



Bölüm 17: Soru Cevap

İşletim Sistemleri



Soru

- Sanal belleğin temel amacı nedir?
 - A) Fiziksel belleğin boyutunu artırmak
 - B) Bellek paylaşımı için bir mekanizma sağlamak
 - C) Süreçlerin fiziksel olarak mevcut olan bellekten daha fazla belleğe erişmesine izin vermek
 - D) Bellek erişim süresini hızlandırmak



Cevap

- Cevap: C
- Sanal bellek, fiziksel belleđi desteklemek ve genişletmek için kullanılan bir mekanizmadır. Bir süreç, ihtiyaç duyduđu verilere fiziksel bellekte erişemiyorsa, işletim sistemi bu verileri sanal bellekten getirir. Bu sayede, süreç daha fazla belleđe erişebilir gibi görünür. Bu, işletim sisteminin fiziksel belleđi verimli bir şekilde yönetmesine ve süreçlerin gereksinimlerini karşılamasına olanak tanır.



Soru

- Fiziksel belleđi sabit boyutlu bloklara bölen ve her blođu gerektiđinde bir sürece atayan bellek yönetimi tekniđi hangisidir?
- A) Sayfalama (paging)
- B) Kesimleme (segmentation)
- C) Takas (swapping)
- D) Parçalanma (fragmentation)



Cevap

- Cevap: A
- Sayfalama, fiziksel belleđi sabit boyutlu bloklara böler ve her blođu bir sürece atar. Bu teknikte, süreçler sabit boyutlu sayfalara bölünür ve her sayfa fiziksel bellek bloklarına atılır. Sayfalama, bellek yönetimi için yaygın olarak kullanılan bir tekniktir ve süreçlere gerektiđi kadar bellek sağlamak için esneklik sunar. Bu sayede, bellek kullanımı optimize edilir ve süreçler arasında bellek paylaşımı daha etkin hale gelir.



Soru

- Bitişik bellek tahsisi kullanmanın başlıca dezavantajı nedir?
 - A) Harici parçalanmaya neden olur
 - B) Karmaşık donanım desteği gerektirir
 - C) Eşzamanlı olarak çalışabilecek süreç sayısını sınırlar
 - D) Sanal belleği destekleyemez



Cevap

- Cevap: A
- Bitişik bellek tahsisi, belleği kesintisiz bloklara ayırır ve her bloğu bir sürece atar. Bu yöntemde, süreçler belleğe yerleştirildiğinde, bellekteki boşluklar parçalanabilir ve kullanılamaz hale gelebilir. Bu duruma "harici parçalanma" denir. Harici parçalanma, bellek kullanımını kısıtlar ve bellek verimliliğini azaltır. Ayrıca, bellek bloklarının arasında boşluklar olduğu için, büyük süreçleri belleğe yerleştirmek zorlaşabilir.



Soru

- Hangi bellek tahsis algoritması, bir isteđi karřılamak için yeterince büyük olan en küçük bellek blođunu ayırarak israfı en aza indirmeyi amaçlar?
- A) En İyi Uygun
- B) İlk Uygun
- C) En Kötü Uygun
- D) Sonraki Uygun



Cevap

- Cevap: A
- En İyi Uygun (Best Fit) bellek tahsis algoritması, bir bellek isteğini karşılamak için mevcut olan en küçük bellek bloğunu bulur. Bu sayede, bellek israfı minimize edilir ve boşa harcanan bellek miktarı azalır. En İyi Uygun algoritması, mevcut boş bellek bloklarını taramak suretiyle çalışır ve istek için en küçük uygun bloğu seçer. Bu yöntemde, bellek boşa harcanmadan etkili bir şekilde kullanılır. Ancak, algoritma maliyetli olabilir çünkü mevcut tüm blokların taranması gerekir.



Soru

- Sanal bellek yönetiminde sayfa tablosunun amacı nedir?
- A) Sanal adresleri fiziksel adreslere eşlemek
- B) Tüm sayfaların içeriğini ana bellekte saklamak
- C) Sayfaların süreçlere tahsisini yönetmek
- D) Bellek sayfaları için erişim izinlerini kontrol etmek



Cevap

- Cevap: A
- Sayfa tablosu, sanal bellek yönetiminde kullanılan kritik bir veri yapısıdır. Sanal bellek kullanıldığında, sürecin sanal adresleri fiziksel bellek adreslerine eşlenmelidir. Bu eşleme işlemi sayfa tablosu tarafından gerçekleştirilir. Sayfa tablosu, her bir sanal bellek sayfasının karşılık geldiği fiziksel bellek bloğunu gösterir. Bir süreç, bir sanal bellek adresine erişmek istediğinde, sayfa tablosu bu adresin fiziksel bellek adresine karşılık geldiği yerin bilgisini sağlar.



Soru

- Tüm mevcut süreçleri tutmak için fiziksel belleğin yeterli olmadığı durumda hangi mekanizma kullanılır?
- A) Boşa çabalama (thrashing)
- B) Sayfa değiştirme (page replacement)
- C) Takas etme (swapping)
- D) Parçalanma (fragmentation)



Cevap

- Cevap: C
- Takas etme (Swapping), fiziksel belleğin yetersiz olduğu durumlarda kullanılır. Eğer işletim sistemi tüm çalışan süreçleri tutmak için yeterli fiziksel belleğe sahip değilse, kullanılmayan süreçleri veya süreçlerin bazı bölümlerini, sabit disk gibi daha yavaş bir depolama alanına taşıyarak belleği boşaltabilir. Bu sayede, önemli süreçler için yer açılır ve sistem verimliliği artırılır. Takas etme, fiziksel belleğin sınırlarını aşan durumlarda sistemin dengesini korumak için önemlidir.



Soru

- Bellek yönetiminde TLB'nin (Dönüşüm Önbellesi) rolü nedir?
 - A) Sık erişilen sayfaların kopyalarını saklamak
 - B) Adres dönüşümünü hızlandırmak için sayfa tablosu girdilerini önbellege almak
 - C) Sayfaların süreçlere tahsisini yönetmek
 - D) Bellek sayfaları için erişim izinlerini kontrol etmek



Cevap

- Cevap: B
- TLB, sanal bellek adreslerini fiziksel bellek adreslerine çevirirken adres dönüşümünü hızlandırmak için kullanılan bir önbellek türüdür. Sayfa tablosundaki girdilerin bir kopyasını saklar ve sıkça kullanılan çevirileri hızlı bir şekilde gerçekleştirmek için bu kopyaları kullanır. Bu, bellek erişim sürelerini azaltır ve sistem performansını artırır.



Soru

- Birden fazla sürecin bellekte bir programın tek bir kopyasını paylaşmasına izin veren bellek yönetimi tekniği hangisidir?
- A) Sayfalama
- B) Kesimleme (segmentation)
- C) Talep Üzerine Sayfalama (demand paging)
- D) Paylaşımlı Bellek



Cevap

- Cevap: D
- Paylaşımlı bellek, farklı süreçler arasında veri ve bilgi paylaşımını kolaylaştıran bir yöntemdir. Bu tekniğin kullanılması, bellek kullanımını optimize eder ve kaynak kullanımını azaltır.



Soru

- Sanal bellek sistemlerinde talep üzerine sayfalamanın kullanılmasının dezavantajı nedir?
 - A) Daha büyük bir sayfa tablosu gerektirir
 - B) Disk G/Ç yükünü aşırı artırır
 - C) Daha yüksek bellek parçalanmasına neden olur
 - D) Çoklu programlamayı desteklemez



Cevap

- Cevap: B
- Talep üzerine sayfalama, bir sürecin çalışma zamanında sadece ihtiyaç duyduğu sayfaların belleğe getirilmesini sağlayarak bellek kullanımını optimize eder. Ancak, bu yöntemin dezavantajı, disk G/Ç (giriş/çıkış) işlemlerinde artışa neden olmasıdır. Çünkü ihtiyaç duyulan sayfalar fiziksel bellekte bulunmadığında, diskten bu sayfaların alınması gerekmektedir.



Soru

- Bellek Yönetim Birimi (MMU) nin rolü aşağıdakilerden hangisi DEĞİLDİR?
 - A) Adres çeviri
 - B) Bellek koruma
 - C) Süreç çizelgeleme
 - D) Sanal bellek yönetimi



Cevap

- Cevap: C
- Süreç çizelgeleme MMU'nun görevlerinden biri değildir. MMU, adres çevirisi, bellek koruması ve sanal bellek yönetimi gibi işlevleri gerçekleştirir. Süreç çizelgeleme, işletim sistemi çekirdeğinin sorumluluğundadır.



Soru

- Aşağıdakilerden hangisi bellek yönetiminde Thrashing'in bir özelliğidir?
 - A) Yüksek disk G/Ç ve düşük CPU kullanımı ile aşırı takas işlemi
 - B) Minimal iş yükü ile bellek kaynaklarının etkin kullanımı
 - C) İkincil depodan ana belleğe hızlı veri transferi
 - D) Parçalanmayı en aza indirerek süreçlere bellek tahsisi



Cevap

- Cevap: A
- Thrashing, aşırı takas işlemlerine neden olur. Bu durumda, işletim sistemi sürekli olarak işlemciye yararlı işler yapmak yerine bellek ve disk arasında sürekli veri alışverişi yapar. Sonuç olarak, yüksek disk G/Ç (giriş/çıkış) işlemleri gerçekleşirken, işlemci düşük bir kullanım oranına sahip olur. Bu nedenle, thrashing durumu genellikle bellek yetersizliğiyle ilgilidir ve işletim sistemi birçok süreci aynı anda sürdürmek için yetersiz bellek alanı nedeniyle verimliliği azaltır.



Soru

- Talep üzerine sayfalama sanal bellek sistemlerinde, hangi sayfa deęiřtirme algoritması, en uzun süre başvurulmayacak olan sayfayı deęiřtirmeyi amalar?
- A) En Düşük En Son Kullanılan (LRU)
- B) İlk Giren İlk Çıkar (FIFO)
- C) Saat Deęiřtirme (Clock)
- D) Optimal Sayfa Deęiřtirme (Optimal)



Cevap

- Cevap: D
- Optimal Sayfa Deęiřtirme algoritması, bir sayfanın gelecekte belirli bir süre boyunca başvurulmayacaęını tahmin ederek, en uygun sayfayı deęiřtirmeyi amaçlar. Bu algoritmanın amacı, gelecekte kullanılmayacak olan sayfaları tahmin ederek, bellek içindeki sayfaları en verimli şekilde kullanmaktır. Ancak, pratikte gerçekteştirilmesi zordur çünkü gelecekteki istekleri önceden tahmin etmek zordur. Bu nedenle, algoritma teorik bir kavram olarak kalır. Dięer sayfa deęiřtirme algoritmaları, gelecekteki istekleri tahmin etmek yerine farklı stratejiler kullanır.



Soru

- Bellek yönetiminde Çalışma Kümesi Modelinin temel amacı nedir?
 - A) Süreçlerde bellek sızıntılarını önlemek
 - B) Süreçlerin gelecekteki bellek ihtiyaçlarını tahmin etmek
 - C) Thrashing'i önlemek için gereken minimum sayfa sayısını belirlemek
 - D) Bir sürecin etkin çalışması için gereken bellek miktarını ölçmek



Cevap

- Cevap: D
- Çalışma Kümesi Modeli, bir sürecin etkin bir şekilde çalışabilmesi için gereken bellek miktarını ölçmek amacıyla kullanılır. Bu model, bir sürecin çalışması sırasında bellekte hangi sayfaların kullanıldığını ve süreç için ne kadar belleğe ihtiyaç duyulduğunu belirler. İşletim sistemi, sürecin çalışma zamanında kullanılan bellek miktarını izler ve bu bilgiye dayanarak süreç için gereken bellek miktarını belirler.



Soru

- Aşağıdakilerden hangisi sanal bellek sistemlerinde sayfa tablosu girdisi DEĞİLDİR?
 - A) Geçerli/Geçersiz Bit
 - B) Kirli Bit
 - C) Sayfa Tablosu Taban Yazmacı
 - D) Çerçeve Numarası



Cevap

- Cevap: C
- Sayfa tablosu girdisinin bileşenleri, bir sayfanın fiziksel bellekteki konumu hakkında bilgi sağlar. Bu bileşenler arasında Geçerli/Geçersiz Bit (Valid/Invalid Bit), Kirli Bit (Dirty Bit) ve Çerçeve Numarası (Frame Number) bulunur. Ancak, Sayfa Tablosu Taban Kaydedici (Page Table Base Register), sayfa tablosunun yerini gösteren bir yazmaçtır. Sayfa tablosu taban kaydedici, sayfa tablosunun bellekteki başlangıç adresini tutar ve işletim sistemi tarafından kullanılır.



Soru

- Bellek tahsisinde Buddy sisteminin amacı nedir?
 - A) Bitişik boş bellek bloklarını birleştirerek parçalanmayı yönetmek
 - B) Tüm süreçleri bellek içine ve dışına takas ederek talep üzerine sayfalama uygulamak
 - C) Süreç büyüklüğüne bağlı olarak bellek kaynaklarını dinamik olarak tahsis etmek
 - D) Sıkça erişilen sayfaları belleğe önceden yükleyerek sayfa hatalarını önlemek



Cevap

- Cevap: A
- Buddy sistemi, bitişik boş bellek bloklarını birleştirerek parçalanmayı yönetir. Bu sistemde, bellek blokları "arkadaşlar" olarak belirli boyutlarda gruplara ayrılır. Bir süreç için istenen bellek miktarı belirlendiğinde, uygun boyuttaki bir blok tahsis edilir. Eğer istenen boyutta bir blok yoksa, daha büyük bir blok, gereksinim duyulan boyuta bölünerek kullanılır veya daha küçük bloklar birleştirilerek ihtiyacı karşılayacak bir blok oluşturulur. Bu şekilde, parçalanma azaltılır ve bellek kullanımı optimize edilir.



Soru

- Hangi bellek tahsis algoritması belleği deęişken boyutlu bölümlere ayırır ve her bölümü sürecin bellek gereksinimlerine göre atar?
- A) Kesimleme (segmentation)
- B) Sayfalama (paging)
- C) En Uygun (best fit)
- D) Sonraki Uygun (next fit)



Cevap

- Cevap: A
- Kesimleme, belleđi deđişken boyutlu bölümlere ayırır ve her bir bölümü sürecin bellek gereksinimlerine göre tahsis eder. Bu yöntemde, her süreç, ihtiyaç duyduđu bellek miktarına uygun olan bir bölüm alır. Bu, bellek kullanımını optimize etmek için esneklik sağlar ve farklı süreçlerin farklı bellek ihtiyaçlarına yanıt verir.



Soru

- İşletim sistemlerinde bellek korumasının amacı nedir?
 - A) Sistem kaynaklarına izinsiz erişimi önlemek
 - B) Belleği birden çok süreç arasında verimli bir şekilde tahsis etmek
 - C) Bellek tahsis işlemi sırasında parçalanmayı en aza indirmek
 - D) Disk G/Ç işlemlerinin performansını artırmak



Cevap

- Cevap: A
- Bellek koruması, işletim sisteminin sistem kaynaklarına (bellek gibi) yetkisiz erişimi önlemesini sağlar. Bu, işletim sisteminin, herhangi bir sürecin diğer süreçlerin bellek alanlarına müdahalesini önleyerek güvenliği sağlamasına yardımcı olur. Bellek koruması, yetkilendirilmiş erişim ve izinsiz erişim arasında bir ayırım yaparak, sistemdeki verilerin bütünlüğünü ve gizliliğini korur.



Soru

- Aşağıdakilerden hangisi Kesimleme kullanmanın faydalarından değildir?
- A) Birden çok süreç arasında kod kesimlerinin paylaşımına olanak tanır
- B) Belleği sabit boyutlu bloklara bölerek bellek tahsisini basitleştirir
- C) Dinamik yükleme ve kütüphane paylaşımını destekler
- D) Farklı ayrıcalıklar atayarak bellek koruması sağlar



Cevap

- Cevap: B
- Kesimleme kullanmanın faydalarından biri belleği sabit boyutlu bloklara bölmek değildir, çünkü bu Segmentation tekniğine özgü değildir. Kesimleme tekniği belleği mantıksal olarak ilgili parçalara böler ve her parçayı bir sürece atar. Bu sayede, birden çok süreç aynı bellek alanını paylaşabilir, dinamik yükleme ve kütüphane paylaşımını destekler, ve farklı ayrıcalıklar atayarak bellek koruması sağlar.



Soru

- Çok seviyeli sayfa tablosu düzeninde, birden fazla sayfa tablosu kullanmanın amacı nedir?
- A) Sayfa tablosunun boyutunu azaltmak ve bellek gereksinimini en aza indirmek
- B) Sık erişilen sayfaları önbelleğe alarak adres çevirisi performansını artırmak
- C) Hiyerarşik bir yapı ile daha büyük sanal adres alanlarını desteklemek
- D) Bellek sıkıştırma ve defragmentation işlemlerini mümkün kılmak



Cevap

- Cevap: C
- Çok seviyeli sayfa tablosu düzeni, daha büyük sanal adres alanlarını desteklemek için kullanılır. Bu düzen, sanal adres alanını hiyerarşik bir yapıda düzenler ve daha büyük adres alanlarını yönetmeyi kolaylaştırır. Bu sayede, çok seviyeli sayfa tablosu düzeni, bellek yönetimini optimize eder ve daha büyük ve karmaşık sanal adres alanlarını destekler.



Soru

- Diğer bellek tahsis tekniklerine göre Buddy sistemin avantajı nedir?
- A) Sabit boyutta bloklar halinde bellek tahsis ederek parçalanmayı en aza indirir
- B) Etkin bellek sıkıştırma ve birleştirme (defragmentation) işlemleri sağlar
- C) Sık erişilen sayfaları önbelleğe alarak belleğe hızlı erişim sağlar
- D) Bellek tahsisini ve serbest bırakma işlemlerini düşük iş yükü ile basitleştirir



Cevap

- Cevap: D
- Buddy sistem, bellek tahsisini yönetmek için kullanılan bir yöntemdir. Bu yöntemde, bellek blokları "arkadaş" olarak adlandırılan ve boyutları birbirine yakın olan bloklar halinde gruplandırılır. Bir süreç bellek tahsis etmek istediğinde, sistem bu süreç için uygun boyutta bir blok bulur veya uygun boyutta bir bloğu böler. Bellek serbest bırakma işlemi ise, artık kullanılmayan bir bloğun işaretlenmesi ve gerektiğinde bu bloğun birleştirilmesiyle gerçekleştirilir. Bu şekilde bellek tahsisini ve serbest bırakma işlemlerini basit ve etkili bir şekilde yönetir.



Soru

- Çok seviyeli sayfa tablosu düzeninde, TLB kullanım amacı nedir?
 - A) Daha hızlı erişim için sıkça erişilen sayfaların kopyalarını saklamak
 - B) Sayfa tablosunun boyutunu azaltmak ve bellek gereksinimini en aza indirmek
 - C) Ana bellek ve ikincil depo arasında sayfa değişimi yaparak talep üzerine sayfalama desteği sağlamak
 - D) Sayfalara farklı erişim izinleri atayarak bellek koruması sağlamak



Cevap

- Cevap: A
- TLB'nin temel amacı, sıkça erişilen sayfaların kopyalarını saklayarak daha hızlı veri transferi sağlamaktır. TLB, sanal bellek sistemlerinde adres dönüşümünü hızlandırmak için kullanılan bir önbellek türüdür. TLB, sıkça erişilen sayfa tablosu girdilerinin bir kopyasını saklar. Bu sayede, bir işlem sanal bellek adresini fiziksel bellek adresine çevirirken, bu çevirme işlemini hızlı bir şekilde gerçekleştirebilir. Eğer TLB'de gerekli bilgi bulunursa, işlem fiziksel bellek adresini TLB'den alır ve doğrudan bu adrese erişir, böylece sayfa tablosuna erişim süresi azalır ve sistem performansı artar.



