KÜTÜPHANE UYGULAMASI

Onur Kuş, Abdülbaki Bayraktar

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[onurkus58@gmail.com](mailto:onurkus58@gmail.com), 61a.baki61@gmail.com

Özet

*Yazılım Laboratuvarı 1 dersi ikinci projesinde Nesneye Yönelik Programlama (Java), İşletim Sistemleri derslerinde edinilen bilgilerin uygulamaya geçirilmesi amaçlanmıştır.*

# Giriş

Sunucu İstek Yoğunluğu Multithread İle Kontrolü projesinde, kullanıcı ile programın arasındaki iletişimi buton yardımlarının ve görsel olarak daha etkili ve anlaşılabilir progress barların olduğu bir ara yüz pencerisi sağlar. Kullanıcı Başla butonuna tıklayarak thread işlemlerini başlatmasıyla ana ve sabit 2 alt sunucular birbiriyle uyumlu (senkronize) şeklinde çalışmaya başlar olur. Aynı anda istek alma ve isteğe cevap verme yani dönüş yapma işlemleri progress barların da varlığıyla net bir şekilde takip edilerek analizi yapılabilir. Zamanla sabit alt sunucular kapasitesin %70’ ni geçtiği anda sonradan oluşup istek almadığı zaman silinebilen alt sunucular oluşur. Sunun sayısını takip eden sayaç sayesinde çalışan anlık sunucu sayısı takip edilir. Durdur butonuyla işlemler anlık olarak durdurulur. Ara yüz penceresinin kapatılmasıyla birlikte program kapanır.

# Temel Bilgiler

Proje gelişiminde;

Tümleşik Geliştirme Ortamı olarak “NetBeans IDE 8.2” kullanılmıştır.

# Tasarım

Sunucu İstek Yoğunluğu Multithread İle Kontrolü programının programlanma aşamaları, altta belirtilen başlıklar altında açıklanmıştır.

## Yazılım Tasarımı

Sunucu İstek Yoğunluğu Multithread İle Kontrolü Projesinin yazılım aşaması bu başlık altında bulunan konular tarafınca geliştirilmiştir.

## Sınıflar

1. **Project(main)**

Main Class’ıdır. Arayuz nesnesi tanımlı olup çağrılmıştır. 10000 istek kapasitesine sahip olan 500 ms zaman aralıklarıyla [1-100] arasında rastgele sayıda istek kabul eden, istek olduğu sürece 200 ms zaman aralıklarıyla [1-50] arasında rastgele sayıda isteğe geri dönüş yapabilen ana sunucun sınıfıdır.

1. **Arayuz**

Bütün sunuculardaki işlemlerin durum bilgilerini JProgressBaryardımıyla ekrana basar. Başla ve Durdur butonlarını barındırır. Sistemde mevcut olan tüm sunucuların bilgilerini canlı olarak göstermektedir.

1. **AltSunucu**

5000 istek kapasitesine sahip olan, 500 ms zaman aralıklarıyla [1-50] arasında rastgele sayıda ana sunucudan istek alan, istek olduğu sürece 300 ms zaman aralıklarıyla [1-50] arasında rastgele sayıda isteğe geri dönüş yapan alt sunucunun sınıfıdır.

1. **AltSunucuDiger**

5000 istek kapasitesine sahip olan, 500 ms zaman aralıklarıyla [1-50] arasında rastgele sayıda ana sunucudan istek alan, istek olduğu sürece 300 ms zaman aralıklarıyla [1-50] arasında rastgele sayıda isteğe geri dönüş yapan diğer alt sunucunun sınıfıdır.

1. **YeniAltSunucu**

Mevcut olan alt sunucuları kontrol eden. Eğer herhangi bir alt sunucunun kapasitesi %70 ve üzerinde ise yeni bir alt sunucu oluşturabilen ve kapasitenin yarısınıyeni oluşturduğu alt sunucuya gönderen. Eğer herhangi bir alt sunucunun kapasitesi %0’a ulaştığında mevcut olan alt sunucu silen sınıftır.

