



**T.C.**  
**ONDOKUZMAYIS ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**VERİ YAPILARI ÖDEV RAPORU**



**Ödev Başlığı: Kuyruk Veri Yapısına Bilgisayar Biliminden Örnek Verme, Bir Örneği Gerçekleme, Öncelikli Kuyruk (Priority Queue) Hakkında Araştırma**

**Ürünün İçermesi Gereken Özellikler: Seçilen Örneğin Desteklediği Operasyonlar**

**Ödev Kazanımları: Kuyruk Veri Yapısı ve Uygulamaları**

**Yapılan Ürünün Özellikleri: Temel Kuyruk Fonksiyonları**

**Ürünün Kaynak Kodu ve Çalıştırılabilir Dosyaları (Linux ve Windows) “21060619\_alper\_karaca.zip” İsimli Sıkıştırılmış Klasörde Mevcuttur.**

# Kuyruk Yapısının Bilgisayar Biliminde Kullanımı

Kuyruk veri yapısının Bilgisayar Biliminde kullanım alanlarına ait örnekler:

- Web sitesine gönderilen isteklerin işlenme süreci.
- WI-FI üzerinden gönderilen paketlerin işleme alınma süreci.
- İşletim sistemlerinde bulunan işlem önceliklendirme sistemi.
- Medya oynatma uygulamalarındaki kuyruk sistemi. (Örn: Spotify)
- Yazıcılardaki yazdırma kuyruğu.

## Web Sitesine Gönderilen İsteklerin İşlenme Süreci:

Kullanıcının cihazından gönderilen paket ağ vasıtasıyla ağ sunucusuna ulaşır ve buradaki bekleme listesinde tutulur. Sunucu kapasitesi doluysa eksilen kısım yeni kullanıcının erişimine ulaşır ve erişim sağlanmış olur.

## WI-FI Üzerinden Gönderilen Paketlerin İşleme Alınma Süreci:

WI-FI teknolojisi radyo frekansları üzerinden çalışır ve günümüzde güncel olarak 2.4GHz ve 5GHz frekanslarında yer almaktadır.

WI-FI 6 teknolojisine kadar gönderilen bütün işlemler gönderim sırasına göre yapılmaktaydı örnek vermek gerekirse :

Telefondan Facebook'a erişmek için gönderilen paketle TV'den Netflix yayınına erişmek için gönderilen paket aynı anda geldiğinde önce birisinin tamamlanması gerekiyor sonra diğer işlem yapılmıyordu yani her seferde tek işlem yapılmıyordu. Bu sebeple gecikmeler oldukça artıyor ve tecrübemizi azaltıyordu. WI-FI 6 ile de bu sorunun önüne geçilmiştir. Gerçekleştirdiğim örnek ise bu sistemdir.

## İşletim Sistemlerindeki Önceliklendirme Sistemi:

İşletim sistemlerinde de işlemler sırayla yapılmaktadır. Eğer bir programı açmak için işlem yaptıysanız önce o işlem yapılır sonra diğer işlem vs. vs. diye gider.

Sistem monitörünü kullandığımız zaman ise bu işlemlere öncelik verilebilir ki burada da öncelikli kuyruk işleme girmektedir. Öncelikli kuyruktan ilereleyen kısımda bahsedeceğim.

## Medya Oynatıcı Programlardaki Kuyruk Sistemi:

Örnek olarak Spotify'i ele alalım. Bir şarkı dinlerken seçenekler arasında başka bir şarkıyı sıraya ekleme seçeneği var. Bu seçenek sayesinde bizim seçtiğimiz o şarkı uygulamanın yapmış olduğu listede en öne alınarak ilk çalınacak şekilde ayarlanır.

## Yazdırma Kuyruğu:

Yazıcıya bir belge gönderildiği zaman o belgenin sonuna gelinmeden (Yazdırma işlemi bitmeden) sırada bekleyen belge yazdırmaya başlanmaz. (Bu durum tek bilgisayar üzerinden gelen bilgiler için geçerli. Eğer bir yazıcıya iki farklı bilgisayardan birer belge gönderilirse bir belge basılırken arada diğer belgeyi de basar.)

# Öncelikli Kuyruk Sistemi Nedir ?

Öncelikli kuyruk sistemi temelde bildiğimiz kuyruk veri yapısı ile aynı olmasına karşın içerisinde bir önceliklendirme sistemi barındırır ve işlemler bu sisteme göre yapılır.

