

# Windows İşletim Sisteminde Esnek Zaman Paylaşımlı Proses Yönetimi Gerçekleşmesi

Onur ÜLGEN<sup>1</sup>

Mutlu AVCI<sup>2</sup>

Çukurova Üniversitesi  
Mühendislik ve Mimarlık Fakültesi  
Bilgisayar Mühendisliği Bölümü  
Balcalı / ADANA

<sup>1</sup>e-posta: onurulgen@gmail.com

<sup>2</sup>e-posta: mavci@cu.edu.tr

*Anahtar Sözcükler: Esnek Zaman Paylaşımı, Tek Yönlü Bağlı Liste, İş Sıralama, Proses Yönetimi*

## Özet

Bu çalışmada Esnek Zaman Paylaşımlı iş sıralama algoritması ve bu algoritma kullanılarak bir proses yöneticisinin Windows işletim sistemi üzerinde C yazılım dili kullanılarak nasıl yazılacağı anlatılmıştır.

## 1. Giriş

Bütün modern bilgisayarlar aynı anda pek çok işi yapabilir gibi görünmektedirler. Genellikle Merkezi İşlemci Birimi (CPU), her prosesi onlarca veya yüzlerce mikro saniye çalıştırarak anahtarlama yapmaktadır. Tek işlemcili sistemlerde aslında bir anda sadece bir işlem yapılmaktadır. Bu hızlı anahtarlama sözde bir paralelliktir. Gerçek bir paralellik ancak iki veya daha fazla işlemci içeren sistemlerde mümkündür. İşletim sistemi tasarımcıları paralellikle kolayca uğraşabilmek için sıralı işlemler modelleri geliştirmişlerdir [1].

Her proses program sayıcısı ve iç durumlarıyla bağımsız olmasına rağmen başka proseslerle ilişkiye sık sık ihtiyaç duyar. Bir proses başka bir proses için giriş olacak çıkışlar üretebilir veya bir girişi bekleyebilir. Bu nedenle her bir proses durum modeline göre; hazır, bloklamış veya çalışır hallerden birinde bulunur [2].

Tüm çalıştırılabilir yazılımlar, işletim sistemi de dâhil olmak üzere sıralı prosesler olarak organize edilir. Bir proses sadece program sayıcı, yazmaç ve değişkenlerin o anki değerlerini içererek çalışan programdır. Her proses kendi zahiri CPU'suna sahip gibidir. Gerçek CPU aslında prosesler arasında ileri geri anahtarlama yapmaktadır. CPU'nun prosesler arası paylaşılmasına iş sıralama adı verilir [3].

İş sıralamasında göz önüne alınması gerekenler; dengeli olması yani tüm proseslerin aynı etkilenmesi ve sonsuz erteleme olmaması, birim zamanda maksimum prosese izin verilmesi, etkileşimli

kullanıcıların yanıt süresinin en aza indirilmesi, kaynak kullanımının dengelenmesi, yanıt süresi ile kaynak kullanımı arasında denge sağlanmasıdır.

İş sıralamada öncelik mekanizması olmalı, sonsuz bekleme önlenmeli, kilit kaynakları kullanan prosesler hemen çalıştırılıp kaynakları bırakmaları sağlanmalı ve çok yüklü durumlarda sistemin çökmesi önlenmelidir.

İş sıralama kriterleri basitçe prosesin giriş/çıkış eğilimi, CPU kullanma eğilimi, etkileşimli yada batch mi olduğu, yanıt alma süresinin acilliği, önceliği, ne sıklıkla sayfa hatası ürettiği, daha yüksek öncelikli proses tarafından yarıda kesilme sıklığı, ne kadar gerçek yürütme zamanı elde ettiği, sonlanmak için ne kadar zamanı kaldığı gibi faktörlerdir.

