Gebze Technical University

Computer Engineering

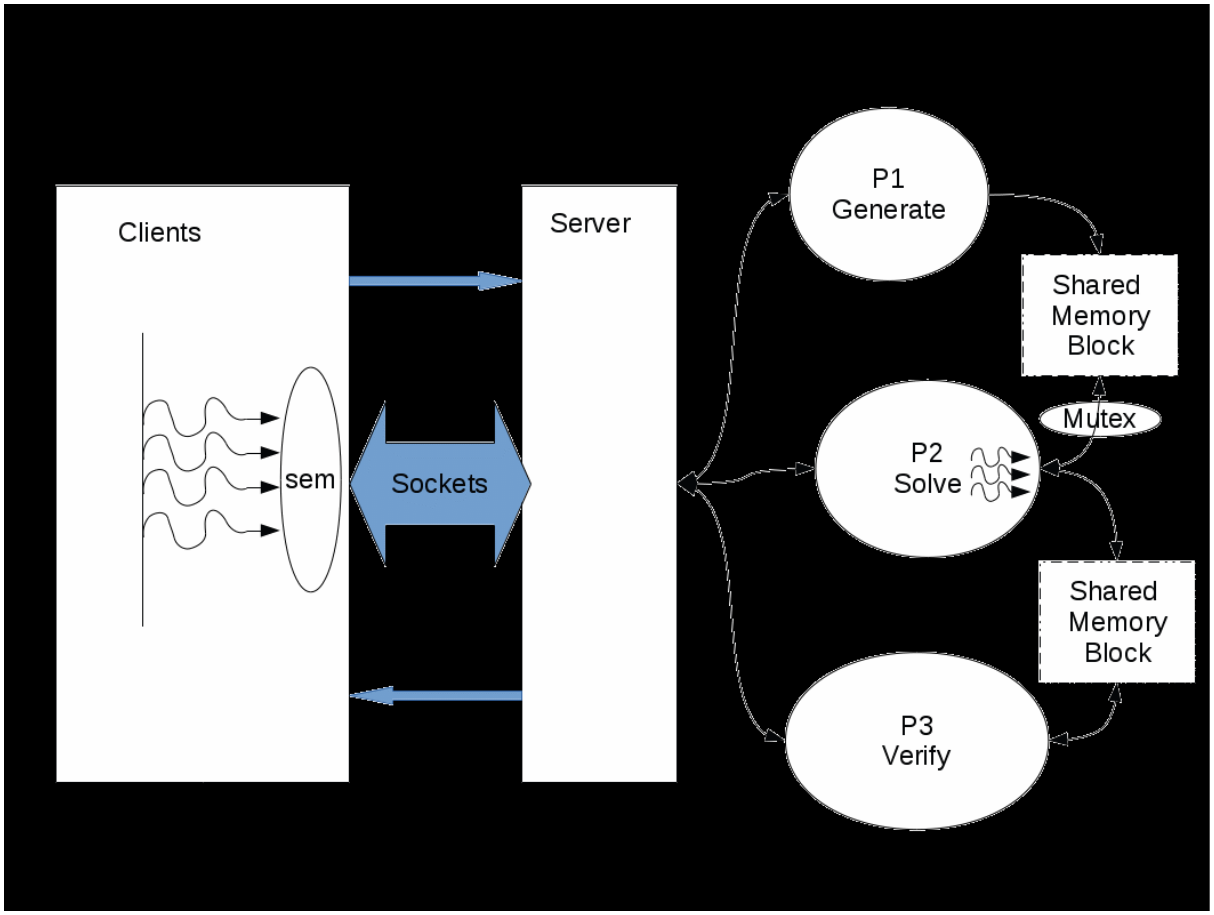
CSE 244

2017 Spring

FINAL PROJECT REPORT

BURAK DEMİRCİ

141044091



Clients:

Ödevde istenilen Clients verilen sayıda thread oluşturup serverden soket üzerinden haberleşerek istenilen matris boyutlarında matris oluşturup bunları denklem yapıp çözmek ve çözümü cliente yollamaktır.