Soru

- Aşağıdakilerden hangisi bellek tahsisinde dış parçalanmaya neden olur?
- A) Ardışık bellek tahsis düzeninde bloklarının tahsisi ve serbest bırakılması
- B) Talep üzerine sayfalamada bellek sayfalarının süreçlere tahsisi
- C) Kesimleme tabanlı bellek yönetiminde bellek kesimlerinin birden fazla süreç arasında paylaşılması
- D) Buddy sistemi tahsis stratejisinde değişken boyutlu bellek bölümlerinin süreçlere tahsisi



Cevap

- Cevap: A
- Dış parçalanmaya sebep olan en muhtemel senaryo, ardışık bellek tahsis düzeninde bellek bloklarının tahsis edilmesi ve serbest bırakılmasıdır. Ardışık bellek tahsis düzeninde, bellek blokları ardışık olarak tahsis edilir ve süreçler bellek blokları arasında yer değiştirir. Ancak, bellek blokları serbest bırakıldığında, bu bloklar arasında boşluklar oluşabilir. Bu boşluklar, yeterince büyük bir süreç için yeterli bellek bloğu olmaması durumunda parçalanmaya neden olabilir. Yani, serbest bırakılan bellek blokları arasında kalan küçük boş alanlar, dış parçalanmaya yol açabilir.



Soru

- Bellek yönetiminde Takas (Swapping) uygulamanın asıl amacı nedir?
 - A) Sanal bellek sistemlerinde sayfa tablosu bakımının iş yükünü azaltmak
 - B) Süreçlerin birbirleriyle bellek kesimlerini paylaşmasını sağlamak için bir mekanizma sağlamak
 - C) Bellek alanı serbest bırakmak için ana bellek ve ikincil depo arasında tüm süreci transfer etmek
 - D) Sıkça erişilen sayfaları ayrı bir önbellekte önbelleğe alarak bellek erişim süresini iyileştirmek



Cevap

- Cevap: C
- Takas (Swapping), tüm sürecin ana bellek ve ikincil depo (genellikle disk) arasında aktarılmasıyla bellek alanı serbest bırakmak için kullanılır. Bir bilgisayarın ana belleği sınırlıdır ve aynı anda birçok süreç çalıştırıldığında, bellek yetersizliği yaşanabilir. Bu durumda, aktif olmayan veya şu anda çalıştırılmayan süreçlerin bellekten çıkarılması ve ikincil depoya (disk) taşınması gerekir. Bu işlem, belleği serbest bırakır ve mevcut süreçlerin bellek ihtiyaçlarını karşılamak için yer açar.



Soru

- Aşağıdaki bellek tahsis algoritmalarından hangisi, bir isteği karşılamak için yeterince büyük olan en büyük bellek bloğunu tahsis ederek boşa harcanan belleği en aza indirmeyi amaçlar?
 - A) En Uygun (best fit)
 - B) İlk Uygun (first fit)
 - C) En Kötü Uygun (worst fit)
 - D) Sonraki Uygun (next fit)



Cevap

- Cevap: C
- En Kötü Uygun (Worst Fit) bellek tahsis algoritması, bir süreç için yeterince büyük olan en büyük bellek bloğunu tahsis ederek boşa harcanan belleği en aza indirmeyi amaçlar. Bu algoritma, mevcut boş bellek blokları arasından en büyük olanını seçer ve bu bloğu sürecin bellek ihtiyacını karşılamak için kullanır. Bu şekilde, küçük boşluklar bırakılmaz ve bellek alanı daha etkin bir şekilde kullanılır. Ancak, bu algoritmanın dezavantajı, küçük boşluklar yerine büyük boşluklar bırakmasıdır, bu da gelecekteki tahsisler için uygun boşlukların azalmasına ve parçalanmanın artmasına neden olabilir.



Soru

- Talep üzerine sayfalamada, basitliđi ve düşük iř y¼k¼ ile bilinen ancak Belady anomalisinden etkilenebilecek hangi algoritmadır?
- A) En Son Kullanılan (LRU)
- B) İlk Giren İlk Çıkar (FIFO)
- C) Optimal Sayfa Yer Deđiřtirme
- D) Saat Yer Deđiřtirme



Cevap

- Cevap: B
- İlk Giren İlk Çıkar (FIFO) sayfa yer değiştirme algoritması, basitliği ve düşük iş yükü ile bilinir ancak Belady Anomalisinden etkilenebilir. FIFO algoritması, bellekte en uzun süredir bulunan sayfanın yerine yeni bir sayfa getirir. Bu, basit bir kuyruk yapısı kullanılarak gerçekleştirilir; yeni gelen sayfalar kuyruğun sonuna eklenir ve yer değiştirme gerektiğinde kuyruğun başındaki sayfa bellekten çıkarılır. Belady Anomali'si, FIFO'nun bazen yeni sayfa getirmesine rağmen sayfa yer değiştirme sayısının artabileceği durumu ifade eder. Yani, FIFO, daha fazla yer değiştirme yaparak performansı olumsuz yönde etkileyebilir.



Soru

- İşletim sistemlerinde bellek yeniden yerleştirme (relocation) işleminin temel amacı nedir?
 - A) Fiziksel belleğin boyutunu artırmak
 - B) Süreçlerin fiziksel adreslerinden bağımsız olarak bellek konumlarına erişmesine izin vermek
 - C) Bellek erişim süresini hızlandırmak
 - D) Bellek parçalanmasını yönetmek



Cevap

- Cevap: B
- Bellek yeniden yerleştirme, işletim sistemlerinde süreçlerin fiziksel adreslerinden bağımsız olarak bellek konumlarına erişmesini sağlar. Bir süreç, çalıştırıldığında fiziksel bellek içinde belirli bir konuma yerleştirilir. Ancak, fiziksel bellek, sınırlı bir kaynaktır ve süreçler arasında paylaşılmalıdır. Ayrıca, bir süreç için ayrılan fiziksel bellek miktarı, sürecin ihtiyaçlarını karşılayabilecek kadar büyük olmayabilir. Bu nedenle, işletim sistemi, süreçleri fiziksel bellekte daha uygun bir konuma taşımak ve bu süreçlere sanal bellek alanları tahsis etmek için bellek yeniden yerleştirme işlemini kullanır. Bu şekilde, süreçler fiziksel bellekte farklı konumlara yerleştirilebilir ve farklı zamanlarda farklı fiziksel bellek konumlarını kullanabilir.



Soru

- Aşağıdakilerden hangisi mantıksal adresleri fiziksel adreslere çevirmek için Bellek Yönetimi birimi (MMU) kullanımını gerektirir?
- A) Ardışık Bellek Tahsisi
- B) Sayfalama
- C) Kesimleme
- D) Parçalanma



Cevap

- Cevap: B
- Sayfalama, sanal bellek sistemlerinde kullanılan bir bellek yönetimi tekniğidir. Bu teknikte, bellek fiziksel olarak bölümlere ayrılmaz; bunun yerine, bellek sanal sayfalara bölünür. MMU, işlemci tarafından oluşturulan mantıksal adresleri, bu sanal sayfalardaki fiziksel adreslere çevirir. Bu işlem, sürecin sanal bellek alanında yer aldığını varsayar ve herhangi bir sayfanın herhangi bir zamanda ana bellekte herhangi bir konuma taşınabileceği anlamına gelir. MMU, işletim sistemi tarafından belirlenen sayfa tablolarını kullanarak bu adres dönüşümünü gerçekleştirir.



Soru

- Talep üzerine sayfalamada, Bellek Yönetimi birimi (MMU) ne işe yarar?
- A) Bellek sayfalarının süreçlere tahsisini yönetir
- B) CPU tarafından oluşturulan mantıksal adresleri ana bellekteki fiziksel adreslere çevirir
- C) Bellek sayfaları için erişim izinlerini kontrol eder
- D) Daha hızlı veri transferi için sıkça erişilen sayfaları önbelleğe alır



Cevap

- Cevap: B
- MMU, sanal bellek sistemlerinde mantıksal adreslerin fiziksel bellek adreslerine çevrilmesinden sorumludur. İşlemci (CPU), programların yürütülmesi sırasında mantıksal adresler üretir. Bu mantıksal adresler, işletim sistemi tarafından belirlenen sayfa tabloları kullanılarak fiziksel bellek adreslerine çevrilir. MMU, bu adres dönüşüm işlemini gerçekleştirir ve işlemcinin erişmek istediği fiziksel bellek adresini belirler. Bu sayede, işlemci, sanal bellek alanında bulunan bir süreç için fiziksel bellek içindeki doğru konumu bulabilir ve süreçler birbirinden izole edilmiş bir şekilde çalışabilir.



Soru

- Bellek yönetiminde yer deęiřtirme yazmaçlarının (relocation register) temel faydası nedir?
- A) Süreçlerin çeviri yapmadan bellek konumlarına erişmesine izin verir
- B) Belleęi sabit boyutta bloklara ayırarak bellek parçalanmasını önler
- C) Bellekte dinamik yükleme ve kütüphanelerin paylaşımını kolaylaştırır
- D) Süreçlerin fiziksel bellek konumlarından baęımsız yürütölmesini sağlar



Cevap

- Cevap: D
- Yer deęiřtirme yazmaçları, süreçlerin bellek içindeki konumlarını izlemek ve gerekirse yeniden yerleřtirmek için kullanılan özel yazmaçlardır. Bu yazmaçlar, süreçlerin fiziksel bellek içindeki konumlarını belirtir ve süreçlerin baęımsız olarak yürütölmesini saęlar. Bu sayede, süreçlerin bellek içindeki konumları deęiřse bile, süreçler çalıřmaya devam edebilir. Bu, iřletim sisteminin bellek yönetimini daha esnek hale getirir.



Soru

- CPU tarafından oluşturulan mantıksal adresleri taban ve limit yazmaçlarını kullanarak fiziksel adreslere çeviren hangi bellek yönetimi tekniğidir?
- A) Sayfalama
- B) Kesimleme
- C) Takas
- D) Ardışık Bellek Tahsisi



Cevap

- Cevap: D
- Ardışık Bellek Tahsisi, CPU tarafından oluşturulan mantıksal adresleri taban ve limit yazmaçlarını kullanarak fiziksel adreslere çeviren bir bellek yönetimi tekniğidir. Belleği ardışık bloklara böler ve her süreç için bir başlangıç adresi ve bir limit belirler. CPU tarafından oluşturulan mantıksal adresler, süreç için atanmış olan başlangıç adresi ve limit kullanılarak doğrudan fiziksel bellek adreslerine çevrilir. Bu, süreçlerin fiziksel bellek içinde belirli bir alanda sınırlı olmasını sağlar. Bu yöntem, basit ve hızlı bir bellek yönetimi tekniği olmasına rağmen, bazı sınırlamaları vardır. Örneğin, bellek parçalanmasını önlemez ve bellek kullanımını optimize etmek için dinamik olarak büyüüp küçülmeyen bloklar kullanır.



Soru

- Bellek yer deęiřtirmede taban ve limit yazmaçlarının kullanılma amacı nedir?
- A) Bellek sayfalarının süreçlere tahsisini yönetmek
- B) Belleęi sabit boyutta bloklara ayırarak bellek parçalanmasını önlemek
- C) Mantıksal adresleri fiziksel adreslere çevirerek bellek korumasını sağlamak
- D) Daha hızlı transfer için sık erişilen sayfaları önbelleęe almak



Cevap

- Cevap: C
- Taban ve limit yazmaçlarının temel amacı, mantıksal adresleri fiziksel adreslere çevirmek ve bellek koruması sağlamaktır. Mantıksal adresler, işlemcinin ürettiği adreslerdir. Ancak, bu adresler doğrudan bellek üzerinde işaret etmezler. Bu nedenle, işlemcinin ürettiği mantıksal adresler, gerçek bellek içindeki fiziksel adreslere çevrilmelidir. Bu dönüşüm işlemi, taban ve limit yazmaçları kullanılarak yapılır. Ayrıca, bellek koruması için de kullanılır. Limit yazmacı, sürecin bellek aralığının boyutunu belirtir ve bu sayede süreç, sınırları aşamaz. Bu, işlemcinin istemeden bellek bölgelerine müdahale etmesini önler ve bellek korumasını sağlar.



Soru

- Aşağıdakilerden hangisi bellek parçalanmasına yol açar?
- A) Ardışık bellek tahsisinde bellek bloklarının tahsisi ve geri alınması
- B) Talep üzerine sayfalamada süreçlere bellek sayfalarının tahsisi
- C) Kesimleme bellek yönetiminde bellek bölümlerinin birden fazla süreç arasında paylaşılması
- D) Buddy tahsis stratejisinde süreçlere değişken boyutlarda bellek bölümlerinin tahsisi



Cevap

- Cevap: A
- Bellek parçalanmasına en muhtemel senaryo, ardışık bellek tahsisinde bellek bloklarının tahsisi ve serbest bırakılmasıdır. Ardışık bellek tahsisinde, bellek blokları birbirini izler, bu nedenle bir blok serbest bırakıldığında, kullanılamaz hale gelecek kadar küçük parçalar kalabilir. Bu durumda, bellek parçalanması oluşur. Bellek parçalanması, birçok küçük parçacığın kullanılamaz hale gelmesi ve belleğin etkin bir şekilde kullanılamaması durumunu ifade eder.



Soru

- Bellek yönetiminde dinamik yer deęiřtirmenin (dynamic relocation) temel amacı nedir?
 - A) Fiziksel belleęin boyutunu artırmak
 - B) Süreçlerin fiziksel adreslerinden baęımsız olarak bellek konumlarına erişmesine izin vermek
 - C) Bellek erişim süresini hızlandırmak
 - D) Bellek parçalanmasını yönetmek



Cevap

- Cevap: B
- Dinamik yer deęiřtirme, bir sürecin fiziksel bellek adresini deęiřtirme işlemidir. Bu işlem, süreci farklı bir fiziksel bellek adresine taşımak ve sürecin bellek içindeki konumunu deęiřtirmek için yapılır. Bu sayede, sürecin fiziksel bellek içindeki konumu deęiřse bile, süreç doęru bir şekilde çalışabilir. Dinamik yer deęiřtirme, işletim sistemi tarafından bellek tahsisi ve geri alınması sırasında kullanılır. Süreçlerin bellek içindeki konumlarının dinamik olarak deęiřtirilmesine izin vererek, işletim sistemi bellek kaynaklarını daha verimli bir şekilde yönetir.



Soru

- Aşağıdaki tekniklerinden hangisi bellek adreslerini çalışma zamanında ayarlamak için dinamik yer değiştirme gerektirir?
- A) Sayfalama
- B) Kesimleme
- C) Ardışık Bellek Tahsisi
- D) Parçalanma



Cevap

- Cevap: B
- Kesimleme tekniđi, belleđi mantıksal olarak farklı bölümlere böler ve her bir bölümü bir sürece atar. Bu teknikte, her süreç, farklı mantıksal adres aralıklarını kullanır. Ancak, işletim sistemi bu mantıksal adresleri, fiziksel bellek adreslerine dönüştürmek ve sürecin bellek içindeki konumunu belirlemek için dinamik yer deđiştirme kullanır. Bu nedenle, kesimleme, bellek adreslerini çalışma zamanında ayarlamak için dinamik yer deđiştirme gerektirir.



Soru

- Dinamik yer deęiřtirmede, bellek adreslerinin alıřma zamanında ayarlanmasının temel faydası nedir?
- A) Adres evirisinin getirdięi ek yk azaltır
- B) Bellek tahsisini optimize ederek bellek paralanmasını en aza indirir
- C) Srelerin daha gvenli bir ortamda yrtlmesini saęlar
- D) Srelerin herhangi bir deęiřiklięe gerek kalmadan mevcut boř bir bellek konumuna yklenmesini saęlar



Cevap

- Cevap: D
- Dinamik yer deęiřtirme, iřletim sistemi tarafından sreç yrtlrken bellek adreslerinin dinamik olarak ayarlanmasını saęlar. Bu, sreçlerin herhangi bir mevcut bellek konumuna deęiřiklik yapmadan yklenmesine olanak tanır. Bu, sreçlerin bellek iindeki konumunu belirlemede ve bellek iinde daha iyi performans gstermesini saęlar.



Soru

- İşletim sistemlerinde dinamik yüklemenin (dynamic loading) temel amacı nedir?
 - A) Kaynak kodunu makine koduna derlemek
 - B) Yürütülebilir kodu yalnızca ihtiyaç duyulduğunda belleğe yüklemek
 - C) Derleme zamanında harici kütüphaneleri bağlamak
 - D) İşletim sistemi başlangıcında süreçler için bellek tahsis etmek



Cevap

- Cevap: B
- Bir program çalıştırıldığında, işletim sistemi öncelikle yürütülebilir kodu diskten belleğe yükler. Ancak, tüm programın belleğe yüklenmesi gerekmez. Özellikle büyük programlar için bu durumda gereksiz bellek kullanımı olabilir. Dinamik yükleme, yürütülebilir kodun yalnızca ihtiyaç duyulduğunda belleğe yüklenmesini sağlar. Dinamik yükleme aynı zamanda programların yürütme süresini de azaltabilir. Örneğin, bir programın tüm bileşenlerini başlangıçta yüklemek yerine, yalnızca kullanıcı bir işlevi çağırdığında ilgili kodu yüklemek daha hızlı yanıt süresi sağlar.



Soru

- Aşağıdakilerden hangisi dinamik bağlamayı (dynamic linking) tanımlar?
- A) Derleme zamanında harici kütüphaneleri bağlamak
- B) Program başlatıldığında kütüphaneleri belleğe yüklemek
- C) Program çalışma zamanında kütüphaneleri programa bağlamak
- D) Dinamik veri yapıları için bellek ayırmak



Cevap

- Cevap: C
- Dinamik bağlama, bir programın çalışma zamanında harici kütüphaneleri programa bağlama işlemidir. Bir program, belirli işlevleri gerçekleştirmek için harici kütüphanelere ihtiyaç duyabilir. Bu kütüphaneler, program tarafından kullanılan işlevleri içerir. Dinamik bağlama, bu kütüphanelerin program tarafından çalışma zamanında yüklenmesini ve programa bağlanmasını sağlar. Bu, programın sadece gerektiği zamanlarda gerekli kütüphaneleri yüklemesini ve bellek kullanımını optimize etmesini sağlar. Dinamik bağlama, bir programın daha küçük boyutta olmasını sağlar çünkü program yalnızca ihtiyaç duyulan kütüphaneleri içerir ve diğer kütüphaneleri çalışma zamanında yükler.



Soru

- Statik bağlamaya kıyasla dinamik yükleme ve bağlamanın temel faydası nedir?
- A) Bellek kullanımının azalması
- B) Daha hızlı program yürütme
- C) Kaynak kullanımında daha büyük esneklik ve verimlilik
- D) Program taşınabilirliğinin artması



Cevap

- Cevap: C
- Dinamik yükleme ve bağlama, bir programın çalışma zamanında harici kütüphaneleri yüklemesine ve programa bağlamasına olanak tanır. Dinamik yükleme ve bağlama, bir programın yalnızca gerektiği zamanlarda harici kütüphaneleri yüklemesine ve programa bağlamasına olanak tanır. Bu, programın çalışma zamanında kütüphane ihtiyaçlarını karşılayabilir ve kaynakların daha etkin kullanılmasını sağlar. Dinamik bağlama, bir programın güncel kütüphane sürümlerini kullanmasını sağlar ve daha esnek bir geliştirme süreci sunar.



