



Adı – Soyadı – Numarası:

Soru 1: Süreçler arası kilitlenme olması için gerekli 4 şart nedir? Kısaca açıklayınız.

Kilitlenme (deadlock), bir grup sürecin, birbirlerinin kaynaklarını bekleyerek sonsuz bekleme durumuna girdiği durumdur. Kilitlenmenin gerçekleşmesi için dört gerekli ve yeterli şart bulunur.

- **Karşılıklı Dışlama (Mutual Exclusion):** En az bir kaynak, aynı anda yalnızca bir süreç tarafından kullanılabilir olmalıdır. Yani, kaynaklar paylaşılamaz ve sadece tek bir süreç tarafından tutulabilir.
- **Kaynak Tutma ve Bekleme (Hold and Wait):** Bir süreç, en az bir kaynağı tutarken aynı zamanda başka kaynakları da bekliyor olmalıdır. Bu durumda süreç, sahip olduğu kaynakları serbest bırakmadan yeni kaynaklar talep eder.
- **Geri Alamama, Kesme Yok (No Preemption):** Süreçlerden tuttukları kaynaklar zorla geri alınamaz. Kaynaklar, sadece işini tamamlayan süreç tarafından serbest bırakılabilir.
- **Döngüsel Bekleme (Circular Wait):** Süreçler arasında döngüsel bir bekleme durumu olmalıdır. Yani, P1 süreci bir kaynağı beklerken, bu kaynağı tutan P2 süreci başka bir kaynağı bekler ve bu döngü devam eder, en sonunda son süreç de P1'in tuttuğu kaynağı bekler.

Bu dört şartın hepsi aynı anda gerçekleşirse, sistemde kilitlenme meydana gelir. Bu şartların herhangi birinin ihlal edilmesi, kilitlenmenin oluşmasını engeller.

Soru 2: Bellek yönetiminde fiziksel adres yerine mantıksal adres kullanımının (bellek soyutlaması) getirdiği avantaj ya da dezavantajlar nedir?

Bellek yönetiminde mantıksal adreslerin (logical address) kullanımı, fiziksel adreslere (physical address) doğrudan erişim yerine bir soyutlama katmanı ekleyerek birçok avantaj ve bazı dezavantajlar sunar.

Mantıksal adresleme, süreçlerin birbirlerinin bellek alanlarına erişimini engeller. İşletim sistemi, her sürecin yalnızca kendi mantıksal adres alanında çalışmasını sağlayarak bellek korumasını gerçekleştirir. Mantıksal adresleme, programcılarının fiziksel bellek detaylarından soyutlanmasını sağlar. Bu soyutlama, uygulamaların taşınabilirliğini artırır ve farklı donanım yapılandırmalarında çalışabilir hale getirir. Mantıksal adresleme, süreçlerin belirli bellek alanlarını paylaşmalarını sağlar. Örneğin, kütüphaneler ve dinamik bağlama (dynamic linking) bu şekilde etkinleştirilir. Mantıksal adresleme, bellek parçalanmasını (fragmentation) azaltarak bellek kullanımını daha verimli hale getirir. Sayfalama (paging) ve kesimleme (segmentation) gibi teknikler, bellek tahsisini ve kullanımını optimize eder. Sanal bellek kullanımı, fiziksel bellek miktarının ötesinde bellek kullanımına izin verir. Bu, disk üzerindeki bellek alanının RAM gibi kullanılmasını sağlar.

Mantıksal adresleme ve sanal bellek yönetimi, ek işlemci döngüleri ve bellek erişim süreleri gerektirir. TLB (Translation Lookaside Buffer) ve sayfa tabloları (page tables) gibi yapılar, bellek erişimlerinde ek gecikmelere neden olabilir. Sayfalama, kesimleme ve bellek koruma mekanizmaları gibi yapılar, işletim sisteminin karmaşıklığını artırır. Mantıksal adresleme ve sanal bellek yönetimi, ek bellek alanı gerektirir. Sayfa tabloları ve segment tabloları gibi yapılar, ek bellek tüketimine yol açar.

Soru 3: Bellekte boş alan yönetimi (free space management) yapabilmek için kullanılacak iki adet veri yapısı öneriniz. Avantaj ve dezavantajlarını tartışınız.

