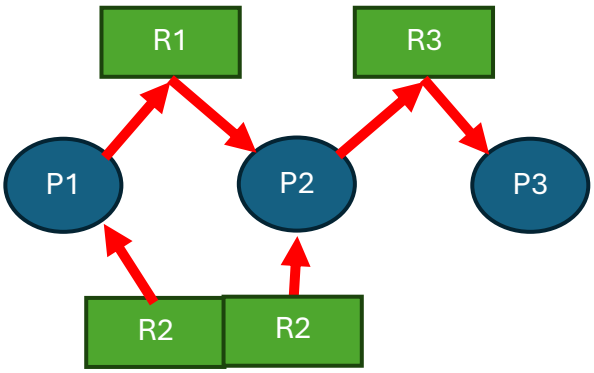
GİRESUN ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
İŞLETİM SİSTEMLERİ DERSİ FİNAL SINAVI

Adı – Soyadı – Numarası:

Soru 1: Aşağıda durumu verilen kaynak tahsis çizgesini (resource allocation graph) çiziniz.
P1, bir R2 kaynağını tutar ve R1 kaynağını bekler.
P2, R1 kaynağını ve bir R2 kaynağını tutar ve R3 kaynağını bekler.
P3, R3 kaynağını tutar.

Kaynak tahsis çizgesi (resource allocation graph), sistemdeki süreçler ve kaynaklar arasındaki ilişkiyi görsel olarak gösteren bir çizge türüdür. Bu çizgede, süreçler ve kaynaklar düğümlerle temsil edilir ve süreçlerden kaynaklara veya kaynaklardan süreçlere doğru kenarlar çizilir. Süreçler P1, P2 ve P3 olarak temsil edilir. Kaynaklar R1, R2 ve R3 olarak temsil edilir.

P1 --> R1 (bekliyor)
P2 --> R3 (bekliyor)
R2 --> P1 (tutuyor)
R1 --> P2 (tutuyor)
R2 --> P2 (tutuyor)
R3 --> P3 (tutuyor)



Soru 2: Banker's algoritması nasıl çalışır ve ne için kullanılır?

Banker's algoritması, kaynak tahsisi ve kilitlenme (deadlock) önleme için kullanılır. Algoritma, Hollandalı bilgisayar bilimci Edsger Dijkstra tarafından geliştirilmiştir ve adını bankacılar tarafından müşterilere kredi verirken kullanılan bir sistemden almıştır. Banker's algoritması, bir sistemin güvenli durumda kalmasını sağlar ve kilitlenmelerin oluşmasını önler.

Banker's algoritması, her sürecin maksimum ihtiyaçlarını önceden belirleyerek ve süreçlerin mevcut kaynak taleplerini değerlendirerek çalışır. Algoritma, kaynak tahsisi yapmadan önce, tahsisin sistemin güvenli durumda kalıp kalmayacağını kontrol eder. Güvenli bir durum, tüm süreçlerin ihtiyaç duyduğu kaynakları alıp tamamlanabileceği ve sonunda kaynakları serbest bırakabileceği bir durumdur.

Soru 3: Bellek yönetim biriminde yer alan yer değiştirme yazmacı (relocation register) görevi nedir?

Bellek yönetim biriminde yer alan yer değiştirme yazmacı (relocation register), işlemci tarafından yürütülen programların mantıksal adreslerini fiziksel adreslere çevirmek için kullanılan önemli bir bileşendir. Programlar, bellek referanslarını mantıksal adresler olarak yapar. Yer değiştirme yazmacı, bu mantıksal adreslere eklenerek fiziksel adreslerin hesaplanmasını sağlar. Örneğin, bir program belirli bir bellek konumuna erişmek istediğinde, mantıksal adres yer değiştirme yazmacının değeri ile toplanarak fiziksel adres elde edilir. Fiziksel adres = Mantıksal adres + Yer değiştirme yazmacının değeri.