Soru

- Aşağıdakilerden hangisi dinamik bağlamaya kıyasla statik bağlamanın bir avantajıdır?
 - A) Daha küçük yürütülebilir boyut
 - B) Azaltılmış bellek gereksinimi
 - C) İyileştirilmiş çalışma zamanı performansı
 - D) Kütüphanelerin daha kolay bakımı ve güncellenmesi



Cevap

- Cevap: A
- Statik bağlama, bir programın derlenmiş sürümüne tüm harici kütüphane kodunu dahil ederek yürütülebilir dosyanın tek bir dosya içinde birleştirilmesini sağlar. Statik bağlama, tüm gerekli kodun birleştirilmesiyle sonuçlanır, bu da yürütülebilir dosyanın daha küçük boyutta olmasını sağlar. Statik bağlama, programın harici kütüphanelere olan bağımlılığını ortadan kaldırır. Bu, programın kütüphaneleri yerel olarak dahil etmesini sağlar. Statik bağlama, bir programın herhangi bir ortamda çalışmasını sağlar, çünkü tüm gerekli kodlar yürütülebilir dosyada mevcuttur. Bu, programın farklı sistemlerde çalışmasını sağlar ve bağımlılıkları en aza indirir.



Soru

- Statik bağlamanın dinamik bağlamaya kıyasla başlıca dezavantajı nedir?
 - A) Artan bellek kullanımı
 - B) Daha yavaş program başlatma süresi
 - C) Harici kütüphanelere bağımlılık
 - D) Kaynak tahsisinde sınırlı esneklik



Cevap

- Cevap: A
- Statik bağlama, tüm gerekli kütüphane kodunu yürütülebilir dosyaya dahil ettiği için bellek kullanımını artırabilir. Her bir programın kopyasında kullanılan kütüphane kodunun birleştirilmesi, bellek kullanımını artırabilir. Özellikle birçok programın aynı kütüphane kodunu paylaştığı durumlarda bu etki daha belirgin olabilir. Statik bağlama, tüm kütüphane kodunu yürütülebilir dosyaya dahil ettiği için yürütülebilir dosyaların boyutunu artırabilir. Bu, programların dağıtımını ve depolanmasını zorlaştırabilir. Statik bağlama, herhangi bir kütüphane güncellemesinin tüm programların yeniden derlenmesini ve yeniden dağıtılmasını gerektirir.



Soru

- İşletim sistemi, dinamik yüklemde hangi rolü oynar?
 - A) Sistem başlangıcında süreçler için bellek tahsis eder
 - B) Program tarafından istendiğinde yürütülebilir kodu belleğe yükler
 - C) Mantıksal adresleri fiziksel adreslere çevirir
 - D) Sistem performansını optimize etmek için bellek sayfalarını yönetir



Cevap

- Cevap: B
- Dinamik yükleme, bir programın çalışma zamanında harici kütüphaneleri ihtiyaç duyulduğunda yüklemesine ve bağlamasına olanak tanır. İşletim sistemi, bu süreçte belleği yönetir ve işletim sistemi, programın belleğe yüklenen kodunu ve verilerini yönetir. Örneğin, bir program bir işlevi kullanmak istediğinde, işletim sistemi bu işlevi içeren kütüphanenin belleğe yüklenmesini sağlar. İşletim sistemi ayrıca, belleğin uygun şekilde kullanılmasını sağlamak için bellek yönetimi politikalarını uygular. Bu politikalar, bellek tahsisini, paylaşımı ve serbest bırakmayı içerir.



Soru

- Dinamik yükleme ve bağlama hakkındaki aşağıdaki ifadelerden hangisi doğrudur?
 - A) Yalnızca program derleme zamanında gerçekleştirilebilir.
 - B) Program performansını artırır ancak bellek kullanımını artırır.
 - C) Programların daha modüler ve esnek olmasına olanak tanır.
 - D) Yalnızca statik bağlama ile birlikte kullanılabilir.



Cevap

- Cevap: C
- Dinamik yükleme ve bağlama, bir programın çalışma zamanında harici kütüphaneleri yüklemesine ve programa bağlamasına olanak tanır. Bu durum, programların daha modüler ve esnek olmasını sağlar çünkü programlar ihtiyaç duydukları kütüphaneleri yalnızca kullanıldıkları zaman yüklerler. Ayrıca, dinamik yükleme ve bağlama, programların daha az bellek kullanmasına olanak tanır çünkü yalnızca ihtiyaç duyulduğunda kütüphane kodu yüklenir ve yürütülür.



Soru

- Aşağıdakilerden hangisi bellek yönetiminde boş alan yönetiminin amacını en iyi şekilde açıklar?
 - A) Bellek kaynaklarını birden çok süreç arasında dağıtmak
 - B) Bellekteki süreçlerin yerleşimini yönetmek ve parçalanmayı en aza indirmek
 - C) Süreçlere tahsis edilmek üzere kullanılabilir bellek bloklarını takip etmek
 - D) Mantıksal adresleri fiziksel adreslere çevirmek



Cevap

- Cevap: C
- Boş alan yönetimi, bellek yönetiminde kullanılabilir bellek bloklarını takip eder. Bu, bellek tahsis işlemlerinde kullanılacak uygun bellek bloklarını bulmayı ve süreçlere tahsis etmeyi sağlar. Örneğin, işletim sistemi genellikle boş alanı bir liste veya veri yapısı aracılığıyla izler ve bellek tahsisi gerektiğinde uygun blokları bulur ve süreçlere tahsis eder. Boş alan yönetimi, bellek yönetiminin önemli bir bileşenidir çünkü işletim sistemi, bellek kaynaklarını etkin bir şekilde kullanmak ve sistem performansını optimize etmek için uygun bellek tahsisi yapmalıdır.



Soru

- Boş alan yönetimde kullanılan yaygın veri yapısı hangisidir?
 - A) Bağlı liste
 - B) Hash tablosu
 - C) İkili ağaç
 - D) Yığın



Cevap

- Cevap: A
- Boş alan yönetimi sırasında, kullanılabilir bellek bloklarını izlemek için genellikle bağlı listeler kullanılır. Bağlı liste, her biri bir bellek bloğunu temsil eden düğümlerden oluşur. Her düğüm, belirli bir bellek bloğunun başlangıç adresini, boyutunu ve bağlantıları içerir. Bağlı listeler, bellek bloklarının eklenmesi, kaldırılması ve güncellenmesi gibi işlemleri etkin bir şekilde yönetmek için kullanılabilir. Örneğin, bir süreç bellek talep ettiğinde, işletim sistemi bağlı listeden uygun bir bellek bloğu bulabilir ve tahsis edebilir. Bellek bloğu kullanıldıktan sonra, işletim sistemi bağlı listeden bu bloğu kaldırabilir veya serbest bırakılan bellek bloğunu izlemek için bağlı liste üzerinde güncelleme yapabilir.



Soru

- Boş alan yönetiminde, iç parçalanma nedir?
 - A) Tahsis edilmiş bloklar arasındaki kullanılmayan bellek
 - B) Bir bellek bloğunun başındaki kullanılmayan bellek
 - C) Tahsis edilmiş bir bellek bloğu içindeki kullanılmayan bellek
 - D) Bir bellek bloğunun sonundaki kullanılmayan bellek



Cevap

- Cevap: C
- İç parçalanma, bir bellek bloğunun, sürecin ihtiyacından daha büyük olduğunda veya süreç boyutu tam blok boyutuna eşit olmadığında oluşur. Bu durumda, işletim sistemi, süreç için daha büyük bir bellek bloğu tahsis eder, ancak süreç bloğun tamamını kullanmaz. Blok içindeki bu kullanılmayan alan, iç parçalanma olarak adlandırılır. İç parçalanmayı azaltmak için, bellek tahsisi yapılırken işletim sistemi, süreç boyutuna mümkün olduğunca yakın bir blok tahsis etmeye çalışır.



Soru

- Hangi bellek tahsis algoritması, istenen boyuta uygun olan ilk kullanılabilir bellek bloğunu seçer?
- A) İlk Uygun (first fit)
- B) Sonraki Uygun (next fit)
- C) En Uygun (best fit)
- D) En Kötü Uygun (worst fit)



Cevap

- Cevap: A
- İlk Uygun (First Fit) bellek tahsis algoritması, bellek bloklarını tararken, istenen boyuta uygun olan ilk boş bellek bloğunu seçer ve süreç için bu bloğu tahsis eder. Oldukça basit ve etkilidir. Bellek blokları tarama sırasında, işletim sistemi, ilk uygun bloğu bulduğunda arama işlemi sona erer ve süreç için bu bloğu tahsis eder. Bu, işletim sisteminin bellek tahsisini hızlı bir şekilde gerçekleştirmesine yardımcı olur. Ancak, daha sonraki tahsisler için daha büyük blokların bırakılmasına neden olabilir, bu da iç parçalanmaya yol açabilir. İlk uygun bloğu seçmek daha sonraki tahsislerde daha kötü eşleşmelere neden olabilir.



Soru

- İlk uygun bellek tahsis algoritmasının temel dezavantajı nedir?
 - A) Aşırı dış parçalanmaya neden olur
 - B) Her tahsis sonrası boş bellek listesini sıralamayı gerektirir
 - C) Tahsis edilen bloklar arasında küçük kullanılamaz boşluklar bırakabilir
 - D) Diğer algoritmalara kıyasla daha yüksek hesaplama maliyetine sahiptir



Cevap

- Cevap: C
- İlk uygun bellek tahsis algoritması, uygun boyuttaki ilk boş bellek bloğunu seçer ve süreç için bu bloğu tahsis eder. Ancak, bu yaklaşım bazen küçük, kullanılamaz boşlukların bırakılmasına neden olabilir. Bu boşluklar, işletim sistemi tarafından yeniden kullanılamaz ve bellek kullanımını optimize etmek için kaynaklarını boşa harcarlar. Bu durum, iç parçalanma olarak adlandırılır ve bellek kullanımını etkili bir şekilde azaltır.



Soru

- Bellek yönetiminde boş alan yönetimi algoritmalarının temel amacı nedir?
- A) Bellek tahsis ve serbest bırakma için gereken zamanı en aza indirmek
- B) Bellek kullanımını maksimize etmek ve parçalanmayı en aza indirmek
- C) Süreçlerin yetkisiz bellek konumlarına erişmesini engellemek
- D) Hızlı erişim için bellek bloklarını ardışık bir şekilde tahsis etmek



Cevap

- Cevap: B
- Boş alan yönetimi algoritmalarının ana hedefi, bellek kullanımını optimize etmek ve parçalanmayı en aza indirmektir. Bu, sistemin toplam bellek kullanımını maksimize etmek ve bellek boşluklarının verimsiz bir şekilde kullanılmasını önlemek için gereklidir. Bellek parçalanması, bellek bloklarının bölünmesi ve verimli bir şekilde kullanılmayan küçük boşluklar bırakılması durumunda oluşur. Bu durum, bellek tahsisi için kullanılabilir bellek miktarını azaltır ve sistemin bellek kullanımını etkili bir şekilde optimize etmesini engeller.



Soru

- Hangi bellek tahsis algoritması, tahsis sonrası en küçük boş alan bırakan bellek bloğunu seçer?
- A) İlk Uygun
- B) Sonraki Uygun
- C) En Uygun
- D) En Kötü Uygun



Cevap

- Cevap: C
- En iyi uygun bellek tahsis algoritması, tahsis edilen boyuttan sonra kalan boş alanın en küçük olduğu bellek bloğunu seçer. Bu algoritma, kullanılabilir boş bellek blokları arasından süreç boyutuna en yakın olanı seçmeye çalışır. Böylece, daha fazla boşluk bırakmadan bellek bloklarının tam olarak kullanılmasını sağlar. Ancak, en iyi uygun algoritması, bellek bloklarının taranması ve boyutlarına göre sıralanması gibi işlemleri gerektirir, bu da bazı durumlarda diğer algoritmalara kıyasla daha fazla hesaplama maliyetine neden olabilir.



Soru

- Boş alan yönetiminde, dış parçalanma nedir?
- A) Tahsis edilmiş bloklar arasında kullanılmayan alan
- B) Bellek bloğunun başlangıcında kullanılmayan alan
- C) Tahsis edilen bir bellek bloğunda kullanılmayan alan
- D) Bellek bloğunun sonunda kullanılmayan alan



Cevap

- Cevap: A
- Dış parçalanma, tahsis edilmiş bellek blokları arasında kullanılmayan boş bellek blokları oluştuğunda ortaya çıkar. Bu durumda, bellek blokları arasında boşluklar oluşur ve bu boşluklar birleştirilerek daha büyük bir bellek bloğu elde edilmez. Bu nedenle, toplam bellek kullanımını azaltır. Dış parçalanma, bellek tahsisi sırasında kullanılmayan küçük boşlukların birikmesiyle oluşabilir. Bu durum, bellek bloklarının toplam kullanılabilirliğini azaltır ve bellek yönetimini karmaşık hale getirebilir.



Soru

- Talebi karşılamak için yeterli toplam bellek alanı mevcut olsa da belleğin ardışık olmaması durumunda hangi tür parçalanma oluşur?
- A) İç parçalanma
- B) Dış parçalanma
- C) Statik parçalanma
- D) Dinamik parçalanma



Cevap

- Cevap: B
- Dış parçalanma, yeterli toplam bellek alanı olduğunda ancak bellek blokları arasında kullanılmayan boşluklar olduğunda ortaya çıkar. Bu durumda, bellek blokları arasında boş alanlar olduğu için, bir talep yerine getirilemez çünkü gereken bellek bloğu ardışık değildir. Dolayısıyla, dış parçalanma, bellek yönetimi sürecinde dikkate alınması gereken önemli bir faktördür ve uygun bellek tahsis algoritmaları kullanılarak en aza indirilmeye çalışılır.



Soru

- Bellek yönetiminde sayfa tablosunun temel işlevi nedir?
 - A) Mantıksal adreslere karşılık gelen fiziksel adresleri saklamak
 - B) Bellek sayfalarının süreçlere tahsisini yönetmek
 - C) Bellek konumlarına yetkisiz erişimi engellemek
 - D) Kütüphanelerin dinamik yüklenmesi ve bağlanmasını kolaylaştırmak



Cevap

- Cevap: A
- Sayfa tablosu, işletim sistemi tarafından kullanılan bir bellek yönetimi veri yapısıdır ve genellikle sanal bellek sistemlerinde kullanılır. Sayfa tablosu, mantıksal adreslerle fiziksel bellek adresleri arasındaki eşlemeyi saklar. Sanal bellek sistemlerinde, her süreç kendi sanal adres uzayını kullanır. Bu sanal adresler, gerçek fiziksel bellek adreslerine eşlenmelidir, bu da sayfa tablosunun rolüdür. Böylece, sayfa tablosu, işlemci tarafından üretilen mantıksal adresleri gerçek fiziksel bellek adreslerine çevirmek için kullanılır. Bu, işlemcinin doğru bellek konumlarına erişmesini sağlar. Sayfa tablosu ayrıca bellek erişim haklarını kontrol etmek ve bellek korumasını uygulamak için de kullanılabilir.



Soru

- Sayfalama ile ilgili aşağıdaki ifadelerden hangisi doğrudur?
- A) Dış parçalanmayı tamamen ortadan kaldırır.
- B) Tüm süreçlerin aynı boyutta olması gerekir.
- C) İç parçalanmaya neden olabilir.
- D) Bellek tahsisi için dinamik yer değiştirme işlemine dayanır.



Cevap

- Cevap: C
- Sayfalama, bir bellek yönetimi tekniği olup, sanal bellek sistemlerinde sıkça kullanılır. Bu sistemde, fiziksel bellek blokları sabit boyutlu parçalara bölünür ve süreçler sanal bellek sayfalarına bölünür. Her bir süreç, fiziksel bellekteki uygun sayfalara eşlenir. Ancak, paging iç parçalanmaya neden olabilir. İç parçalanma, bir süreç tarafından kullanılmayan küçük boşluklara neden olur. Dolayısıyla, paging dış parçalanmayı tamamen ortadan kaldırmaz; ancak, iç parçalanmayı azaltabilir. Sayfa boyutu seçimi, iç parçalanmayı en aza indirmek için önemlidir.



Soru

- Sayfa tablosu girdisinin amacı nedir?
 - A) Sayfanın mantıksal adresini saklamak
 - B) Sayfa çerçevesinin fiziksel adresini saklamak
 - C) Sayfa boyutunu saklamak
 - D) Sayfa için erişim izinlerini saklamak



Cevap

- Cevap: B
- Sayfa tablosu, her bir sayfanın fiziksel bellek içindeki konumunu belirlemek için kullanılır. Sayfa tablosu, mantıksal adreslerle fiziksel bellek adresleri arasındaki eşlemeyi sağlar. Her bir sayfa tablosu girdisi, bir mantıksal sayfayı (veya sayfa numarasını) fiziksel bellekteki bir çerçeveye eşler. Bir sayfa tablosu girdisi, genellikle birkaç alan içerir, ancak en önemli alan fiziksel adres alanıdır.



Soru

- Sayfalamanın ardışık bellek tahsisine kıyasla temel avantajı nedir?
 - A) Azaltılmış iç parçalanma
 - B) Daha hızlı bellek erişim süresi
 - C) Geliştirilmiş bellek koruması
 - D) Bellek tahsisinde daha büyük esneklik



Cevap

- Cevap: D
- Paging, ardışık bellek tahsisine (contiguous memory allocation) kıyasla daha fazla esneklik sağlar. Ardışık bellek tahsisinde, bellek, süreçler arasında ardışık ve kesintisiz bloklara bölünür. Bu, süreçlerin bellek içinde sıkışıp kalmasına ve bellek kullanımını kısıtlamasına neden olur. Ancak, paging kullanıldığında, bellek blokları sabit boyutlu parçalara bölünür (sayfalar) ve süreçler bu sayfalar arasında serbestçe yer değiştirebilir. Paging'in esnekliği, süreçlere bellek içinde daha serbestçe yer bulma imkanı sağlar. Bir süreç, sadece gerektiğinde ihtiyacı olan sayfaları belleğe yükleyebilir ve gereksiz sayfaları bellekten çıkarabilir. Ayrıca, süreçler bellekte rastgele yerleştirilebilir.



Soru

- TLB hakkında aşağıdaki ifadelerden hangisi doğrudur?
- A) Mantıksal adreslere eşlenmiş fiziksel adresleri içerir.
- B) Ana belleğin bir parçasıdır.
- C) Sayfa tablosuna erişildikten sonra erişilir.
- D) Kesimlemeyi yönetmek için kullanılır.



Cevap

- Cevap: A
- TLB, mantıksal adreslerin fiziksel bellek adreslerine dönüştürülmesinde hızlı erişim sağlayan bir önbellek türüdür. Mantıksal adresler, işlemci tarafından kullanılan adresleme düzenidir, ancak bu adresler doğrudan fiziksel bellek adreslerine karşılık gelmezler. Bu nedenle, bir işlemci, bir sürecin belleğe erişim taleplerini işlemek için mantıksal adresleri kullanır ve bu adreslerin karşılık gelen fiziksel adreslerini belirlemek için de TLB'yi kullanır. TLB, mantıksal adresleri doğrudan fiziksel adreslere çevirerek işlemcinin bellek erişimini hızlandırır. Bu, sayfa tablosuna (page table) her erişimde fiziksel belleğe doğrudan erişim gerektiğinden daha hızlı bir işlemi mümkün kılar.