Serbest blok listesi (free block list), serbest bellek bloklarının boyut ve adres bilgilerini içeren bir liste yapısıdır. Bu yapı, tipik olarak blok boyutlarına veya adreslerine göre sıralanabilir. Uygun algoritmalarla, belirli boyuttaki bloklara hızlıca erişim sağlanabilir. Bellek blokları serbest bırakıldığında veya tahsis



edildiğinde liste güncellenebilir. Liste, bellek bloklarının dinamik olarak eklenip çıkarılmasını destekler. Uygun boyuttaki bir blok bulunması zaman alabilir, ancak optimize edilebilir. Liste yapısını tutmak için ekstra bellek alanı gerekir.

Bit eşlem (bitmap), bellek bloklarının dolu veya boş olup olmadığını gösteren bitlerden oluşan bir dizidir. Her bit, belirli bir bellek bloğunun durumunu temsil eder (0: boş, 1: dolu). Çok küçük bir bellek kullanarak büyük bellek alanlarının yönetimini sağlar. Uygun algoritmalarla, boş bloklar hızlıca bulunabilir. Bit eşlem yapısı, uygulama ve yönetim açısından oldukça basittir. Bit eşlem genellikle sabit boyutlu bloklarla çalışır, bu da bellek parçalanmasına yol açabilir. Çok büyük bellek alanları için bit eşlem yönetimi karmaşık olabilir.

Soru 4: Sayfa Tablosu'nun (Page Table) görevi nedir? Performans sorunlarına yönelik yapılan donanımsal ve yazılımsal değişiklikler nelerdir?

Fiziksel Adres Dönüşümü: Sayfa tablosu, sanal bellek adreslerini fiziksel bellek adreslerine dönüştürmek için kullanılır. Sanal bellek, işletim sistemi tarafından yönetilen bir alanı ifade ederken, fiziksel bellek ise gerçek RAM (Random Access Memory) üzerindeki fiziksel adresleri temsil eder. İşlemci sanal bellek adreslerine erişirken, sayfa tablosu aracılığıyla bu adresler fiziksel bellek adreslerine çevrilir.

Sayfa İzleme ve Yönetim: Sayfa tablosu, her sanal sayfa için fiziksel bellekteki karşılık gelen sayfa çerçevesini (page frame) izler. Sanal bellekte bulunan her bir sayfa, fiziksel bellekte bir veya daha fazla sayfa çerçevesine karşılık gelir. Sayfa tablosu bu eşleşmeyi takip eder ve gerektiğinde sayfa değişimi (page swapping) işlemlerini yönetir.

Hafıza Yönetimi: İşletim sistemi, sayfa tablosu üzerinden sanal belleği fiziksel belleğe ve bellek dışı depolama (disk gibi) arasında yönetir. Sayfa tablosu, sanal bellekteki sayfaların fiziksel bellekteki yerlerini belirler ve yönetirken, bu işlem sırasında hafıza kullanımı ve veri erişim süreçlerini optimize eder.

Performans sorunlarına yönelik yapılan donanımsal ve yazılımsal değişiklikler ise şunları içerebilir:

Bellek Hiyerarşisi: Bellek erişim hızını artırmak için daha hızlı bellek türlerinin kullanılması (örneğin, önbellek belleği).

Bellek Yönetimi Birimleri: İyileştirilmiş bellek yönetim birimleri ve algoritmaları kullanarak bellek erişimini optimize etmek.

İşlemci Tasarımı: Donanım düzeyinde, sayfa tablosu erişimlerini hızlandırmak için işlemci tasarımında değişiklikler yapmak.

Bellek Yönetimi Algoritmaları: Daha verimli sayfa değişim (page swapping) ve sayfa değiştirme (page replacement) algoritmalarının kullanılması.

Sanal Bellek Yönetimi Politikaları: Bellek yönetimi politikalarının gözden geçirilmesi ve performans iyileştirmeleri için yeni politikaların geliştirilmesi.

Optimizasyon ve Ayarlamalar: İşletim sistemi seviyesinde yapılan optimizasyonlar, bellek kullanımını etkinleştirmek ve veri erişim süreçlerini hızlandırmak için yapılan ayarlamalar.

