

SYSTEM PROGRAMMING FINAL REPORT

Furkan Erdöl 131044065

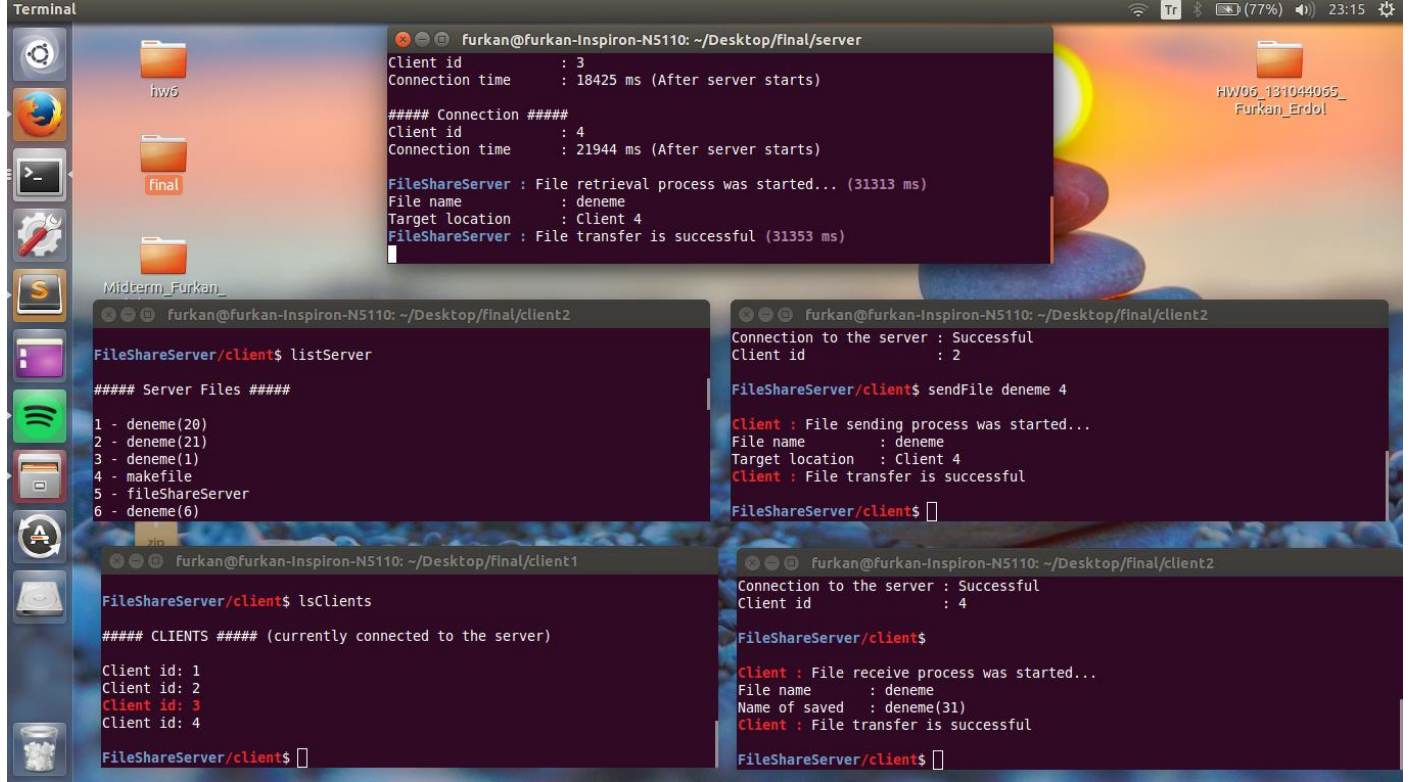
Final projesine başlarken midterme başlarken olduğu kadar zorlanmadım. Midterm projesinde Client-server yapısı hakkında bir çok şey öğrendiğimi düşünüyorum. Bu projeye başlarken socket yapısı üzerinde bilgim çok azdı. Öncelikle işe socket yapısı hakkında araştırma yaparak başladım. Sonrasında projede istenen “file transfer protocol” konusunda internet üzerinden bir çok kaynaktan araştırmalar yaptım. Bu iki konu hakkında final projesine başlamak için yeterli bilgim artık vardı. Sonrasında belki de en önemli kısım olan tasarım konusu üzerinde kafa yordum. Bu proje için kodlama kısmını daha rahat yapabileceğim aynı zamanda performans ve verimlilik konusunda elimden geleni yapabileceğim bir tasarım oluşturdum.