Clients Usage: <#of columns of A, m> <#of rows of A, p> <#of clients, q> <#portNo, pt>

./client 4 3 5 1245 gibi çalışmaktadır

Clients çalışma sistemi ilk öncelikle verilen thread sayısında thread oluşturmak ve bu threadleri serverle iletişime geçirmektir. Serverle iletişime geçerken semefor kullanıldı nedeni ise threadlerin hepsinin aynı anda bağlanmasını engllemek kullanılan semafor sayesinde düzenli şekilde iletişime geçilip. Matris boyutları ve clientsin pid si servere gönderildi. Pid göndermemdeki sebep ise server öldüğünde clientin de ölmesini sağlamak.Server bütün işeri yapıp cliente sonuçları yolladığında client sonuçları log dosyasına yazar. Ayrıyeten clientslerin ortalama bağlanma süreleri ve bu sürelerin standart sapmaları hem CTRL+C yakalandığında yapıyor hemde normal client bittiğinde yapıyor. Oluşan log dosyaları clientin pid.log şeklinde oluşturulmuştur.

Ödev multi client şekildede denenmiştir.

Server:

Ödevde iki çeşit yapıda server istendi biri thread_per_request diğeri ise workerpool bunlar tek fonksiyonda yazılmış olup sadece Usagelere farklı değer verilerek çalıştırılabilir.

Usage for worker_pool : <port #, id> <thpool size, k >

Usage for thread_per_request : <port #, id>

thread_per_request :

Bu yapı clientten gelen her istek için server bir tane thread oluşturur ve soket üzerinden o clientle bağlantıya geçilir. Burada oluşabilecek karmaşıklıkları engellemek amacıyla semafor kullanıldı.

Oluşan her thread " void Connection(void *n) " isimli fonksiyona gider bu fonksiyon ise işlemleri yapmak için " void calculate(int col,int row,int communfd) " fonksiyonuna yönlenir buradaki communfd server ile client arasında ki yol, col ve row ise istenilen matrisin column ve row'u dur.

Bu fonksiyon P1 , P2 ve P3 işlemlerini paralel yapmaktadır içerisinde 3 fork işlemi yapıldı ve parent proccess bu forkların en altında wait(NULL) yapmaktadır. Aynı anda forka giren işlemler paralel çalışmaktadır. Üretilen matrisler Bir shared memorye yazılmaktadır yazıldıktan sonra bunu okumak için giden P2 fork'u ilk öncelikle 3 farklı işlem için aynı anda thread açmaktadır. Vu threadler bu shared memory'den matrisleri okurken sıkıntı çıkmaması için mutex yapısı kullanıldı okumadan önce phthread_mutex_lock yapıldı okundukduktan sonra phthread_mutex_unlock yapıldı okuma sırasında karışıklıklar engelendi. Buradaki matematiksel işlemler yapılamayıp yerine random olarak matrisler üretilerek bu random matrisler başka bir shared memory bloğuna yazıldı yazılırken yine mutex yapısı kullanıldı. Elde edilen bu 3 farklı matris ise P3 forkundan okundu ve

" void errorCheck(int *result,int *data,int communfd, int col, int row)"

Hata kontrolü ve yazma işlemleri için bu fonksiyona yönlendirildi. Bu fonksiyon pdfte verilen hata kontrollerini uygulayarak aşağıda verilen işlemleri her bir sonuç için gerçekleştirmiştir.

$$e = Ax_d - b \qquad e = Ax_d - b$$

void WriteLog(int *all, int size,int col, int row)

fonksiyonu ise servere yazılması gereken değerleri yazmıştır.

Workerpool:

