
13
14 Matrix B:
15 0.0000
16 0.0000
17 0.0000
18 0.0000
19 0.0000
20 0.0000
21
22
23 Matrix X1:
24 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
25
26 Matrix X2:
27 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
28
29 Matrix X3:
30 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
31
```



## 5. ÇALIŞTIRILMA ŞEKLİ:

```
user@user-VirtualBox: ~/Desktop/FINAL_131044019/131044019_SERVER
gcc server.c restart.c uici.c uiciname.c -o server -lm -lpthread
user@user-VirtualBox:~/Desktop/FINAL_131044019/131044019_SERVER$ ./server 8632 4

iPortNumber : 8632
SERVER ID: 13213

user@user-VirtualBox: ~/Desktop/FINAL_131044019/131044019_CLIENTS
gcc clients.c restart.c uici.c uiciname.c -o clients -lm -pthread
user@user-VirtualBox:~/Desktop/FINAL_131044019/131044019_CLIENTS$ ./clients 8632 4 3 2
```

**PORTID THREAD POOL SIZE**

**SERVER PID**

**PORTID COL ROW NUMOFTHREAD**

## 6. ÇALIŞTIRILMA SONRASI

```
user@user-VirtualBox: ~/Desktop/FINAL_131044019/131044019_SERVER
Matrix X1:
0.0000 0.0000 0.0000 0.0000
time for x1: 0.0000

Matrix X2:
0.0000 0.0000 0.0000 0.0000
time for x2: 3.0000

Matrix X3:
0.0000 0.0000 0.0000 0.0000
time for x3: 0.0000

timeAverage: 1.0000
e1: 0.0000
e2: 0.0000
e2: 0.0000

waited request...

user@user-VirtualBox: ~/Desktop/FINAL_131044019/131044019_CLIENTS
Matrix X1:
0.0000 0.0000 0.0000 0.0000
time for x1: 0.0000

Matrix X2:
0.0000 0.0000 0.0000 0.0000
time for x2: 3.0000

Matrix X3:
3.0000 3.0000 3.0000 0.0000
time for x3: 0.0000

timeAverage: 1.0000
```

**SERVER**

**CLIENT**

## 7. CTRL-C, CLIENT:

```
user@user-VirtualBox: ~/Desktop/FINAL_131044019/131044019_SERVER
0.0000
0.0000

Matrix X1:
9.0000 4.0000 0.0000 0.0000
time for x1: 0.0000

Matrix X2:
0.0000 0.0000 0.0000 0.0000
time for x2: 3.0000

Matrix X3:
3.0000 3.0000 3.0000 0.0000
time for x3: 0.0000

timeAverage: 1.0000

e1: 0.0000
e2: 0.0000
e2: 7.0000

```

```
user@user-VirtualBox: ~/Desktop/FINAL_131044019/131044019_CLIENTS
4.0000
0.0000

