



SAKARYA ÜNİVERSİTESİ

İŞLETİM SİSTEMLERİ PROJE RAPORU - 19 Grubu.

Ders: İşletim Sistemleri .

Github linki :<https://github.com/allabban/Isletim19.git>

Öğrenci No - Adı Soyadı -Ders Grubu:

G231210559 Yusuf Demir 2B

G211210574 Salman Abdulsamee 2B

G231210563 Abdullah Allabban 2C

G231210554 Ammar Ajam 2C

G231210571 Abdulrhman Ghanoum 1B

Konu: FreeRTOS ile Öncelik Tabanlı ve Round-Robin Destekli İşlemci Zamanlayıcı Simülasyonu

-Giriş-

Bu projenin amacı, işletim sistemlerinin temel bileşenlerinden biri olan işlemci zamanlayıcı (CPU Scheduler) mekanizmasının çalışma prensiplerini simüle etmektir. Proje, gerçek zamanlı işletim sistemi çekirdeği olan **FreeRTOS** kütüphanesi kullanılarak geliştirilmiştir. Simülasyon, farklı önceliklere ve çalışma sürelerine (burst time) sahip işlemlerin (task) yönetimini, kuyruk yapılarını ve bağlam değiştirme (context switch) maliyetlerini modellemektedir.

-Geliştirme Ortamı ve Altyapı (WSL Kullanımı)-

Proje, Windows işletim sistemi üzerinde Linux çekirdek uyumluluğu sağlayan **WSL (Windows Subsystem for Linux)** ortamında geliştirilmiş ve test edilmiştir. Bu yöntem, Linux tabanlı sistem çağrılarını ve FreeRTOS portunu yerel bir Linux dağıtımı (Ubuntu) üzerindeyim gibi performanslı bir şekilde çalıştmak amacıyla tercih edilmiştir.

- **Platform:** WSL (Ubuntu)
- **Derleyici:** GCC (GNU Compiler Collection)
- **Build Sistemi:** GNU Make
- **Kütüphane:** FreeRTOS (Linux Port)

-Zamanlayıcı Algoritması ve Tasarım-

Simülasyon, "Çok Seviyeli Kuyruk" (Multi-Level Queue) mantığına benzer hibrit bir algoritma kullanmaktadır. `schedulers.c` içerisindeki `selectNextTask` fonksiyonu şu öncelik sıralamasını takip eder:

1. **Yüksek Öncelik (Priority 0):** Sistemin en kritik işlemidir, varsa derhal çalıştırılır.
2. **Orta Öncelik (Priority 1-2):** Standart öncelik tabanlı sıralama ile çalışır.

3. Round Robin (Priority 3): 3. öncelik seviyesindeki işlemler, adil kullanım sağlamak amacıyla **Round Robin (RR)** algoritması ile sırayla işlenir. Kod içerisinde `lastRRIndex` değişkeni ile son kalınan işlem takip edilir.