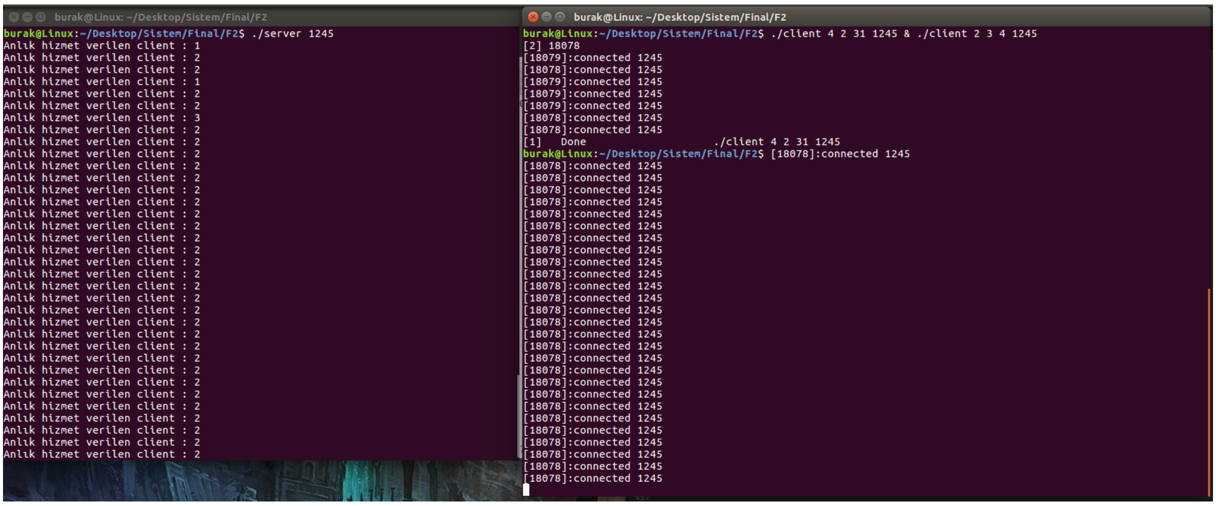
Bu yöntem ise sadece clientlerin bağlanırken threadleri her request için değilde başta verilen thread kadar oluşturup bu threadleri sürekli kullanmasıyla oluşan sistemdir diğer bütün işlemler thread_per_request ile aynıdır.

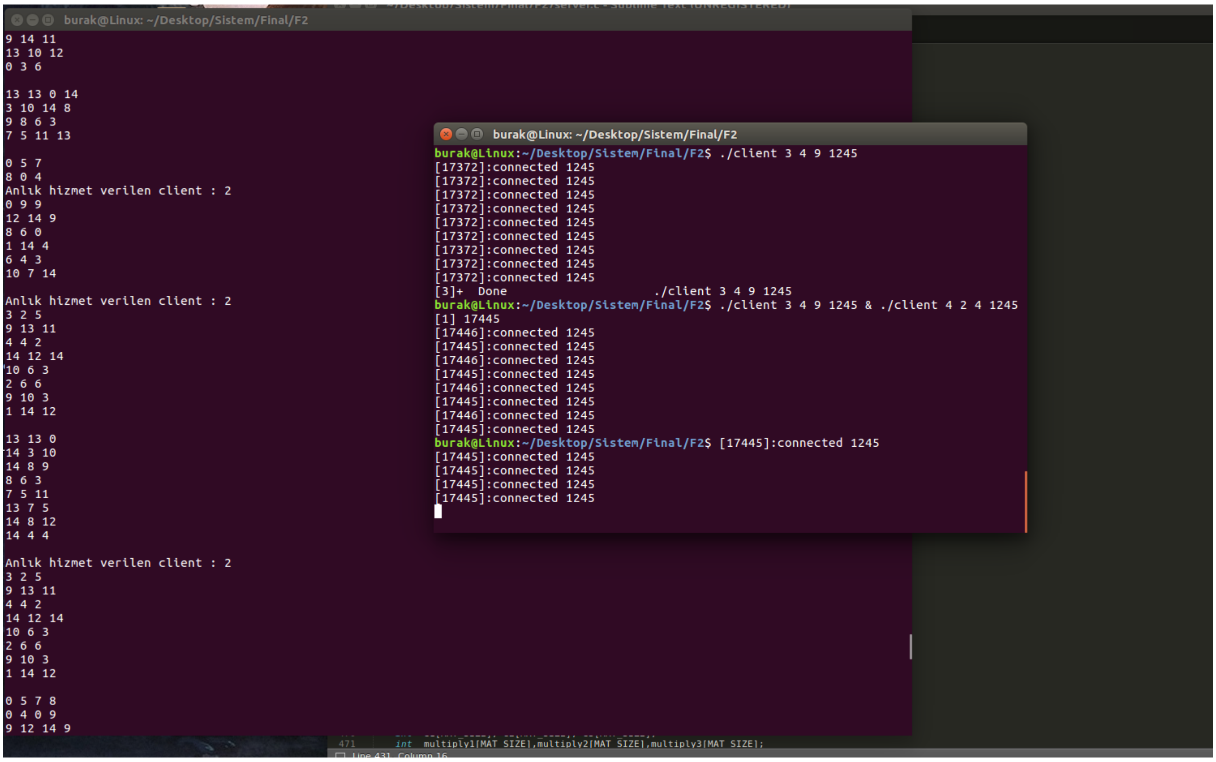
CTRL+C sinyali kontrol edilmiş geldiği anda servere bağlı olan bütün clientleri öldürüp kendisi ölerek sonlanır.