Soru

- TLB, bellek erişim performansını nasıl artırır?
 - A) Sayfa tablosunun boyutunu azaltarak
 - B) Daha hızlı çeviri için sık erişilen sayfa tablosu girdilerini saklayarak
 - C) Bellek kesimlerini verimli bir şekilde yöneterek
 - D) Ana bellek ile ikincil depo arasında bellek sayfalarını değiştirerek



Cevap

- Cevap: B
- TLB, bellek erişim performansını artırmak için kullanılan bir önbellek türüdür. TLB, mantıksal adreslerin fiziksel bellek adreslerine dönüştürülmesini hızlandırır. Bunun temel nedeni, TLB'nin sık erişilen sayfa tablosu girdilerini saklamasıdır. İşlemci, bir programın mantıksal adresine eriştiğinde, bu adresi fiziksel bellek adresine çevirmek için sayfa tablosuna başvurur. Ancak, bu işlem zaman alıcı olabilir, çünkü sayfa tablosu büyük olabilir ve her bellek erişiminde tamamen taraması gerekebilir. Bu durumu hafifletmek için sık erişilen sayfa tablosu girdileri, TLB içinde saklanır. Bu, işlemcinin sık erişilen bellek adreslerini doğrudan TLB'den almasını sağlar, böylece sayfa tablosuna erişme gereksinimini azaltır.



Soru

- Bellek yönetiminde, kesimleme nedir?
 - A) Ana belleği sabit boyutlu bloklara bölmek
 - B) Ana belleği değişken boyutlu bloklara bölmek
 - C) Mantıksal adresleri fiziksel adreslere çevirme
 - D) TLB girdilerini yönetme



Cevap

- Cevap: B
- Kesimleme, ana belleđi deđişken boyutlu bloklara bölme işlemidir. Bellek kesimleri, bir programın mantıksal adres alanını mantıksal olarak ilgili parçalara bölen farklı boyutlarda bellek bölgeleridir. Bu kesimler, programın farklı bölümlerini (örneğin kod, veri, yığın) temsil eder ve her biri farklı boyutlarda olabilir. Kesimleme, bir programın bellek gereksinimlerine daha iyi uyarlanmasını sağlar. Örneđin, bir programın kod segmenti diğerlerinden daha büyük olabilirken, veri segmenti daha küçük olabilir. Bu nedenle, kesimleme, esnek bir bellek yönetimi stratejisi sunar.



Soru

- TLB içinde istenen adres çevirisi bulunamazsa ne olur?
- A) CPU, çeviriyi ana bellekteki sayfa tablosundan alır.
- B) CPU, bir sayfa hatası oluşturur.
- C) CPU, belirli bir gecikmeden sonra tekrar TLB önbelleğini arar.
- D) CPU, çeviriyi başka bir CPU çekirdeğinin TLB önbelleğinden alır.



Cevap

- Cevap: A
- Eğer TLB'de istenen adres çevirisi bulunamazsa, CPU bu çeviriyi ana bellekteki sayfa tablosundan alır. Her adres çevirisi TLB önbelleğinde bulunmayabilir. Bu durumda, CPU, gerekli çeviriyi ana bellekteki sayfa tablosundan alarak işlemi tamamlar. Bu işlem, TLB'nin önbellek dışı bir çeviri talebini ele alma mekanizmasıdır ve bellek erişim süresini biraz artırır, ancak sistem performansını ciddi şekilde etkilemez.



Soru

- Aşağıdakilerden hangisi sayfa hatasını tetikler?
 - a. Bellekte bulunan bir sayfaya erişme
 - b. Bellekte bulunmayan bir sayfaya erişme
 - c. Bellekte kilitli olan bir sayfaya erişme
 - d. Salt okunur olarak işaretlenmiş bir sayfaya erişme



Cevap

- Cevap: b
- Bellekte bulunmayan bir sayfaya erişmeye çalışmak, sayfa hatasına neden olur çünkü işletim sistemi, talep edilen sayfanın diskten belleğe yüklenmesi gerektiğini anlar. Sayfa hatası işlemi, istenen sayfanın belleğe yüklenmesiyle tamamlanır ve ardından süreç yeniden başlatılır. Sayfa hataları, talep edilen verinin bellekte olmayabileceği durumlarda ortaya çıkar ve sanal bellek sistemlerinde sıkça görülür. Bu hatanın önlenmesi için sayfalar önceden yüklenmez ve sadece gerçekten gerektiğinde belleğe yüklenir.



Soru

- Bir sayfa hatası meydana geldiğinde işletim sistemi ne yapar?
 - a. Süreci hemen sonlandırır
 - b. Tüm süreci disk belleğine taşır
 - c. İstenen sayfa için yeni bir sayfa çerçevesi ayırır
 - d. İstenen sayfayı ikincil depodan alır



Cevap

- Cevap: d
- Sayfa hatası, istenen veri sayfasının bellekte bulunmadığını belirtir. İşletim sistemi, sayfa hatası oluştuğunda bu hatayı ele alır ve istenen sayfayı diskten belleğe yüklemek için gerekli işlemleri gerçekleştirir. İkincil depo (genellikle sabit disk) üzerinde bulunan istenen sayfa, diskten belleğe kopyalanır. Ardından, süreç yeniden başlatılır ve bu kez istenen veri bellekte bulunur. Sayfa hataları, sanal bellek sistemlerinde sıkça karşılaşılan durumlardır ve veriye erişim gerektiğinde otomatik olarak işletim sistemi tarafından yönetilirler.



Soru

- Aşağıdakilerden hangisi yüksek sayfa hata oranının sonuçlarından değildir?
 - a. Sistem performansının azalması
 - b. Disk G/Ç'sinin artması
 - c. CPU kullanımının artması
 - d. Bellek kullanımının artması



Cevap

- Cevap: c
- Yüksek bir sayfa hata oranı, genellikle sistem performansının azalmasına, artan disk G/Ç işlemlerine ve artan bellek kullanımına neden olur. Ancak, sayfa hata oranının artması, CPU kullanımının doğrudan artmasına neden olmaz. Bunun yerine, CPU'nun iş yükünü artırabilecek ek disk G/Ç işlemleri ve bellek erişimiyle ilgili işlemler olabilir. Sayfa hata oranı genellikle sistem yükünü artırır, ancak bu, CPU kullanımının doğrudan bir sonucu değildir.



Soru

- Çok seviyeli bir sayfa tablosunda, en dıştaki sayfa tablosunun amacı nedir?
 - a. Sanal sayfaları disk adreslerine eşlemek
 - b. Sayfa tablosunun boyutunu azaltmak
 - c. Sayfalar için erişim izinlerini yönetmek
 - d. Sanal sayfaları bellekte fiziksel sayfalara eşlemek



Cevap

- Cevap: d
- Çok seviyeli bir sayfa tablosunda, en dıştaki sayfa tablosu, sanal adres uzayındaki sanal sayfaları fiziksel bellekteki fiziksel sayfalara eşler. Bu, sanal bellek yönetiminde adres çevirisi için gerekli olan yapısal bir düzendir. En dıştaki sayfa tablosu, sanal adres uzayının belirli bölümlerini daha içteki sayfa tablolarına yönlendirir, böylece belirli bir sanal sayfa numarası, fiziksel bellekteki doğru sayfa numarasını bulmak için kullanılabilir. Bu, bellek erişimi sırasında gerekli olan adres dönüşümünü sağlar ve fiziksel bellek içindeki yerleri belirler.



Soru

- FIFO (First-In-First-Out) sayfa yer deęiřtirme algoritmasının kullanımının bir dezavantajı hangisidir?
 - a. Karmařık hesaplama gerektirir
 - b. Belady anomalisiyle karřılařır
 - c. Her zaman en son kullanılan sayfayı seęer
 - d. Sayfa tablosuna sık g¼ncellemeler gerektirir



Cevap

- Cevap: b
- FIFO sayfa yer deęiřtirme algoritması, en eski giren sayfayı önce çıkartır. Belady Anomalisi, FIFO'nun beklenmedik bir řekilde performansının kötüleřebileceęi durumu ifade eder. Bu durumda, daha fazla fiziksel belleęin kullanılmasına raęmen, daha fazla sayıda sayfa hatası alınabilir. Yani, FIFO'nun performansı arttıkça, sayfa hatası sayısının artması beklenir. Daha iyi bir performans elde etmek için kullanılan fiziksel bellek miktarının artmasına raęmen, daha fazla sayfa hatası alınabilir.



Soru

- Bir sistem aynı sayfa için birden fazla sayfa hatasını nasıl önler?
 - a. Bellekte sayfayı kilitleyerek
 - b. Baştan sona yazma (write-through) ön bellek politikası kullanarak
 - c. TLB kullanarak
 - d. Sayfayı kirli olarak işaretleyerek



Cevap

- Cevap: a
- Bir sayfa için birden fazla sayfa hatasını önlemek için sistem, sayfayı bellekte kilitlet. Bu, sayfanın bellekte tutulmasını ve erişiminin diğer işlemler tarafından değiştirilmesini engeller. Eğer sayfa bellekte kilidini kaybederse ve değiştirilirse, bu durum sayfa hatasına neden olabilir ve bu da gereksiz bellek giriş/çıkış işlemlerine yol açabilir. Dolayısıyla, sayfa bellekte kilitleme, aynı sayfa için birden fazla sayfa hatasını engellemenin etkili bir yoludur. Diğer seçenekler, TLB adres çevirisini hızlandırırken, kirli sayfaların işaretlenmesi sayfanın değiştiğini belirtir.



Soru

- Ters çevrilmiş (inverted) sayfa tabloları kullanmanın temel avantajı nedir?
 - a. Azaltılmış bellek gereksinimi
 - b. Daha hızlı sayfa tablosu araması
 - c. İyileştirilmiş önbellek tutarlılığı
 - d. Sanal bellek parçalanması için daha iyi destek



Cevap

- Cevap: a
- Ters çevrilmiş sayfa tablolarının en önemli avantajı, azaltılmış bellek gereksinimidir. Geleneksel bir sayfa tablosunda, her süreç için bir sayfa tablosu bulunur ve bu tablolar genellikle büyük olabilir, çünkü her biri tüm fiziksel bellek sayfalarını yönetir. Ancak ters çevrilmiş sayfa tablolarında, her bir sayfa girdisi genellikle bir süreç ve sayfa numarası çiftini eşler, böylece her süreç için tek bir girdiye ihtiyaç duyulur. Diğer seçenekler, sayfa tablosu aramasının hızı TLB kullanımıyla ilgilidir, önbellek tutarlılığı önbellek yönetimiyle ilgilidir ve sanal bellek parçalanması sayfa yer değiştirme algoritmaları ile ilgilidir.



Soru

- En düşük sayfa hata oranını garanti eden algoritma hangisidir?
 - a. FIFO (İlk Giren İlk Çıkar)
 - b. Optimal
 - c. LRU (En Son Kullanılan)
 - d. LFU (En Az Kullanılan)



Cevap

- Cevap: b
- Optimal sayfa yer deęiřtirme, verilen bir iř iin en dūřuk sayfa hata oranını garanti eder. Bu algoritma, gelecekte hangi sayfaların iřlemci tarafından kullanılacaęını bilir. Bu nedenle, en az sayıda sayfa hatası oluřacak Őekilde sayfaları yer deęiřtirir. Ancak, pratikte uygulanması zordur, ünkü gerek zamanlı olarak gelecekte hangi sayfaların kullanılacaęını bilmek mūmkūn deęildir. Bununla birlikte, dięer sayfa yer deęiřtirme algoritmalarının performansını Őlmek iin bir referans olarak kullanılabilir. FIFO, en eski giren sayfayı ıkarır ve Belady Anomalisi gibi durumlarda performansı kŐtūleřebilir. LRU, en son kullanılan sayfayı yer deęiřtirir ve belirli kořullarda sayfa hatalarına neden olabilir. LFU, en az kez kullanılan sayfayı yer deęiřtirir, ancak bazı durumlarda performansı dūřuk olabilir.



Soru

- Hangi sayfa yer deęiřtirme algoritması "thrashing" ile karřılařır?
 - a. FIFO (İlk Giren İlk Çıkar)
 - b. LRU (En Son Kullanılan)
 - c. Optimal
 - d. Çalışma Kümesi



Cevap

- Cevap: a
- "Thrashing", bilgisayarın işlem hızının dramatik düştüğü ve işlemci, disk ve diğer kaynaklar arasında sürekli bir rekabetin olduğu durumdur. Bu durumda, işlemci, yetersiz fiziksel bellek nedeniyle, sürekli olarak sayfa değişimleri yapmak zorunda kalır. FIFO (İlk Giren İlk Çıkar) sayfa yer değiştirme algoritması, özellikle çalışma kümelerini takip etmediğinde veya yetersiz fiziksel bellek varken kullanıldığında, en eski giren sayfayı çıkarttığı için, sayfalar sıklıkla tekrar yüklenebilir. Diğer seçenekler, daha sık kullanılan sayfaları koruyarak işlemci tarafından daha etkin bir şekilde kullanılır. Çalışma Kümesi, işlemci tarafından belirli bir zaman diliminde kullanılan sayfaları izlemek için kullanılan bir kavramdır ve "thrashing" oluşumunu engellemeye yardımcı olur.



Soru

- Hangi sayfa yer deęiřtirme algoritması sayfa erişimlerinin yakınlığını dikkate alır?
 - a. FIFO (İlk Giren İlk Çıkar)
 - b. Optimal
 - c. LRU (En Son Kullanılan)
 - d. Saat (Clock)



Cevap

- Cevap: c
- LRU (Least Recently Used - En Son Kullanılan), sayfa erişimlerinin yakınlığını dikkate alır. En son kullanılan sayfanın yerine en uzun süredir erişilmemiş olan sayfanın değiştirilmesini sağlar. Bir sayfaya erişildiğinde, o sayfanın en son kullanıldığı zamana kadar geçen süre ölçülür ve kullanılan sayfa olarak işaretlenir. LRU algoritması, en son kullanılan sayfaların bellekte kalma olasılığını artırırken, nadiren kullanılan sayfaların bellekten çıkarılma olasılığını artırır. Bu, sayfa hatalarını azaltır. Diğer seçenekler, FIFO (İlk Giren İlk Çıkar), en eski sayfayı değiştirir. Optimal algoritma, gelecekteki tüm sayfa erişimlerini öngörerek en iyi kararı verir, Saat (Clock) algoritması, bir referans biti kullanarak döngü içinde sayfaları yer değiştirir.



Soru

- Saat sayfa yer deęiřtirme algoritmasında, el iřaretçisi (hand pointer) neyi iřaret eder?
 - a. Deęiřtirilecek sonraki sayfayı
 - b. En son eriřilen sayfayı
 - c. Geçerli zamanı
 - d. Toplam sayfa hata sayısını



Cevap

- Cevap: a
- Saat (Clock) sayfa yer deęiřtirme algoritması, dairesel olarak dñzenlenmiř bir sayfa çerçevesi dizisi kullanır. "clock hand", bu dizide bir noktayı temsil eder ve sayfa çerçevelerini dolařır. Algoritma, bu el iřaretçisinin iřaret ettięi sayfa çerçevesine bakar ve eęer kullanımdaysa, bir sonraki sayfa çerçevesine geçer. Eęer sayfa çerçevesi kullanılmıyorsa, bu sayfa çerçevesi deęiřtirilmek üzere seęilir. Yani, el iřaretçisi, bir sonraki deęiřtirilecek sayfa çerçevesini gösterir.



Soru

- Hangi algoritma, sık kullanılan sayfaların bellekte daha uzun süre kalmasına neden olan "aging" problemine duyarlıdır?
 - a. FIFO (İlk Giren İlk Çıkar)
 - b. LRU (En Son Kullanılan)
 - c. Optimal
 - d. LFU (En Az Kullanılan)



Cevap

- Cevap: d
- LFU (Least Frequently Used - En Az Kullanılan) sayfa yer deęiřtirme algoritması, sayfa erişimlerinin sıklığına göre sayfaların sıralandığı ve en az sıklıkla kullanılan sayfanın bellekten çıkarıldığı bir algoritmadır. Ancak, LFU algoritması "aging" problemine duyarlıdır. "Aging" problemi, bir sayfanın sıklığı zamanla deęiřtiğinde, LFU'nun sıkça kullanılan sayfaları doęru bir şekilde tespit etmekte zorlanmasına neden olur. Özellikle, bir sayfanın sıklığı arttıkça, LFU algoritması bu sayfayı hala nadiren kullanılan bir sayfa olarak görebilir ve onu bellekten çıkartmaz, böylece sıkça kullanılan sayfalar bellekte daha uzun süre kalır.



Soru

- Clock sayfa yer deęiřtirme algoritmasının FIFO'ya göre avantajı nedir?
 - a. Daha düşük uygulama karmařıklığı
 - b. Garantili optimal performans
 - c. Thrashing riskinin azaltılması
 - d. Önbelleęin daha verimli kullanımı



Cevap

- Cevap: a
- Clock (Saat) sayfa yer deęiřtirme algoritması, FIFO'ya gre bazı avantajlara sahiptir. En nemli avantajı, daha dřk uygulama karmařıklıęıdır. FIFO algoritması, geleneksel bir kuyruk yapısı kullanır ve bellekten ıkarılacak sayfa her zaman en eski giren sayfadır. Bu nedenle, FIFO algoritması iin bir kuyruk yapısı tutulması ve bu kuyruęun gncellenmesi gereklidir. Ancak Clock algoritması, FIFO'ya benzer bir dairesel yapı kullanır, ancak ek olarak bir "el iřaretisi" (clock hand) kullanır. Bu iřareti, dairesel olarak ilerler ve her seferinde bir sayfa erevesine iřaret eder. Dolayısıyla, bellekten ıkarılacak sayfa erevesini bulmak iin FIFO'ya kıyasla daha az hesaplama yapılır.



Soru

- Basitliđi nedeniyle genellikle gerek zamanlı sistemlerde kullanılan algoritma hangisidir?
 - a. LRU (En Son Kullanılan)
 - b. FIFO (İlk Giren İlk ıkar)
 - c. Optimal
 - d. LFU (En Az Kullanılan)



Cevap

- Cevap: b
- FIFO (İlk Giren İlk Çıkar) sayfa yer deęiřtirme algoritması, basitlięi nedeniyle genellikle gerek zamanlı sistemlerde tercih edilir. Bu algoritma, belleęe ilk giren sayfanın bellekten ıkarılmasını gerektirir. FIFO'nun uygulanması oldukça basittir ünkü sadece bir kuyruk yapısı kullanır ve yeni bir sayfa belleęe eklendięinde, en eski giren sayfanın yerini deęiřtirir. Gerek zamanlı sistemler, belirli zaman kısıtlamalarına uymak zorunda oldukları iin basit ve ngörülebilir algoritmaları tercih ederler.



Soru

- Aşağıdakilerden hangisi, sayfa referanslarının son geçmişine bakarak dinamik olarak değişen bir politika uygular?
 - a. Optimal
 - b. Çalışma Kümesi
 - c. LRU (En Son Kullanılan)
 - d. LFU (En Az Kullanılan)



Cevap

- Cevap: b
- Çalışma Kümesi (Working Set) algoritması, sayfa referanslarının geçmişine dayalı olarak sayfa yer değiştirme politikasını dinamik olarak ayarlayarak çalışır. Bu algoritma, her bir işlemcinin çalışma kümesini (işlemcinin belirli bir zaman diliminde kullandığı sayfaların kümesi) izler ve bu kümedeki sayfaları bellekte tutar. İşlemcinin çalışma kümesi, zamanla değişebilir ve bu nedenle çalışma kümesi algoritması, bu değişikliklere uyum sağlamak için sayfa yer değiştirme politikasını ayarlar. Özellikle, iş yükünün zamanla değiştiği veya öngörülemeyen sayfa erişim örüntülerine sahip olunan durumlarda kullanışlıdır.