Örnek vermek gerekirse yemek kuyruğunda oluşan kuyruğa girecekken tanıdık birisi varsa onun yanına gidip diğerlerinin önüne geçmek.

Bilgisayarı kapatmadan önce açık işlemlerin kapatılması vb.

Bankalardaki sıra sistemi de buna benzer. Seçtiğiniz alana göre bir numara verilir ve n tane vezneden hangisinde size sıra gelirse oraya gidersiniz.

## Kuyruk Veri Yapısı Sorunları Nasıl Çözer?

Kuyruk dediğimiz veri yapısı “İlk Giren İlk Çıkar(FIFO)” prensibiyle çalışır yani bizim yapılmasını istediğimiz ilk işlem en başa gönderilir ve bekletilir ardından diğer işlemler bunun ardı sıra dizilir. Ve işlem sırasına göre tamamlanır. Bu veri yapısının gerçek hayattan örnekleri ise asansörler, hastane sıraları denebilir.

Asansörlerde öncelikli kuyruk sistemini özetlemek gerekirse şöyle bir örnek geçelim:

Bir x kişisi 6 katlı apartmanın 3. katında otururken y kişisi ise 4. katta oturuyor olsun. Asansöre ilk binen x kişisi 0'a basıyor ve bu sırada y kişisi ise asansörü çağırıyor. Kapı kapanmadan (işlemler tamamlanmaya başlamadan önce) x kişisi önce basmış olsa bile asansör 4. kata çıkıp y kişisini alır ardından eğer y kişisi 2.katta inecekse önce 2.katta durur sonra 0'a ulaşır. Burada öncelikli kuyruk yapısını örnekleyen ise 0'a basılma işlemi ilk yapılmış olsa bile en son tamamlanan oldu diyebiliriz yani önceliği düşüktü.

# WI-FI Sisteminin Gerçeklenmesi

**private static int Menu()**

```
Menu()

private static int Menu()
{
    Console.Clear();
    Console.WriteLine(

"|-----|"

    );
    Console.WriteLine(
        "|1)Router'e İşlem Gönder\t2)Bekleyen İşlem Sayısını Görüntüle\t3)Sıradaki İşlemi Görüntüle|"
    );
    Console.WriteLine(
        "|4)Sıradaki İşlemi Tamamla\t5)Sıradaki İşlemleri Görüntüle\t6)Bütün İşlemleri Tamamla  |"
    );
    Console.WriteLine("|7)İşlem Araması Yap \t\t\t\t\t8)Çıkış Yap\t\t  |");
    Console.WriteLine(

"|-----|"

    );
    Console.Write("İşlem: ");
    return Convert.ToInt32(Console.ReadLine());
}
```

**Kullanıcının işlemlerini yapması için gereken ana menüyü oluşturdum ve seçeneği değer olarak döndürmesini sağladım. Ayrıca Console.Clear() ile de menü her yüklendiği zaman konsol ekranını temizleyerek kullanıcıya güzel bir ortam ayarladım.**

## public static void Main()

```
public static void Main()
{
    //Bu örnek WIFI-5 ve öncesi nesiller için geçerlidir. WIFI-6 ile paketler senkron şekilde
    işlendiğinden kuyruk yapısına olan benzerliği de azalmıştır.
    Queue wifi_queue = new();
    while (true)
    {
        int m_select = Menu();
        switch (m_select)
        {
            case 1:
                Console.Write("İşlem Adı: ");
                wifi_queue.Enqueue(Console.ReadLine());
                Console.WriteLine("Devam Etmek İçin Herhangi Bir Tuşa Basınız");
                Console.ReadKey();

                break;
            case 2:
                Console.WriteLine($"Sırada Bekleyen İşlem Sayısı: {wifi_queue.Count}");
                Console.WriteLine("Devam Etmek İçin Herhangi Bir Tuşa Basınız");
                Console.ReadKey();
                break;
            case 3:
                if (wifi_queue.Count > 0)
                    Console.WriteLine($"Sıradaki İşlem: {wifi_queue.Peek()}");
                else
                    Console.WriteLine("Sırada İşlem Yok.");
                Console.WriteLine("Devam Etmek İçin Herhangi Bir Tuşa Basınız");
                Console.ReadKey();
                break;
        }
    }
}
```

İlk olarak wifi\_queue adında yeni bir kuyruk yapısı oluşturdum ve yapılacak işlemler ekranını sürekli çalışacak şekilde ayarladım. m\_select adındaki değişkende Menu() fonksiyonunu çalıştırarak kullanıcıya menüyü çıkarttım ve işlem seçtirttim.

