

Süreç Senkronizasyonu Nedir?

Çalışma Kağıdı

Süreç senkronizasyonu, birden fazla sürecin paylaşılan kaynaklara aynı anda erişmesini engelleyerek veya düzenleyerek veri yarışlarını (race conditions) önleyen ve sistemin doğru çalışmasını sağlayan bir mekanizmadır.

Sorular

1. Aşağıdakilerden hangisi süreç senkronizasyonu için kullanılan bir mekanizma DEĞİLDİR?

- A) Mutex
- B) Semafor
- C) İş Parçacığı (Thread)
- D) Kondisyon Değişkeni (Condition Variable)

2. Kritik bölgeye girmeden önce hangi işlem gerçekleştirilmelidir?

- A) Kaynak serbest bırakılmalı
- B) Kilidi elde etme (giriş protokolü)
- C) Süreç sonlandırılmalı
- D) Veri yedeklenmeli

3. Üretici-Tüketici probleminde, üretici dolu tampona veri eklemeye çalıştığında ne olur?

- A) Veri otomatik olarak silinir.
- B) Üretici süreci beklemeye alınır.
- C) Tüketici süreci uyarılır.
- D) Hata mesajı verilir.

4. İki sürecin paylaşılan bir değişkene erişimi nasıl senkronize edilir?

5. Üretici-Tüketici (Producer-Consumer) problemi nedir ve nasıl çözülür?

6. Tanımla: Süreç Senkronizasyonu Temel Amacı Nedir?

7. Tanımla: Kritik Bölge (Critical Section) Nedir?

8. Tanımla: Mutex Nedir?

Cevap Anahtarı

1. C) İş Parçacığı (Thread) — İş parçacığı (thread), bir sürecin alt birimidir ve senkronizasyon mekanizması değildir. Mutex, semafor ve kondisyon değişkenleri senkronizasyon mekanizmalarıdır.
2. B) Kilidi elde etme (giriş protokolü) — Kritik bölgeye girmeden önce, o bölgeye sadece bir sürecin girebilmesini sağlamak için bir giriş protokolü (genellikle kilidi alma) uygulanır.
3. B) Üretici süreci beklemeye alınır. — Üretici, eğer tampon doluysa, 'boş_yer' semaforunun değerinin sıfır olması nedeniyle beklemeye alınır ve bir tüketici veri aldığı anda tekrar çalışmaya başlar.
4. 1. Kritik bölge (critical section) tanımlanır: Paylaşılan değişkene erişilen kod parçasıdır. 2. Senkronizasyon mekanizması seçilir: Örneğin, mutex (mutually exclusive) kilitler veya semaforlar kullanılabilir. 3. Süreçler, kritik bölgeye girmeden önce kilidi almaya çalışır. Kilit alınırsa kritik bölgeye girilir, aksi halde beklenir. 4. Kritik bölgeden çıkarken kilit serbest bırakılır, böylece başka bir süreç kilidi alabilir.
5. 1. Problem: Bir üretici süreç, bir tampona (buffer) veri üretir ve bir tüketici süreç bu tampondan veriyi tüketir. Tamponun boyutu sınırlıdır. 2. Senkronizasyon İhtiyacı: Üretici dolu tampona veri ekleyemez, tüketici boş tampondan veri alamaz. 3. Çözüm: İki adet semafor kullanılır: 'boş_yer' (üretici için) ve 'dolu_yer' (tüketici için). Ayrıca tamponun erişimi için bir mutex kullanılır. Üretici 'boş_yer' semaforunu azaltır, tüketici 'dolu_yer' semaforunu azaltır. Her ikisi de tampon erişimi için mutex kullanır.
6. Veri tutarlılığını ve sistem kararlılığını sağlamak, veri yarışlarını önlemek.
7. Bir süreçte, paylaşılan kaynaklara erişilen kod parçasıdır.
8. Bir kaynağın aynı anda yalnızca bir süreç tarafından erişilebilmesini sağlayan bir kitleme mekanizmasıdır (karşılıklı dışlama).

Bounlu

Tüm kartlar, adım adım çözümler ve AI hoca desteği Notek uygulamasında.
Sınav tarihlerini Promy otomatik hatırlatıcıya çevirir.