GİRESUN ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
İŞLETİM SİSTEMLERİ DERSİ FİNAL SINAVI

Adı – Soyadı – Numarası:

Soru 1: Bilgisayarda birçok süreç (process) aynı anda çalışıyor sanırız. Fiziksel olarak sadece 1 adet işlemci bulunmasına rağmen birden fazla sürecin aynı anda çalışıyor görünmesi nasıl sağlanır?

Bilgisayar sistemlerinde, fiziksel olarak yalnızca bir işlemci (CPU) bulunmasına rağmen birden fazla sürecin aynı anda çalışıyor gibi görünmesi, çoklu görev (multitasking) ve bağlam anahtarlama (context switching) mekanizmaları sayesinde sağlanır. Bu süreç, işletim sisteminin süreç yönetimi ve çizelgeleme (scheduling) yetenekleriyle gerçekleştirilir.

Soru 2: Bilgisayarda çalışan süreçlere ihtiyaç duymaları halinde, gerçekte var olan fiziksel bellek miktarından daha fazla bellek alanı nasıl tahsis edilir?

Bilgisayarda çalışan süreçlere, fiziksel bellek (RAM) miktarından daha fazla bellek alanı tahsis edilmesi, sanal bellek (virtual memory) mekanizması sayesinde mümkün olur. Sanal bellek, işletim sisteminin süreçlere fiziksel bellekten bağımsız olarak geniş bir bellek alanı sunmasını sağlar. Bu, hem fiziksel belleğin verimli kullanılmasını hem de süreçlerin ihtiyaç duyduğu bellek miktarının karşılanmasını mümkün kılar.

Soru 3: Kilitlenme (deadlock) nedir? Bir sistemde kilitlenmenin oluşması için gerekli dört koşul nelerdir? Her koşulu, gerçek dünyadan bir örnekle (örneğin, bir trafik senaryosu) açıklayınız.

Kilitlenme, bir işletim sisteminde birden fazla sürecin, birbirlerinin serbest bırakmasını beklediği kaynaklar nedeniyle süresiz olarak bekleme durumunda kalmasıdır. Bu durumda, hiçbir süreç ilerleyemez ve sistemde bir tıkanıklık oluşur. Kilitlenme, özellikle paylaşılan kaynakların (örneğin, bellek, G/Ç cihazları, kilitler) kullanıldığı çoklu süreç sistemlerinde yaygın bir sorundur. Örnek: İki sürecin (P1 ve P2) iki kaynağı (R1 ve R2) paylaşmaya çalıştığını düşünelim. P1, R1'i tutuyor ve R2'yi bekliyor; P2, R2'yi tutuyor ve R1'i bekliyor. Her iki süreç de diğerinin kaynağı serbest bırakmasını beklediği için ilerleyemez, bu da kilitlenmeye yol açar.

Kilitlenme, aşağıdaki dört koşulun aynı anda sağlanması durumunda meydana gelir. Bu koşullar, Coffman koşulları olarak bilinir:
1. Karşılıklı Dışlama (Mutual Exclusion): En az bir kaynağın, aynı anda yalnızca bir süreç tarafından kullanılabilecek şekilde tutulması gerekir. Yani, kaynak paylaşılamaz ve bir süreç kaynağı kullandığında, diğer süreçler bu kaynağı kullanamaz.
2. Tut ve Bekle (Hold and Wait): Bir süreç, en az bir kaynağı tutarken (kullanırken), aynı anda başka bir kaynağı elde etmek için beklemelidir. Yani, süreç mevcut kaynakları serbest bırakmadan yeni kaynaklar talep eder.
3. Önleme Yok (No Preemption): Bir süreç tarafından tutulan kaynaklar, o süreç gönüllü olarak serbest bırakmadıkça zorla alınamaz (önlenemez). İşletim sistemi, kaynağı süreçten geri alamaz.
4. Dairesel Bekleme (Circular Wait): Bir grup süreç, bir döngü oluşturacak şekilde birbirlerinin tuttuğu kaynakları bekler. Örneğin, P1 R2'yi, P2 R3'ü, P3 R1'i beklerken, P1 R1'i, P2 R2'yi, P3 R3'ü tutuyordur.

Soru 4: Kilitlenme koşullarından her birini önlemek için uygulanabilecek bir strateji öneriniz.

1. Karşılıklı Dışlamayı (Mutual Exclusion) Önleme: Kaynakların yalnızca bir süreç tarafından kullanılmasını gerektiren karşılıklı dışlama koşulunu ortadan kaldırmak için, kaynaklar birden fazla sürecin eşzamanlı erişimine açık hale getirilir. Örneğin, bir yazıcı kaynağı için, yazdırma işleri bir kuyruk (spooler) aracılığıyla yönetilebilir, böylece yazıcıya doğrudan erişim yerine işler sırayla işlenir.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 02.06.2025

2. Tut ve Bekle (Hold and Wait) Koşulunu Önleme: Bir sürecin kaynak tutarken başka bir kaynağı beklemesini önlemek için, süreçlerin ihtiyaç duyacağı tüm kaynakları yürütme öncesinde bir kerede tahsis etmesi zorunlu kılınır. Eğer bir süreç gerekli tüm kaynakları alamıyorsa, hiçbir kaynağı almadan bekler.