NOT: uici.c , uiciname.c ve u_open.c fonksiyonları kitap kodundan alınmıştır buradaki bazı fonksiyonlar kullanılmıştır.

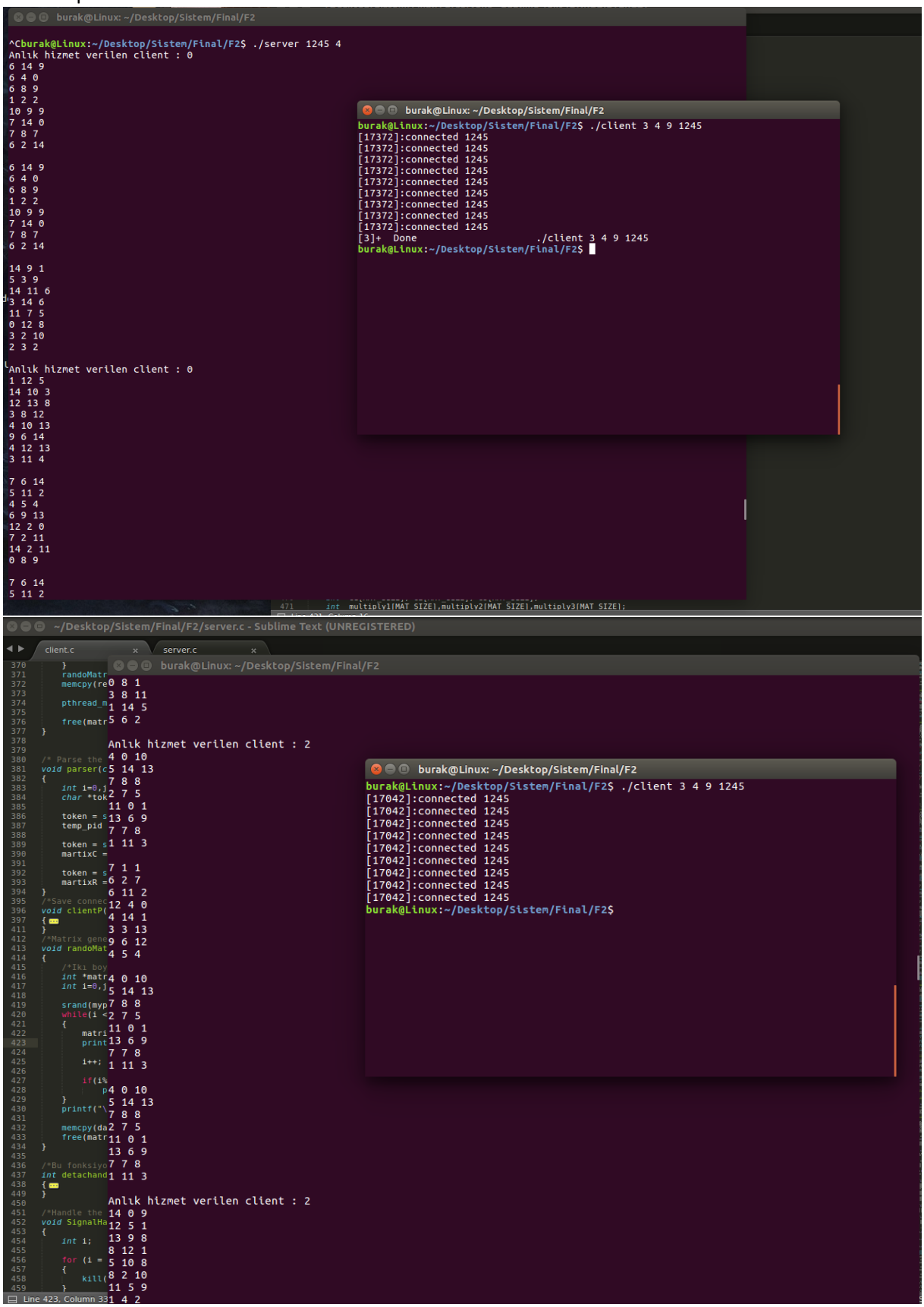
TEST

MULTİ- Client - Single Client

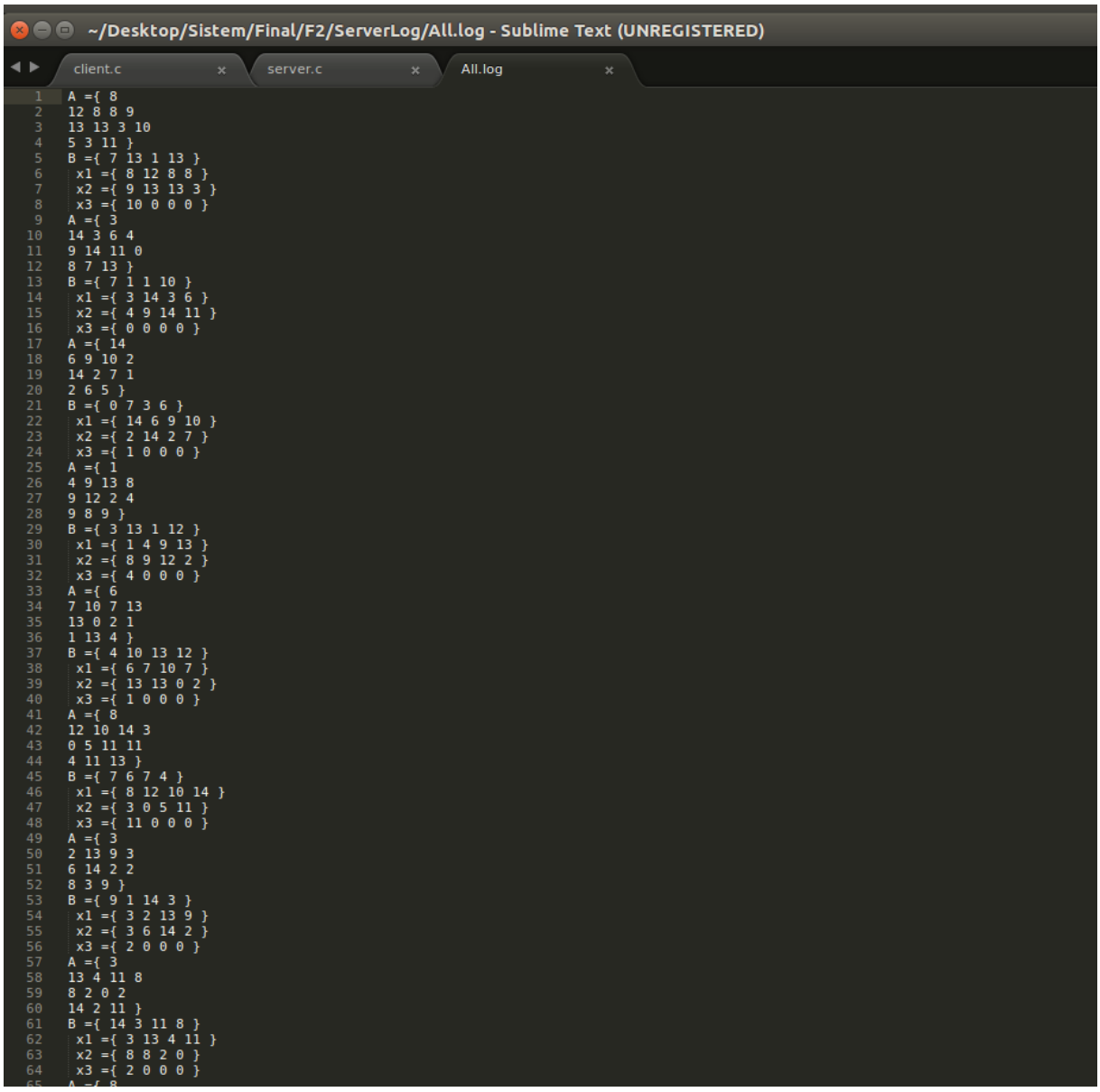




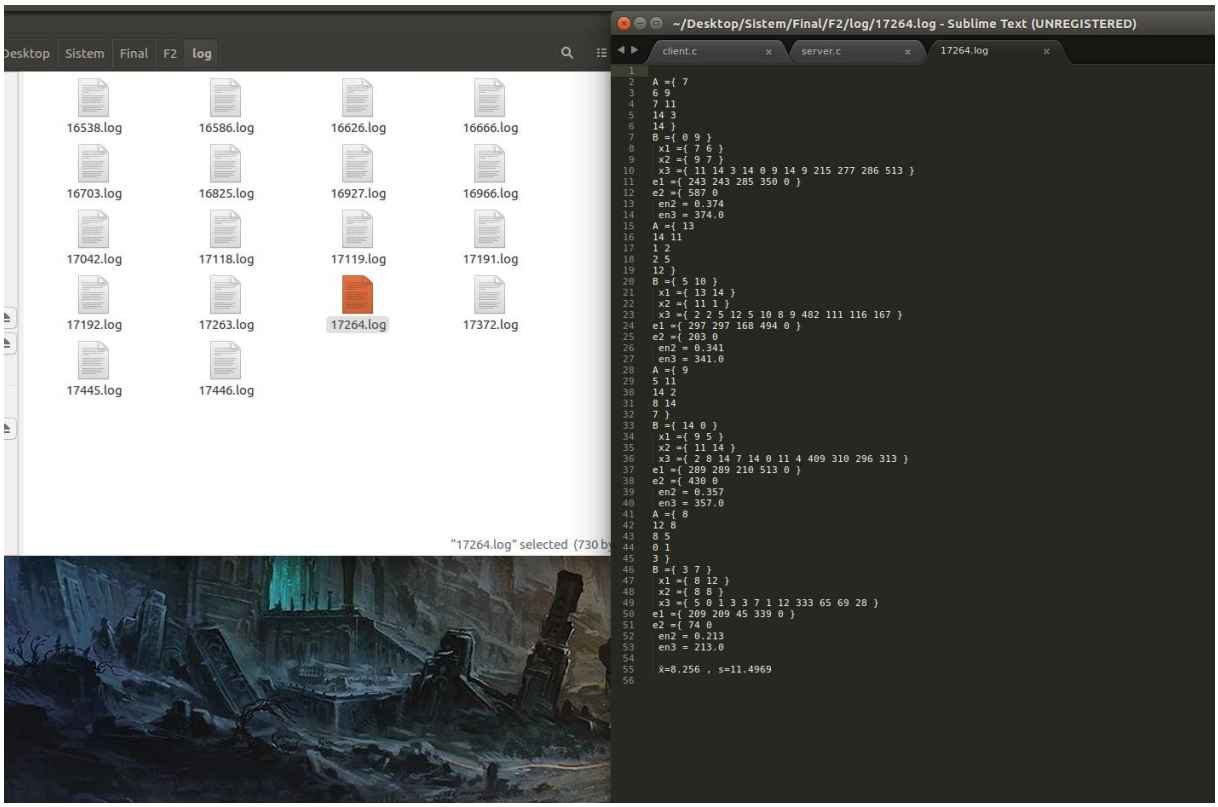
Worker pool


burak@Linux: ~/Desktop/Sistem/Final/F2
^Cburak@Linux:~/Desktop/Sistem/Final/F2$ ./server 1245 4
Anlık hizmet verilen client : 0
burak@Linux: ~/Desktop/Sistem/Final/F2
burak@Linux:~/Desktop/Sistem/Final/F2$ ./client 3 4 9 1245
[17372]:connected 1245
[3]+ Done ./client 3 4 9 1245
burak@Linux:~/Desktop/Sistem/Final/F2$
~/Desktop/Sistem/Final/F2/server.c - Sublime Text (UNREGISTERED)
client.c
server.c
burak@Linux: ~/Desktop/Sistem/Final/F2
Anlık hizmet verilen client : 2
burak@Linux: ~/Desktop/Sistem/Final/F2
burak@Linux:~/Desktop/Sistem/Final/F2$ ./client 3 4 9 1245
[17042]:connected 1245
burak@Linux:~/Desktop/Sistem/Final/F2$
Line 423, Column 33