### 1.Seçenek Seçilirse:

Kullanıcıda gönderilecek işlem ya da paketin adını alıyor ve bunu kuyruğa ekliyor. Ardından herhangi bir tuşa basmasının ardından ana menüye dönüyor.

### 2.Seçenek Seçilirse:

Kullanıcıya sırada bekleyen toplam işlem sayısını gösteriyor ve herhangi bir tuşa basılmasının ardından ana menüye dönüyor.

### 3.Seçenek Seçilirse:

İlk sıradaki işlemi gösteriyor ve herhangi bir tuşa basılmasının ardından ana menüye dönüyor.

```
main()

case 4:
    if (wifi_queue.Count > 0)
        Console.WriteLine($"Tamamlanan İşlem: {wifi_queue.Dequeue()}");
    else
        Console.WriteLine("Sırada İşlem Yok.");
    Console.WriteLine("Devam Etmek İçin Herhangi Bir Tuşa Basınız");
    Console.ReadKey();
    break;
case 5:
    if (wifi_queue.Count > 0)
    {
        foreach (object obj in wifi_queue)
            Console.WriteLine(obj.ToString());
    }
    else
        Console.WriteLine("Sırada İşlem Yok.");
    Console.WriteLine("Devam Etmek İçin Herhangi Bir Tuşa Basınız");
    Console.ReadKey();
    break;
case 6:
    wifi_queue.Clear();
    Console.WriteLine("Tüm İşlemler Tamamlandı !");
    Console.WriteLine("Devam Etmek İçin Herhangi Bir Tuşa Basınız");
    Console.ReadKey();
    break;
```

#### 4.Seçenek Seçilirse:

Öncelikle kuyruğu kontrol ediyor. (Boş kuyruktan eleman silmeye çalışılırsa Unhandled Exception hatası ile uygulama çöker.)

Eğer kuyrukta yapılacak iş varsa baştaki işi yapıp hangi iş olduğunu gösteriyor. Eğer kuyruk boşsa yapılacak işlem olmadığını belirtiyor.

İki işlemden biri yapıldıktan sonra da herhangi bir tuşa basılmasının ardından ana menüye dönüyor.

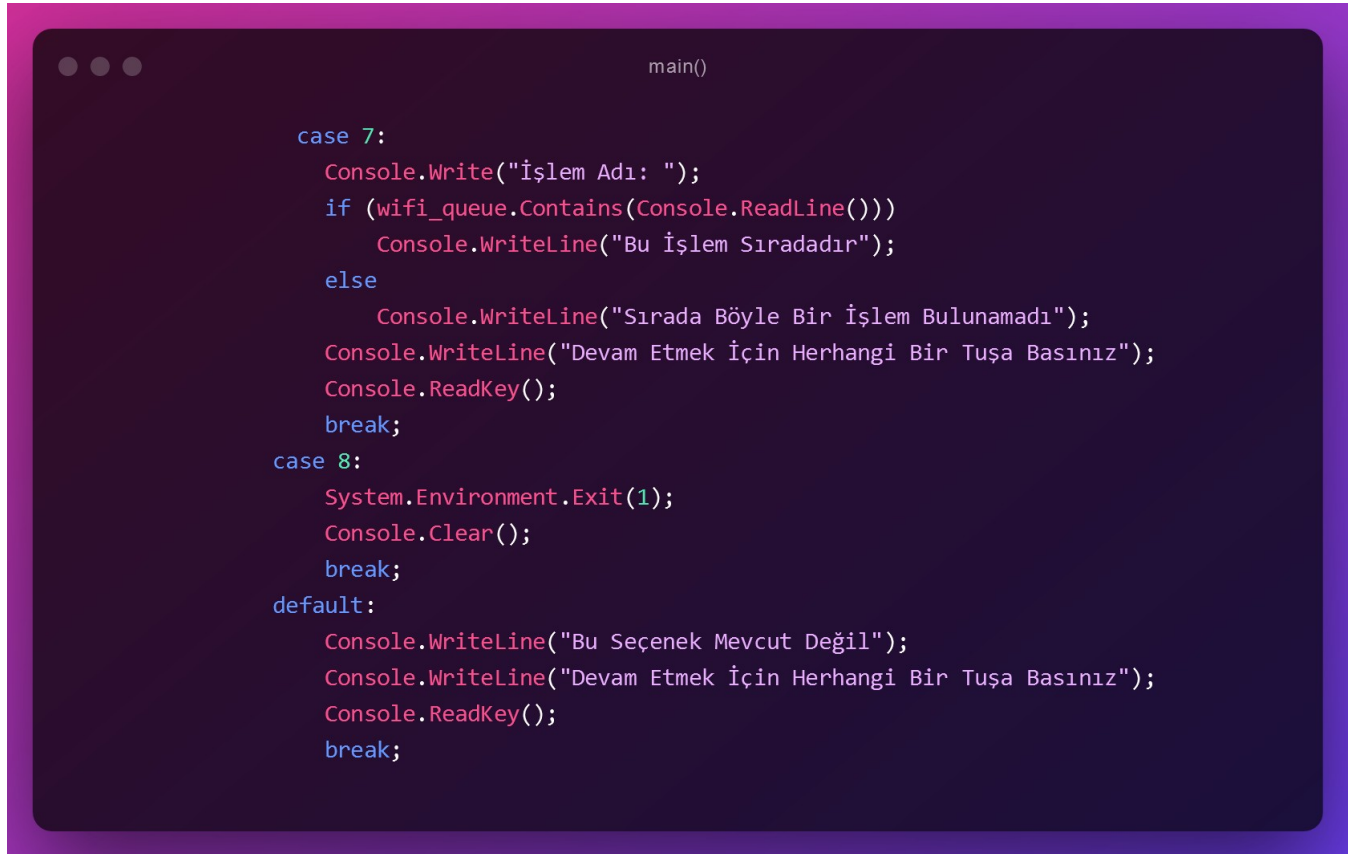
#### 5.Seçenek Seçilirse:

