

Gebze Technical University  
Computer Engineering  
CSE 344  
2019 Spring

System Final Report

Emirhan Karagözoğlu  
151044052

- BibakBOXServer

İstenen sistemin server tarafı. Temel olarak kullanıcı tarafından belirlenen thread pool size kadar thread ile kullanıcıya hizmet sunmaktır.

Kullanıcının belirlediği dosya konumunda sisteme bağlanan client dosyalarının yedeğini tutmaktadır. Servera bağlı clientlarda yapılan anlık değişimler direkt olarak server tarafına yansıtılmaktadır. Ayrıca client bağlı olmadığı zaman yaptığı değişimler, client servera bağlandığında server tarafına yansıtılmaktadır. Threadler client bağlı olduğu sürece o clienta hizmet vermekle görevlidir. Client bağlantıyı kaparana kadar yada server kapanana kadar, her thread kendi clientına hizmet vermektedir. Server başlatıldıktan sonra sonlandırmak için ctrl+c yapılmalıdır. Server kapanırken tüm bağlı clientların serverla bağlantısını koparmasını sağlar ve kendini ancak o zaman sonlandırır. Threadler paralel olarak çalışmaktadır. Thread pool size 'ından fazla client gelmesi durumunda bağlanan bu clientlar beklemede kalır ve herhangi bir thread boşta çıktığında hizmete alınır. Herbir client dosyasının içerisinde yapılan işlemleri detaylı şekilde gösteren birer log dosyası tutulmaktadır. İşlem yapılan ögenin türü (directory, file) , yapılan işlemin türü (add, remove, update) , işlemin gerçekleştirilme zamanı ve ögenin ismi log dosyasında tutulan bilgilerdir.

- BibakBOXClient

İstenen sistemin client tarafı. IP ve port ile servera bağlanarak parametre olarak verdiği directory'i server tarafına yollar. Client ta aynı server gibi sürekli çalışır haldedir ve sadece ctrl+c yapılarak durdurulur. Her 1 saniyede bir belirlenen directory'i servera yollar. Eğer bir değişiklik varsa server tarafı bunları algılar ve server tarafında bulunan client directory'sine bunları uygular.

### Genel Sistem Değerlendirmesi

Sistemin hem server hem client tarafında :

- Hiçbir memory leak veya valgrind error yoktur.
- Signal handling sorunsuz şekilde yapılmaktadır.
- Error check ve usage bulunmaktadır.

Sistem farklı parametreler ile birçok kez test edildi. Bunlardan 5 tanesi aşağıda sıralanmıştır.

- Case 1

Server thread pool size 10 ile çalıştırıldı. Ardından 10 client çalıştırıldı. Bütün clientların dosyalarının paralel olarak aynı anda kopyalandığı gözlemlendi. Server ctrl+c ile sonlandırıldığında tüm clientların sonlandırıldığı ve sonrasında serverında sonlandırıldığı gözlemlendi.

- Case 2

Server thread pool size 20 ile çalıştırıldı. Ardından 5 client çalıştırıldı. Clientler bağlı iken client dosyalarında değişiklikler yapıldı ve bunları server tarafına yansıdığı gözlemlendi. Clientlar teker teker ctrl+c ile kapatıldı. Ardında server sorunsuz şekilde kapatıldı.

- Case 3

Server thread pool size 30 ile çalıştırıldı. Ardından 25 client çalıştırıldı. Clientlar bağlı iken 10 client daha çalıştırıldı. 5 client beklemede kaldığı gözlemlendi. Çalışır haldeki 30 clientın 20 tanesi teker teker sonlandırıldı. Her client sonlandığında beklemede olan 5 client teker teker servera bağlandı. En son server ctrl+c ile sonlandırıldığında tüm bağlı clientlar sonlandırıldı ve server kapandı. Paralel olarak 30 totalde 35 tane clienta sorunsuz hizmet verildi.

- Case 4

Server thread pool size 50 ile çalıştırıldı. Ardından 40 client çalıştırıldı. Bağlı olan clientlardan bazıları ile aynı isimde clientlar bağlanmaya çalıştı ve sistem hata mesajı döndürdü. Aynı anda aynı isim ile sisteme bağlı birden fazla client bulunamaz. Ardından bağlı olan 10 client sonlandırıldı ve tekrar bağlantı kuruldu. Sorunsuz olarak çalıştı. Server sonlandırıldığında tüm bağlı clientlar sonlandırıldı ve server kapandı.

- Case 5

Server thread pool size 1 ile çalıştırıldı. Ardından farklı bilgisayardan 1 client servera bağlandı. Server çöktü client koptu. Hayatımda görmediğim hatalar gördüm.

```
--7190-- si_code=1; Faulting address: 0x7A5814C6; sp: 0x655d3e38
valgrind: the 'impossible' happened:
  Killed by fatal signal
/
host stacktrace:
```