LOG DOSYALARI




```
A ={ 8
12 8 8 9
13 13 3 10
5 3 11 }
B ={ 7 13 1 13 }
 x1 ={ 8 12 8 8 }
 x2 ={ 9 13 13 3 }
 x3 ={ 10 0 0 0 }
A ={ 3
14 3 6 4
9 14 11 0
8 7 13 }
B ={ 7 1 1 10 }
 x1 ={ 3 14 3 6 }
 x2 ={ 4 9 14 11 }
 x3 ={ 0 0 0 0 }
A ={ 14
6 9 10 2
14 2 7 1
2 6 5 }
B ={ 0 7 3 6 }
 x1 ={ 14 6 9 10 }
 x2 ={ 2 14 2 7 }
 x3 ={ 1 0 0 0 }
A ={ 1
4 9 13 8
9 12 2 4
9 8 9 }
B ={ 3 13 1 12 }
 x1 ={ 1 4 9 13 }
 x2 ={ 8 9 12 2 }
 x3 ={ 4 0 0 0 }
A ={ 6
7 10 7 13
13 0 2 1
1 13 4 }
B ={ 4 10 13 12 }
 x1 ={ 6 7 10 7 }
 x2 ={ 13 13 0 2 }
 x3 ={ 1 0 0 0 }
A ={ 8
12 10 14 3
0 5 11 11
4 11 13 }
B ={ 7 6 7 4 }
 x1 ={ 8 12 10 14 }
 x2 ={ 3 0 5 11 }
 x3 ={ 11 0 0 0 }
A ={ 3
2 13 9 3
6 14 2 2
8 3 9 }
B ={ 9 1 14 3 }
 x1 ={ 3 2 13 9 }
 x2 ={ 3 6 14 2 }
 x3 ={ 2 0 0 0 }
A ={ 3
13 4 11 8
8 2 0 2
14 2 11 }
B ={ 14 3 11 8 }
 x1 ={ 3 13 4 11 }
 x2 ={ 8 8 2 0 }
 x3 ={ 2 0 0 0 }
A ={ 8
```


Desktop
Sistem
Final
F2
log
16538.log
16586.log
16626.log
16666.log
16703.log
16825.log
16927.log
16966.log
17042.log
17118.log
17119.log
17191.log
17192.log
17263.log
17264.log
17372.log
17445.log
17446.log
"17264.log" selected (730 by
~/Desktop/Sistem/Final/F2/log/17264.log - Sublime Text (UNREGISTERED)
client.c
server.c
17264.log
A ={ 7
6 9
7 11
14 3
14 }
B ={ 0 9 }
x1 ={ 7 6 }
x2 ={ 9 7 }
x3 ={ 11 14 3 14 0 9 14 9 215 277 286 513 }
e1 ={ 243 243 285 350 0 }
e2 ={ 587 0
en2 = 0.374
en3 = 374.0
A ={ 13
14 11
1 2
2 5
12 }
B ={ 5 10 }
x1 ={ 13 14 }
x2 ={ 11 1 }
x3 ={ 2 2 5 12 5 10 8 9 482 111 116 167 }
e1 ={ 297 297 168 494 0 }
e2 ={ 203 0
en2 = 0.341
en3 = 341.0
A ={ 9
5 11
14 2
8 14
7 }
B ={ 14 0 }
x1 ={ 9 5 }
x2 ={ 11 14 }
x3 ={ 2 8 14 7 14 0 11 4 409 310 296 313 }
e1 ={ 289 289 210 513 0 }
e2 ={ 430 0
en2 = 0.357
en3 = 357.0
A ={ 8
12 8
8 5
0 1
3 }
B ={ 3 7 }
x1 ={ 8 12 }
x2 ={ 8 8 }
x3 ={ 5 0 1 3 3 7 1 12 333 65 69 28 }
e1 ={ 209 209 45 339 0 }
e2 ={ 74 0
en2 = 0.213
en3 = 213.0
x̄=8.256 , s=11.4969