

CMP3001 İşletim Sistemleri Proje (Senkronizasyon)

Süre: 24 Aralık Pazartesi (23:59)

BARIYER UYGULAMASI

Engeller, birden çok işlemin / iş parçacığının kodda belirli bir konumu diğerine kadar beklemesinin bir yoludur.

sistemdeki konu başlıkları aynı yere ulaştı. Bu ilkel, bir programcının yapmasına izin verir.

Bir uygulamanın yürütülmesini sağlamak için kullanılabilecek tek bir senkronizasyon noktası uygulayın.

adım davranışını kilitler. Bu işlemler paralel olarak dağıtılmış sistemlerde son derece popülerdir.

birçok paralel uygulamada baskın özellik. En sık dağıtılmış olarak görülürler

hafıza sistemlerinde, paylaşılan hafıza ortamlarında da kullanımları vardır.

Neden bir engel kullandığına bir örnek olarak, fiziksel olarak simüle eden bilimsel bir uygulama düşünün.

Zamanla olayları. Simülasyon, her biri verilenleri temsil eden bir dizi timesteps'e bölünmüştür.

simüle zamanın sınırı. Yürütülen simülasyon kodu her zaman aralığı için tamamen aynıdır,

ve simüle edilen sistemin o andaki durumunu temsil eden paylaşılan bir tamponu belirli bir noktada değiştirir.

zamanında. Şimdi bir çekirdek diğerleriyle senkronize edilmezse ve simüle etmeye başladığında ne olacağını hayal edin

geri kalan çekirdeklerden daha farklı bir zaman aralığı. Çünkü simülasyon tek bir paylaşılan tampon kullanıyor

durum verilerini depolamak için, simülasyon artık tamponun temsil edilmediğinden geçerli olmayacaktır.

Belirli bir zamanda doğru durumda. Bunun olmasını önlemek için, böyle bir uygulamanın ihtiyacı var

Her çekirdeğin aynı zaman aralığında çalıştığından emin olmak için. Bir çekirdek diğerlerinden önce biterse

diğerinden önce bir sonraki zaman adımına geçmek yerine diğerlerinin bitmesini beklemek gerekiyor.

çekirdekler. Engeller bu davranışı sağlamanın bir yolunu sunar.

Bu projede, ilkel basit bir bariyer uygulamanız gerekecektir. Bariyer ile başlatılacak

Bariyerin beklemesi gereken iplik sayısını gösteren belirli bir sayı.

Bir iş parçacığı bariyere gelişini işaret edecek ve bariyer () 'i çağırarak sorumlu olacak

iş parçacığı, uygun iş parçacığı sayısına kadar yürütmeye devam etmez (

init işlevi) aynı yere ulaştı. Tüm dışlar bariyere ulaştığında, bunlar

devam etmesine izin verilecek.

Kullandığınız kütüphane tarafından sağlanan Semafor ve / veya Mutex'i kullanmanız beklenir. (Örneğin

java.util.concurrent.Sava'da Java görüntülemesi)

Uygulama Fonksiyonları:

barrier_init

bariyer

Bariyer kullanımı için örnek sözde kod:

P0	P1	P2
{	{	{
A [0] = hesap 1 (0);	A [1] = hesap 1 (1);	A [2] = hesaplanan (2);
Bariyer ();	Bariyer ();	Bariyer ();
B [0] = hesaplanan (0, A);	B [1] = hesaplanan (1, A);	B [2] = hesaplanan (2, A);
}	}	}

```
ana(){
    barrier_init (3)

    P0 oluşturun;
    P1 oluşturun;
    P2 oluşturun;
    .
    .
    .
    B dizisini yazdır
}
```

Gönderim :

Yine de son teslim tarihinden önce **öğrenmeden** önce kodunuzu göndermelisiniz . Hiçbir gönderi kabul edilmeyecek

son teslim tarihinden sonra.

Not Verme:

Notunuz, kodun manuel olarak incelenmesine ve verilen soruya yazılı bir cevaba dayanacaktır.

Final sınavı sırasında.

Proje Notu = Kod% 60 + Proje Sorusu% 40