

MÜHENDİSLİK FAKÜLTESİ

YAZILIM MÜHENDİSLİĞİ YÜKSEK LİSANS

TEZSİZ YÜKSEK LİSANS DÖNEM PROJESİ

C# VE JAVA DİLLERİNDE ASYNC PROGRAMLAMANIN PERFORMANSA OLAN ETKİSİNİN İNCELENEMESİ

HAZIRLAYAN

Türhan Yıldırım

DANIŞMAN ÖĞRETİM ÜYESİ

Dr. Öğr. Üyesi Bilgin Avenoğlu

2022

# ETİK İLKELERE UYGUNLUK BEYANI

Dönem proje yazma sürecinde bilimsel ve etik ilkelere uyduğumu, yararlandığım tüm kaynakları kaynak gösterme ilkelerine uygun olarak kaynakçada belirttiğimi ve bu bölümler dışındaki tüm ifadelerin şahsıma ait olduğunu beyan ederim.

İmza Adı Soyadı

**C# VE JAVA DİLLERİNDE ASYNC PROGRAMALAMANIN PERFORMANSA OLAN ETKİSİNİN İNCELENMESİ**

**Türhan Yıldırım**

**AHMET YESEVİ ÜNİVERSİTESİ**

**YAZILIM MÜHENDİSLİĞİ YÜKSEK LİSANS BÖLÜMÜ**

**2022**

# ÖZET

**Anahtar Kelimeler:** Sync, Async, Programalama, Performans

**Danışman:** Dr. Öğr. Üyesi Bilgin Avenoğlu

**DÖNEM PROJESİNİN İNGİLİZCE ADI BU BÖLÜME YAZILMALIDIR**

**Türhan Yıldırım**

**AHMET YESEVI UNIVERSITY**

**SOFTWARE ENGINEERING MASTER’S DEPARTMENT**

**2022**

# ABSTRACT

**Keywords:** Sync, Async, Programming, Performance

**Advisor:** Dr. Öğr. Üyesi Bilgin Avenoğlu

# İÇİNDEKİLER

[ETİK İLKELERE UYGUNLUK BEYANI ii](#_Toc511318601)

[ÖZET iii](#_Toc511318602)

[ABSTRACT iv](#_Toc511318603)

[İÇİNDEKİLER v](#_Toc511318604)

[ŞEKİLLER LİSTESİ vi](#_Toc511318605)

[TABLOLAR LİSTESİ vii](#_Toc511318606)

[SİMGELER VE KISALTMALAR viii](#_Toc511318607)

[BÖLÜM I GİRİŞ 1](#_Toc511318608)

[1.1. Problem 1](#_Toc511318609)

[1.2. Araştırmanın Amacı 1](#_Toc511318610)

[1.3. Araştırmanın Önemi 1](#_Toc511318611)

[1.4. Sayıltılar 1](#_Toc511318612)

[1.5. Sınırlılıklar 1](#_Toc511318613)

[1.6. Tanımlar 1](#_Toc511318614)

[BÖLÜM II KAVRAMSAL ÇERÇEVE 2](#_Toc511318615)

[2.1. Literatür Taraması 2](#_Toc511318616)

[BÖLÜM III YÖNTEM 3](#_Toc511318619)

[3.1. Araştırmanın Modeli 3](#_Toc511318620)

[3.2. Evren ve Örneklem 3](#_Toc511318621)

[3.3. Veri Toplama Araçları 3](#_Toc511318622)

[3.4. Verilerin Toplanması 3](#_Toc511318623)

[3.5. Verilerin Analizi 3](#_Toc511318624)

[BÖLÜM IV BULGULAR VE YORUM 4](#_Toc511318625)

[4.1. Birinci araştırma sorusuna(alt problem) ilişkin bulgular. 4](#_Toc511318626)

[4.2. İkinci araştırma sorusuna(alt problem) ilişkin bulgular. 4](#_Toc511318627)

[4.3. Üçüncü araştırma sorusuna(alt problem) ilişkin bulgular. 4](#_Toc511318628)