4. Düşük Öncelik (Priority 4): Sadece üst seviyelerde işlem kalmadığında devreye girer.

Ayrıca, her işlem için 20 saniyelik bir **Zaman Aşımı (Timeout)** mekanizması uygulanmıştır. İşlem CPU'yu belirtilen sürede alamazsa sistemden atılır (`vTaskDelete`).

```
aws232@ABDULRHMAN:/mnt/c/Users/aws23/Downloads/Isletim19-main/Isletim19-main$ ./freertos_sim
0.0000 sn proses başladı      (id:0000 öncelik:1 kalan süre:2 sn)
1.0000 sn proses askıda      (id:0000 öncelik:2 kalan süre:1 sn)
1.0000 sn proses başladı      (id:0001 öncelik:0 kalan süre:1 sn)
2.0000 sn proses sonlandı    (id:0001 öncelik:0 kalan süre:0 sn)
2.0000 sn proses başladı      (id:0003 öncelik:0 kalan süre:3 sn)
3.0000 sn proses yürütülüyor (id:0003 öncelik:0 kalan süre:2 sn)
4.0000 sn proses yürütülüyor (id:0003 öncelik:0 kalan süre:1 sn)
5.0000 sn proses sonlandı    (id:0003 öncelik:0 kalan süre:0 sn)
5.0000 sn proses başladı      (id:0006 öncelik:0 kalan süre:4 sn)
6.0000 sn proses yürütülüyor (id:0006 öncelik:0 kalan süre:3 sn)
7.0000 sn proses yürütülüyor (id:0006 öncelik:0 kalan süre:2 sn)
8.0000 sn proses yürütülüyor (id:0006 öncelik:0 kalan süre:1 sn)
9.0000 sn proses sonlandı    (id:0006 öncelik:0 kalan süre:0 sn)
9.0000 sn proses başladı      (id:0007 öncelik:0 kalan süre:4 sn)
10.0000 sn proses yürütülüyor (id:0007 öncelik:0 kalan süre:3 sn)
11.0000 sn proses yürütülüyor (id:0007 öncelik:0 kalan süre:2 sn)
12.0000 sn proses yürütülüyor (id:0007 öncelik:0 kalan süre:1 sn)
13.0000 sn proses sonlandı    (id:0007 öncelik:0 kalan süre:0 sn)
13.0000 sn proses başladı      (id:0008 öncelik:0 kalan süre:2 sn)
14.0000 sn proses yürütülüyor (id:0008 öncelik:0 kalan süre:1 sn)
15.0000 sn proses sonlandı    (id:0008 öncelik:0 kalan süre:0 sn)
15.0000 sn proses başladı      (id:0010 öncelik:0 kalan süre:3 sn)
16.0000 sn proses yürütülüyor (id:0010 öncelik:0 kalan süre:2 sn)
17.0000 sn proses yürütülüyor (id:0010 öncelik:0 kalan süre:1 sn)
18.0000 sn proses sonlandı    (id:0010 öncelik:0 kalan süre:0 sn)
18.0000 sn proses başlandı     (id:0016 öncelik:0 kalan süre:4 sn)
19.0000 sn proses yürütülüyor (id:0016 öncelik:0 kalan süre:3 sn)
20.0000 sn proses yürütülüyor (id:0016 öncelik:0 kalan süre:2 sn)
21.0000 sn proses yürütülüyor (id:0016 öncelik:0 kalan süre:1 sn)
21.0000 sn proses zamanaşımı (id:0000 öncelik:2 kalan süre:1 sn)
21.0000 sn proses zamanaşımı (id:0002 öncelik:3 kalan süre:2 sn)
21.0000 sn proses zamanaşımı (id:0004 öncelik:2 kalan süre:2 sn)
22.0000 sn proses sonlandı    (id:0016 öncelik:0 kalan süre:0 sn)
22.0000 sn proses başlandı     (id:0017 öncelik:0 kalan süre:4 sn)
22.0000 sn proses zamanaşımı (id:0005 öncelik:2 kalan süre:3 sn)
23.0000 sn proses yürütülüyor (id:0017 öncelik:0 kalan süre:3 sn)
24.0000 sn proses yürütülüyor (id:0017 öncelik:0 kalan süre:2 sn)
24.0000 sn proses zamanaşımı (id:0009 öncelik:2 kalan süre:4 sn)
25.0000 sn proses yürütülüyor (id:0017 öncelik:0 kalan süre:1 sn)
25.0000 sn proses zamanaşımı (id:0011 öncelik:3 kalan süre:2 sn)
26.0000 sn proses sonlandı    (id:0017 öncelik:0 kalan süre:0 sn)
26.0000 sn proses başlandı     (id:0019 öncelik:0 kalan süre:4 sn)
26.0000 sn proses zamanaşımı (id:0012 öncelik:3 kalan süre:2 sn)
```