Matrix X1:
0.0000 0.0000 0.0000 0.0000
time for x1: 0.0000

Matrix X2:
0.0000 0.0000 0.0000 0.0000
time for x2: 0.0000

Matrix X3:
0.0000 0.0000 0.0000 0.0000
time for x3: 5.0000

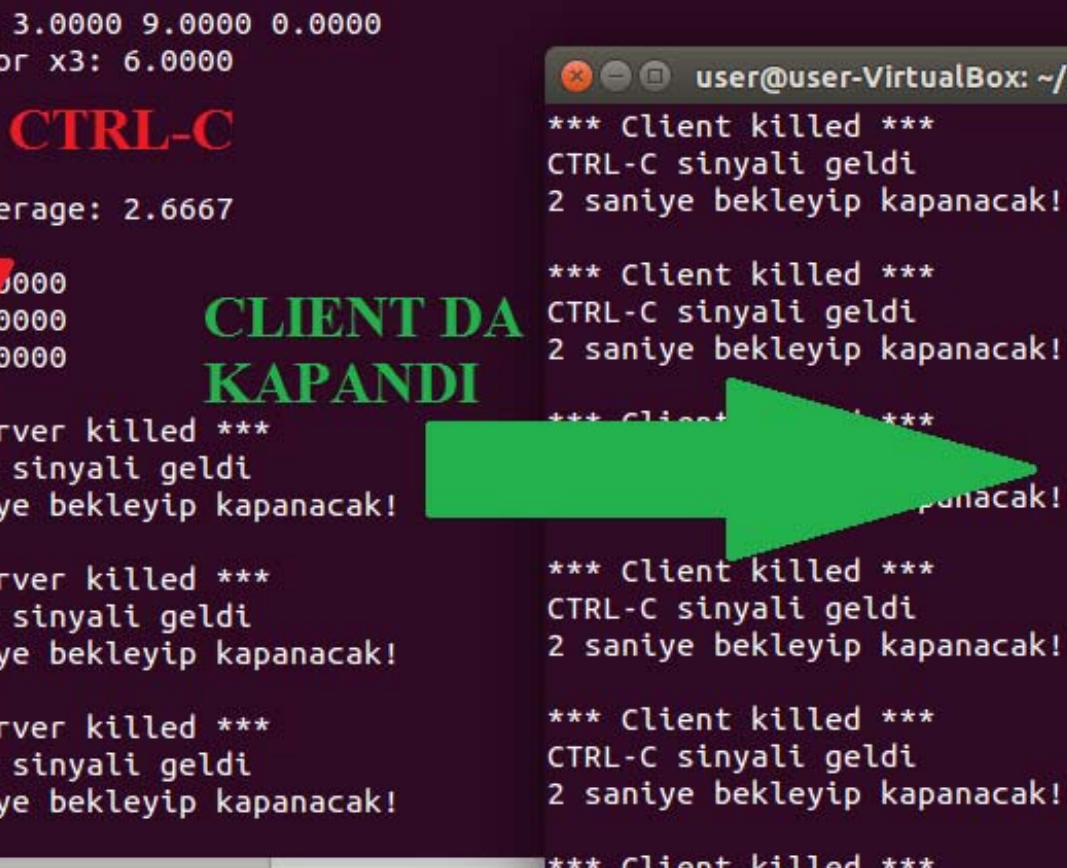
timeAverage: 1.6667

e1: 5.0000
e2: 5.0000
e2: 5.0000
user@user-VirtualBox:~/Desktop/FINAL_131044019/131044019_CLIENTS$ ^C
user@user-VirtualBox:~/Desktop/FINAL_131044019/131044019_CLIENTS$
```

**CTRL-C yapmaya vakit kalmadan işlemi gerçekleştiriyor**

Browse Network  
Connect to Server

## 8. CTRL-C, SERVER:



user@user-VirtualBox: ~/Desktop/FINAL\_131044019/131044019\_SERVER

```
Matrix X3:
8.0000 3.0000 9.0000 0.0000
time for x3: 6.0000

CTRL-C

average: 2.6667

0.0000
e2.0000
e2.0000
^C
*** Server killed ***
CTRL-C sinyali geldi
2 saniye bekleyip kapanacak!

*** Server killed ***
CTRL-C sinyali geldi
2 saniye bekleyip kapanacak!

*** Server killed ***
CTRL-C sinyali geldi
2 saniye bekleyip kapanacak!
```

**CLIENT DA KAPANDI**

user@user-VirtualBox: ~/Desktop

```
*** Client killed ***
CTRL-C sinyali geldi
2 saniye bekleyip kapanacak!

*** Client killed ***
CTRL-C sinyali geldi
2 saniye bekleyip kapanacak!

*** Client killed ***
CTRL-C sinyali geldi
2 saniye bekleyip kapanacak!

*** Client killed ***
CTRL-C sinyali geldi
2 saniye bekleyip kapanacak!

*** Client killed ***
CTRL-C sinyali geldi
2 saniye bekleyip kapanacak!
```

Browse Network

Connect to Server



## 9. BIRDEN FAZLA CLIENT:

```
user@user-VirtualBox: ~/Desktop/FINAL_131044019/131044019_SERVER
7.0000
3.0000
SERVER
Matrix X1:
0.0000 0.0000 0.0000
time for x1: 5.0000
Matrix X2:
5.0000 5.0000 0.0000
time for x2: 0.0000
Matrix X3:
0.0000 0.0000 0.0000
time for x3: 5.0000
timeAverage: 3.3333
e1: 5.0000
e2: 5.0000
e2: 0.0000
[ ]
Browse Network
Connect to Server

user@user-VirtualBox: ~/Desktop/FINAL_131044019/131044019_CLIENT1
4.0000
0.0000
CLIENT1
Matrix X1:
0.0000 0.0000 0.0000 0.0000
time for x1: 0.0000
Matrix X2:
0.0000 0.0000 0.0000 0.0000
time for x2: 0.0000
Matrix X3:
0.0000 0.0000 0.0000 0.0000
time for x3: 5.0000
timeAverage: 1.6667
e1: 5.0000
e2: 5.0000
e2: 5.0000
user@user-VirtualBox: ~/Desktop
user@user-VirtualBox: ~/Desktop

user@user-VirtualBox: ~/Desktop/FINAL_131044019/131044019_CLIENT2
0.0000
0.0000
0.0000
CLIENT2
Matrix X1:
5.0000 5.0000 5.0000
time for x1: 684.0000
Matrix X2:
0.0000 0.0000 0.0000
time for x2: 0.0000
Matrix X3:
0.0000 0.0000 3.0000
time for x3: 5.0000
timeAverage: 229.6667
```

## 4. ÖNEMLİ:

uici.h, uiciname.h ve restart.h kütüphaneleri kullanıldı.

Kütüphanelerin c dosyaları ve h dosyaları klasörün içinde var.

Yararlanılan kaynaklar kullanıldığı .c uzantılı dosya da en üstte yorum içinde yazıldı.