[BÖLÜM V SONUÇ, TARTIŞMA VE ÖNERİLER 5](#_Toc511318629)

[5.1. Sonuç 5](#_Toc511318630)

[5.2. Tartışma 5](#_Toc511318631)

[5.3. Öneriler 5](#_Toc511318632)

[KAYNAKÇA 6](#_Toc511318633)

[EKLER 7](#_Toc511318634)

# ŞEKİLLER LİSTESİ

# TABLOLAR LİSTESİ

# SİMGELER VE KISALTMALAR

# BÖLÜM I **GİRİŞ**

## Problem

Günümüzde yeni trend olan asenkron(async) yazılım geliştirme mantığı popüler dillerden olan C# ve Java için gerçekten ifade edildiği gibi her koşulda performans artışı sağlamakta mıdır?

## Araştırmanın Amacı

Asenkron programlama ile vaat edilen performans artışı işlem türüne göre farklılık gösteriyor mu tespit etmek istiyoruz. İşlem gücü (CPU) gerektiren ve dosya okuma-yazma (File IO) işlemi yapan işlemler karşılaştırılacaktır.

## Araştırmanın Önemi

Gerçek bir performans artışı sağlanabilecek mi bilinmeden geliştirilen yazılım ilk günden asenkron(async) olarak yazılmaya çalışılmakta ve bu da ek geliştirme sürecin de hem sürenin hem de kompleksitenin artmasına sebep olmaktadır.

## Sayıltılar

Herhangi bir sayıltı bulunmamaktadır.

## 1.5. Sınırlılıklar

Asenkron olarak geliştirilmek istenen tüm yazılımlar için geçerli bir durumdur.

## Tanımlar

Aynı işlemi defalarca kez tekrar eden ya da ya da çok fazla işlem yükü olan uygulamalar/yazlımlar için bu durum geçerlidir. Tek kullanıcısı olan, zaman sınırı olmayan yani işlemin ne kadar sürede yapıldığının önemi olmayan yazılımlar için geçerli değildir.

# BÖLÜM II KAVRAMSAL ÇERÇEVE

## 2.1. Literatür Taraması

Literatür taramalarına göre asenkron programlama üzerine resmi çokça yazı/makale bulunmasına rağmen uygulamalar üzerinde direk oluşacak performans etkisi için net bir açıklama bulunmamakta.

Genel literatür asenkron programlama modelleri ve uygulanışları üzerine yoğunlaşırken, linkte belirttiğim araştırma doğru bir şekilde geliştirilen asenkron uygulamalar için %30,7 gibi çok ciddi bir performans kazancı olabileceğini bildirmektedir.(Castillo et al., 2019)

Asenkron işlem mantığına istinaden yayınlanan başka bir yayımda ise, asenkron programlama algoritmalarının senkronize işlem algoritmalarına göre 1,5-12 kat arasında performans kazancı sağladığını göstermektedir.(Shi et al., 2019)

# BÖLÜM III **YÖNTEM**

## 3.1. Araştırmanın Modeli

Çalışmanın yapılacağı bilgisayar bilgileri

* AMD Ryzen 3900X (12 Fiziksel Çekirdek) İşlemci
* 32 GB 3200MHZ DDR4 Ram
* Samsung 970Evo Plus 512GB SSD Sabit Disk
* Windows 11 (21H2 – OS Build 22000.613) İşletim sistemi

C# için geliştirilecek uygulama ile ilgili bilgiler

* Visual Studio 2022 (17.1.5)
* .NET 6.0
* .NET SDK Version 6.0.202
* Console uygualması yazılacak
* Asenkron kısmında ilgili methodlar Task.Run şeklinde çağrılacaktır

Java uygulaması için kullanılacak uygulama bilgileri

* Daha sonra eklenecektir

## 3.2. Evren ve Örneklem

Veri toplama methodlarından elde edilecek sonuçlar kullanılacaktır.

## 3.3. Veri Toplama Araçları

Veri toplanması için 2 temel method yazılacak.