Soru 5: Adres alanı: 32 bit olsun. Sayfa boyutu: 1 KB olsun. Sayfa tablosunda her bir eleman 32 bit (4 byte) yer kaplasın. Bu durumda sayfa tablosunun boyutu ne kadardır?

Adres alanı: 32 bit

Sayfa boyutu: 1 KB (1024 byte)

Sayfa tablosunda her bir eleman 32 bit (4 byte) yer kaplar.



1 KB'lik sayfa boyutu olduğunda, adres alanı içinde $2^{32-10} = 2^{22}$ tane sayfa bulunur (her biri 1 KB).
Toplam sayfa sayısı = 2^{22}
Her bir sayfa tablosu ögesi = 4 byte
Sayfa tablosunun toplam boyutu = $2^{22} \times 4 \text{ byte} = 16 \text{ Mbyte}$

Soru 6: Sayfa yer değiştirme (Page Replacement) algoritmalarından dört tanesini açıklayarak yazınız.

Optimal sayfa yer değiştirme algoritması, ideal bir durumda hangi sayfanın diskten çıkarılması gerektiğini belirlemek için kullanılır. Algoritma, her bir sayfa erişiminde gelecekteki kullanımı tahmin ederek en uzun erişim süresine sahip olan sayfayı seçer.

İkinci şans sayfa yer değiştirme algoritması, kullanılan ve kullanılmayan sayfaları ayırt etmek için "referans" veya "ikinci şans" biti kullanır. Algoritma, fiziksel bellekte bulunan her sayfa için bir referans biti bulundurur. İlk önce işaretlenmiş olup olmadığı kontrol edilir ve eğer işaretlenmişse, işareti kaldırılır ve sayfa döngüsü devam eder.

En eski kullanılan (LRU – Least recently used) sayfa yer değiştirme algoritması, en son erişilen sayfanın en az kullanılan sayfa olduğunu varsayar. Yani, en son kullanılan sayfaların en erken döndürülmesini bekler. Bu algoritma, fiziksel bellekteki her sayfaya bir zaman damgası atayarak çalışır ve her erişimde bu damgayı günceller.

Çok kullanılan (MRU – Most recently used) sayfa yer değiştirme algoritması, en son erişilen sayfanın en çok kullanılan sayfa olduğunu varsayar. Yani, en son kullanılan sayfaların en geç döndürülmesini bekler. Bu algoritma, LRU algoritmasının tam tersidir.

Çalışma kümesi (Working Set) yer değiştirme algoritması, programın çalışma kümesini belirlemek için bir zaman penceresi (sliding window) kullanır. Bu algoritma, programın gereksinim duyduğu sayfaların birbirleriyle etkileşimini dikkate alır.

Soru 7: Bir dosyaya erişmek için adres (path) kullanılır. Mutlak ve bağıl adres nedir?

Mutlak adres, dosya veya dizinin kök dizininden başlayarak tam yolunu belirtir. Mutlak adresler, dosyanın veya dizinin tam yerini belirler ve genellikle sistem genelinde benzersizdir.

Örneğin, UNIX/Linux işletim sistemlerinde mutlak adres / ile başlar ve dizinler arası ilerler:

- /home/user/documents/example.txt
- /var/log/syslog

Windows işletim sisteminde mutlak adres C:\ veya başka bir sürücü harfi ile başlar:

- C:\Users\user\Documents\example.txt
- D:\Program Files\program.exe

Bağıl adres, mevcut çalışma dizininden dosya veya dizine olan yolunu belirtir. Bağıl adresler genellikle mevcut süreç veya uygulama tarafından çalışma dizini belirlenir ve bu dizinden dosyalara erişim sağlar.

Örneğin, bir çalışma dizini ~/documents/ olarak varsayıldığında:

- example.txt (~/documents/example.txt anlamına gelir)
- subfolder/image.jpg (~/documents/subfolder/image.jpg anlamına gelir)

Soru 8: Güç kesintilerine karşı dosya sistemini korumanın yöntemlerini kısaca açıklayınız.



Günlükleme, dosya sisteminde yapılan her değişikliğin bir günlük (journal) dosyasına kaydedilmesi yöntemidir. Bu sayede, bir güç kesintisi veya sistem çökmesi durumunda, sistem yeniden başlatıldığında yarıda kalan işlemler tamamlanabilir veya geri alınabilir.

