



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**İŞLETİM SİSTEMLERİ PROJE RAPORU**

**B211210351- Muhammad Yusuf**

**SABİR**

**Y225012058- İhsan Eren**

**DELİBAŞ**

**G2312103737- Nadir ŞENSOY**

**G171210552- Shahd Numa**

**YALDANI**

**B211210565- Mikail OSMANOV**

**GitHub:** <https://github.com/KidDikson/isletim-sistemleri/blob/main/Operating-Systems-Project-main/Rapor/Odev%20Raporu%20.pdf>

**SAKARYA**

**Aralık, 2023**

**İşletim Sistemleri Dersi**

---

## Özet

Java programlama dili kullanılarak sınırlı kullanılabilir kaynakların kısıtlamaları içinde çalışan dört seviyeli öncelikli proses görevlendiricisine sahip bir çoklu programlama sistemi simüle edilmiştir.

**Not:** Ödevde istenilen jar dosyası projenin **dist** klasörü içerisinde.

**Github:** <https://github.com/KidDikson/isletim-sistemleri/blob/main/Operating-Systems-Project-main/Rapor/Odev%20Raporu%20.pdf>

Anahtar Kelimeler: Dispatcher, FCFS, proses, round robin, kuantum, geri besleme, java, zaman aşımı

---

## GELİŞTİRİLEN YAZILIM

Program çalıştırıldığında giriş dosyasından okunan prosesler Dispatcher nesnesinde bulunun ve prosesleri tutan kuyruğa yerleştirilir. Proseslerin varış zamanı, öncelik değeri, işlenme zamanı gibi özellikleri vardır. Kuyruğa yerleştirilen prosesler özelliklerine göre sıralanarak uygun öncelikli kuyruklara yerleştirilir.

### Gelen Procces'lerin Sıralanması ve Kuyruklara Alınması

Giris dosyasından gelen process'leri öncelik değerlerine göre öncelikli kuyruklara yerleştiriyoruz. Dört öncelik değeri için dört adet kuyruğumuz bulunmakta.

Ardından process'ler karışık sırayla geldiği için ArrivalTime değişkenlerine göre sıraladık. Bu bizi process'leri her seferinde aramaktan kurtardı. Time değişkenine göre ilk elemana bakılıyor eğer ilk elemanın ArrivalTime'ı, Time değerimizden büyükse devamına bakmaya gerek kalmamaktadır. Sıralama fonksiyonunun içine bakacak olursak; Gelen kuyruğu SelectionShort algoritmasıyla beraber ArrivalTime'a göre sıraladık.

### Görevlendiricinin Çalışması

Görevlendirici, sıfır öncelik değerine sahip prosesleri bulunduran kuyruğu gezmekle başlar. Görevlendiriciye gelen proses varsa bu proses FCFS algoritmasıyla işletilir. Sonlanana kadar devam eder. Bu kuyruk gezilirken aynı zamanda diğer kuyruklar da kontrol edilir. Kontrol için feedback() ve timeOutCheck fonksiyonları çağrılır. Eğer sıfır öncelikli proseslerin bulunduğu kuyrukta proses yoksa feedback fonksiyonunda öncekik sırasına göre prosesler

çalıştırılır. Kuantum değeri bir olduğu için feedback’de her proses bir saniye çalışır ve prosesin önceliği düşürülerek askıya alınır. Askıya alınan prosesler farklı bir kuyrukta tutulur ve tekrar sıra kendine gelene kadar o kuyrukta bekler.

Tüm bu işlemler timer değişkeniyle senkronize edilir. `timeOutCheck` fonksiyonuyla da yirmi saniye çalışmayan prosesler tespit edilerek kuyruktan çıkarılır ve prosesin zaman aşımına uğradığını gösteren mesajlar konsola yazılır.

## Olası İyileştirmeler

Görevlendiricide kullanılan sıralama algoritmasında daha efektif bir seçim yapabiliriz. Onun haricinde görevlendirici şuanda sadece 4 seviye için çalışmaktadır daha yüksek seviyelerde ek olarak kod yazmamız gerekli ve öncelik değiştirmek istediğimizde maliyeti yüksek olabilir.

## Procces Sınıfı

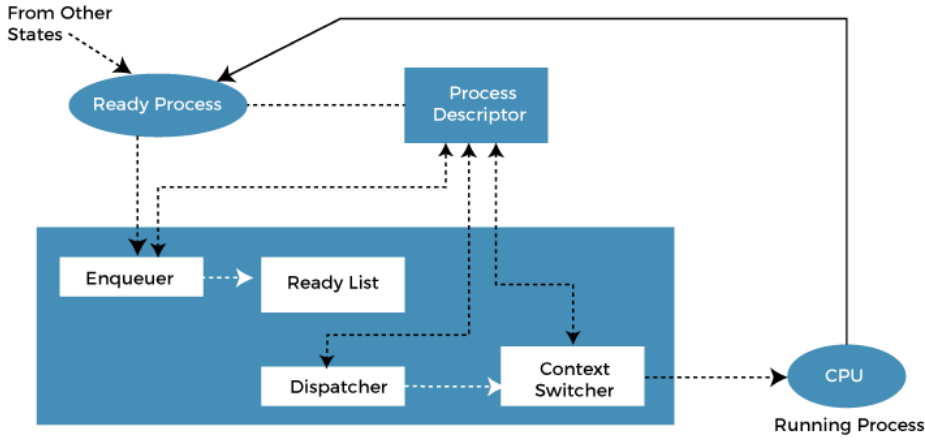
Dosyadan okunan her proses için kendi yazdığımız proses sınıfından nesneler oluştururuz ve gerekli değişkenlerin atamasını yaparız.

Görevlendirici tarafından prosesin çalışması için proses sınıfımızın `execute` fonksiyonu çağrılmalıdır. Bu fonksiyon Java proseslerini kullanarak daha önce oluşturduğumuz jar dosyasını çalıştırır ve konsola prosesin bilgileri yazdırılır. Java proses `Runtime.exec()` kullanılarak çağrılır.

Proses mesajlarının renkli çıkabilmesi için `ColoredSystemOutPrintLn` sınıfındaki ANSI escape sequence’ları kullanılmıştır. Her yeni proses oluştuğunda `COLORS` array’indeki ANSI kodlarından biri o prosese özel bir `colorId` property’si şeklinde atanır. Bu `colorId`’ler, proses mesajlarının stringlerinde başa gelir. Proses mesajları ekrana renkli bir şekilde çıkar.

## Görevlendirici Karşılaştırması

Görevlendirici modülünün işletim sistemi üzerindeki görevi gelen işlemlerin CPU'yu kullanımını düzenler, process'leri uygun pozisyonlara koyar ve kullanıcı modunu değiştirmeyi sağlar. Görevlendirici modülleri tek bir amaca hizmet etmediği için birden fazla çeşiti vardır. Bu yüzden kendi ihtiyaçlarımıza özel olarak görevlendirici algoritması yazabiliriz. İşletim sisteminin kullandığı görevlendirici şemasına bakacak olursak (bkz Görsel:1), gelen process'ler hazır kuyruğuna alınıyor ve sırası gelen process çalıştırılmak üzere CPU'ya yönlendiriliyor.



Bizim görevlendirici mekanizmamıza bakacak olursak :