Esnek Zaman Paylaşımlı iş sıralama algoritmasında işler kuyruğa alınır. Sırası gelen iş önceden belirlenen bir süre boyunca çalıştırılıp kuyruğun sonuna atılır. Bütün işler tamamlanana kadar bu işlem tekrar edilir.

Bu çalışmada amaç, Windows işletim sisteminde var olan proses yönetimini kullanmak yerine, çalıştırılacak olan proseslerin bu çalışmada geliştirilen Esnek Zaman Paylaşımlı iş sıralama algoritmasına göre yönetilmesini sağlamaktır.

Bu işlemi gerçekleştirmek için C yazılım dilinde kod geliştirilmiş, Windows işletim sistemi üzerinde çalıştırılarak Esnek Zaman Paylaşımlı Proses Yönetimi gerçekleştirilmiştir.

## 2. Esnek Zaman Paylaşımlı İş Sıralama Algoritması

Esnek Zaman Paylaşımlı iş sıralama algoritması modern işletim sistemlerinde kullanılan iş sıralama algoritmalarının temelini oluşturur. Modern işletim sistemleri birden çok prosesi aynı anda çalıştırmış gibi göstermek için işleri bir kuyruğa alıp sırası gelen işi belli bir süre boyunca çalıştırmalar. Bu süreyle

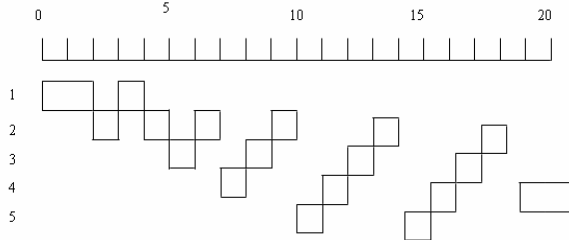
kuantum süresi denir. Bu süre insan gözüyle algılanamayacak kadar küçük olduğu için de sanki bütün prosesler aynı anda çalışıyormuş gibi görülür.

Tablo 1’de beş adet farklı proses geliş anları ve hizmet süreleri görülmektedir.

Proses	Geliş Anı (sn)	Hizmet Süresi (sn)
P1	0	3
P2	2	6
P3	4	4
P4	6	5
P5	8	2

**Tablo 1.** Prosesler, Geliş Anları ve Hizmet Süreleri

Tablo 1’de verilen bu beş adet proses için Esnek Zaman Paylaşımli iş sıralama algoritması prosesleri Şekil 1’deki gibi çalıştırır. Bu aynı zamanda günümüz modern bilgisayarlarının uygulayacağı iş sıralama yapılarından biridir.



**Şekil 1.** Dönüşümlü Bir İş Sıralama Zaman Diyagramı

Esnek Zaman Paylaşımli iş sıralama algoritması işleri bir kuyruğa alır. Kuyruk yapısı için genellikle Tek Yönlü Bağlı Listeler kullanılır.



**Şekil 2.** FIFO Tek Yönlü Bağlı Liste Yapısı

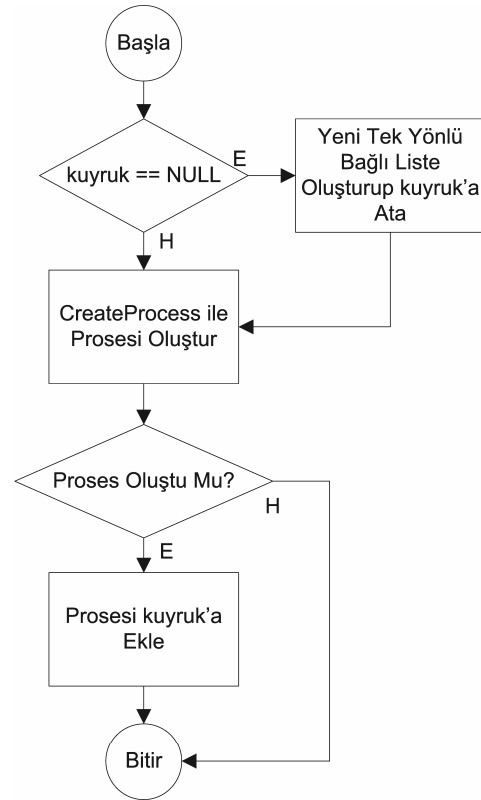
Listenin yapısı Şekil 2’deki gibidir. Yapının veri olarak adlandırılan ilk elemanı proses adresini tutar. İkinci eleman ise sonraki prosesin adresini saklayan bağlantılı liste elemanının adresini saklayan işaretçidir.