Soru

- Saat sayfa yer deęiřtirme algoritmasında, iřaretçi nasıl hareket eder?
 - a. Saat yönünde
 - b. Saat yönünün tersine
 - c. Sayfa referans bitine baęlı olarak
 - d. Rastgele



Cevap

- Cevap: a
- Saat sayfa yer deęiřtirme algoritmasında, iřaretçi (clock hand) saat yönünde hareket eder. Bu algoritma, sayfa çerçevelerini dairesel bir yapıda düzenler ve iřaretçi de bu dairesel yapıda dolařır. Her sayfa erişiminde, iřaretçi bir sonraki sayfa çerçevesine ilerler. Bu iřlem, dairesel yapıda bir tur tamamlanana kadar devam eder. Eęer bir sayfa çerçevesi kullanılmıyorsa, iřaretçi o sayfa çerçevesine gelince, sayfa yer deęiřtirme iřlemi yapılır. Bu řekilde, saat yönünde hareket eden el iřaretçisi, bir sayfa çerçevesinin seęilmesi için kullanılır.



Soru

- Saat sayfa yer deęiřtirme algoritmasında, bir sayfa hatası oluřtuęunda ve mevcut sayfanın referans biti ayarlı ise ne yapılır?
 - a. Sayfa deęiřtirilir
 - b. Referans biti temizlenir
 - c. El iřaretçisi bir sonraki sayfaya hareket eder
 - d. Sayfa bellekte tutulur



Cevap

- Cevap: d
- Saat sayfa yer deęiřtirme algoritmasında, bir sayfa hatası meydana geldiğinde ve mevcut sayfanın referans biti ayarlandığında, sayfa bellekte tutulur. Clock algoritması, bir sayfanın bellekte kalıp kalmayacağını belirlemek için referans bitini kullanır. Referans biti, bir sayfanın son erişimini gösterir. Eğer bir sayfanın referans biti ayarlanmışsa, bu, sayfanın son bir döngü süresi içinde kullanıldığı anlamına gelir ve dolayısıyla sayfanın bellekte kalmasının devam etmesi gerektięi anlamına gelir. Bu nedenle, Clock algoritması, sayfa hatası meydana geldiğinde ve referans biti ayarlanmışsa, bu sayfanın bellekte kalmasını sağlar.



Soru

- Hangi sayfa yer deęiřtirme algoritması, en uzun süredir kullanılmayan sayfayı atma prensibine dayanır?
 - a. FIFO (İlk Giren İlk Çıkar)
 - b. Optimal
 - c. LRU (En Son Kullanılan)
 - d. Saat (Clock)



Cevap

- Cevap: c
- LRU (Least Recently Used - En Son Kullanılan) algoritması, en uzun süredir kullanılmayan sayfayı atma prensibine dayanır. En son kullanılan sayfanın yerine en uzun süredir erişilmemiş olan sayfanın değişimini gerektirir. Bir sayfanın son erişiminin üzerinden ne kadar fazla zaman geçmişse, bir sonraki yer değiştirme adayı olur. LRU algoritması, bellekte en az kullanılan veya en uzun süredir kullanılmayan sayfaları çıkarma eğilimindedir. LRU algoritması, sayfa yer değiştirme stratejileri arasında popülerdir, genellikle etkilidir ve özellikle sayfa erişim örüntülerinin dinamik olduğu durumlarda iyi performans gösterir. Ancak, uygulama maliyeti yüksektir, her bir sayfanın son erişimini takip eder.



Soru

- Optimal sayfa yer deęiřtirme algoritmasının başlıca dezavantajı nedir?
 - a. Yüksek uygulama karmařıklıęı
 - b. Deęiřen bellek eriřim örüntülerine uyum saęlayamama
 - c. Thrashing'e duyarlı olma
 - d. Gelecekteki sayfa referans bilgilerine baęımlılık



Cevap

- Cevap: d
- Optimal sayfa yer deęiřtirme algoritması, gelecekteki sayfa erişimlerini önceden bilerek en iyi sayfa yer deęiřtirme kararını verir. Bu nedenle, dięerlerinden teorik olarak daha iyi performans saęlar. Ancak, gelecekteki sayfa erişim bilgilerine baęımlıdır. Gelecekteki sayfa erişimleri önceden bilinemedięi için Optimal algoritması pratik uygulamalarda kullanılmaz.



Soru

- LFU (En Az Kullanılan) sayfa yer deęiřtirme algoritmasında, sayfa erişim sıklığı nasıl belirlenir?
 - a. Sayfa hata sayısını sayarak
 - b. Sayfa erişim zamanlarını takip etmek için bir zamanlayıcı kullanarak
 - c. Her sayfa için bir sayaç tutarak
 - d. Sayfa referans geçmişini analiz ederek



Cevap

- Cevap: c
- LFU (En Az Kullanılan) sayfa yer deęiřtirme algoritması, sayfa eriřim sıklıęını belirlemek için her bir sayfa için bir sayaç tutar. Her sayfa bir kez belleęe alındıęında, bařlangıçta sayaç deęeri 1 olarak atanır. Daha sonra, sayfaya her eriřimde, ilgili sayfa için sayaç bir artırılır. Bu řekilde, LFU algoritması, her bir sayfanın kaç kez eriřildięini izler ve en az sıklıkla kullanılan sayfaları belirler. Sayfa eriřim sıklıęına dayalı olarak sayfa yer deęiřtirme kararları verilirken, LFU algoritması, her bir sayfanın eriřim sıklıęını doęrudan göz önünde bulundurur. Sıklıkla kullanılan sayfaların bellekte kalma olasılıęını artırırken, nadiren kullanılan sayfaların bellekten çıkarılma olasılıęını artırır.



Soru

- "Yaşlanma" sorununa yatkın olan sayfa yer değiştirme algoritması hangisidir? (eski sayfaların istenenden daha uzun süre bellekte tutulma olasılığı)
 - a. FIFO (İlk Giren İlk Çıkar)
 - b. Optimal
 - c. LRU (En Son Kullanılan)
 - d. LFU (En Az Kullanılan)



Cevap

- Cevap: d
- LFU (En Az Kullanılan) sayfa yer deęiřtirme algoritması, yařlanma problemine yatkındır. "Yařlanma" problemi, bir sayfanın sıklıęının zamanla arttıęı durumlarda ortaya ıkar ve LFU'nun bu deęiřen davranıřa uyum saęlayamaması sonucu oluřur. LFU algoritması, en az kullanılan sayfaları bellekten ıkarmaya eęilimlidir; ancak, bir sayfanın sıklıęı zamanla artarsa, LFU bu sayfayı hala nadiren kullanılan bir sayfa olarak algılayabilir ve bellekten ıkarmayabilir. Sonu olarak, eski sayfalar beklenenden daha uzun sre bellekte tutulabilir.



Soru

- Saat sayfa yer deęiřtirme algoritmasının temel dezavantajı nedir?
 - a. Sayfa erişim örüntülerini dikkate almaz
 - b. Sayfa referans bitlerinin sık sık güncellenmesini gerektirir
 - c. Belady anomalisiyle karşılaşır
 - d. Sık erişilen sayfaları gereksiz yere bellekte tutabilir



Cevap

- Cevap: a
- Saat sayfa yer deęiřtirme algoritması, sayfa eriřim örüntülerini dikkate almaz. Bir sayfaya ne sıklıkta eriřildięini veya en son ne zaman eriřildięini göz önünde bulundurmaz. Saat algoritması, bir dairesel liste yapısı kullanır ve bir referans bitine dayalı olarak sayfa yer deęiřtirme kararları verir. Ancak, bu kararlar, sayfanın gerçek kullanım sıklığına veya eriřim zamanına dayanmaz.



Soru

- FIFO sayfa yer deęiřtirme algoritmasının temel avantajı nedir?
 - a. En düşük sayfa hata oranını garanti eder
 - b. Thrashing'e karřı dirençlidir
 - c. Uygulama karmařıklığı düşüktür
 - d. Bellek erişim örüntülerinin deęişmesine iyi uyum sağlar



Cevap

- Cevap: c
- FIFO (First-In, First-Out - İlk Giren, İlk Çıkar) sayfa yer deęiřtirme algoritması, uygulama karmařıklığı açısından dūřüktür. Bu algoritma, basit bir kuyruk yapısı kullanır ve bellekteki en eski sayfayı, yani ilk giren sayfayı çıkarır. Yeni bir sayfa belleęe eklendięinde, FIFO algoritması en eski sayfanın yerini deęiřtirir. FIFO'nun uygulanması kolaydır çünkü karmařıklığı dūřüktür. Tek yapılması gereken, belleęe yeni bir sayfa eklendięinde, en eski sayfanın yerine yeni sayfanın eklenmesidir.



Soru

- FAT32 dosya sistemi tarafından desteklenen maksimum dosya boyutu nedir?
- A) 2 GB
- B) 4 GB
- C) 16 GB
- D) 32 GB



Cevap

- Cevap: B
- FAT32 dosya sistemi, bir dosyanın maksimum boyutunu 4 GB olarak sınırlar. Bu, tek bir dosyanın 4 gigabaytı geçemeyeceği anlamına gelir. FAT32 dosya sistemi, dosya boyutu sınırlamaları açısından FAT16'dan daha esnek olsa da, modern dosya sistemleri olan NTFS (Windows) veya exFAT (genel olarak taşınabilir aygıtlarda kullanılır) gibi diğer dosya sistemleri daha büyük dosya boyutlarına izin verir.



Soru

- Aşağıdakilerden hangisi ext4 dosya sisteminin bir özelliğidir?
 - A) 4 TB'a kadar dosya boyutunu destekler
 - B) Maksimum birim (volume) boyutu 16 TB'dir
 - C) Dosya adları 256 karakterle sınırlıdır
 - D) Günlükleme (journaling) desteği vardır



Cevap

- Cevap: D
- Ext4 dosya sistemi, günlükleme (journaling) desteği ile bilinir. Günlükleme, bir dosya sisteminde yapılan değişikliklerin bir günlük dosyasına kaydedilmesi işlemidir. Bu, sistemde beklenmedik bir güç kesintisi veya çökme gibi olaylar meydana geldiğinde dosya sistemi bütünlüğünü korumaya yardımcı olur. Ext4 dosya sistemi, dosya adlarını 255 karakterle sınırlar.



Soru

- Unix dosya sistemine göre, "chmod 755 dosya.txt" komutu ne yapar?
- A) Sahibine okuma, yazma ve yürütme izinleri verir, grup ve diğerlerine ise sadece okuma ve yürütme izinleri verir
- B) Sahip, grup ve diğerlerine okuma ve yürütme izinleri verir
- C) Sahibine tüm izinleri verir, grup ve diğerlerine ise sadece okuma izni verir
- D) Dosyadan tüm izinleri kaldırır



Cevap

- Cevap: A
- Bu komut dosya.txt adlı dosyaya izinler atar. "chmod" komutu, dosya veya dizinlerin izinlerini değiştirmek için kullanılırken, "755" ise verilecek izinleri belirtir.
- Dosya sahibi (owner): $7 = 4$ (okuma) + 2 (yazma) + 1 (yürütme).
- Grup (group): $5 = 4$ (okuma) + 1 (yürütme).
- Diğerleri (others): $5 = 4$ (okuma) + 1 (yürütme).



Soru

- MacOS'ta hangi dosya sistemi kullanılır?
- A) NTFS
- B) FAT32
- C) APFS
- D) exFAT



Cevap

- Cevap: C
- APFS, macOS ve diğer Apple cihazlarında (iOS, iPadOS, tvOS, watchOS) varsayılan dosya sistemidir. APFS, HFS+ (Hierarchical File System Plus) dosya sistemine kıyasla birçok avantaj sunar. Özellikle, performans, veri bütünlüğü, dosya şifreleme ve depolama alanı yönetimi gibi alanlarda iyileştirmeler getirir.



Soru

- Dosya sisteminde gnlkleme (journaling) zelliđinin temel amacı nedir?
- A) Dosyaları daha verimli depolayabilmek iin sıkıřtırmak
- B) Dosya sistemi meta verilerinin bir yedeđini sađlamak
- C) kme veya g kesintisi durumunda kurtarma iin dosya sistemi deđiřikliklerinin bir gnlgn tutmak
- D) Gvenlik amalarıyla dosyaları řifrelemek



Cevap

- Cevap: C
- Dosya sisteminde günlükleme (journaling), dosya sistemi deęişikliklerinin bir günlüğünü tutar. Bu, dosya sisteminde meydana gelen deęişikliklerin bir günlük dosyasına kaydedilmesi anlamına gelir. Eđer bir çökme veya güç kesintisi gibi beklenmedik bir durum olursa, dosya sistemi bu günlük dosyasını kullanarak son yapılan deęişiklikleri geri alabilir ve dosya sisteminin bütünlüğünü koruyabilir. Bu, veri kaybını önlemeye yardımcı olur ve dosya sisteminin kurtarılabilirliğini artırır.



Soru

- Aşağıdakilerden hangisi dosya sistemlerinde kullanılan bir dosya tahsis yöntemi türü DEĞİLDİR?
- A) Ardışık tahsis (contiguous)
- B) Bağlı tahsis (linked)
- C) İndeksli tahsis (indexed)
- D) Bölümlenmiş tahsis (partitioned)



Cevap

- Cevap: D
- Bölümlenmiş tahsis (Partitioned allocation) dosya sistemlerinde kullanılan bir dosya tahsis yöntemi türü değildir. Ardışık tahsis (Contiguous allocation), dosyaların ardışık bir şekilde yerleştirilmesini sağlayan bir yöntemdir. Bağlı tahsis (Linked allocation), dosyaların bağlı listeler şeklinde yerleştirilmesini sağlayan bir yöntemdir. İndeksli tahsis (Indexed allocation), dosyaların indeks tabloları aracılığıyla yerleştirilmesini sağlayan bir yöntemdir.



Soru

- Aşağıdakilerden hangisi FAT dosya sisteminin bir kısıtlamasıdır?
 - A) Dosya parçalanması için sınırlı destek
 - B) Her dizinde sınırlı sayıda dosya
 - C) Uzun dosya isimlerini desteklemez
 - D) Dosya izinlerini desteklemez



Cevap

- Cevap: B
- FAT dosya sisteminin bir kısıtlaması, her dizinde sınırlı sayıda dosya olmasıdır. FAT dosya sistemi, FAT12, FAT16 ve FAT32 olmak üzere farklı sürümlerde mevcuttur ve her bir sürüm, dosya ve dizinler için belirli bir maksimum sayıda girdiye izin verir. Özellikle eski FAT dosya sistemleri, her dizin için sınırlı sayıda dosya ve alt dizin girdisi destekler. Bu, büyük ve karmaşık dosya yapılarına sahip sistemlerde sınırlamalara neden olur.



Soru

- Unix dosya sisteminde, "ls -l" komutu neyi görüntüler?
- A) Sadece dosya adlarını
- B) Dosyalar hakkında izinler, sahiplik, boyut ve değiştirilme zamanı dahil detaylı bilgiler
- C) Gizli dosyalar dahil tüm dosyaların listesini
- D) Disk kullanımının özetini



Cevap

- Cevap: B
- "ls -l" komutu, dosyalar hakkında ayrıntılı bilgileri görüntüler. Bu, dosya adlarının yanı sıra her dosya için izinler, sahiplik, boyut ve değiştirilme zamanı gibi detaylı bilgileri gösterir. Bu komut, kullanıcıya bir dizindeki dosyalar hakkında daha kapsamlı bilgi sunar ve bu bilgiler dosyaları tanımlamak, erişmek veya yönetmek için önemlidir.



Soru

- Dağıtık dosya sistemi kullanmanın temel avantajı nedir?
 - A) Merkezi depolama nedeniyle artan performans
 - B) Dosyaların şifrelenmesi yoluyla artan güvenlik
 - C) Artan güvenilirlik ve hata toleransı
 - D) Kullanıcılar için basitleştirilmiş dosya yönetimi



Cevap

- Cevap: C
- Dağıtık dosya sistemi kullanmanın temel avantajı, artan güvenilirlik ve hata toleransıdır. Bu sistemler, verileri birden fazla sunucu veya depolama cihazı arasında dağıtarak, herhangi bir sunucu veya depolama cihazının arızalanması durumunda bile veriye erişimi sürdürebilirler. Bu, sistemlerin daha yüksek bir çalışma sürekliliği sağlamasına ve veri kaybını en aza indirmesine olanak tanır.



Soru

- Hangi dosya sistemi özelliği, ardışık depolama alanlarını ayırarak parçalanmayı azaltmaya yardımcı olur?
- A) Birleştirme (Defragmentation)
- B) Günlükleme (Journaling)
- C) Blok alt tahsis (Block suballocation)
- D) Ön tahsis (Preallocation)



Cevap

- Cevap: D
- Bu özellik, ön tahsis (preallocation) olarak adlandırılır. Ön tahsis, dosya sistemi tarafından ardışık depolama alanlarının belirli bir dosya için önceden ayrılması işlemidir. Bu, dosyanın büyüdüğü veya değiştiğinde, dosyanın parçalanmasını önler ve ardışık depolama alanları kullanımını sağlar. Bu sayede, performans artar ve dosya sistemi üzerindeki disk parçalanması azalır.



Soru

- Kopya-Üzerine-Yazma (Copy-on-Write - COW) tekniğinin bir dezavantajı nedir?
 - A) Artan disk alanı kullanımı
 - B) Yavaş dosya erişim hızı
 - C) Büyük dosyalara sınırlı destek
 - D) Günlükleme ile uyumsuzluk



Cevap

- Cevap: A
- Kopya-Üzerine-Yazma (Copy-on-Write - COW) tekniđi, bir dosyaya yazma işlemi yapılmadan önce kopyalanarak, orijinal dosyanın deđişmeden kalmasını sağlar. Bu, özellikle bir dosyanın birden çok kullanıcısı veya işlemi olduğunda yararlıdır, çünkü her işlem orijinal dosyayı deđiştirmez, deđişikliklerini kendi kopyasında yapar. Ancak, COW tekniđinin bir dezavantajı artan disk alanı kullanımıdır. Her yazma işlemi sırasında yeni bir kopya oluşturulduğu için, diskte fazladan yer tutabilir. Özellikle büyük dosyalar veya sık sık deđiştirilen dosyalar için, zamanla bu ekstra kopyalar önemli miktarda disk alanını işgal edebilir.



Soru

- Hangi dosya sistemi, ayrıntılı izinler için Erişim Kontrol Listelerini (ACL) destekler?
- A) ext4
- B) HFS+
- C) ZFS
- D) NTFS



Cevap

- Cevap: D
- NTFS (New Technology File System), ayrıntılı izinler sağlamak için Erişim Kontrol Listeleri (ACL'ler) kavramını destekler. ACL'ler, her dosya veya dizin için özelleştirilmiş izinler atamaya olanak tanır. Bu, kullanıcıların, grupların veya diğer varlıkların (örneğin, bir hizmet hesabı) belirli dosyalar veya dizinler üzerindeki erişimini çok daha ayrıntılı bir şekilde kontrol edebilmeyi sağlar.



Soru

- Unix benzeri dosya sistemlerinde inode'un amacı nedir?
 - A) Dosya verilerini saklamak
 - B) Dosya izinlerini yönetmek
 - C) Dosya meta verilerini saklamak
 - D) Dosya erişim zamanını izlemek



Cevap

- Cevap: C
- inode'lar, Unix benzeri dosya sistemlerinde dosyaların meta verilerini saklamak için kullanılan yapısal bir bileşendir. Her dosya veya dizin için bir tür kimlik numarası gibi işlev görür ve dosyanın fiziksel konumu, boyutu, erişim izinleri, sahibi, değiştirme ve erişim zamanları gibi meta verileri içerir. Bu nedenle, inode'lar dosya sistemi tarafından dosya ile ilgili temel bilgilerin depolanmasını ve yönetilmesini sağlar. Dosyanın kendisi değil, dosyanın özellikleri hakkında bilgi içerirler.