Yer değiştirme yazmacı, süreçlerin yalnızca kendisine tahsis edilmiş bellek alanlarına erişmelerini sağlar. Süreçler, kendi mantıksal adres alanları içinde serbestçe çalışabilirler ancak işletim sistemi tarafından tahsis edilen bellek alanının dışına çıkmaları engellenir.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 25.06.2024

Yer değiştirme yazmacı, aynı fiziksel bellek alanını farklı mantıksal adreslerle erişen süreçlere izin verir. Bu, ortak kullanılan kütüphaneler veya paylaşılan bellek alanları gibi senaryolarda kullanışlıdır.

Programlar, bellekte herhangi bir fiziksel konuma yerleştirilebilir ve çalıştırılabilir. Yer değiştirme yazmacı, programın başlangıç adresine bağlı olarak mantıksal adresleri doğru fiziksel adreslere çevirdiğinden, programların taşınabilirliği artırılır.

Soru 4: Statik ve dinamik kütüphane arasındaki farklar nelerdir?

Statik kütüphaneler, programın derleme zamanı sırasında kütüphane kodlarının doğrudan uygulamanın yürütülebilir dosyasına dahil edilerek kullanılır. Genellikle .lib (Windows) veya .a (Unix/Linux) uzantılı dosyalar olarak bulunur. Statik kütüphaneler, kütüphane kodlarının yürütülebilir dosyaya dahil edilmesi nedeniyle programın çalışması için ek dosyalara ihtiyaç duymaz. Kütüphane kodu derleme zamanı sırasında dahil edildiği için, kütüphanede yapılan değişiklikler uygulamayı etkilemez. Statik kütüphaneler, kütüphane kodlarının uygulamaya dahil edilmesi nedeniyle yürütülebilir dosya boyutunu artırır. Aynı kütüphaneyi kullanan birden fazla uygulama, her biri kendi kopyasını içerdiği için bellek kullanımında artışa neden olabilir. Kütüphanede bir hata veya güvenlik açığı varsa, her uygulamanın yeniden derlenmesi ve dağıtılması gerekir.

Dinamik kütüphaneler, programın çalıştırma zamanı sırasında yüklenir ve bağlanır. Genellikle .dll (Windows) veya .so (Unix/Linux) uzantılı dosyalar olarak bulunur. Dinamik kütüphaneler, kütüphane kodlarının yürütülebilir dosyaya dahil edilmemesi nedeniyle daha küçük dosya boyutlarına sahiptir. Aynı kütüphaneyi kullanan birden fazla uygulama, kütüphaneyi paylaşarak bellek kullanımını optimize eder. Kütüphanede bir hata veya güvenlik açığı varsa, sadece kütüphane dosyasının güncellenmesi yeterlidir. Dinamik kütüphaneler, programın çalışması için gerekli ek dosyalar olduğu için bu kütüphanelerin hedef sistemde mevcut olması gerekir. Kütüphanede yapılan değişiklikler, uyumsuzluklara veya beklenmeyen davranışlara neden olabilir.

Soru 5: Bellekte boş alanların bir veriyapısında tutulması gerekir. İki adet veri yapısı öneriniz.

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 6: Bellekte harici ve dahili parçalanma nedir?

Harici parçalanma, bellekte yeterince birçok küçük boşluk (fragment) bulunmasına rağmen, bu boşlukların gereksinim duyulan büyük bir boşluğu karşılayamayacak kadar dağılmış olması durumudur. Uygulamaların talep ettiği bellek blokları serbest veya ayrılmış olabilir, ancak bu boşluklar uygulamaların ihtiyacı olan

büyük blokları sağlayacak şekilde birleştirilemez. Bellek tahsisi ve serbest bırakma işlemleri sonucunda, boşluklar bellekte dağınık bir şekilde dağılmış olabilir. Harici parçalanma durumunda, bazı durumlarda küçük boşluklar kullanılabilirken, büyük bloklar için yeterli boşluk sağlanamayabilir. Bellek Yeniden Düzenleme (Memory Compaction): Bellekteki blokların yeniden düzenlenmesi ve boşlukların birleştirilmesiyle büyük bloklar için yeterli boşluk sağlanabilir. İleri Yönlü Tahsis (Forward Allocation): Bellekteki bir sonraki boş bloğun bir sonraki talep edilen blok için kullanılması.