3. Önleme Yok (No Preemption) Koşulunu Önleme: Bir sürecin tuttuğu kaynakların, işletim sistemi tarafından zorla geri alınabileceği bir mekanizma uygulanır. Eğer bir süreç, ihtiyaç duyduğu başka bir kaynağı alamıyorsa, mevcut kaynakları serbest bırakılır ve başka bir sürece tahsis edilir. Daha sonra süreç, tüm kaynakları tekrar talep edebilir.

4. Dairesel Bekleme (Circular Wait) Koşulunu Önleme: Dairesel beklemeyi önlemek için, kaynaklara bir sıralama (örneğin, numaralandırma) atanır ve süreçlerin kaynakları yalnızca bu sırayla talep etmesi zorunlu kılınır. Böylece, bir süreç daha düşük sıralı bir kaynağı tutarken daha yüksek sıralı bir kaynağı talep edemez, bu da dairesel bekleme zincirini kırar.

Soru 5: İşletim sisteminde bellek yönetimi (memory management) bileşenine neden ihtiyaç duyulur?

Bilgisayarda çalışan her süreç (process), kodunu, verilerini ve yığınını (stack) saklamak için bellek alanına ihtiyaç duyar. Fiziksel bellek (RAM) sınırlı bir kaynaktır ve birden fazla süreç bu kaynağı paylaşmak zorundadır. Bellek yönetimi, hangi sürecin hangi bellek bölgesini kullanacağını belirler ve belleğin süreçlere adil ve verimli bir şekilde tahsis edilmesini sağlar.

Her süreç, kendi bellek alanında çalışmalı ve diğer süreçlerin bellek alanına izinsiz erişmemelidir. Bellek yönetimi, süreçler arasında izolasyon sağlar ve bir sürecin başka bir sürecin verilerini yanlışlıkla veya kötü niyetle değiştirmesini önler.

Süreçler bellek tahsis edip serbest bıraktıkça, fiziksel bellekte boşluklar (parçalanma) oluşabilir. Dahili parçalanma (internal fragmentation), tahsis edilen bellek alanının tam kullanılmaması durumunda; harici parçalanma (external fragmentation), bellekteki boş alanların küçük ve dağınık olması durumunda ortaya çıkar. Bellek yönetimi, bu parçalanmayı en aza indirerek belleğin verimli kullanılmasını sağlar.

Fiziksel bellek, süreçlerin toplam bellek taleplerini karşılamakta yetersiz kalabilir. Bellek yönetimi, sanal bellek mekanizmasıyla süreçlere fiziksel bellekten daha büyük bir adres alanı sunar. Bu, sabit diskteki takas alanı (swap space) kullanılarak yapılır ve süreçlerin fiziksel bellek sınırlarını aşması mümkün olur.

Bazı durumlarda, süreçlerin ortak verilere veya kütüphanelere erişmesi gerekir (örneğin, paylaşılan kütüphaneler gibi). Bellek yönetimi, bu paylaşımı güvenli ve verimli bir şekilde organize eder, böylece aynı veri birden fazla kopya yerine tek bir bellek bölgesinde tutulur.

Soru 6: Bellek yönetiminde mantıksal adres ve fiziksel adres kavramlarını açıklayınız.

Mantıksal adres, bir sürecin kendi perspektifinden gördüğü ve kullandığı adreslerdir. Süreç, belleğe erişirken fiziksel belleğin gerçek konumlarını bilmez; bunun yerine, kendi sanal adres alanında tanımlı mantıksal adresleri kullanır. Mantıksal adresler, genellikle bir süreç başlatıldığında 0'dan başlayan bir adres aralığıdır.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 02.06.2025

Fiziksel adres, belleğin (RAM) gerçek, donanımsal konumunu temsil eden adrestir. Fiziksel adresler, fiziksel belleğin belirli bir baytına veya bellek hücresine doğrudan işaret eder ve donanım (örneğin, RAM çipi) tarafından anlaşılır.

İşletim sistemi, mantıksal adresleri fiziksel adreslere çevirir. Bu işlem, Bellek Yönetim Birimi (MMU) ve sayfa tabloları (page tables) aracılığıyla gerçekleştirilir. Mantıksal adresler, süreçlere kullanıcı dostu ve güvenli bir arayüz sunarken, fiziksel adresler, donanımın gerçek kaynaklarını yönetir.

Soru 7: Bellekte boş alanlar hangi veri yapıları ile tutulabilir? Açıklayınız.