Soru

- Aşağıdakilerden hangisi Btrfs dosya sisteminin özelliği değildir?
- A) Kopya-Üzerine-Yazma (copy-on-write)
- B) Anlık görüntü desteği (snapshot)
- C) Şeffaf sıkıştırma (transparent)
- D) Kaplam-tabanlı tahsis (extent based)



Cevap

- Cevap: D
- Btrfs (B-tree file system), modern bir dosya sistemidir ve birçok gelişmiş özelliği içerir. Şeffaf sıkıştırma, dosyaların otomatik olarak sıkıştırılmasını sağlar, böylece disk alanı daha etkin bir şekilde kullanılabilir. Kopya-Üzerine-Yazma (Copy-on-Write - COW) ve anlık görüntü desteği Btrfs dosya sisteminin temel özelliklerindedir. Bu özellikler, dosyaların güvenliği ve yönetimini sağlamak için önemlidir.



Soru

- Dosya sistemleri bağlamında "RAID" kısaltması neyi ifade eder?
- A) Bağımsız Disklerin Yedeklenmiş Dizisi (Redundant Array of Independent Disks)
- B) Hızlı Erişim ve Sızma Tespiti (Rapid Access and Intrusion Detection)
- C) Bellekteki Verilerin Güvenilir Tahsisi (Reliable Allocation of In-memory Data)
- D) Yazmadan Sonra Okuma Atomik Disk (Read-after-Write Atomic Disk)



Cevap

- Cevap: A
- RAID (Redundant Array of Independent Disks), disklerin paralel olarak çalıştığı ve veri depolama ve korumanın geliştirilmiş bir yöntemidir. Bu terim, "Redundant" (Yedekli), "Array" (Dizisi), "Independent" (Bağımsız) ve "Disks" (Diskler) kelimelerinin baş harflerinden oluşur. RAID, bir dizi farklı disk sürücüsünün bir araya getirilmesiyle oluşturulan bir depolama teknolojisidir. Bu teknoloji, veriye yedekleme, yüksek performans, artan veri erişilebilirliği veya her üçünü de bir arada sağlamak için kullanılabilir.



Soru

- Flash tabanlı depolama aygıtları, USB sürücüler ve SD kartlar için tasarlanmış olan dosya sistemi hangisidir?
- A) JFS
- B) F2FS
- C) ReiserFS
- D) XFS



Cevap

- Cevap: B
- F2FS (Flash-Friendly File System), özellikle flash tabanlı depolama aygıtları için tasarlanmıştır. Geleneksel dosya sistemleri, sabit disk sürücülerini gibi dönme tabanlı depolama ortamlarına odaklanmışken, F2FS özellikle flash bellek aygıtlarının özelliklerine uygun tasarlanmıştır. F2FS, flash bellek aygıtlarının performansını artırmak, ömrünü uzatmak ve dosya sistemi üzerindeki yazma işlemlerini daha etkin hale getirmek için optimizasyonlar içerir. Örneğin, yazma işlemlerini azaltmak için veri bloklarını hızlı bir şekilde silebilir ve birçok flash bellek aygıtının yerleşik önbellek ve hata düzeltme özelliklerini etkin bir şekilde kullanabilir.



Soru

- Unix dosya sisteminde "chown" komutunun amacı nedir?
- A) Dosya sahipliğini değiştirmek
- B) Dosya izinlerini değiştirmek
- C) Dosya konumunu değiştirmek
- D) Dosya içeriğini değiştirmek



Cevap

- Cevap: A
- "chown" (change owner) komutu, dosyanın sahipliğini deęiřtirmek için kullanılır. Dosya sahibi, dosyanın tam izinlerine ve dięer kontrol düzeylerine (örneğin, grup izinleri) erişime sahiptir. "chown" komutu, bir kullanıcının bir dosyanın sahibini veya bir dosyanın sahibi olmayan bir kullanıcıya atamak için kullanılabilir.



Soru

- Dosya sistemlerinde "seyrek dosya" (sparse) özelliğinin amacı nedir?
- A) Yalnızca boş olmayan (non-empty) bloklar için depolama alanı ayırmak
- B) Depolama gereksinimlerini azaltmak için dosyaları sıkıştırmak
- C) Güvenlik için dosya içeriğini şifrelemek
- D) Gelecekteki dosya büyümesi için ekstra alan ayırmak



Cevap

- Cevap: A
- "Seyrek dosya" (sparse file) özelliđi, dosya sistemlerinde yalnızca boş olmayan bloklar için depolama alanı tahsis etmenin bir yolunu sağlar. Bu, dosyanın belirli bir boyutta olmasına rağmen, gerçek depolama alanının dosyanın içeriğindeki boşluklara ve dolu olmayan bölümlere göre dinamik olarak deđişmesine olanak tanır. Seyrek dosyalar, genellikle büyük dosyaların saklanması veya veri yedeklemesi gibi durumlarda kullanışlıdır. Örneđin, bir seyrek dosya 1 TB boyutunda olabilir, ancak gerçekte sadece birkaç megabayt veri içerebilir. Bu tür dosyalar, boşlukları dolu bloklar gibi işleme alır, ancak depolama alanı gereksinimlerini minimize eder.



Soru

- Dosya sistemleri bağlamında, "inode" terimi ne anlama gelmektedir?
- A) İndeks Düğümü (index node)
- B) İç Düğüm (internal node)
- C) Bilgi Düğümü (information node)
- D) İndeksli Düğüm (indexed node)



Cevap

- Cevap: A
- "Inode" (Index Node), dosya sistemlerinde dosyaların ve dizinlerin meta verilerini saklamak için kullanılır. Her dosya veya dizin için bir tür kimlik numarası gibi işlev görür ve dosyanın fiziksel konumu, boyutu, erişim izinleri, sahibi, değiştirme ve erişim zamanları gibi meta verileri içerir.



Soru

- Aşağıdakilerden hangisi bir dosyanın birden fazla dosya adıyla erişilmesine izin verir?
- A) Katı bağlantılar (hard links)
- B) Yumuşak bağlantılar (soft links)
- C) Sembolik bağlantılar (symbolic links)
- D) Bağlantı noktaları (junction points)



Cevap

- Cevap: A
- "Kati bağlantılar" (Hard links), bir dosyanın birden fazla dosya adıyla (yol adı) erişilmesine izin veren bir dosya sistemi özelliğidir. Bir dosyanın kati bağlantıları, dosya sistemine ekstra bir dosya adı ekleyerek oluşturulur. Bu dosya adları, orijinal dosya adıyla aynı dosya içeriğine ve aynı disk konumuna işaret eder. Bir dosyanın birden fazla yerde kullanılmasına izin verir ve dosyanın silinebilmesi için tüm bağlantıların kaldırılması gerekir. Bununla birlikte, aynı dosyanın farklı adlar altında farklı bağlantıları olduğundan, bir adı değiştirme veya silme işlemi, diğer adlarla ilişkili dosyaları etkilemez.



Soru

- İşletim sistemlerinde "mount" komutunun amacı nedir?
 - A) Aygıt sürücülerini yüklemek için
 - B) Disk bölümlerini başlatmak için
 - C) Bir dosya sistemini dosya hiyerarşisinde bir dizine bağlamak için
 - D) Sanal bellek alanı oluşturmak için



Cevap

- Cevap: C
- "Mount" komutu, işletim sistemlerinde bir dosya sistemini belirli bir dizine bağlamak için kullanılır. Bu işlem, belirli bir disk bölümündeki dosyaların, işletim sistemi tarafından belirli bir dizin altında erişilebilir hale getirilmesini sağlar. Bu nedenle, "mount" komutu, bir dosya sistemiyle belirli bir dizinin ilişkilendirilmesini gerçekleştirir. Örneğin, bir USB flash sürücüsü bir Linux sisteme takıldığında, bu sürücünün içeriğine erişmek için "mount" komutu kullanmak gerekir. "mount" komutu, USB sürücüsündeki dosya sistemini, önceden belirlenmiş bir dizine (örneğin, "/mnt/usb") bağlayarak, bu dizindeki dosyalara erişim sağlar.



Soru

- Hangi dosya sistemi, dosya sisteminin zamanında (point-in-time) bir kopyasını oluşturmak için "snapshot" özelliğini destekler?
- A) XFS
- B) Btrfs
- C) UFS
- D) JFS



Cevap

- Cevap: B
- "Snapshot" kavramı, bir dosya sisteminin belirli bir anında tüm dosyalarının bir kopyasını oluşturmayı sağlayan bir özelliktir. Bunu yaparak, dosya sisteminin bir "anlık görüntüsü" oluşturulur ve bu görüntü, dosyaların o belirli zamandaki durumunu korur. Bu, dosyaların ilerideki değişikliklerinden etkilenmeyen bir kopya sağlar. Btrfs (B-Tree File System), "snapshot" özelliğini destekler. Btrfs, Linux işletim sistemlerinde yaygın olarak kullanılır. Bu "snapshot" daha sonra geri dönebilmek veya yedekleme amacıyla kullanılabilir.



Soru

- Geleneksel dosya sistemlerine kıyasla Günlüklü Yapılandırılmış Dosya Sistemi'nin (LFS) başlıca avantajı nedir?
- A) Daha hızlı dosya erişim hızı
- B) Veri bozulma riskinin azalması
- C) Geliştirilmiş disk alanı kullanımı
- D) Büyük dosyalar için daha iyi destek



Cevap

- Cevap: B
- Günlüklü Yapılandırılmış Dosya Sistemi (LFS), geleneksel dosya sistemlerine kıyasla veri güvenliğini artırır. LFS'nin avantajı, veri bütünlüğünü korumak ve veri kaybını en aza indirmek için günlükleme yöntemlerine dayanmasıdır. Geleneksel dosya sistemlerinde, bir dosya üzerinde yapılan değişiklikler doğrudan dosya üzerine yazılırken veri bütünlüğünü tehlikeye atabilir. Örneğin, bir kesinti veya sistem çökmesi sırasında verinin kaybolma, bozulma riski vardır. Ancak LFS'de, değişiklikler önce bir günlük dosyasına (log) yazılır ve sonra ana depolama alanına taşınır. Bu, veri güvenliği açısından daha güvenli bir yaklaşım sağlar, çünkü dosya değişiklikleri günlükte saklanır.



Soru

- Unix benzeri işletim sistemlerinde "truncate" komutunun amacı nedir?
 - A) Yeni bir dosya oluşturmak
 - B) Bir dosyayı kaldırmak
 - C) Bir dosyanın boyutunu daraltmak veya genişletmek
 - D) Dosya izinlerini değiştirmek



Cevap

- Cevap: C
- "Truncate" komutu, bir dosyanın boyutunu deęiřtirmek için kullanılır. Bu komut, bir dosyanın boyutunu belirli bir boyuta kesmek veya bir dosyayı belirli bir boyutta genişletmek için kullanılır. Dosya boyutu genişletildiğinde, eklenen bölümler genellikle sıfırlanır veya boş deęerlerle doldurulur. Örneęin, bir dosyanın içerięi silinmeden boyutunu daraltmak için "truncate" komutu kullanılabilir. Ayrıca, bir dosyanın boyutunu genişletmek ve daha fazla veri eklemek gerektiğinde de kullanılabilir.



Soru

- Veri yedekleme ve hata toleransı sağlayan dosya sistem özelliği hangisidir?
- A) Günlükleme
- B) RAID
- C) ACL'ler
- D) Anlık Görüntüler



Cevap

- Cevap: B
- RAID (Redundant Array of Independent Disks), dosya sistemlerinde veri yedekleme ve hata tolere edebilirlik sağlar. RAID, birden fazla disk sürücüsünü bir araya getirerek veri parçalarını depolar ve bu parçaları farklı disk sürücülerine dağıtarak veri artıklığı oluşturur. RAID'de kullanılan bir yöntem, parite bilgisi kullanarak veri artıklığı sağlamaktır. Parite, RAID dizisi içindeki veri blokları arasında bir tür kontrol verisidir. Bu parite bilgisi, diğer veri bloklarından elde edilir ve veri bozulması durumunda orijinal veriyi geri yüklemek için kullanılır.



Soru

- ZFS sisteminin Kopya-üzerine-Yazma (Copy-on-Write) mekanizmasının başlıca avantajı nedir?
- A) Okuma performansının artması
- B) Yazma genişlemesinin (amplification) azalması
- C) Düşük depolama maliyeti
- D) Veri bütünlüğünün artması



Cevap

- Cevap: D
- ZFS (Zettabyte File System), veri bütünlüğünü sağlamak ve veri kaybını önlemek için güçlü bir dosya sistemidir. Bu özelliğin temelinde Kopya-üzerine-Yazma (COW) mekanizması bulunur. ZFS'de bir dosyanın üzerine yazıldığında, öncelikle ilgili blokların kopyaları alınır ve yeni veri bu kopyalar üzerine yazılır. Eski veri blokları değiştirilmez ve bir kopya oluşturulur. Bu, veri bütünlüğünü artırır çünkü orijinal veriye dokunulmadan yeni veri yazılabilir. Bu mekanizma aynı zamanda veri güvenliği ve geri dönüşü sağlar. Yazma genişlemesinin azalması genellikle SSD'lerde kullanılan bir terimdir.



Soru

- Unix benzeri bir dosya sisteminde "inode tablosu"nun başlıca amacı nedir?
- A) Dosya sistemindeki her dosya hakkında meta verileri depolamak
- B) Kurtarma için dosya sistemi değişikliklerinin günlüğünü tutmak
- C) Dosya isimlerini bunların karşılık gelen disk adreslerine eşlemek
- D) Dosya depolama için disk alanı tahsis etmek ve yönetmek



Cevap

- Cevap: A
- Inode tablosu, her dosya ve dizin için ayrıntılı meta verilerin (metadata) depolandığı bir veri yapısıdır. Bu meta veriler dosyanın adını, boyutunu, sahibini, izinlerini, oluşturma ve değiştirme tarihlerini ve dosyanın fiziksel konumunu içerir. Her dosya ve dizin için bir inode girdisi bulunur ve bu girdi o dosya veya dizinin tüm özelliklerini tanımlar. Dosya sistemine eklenen her dosya veya dizin için bir inode oluşturulur ve bu inode tablosuna eklenir.



Soru

- Dosya sistemlerinde "süper bloğun" (superblock) amacı nedir?
- A) Dosya sisteminin tamamı hakkında boyut ve durum gibi meta verileri depolamak
- B) Tahsis edilebilecek boş disk bloklarının bir listesini tutmak
- C) Günlüklenme amaçları için dosyalarda yapılan değişiklikleri izlemek
- D) Küçük dosyaların içeriğini doğrudan süper blok içinde depolamak



Cevap

- Cevap: A
- Dosya sistemi süper bloğu, dosya sisteminin tamamıyla ilgili meta verileri depolayan bir veri yapısıdır. Bu meta veriler dosya sisteminin boyutunu, kullanılan ve kullanılmayan alanları, dosya sisteminin durumunu (örneğin, bağlı veya bağlı olmayan) ve dosya sisteminin diğer özelliklerini içerir. Dosya sisteminin başlangıcında bulunur ve genellikle dosya sistemi tarafından kullanılan sabit bir konumda bulunur. Dosya sistemi başlatıldığında veya bağlandığında, işletim sistemi bu süper bloğa erişir ve dosya sistemi hakkındaki önemli bilgileri okur.



Soru

- Microsoft tarafından sunulan ve büyük miktarda veri ve akan medya (streaming) dosyalarını depolamak için optimize edilen dosya sistemi hangisidir?
- A) NTFS
- B) FAT32
- C) exFAT
- D) ReFS



Cevap

- Cevap: D
- ReFS (Resilient File System), Microsoft'un sunucu ve depolama çözümleri için geliştirdiği dosya sistemidir. Büyük miktarda veri ve akan medya dosyalarını depolamak için optimize edilmiştir. ReFS, geleneksel dosya sistemlerine göre daha sağlam ve dayanıklı bir yapı sunar. Veri bütünlüğünü korumak için çeşitli özellikler içerir ve büyük veri depolama gereksinimlerini karşılamak için tasarlanmıştır. Özellikle, büyük dosya ve veri kümeleriyle çalışırken, performansı artırır ve güvenilirliği sağlar. ReFS, yüksek performans ve güvenilirlik sağlamak için tasarlanmıştır, bu nedenle büyük veri merkezlerinde, bulut depolama sistemlerinde ve medya yayıncılığı gibi alanlarda tercih edilir.



Soru

- Hangi Linux dosya sistemi anlık görüntüleme (snapshotting), sıkıştırma ve RAID işlevselliğini yerleşik olarak destekler?
- A) ext2
- B) ext4
- C) Btrfs
- D) XFS



Cevap

- Cevap: C
- Btrfs (B-tree File System), Linux için geliştirilmiş bir dosya sistemidir ve snapshotting, sıkıştırma ve yerleşik RAID işlevselliği gibi modern özellikleri destekler. Btrfs, dosya sistemlerinin güncel ihtiyaçlarına uyum sağlamak amacıyla tasarlanmıştır. Anlık görüntüleme (snapshotting) özelliği, dosya sistemini belirli bir zamandaki "anlık görüntüsünü" oluşturarak dosyaların durumunu koruma ve geri dönüş yapma imkanı sağlar. Sıkıştırma, depolama alanını daha verimli bir şekilde kullanmak ve diskteki veri miktarını azaltmak için kullanılır. Yerleşik RAID işlevselliği, disk arızalarına karşı yüksek düzeyde veri koruması sağlamak için RAID düzeylerini doğrudan dosya sistemi seviyesinde yönetme olanağı sağlar.



Soru

- ReiserFS dosya sisteminin geleneksel Linux dosya sistemleri (örneğin ext4) karşısındaki başlıca avantajı nedir?
 - A) Büyük dosyalar için geliştirilmiş destek
 - B) Parçalanmaya karşı direnç
 - C) Büyük depolama dizileri için artırılmış ölçeklenebilirlik
 - D) Rastgele G/Ç işlemleri için daha yüksek performans



Cevap

- Cevap: D
- ReiserFS, özellikle yüksek performanslı ve rastgele G/Ç işlemleri için optimize edilmiş bir Linux dosya sistemidir. Bu nedenle, ReiserFS'nin geleneksel Linux dosya sistemlerine göre en önemli avantajı, rastgele G/Ç işlemleri için daha yüksek performans sağlamasıdır. ReiserFS, dosya sistemi verilerini depolamak için daha verimli bir B-ağacı (B-tree) yapısı kullanır. Bu yapının bir sonucu olarak, büyük dosyaları işlemek ve rastgele G/Ç işlemlerini daha hızlı gerçekleştirmek için uygun hale gelir. Bu özellik, veritabanları, web sunucuları ve yüksek performanslı uygulamalarda önemlidir.



Soru

- Hangi Linux dosya sisteminde veri ve meta verilerin veri bütünlüğünü sağlamak için yerleşik sağlama toplamı (checksum) bulunur?
- A) ext4
- B) XFS
- C) Btrfs
- D) JFS



Cevap

- Cevap: C
- Btrfs (B-tree File System), Linux için geliştirilmiş modern bir dosya sistemidir ve veri bütünlüğünü sağlamak için yerleşik sağlama toplama (checksums) kullanır. Sağlama toplama, dosya sistemi içindeki verilerin ve meta verilerin doğruluğunu kontrol etmek için kullanılan özet değerlerdir. Btrfs, her veri ve meta veri bloğu için bir değer hesaplar ve bu değerlerin depolama alanına yazılmasını sağlar. Bu sayede, dosya sistemi herhangi bir veri bütünlüğü hatası algıladığında, bu hatayı otomatik olarak düzeltebilir veya etkilenen veriyi doğru bir kopyadan geri yükleyebilir.



