

SUSHİ BAR PROBLEM

SUSHİ BAR PROBLEMİ NEDİR ?

Sushi Bar problemini basitçe anlatmak gerekirse, bir barda 5 sandalyelik bir masa bulunmaktadır. Biz bu masaya oturmak istemekteyiz, fakat masadaki kişilerin sayısına bakmamız gerek.

Masada oturan kişi sayısı 0, 1, 2, 3, 4 olduğu durumda; yani en az 1 kişilik bir boşluk olduğu an direk olarak masaya oturabilmekteyiz. Fakat masadaki kişi sayısı 5 olduğu zaman ise, bu 5 kişinin beraber geldiği anlamına gelir. Dolayısıyla masadaki 5 kişinin hepsinin kalkmasını beklemek zorundayız. Masadaki herkes kalkıp masadaki kişi sayısı 0 olduğunda ise masaya oturabileceğiz.

Masadaki kişi sayısına göre bizim o an yemek yiyip yiyemeyeceğimizi hesaplayan, bir java kodu yazacağız. Bu kodda, masadaki kişi sayısı 5 olmadığı her an yemek yiyebileceğimizi göstermesini istemekteyiz.

Ayrıca Masadaki kişi sayısı 5 olduğu anda ise sıra beklememizi, masa boşaldığı anda ise o an masaya oturabileceğimizi göstermesini istemekteyiz.

SUSHİ BAR PROBLEMİ ÇÖZÜM 1:

Bu kodda, öncelikle yiyen ve bekleyen değişkenlerini oluşturduk. Bunlar kaç kişinin yediğini ve kaç kişinin sırada beklediğini göstermektedir. Eğer bekleme durumu varsa bekleme değişkeni 1 artacak. Bunun için sandalyelerin 5'inin de dolması gerekmektedir. Eğer bu koşul sağlanmazsa bekleme durumu gerçekleşmeyecektir. Sonrasında ise Deadlock önlemek amacıyla semaforları kullandık. Bekleme durumu yoksa, yani sandalyeler boşta ise direk yemeği yiyebileceğiz. Bunun için de eating değeri 1 artacak ve aynı zamanda eating değerinin 5'e eşit olup olmadığına bakılıyor. Bunu da sushibar sınıfında oluşturduğumuz setMustWait metodu ile gerçekleştiriyoruz. Ardından her yeme ve ayrılma durumunda kaç kişinin yeme halinde, kaç sandalyenin boşta olduğunu gösteren bir ifade yazdırıyoruz. Yiyen kişinin 0 olması durumunda bekleyen sayısı 5'ten küçükse o kadar kişiyi; 5'ten büyükse 5 tane bekleyen kişiyi yeme durumuna alıyoruz. Bekleyen sayısını buna göre azaltıp yiyen sayısını buna göre arttırıyoruz. Tabii aynı zamanda eating sayısının 5'e eşitlik durumunu kontrol ediyoruz. İşlemi sorunsuz halletmek için block semaforunu sonlandırıp ardından mutex'i sonlandırmış oluyoruz.

SUSHİ BAR PROBLEMİ ÇÖZÜM 2:

İkinci yol olarak ise bekleyen kişi yoksa doğrudan eating sayısı 1 artacak ve eating değeri 5'e eriştiği anda waiting değerleri 1 artacak. Burada da waiting değeri eating değeri azaldıkça azalacak. Yine deadlock olması olasılığına karşılık semaforları acquire ediyoruz.

SUSHİ BAR YANLIŞ ÇÖZÜM:

Problem 7. satırdadır. Eğer Müşteeri bara dolu iken gelirse muutexten vazgeçer vve diğer müşterilerin bardan ayrılmasını bekler.

Son müşteride ayrıldığında block sinyali yollar. Sinyali alan diğer bekleyen müşteriler beklemeyi bitirir.

Fakat müşteriler uyandığında mutex geri gelir ve bunun anlamı yeni iş parçacıklarıyla rekabet etmektir.

Yeni iş parçacıklar geldiğinde ilk mutexi alır vve tüm yerlere dahha önce bekleyen iş parçacıklarını koyar. 5 ten fazla

iş parçacığı için mümkündür eş zamanlı olarak kritik bölümde olmak senkronizasyonu ihlal eden kısıtlamadır.

HAZIRLAYANLAR

17253064	Hüseyin Can Şimşek
18253042	Ali Mert Kahraman
17253040	Yusufkan Sürmeliöğlu
19253519	Arda Yakakayı
18253508	Muhammed Fethullah Sağlam