# Kazanımlar

**1. Thread işlemleri**

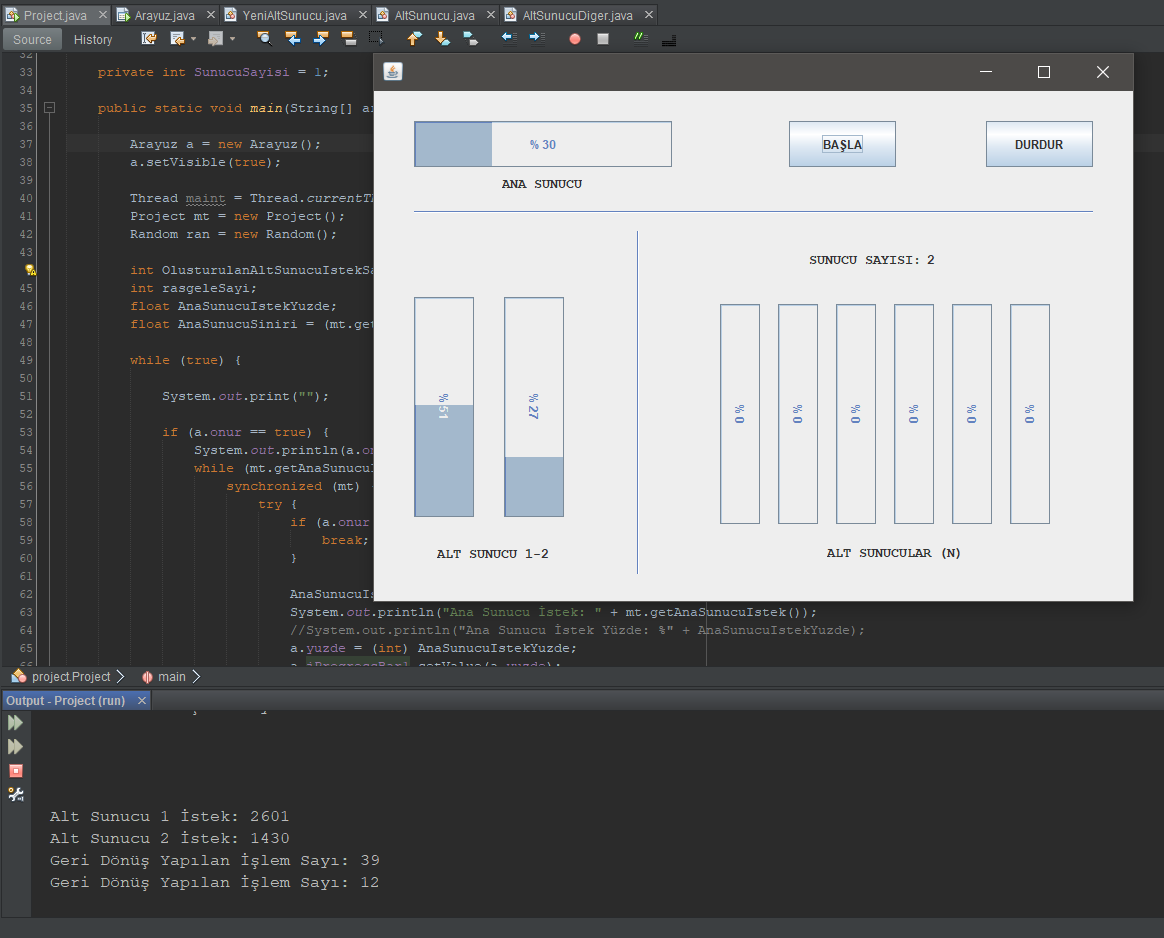
Bu proje ile, Thread kullanarak birden fazla işlemlerin kısa sürede ve birbiriyle uyumlu (senkronize) şekilde çalışmasının kullanmayı öğrendik.

**2. JFrame yapısı**

Java’da işlem verilerinden yararlanarak buton-progressbar iletişiminin jframe üzerinde kullanılması için nasıl bir yol izleneceğini öğrendik.

**3. JProgressBar yapısı**

Bu proje ile, verilerin görsel olarak daha sağlıklı gösterilmesinin ve nasıl kullanılmasının deneyimleme şansı bulduk.

****

# Algoritma

Sunucu İstek Yoğunluğu Multithread İle Kontrolü Programı, kullanıcı ara yüzünün açılmasıyla başlar. Kullanıcı Başla butonuna tıklayarak thread işlemlerini başlatmış olur. Kullanıcının butona basıp işlemlerin başlamasının akabinde arayüzdeki ana sunucu ve 2 sabit alt sunucunun progress barları artmaya başlar. Bu işlemler (istek alma-isteğe geri dönüş yapma) birbiyle uyumlu (senkronize) çalışır. Daha sonra sabit alt sunucuların istek sayısı kapasitenin %70’ ini geçip-geçmemesi kontrol edilir. Geçtiği durumlarda yeni alt sunucular oluşur. Bu oluşan alt sunucuların Alt Sunucular (N) progress barları tarafından takip edilir. Sonrada oluşan sabit sunucuların dışındaki (child) alt sunucuların sabit alt sunuculardan farkı istek sayıları sıfırlandığı esnada silinir (thread) progressbarın üzerinden kısacası %0 düşer. Kullanıcı isteğine bağlı olarak ara yüz penceresindeki Durdur butonuna tıklanarak işlemler anlık durumda durur. Ara yüz penceresinin kapatılmasıyla program durur.

# Ekran Çıktısı

# Akış Diyagramı

# Kaynakça

1. <https://www.youtube.com/watch?v=tLZT-S_CTh8>
2. <https://www.youtube.com/watch?v=_JYrDNWoiXs>
3. <https://medium.com/gokhanyavas/javada-multithreading-bbc6a9181772>
4. <https://www.geeksforgeeks.org/multithreading-in-java/>

1. <https://www.youtube.com/watch?v=Vjd-Enj48ls>
2. <https://www.youtube.com/watch?v=0CU7Ktk9MwU&>
3. <https://www.youtube.com/watch?v=XTQs9OvFCAQ&>
4. <https://www.youtube.com/watch?v=Ytt_ZuVOt2s&>
5. <https://www.youtube.com/watch?v=EckXuq-4Y_o&>