İlk önce bu projede kullanacağım componentleri belirledim. Fork yerine thread kullanmaya karar verdim. Bu durumun hem maliyet hem performans konusunda daha verimli olacağını düşündüm. Threadler üzerine çok pratik yapamadığım için sınırlı bilgim vardı. Bu projede thread kullanmam benim içinde threadlerin çalışma mantığını tam anlamıyla öğrenmek ve pratik yapmak için iyi bir şans oldu. Kullanacağım yapıları da belirledikten sonra artık tam anlamıyla kodlama kısmına yoğunlaşmanın zamanı geldi.

Algoritma

Client-server arası iletişimi socket bağlantısı ile sağladım. Başlarken server ın bir port numarasına client ın ise server ın ip adresi ve port numarasına ihtiyacı var. Server a bağlanan clientların bilgisini bir array de tuttum. Array in index sayılarının bir fazlası da client ın id numarasını temsil ediyor. İletişim ve senkronizasyon için gerekli olan mutex, socket, ve threadlerin bilgilerini de bu client array ine paralel farklı array lerde tuttum. Server a bağlanan her bir client için onunla ilgilenmesi gereken helper server adı altında bir thread oluşturdum. Artık client ın isteklerini bu thread yerine getirecek. Dosya gönderme kısmını ise byte byte okuduğum dosyayı server a gönderdim. Eğer başka bir client a göndermek istiyorsam da bunu server a belirterek yine server üzerinden transfer işlemi gerçekleştirdim. Client kısmında ise main thread im istekleri commend olarak alıyor. Başlangıçta bir de thread oluşturunca. Bu thread in amacı sürekli server ı dinleyerek server dosya göndermek istediği zaman veya başka bir message gönderdiği zaman bu thread işlemi gerçekleştiriyor. Server ve client tarafında senkronizasyon işlemi mutex ile sağladım. Server da her client için bir mutex oluşturdum. Client tarafında ise kedisini için bir tane mutex oluşturdum. Sinyal mevzularına gelirse elimden geldiği kadar sinyalleri ve beklenmedik bağlantı kopmaları, vs durumları handle etmeye çalıştım. Bu konu üzerinde çok durdum. Kodun mümkün olduğu kadar stabil ve hatalara kapalı olabilmesi için çaba sarfettim. Son olarak pdf te yazan bütün kurallara uydum ve gerekli bütün özellikleri client-server yapısı üzerine kurmaya çalıştım.

Bu final projesinin bana kazandırdıklarına gelecek olursam; Gerçekten faydalı bir proje olduğunu düşünüyorum. Yaparken ciddi çok zorlandım fakat gerekli component'ları öğrenip uygulamalarını doğru bir şekilde yaptıktan sonra sistem programlamanın complex fakat bir o kadar zevkli bir ders olduğunu midterm den sonra bir kez daha anladım. Bu projenin bana olumlu açıdan bir çok getirisi oldu.



```
Terminal
furkan@furkan-Inspiron-N5110: ~/Desktop/final/server
Client id      : 3
Connection time : 18425 ms (After server starts)

#### Connection ####
Client id      : 4
Connection time : 21944 ms (After server starts)

FileShareServer : File retrieval process was started... (31313 ms)
File name       : deneme
Target location  : Client 4
FileShareServer : File transfer is successful (31353 ms)

furkan@furkan-Inspiron-N5110: ~/Desktop/final/client2
FileShareServer/client$ listServer

#### Server Files ####
1 - deneme(20)
2 - deneme(21)
3 - deneme(1)
4 - makefile
5 - fileShareServer
6 - deneme(6)

furkan@furkan-Inspiron-N5110: ~/Desktop/final/client1
FileShareServer/client$ lsClients

#### CLIENTS #### (currently connected to the server)

Client id: 1
Client id: 2
Client id: 3
Client id: 4
FileShareServer/client$

furkan@furkan-Inspiron-N5110: ~/Desktop/final/client2
Connection to the server : Successful
Client id      : 2

FileShareServer/client$ sendFile deneme 4

Client : File sending process was started...
File name       : deneme
Target location  : Client 4
Client : File transfer is successful

FileShareServer/client$

furkan@furkan-Inspiron-N5110: ~/Desktop/final/client2
Connection to the server : Successful
Client id      : 4

FileShareServer/client$

Client : File receive process was started...
File name       : deneme
Name of saved    : deneme(31)
Client : File transfer is successful

FileShareServer/client$
```