İşletim sistemlerinde bellek yönetiminde, boş bellek alanlarının (free memory) izlenmesi ve tahsis edilmesi için çeşitli veri yapıları kullanılır. Bu veri yapıları, boş alanların konumlarını, boyutlarını ve tahsis durumlarını etkili bir şekilde temsil ederek belleğin verimli kullanılmasını sağlar. Boş alanları tutmak için kullanılan başlıca veri yapıları bağlı liste (linked list), bit eşleme (bitmap), ağaç yapıları (tree structures) gibi yapılardır.

Soru 8: Bellek yönetiminde dahili ve harici parçalanma arasındaki farkı açıklayınız.

Dahili parçalanma, bir sürece tahsis edilen bellek bloğunun, sürecin ihtiyaç duyduğundan daha büyük olması ve tahsis edilen alanın bir kısmının kullanılmadan kalması durumunda meydana gelir. Bu, tahsis edilmiş bir bellek bloğu içinde "boş" kalan alanların varlığıdır.

Harici parçalanma, bellekte toplamda yeterli boş alan olmasına rağmen, bu boş alanların küçük ve dağınık parçalar halinde olması nedeniyle bir sürece tahsis edilememesi durumunda meydana gelir. Bu, boş bellek bloklarının ardışık olmaması ve büyük bir süreç için uygun bir bloğun bulunamamasıdır.

Soru 9: Fiziksel bellekte olmayan bir sayfaya erişilmeye çalışıldığında neler olur? Adım adım açıklayınız.

Bir süreç, programında belirli bir mantıksal adrese (örneğin, 0x1234) erişmeye çalışır. Bu adres, sanal adres alanındaki bir sayfaya karşılık gelir. Süreç, bu adresin fiziksel bellekte olup olmadığını bilmez; yalnızca kendi sanal adres alanını kullanır.

CPU, mantıksal adresi fiziksel adrese çevirmek için Bellek Yönetim Birimi (MMU)'ni kullanır. MMU, süreç için oluşturulan sayfa tablosuna (page table) bakar ve ilgili sanal sayfanın fiziksel bellekteki karşılık gelen çerçevesini (frame) arar.

MMU, sayfanın fiziksel bellekte olmadığını tespit ettiğinde, bir sayfa hatası kesmesi üretir. Bu kesme, CPU'nun mevcut talimatı durdurmasını ve kontrolü işletim sistemine devretmesini sağlar.

İşletim sisteminin sayfa hatası işleyici (page fault handler) devreye girer ve hatanın türünü analiz eder. İşletim sistemi, erişilmeye çalışılan sayfanın nerede olduğunu ve nasıl getirileceğini belirler.

Eğer fiziksel bellek doluysa, işletim sistemi yeni sayfa için yer açmak zorundadır. Bu, bir veya daha fazla mevcut sayfanın fiziksel bellekten çıkarılmasını gerektirir.

İşletim sistemi, eksik sayfayı sabit diskten (takas alanı veya dosya sistemi) fiziksel belleğe yükler.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 02.06.2025

Sayfa fiziksel belleğe yüklendikten sonra, işletim sistemi süreci yeniden başlatır.

Soru 10: Sayfa yer değiştirme (page replacement) algoritmalarından 3 tanesini açıklayınız.

İlk Giren İlk Çıkar (First-In, First-Out - FIFO): FIFO, fiziksel bellekte en uzun süredir bulunan sayfanın çıkarılmasını seçer. Bellek, bir kuyruk (queue) gibi yönetilir; yeni bir sayfa eklendiğinde kuyruğun sonuna yerleştirilir ve yer açılması gerektiğinde kuyruğun başındaki (en eski) sayfa çıkarılır.

Optimal Sayfa Yer Değiştirme (Optimal Page Replacement): Optimal algoritma, gelecekte en uzun süre kullanılmayacak sayfanın çıkarılmasını seçer. Bu, teorik olarak en az sayfa hatasını üreten algoritmadır, ancak gelecekteki sayfa referanslarını bilmek gerektiğinden pratikte uygulanamaz.

En Eskiden Kullanılan (Least Recently Used - LRU): LRU, en uzun süredir kullanılmayan sayfanın çıkarılmasını seçer. Bu, geçmiş kullanım desenlerine dayanarak, yakın zamanda kullanılan sayfaların tekrar kullanılma olasılığının yüksek olduğu varsayımına dayanır (yerellik ilkesi(locality)).

En Sık Kullanılan (Most Frequently Used - MFU): MFU, en sık kullanılan sayfanın çıkarılmasını seçer. Bu, sık kullanılan bir sayfanın zaten çok kez erişildiği ve yakın gelecekte daha az kullanılabileceği varsayımına dayanır.

Saat Algoritması (Clock Algorithm) / İkinci Şans (Second Chance): Saat algoritması, LRU'nun daha basit bir yaklaşık versiyonudur. Sayfalar bir dairesel liste (saat) içinde tutulur ve her sayfanın bir referans biti (reference bit) vardır. Yakın zamanda kullanılan sayfalar korunur, kullanılmayanlar çıkarılır.