

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

BIL461/062 İşletim Sistemleri

2021-2022 Yaz Dönemi

Ödev 2

Konu: C, Unix/Linux Multithreading, Pthreads, Process Synchronization

Görev: Meşhur Dining Philosophers problemi yeme-düşünme döngüsünden oluşmaktadır. Her filozofun solunda ve sağında birer tane chopstick olduğu kabulü vardır. Dolayısıyla, aynı anda bu kaynak komşu iki filozof arasında eş zamanlı olarak kullanılamaz. Klasik problemde ortadaki pirinç pilavının sınırsız miktarda olduğu kabulü vardır. Gıda krizi filozofları da vurduğundan ortadaki pilav artık sınırsız değildir. Problemin yeni tanımında bir de garson vardır ve garson belirli aralıklarla o günkü bütçeye uygun olacak şekilde belli sayıda porsiyonu masaya koyar.

Program: Garson ve her bir filozof için ayrı birer thread olacak şekilde yukarıdaki problemi çözen bir C programı yazın. Program başlarken komut satırından filozof sayısını parametre olarak alsın. Garson thread ise her seferinde masaya en az 1 en fazla ise filozof sayısı kadar rasgele miktarda porsiyon koysun. Masadaki pilav bitmeden garson servis yapmaz, yani elinde chopstick ile pilavın gelmesini bekleyen filozof manzaraları görebilirsiniz. Filozoflar arası porsiyon bölüşmek yoktur. Bir porsiyon bir kişiliktir. Aynı turda yeterli porsiyon olsa bile bazı filozoflar birden fazla sayıda (yedi, düşündü, tekrar yedi gibi) yiyebilirken diğerleri şansı yoksa o gün aç kalabilir. Her bir günü simüle etmek için garson her 10 ms de bir servis yapsın. Programınız Unix tabanlı olarak Pthreads kütüphanesini kullanmalıdır. Pthreads kütüphanesindeki hazır mutex, semafor ve monitor yapılarını kullanabilirsiniz.

En başta tüm filozofların thread öncelikleri aynıdır. 1000 gün için simülasyon yapılır ve garsonun yaptığı servislerdeki toplam porsiyon sayısı, hangi filozof toplamda kaç porsiyon yemiş ve her bir filozof için üst üste en fazla kaç gün hiç yiyememiş bilgileri standart çıktıya yazdırılmalıdır. Ardından bir numaralı filozof başfilozof yapılmalıdır ve program simülasyona en baştan başlamalıdır. Baş filozof sadece thread önceliği yüksek olmalıdır. 1000 günlük ikinci bir simülasyonunun ardından önceki turda yazdırılan bilgiler standart çıktıya yazdırılmalı ve program sonlanmalıdır. Böylelikle, uygulama katmanında aynı işi yapan threadler arasında işletim sistemi tarafından yapılan önceliklendirmenin etkilerini inceleme şansı yakalamış olacaksınız.

Gönderim: Bir adet C dosyası (program) göndermeniz gerekmektedir.

Gönderimler <https://uzak.etu.edu.tr> ye dosya yükleme şeklinde olacaktır. Email gönderimleri kabul edilmemektedir.

Not: Readme dosyası göndermenin her zaman profesyonellik olduğunu asla aklınızdan çıkarmayın.