T.C.

GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

SYSTEM PROGRAMING FINAL

FÜRKAN YILDIZ

141044031

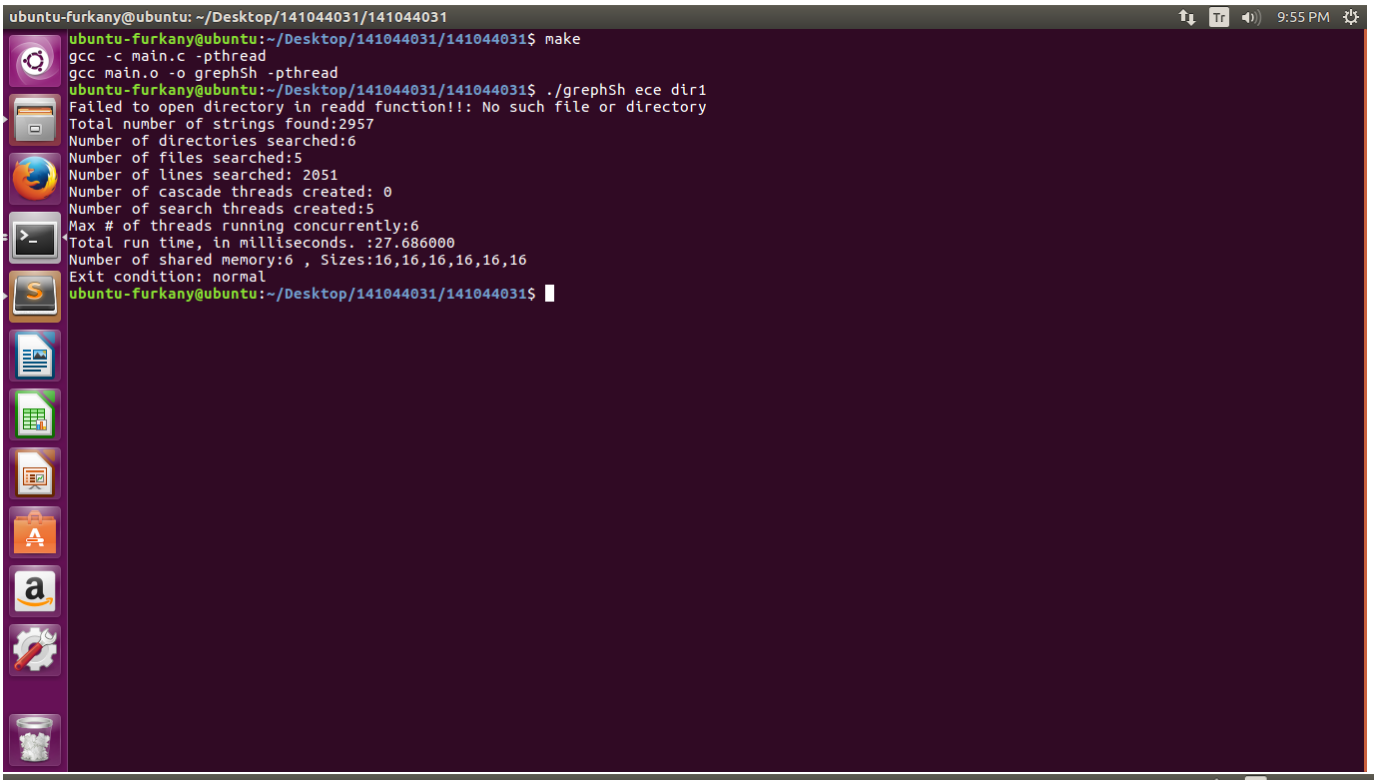
Bu projede bizden istenilen 2 adet program yazılmasıydı, server ve client. Programlar şu şekilde çalıştırılabilmekte,

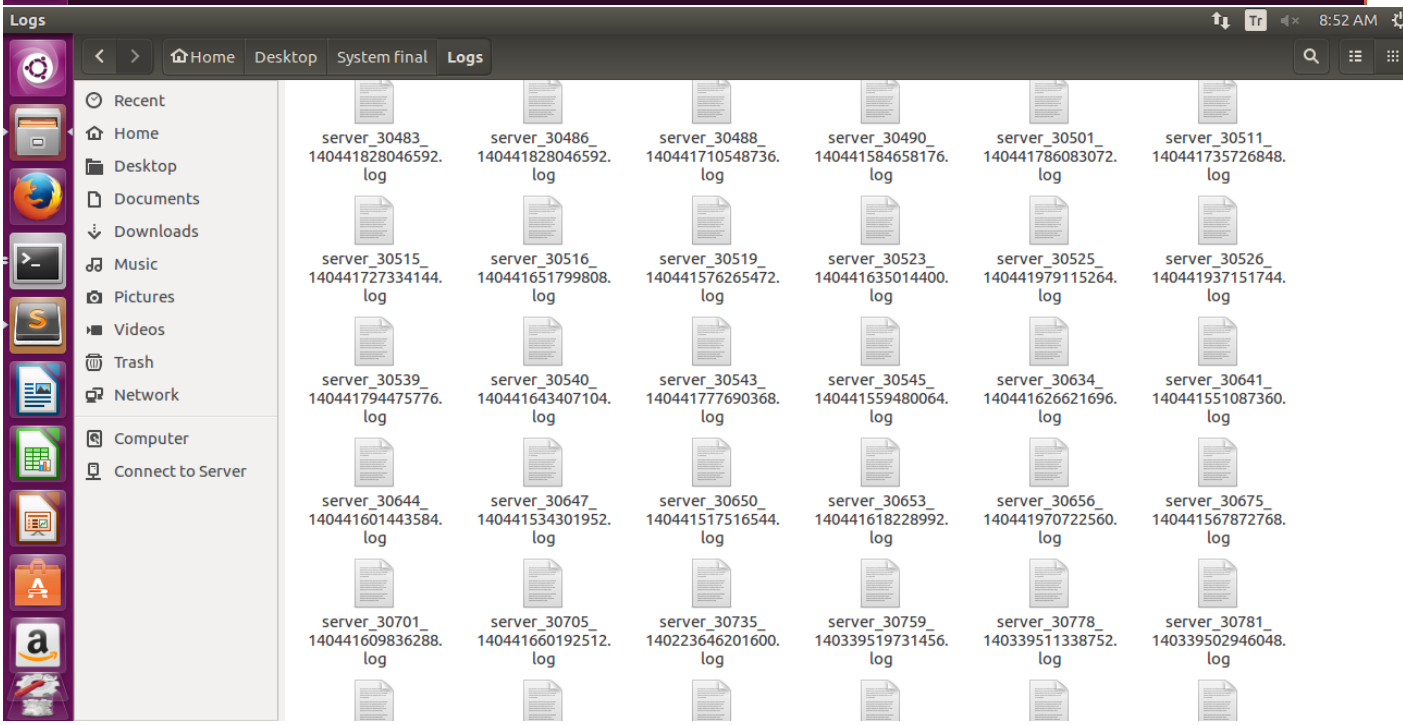
./server <port #, id> <thpool size, k >

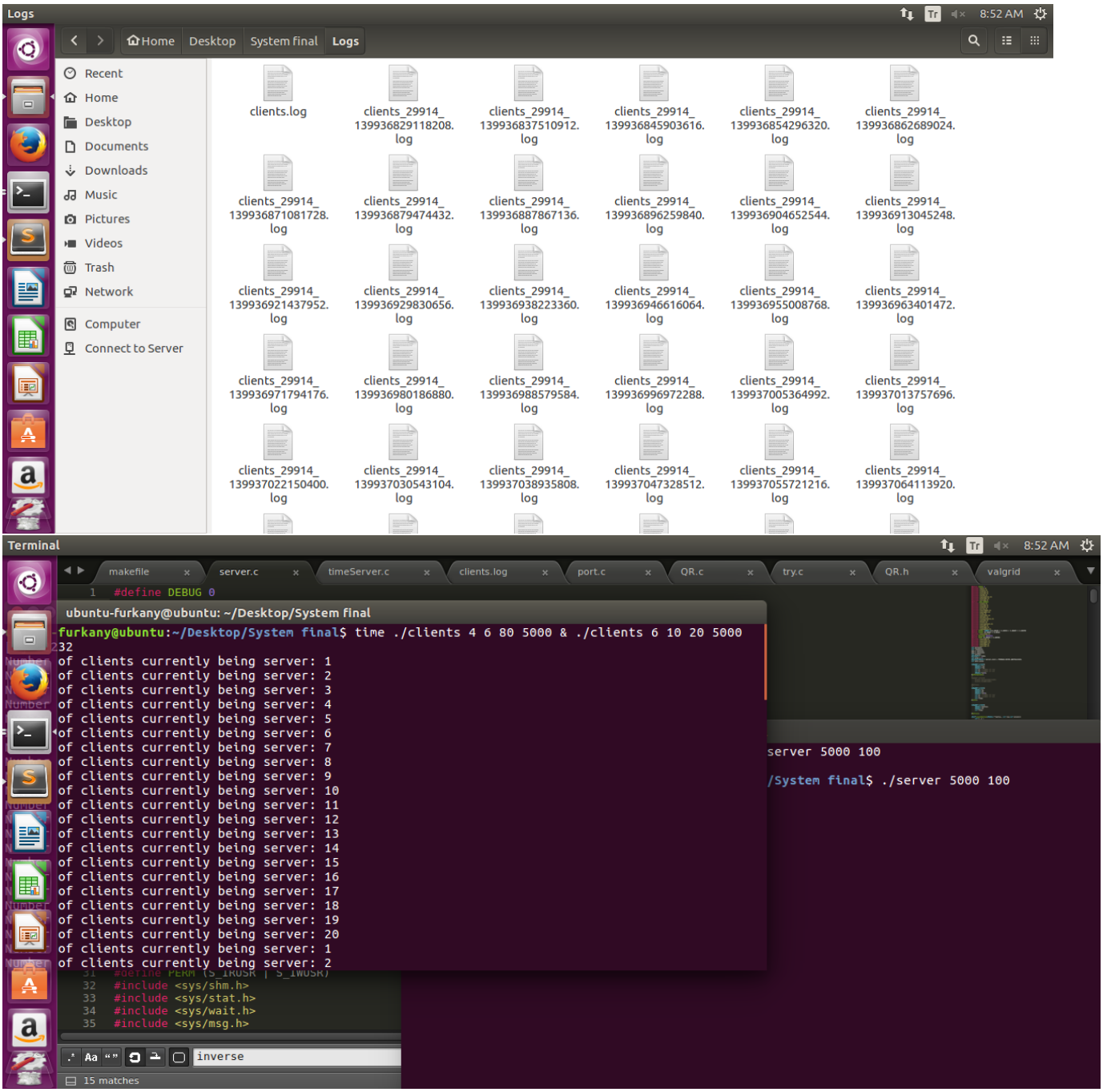
./clients <#of columns of A, m> <#of rows of A, p> <#of clients, q> < port no>

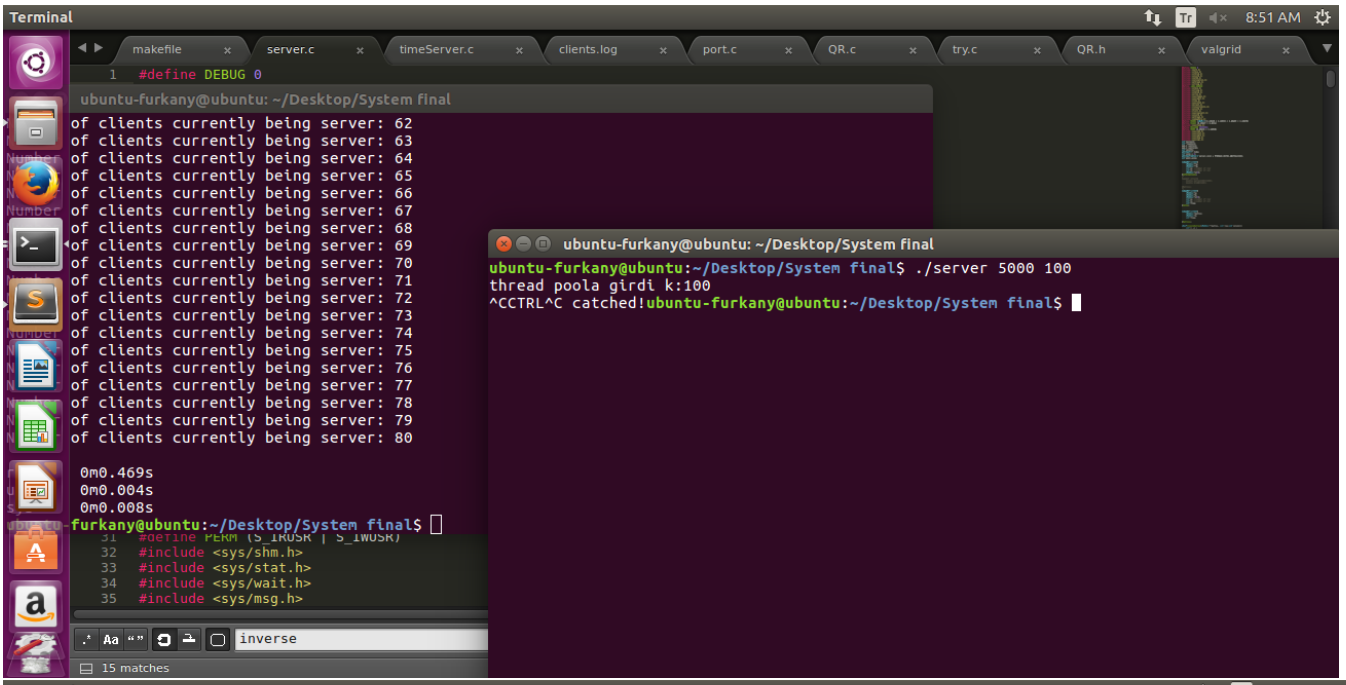
Server iki farklı şekilde implement edildi, kendisine thread pool size'ı olarak sıfır geldiğinde, servera bağlanan her yeni client için yeni bir thread oluştururken, sıfırdan büyük bir sayı girildiğinde ise bir threadpool'u oluşturuluyor ve threadler kendilerine clientlerin bağlanmasını bekliyorlar. Her request geldiğinde thread oluşturacağımız versiyonda threadler detachable