### 3. Esnek Zaman Paylaşımli Proses Yönetiminin Gerçekleştirilmesi

#### 3.1 Esnek Zaman Paylaşımli Proses Yöneticisine Proses Ekleme

Windows’a bir proses eklenmek istendiğinde CreateProcess Uygulama Programlama Arayüzü (API) çağrılır. Bu API Windows’un proses yöneticisinin kuyruğuna yeni bir proses ekler. Eğer bu API olmasaydı Windows’ta program çalıştırılmazdı.

Esnek Zaman Paylaşımli Proses Yöneticisi de benzer bir mantıkla tasarlanmıştır. Yöneticiye proses eklenmesi için bir API gereklidir yoksa yönetici hiçbir iş yapmadan bekleyecektir.



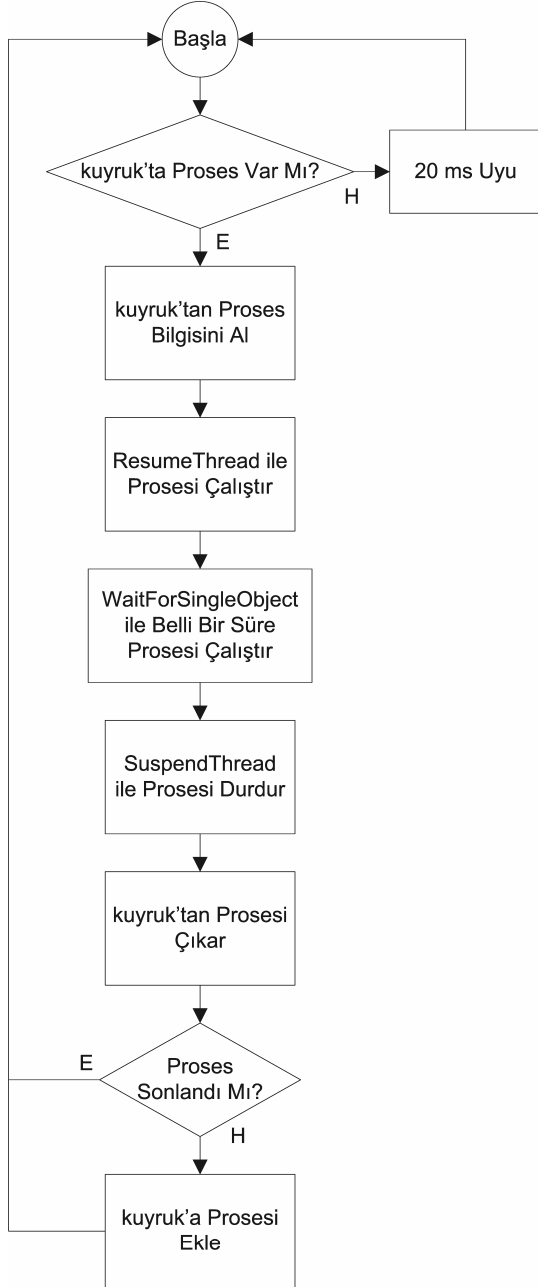
**Şema 1.** Proses Ekleme Fonksiyonunun Akış Şeması

Şema 1’deki Esnek Zaman Paylaşımli Proses Yöneticisi’nin Proses Ekleme fonksiyonu ilk önce *kuyruk* değişkeninin NULL olup olmadığını kontrol eder. Eğer NULL ise yeni bir Tek Yönlü Bağlı Liste oluşturup *kuyruk* değişkenine atar.