Soru

- Linux'ta, hızlı günlükleme (journaling) ve kurtarma yetenekleri nedeniyle genellikle kök (/) bölümleri için hangi dosya sistemi kullanılır?
- A) ext3
- B) XFS
- C) Btrfs
- D) ReiserFS



Cevap

- Cevap: A
- ext3, Linux'ta kullanılan bir dosya sistemidir ve özellikle kök (/) bölümleri için tercih edilir. Bu dosya sistemi, hızlı günlükleme (journaling) ve kurtarma yetenekleri sağlar. Günlükleme (journaling), dosya sistemi üzerindeki değişikliklerin bir günlük dosyasına kaydedilmesini içeren bir tekniktir. Bu sayede, beklenmedik sistem kapanmaları veya çökme durumlarında dosya sistemi bütünlüğü korunur ve veri kaybı önlenir. ext3, bu günlükleme tekniklerini kullanarak dosya sistemi üzerindeki veri kaybını en aza indirir ve dosya sistemi bütünlüğünü sağlar. Kök (/) bölümü, işletim sisteminin temel dosya sistemi ve sistem dosyalarının bulunduğu bölümdür. Bu nedenle, kök bölümü için güvenilir ve hızlı bir dosya sistemi tercih edilir.



Soru

- HFS+ dosya sisteminin FAT32 dosya sistemine kıyasla avantajı nedir?
- A) Büyük dosya boyutları için daha iyi destek
- B) Windows sistemleriyle geliştirilmiş uyumluluk
- C) Veri depolama için artırılmış güvenilirlik
- D) Günlüklenme için yerel destek



Cevap

- Cevap: D
- HFS+ dosya sisteminin FAT32'ye göre avantajı, günlükleme (journaling) için yerel destek sağlamasıdır. HFS+ dosya sistemi, günlükleme özelliğini destekler. Günlükleme özelliği, dosya sisteminin bütünlüğünü korur ve veri kaybını azaltır. Bazı durumlarda, günlükleme işlemi dosya sistemi performansını azaltabilir, ancak genel olarak, bu küçük bir etkidir ve veri bütünlüğünün sağlanması göz önüne alındığında avantajları ağır basar. FAT32 dosya sistemi, günlükleme özelliğini desteklemez. Bu nedenle, FAT32'de dosya sistemi bütünlüğünü sağlamak daha zordur.



Soru

- Extended File Allocation Table (exFAT) dosya sistemi uzantısının temel amacı nedir?
 - A) Dosya sıkıştırması için artırılmış destek
 - B) Windows ve büyük depolama cihazları ile geliştirilmiş uyumluluk
 - C) Veri güvenliği için yerleşik şifreleme desteği
 - D) Gelişmiş veri yinelenme teknikleri



Cevap

- Cevap: B
- Extended File Allocation Table (exFAT) dosya sistemi uzantısının amacı, Windows ve büyük depolama aygıtları ile uyumluluğunun artırılmasıdır. exFAT, Windows işletim sistemine daha iyi uyumluluk sağlamak için geliştirilmiştir. FAT32 gibi, exFAT de dosya ve depolama aygıtları arasında dosya transferini kolaylaştırır. Ayrıca, FAT32'nin aksine, exFAT dosya boyutu sınırlarını genişletir. exFAT, Windows yanı sıra diğer işletim sistemleriyle de uyumludur. Bu, aynı depolama cihazının farklı işletim sistemlerinde kullanılabilmesini sağlar. FAT32, tek bir dosya için 4 GB'lık boyut sınırına sahiptir. Bu, büyük medya dosyaları için uygun olmayabilir. exFAT bu sınırları genişleterek, dosyanın daha büyük olmasına izin verir. exFAT, FAT32'den daha modern bir dosya sistemi olarak kabul edilir.



Soru

- Dosya sistemleri için Şeffaf Sıkıştırma Uzantısının (TCE) temel özelliği nedir?
- A) Sıkıştırılmış dosyaların otomatik şifrelenmesi
- B) Depolama alanını azaltmak için dosya verilerinin gerçek zamanlı sıkıştırılması
- C) Sıkıştırılmış dosyaların anlık olarak açılmasını destekleme
- D) Sıkıştırılmış dosya yedeklemesi için bulut depolama hizmetleriyle entegrasyon



Cevap

- Cevap: B
- Şeffaf Sıkıştırma Uzantısı (TCE) dosya sistemlerinde gerçek zamanlı sıkıştırma özelliğine sahip olmasıdır. Dosyaları saklarken veya aktarırken otomatik olarak sıkıştırır. Bu, depolama alanından tasarruf sağlar çünkü sıkıştırılmış dosyalar daha az yer kaplar. Gerçek zamanlı sıkıştırma, dosya sistemlerinin performansını etkileyebilir. Ancak, modern sıkıştırma algoritmaları genellikle bu etkiyi minimumda tutar ve sıkıştırma işlemlerini arka planda yürütür. TCE, dosyaları sıkıştırırken veri bütünlüğünü korur. Bu, sıkıştırılmış dosyaların açılması veya kullanılması sırasında veri kaybını önler. Özellikle büyük ölçekli sistemlerde, depolama alanı sıkıştırma ile önemli ölçüde tasarruf edilebilir.



Soru

- Günlük Yapılı (log structured) Dosya Sistemi (LSFS) uzantısı, depolama optimizasyonu açısından ne sunar?
- A) Disk bloklarının otomatik olarak birleştirilmesi
- B) Depolama tahsisi sürekli izlenmesi ve ayarlanması
- C) Dosya değişikliklerinin ardışık günlüklenmesi ile yazma performansının artırılması
- D) Disk alanı kullanımını azaltmak için dosya verilerinin şeffaf sıkıştırılması



Cevap

- Cevap: C
- LSFS, dosya sistemi üzerinde yapılan değişiklikleri ardışık olarak günlükler. Bu, dosyaların değişikliklerinin sıralı olarak kaydedilmesi anlamına gelir. Sıralı olarak kaydedilen dosya değişiklikleri, diske yazma işlemlerini optimize eder ve disk erişimini iyileştirir. Özellikle sık sık dosya yazma işlemlerinin gerçekleştiği uygulamalarda, LSFS'in bu özelliği performansı artırır. LSFS, silinen dosyaların yerini geri kazanmak için disk boşaltma işlemlerini yönetir. Bu, diskte boş alanın etkin bir şekilde kullanılmasını sağlar ve performansı artırır. Ardışık günlüklenme, dosyaların diske ardışık olarak yazılmasını sağladığından, parçalanmayı azaltır. Bu da okuma işlemlerini hızlandırır.



Soru

- Dağıtık Dosya Sistemi (DFS) uzantısının birincil özelliği nedir?
- A) Veri bütünlüğünün artırılması için merkezi depolama yönetimi
- B) Depolama düğümleri arasında dosyaların şeffaf olarak kopyalanması
- C) Dosya düzeyinde şifreleme ve erişim kontrolü için geliştirilmiş destek
- D) Dağıtık ağlar üzerinde dosya değişikliklerinin gerçek zamanlı senkronizasyonu



Cevap

- Cevap: B
- DFS, dosyaların birden çok depolama düğümü arasında şeffaf bir şekilde kopyalanmasını sağlar. Bir dosyanın tek bir depolama düğümünde bulunmasına gerek yoktur. DFS, dosyaları farklı düğümler arasında otomatik kopyalar ve kullanıcılar için bu işlemi şeffaf hale getirir. Kullanıcılar, dosyalarını tek bir merkezi depolama noktasına yüklemek zorunda kalmaz, bu da veri erişimini ve yönetimini kolaylaştırır. Bir depolama düğümünde arıza olursa, kullanıcılar dosyalara diğer düğümlerden erişebilirler. Bu, veri kaybını önler ve hizmet kesintilerini azaltır. DFS, dosya erişimini dağıtarak performansı artırır. Aynı anda birden çok kullanıcı tarafından erişilen dosyalar için yükü dengeleyebilir. DFS, yeni depolama düğümlerinin kolayca eklenmesine olanak tanır.



Soru

- Windows'ta ".exe" uzantısı ile ilişkilendirilen dosya türü nedir?
 - A) Metin dosyası
 - B) Yürütülebilir dosya
 - C) Görüntü dosyası
 - D) Ses dosyası



Cevap

- Cevap: B
- ".exe" uzantısı yürütülebilir dosyalarla ilişkilendirilir. Bilgisayar tarafından doğrudan çalıştırılabilir dosyalardır. Bir ".exe" dosyası çift tıklandığında, ilgili program başlatılır. Örneğin, tarayıcılar, ofis programları, oyunlar ".exe" dosyaları aracılığıyla çalıştırılır. ".exe" dosyaları, zararlı veya kötü amaçlı yazılım tarafından da kullanılabilir. Bu nedenle, ".exe" dosyalarını açarken veya çalıştırırken dikkatli olmak önemlidir ve güvenilir kaynaklardan veya güvenlik yazılımı tarafından onaylanmış dosyaları çalıştırmak önemlidir.



Soru

- Unix tabanlı sistemlerde sıkıştırılmış arşiv dosyasını gösteren dosya uzantısı hangisidir?
- A) .zip
- B) .tar.gz
- C) .iso
- D) .rar



Cevap

- Cevap: B
- ".tar.gz" uzantısı, Unix tabanlı sistemlerde sıkıştırılmış arşiv dosyasını temsil eder. ".tar.gz" uzantısı, iki farklı sıkıştırma işlemi olan "tar" ve "gzip" tarafından oluşturulan bir arşiv dosyasını belirtir. "tar" komutu, dosyaları bir araya getirir ve tek bir dosya veya klasör halinde bir arşiv oluşturur. Bu arşiv dosyası daha sonra "gzip" sıkıştırma programıyla sıkıştırılır, bu da ".tar.gz" uzantılı sıkıştırılmış bir arşiv dosyasını oluşturur.



Soru

- ".mp3" uzantısıyla tanımlanan dosya türü nedir?
- A) Görüntü dosyası
- B) Video dosyası
- C) Ses dosyası
- D) Belge dosyası



Cevap

- Cevap: C
- ".mp3" uzantılı dosyalar, ses içeriğini depolayan dosyalardır. Ses verilerini sıkıştırarak depolayan bir formatı temsil eder. MP3 formatı, ses verilerini sıkıştırırken kaliteyi koruyarak dosya boyutunu küçültür. Bu, daha küçük dosya boyutlarıyla yüksek kalitede ses kaydı sağlar ve internet üzerinden ses içeriğinin paylaşılmasını ve aktarılmasını kolaylaştırır.



Soru

- ".pdf" uzantısı hangi dosya türünü temsil eder?
- A) Görüntü dosyası
- B) Ses dosyası
- C) Belge dosyası
- D) Video dosyası



Cevap

- Cevap: C
- ".pdf" uzantılı dosyalar, Portable Document Format (Taşınabilir Belge Biçimi) adı verilen dosya biçimini kullanarak belgeleri depolar. Bu dosya biçimi, belgelerin farklı bilgisayarlar ve işletim sistemleri arasında tutarlı bir şekilde görüntülenmesini sağlar. PDF dosyaları metin, görüntü, grafikler, tablolar ve diğer içerikleri bir araya getirebilir. Bu nedenle, PDF formatı, broşürler, raporlar, kitaplar, sunumlar ve diğer birçok belge türü için yaygın olarak kullanılır. PDF formatı, belgelerin taşınabilirliğini sağlar. Farklı cihazlarda ve platformlarda açılabilir ve görüntülenebilir. Belgenin orijinal biçimini korur. Bir belge PDF olarak kaydedildiğinde, içeriği orijinal biçiminde korunur ve başkaları tarafından aynı şekilde görüntülenir.



Soru

- Microsoft Excel'de elektronik tablo dosyaları için hangi dosya uzantısı kullanılır?
- A) .xls
- B) .docx
- C) .ppt
- D) .txt



Cevap

- Cevap: A
- ".xls" uzantısı, Microsoft Excel'in eski sürümlerinde kullanılan standart elektronik tablo dosya biçimidir. ".xls" dosyaları, elektronik tablo verilerini depolar. Bu dosyalar, tabloları, grafikleri, formülleri ve diğer verileri içerebilir. Microsoft Excel 2007'den itibaren, ".xlsx" uzantısı XML tabanlı Office Open XML (OOXML) biçimine geçiş yapmıştır. Bu, daha iyi sıkıştırma, daha iyi veri bütünlüğü sağlar.



Soru

- ".avi" uzantısı hangi dosya türünü belirtir?
- A) Ses dosyası
- B) Video dosyası
- C) Görüntü dosyası
- D) Arşiv dosyası



Cevap

- Cevap: B
- AVI (Audio Video Interleave), ses ve görüntü içeriğini aynı dosya içinde birleştirir. Bu, dosyanın hem sesi hem de görüntüyü içerebileceği anlamına gelir. Dolayısıyla, ".avi" dosyaları genellikle filmler, video klipler, animasyonlar ve diğer video içeriklerini depolamak için kullanılır. AVI formatı, ses ve video kalitesini korurken dosya boyutunu yönetmek için farklı sıkıştırma yöntemlerini destekler.



Soru

- Linux'ta genellikle bir kabuk betik (shell script) dosyasını gösteren hangi dosya uzantısıdır?
- A) .sh
- B) .exe
- C) .bat
- D) .cmd



Cevap

- Cevap: A
- Kabuk betikleri, kabuk (shell) tarafından yürütülen komut dizilerini içeren metin dosyalarıdır. Bu betik dosyaları, sistem yönetimi, otomasyon ve diğer görevler için kullanılır. ".sh" uzantısı, bir dosyanın bir kabuk betiği olduğunu belirtir. Metin dosyası biçimindedir ve bir dizi kabuk komutunu içerir. Uygun izinlerle işaretlendiğinde doğrudan yürütülebilir hale gelir. Bu, kullanıcının dosyayı çalıştırmasını ve betiğin içindeki komutların gerçekleştirilmesini sağlar. ".sh" uzantılı dosyalar, farklı kabuklarda (örneğin, Bash, Zsh, KornShell) çalışabilir.



Soru

- ".jpg" uzantısı hangi dosya türünü temsil eder?
- A) Ses dosyası
- B) Video dosyası
- C) Görüntü dosyası
- D) Belge dosyası



Cevap

- Cevap: C
- ".jpg" uzantılı dosyalar, dijital görüntüleri depolamak için yaygın olarak kullanılan bir dosya biçimidir. Bu dosyalar, fotoğraflar, çizimler, grafikler ve diğer görsel içerikleri saklamak için kullanılır. JPEG (Joint Photographic Experts Group) olarak bilinen ".jpg" formatı, sıkıştırılmış bir görüntü biçimidir. Bu sıkıştırma, görüntünün dosya boyutunu küçültürken, görsel kaliteyi korumaya çalışır. Farklı renk derinliklerini ve kalite seviyelerini destekler. JPEG formatı, geniş bir kullanıcı tabanı tarafından desteklenir ve birçok cihaz ve platformda yaygın olarak kullanılır.



Soru

- Bloklamalı G/Ç (I/O) hakkında hangisi doğrudur?
 - A) Birden fazla G/Ç işleminin aynı anda işlenmesine izin verir.
 - B) G/Ç işlemi tamamlanana kadar çağırılan işlemi bekletir.
 - C) CPU G/Ç beklerken asla kesintiye uğramaz.
 - D) Bloklama olmayan G/Ç olarak da bilinir.



Cevap

- Cevap: B
- Bloklamalı G/Ç (I/O), işlemcinin bir G/Ç işlemi tamamlanana kadar beklemesine neden olan bir G/Ç yöntemidir. Örneğin, dosya okuma veya yazma işlemi sırasında, işlemci dosya işlemini tamamlamadan diğer işlemlere devam etmez. Bloklamalı G/Ç, işlemci zamanının kaybına neden olabilir. Bloklamalı G/Ç işlemi sıralı olarak gerçekleşir. Yani, bir G/Ç işlemi başlatıldığında, diğer işlemler sırayla beklemek zorunda kalır. Bloklamalı G/Ç, tek iş parçacıklı (single-threaded) veya tek işlemcili sistemlerde verimlilik kaybına neden olabilir.



Soru

- Aşağıdakilerden hangisi Doğrudan Bellek Erişimi'nin (DMA) özelliklerinden değildir?
- A) Veri transferi sırasında CPU üzerindeki yükü azaltır.
- B) Her veri transferi için CPU müdahalesini gerektirir.
- C) Çevre birimlerinin, CPU'yu dahil etmeden doğrudan belleğe/veya bellekten veri transfer etmesine izin verir.
- D) CPU'ya G/Ç görevlerini aktararak genel sistem performansını artırır.



Cevap

- Cevap: B
- DMA, veri transferi işlemlerinde CPU'nun rolünü azaltır. Geleneksel olarak, veri transferi işlemleri, CPU tarafından doğrudan yönlendirilir ve bu CPU'nun kaynaklarını kullanır. Ancak DMA kullanılarak, bu işlemler CPU müdahalesi olmadan gerçekleştirilebilir, bu da CPU'nun diğer görevlere odaklanmasına olanak tanır. DMA, her veri transferi için CPU müdahalesini gerektirmez. Bunun yerine, DMA denetleyicisi, veri transferini doğrudan yürütür ve CPU'ya sadece gerekli olduğunda bilgi verir. DMA, çevre birimlerinin (örneğin, disk sürücüler, ağ kartları) doğrudan belleğe veya bellekten veri transfer etmesine izin verir.



Soru

- Aşağıdakilerden hangisi Asenkron G/Ç'nin bir özelliğidir?
- A) G/Ç işlemi tamamlanana kadar çağırın süreci bloke eder.
- B) İşlemin tamamlandığının kontrolü için çağırın süreç tarafından sorgulama gerektirir.
- C) Senkronize G/Ç olarak da bilinir.
- D) G/Ç işlemlerinin hemen tamamlanmasını garanti eder.



Cevap

- Cevap: B
- Asenkron G/Ç, bir işlem sırasında G/Ç işlemlerinin arka planda tamamlanmasına olanak tanıyan bir yöntemidir. G/Ç işlemi başlatıldıktan sonra işlemci beklemek zorunda kalmaz. İşlemi başlatan süreç, G/Ç işleminin tamamlandığını kontrol etmek için açık bir sorgulama yapar. Süreçlerin G/Ç işlemlerini başlatırken diğer süreçlerle aynı anda devam etmelerine olanak tanır. Bu, sistemdeki kaynakların daha etkili bir şekilde kullanılmasını sağlar.



Soru

- İşletim sistemi içinde aygıt sürücüsünün amacı nedir?
 - A) Aygıtları yönetmek için grafiksel kullanıcı arayüzü sağlamak.
 - B) Uygulama taleplerini donanım aygıtları tarafından anlaşılabilir komutlara çevirmek.
 - C) G/Ç işlemleri için sistem kaynaklarını tahsis etmek.
 - D) G/Ç yoğun işlemler için CPU çizelgelemesini optimize etmek.



Cevap

- Cevap: B
- Aygıt sürücüsü, uygulamaların aygıtlarla iletişim kurmak için kullandığı yüksek düzeyli komutları, donanım aygıtları tarafından anlaşılabilir düşük düzeyli komutlara dönüştürür. Bu, uygulamaların donanım aygıtlarına erişmesini ve onlarla etkileşim kurmasını sağlar. Donanım aygıtlarının doğru şekilde çalışmasını sağlamak için donanım kaynaklarını yönetir. Donanım aygıtlarının performansını artırmak için optimize edilmiş sürücü kodu sağlar. Donanım cihazlarında oluşabilecek hataları algılar ve işletim sistemi veya kullanıcıya bu hataları bildirir. İşletim sistemi ve aygıtlar arasında uyumluluğu sağlar.