Dahili parçalanma, bellekte tahsis edilen bir bloğun gerçek ihtiyacından daha büyük olması durumudur. Bu durumda, tahsis edilen blok içinde kullanılmayan veya gereksiz boşluklar bulunur. Bellekte belirli bir boyutta bir blok talep edilir, ancak bu blok gereksinimlerin altında kalan küçük bir veri setiyle doldurulur. Dinamik bellek tahsisi sırasında, talep edilen bloğun genellikle belirli bir minimum boyutta olması gerekliliği vardır, bu da gereksiz boş alanlara neden olabilir. Talep edilen bloğun gerçek boyutuna daha yakın olan bir boyutla tahsis edilmesi gerekir.

Soru 7: Erişilmek istenen sayfa bellekte yok ise hatalı sayfa (page fault) işlemi nasıl ele alınır?

Erişilmek istenen sayfa bellekte bulunmadığında (yani hatalı sayfa durumu), işletim sistemi bu durumu ele almak ve sayfayı fiziksel belleğe getirmek için belirli adımlar izler. İşletim sistemi veya bellek yöneticisi, bir programın sanal bellekteki bir sayfaya erişmeye çalıştığını tespit eder, ancak bu sayfa fiziksel bellekte (RAM) bulunmamaktadır. İlk olarak, bellekte boş çerçeve yoksa işletim sistemi bir sayfa değiştirme (page replacement) algoritması kullanarak bir veya daha fazla sayfayı RAM'den çıkararak boş bir çerçeve elde eder. Hatalı sayfa durumunda, disk üzerindeki sanal bellek sayfası (page file veya swap dosyası gibi) belleğe getirilir. Disk üzerindeki sayfa, ilgili fiziksel çerçeveye kopyalanır ve sayfa tablosu güncellenir. Bu işlem bir giriş/çıkış (I/O) işlemi gerektirir. Sayfa tablosunda ilgili sanal sayfa öğesi, artık fiziksel bellekteki doğru çerçeveyi (frame) işaret eder. Eğer sayfa değiştirme işlemi yapıldıysa, değiştirilen sayfa disk üzerinde de güncellenmelidir. Sayfa diskten RAM'e başarıyla getirildikten ve sayfa tablosu güncellendikten sonra, işletim sistemi komutu yeniden yürütür ve erişimi yeniden dener.

Soru 8: Adres alanı: 32 bit olsun. Sayfa boyutu: 4 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: 4 KB (4096 byte)
Sayfa tablosunda her bir eleman 32 bit (4 byte) yer kaplar.
4 KB'lik sayfa boyutu olduğunda, adres alanı içinde $2^{32-12} = 2^{20}$ tane sayfa bulunur (her biri 4 KB).
Toplam sayfa sayısı = 2^{20}
Her bir sayfa tablosu öğesi = 4 byte
Sayfa tablosunun toplam boyutu = 2^{20} x 4 byte = 4 Mbyte

Soru 9: Talebe bağlı sayfalama (demand paging) özelliği nedir?

Talebe bağlı sayfalama (demand paging), sanal bellek yönetimi içinde bir stratejidir ve modern işletim sistemlerinin çoğunda kullanılır. Bu özellik, fiziksel bellek (RAM) kullanımını optimize eder ve yalnızca gerektiği zaman sayfaları (veya blokları) belleğe yükler. Talebe bağlı sayfalama, sanal belleğe erişim talepleri doğrultusunda sayfaların (veya blokların) fiziksel belleğe yüklenmesini gerektirir. Bir programın çalıştırılması sırasında, işletim sistemi yalnızca gerektiği zaman sayfa yükler. Bu işlem sayesinde, programın tamamı bellekte olmadan başlatılabilir ve kullanıcının gerçekten eriştiği sayfalar bellekte tutulur. Bir sayfa talebi geldiğinde, sayfa tablosu kullanılarak ilgili sayfanın fiziksel bellekteki konumu belirlenir ve diskten RAM'e taşınır.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 25.06.2024

Soru 10: 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 11: Metin (text) ve ikili (binary) dosya tipi farkları nedir?

