

# İşletim Sistemleri Proje 3 Raporu

*Hazırlayan: 1821221005 Furkan Gündoğan*

## 1. Projenin Adı: C Dili İle Thread Senkronizasyonu

## 2. Proje Konusu: Senkronize edilmiş iki thread ile matrix oluşturma ve elemanlarını toplama işleminin kullanıcı çıkış yapana kadar yapılması.

## 3. Tamamlanan Talimatlar:

3.1 Matrixcalc.c isimli programımız sürekli olarak kullanıcıdan matrix adedi verisi istemektedir.

3.2 Bu programda üç adet thread kullandım:

- th\_matrix\_creator : Matrix oluşturma ve ekrana yazma.
- th\_matrix\_sum : Matrix elemanlarının toplayıp ekrana yazma.
- th\_scan : Kullanıcıdan sürekli adet isteme.

3.3 Bu threadlere ek olarak senkronizasyon için kullanacağımız üç adet lock mekanizması oluşturduk.

- locksum
- lockcreator
- lockscan

3.4 Ana programın başında threadler oluşturuluyor. Lock mekanizmaları init ediliyor. Daha sonra threadler “matrixadet” parametresiyle create ediliyorlar. Threadler join edilerek program sonlanana kadar threadlerin birbirini beklemesi sağlanıyor.

3.5 Th\_scan threadi ile işlemimiz başlıyor; bu thread adetAl() fonksiyonunu çalıştırmakta. Bu fonksiyon ile th\_scan önce kendini kilitliyor ardından kullanıcıdan bir veri girişi alıyor. Bu girişte problem yoksa creator’ın kilidi açılıyor.

3.6 Th\_matrix\_creator içerisinde sonsuz döngüde çalışırken matrixadet sayısı kadar işlem yapmakta. İlk olarak kendini kitlemekte daha sonra matris oluşturup ekrana basıyor ve bu matrisi toplaması için locksum’ın kilidini açıyor.

3.7 Locksum ise önce kendini kilitliyor ve sonsuz döngüde matrix adedi kadar dönerek oluşan matrisin elemanlarını toplar, ardından bu toplamı ekrana basar. Daha sonra sıradaki matrisin oluşması için(creator’ın çalışması için) lockcreator’ın kilidini açar.

3.8 Locksum son matrisi de yazdırdıktan scan’ın kilidini açıyor ki program başa dönsün. Scan threadi içerisinde kullanıcı 0’den küçük bir değer ya da harf girerse program threadler cancel edilerek ve kilit mekanizmaları destroy edilerek kapatılıyor.

3.9 Programda kullandığımız bir diğer teknoloji ise sinyal mekanizması. Ctrl+C kombinasyonuna göre ayarladığımız sinyalimiz program esnasında bu tetikleme yapılırsa kapat() fonksiyonunu çağırıyor. Kapat fonksiyonu da threadleri cancel edip kilit mekanizmalarını destroy ediyor ve programı kapatıyor.

#### **4. Thread Kullanmanın Avantajları**

Bu projede senkronize threadler kullanarak işlemlerimizi yaptık. Threadler fork yapılmış bir prosese göre çok daha hafiftir. Çünkü prosesin tamamen kopyalanması gerekmez ve daha hızlı işlemler yapar. Bir proses için birden fazla thread kullanımı işlem performansını artıracaktır. Çünkü threadler oluşturulduğu prosesin kaynağına erişerek kendileri arasında da iyi bir iletişim gerçekleştirmiş olurlar.

#### **5. Multithread Senkronizasyon**

Birden fazla thread kullanmak iş yükünü hafifletip performans sağlasa da iyi bir senkronizasyon istemektedir. Kritik bölgeleri belirleyip hangi prosesin işlem yapması gerekiyorsa ona göre lock mekanizmalarımızı oluşturmalıyız. Bu programda da oluşturduğum üç thread için bu işlemleri uygulamaya çalıştım.

#### **6. Gerçekleştirilemeyenler ve Sınırlandırmalar:**

- Projeyi gerçekleştirirken kullanıcı girişi için de bir thread eklemiştim ve ona göre senkronizasyonumu ayarladım. Ancak programdan çıkış yapılacağı vakit bir thread içerisinden exit çalıştırdım. Exit yanında thread cancel ve lock destroy işlemleri de yapıldı ancak exit'i çalıştıran thread cancel yapılamadı, program kitleniyordu. Bu sebeple kapat() fonksiyonunun kopyasını bu threadin içerisinde kendisi hariç olacak şekilde tekrar yazdım.

#### **7. Yardım Aldığım Kaynaklar**

- Laboratuvar dersleri dökümanları ve örnek programlar.