

T.C. İSTANBUL NİŞANTAŞI ÜNİVERSİTESİ
MÜHENDİSLİK MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
DERS: EBLM341 – İŞLETİM SİSTEMLERİ
KONU: İŞLEMCİ ZAMANLAMA ALGORİTMALARI ANALİZİ
HAZIRLAYAN: Harun ALTUNOK 20242013111

Bu projede, çoklu görev (multitasking) işletim sistemlerinin temel bileşeni olan işlemci zamanlama algoritmaları analiz edilmiştir. Amaç, farklı yük durumlarında (Case 1 ve Case 2) algoritmaların performansını kıyaslamaktır.

Simülasyon **Python** programlama dili kullanılarak geliştirilmiştir. Projenin en önemli özelliği **Multithreading** yapısıdır. Her algoritma ayrı bir iş parçacığı (thread) olarak tasarlanmış ve tüm algoritmalar eş zamanlı (concurrent) olarak çalıştırılmıştır. Veri tutarlılığını sağlamak için deepcopy yöntemi kullanılmıştır.

Zaman Tabloları (Gantt Charts) Case 1 ve Case 2 veri seti çok büyük olduğu için detaylı zaman çizelgeleri ekteki dosyalarda sunulmuştur."

Case 1 Performans Karşılaştırma Tablosu

| Algoritma | Ort. Bekleme (sn) | Ort. Tamamlanma (sn) | CPU Verimliliği (%) | Context Switch Sayısı | Throughput (T=200) |
|--------------------|-------------------|----------------------|---------------------|-----------------------|--------------------|
| FCFS | 813.495 | 823.995 | %99.943 | 199 | 19 |
| Round Robin | 1091.700 | 1102.200 | %99.924 | 599 | 14 |
| Priority (Non-Pre) | 824.770 | 835.270 | %99.943 | 199 | 21 |
| Priority (Pre) | 833.630 | 844.130 | %99.943 | 201 | 20 |
| SJF (Non-Pre) | 537.425 | 547.925 | %99.943 | 199 | 42 |
| SJF (Preemptive) | 537.005 | 547.505 | %99.942 | 212 | 42 |

En İyi Performans (SJF): Tablodan görüldüğü üzere, **SJF (Shortest Job First)** algoritmaları (hem Preemptive hem Non-Preemptive), ortalama bekleme süresini **~537 saniye** seviyelerine çekerek en iyi performansı göstermiştir. Diğer algoritmalar 800-1000 saniye civarındayken, SJF'nin kısa işleri öne alması sistemin genel akışını hızlandırmıştır.

En Yüksek İş Tamamlama (Throughput): T=200 anına kadar en çok işi bitiren algoritma yine **42 işlem** ile SJF olmuştur. Bu, yoğun sistemlerde SJF'nin verimliliğini kanıtlar.

Round Robin ve Maliyet: Round Robin algoritması, her işleme eşit süre tanıdığı için süreçleri sık sık bölmüş, bu da **Context Switch sayısını 599'a** fırlatmıştır (Diğerlerinde bu sayı ~200 civarındır). Bu durum, işlemci üzerinde ek yük (overhead) oluşturmuş ve ortalama bekleme süresini **1091 saniyeye** kadar çıkarmıştı

Case 2 Performans Karşılaştırma Tablosu

| Algoritma | Ort. Bekleme (sn) | Ort. Tamamlanma (sn) | CPU Verimliliği (%) | Context Switch Sayısı | Throughput (T=200) |
|--------------------|-------------------|----------------------|---------------------|-----------------------|--------------------|
| FCFS | 418.000 | 428.500 | %99.991 | 99 | 18 |
| SJF (Non-Pre) | 268.390 | 278.890 | %99.991 | 99 | 42 |
| SJF (Preemptive) | 267.860 | 278.360 | %99.990 | 110 | 42 |
| Round Robin | 550.870 | 561.370 | %99.972 | 299 | 11 |
| Priority (Non-Pre) | 409.630 | 420.130 | %99.991 | 99 | 19 |
| Priority (Pre) | 411.390 | 421.890 | %99.990 | 100 | 19 |

En Düşük Bekleme Süresi: Case 2 veri setinde de **SJF (Shortest Job First)** algoritmaları en başarılı sonucu vermiştir. Özellikle **Preemptive SJF**, **267.860 sn** ortalama bekleme süresi ile en iyi performansı sergilemiş, onu **268.390 sn** ile Non-Preemptive SJF takip etmiştir.

Throughput (İş Tamamlama) Farkı: T=200 anına kadar **SJF algoritmaları 42 işlem** tamamlayarak sistem verimliliğini maksimize etmiştir. Buna karşılık Round Robin sadece **11 işlem** bitirebilmiştir. Bu durum, kısa işlerin öne alınmasının sistem çıktısını (throughput) ne kadar artırdığını net bir şekilde göstermektedir.

Context Switch Maliyeti: Round Robin algoritması, zaman paylaşımı (Time Quantum = 4) nedeniyle **299 kez** bağlam değiştirme (Context Switch) yapmıştır. Bu sayı diğer algoritmalarda ortalama 99-110 civarındadır. Yüksek bağlam değiştirme sayısı, Round Robin'in ortalama bekleme süresini **550.870 saniyeye** kadar yükseltmiş ve CPU verimliliğini diğerlerine kıyasla çok az da olsa düşürmüştür (%99.972).