Metin dosyaları, insan tarafından okunabilir karakterleri (harfler, rakamlar, semboller, boşluklar vb.) içerir. Genellikle ASCII veya Unicode gibi karakter kodlamalarıyla kodlanır. Satır sonu karakterleri (newline, carriage return gibi) ile ayrılmış satırlardan oluşur. Düz metin içerir ve metin düzenleyiciler veya metin işleme programları tarafından kolayca düzenlenebilir.

İkili dosyalar, bilgisayar tarafından doğrudan anlaşılabilir verileri içerir. Genellikle işlenmiş veri, resim, ses, video gibi yapılandırılmış veri türlerini içerir. Belirli bir yapıya veya veri türüne göre düzenlenmiş bit düzeyinde verileri içerir. Özel kodlama veya yapılandırma ile saklanır ve genellikle doğrudan insanlar tarafından okunması zor veya imkansızdır. Daha küçük boyutlarda olabilir çünkü veri türlerine veya yapılarına göre optimize edilmiş veri saklama yöntemleri kullanılabilir.

Soru 12: r w x dosya izinleri ne anlama gelir?

Dosya izinleri UNIX/Linux işletim sistemlerinde dosyalara veya dizinlere erişim yetkilerini belirlemek için kullanılır. Bu izinler üç kategoriye ayrılır: kullanıcı (owner), grup (group) ve diğer (others). Her kategori için izinler r (read), w (write) ve x (execute) şeklinde belirtilir.

r (read - okuma izni): Dosya içeriğini okuma iznidir. Eğer bir dizinse, dizinin içeriğini listeleme iznidir.
w (write - yazma izni): Dosyaya yazma iznidir. Eğer bir dizinse, dizine dosya ekleme, dosya silme veya dosya adı değiştirme gibi işlemleri yapma iznidir.
x (execute - çalıştırma izni): Dosyanın yürütme (execute) iznidir. Eğer bir dizinse, dizinde bulunan dosyaları veya alt dizinleri çalıştırma iznidir.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 25.06.2024

Soru 13: Dosya kopyalayan bir program yazacak olsanız, bayt bayt mı yoksa 4 KB'lık parçalar halinde mi işlem yaparsınız? Hangi sistem çağrılarını kullanırsınız? Neden?

Dosyayı 4 KB'lık parçalar halinde okuyup yazmak, dosya sistemi ve disk sürücüsü için daha etkili bir işlem şeklidir. Diskler 4 KB veya daha büyük bloklar halinde veri okuma/yazma işlemlerinde daha iyi performans sağlar.

Dosya kopyalama işlemi için aşağıdaki sistem çağrıları kullanılır:

- open(): Kaynak ve hedef dosyayı okumak için açmak.
- read(): Kaynak dosyadan veri okumak için kullanılır.
- write(): Hedef dosyaya veri yazmak için kullanılır.
- close(): Dosyaların kapatılması ve işlem sonrası kaynakların serbest bırakılması.

Soru 14: Bir bilgisayara kaç adet işletim sistemi kurulabilir? MBR ne işe yarar?

MBR, sabit disk üzerinde bulunan ve bilgisayarın önyükleme sürecini başlatan bölümüdür. MBR, bilgisayar açıldığında işletim sistemi veya önyükleme yöneticisi (bootloader) gibi önyükleme yazılımını bulundurur. Bu kod, bilgisayar açıldığında ilk olarak yüklenen ve çalışan kod parçasıdır. MBR, sabit disk üzerinde bulunan bölüm (partition) tablosunu içerir. Bu tablo, disk üzerindeki bölümlerin başlangıç ve bitiş konumlarını, boyutlarını ve dosya sistemlerini tanımlar. MBR, önyükleme kodunu yükledikten sonra, önyükleme yöneticisine (bootloader) veya doğrudan işletim sistemine yönlendirme yapar. Önyükleme yöneticisi, kullanıcının seçtiği işletim sistemi veya işletim sistemleri arasında geçiş yapılmasını sağlar.

