

## İşletim Sistemleri Dersi Uygulama Ödevi

Bu ödev, işletim sistemleri dersinin thread'ler ve semaforlar konularını pekiştirmek amacıyla hazırlanmıştır. Öğrencilerin thread kullanımı, thread senkronizasyonu ve semaforlarla kaynak yönetimi konularında pratik yapmaları hedeflenmektedir.

Her iki soru da bireysel olarak çözülmelidir. Kodların doğru çalıştığından emin olunmalı ve gereken çıktılar elde edilmelidir. Ödev, bireysel olarak tamamlanacaktır ve her öğrenci kendi kodunu yazıp test etmekle yükümlüdür. Ödevin son teslim tarihi **21.12.2024 Saat:17:00** olup, teslimler **edestek2.kocaeli.edu.tr** adresine yapılacaktır. Belirtilen tarihten sonra yapılan teslimatlar değerlendirmeye alınmayacaktır. Öğrencilerin kodlarını, derlenebilir ve çalıştırılabilir bir şekilde teslim etmeleri gerekmektedir. Ayrıca, kodlarını açıklayan kısa bir **README dosyası** eklemeleri beklenmektedir.

### Kodlama Dilleri:

Öğrenciler, ödevde aşağıdaki programlama dillerinden birini kullanabilirler:

- **C:** Sistem programlama ve thread yönetimi konusunda güçlü olduğu için önerilir.
- **C++:** C diline benzer bir yapıya sahip olup, daha modern özellikler içerir.
- **Java:** Thread sınıfı ve semafor gibi senkronizasyon araçları için kullanıcı dostu bir API sağlar.
- **Python:** Kolay anlaşılır bir sözdizimine sahip olduğu için tercih edilebilir, ancak performans gerektiren uygulamalarda dikkat edilmelidir.

Her öğrencinin seçtiği dilin temel özelliklerini kullanarak doğru ve verimli bir kod yazması beklenmektedir. Teslim edilen dosyanın adı şu formatta olmalıdır: ad\_soyad\_ogrencino.zip (örnek: ahmet\_yilmaz\_123456.zip).

## Soru 1: Çoklu Thread Kullanımı ile Basit Sayma Problemi

**Amaç:** Thread'lerin oluşturulması ve temel kullanımını öğrenmek.

**Senaryo:** Bir sınıfta toplam **50** öğrenci vardır ve bu öğrenciler **5** farklı gruba ayrılmıştır. Her grup bir temsilci seçecektir. Öğrenciler sırayla temsilci olmak için başvuruda bulunur ve thread'ler kullanılarak bu süreç simüle edilir.

### Görev:

- Her grup için bir thread oluşturun (**5 thread**).
- Her thread **10** öğrenciyi temsil edecek.
- Program, her thread'in kaç öğrenci kontrol ettiğini ekrana yazdırmalıdır.
- Son olarak, her gruptan bir temsilci seçildiğini ekrana yazdırın.

**Beklenen Çıktı:**

*Grup 1: 10 öğrenci kontrol edildi.*

*Grup 2: 10 öğrenci kontrol edildi.*

*Grup 3: 10 öğrenci kontrol edildi.*

*Grup 4: 10 öğrenci kontrol edildi.*

*Grup 5: 10 öğrenci kontrol edildi.*

*Her grup bir temsilci seçti.*

**Soru 2: Thread ve Semafor Kullanarak Yemekhane Kuyruğu Senkronizasyonu**

**Amaç:** Thread senkronizasyonu ve semafor kullanımını öğrenmek.

**Senaryo:** Bir üniversite yemekhanesinde 3 tezgah bulunmaktadır. Her tezgahta aynı anda yalnızca 5 öğrenciye hizmet verilebilir. 50 öğrenci yemek almak için sıradadır. Tezgahlara aynı anda en fazla 5 öğrenci erişebilir ve diğer öğrenciler sırada beklemek zorundadır.

**Görev:**

- 3 tezgahı ve 50 öğrenciyi temsil eden bir program yazın.
- Her tezgah bir thread olarak temsil edilmelidir.
- Her öğrenci bir tezgaha erişmek için bir semafordan izin almalıdır.
- Bir tezgah dolduğunda öğrenciler başka bir tezgaha yönlendirilmelidir.

**İpuçları:**

- Semaforları tezgah kapasitesini kontrol etmek için kullanabilirsiniz.
- Her tezgah dolduğunda veya tüm öğrenciler hizmet aldığında program sonlanmalıdır.

**Beklenen Çıktı:**

*Tezgah 1: 1 öğrenci yemek aldı. Kalan öğrenci: 49*

*Tezgah 2: 1 öğrenci yemek aldı. Kalan öğrenci: 48*

*Tezgah 1: 1 öğrenci yemek aldı. Kalan öğrenci: 47*

*...*

*Tezgah 1 doldu, öğrenciler bekliyor.*

*Tezgah 2 doldu, öğrenciler bekliyor.*

*Tezgah 3: 1 öğrenci yemek aldı. Kalan öğrenci: 25*

*...*

*Tüm öğrenciler yemek aldı. Program sonlanıyor.*

**Değerlendirme Kriterleri:**

1. Thread'lerin doğru bir şekilde oluşturulması ve çalıştırılması.
2. Semafor kullanımı ve senkronizasyonun sağlanması.
3. Kodların okunabilirliği ve yorumlarla açıklanması.
4. Çıktıların beklenen şekilde elde edilmesi.