25.0000 sn proses zamanaşımı (id:0011 öncelik:3 kalan süre:2 sn)
26.0000 sn proses sonlandı (id:0017 öncelik:0 kalan süre:0 sn)
26.0000 sn proses başladı (id:0019 öncelik:0 kalan süre:4 sn)
26.0000 sn proses zamanaşımı (id:0012 öncelik:3 kalan süre:2 sn)
26.0000 sn proses zamanaşımı (id:0013 öncelik:1 kalan süre:2 sn)
27.0000 sn proses yürütülüyor (id:0019 öncelik:0 kalan süre:3 sn)
28.0000 sn proses yürütülüyor (id:0019 öncelik:0 kalan süre:2 sn)
28.0000 sn proses zamanaşımı (id:0014 öncelik:1 kalan süre:4 sn)
29.0000 sn proses yürütülüyor (id:0019 öncelik:0 kalan süre:1 sn)
29.0000 sn proses zamanaşımı (id:0015 öncelik:3 kalan süre:4 sn)
30.0000 sn proses sonlandı (id:0019 öncelik:0 kalan süre:0 sn)
30.0000 sn proses başladı (id:0024 öncelik:1 kalan süre:2 sn)
31.0000 sn proses askıda (id:0024 öncelik:2 kalan süre:1 sn)
31.0000 sn proses başladı (id:0018 öncelik:2 kalan süre:2 sn)
32.0000 sn proses askıda (id:0018 öncelik:3 kalan süre:1 sn)
32.0000 sn proses başladı (id:0022 öncelik:2 kalan süre:3 sn)
33.0000 sn proses askıda (id:0022 öncelik:3 kalan süre:2 sn)
33.0000 sn proses yürütülüyor (id:0024 öncelik:2 kalan süre:1 sn)
34.0000 sn proses sonlandı (id:0024 öncelik:2 kalan süre:0 sn)
34.0000 sn proses yürütülüyor (id:0018 öncelik:3 kalan süre:1 sn)
35.0000 sn proses sonlandı (id:0018 öncelik:3 kalan süre:0 sn)
35.0000 sn proses başladı (id:0020 öncelik:3 kalan süre:3 sn)
36.0000 sn proses askıda (id:0020 öncelik:4 kalan süre:2 sn)
36.0000 sn proses başladı (id:0021 öncelik:3 kalan süre:2 sn)
37.0000 sn proses askıda (id:0021 öncelik:4 kalan süre:1 sn)
37.0000 sn proses yürütülüyor (id:0022 öncelik:3 kalan süre:2 sn)
38.0000 sn proses askıda (id:0022 öncelik:4 kalan süre:1 sn)
38.0000 sn proses başladı (id:0023 öncelik:3 kalan süre:2 sn)
39.0000 sn proses askıda (id:0023 öncelik:4 kalan süre:1 sn)
39.0000 sn proses yürütülüyor (id:0020 öncelik:4 kalan süre:2 sn)
40.0000 sn proses askıda (id:0020 öncelik:5 kalan süre:1 sn)
40.0000 sn proses yürütülüyor (id:0021 öncelik:4 kalan süre:1 sn)
41.0000 sn proses sonlandı (id:0021 öncelik:4 kalan süre:0 sn)
41.0000 sn proses yürütülüyor (id:0022 öncelik:4 kalan süre:1 sn)
42.0000 sn proses sonlandı (id:0022 öncelik:4 kalan süre:0 sn)
42.0000 sn proses yürütülüyor (id:0023 öncelik:4 kalan süre:1 sn)
43.0000 sn proses sonlandı (id:0023 öncelik:4 kalan süre:0 sn)
43.0000 sn proses yürütülüyor (id:0020 öncelik:5 kalan süre:1 sn)
44.0000 sn proses sonlandı (id:0020 öncelik:5 kalan süre:0 sn)

-İstatistiksel Analiz ve Görselleştirme-

Projenin son aşamasında, simülasyonun verimliliğini ölçmek adına detaylı bir raporlama modülü eklenmiştir. Simülasyon tamamlandığında konsola yesil renk kodları ile dikkat çekici bir özet tablo basılmaktadır.