yapıldı, böylelikle thread arrayi tutulmadı ve join işlemine gerek kalmadı. Thread pool implementasyonunda ise threadlere her request bağlandığında artan bir count tutuldu ve bu counta göre global olarak tutulan tid arrayi, ^C sinyali geldiğinde join edildi. Server ile client Socket ile birbirine bağlandı, aradaki bilgi akışını doğru şekilde sağlayabilmek için semafor koyuldu. Aradaki bilgi akışı ise clientten servera matrisin boyutları ve serverdan cliente tüm işlemlerin ardından oluşturulan matrislerdir. Server request aldığında (client'te oluşturulan threadlerden birisi connect olmak istediğinde servera) server tarafından 3 adet process oluşturuluyor, 1. Process (p1), clientten gelen matris boyutlarına göre bir matris üretiyor, üretilen bu matris shared memory ve mutex ile 2. Process'e aktarılıyor. Bu işlemler yapılırken, öncelikle shared memory oluşturuldu, shared memorye bir struct pointeri koyuldu ve bu structun içerisinde A matrisinin ve B matrisin pointerları ve bir flag var. Shared memoryde matrisler tek boyutlu olarak aktarılması tercih edildi. Bunun için structun, A matrisinin ve B matrisinin boyutlarının toplamında bir shared memory allocate edildi. (Burada ayrılan shared memory 3'e bölünülmüş olarak düşünüldü, struct için, A matrixi için ve B matrixi için. Daha sonra structun bu elamanlarına shared memorydeki adresleri atandı, örneğin structdaki A matrixinin başlangıç adresi olarak allocate edilen yerin sizeof(struct) kadar ilerisi gösterildi, B matrisinin başlangıç adresi olarak ise allocate edilen yere, sizeof(struct) ve A matrixinin boyutu eklenerek gösterildi. Böylece, shared memorye koyduğumuz structların içerisindeki pointerlar hafızada bir yere sahip oldular. Structurede ayrıca bir adet flag var bu flag, p1 den yazıldığı zaman p2nin haberi olup işlemlerini devam