

# CASE1 PROJE RAPORU

**Ders Adı:** İşletim Sistemleri

**Ödev Konusu:** İşlemci Zamanlama

**Hazırlayan:** Ahmet Eren Kavan

---

## 1. GİRİŞ

Bu projenin amacı, işletim sistemlerinde kullanılan temel CPU çizelgeleme (CPU Scheduling) algoritmalarının davranışlarını simüle etmek ve performanslarını karşılaştırmalı olarak analiz etmektir. Proje kapsamında FCFS, SJF (Preemptive/Non-Preemptive), Priority (Preemptive/Non-Preemptive) ve Round Robin algoritmaları Python dili kullanılarak kodlanmış ve case1.csv veri seti üzerinde test edilmiştir.

## 2. DURUM 1 (CASE 1) ANALİZİ

### 2.1. Veri Seti ve Yöntem

Case 1 veri seti üzerinde 6 farklı algoritma çalıştırılmıştır. Performans ölçümü için aşağıdaki metrikler dikkate alınmıştır:

- **Ortalama Bekleme Süresi (Average Waiting Time):** Süreçlerin hazır kuyruğunda geçirdiği ortalama süre.
- **Ortalama Tamamlanma Süresi (Average Turnaround Time):** Sürecin sisteme girişinden bitişine kadar geçen süre.
- **İş Tamamlama Sayısı (Throughput):** Belirli zaman aralıklarında ( $T=50, 100, 150, 200$ ) tamamlanan toplam işlem sayısı.
- **CPU Verimliliği:** İşlemcinin aktif olarak çalıştığı sürenin toplam süreye oranı.
- **Bağlam Değiştirme (Context Switch):** İşlemcinin bir süreçten diğerine geçiş sayısı.

### 2.2. Zaman Tabloları (Gantt Şeması)

Algoritmaların süreçleri hangi sırayla çalıştığını gösteren detaylı zaman akışları, simülasyon sonucunda üretilen .txt dosyalarında (Örn: sonuc\_fcfs\_case1.txt, sonuc\_round\_robin\_case1.txt vb.) "a) Zaman Tablosu" başlığı altında sunulmuştur.

## 2.3. Performans Karşılaştırma Tablosu

Aşağıdaki tablo, `case1.csv` veri seti kullanılarak elde edilen tüm algoritmaların performans sonuçlarını özetlemektedir.

Algoritma	Bekleme Süresi (Ort. / Maks.)	Tamamlanma Süresi (Ort. / Maks.)	T=50	T=100	T=150	T=200	CPU Verim.	Context Switch
FCFS	813.59 / 1683.20	824.09 / 1703.20	9	13	16	19	%99.94	200
Preemptive SJF	537.11 / 1863.21	547.61 / 1883.21	11	22	32	42	%99.94	215
Non-Preempt SJF	537.52 / 1863.19	548.02 / 1883.19	11	22	32	42	%99.94	200
Round Robin (q=10)	1003.65 / 1683.30	1014.15 / 1703.30	9	10	10	18	%99.94	300
Preempt Priority	833.73 / 1689.20	844.23 / 1707.20	6	11	14	20	%99.94	202
Non-Preempt Priority	824.87 / 1689.20	835.37 / 1707.20	8	13	16	21	%99.94	200

(Değerler milisaniye cinsindendir. Throughput değerleri kümülatif işlem sayısını ifade eder.)

## 2.4. Sonuçların Değerlendirilmesi ve Yorumlar

Elde edilen veriler ışığında Case 1 için yapılan analizler aşağıdadır:

### 1. Bekleme Süresi Performansı:

- Sistemin en iyi performansını Preemptive SJF algoritması (**537.11 ms** ortalama bekleme süresi) ile göstermiştir. Kısa süreli işlerin öne alınması, kuyruktaki ortalama bekleme süresini ciddi oranda düşürmüştür.
- En kötü performans ise Round Robin (**1003.65 ms**) algoritmasında gözlemlenmiştir. Seçilen Quantum süresinin (10 birim), veri setindeki işlem sürelerine kıyasla küçük kalması ve sık bağlam değiştirme yapılması, süreçlerin tamamlanmasını geciktirmiştir.

### 2. Throughput (İşlem Hızı) Analizi:

- Sistemin iş üretme kapasitesi incelendiğinde, SJF algoritmaları (hem Preemptive hem Non-Preemptive) **T=200** anına kadar **42** adet işlem tamamlayarak diğer algoritmala büyük fark atmıştır.
- Aynı süre zarfında Round Robin sadece **18** işlem bitirebilmiştir. Bu durum, SJF'nin sistem yükünü hafifletmekte ve yanıt süresini kısaltmakta çok daha başarılı olduğunu göstermektedir.

### **3. Starvation (Açlık) Riski:**

- Tablo dikkatle incelendiğinde ilginç bir tezatlık göze çarpmaktadır: Ortalama bekleme süresinde en iyi olan **SJF**, maksimum bekleme süresinde (**1863.21 ms**) en yüksek (en kötü) değere sahiptir.
- Bu durum, SJF algoritmasının kısa işleri sürekli öne alırken, uzun süreli bir işi çok uzun süre beklettiğini (Starvation) kanıtlamaktadır. FCFS ve Round Robin ise daha adil (Fair) davranışarak maksimum bekleme sürelerini daha dengeli tutmuştur.

### **4. Bağlam Değiştirme (Context Switch) Maliyeti:**

- Sisteme en çok yük bindiren algoritma **300** adet bağlam değiştirme ile **Round Robin** olmuştur.
- Non-Preemptive yöntemler, işlem sayısı kadar (200) kesme yaparken; Preemptive SJF (**215**) ve Preemptive Priority (**202**) algoritmaları, sisteme yeni ve daha öncelikli bir iş geldiğinde mevcut işi kestiği için fazladan bağlam değiştirme maliyeti oluşturmuştur.

**5. Genel Değerlendirme:** Bu veri seti için **Preemptive SJF**, işlemciyi en verimli kullanan ve ortalama bekleme sürelerini minimize eden en uygun algoritma olarak öne çıkmaktadır. Ancak uzun süreli işlemlerin gecikmesi (Starvation) kabul edilemez bir durumsa, daha adil bir dağılım için FCFS tercih edilebilir. Round Robin bu senaryoda ( $q=10$  için) verimsiz kalmıştır.