Bir bilgisayara kaç adet işletim sistemi kurulabileceği, bilgisayarın disk alanı, işletim sistemi desteği ve disk bölümlendirme yöntemine bağlı olarak değişiklik gösterebilir. Modern bir bilgisayara tek disk üzerinde 4 adet işletim sistemi kurulabilir.

Soru 15: Bitişik yer (contiguous) tahsisinin avantaj ve dezavantajları nelerdir?

Bitişik yer tahsis yöntemi, dosya sistemlerinde dosyaların disk üzerinde ardışık veya bitişik bloklara yerleştirilmesi anlamına gelir. Bitişik yer tahsisinde dosya blokları diskte ardışık olarak yerleştirildiği için dosyanın tamamına hızlı erişim sağlanabilir. Disk arama hareketi (disk seek) minimum düzeyde tutulur, bu da dosya erişim hızını artırır. Dosya sistemleri için yönetimi kolaylaştırır. Dosyanın blokları arasında bağlantı yoktur ve dosyanın diskteki konumu doğrudan hesaplanabilir. Bitişik yer tahsisinde dosyaların blokları ardışık olduğu için disk parçalanması daha az olabilir. Dosya sistemi, yeni dosyaları ve büyütülen dosyaları diskteki boş ardışık alanlara yerleştirebilir. Disk üzerinde büyük, ardışık boş alanları bulmak zor olabilir. Bu durum, dosya sistemlerinin boş alan yönetimini zorlaştırabilir ve boş alanın parçalanmasına neden olabilir. Dosya büyüdükçe, bitişik yer tahsisinde dosya bloklarının art arda konulması gerektiği için dosyanın büyüme potansiyeli sınırlı olabilir. Dosyaların silinmesi veya boyutlarının değiştirilmesi sonucunda diskte boş kalan ardışık alanlar parçalanabilir.

Soru 16: Bağlı liste (linked list) kullanılarak tahsisin avantaj ve dezavantajları nelerdir?

Bağlı liste (linked list) kullanılarak yapılan tahsis, disk bloklarının dosya sisteminde bir zincir şeklinde birbirine bağlanmasıdır. Her blok, bir sonraki bloğun adresini içerir. Dosya bloklarının disk üzerinde ardışık olmasına gerek yoktur. Bu nedenle, disk üzerinde parçalanmış boş alanlar daha verimli kullanılabilir. Dosyanın büyümesi gerektiğinde, mevcut boş bloklardan herhangi biri kullanılabilir. Dosyanın bitişik olması gerekmeyen yapısı, dosya genişletme işlemini kolaylaştırır. Bağlı liste yöntemi, disk üzerinde küçük,

dağınık boş alanları kullanarak daha verimli bir boş alan yönetimi sağlar. Yönetim yapıları ve algoritmaları daha basittir, bu nedenle dosya sistemi uygulamaları daha kolay olabilir. Her dosya bloğu, bir sonraki bloğun adresini içerir ve bu da dosyanın tüm bloklarına erişmek için her bir bloğu sırasıyla okuma gerektirir. Büyük dosyalar için bu durum erişim hızını yavaşlatabilir. Bloklar disk üzerinde dağınık yerleştirildiğinde, disk okuma kafası sürekli hareket etmek zorunda kalır, bu da erişim süresini artırır. Bir blokta meydana gelen herhangi bir bozulma veya bağlantı hatası, tüm dosyanın erişilemez hale gelmesine neden olabilir. Bu, özellikle dosya sisteminde hata tolerans mekanizmaları yoksa büyük bir risk oluşturur.