ettirmesini sağlıyor. Flag 0 yapıldı bunun anlamı, matrixlerin henüz doldurulmadığıdır. Allocation işlemlerinin ardından, p1 p2 ve p3 aynı anda create edildi, p1 de hafızada yerleri ayrılmış olan matrixler, random olarak dolduruldu ve structurede tutulan flag 1 yapıldı. Flag'in 1 yapılmasını bekleyen p2 processi hemen işlemlerine başladı. Öncelikle shared memory ile tek boyutta gelen matrixleri 2 boyuta çevirdi. Ardından yeni bir structureye eklenerek, bu structurenin içerisine A, B matrixleri ve p, m bilgileri eklendi. Ayrıca bu structureye aradığımız x değeride eklendi, 3 farklı çözüm fonksiyonumuz olduğundan bu x, 3 boyutlu pointer arrayi, x i eklememizin nedeni, çözüm üreten fonksiyonlara gönderdiğimizde bulunan x değerinin structureye yazılması ve oradan kolaylıkla erişebilmemiz. 3 çözüm methodumuzdan ise ancak birisini yazabildim "pseudo_inverse", fakat bu noktada amacımızın matrix çözümlerini 3 farklı şekilde üretebilmek değil, aynı anda çalışan 3 threadi yönetebilmek olduğunu düşündüğümden, 3 thread oluşturdum ve 3 ünüde pseudeo_inverse ile çalıştırdım. Çözüm fonksiyonlarını verdiğimiz threadlerimiz işlerini bitirdiğinde, threadlere parametre olarak verdiğimiz structuredeki x değerleri başarılı bir şekilde dolduruldu. Bu sayede p2 de "Logs/server_pid_threadID.log" işiminde bir log dosyası açılarak içerisine matlab formatında A, B, x1(1. Çözüm yolu ile çözüldüğünde üretilen x matrisi),x2(2. Çözüm yolu ile çözüldüğünde üretilen x matrisi) ve x3 bastırıldı. P1 ile P2 arasındaki configurasyon için ise çok lazım olmasada Mutex kullanıldı. P2, solveyi ürettikten sonra ise, P3 üretilen bu solvenin hata payını hesaplamak adına P2 ve P3 arasındaki shared memoryden A , B, x1 , x2 ve x3 matrixleri yollandı. Shared memoryden allocation işlemleri P1 ve P2 arasında yapıldığı gibi yapıldı.