Hesaplanan Metrikler:

- **Total Turnaround Time (Toplam Geri Dönüş Süresi):** İşlemin geliş anından bitiş anına kadar geçen süre.
- **Total Waiting Time (Toplam Bekleme Süresi):** İşlemin hazır kuyruğunda beklediği süre.
- **Dropped Tasks:** Zaman aşımı nedeniyle iptal edilen işlem sayısı.
- **Ortalama Süreler:** Başarılı işlemlerin ortalama bekleme ve tamamlanma süreleri hesaplanarak sistemin genel performansı ortaya konmuştur.

Görselleştirmeyi iyileştirmek için ANSI kaçış kodları kullanılarak her işlem (task) farklı bir renkle (Mavi, Kırmızı, Yeşil, vb.) loglanmıştır.

```
Simulasyon Tamamlandı.
-----
SIMULATION SUMMARY
-----
Total Simulation Time : 44 seconds
Total Tasks Processed : 25
Tasks Completed       : 15
Tasks Dropped (Timeout): 10
Avg Turnaround Time   : 14.00 sec
Avg Waiting Time      : 11.13 sec
-----
```

-Derleme ve Çalıştırma Adımları-

Proje, WSL terminali üzerinde Makefile kullanılarak derlenmektedir. Aşağıdaki komutlar ile simülasyon başlatılır:

1. Derleme (Compile):

```
make
```

Bu komut, kaynak kodları (*scheduler.c*, *main.c*, *task.c*) FreeRTOS kütüphanesi ile linkleyerek çalıştırılabilir dosyayı oluşturur.

2. Çalıştırma (Run):

`./freertos_sim`

Bu komut, *giris.txt* dosyasındaki senaryoyu okuyarak simülasyonu başlatır.

Tartışma ve Olası İyileştirmeler

6.1. Gerçek İşletim Sistemleri ile Karşılaştırma

Bu projede kullanılan "Çok Seviyeli Kuyruk" (Multi-Level Queue) yapısı, Linux veya Windows gibi modern işletim sistemlerinin kullandığı karmaşık zamanlayıcılara (örn. Linux CFS veya Windows Multilevel Feedback Queue) temel teşkil eder. Gerçek işletim sistemlerinde de, bu projede olduğu gibi, etkileşimli işlemler (I/O bound) ile işlemci yoğunluklu işlemler (CPU bound) arasında denge kurmak için öncelik seviyeleri dinamik olarak değiştirilir. Ancak, gerçek sistemlerde "starvation" (açlık) durumunu önlemek için "aging" (yaşlandırma) gibi daha karmaşık mekanizmalar kullanılır. Bizim simülasyonumuzda ise basit bir zaman aşımı (timeout) mekanizması ile sistemin tikanması engellenmiştir.

6.2. Sistemin Eksiklikleri ve Önerilen İyileştirmeler

Simülasyonumuz başarılı bir şekilde çalışsa da bazı geliştirilmeye açık noktaları vardır:

- Dinamik Öncelik Yönetimi:** Şu anki yapıda görevler belirli kurallara göre kuyruk değiştirir. Daha gelişmiş bir "aging" algoritması eklenerek, uzun süre bekleyen düşük öncelikli görevlerin önceliği zamanla artırılabilir.
- Bellek Yönetimi:** Projede statik dizi (*taskList*) kullanılmıştır. Bunun yerine dinamik bellek tahsisini (*Heap allocation*) kullanılarak sistemin bellek kullanımı optimize edilebilir ve MAX_TASKS sınırı kaldırılabilir.
- Geri Besleme Mekanizması:** Feedback kuyruğu arasındaki geçişler daha hassas zaman dilimleriyle (micro-second precision) yönetilebilir.

-Sonuç-

Yapılan simülasyon sonucunda, FreeRTOS'un görev yönetimi fonksiyonları (`xTaskCreate`, `vTaskSuspend`, `vTaskResume`) başarıyla kullanılmış ve işletim sistemlerindeki zamanlama algoritmalarının davranışları gözlemlenmiştir. Eklenen istatistik modülü sayesinde, algoritmanın işlem yoğunluğu altındaki performansı sayısal verilerle kanıtlanmıştır.