Soru 17: Güç kesintilerine karşı dosya sistemini korumanın yöntemleri neler olabilir?

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 18: DMA nedir? Hangi durumlarda kullanmak avantajlıdır ya da değildir?

DMA (Direct Memory Access), bilgisayar sistemlerinde veri transferlerini hızlandırmak ve CPU yükünü azaltmak için kullanılan bir teknolojidir. DMA, verilerin bellek (RAM) ve G/Ç aygıtları (diskler, ağ kartları, vb.) arasında doğrudan transferini sağlar.

CPU, DMA denetleyicisine bir transfer işlemi başlatması için komut verir. Bu komut, kaynak ve hedef adresler ile transfer edilecek veri miktarını içerir. DMA denetleyicisi, veri transferini doğrudan bellek ve G/Ç aygıtı arasında gerçekleştirir. Transfer işlemi tamamlandığında, DMA denetleyicisi CPU'ya bir kesme (interrupt) göndererek işlemin tamamlandığını bildirir.

Disk okuma/yazma, ağ veri transferleri ve grafik işlemleri gibi büyük veri bloklarının hızlı bir şekilde aktarılması gereken durumlarda avantajlıdır. Küçük veri transferleri ve düşük bant genişliği gerektiren işlemlerde. CPU tarafından yapılan veri transferleri daha verimli olabilir.

Soru 19: RAID ne işe yarar? RAID konfigürasyonları neler?

RAID (Redundant Array of Independent Disks), birden fazla fiziksel disk sürücüsünü birleştirerek veri depolama güvenilirliğini, performansını ve kapasitesini artırmak için kullanılır.

RAID 0 (Striping) Veriler, bloklar halinde iki veya daha fazla disk arasında bölünerek yazılır. Her disk, veri bloğunun bir kısmını içerir.

RAID 1 (Mirroring) Veriler, iki veya daha fazla diske aynı anda kopyalanır (aynalanır). Her disk, tam veri kopyasını içerir.

RAID 5 (Striping with Parity) Veriler, üç veya daha fazla disk arasında bölünerek yazılır. Parite bilgisi, verilerin yeniden oluşturulabilmesi için disklere dağıtılır.

RAID 6 (Striping with Double Parity) RAID 5'e benzer, ancak parite bilgisi iki farklı diske yazılır. Bu, iki diskin aynı anda arızalanmasına karşı koruma sağlar.

RAID 10 (RAID 1+0 veya RAID 0+1) RAID 1 ve RAID 0 kombinasyonudur. Veriler önce aynalanır (RAID 1) ve sonra bu aynalanmış veriler şeritlenir (RAID 0).

Soru 20: Diskte bir dosya değişik sektörlerde bulunabilir. Döner diske sahip bir diskte okuma işlemini kısa sürede bitirebilmek için neler yapılabilir?

Döner diske (hard disk drive - HDD) sahip bir sistemde okuma işlemlerini kısa sürede bitirebilmek için çeşitli stratejiler ve teknikler kullanılabilir. Bu teknikler, diskin mekanik hareketlerini optimize etmeyi ve veri erişim süresini minimize etmeyi hedefler.

- FCFS (First-Come, First-Served): İstekleri alınma sırasına göre işler. Basit ama optimal değil.
- SSTF (Shortest Seek Time First): En kısa arama süresine sahip isteği önce işler. Performansı iyileştirir ama açlık (starvation) riski vardır.
- SCAN (Elevator Algorithm): Disk kafası bir yönde hareket ederken sırayla istekleri işler ve uç noktaya ulaştığında yön değiştirir.
- C-SCAN (Circular SCAN): SCAN algoritmasının bir varyasyonudur. Disk kafası bir uçtan diğer uca gidip tekrar başa döner ve aynı yönde hareket etmeye devam eder.
- LOOK ve C-LOOK: SCAN ve C-SCAN'in bir varyasyonu olup, uç noktaya ulaşmadan son isteğe kadar hareket eder ve yön değiştirir.