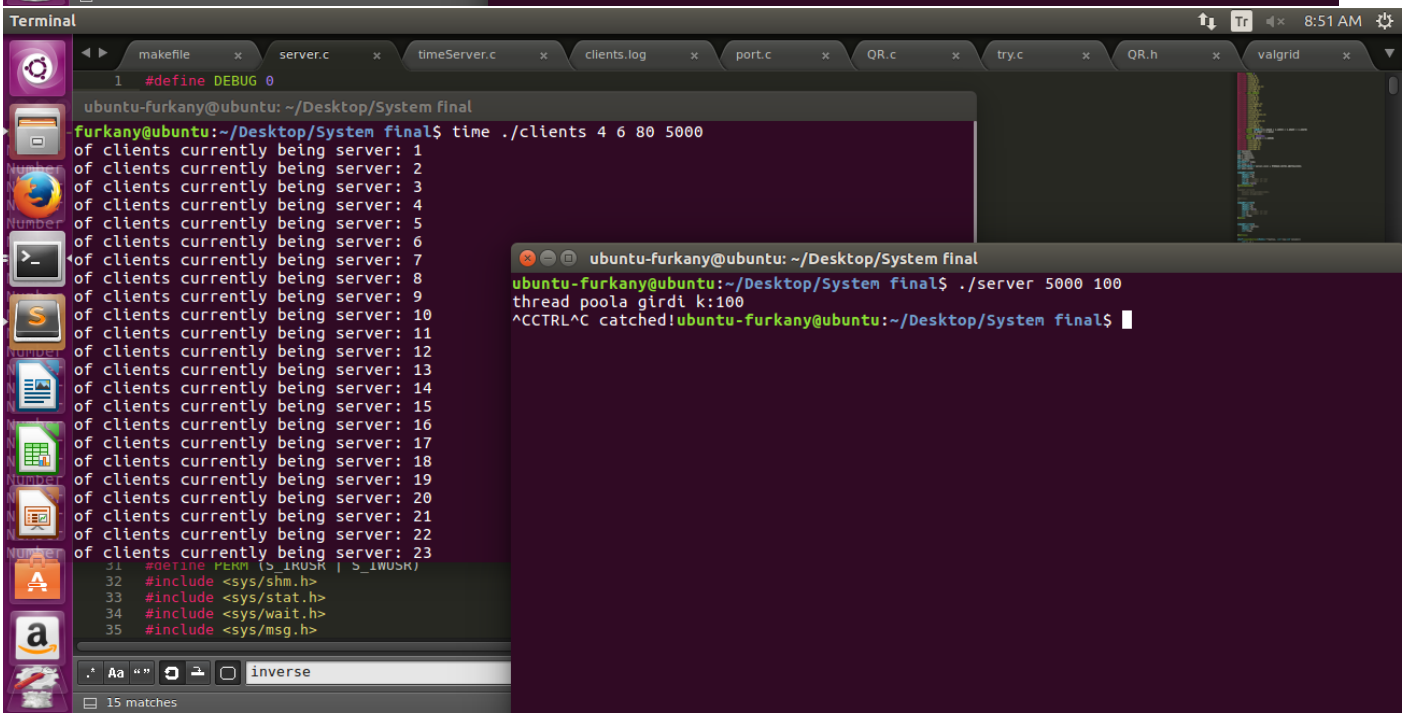
Thread pool yapılan, diğer duruma göre çok daha hızlıdır, çünkü başlangıçta allocation yapılıyor, programın ortasında ha bire allocation yapılmıyor.

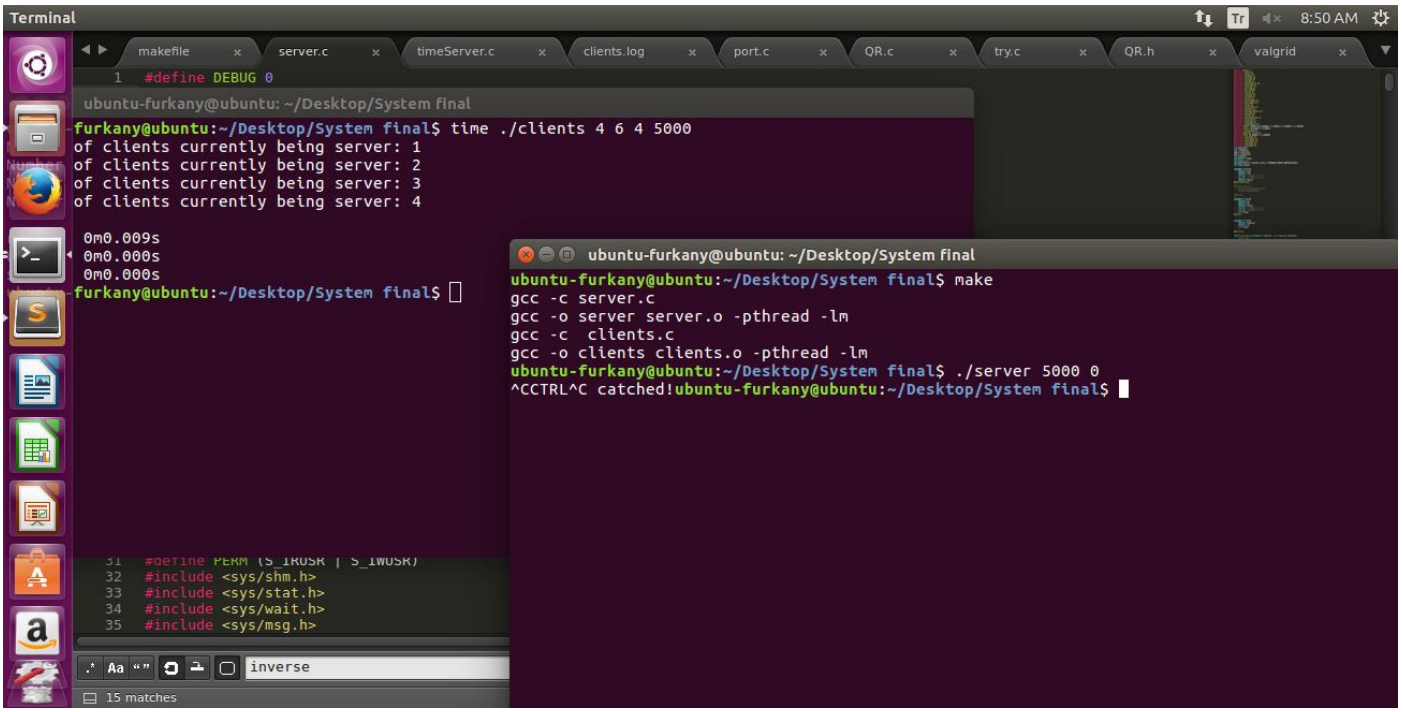





Logs
8:52 AM
Home
Desktop
System final
Logs
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
Connect to Server
clients.log
clients_29914_139936829118208.log
clients_29914_139936837510912.log
clients_29914_139936845903616.log
clients_29914_139936854296320.log
clients_29914_139936862689024.log
clients_29914_139936871081728.log
clients_29914_139936879474432.log
clients_29914_139936887867136.log
clients_29914_139936896259840.log
clients_29914_139936904652544.log
clients_29914_139936913045248.log
clients_29914_139936921437952.log
clients_29914_139936929830656.log
clients_29914_139936938223360.log
clients_29914_139936946616064.log
clients_29914_139936955008768.log
clients_29914_139936963401472.log
clients_29914_139936971794176.log
clients_29914_139936980186880.log
clients_29914_139936988579584.log
clients_29914_139936996972288.log
clients_29914_139937005364992.log
clients_29914_139937013757696.log
clients_29914_139937022150400.log
clients_29914_139937030543104.log
clients_29914_139937038935808.log
clients_29914_139937047328512.log
clients_29914_139937055721216.log
clients_29914_139937064113920.log
Terminal
8:52 AM
makefile
server.c
timeServer.c
clients.log
port.c
QR.c
try.c
QR.h
valgrid
#define DEBUG 0
ubuntu-furkany@ubuntu: ~/Desktop/System final
furkany@ubuntu:~/Desktop/System final$ time ./clients 4 6 80 5000 & ./clients 6 10 20 5000
32
of clients currently being server: 1
of clients currently being server: 2
of clients currently being server: 3
of clients currently being server: 4
of clients currently being server: 5
of clients currently being server: 6
of clients currently being server: 7
of clients currently being server: 8
of clients currently being server: 9
of clients currently being server: 10
of clients currently being server: 11