Kuyruk kontrol ediliyor ve işlem varsa sıradaki işlemler basılıyor eğer işlem yoksa işlem yok mesajı veriliyor.

İki işlemden biri yapıldıktan sonra da herhangi bir tuşa basılmasının ardından ana menüye dönüyor.

#### 6.Seçenek Seçilirse:

**Kuyruktaki tüm işlemleri siliyor (tamamlıyor).**



```
main()

case 7:
    Console.Write("İşlem Adı: ");
    if (wifi_queue.Contains(Console.ReadLine()))
        Console.WriteLine("Bu İşlem Sıradadır");
    else
        Console.WriteLine("Sırada Böyle Bir İşlem Bulunamadı");
    Console.WriteLine("Devam Etmek İçin Herhangi Bir Tuşa Basınız");
    Console.ReadKey();
    break;
case 8:
    System.Environment.Exit(1);
    Console.Clear();
    break;
default:
    Console.WriteLine("Bu Seçenek Mevcut Değil");
    Console.WriteLine("Devam Etmek İçin Herhangi Bir Tuşa Basınız");
    Console.ReadKey();
    break;
```

#### **7.Seçenek Seçilirse:**

Önce kuyruğu kontrol ediyor ardından verilen parametreye göre kuyrukta arama yapıyor. Eğer kuyrukta eşleşen işlem varsa bu işlemin sırada olduğunu yoksa da böyle bir işlem olmadığını belirtiyor.

İki işlemten biri yapıldıktan sonra da herhangi bir tuşa basılmasının ardından ana menüye dönüyor.

#### **8.Seçenek Seçilirse:**

Konsolu temizleyip 1 koduyla uygulamayı kapatıyor.

#### **Yanlış Seçenek Seçilirse:**

Seçenek mevcut değil çıktısını verip herhangi bir tuşa basılmasının ardından ana menüye dönüyor.

# Uygulama Çıktıları

- Ana Menü:

```
Windows PowerShell
|1)Router'e İşlem Gönder      2)Bekleyen İşlem Sayısını Görüntüle  3)Sıradaki İşlemi Görüntüle|
|4)Sıradaki İşlemi Tamamla  5)Sıradaki İşlemleri Görüntüle     6)Bütün İşlemleri Tamamla |
|7)İşlem Araması Yap       8)Çıkış Yap                          |
İşlem: |
```

- İşlem Gönder:

```
Windows PowerShell
|1)Router'e İşlem Gönder      2)Bekleyen İşlem Sayısını Görüntüle  3)Sıradaki İşlemi Görüntüle|
|4)Sıradaki İşlemi Tamamla  5)Sıradaki İşlemleri Görüntüle     6)Bütün İşlemleri Tamamla |
|7)İşlem Araması Yap       8)Çıkış Yap                          |
İşlem: 1
İşlem Adı: IP Kamera Paketi|
```