Esnek Zaman Paylaşımli Proses Yöneticisi prosesi kendi oluşturacağı için fonksiyon parametre olarak prosesin disk üzerinde nerede tutulduğu bilgisini alır. Bu bilgiyi kullanarak CreateProcess fonksiyonuyla prosesi durdurulmuş olarak oluşturmaya çalışır. Eğer oluşturma işlemi başarılı olursa prosesi kuyruğa ekler.

### 3.2 Esnek Zaman Paylaşımlı Proses Yöneticisi Ana Programı

Esnek Zaman Paylaşımlı Proses Yöneticisinin görevi kuyruқта sırası gelen prosesi, belli bir süre boyunca çalıştırmaktır. Bu süre sonunda eğer proses sonlanmadıysa yeniden çalıştırılmak üzere kuyruğun sonuna atılır. Bu işlem kuyruқтаki proseslerin hepsi sonlanana kadar devam eder. Kuyruқтаki prosesler bitince de kuyruğa bir proses gelene kadar uyur.



Şema 2. Ana Program Akış Şeması

Program bir sonsuz döngüde çalışmaktadır. Sürekli hafızada kalıp gelen isteklere cevap verebilmesi için böyle tasarlanmıştır.

Program, eğer *kuyruk'ta* proses yok ise 20 ms uyuyup en başa dönmektedir. 20 ms beklemesinin sebebi bu sürenin Windows'un kuantum süresi olmasıdır. Yani

Windows bu süre dolunca çalıştırdığı ipliği durdurup diğer ipliğe geçiş yapar. Hem işlemciyi meşgul etmeden hem de eklenen prosesi fazla bekletmeden çalıştırabilmek için bu süre seçilmiştir.

Program eğer kuyruk'ta proses bilgisi varsa kuyruktan proses bilgisini alır ve ResumeThread fonksiyonu ile önceden oluşturulmuş prosesi devam ettirir. WaitForSingleObject fonksiyonuyla belli bir süre prosesi çalıştırır. Bu süre dolduğunda SuspendThread fonksiyonuyla prosesi durdurur. Kuyruktan çalıştırılan prosese ait bilgileri çıkarır. Eğer proses sonlanmadıysa prosesi kuyruğun sonuna ekler.

Program tüm bu işlemleri tüm prosesler sonlanana kadar tekrar eder.

### 4. Sonuç

Esnek Zaman Paylaşımlı Proses Yöneticisi Windows işletim sisteminin ve günümüz modern işletim sistemlerinin proses yöneticilerinin en temel halidir. Geliştirilerek bu işletim sistemlerinin proses yöneticilerinin yerini alabilir.

Esnek Zaman Paylaşımlı Proses Yöneticisi prosesleri kullanıcı tarafından belirlenen bir zaman boyunca çalıştırmak için kullanılabilir. Windows'un proses yöneticisi bu zamanı kendi belirler fakat Esnek Zaman Paylaşımlı Proses Yöneticisi'nde zamanı kullanıcı belirler. Bu da daha özelleştirilebilir bir proses yöneticisi anlamına gelir.

Bu çalışmada Esnek Zaman Paylaşımlı iş sıralama algoritmaları kullanılarak proseslerin nasıl yönetileceği uygulamalı olarak anlatılmış ve Windows işletim sistemi üzerinde Esnek Zaman Paylaşımlı Proses Yöneticisi yazılımının C yazılım dili ile nasıl gerçekleştirileceği açıklanmıştır.

### Kaynaklar

- [1] A. Tanenbaum and A. Woodhull, "Operating Systems Design and Implementation", Prentice Hall, 2006
- [2] W. Stallings, "Operating Systems: Internals and Design Principles", Prentice Hall, 2004
- [3] A. Silberschatz, "Operating System Concepts", Wiley, 2004