

Süreçler ve İş Parçacıkları Nedir?

Çalışma Kağıdı

Süreç (process), kendi bağımsız bellek alanına sahip çalışan bir programdır. İş parçacığı (thread) ise bir sürecin içinde yer alan ve aynı bellek alanını paylaşan daha küçük yürütme birimidir.

Sorular

1. Hangi yapı, kendi bağımsız bellek alanına sahiptir?
A) İş Parçacığı (Thread)
B) Süreç (Process)
C) Kütüphane (Library)
D) Soket (Socket)
2. Hangi durum, birden fazla iş parçacığının aynı anda aynı veriye erişmeye çalıştığına ortaya çıkar?
A) Bellek sızıntısı
B) Yarış durumu (Race Condition)
C) İşlemci aşırı yüklenmesi
D) Ağ tıkanıklığı
3. İş parçacıklarının süreçlere göre temel avantajı nedir?
A) Daha fazla güvenlik
B) Daha az kaynak tüketimi ve daha hızlı geçiş
C) Bağımsız hata toleransı
D) Daha kolay hata ayıklama
4. Bir web tarayıcısı (örneğin Chrome) çalışırken süreç ve iş parçacığı nasıl temsil edilir?
5. Bir metin editöründe 'kaydet' işlemi ve 'yazı tipi değiştirme' işlemi nasıl yürütülebilir?
6. Çok çekirdekli bir işlemcide süreç ve iş parçacıklarının avantajları nelerdir?
7. Tanımla: Süreç (Process) Nedir?
8. Tanımla: İş Parçacığı (Thread) Nedir?
9. Tanımla: Süreçler Arası İletişim (IPC) Nedir?

Cevap Anahtarı

1. B) Süreç (Process) — Süreçler, diğer süreçlerden izole edilmiş kendi bellek alanlarına sahipken, iş parçacıkları aynı sürecin belleğini paylaşır.
2. B) Yarış durumu (Race Condition) — Yarış durumu, birden fazla iş parçacığının paylaşılan bir kaynağa erişim sırasının sonucunun beklenmedik olmasına neden olduğunda meydana gelir.
3. B) Daha az kaynak tüketimi ve daha hızlı geçiş — İş parçacıkları, süreçlere göre daha az bellek ve kaynak kullanır ve aralarında geçiş yapmak (context switching) daha hızlıdır.
4. Her bir Chrome sekmesi genellikle ayrı bir süreç olarak çalışır. Bu, bir sekmenin çökmesi durumunda diğerlerini etkilememesini sağlar. Ancak, her sekmenin içindeki farklı görevler (örneğin JavaScript çalıştırma, sayfa yükleme) kendi iş parçacıkları tarafından yönetilir. Bu iş parçacıkları, aynı sekmenin bellek alanını paylaşır.
5. Metin editörü bir süreçtir. 'Kaydet' işlemi arka planda ayrı bir iş parçacığı tarafından yürütülebilirken, kullanıcı arayüzü (yazı tipi değiştirme gibi) ana iş parçacığı tarafından yönetilebilir. Bu sayede kullanıcı, kaydetme işlemi devam ederken bile programı kullanmaya devam edebilir.
6. Çok çekirdekli işlemcilerde, birden fazla süreç veya aynı sürecin birden fazla iş parçacığı aynı anda farklı çekirdeklerde çalıştırılabilir. Bu, genel performansı artırır ve görevlerin daha hızlı tamamlanmasını sağlar. İş parçacıkları, süreçlere göre daha hafif oldukları için daha kolay ve hızlı bir şekilde yeni iş parçacıkları oluşturulup yönetilebilir.
7. Kendi bağımsız adres alanına sahip, çalışan bir program örneğidir. Kaynakları (bellek, dosya tanıtıcıları vb.) diğer süreçlerden izole edilmiştir.
8. Bir sürecin içindeki yürütme birimidir. Aynı sürecin diğer iş parçacıklarıyla bellek alanını ve kaynakları paylaşır.
9. Farklı süreçlerin birbirleriyle veri alışverişi yapmasını sağlayan mekanizmalardır (örn. borular, paylaşılan bellek).

Bounlu

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