

Bekleme Statistics ↺

	fcfs	pjf_preemptive	rr	pjf	sjf_preemptive	sjf
ortalama	418	411.39	561.54	409.63	267.86	268.39
maksimum	851	836	944	836	926	926

Analiz

- **Ortalama:** En yüksek RR (561.54), en düşük SJF_Preemptive (267.86).
- **Maksimum:** En yüksek RR (944), en düşük PJF_Preemptive ve PJF (836).

Açıklama

Bu durum, **SJF (Shortest Job First)** algoritmasının toplam bekleme süresini minimize etmek üzere tasarlanmış olmasıyla, **RR (Round Robin)** algoritmasının ise zaman dilimleme (time-slicing) ve sık bağlam geçişleri (context switch) nedeniyle genel bekleme süresini artırmasıyla açıklanabilir.

Tamamlama Statistics ↺

	fcfs	pjf_preemptive	rr	pjf	sjf_preemptive	sjf
ortalama	428.5	421.89	572.04	420.13	278.36	278.89
maksimum	853	854	963	854	946	946

Analiz

- **Ortalama:** En yüksek rr (572.04), en düşük sjf_preemptive (278.36).
- **Maksimum:** En yüksek rr (963), en düşük fcfs (853).

Açıklama

Bu durum, **SJF (Shortest Job First)** algoritmalarının kısa işlere öncelik vererek ortalama tamamlama süresini optimize etmesi, **RR (Round Robin)** algoritmasının ise işleri bölerek her birinin sistemde kalma süresini (turnaround time) uzatması ile açıklanabilir. Maksimum değerlerde **FCFS**'in düşük çıkması, uzun süreçlerin bölünmeden hızla tamamlanabildiğini gösterir.

Troughput Statistics

	fcfs	pjf_preemptive	rr	pjf	sjf_preemptive	sjf
T=50	5	4	3	5	10	10
T=100	10	8	7	9	21	21
T=150	14	15	10	15	31	30
T=200	18	19	13	19	42	42

En Hızlı: SJF / SJF_P (42 iş)

- Neden:** En kısa işleri önceliklendirerek CPU'nun işleri hızla eritmesini sağlar; verimlilikte (throughput) her zaman en üsttedir.

Orta: PJF / PJF_P (19 iş) ve FCFS (18 iş)

- Neden:** Öncelik veya geliş sırasına odaklanırlar. Uzun bir iş araya girdiğinde diğerlerini bekleterek toplam bitirme sayısını sınırlar.

En Yavaş: RR (13 iş)

- Neden:** İşlemleri sürekli bölüp aralarında geçiş yaptığı (context switching) için zaman kaybeder. Bu tabloda verimliliği en düşük algoritmadır.

CPU verimi ⇄

	fcfs	pjf_preemptive	rr	pjf	sjf_preemptive	sjf
num_switches	99	100	549	99	110	99
cpu_utilization	99.9058	99.9049	99.4799	99.9058	99.8953	99.9058

En Yüksek Verim: FCFS, PJF ve SJF algoritmaları, en düşük anahtar değişimi (99) ve en yüksek CPU kullanımı (%99.9058) ile en verimli sonuçları vermiştir.

En Düşük Verim: Round Robin (RR), çok yüksek anahtar değişimi (549) nedeniyle CPU kullanımında en düşük performansı (%99.4799) sergilemiştir.

Nedeni: Bağlam değişimi (context switch) sayısı arttıkça CPU verimliliği düşmektedir. Basit kuyruklama yöntemleri bu veri setinde daha başarılıdır.