of clients currently being server: 12
of clients currently being server: 13
of clients currently being server: 14
of clients currently being server: 15
of clients currently being server: 16
of clients currently being server: 17
of clients currently being server: 18
of clients currently being server: 19
of clients currently being server: 20
of clients currently being server: 1
of clients currently being server: 2
server 5000 100
/System final$ ./server 5000 100
#include <sys/shm.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/msg.h>
inverse
15 matches


Terminal
8:51 AM
makefile
server.c
timeServer.c
clients.log
port.c
QR.c
try.c
QR.h
valgrid
1 #define DEBUG 0
ubuntu-furkany@ubuntu: ~/Desktop/System final
of clients currently being server: 62
of clients currently being server: 63
of clients currently being server: 64
of clients currently being server: 65
of clients currently being server: 66
of clients currently being server: 67
of clients currently being server: 68
of clients currently being server: 69
of clients currently being server: 70
of clients currently being server: 71
of clients currently being server: 72
of clients currently being server: 73
of clients currently being server: 74
of clients currently being server: 75
of clients currently being server: 76
of clients currently being server: 77
of clients currently being server: 78
of clients currently being server: 79
of clients currently being server: 80
0m0.469s
0m0.004s
0m0.008s
furkany@ubuntu:~/Desktop/System final$
ubuntu-furkany@ubuntu: ~/Desktop/System final
ubuntu-furkany@ubuntu:~/Desktop/System final$ ./server 5000 100
thread poola girdi k:100
^CCTRL^C catched!ubuntu-furkany@ubuntu:~/Desktop/System final$
#include <sys/shm.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/msg.h>
inverse
15 matches


Terminal
8:51 AM
makefile
server.c
timeServer.c
clients.log
port.c
QR.c
try.c
QR.h
valgrid
1 #define DEBUG 0
ubuntu-furkany@ubuntu: ~/Desktop/System final
furkany@ubuntu:~/Desktop/System final$ time ./clients 4 6 80 5000