Copy-on-Write, verinin doğrudan üzerine yazmak yerine, yeni veriyi yeni bir konuma yazar ve ardından referansları günceller. Bu yöntem, veri tutarlılığını sağlar.

Verilerin birden fazla kopyasını farklı disklerde saklamak. RAID 1 (Mirroring) bu yöntemi kullanır.

Soru 9: Bellek Eşlemeli Giriş Çıkışın avantajları ve dezavantajları nelerdir?

Bellek eşlemeli giriş çıkış (Memory-mapped I/O) yönteminde, G/Ç aygıtlarına erişim, cihazların bellekteki belirli adreslere eşlenmesiyle yapılır. Bu yöntem, G/Ç aygıtlarını doğrudan bellek adres alanına dahil eder, böylece CPU, G/Ç aygıtlarına erişim için standart bellek erişim komutlarını kullanabilir.

G/Ç aygıtlarına erişim, bellek erişimi gibi olduğu için, yazılım geliştiricilerinin aygıtlarla etkileşim kurmasını basitleştirir. Bu durum, düşük seviyeli G/Ç işlemlerini yönetmeyi kolaylaştırır. G/Ç işlemleri için ek komut setine ihtiyaç duyulmaz. CPU, bellek işlemleriyle G/Ç aygıtlarına erişebilir, bu da komut kümesini sadeleştirir. Bellek erişim hızında G/Ç işlemleri yapılabilir, bu da yüksek hızlı veri transferleri için avantaj sağlar. Aygıt sürücülerinin yazılması ve yönetimi daha kolaydır.

G/Ç aygıtlarının bellek alanını kullanması, bellek adres alanı tahsisinde karmaşıklığa neden olabilir ve adres çakışmaları yaşanabilir. Bellek erişim hataları G/Ç aygıtlarına zarar verebilir. Bu durum, bellek ve G/Ç aygıtları arasında koruma ve güvenlik sorunlarına yol açabilir. Bellek eşlemeli G/Ç desteklemek için donanımın uygun şekilde tasarlanması gerekir. Bu da ek maliyet ve tasarım karmaşıklığı getirebilir.

Soru 10: Giriş çıkış birimlerinde sıkça karşılaşılan veri kayıplarını/bozulmalarını önlemek için kullanılan yöntemleri kısaca açıklayınız.

Hata Kontrolü (Error Checking): Veri iletimi sırasında hata kontrolü algoritmaları kullanılarak veri bütünlüğü sağlanır. Örneğin, CRC (Cyclic Redundancy Check) veya checksum gibi yöntemlerle veri paketleri üzerinde hata algılaması yapılır. Bu yöntemler, iletilen verinin doğru olup olmadığını kontrol ederek veri kaybını veya bozulmasını tespit eder.

Tekrarlama (Retransmission): Veri iletimi sırasında hata oluşması durumunda iletişim protokolleri, iletimi tekrarlayarak veri kaybını telafi etmeye çalışır. Örneğin, TCP (Transmission Control Protocol) gibi protokollerde veri paketlerinin doğruluğu sağlanamazsa, kayıp olan paketler yeniden gönderilir.

Akış Kontrolü (Flow Control): Veri iletim hızının alıcı ve gönderici arasında dengelenmesini sağlayarak, veri kaybını ve bozulmasını önler. Bu yöntem, iletim sırasında veri akışını düzenleyerek aşırı yüklenme veya hız farklılıklarından kaynaklanan veri kayıplarını önler.

Paketleme ve Tamponlama (Packetization and Buffering): Veri paketlerinin doğru bir şekilde paketlenmesi ve hedef cihazda geçici olarak tamponlanmasıyla veri kaybı önlenir. Tampon, hızlı veri akışında veya anlık yüksek veri taleplerinde verilerin geçici olarak depolanarak düzenlenmesini sağlar.

CRC (Cyclic Redundancy Check): CRC, veri iletimi sırasında kullanılan bir hata kontrol algoritmasıdır. Gönderici ve alıcı arasında iletilen verinin bütünlüğünü sağlamak için kullanılır. CRC algoritması, veri



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

blokları üzerinde matematiksel işlemler yaparak bir hata dedektörü olarak çalışır ve veri bozulmalarını tespit eder.