- Bekleyen İşlem Sayısı:

```
Windows PowerShell
|1)Router'e İşlem Gönder      2)Bekleyen İşlem Sayısını Görüntüle  3)Sıradaki İşlemi Görüntüle|
|4)Sıradaki İşlemi Tamamla  5)Sıradaki İşlemleri Görüntüle     6)Bütün İşlemleri Tamamla |
|7)İşlem Araması Yap       8)Çıkış Yap                          |
İşlem: 2
Sırada Bekleyen İşlem Sayısı: 2
Devam Etmek İçin Herhangi Bir Tuşa Basınız
```



- Sıradaki İşlem:

```
Windows PowerShell

1)Router'e İşlem Gönder      2)Bekleyen İşlem Sayısını Görüntüle      3)Sıradaki İşlemi Görüntüle
4)Sıradaki İşlemi Tamamla    5)Sıradaki İşlemleri Görüntüle            6)Bütün İşlemleri Tamamla
7)İşlem Araması Yap          8)Çıkış Yap

İşlem: 3
Sıradaki İşlem: IP Kamera Paketi
Devam Etmek İçin Herhangi Bir Tuşa Basınız
```

- Sıradaki İşlemleri Görüle

```
Windows PowerShell

1)Router'e İşlem Gönder      2)Bekleyen İşlem Sayısını Görüntüle      3)Sıradaki İşlemi Görüntüle
4)Sıradaki İşlemi Tamamla    5)Sıradaki İşlemleri Görüntüle            6)Bütün İşlemleri Tamamla
7)İşlem Araması Yap          8)Çıkış Yap

İşlem: 5
IP Kamera Paketi
Spotify Paketi
Devam Etmek İçin Herhangi Bir Tuşa Basınız
```

- Sıradaki İşlemi Tamamla

```
Windows PowerShell

1)Router'e İşlem Gönder      2)Bekleyen İşlem Sayısını Görüntüle      3)Sıradaki İşlemi Görüntüle
4)Sıradaki İşlemi Tamamla    5)Sıradaki İşlemleri Görüntüle            6)Bütün İşlemleri Tamamla
7)İşlem Araması Yap          8)Çıkış Yap

İşlem: 4
Tamamlanan İşlem: IP Kamera Paketi
Devam Etmek İçin Herhangi Bir Tuşa Basınız
```

- İşlem Araması Yap

```
Windows PowerShell

1)Router'e İşlem Gönder      2)Bekleyen İşlem Sayısını Görüntüle      3)Sıradaki İşlemi Görüntüle
4)Sıradaki İşlemi Tamamla    5)Sıradaki İşlemleri Görüntüle            6)Bütün İşlemleri Tamamla
7)İşlem Araması Yap          8)Çıkış Yap

İşlem: 7
İşlem Adı: Spotify Paketi
Bu İşlem Sıradadır
Devam Etmek İçin Herhangi Bir Tuşa Basınız
```

- Bütün İşlemleri Tamamla

```
Windows PowerShell

1)Router'e İşlem Gönder      2)Bekleyen İşlem Sayısını Görüntüle      3)Sıradaki İşlemi Görüntüle
4)Sıradaki İşlemi Tamamla    5)Sıradaki İşlemleri Görüntüle            6)Bütün İşlemleri Tamamla
7)İşlem Araması Yap          8)Çıkış Yap

İşlem: 6
Tüm İşlemler Tamamlandı !
Devam Etmek İçin Herhangi Bir Tuşa Basınız
```

- Çıkış Yap

```
Windows PowerShell

1)Router'e İşlem Gönder      2)Bekleyen İşlem Sayısını Görüntüle      3)Sıradaki İşlemi Görüntüle
4)Sıradaki İşlemi Tamamla    5)Sıradaki İşlemleri Görüntüle            6)Bütün İşlemleri Tamamla
7)İşlem Araması Yap          8)Çıkış Yap

İşlem: 8
r[ alpeerkaraca from Kirito][ 2:56.619s][ RAM: 12/16GB][ Saturday at 4:25:41 PM][ main ≡ ?4][ Error, check yo
ur command]
L[~\Documents\data-structures-assignment\Assignment-3\bin\Debug\net6.0]
└─┘
```

# KAYNAKÇA

<https://hasscript.com/908/öncelikli-kuyruk-prioty-queue-nedir>

<https://www.naukri.com/learning/articles/queue-data-structure-types-implementation-applications/>