* İşlemci yükü gereken method, içeriğinde basitçe 1’den 100’e kadar olan sayıların karelerini hesaplayacak
* Dosya okuma işlemi yapacak olan method, içeriğinde 1MB metinsel veri içeren bir txt dosyasını okuyacak

Bu iki temel method için C# ve Java dillerinde, ilgili methodları 1000’er kez senkron ve asenkron çağıran methodlar hazırlanacak. Bu işlem her bir durum için 10’ar kez tekrarlanacaktır.

* Senkron bir şekilde 1’den 100’e kadar olan sayıların karelerini hesaplayan method 1 dakika boyunca çağrılacaktır ve işlemin kaç milisaniyede tamamlandığı alınacaktır.
* Senkron bir şekilde 1MB metinsel veri içeren txt dosya okunacak ve işlemin kaç milisaniyede tamamlandığı alınacaktır.
* Asenkron bir şekilde 1’den 100’e kadar olan sayıların karelerini hesaplayan method 1 dakika boyunca çağrılacaktır ve işlemin kaç milisaniyede tamamlandığı alınacaktır.
* Asenkron bir şekilde 1MB metinsel veri içeren txt dosya okunacak ve işlemin kaç milisaniyede tamamlandığı alınacaktır.

## 3.4. Verilerin Toplanması

## 3.5. Verilerin Analizi

C# uygulaması için elde edilen veriler alttadır

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dosya Okuma (milisaniye) | | Hesaplama (milisaniye) | |
| Deneme | Senkron | Asenkron | Senkron | Asenkron |
| 1 | 1407 | 1430 | 1603 | 59 |
| 2 | 1101 | 1473 | 1626 | 3 |
| 3 | 1071 | 1414 | 1620 | 3 |
| 4 | 1075 | 1493 | 1617 | 3 |
| 5 | 1066 | 1383 | 1643 | 2 |
| 6 | 1062 | 1438 | 1586 | 2 |
| 7 | 1072 | 1423 | 1595 | 2 |
| 8 | 1466 | 1397 | 1570 | 2 |
| 9 | 1437 | 1439 | 1604 | 3 |
| 10 | 1540 | 1444 | 1608 | 2 |
| Ortalama | 1229.7 | 1433.4 | 1607.2 | 8.1 |

Java uygulaması için elde edilen veriler alttadır

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dosya Okuma (milisaniye) | | Hesaplama (milisaniye) | |
| Deneme | Senkron | Asenkron | Senkron | Asenkron |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| Ortalama |  |  |  |  |

# BÖLÜM IV BULGULAR VE YORUM

## 4.1. Birinci araştırma sorusuna(alt problem) ilişkin bulgular.

## 4.2. İkinci araştırma sorusuna(alt problem) ilişkin bulgular.

## 4.3. Üçüncü araştırma sorusuna(alt problem) ilişkin bulgular.

.

.

.

.

.

# BÖLÜM V SONUÇ, TARTIŞMA VE ÖNERİLER

## 5.1. Sonuç

## 5.2. Tartışma

## 5.3. Öneriler

# KAYNAKÇA

Castillo, E., Jain, N., Casas, M., Moreto, M., Schulz, M., Beivide, R., Valero, M., & Bhatele, A. (2019). Optimizing computation-communication overlap in asynchronous task-based programs. *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPOPP*. https://doi.org/10.1145/3293883.3295720

Shi, H., Zhao, Y., Zhang, B., Yoshigoe, K., & Chang, F. (2019). Effective Parallel Computing via a Free Stale Synchronous Parallel Strategy. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2936820

# EKLER

Graphical user interface, text, application

Description automatically generatedC# Uygulama Kodları

Graphical user interface, text, application

Description automatically generatedSync Akış Uygulama Kodları

Graphical user interface, application

Description automatically generatedAsync Akış Uygulama Kodları

Graphical user interface, text, application

Description automatically generatedAna İşlem Method Kodları

Örnek Test Çıktısı

Graphical user interface

Description automatically generated with low confidence