of clients currently being server: 1
of clients currently being server: 2
of clients currently being server: 3
of clients currently being server: 4
of clients currently being server: 5
of clients currently being server: 6
of clients currently being server: 7
of clients currently being server: 8
of clients currently being server: 9
of clients currently being server: 10
of clients currently being server: 11
of clients currently being server: 12
of clients currently being server: 13
of clients currently being server: 14
of clients currently being server: 15
of clients currently being server: 16
of clients currently being server: 17
of clients currently being server: 18
of clients currently being server: 19
of clients currently being server: 20
of clients currently being server: 21
of clients currently being server: 22
of clients currently being server: 23
ubuntu-furkany@ubuntu: ~/Desktop/System final
ubuntu-furkany@ubuntu:~/Desktop/System final$ ./server 5000 100
thread poola girdi k:100
^CCTRL^C catched!ubuntu-furkany@ubuntu:~/Desktop/System final$
#include <sys/shm.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/msg.h>
inverse
15 matches


Terminal
8:50 AM
makefile
server.c
timeServer.c
clients.log
port.c
QR.c
try.c
QR.h
valgrid
1 #define DEBUG 0
ubuntu-furkany@ubuntu: ~/Desktop/System final
furkany@ubuntu:~/Desktop/System final$ time ./clients 4 6 4 5000
of clients currently being server: 1
of clients currently being server: 2
of clients currently being server: 3
of clients currently being server: 4
0m0.009s
0m0.000s
0m0.000s
furkany@ubuntu:~/Desktop/System final$
ubuntu-furkany@ubuntu: ~/Desktop/System final
ubuntu-furkany@ubuntu:~/Desktop/System final$ make
gcc -c server.c
gcc -o server server.o -pthread -lm
gcc -c clients.c
gcc -o clients clients.o -pthread -lm
ubuntu-furkany@ubuntu:~/Desktop/System final$ ./server 5000 0
^CCTRL^C catched!ubuntu-furkany@ubuntu:~/Desktop/System final$
#include <sys/shm.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/msg.h>
inverse
15 matches