Soru

- Unix benzeri işletim sistemlerinde "select" sistem çağrısının amacı nedir?
 - A) Kullanıcıdan girdi okumak.
 - B) Bir veya daha fazla G/Ç işleminin tamamlanmasını beklemek.
 - C) Bir komutu arka planda çalıştırmak.
 - D) Yeni bir süreç oluşturmak.



Cevap

- Cevap: B
- "select" sistem çağrısı, birden fazla dosya tanımlayıcısı ve bir zaman aşımı değerini alarak, dosyalar üzerindeki G/Ç işlemlerinin tamamlanmasını bekler. Bu, tek bir işlemde birden fazla dosyadan gelen verileri izlemeyi ve işlemeyi mümkün kılar. Birden fazla G/Ç işlemi arasında geçiş yapılmasını ve verimliliğin artırılmasını sağlar. Bir sunucu, birden fazla istemciden gelen bağlantıları kabul etmek için "select" çağrısını kullanabilir. Bir zaman aşımı değeri belirterek, belirli bir süre içinde G/Ç işlemlerinin tamamlanmasını bekler. Sistemdeki kaynakların verimli bir şekilde kullanılmasını sağlar.



Soru

- Dosya G/Ç işleminde "seek" işleminin önemi nedir?
 - A) Dosya içindeki mevcut konumu alır.
 - B) Dosya işaretçisini dosya içinde belirtilen bir konuma taşır.
 - C) Dosya önbelleğinin içeriğini diske boşaltır.
 - D) G/Ç işlemi tamamlandıktan sonra dosyayı kapatır.



Cevap

- Cevap: B
- Her dosya, dosyanın içindeki belirli bir konumu izlemek için bir dosya işaretçisine sahiptir. "seek" işlemi, bu dosya işaretçisini belirli bir konuma taşır. Bu, dosya içinde gezinmeyi ve istenen verilere erişmeyi sağlar. Dosya işaretçisinin taşınması, dosya içinde belirli bir konuma gidilmesini sağlar. Bu, dosya içinde arama yapmak veya belirli bir veri bloğuna erişmek için gereklidir. Örneğin, bir dosyada belirli bir yerden okumak veya yazmak için dosya işaretçisinin konumlandırılması gerekir. Dosya işaretçisinin belirli bir konuma taşınması, dosya içindeki verilere daha hızlı ve daha etkili bir şekilde erişimi mümkün kılar.



Soru

- Hangi G/Ç çizelgeleme algoritması, G/Ç isteklerinin tamamlanmasını beklemek için harcanan toplam zamanı en aza indirmeyi amaçlar?
- A) İlk Gelen, İlk Hizmet (FCFS)
- B) En Kısa Arama Zamanı Önce (SSTF)
- C) Round Robin
- D) İlk Giren, İlk Çıkan (FIFO)



Cevap

- Cevap: B
- SSTF algoritması, mevcut başlık konumuna en yakın veri bloğuna erişmek için bir sonraki G/Ç isteğini seçer. Bu, başlık konumuna hareket süresini azaltarak G/Ç isteklerinin hizmet edilmesi için geçen toplam zamanı azaltır. SSTF, mevcut disk başlık konumunu optimize ederken aynı zamanda işlemci kullanımını maksimize etmeyi amaçlar. Adil bir G/Ç hizmeti dağılımı sağlar ve tüm G/Ç isteklerinin hızlı bir şekilde hizmet edilmesini sağlar.



Soru

- İşletim sisteminde tampon önbelleğinin (buffer cache) amacı nedir?
 - A) İkincil depolama aygıtlarından sık erişilen verileri saklamak.
 - B) Sıkça yürütülen süreçler için talimatları önbelleğe almak.
 - C) Aygıt sürücülerini için bellek tahsisini yönetmek.
 - D) G/Ç yoğun süreçler için CPU çizelgelemesini optimize etmek.



Cevap

- Cevap: A
- Tampon önbelleği, sıkça erişilen verileri hızlı bir şekilde erişilebilir konuma getirerek, veri erişim hızını artırır. Sıkça erişilen verilerin önbelleğe alınması, disk erişimini azaltır. Çünkü veriler önbellekte bulunur ve doğrudan diskten okunmak yerine önbellekten alınır. Önbellekten veri alındığında, işletim sistemi bu verinin kopyasını diskteki orijinal konumundan alır. Bu, veri bütünlüğünü korur ve veri kaybını önler. Birden fazla işlemcinin veya sürecin aynı verilere erişmesini kolaylaştırır. Bu, veri paylaşımını ve işbirliğini artırır.



Soru

- Aşağıdakilerden hangisi geçerli bir G/Ç aygıtı kategorisi değildir?
 - A) Blok tabanlı aygıtlar
 - B) Karakter tabanlı aygıtlar
 - C) Ağ aygıtları
 - D) Sıralı aygıtlar



Cevap

- Cevap: D
- Blok tabanlı aygıtlar, veri depolama ve erişimini bloklar halinde yapar. Örneğin, sabit diskler ve SSD'ler blok aygıtlarına örnektir. Karakter tabanlı aygıtlar, karakterlerin veya baytların seri bir şekilde okunup yazılmasını sağlar. Örneğin, klavye ve fare gibi giriş aygıtları ve yazıcılar gibi çıkış aygıtları karakter tabanlı aygıtlara örnektir. Ağ aygıtları, bilgisayarlar arasında veri iletişimini sağlar. Örneğin, ağ arabirimi kartları (NIC'ler), router'lar, switch'ler ve modemler ağ aygıtlarına örnektir. Bu aygıtlar, ağ üzerinde iletişim kurmak için veri paketlerini kullanır.



Soru

- G/Ç işlemlerinde bir aygıt denetleyicisinin temel amacı nedir?
- A) CPU'dan gelen yüksek seviyeli komutları yorumlamak ve bunları aygıtta çalıştırmak.
- B) CPU ile aygıt arasındaki fiziksel bağlantıyı yönetmek.
- C) CPU ve aygıt tarafından kullanılan farklı temsiller arasında veri dönüşümü yapmak.
- D) Aygıt ile etkileşimde bulunmak için bir kullanıcı arayüzü sağlamak.



Cevap

- Cevap: A
- Aygıt denetleyicisi, CPU tarafından gönderilen komutları alır ve bu komutları aygıtta gerçekleştirir. Örneğin, dosya okuma veya yazma komutları gibi işlemler, aygıt denetleyicisi tarafından yönetilir ve gerçekleştirilir. Aygıt denetleyicisi, ilgili aygıtın durumunu ve yapılandırmasını yönetir. Bu, aygıtın etkin bir şekilde kullanılmasını sağlar ve uygun işlem için hazır durumda olmasını sağlar. CPU ve aygıt arasında veri aktarımını yönetir. Aygıtta oluşabilecek hataları algılar ve bunları işler. Bu, hataların raporlanmasını ve giderilmesini sağlar, böylece sistem kararlılığı ve güvenilirliği artar.



Soru

- Düzenli G/Ç işlemlerine göre scatter-gather G/Ç'nin başlıca avantajı nedir?
- A) Verilerin aynı anda birden fazla aygıt arasında aktarılmasına izin verir.
- B) Büyük veri transferlerini yönetme üzerindeki iş yükünü, ayrı önbellekler kullanarak azaltır.
- C) Aygıt sürücülerine gerek duymaz.
- D) CPU ve G/Ç aygıtları arasında gerçek zamanlı senkronizasyon sağlar.



Cevap

- Cevap: B
- Scatter-gather G/Ç, büyük veri bloklarını işlerken veri transferini daha verimli hale getirir. Bu işlem, veriyi ayrı parçalara böler ve bu parçaları farklı önbelleklerde toplar. Bu, veri aktarımı sürecini optimize eder ve veri aktarımı için gerekli olan işlemci zamanını azaltır. Her bir veri bloğunun ayrı ayrı işlenmesini sağlar ve büyük veri bloklarının tamamının tek bir önbelleğe sığdırılmasını önler. Veri kopyalama maliyetini azaltır. Çünkü veri, birden fazla önbellek arasında kopyalanmaz, bunun yerine veri doğrudan kaynaktan hedefe aktarılır.



Soru

- Çoklu görev ortamlarında G/Ç aygıtlarının CPU'nun dikkatini ele geçirmesini önlemek için hangi teknik kullanılır?
- A) Kesme tabanlı G/Ç
- B) Sorgulama
- C) DMA (Doğrudan Bellek Erişimi)
- D) Spinlock'lar



Cevap

- Cevap: A
- Kesme tabanlı G/Ç'de, bir G/Ç aygıtı işlemi tamamladığında, aygıt CPU'ya bir kesme sinyali gönderir. CPU, bu kesmeyi alır almaz, mevcut süreci askıya alır ve G/Ç aygıtının işlemi tamamlamasına yardımcı olmak için gerekli işlemleri gerçekleştirir. Çoklu görev ortamlarında, birden fazla süreç aynı anda çalışır. Kesme tabanlı G/Ç, bu süreçlerin birbirini engellemeden çalışmasını sağlar. CPU'nun zamanını etkin bir şekilde kullanmasını sağlar. G/Ç cihazlarıyla etkileşimi kolaylaştırır. Çünkü aygıtlar, işlemlerini tamamladıklarında hemen CPU'ya bir kesme gönderirler ve CPU, bu kesintiyi hemen işleyebilir.



Soru

- G/Ç işlemleri bağlamında, bir semaforun amacı nedir?
- A) Birden çok süreç arasında paylaşılan kaynaklara erişimi senkronize etmek.
- B) G/Ç işlemleri sırasında oluşan hataları işlemek.
- C) Aygıt sürücüleri için bellek tahsisini yönetmek.
- D) G/Ç aygıtlarıyla etkileşim için kullanıcı dostu bir arayüz sağlamak.



Cevap

- Cevap: A
- Birçok süreç, aynı anda paylaşılan kaynaklara erişmeye çalışabilir. Bu kaynaklar arasında bellek bölgeleri, dosya kaynakları veya G/Ç aygıtları gibi unsurlar bulunabilir. Semaforlar, bu kaynaklara erişimi kontrol ederek, bir sürecin kaynağa erişmeyi beklemesini veya diğer süreçlerin kaynağa erişmesini engellemeyi sağlar. Eşzamanlı çalışan süreçler arasında, paylaşılan kaynaklara erişim sırasında çakışmalar olabilir. Semaforlar, bu çakışmaları engelleyerek ve kaynaklara güvenli bir şekilde erişimi sağlayarak eşzamanlılık sorunlarını çözmeye yardımcı olur. Genellikle kritik bölge adı verilen bir kod parçasının güvenli bir şekilde yürütülmesini sağlamak için kullanılır.



Soru

- İşletim sistemlerinde halka önbelleği (ring buffer) G/Ç performansını nasıl artırır?
- A) Donanım yazmaçlarına doğrudan erişerek aygıt sürücülerine gereksinimini azaltır.
- B) Sürekli ve döngüsel veri depolamasına izin vererek çekişme (contention) ve ek yükü (overhead) en aza indirir.
- C) Sürekli olarak G/Ç aygıtlarını sorgulayarak kesme gereksinimini ortadan kaldırır.
- D) Birden fazla G/Ç aygıtı arasında gerçek zamanlı senkronizasyon sağlar.



Cevap

- Cevap: B
- Halka önbelleği, veriyi sürekli ve döngüsel bir şekilde depolar. Bu, veri depolama alanının sonuna ulaşıldığında, yeni veri eski verinin üzerine yazılır. Bu sürekli döngüsel işlem, veri depolama alanının etkin kullanılmasını sağlar ve veri depolama alanının boşa harcanmasını önler. Çekişme ve üstünlüğü en aza indirir. Çünkü halka önbelleği, veriyi sürekli olarak depolar ve süreçler arasında veri aktarımını sağlar. Bu, veri aktarımında çakışma olasılığını azaltır ve G/Ç işlemlerinin daha düzenli ve etkin bir şekilde gerçekleştirilmesini sağlar. Veri erişimi sırasında hiçbir zaman duraksama olmaz. Bu, veri erişim sürelerinin kısalmasını sağlar. Veri kaybını önler. Veri, halka önbelleğine yazıldığında, en eski veriler otomatik olarak silinir ve yerine yeni veriler yazılır.



Soru

- Bellek eşlemeli G/Ç (memory-mapped I/O) ile G/Ç eşlemeli G/Ç (I/O-mapped I/O) adresleme bakımından nasıl farklılık gösterir?
- A) Bellek eşlemeli, bellek ve G/Ç aygıtları için ayrı adres alanları kullanırken, G/Ç eşlemeli birleşik bir adres alanını paylaşır.
- B) Bellek eşlemeli, aygıtın yazmaçlarını doğrudan CPU'nun adres alanına eşlerken, G/Ç eşlemeli aygıtlara erişmek için özel komutlar kullanır.
- C) Bellek eşlemeli, özel donanım desteği gerektirirken, G/Ç eşlemeli yalnızca yazılım emülasyonuna dayanır.
- D) Bellek eşlemeli, belirli CPU mimarilerine özelken, G/Ç eşlemeli evrensel olarak desteklenir.



Cevap

- Cevap: B
- Bellek eşlemeli G/Ç'de, aygıtın yazmaçları doğrudan CPU'nun adres alanına eşlenir. Yani, aygıta erişmek için normal bellek erişim komutları kullanılabilir. G/Ç aygıtlarına veri yazmak veya okumak için özel komutlara gerek yoktur. G/Ç aygıtı yazmaçları, normal bellek adresleri gibi işlenir.
- G/Ç eşlemeli G/Ç'de ise, aygıtlara erişmek için ayrı bir adres alanı kullanılır. CPU, aygıta erişmek için özel komutları kullanır. Bu komutlar, G/Ç aygıtlarına erişmek için normal bellek erişim komutları değildir.



Soru

- Aşağıdakilerden hangisi vektörel kesme-tabanlı G/Ç'nin bir özelliğidir?
- A) CPU'nun düzenli olarak aygıtları sorgulamasını gerektirir.
- B) Birden çok aygıtın tek bir kesme isteği hattını paylaşmasına izin verir.
- C) Aygıtlar arası doğrudan veri transferi için DMA kullanır.
- D) Gerçek zamanlı işletim sistemlerinde yaygın olarak kullanılır.



Cevap

- Cevap: B
- Vektörize kesme-tabanlı G/Ç, birden çok aygıtın tek bir kesme isteği hattını paylaşmasına izin veren bir yöntemidir. Çok sayıda aygıtın aynı anda kesme talebinde bulunması durumunda kullanışlıdır çünkü her aygıt için ayrı bir kesme hattı gerektirmeden, tek bir kesme hattı üzerinden birden çok aygıtın kesme isteklerini işlemeyi sağlar. Özellikle yoğun G/Ç iş yüklerini etkin bir şekilde yönetmek için kullanılır. Bu, sistem tasarımını basitleştirir ve donanım kaynaklarını daha verimli kullanılmasını sağlar.



Soru

- Disk çizelgeleme algoritmaları bağlamında, hangisi talepleri mevcut disk kafa konumuna yakınlığına göre önceliklendirir?
- A) SCAN
- B) C-LOOK
- C) LOOK
- D) SSTF



Cevap

- Cevap: D
- SSTF, "En Yakın Talep Önce" anlamına gelir (Shortest Seek Time First). Bu algoritma, mevcut disk kafa konumuna en yakın olan veri taleplerini önceliklendirir. Yani, disk kafası bir konumdan diğerine hareket ederken, en yakın olan veri taleplerini öncelikle işler. SSTF'nin avantajı, disk hareketlerini minimize etmesidir. Bu, ortalama erişim süresini kısaltır ve disk performansını artırır. Ancak, bu algoritmanın dezavantajı, taleplerin bazen uzak bölgelerde sıkışıp kalmasına neden olabileceği ve bu durumun adil bir şekilde davranmasını etkileyebileceğidir.



Soru

- Disk G/Ç işlemlerinde "arama zamanı" (seek) terimi ne anlama gelir?
- A) Disk ve bellek arasında veri aktarımı için harcanan zaman.
- B) Disk kolunun istenen izin üzerine konumlanması için harcanan zaman.
- C) CPU'nun bir disk G/Ç isteğini işlemesi için harcanan zaman.
- D) Diskin gerekli dönme hızına ulaşmak için harcanan zaman.



Cevap

- Cevap: B
- "Arama zamanı", bir disk G/Ç işleminde, disk kolu veya okuyucunun istenen iz üzerine konumlanması için harcanan zamandır. Bu, disk kafasının bir izden diğerine hareket etmesi ve istenen veriye erişmek için gerekli konumu alması anlamına gelir. Arama zamanı, disk erişim süresinin önemli bir bileşenidir. Daha kısa arama süreleri, daha hızlı disk erişimi anlamına gelir ve dolayısıyla daha iyi bir performans sağlar. Disk çizelgeleme algoritmaları, arama zamanını minimize etmek için optimize edilir, böylece disk kafası mümkün olduğunca az yer değiştirir ve veri taleplerini hızlı bir şekilde işler.



Soru

- Bir işletim sistemindeki G/Ç çizelgeleyicinin temel işlevi nedir?
- A) CPU kaynaklarının G/Ç-odaklı süreçlere tahsisini yönetmek.
- B) Bekleyen G/Ç isteklerini önceliklendirerek disk performansını optimize etmek.
- C) Giriş/çıkış aygıtlarıyla etkileşim için bir kullanıcı arayüzü sağlamak.
- D) Giriş/çıkış süreçleri için erişim izinlerini kontrol etmek.



Cevap

- Cevap: B
- G/Ç çizelgeleyicisi, bekleyen G/Ç isteklerini önceliklendirir. Hangi G/Ç isteklerinin öncelikli olarak işleneceğini belirler. Bu, daha kritik veya sık kullanılan veri taleplerine öncelik verilmesini sağlar. Özellikle disk erişiminde, hangi G/Ç isteklerinin ne zaman işleneceğini belirler. Önbellek kullanımını optimize etmek için özel stratejiler kullanır.



Soru

- Hangi G/Ç tekniđi, CPU ve G/Ç aygıtı arasında sürekli sorgulama veya kesmelere gerek duymadan veri transferine izin verir?
- A) Doğrudan Bellek Erişimi (DMA)
- B) Programlı G/Ç
- C) Kesme-tabanlı G/Ç
- D) Bellek Eşlemeli G/Ç



Cevap

- Cevap: A
- Doğrudan Bellek Erişimi (DMA), CPU'nun doğrudan müdahale etmeden, G/Ç aygıtı ve bellek arasında veri transferi yapmasını sağlayan bir G/Ç tekniğidir. Bu, sürekli sorgulama veya kesmelere gerek olmadan, veri transferini daha verimli bir şekilde gerçekleştirmeyi mümkün kılar. DMA, CPU'nun müdahalesi olmadan doğrudan G/Ç aygıtı ve bellek arasında veri transferi gerçekleştirir. DMA, genellikle blok transferler yapar. Yani, tek bir komutla birden çok veri parçası (blok) transfer edilebilir. Sürekli sorgulama veya kesinti gerektirmediği için, DMA veri transferi daha verimli ve hızlıdır. DMA, özellikle yüksek bant genişliği gerektiren uygulamalarda yaygın olarak kullanılır.



Soru

- Aşağıdakilerden hangisi spooling'i doğru bir şekilde tanımlar?
- A) Simultaneous Peripheral Operations Online'nin kısaltmasıdır.
- B) Birden çok G/Ç işleminin aynı anda işlenmesine izin verir.
- C) İşleme hızını dengelemek için verilerin geçici bir kuyruğa depolanması.
- D) Toplu işlem sistemlerinde yazıcı çıktısını yönetmek için kullanılır.



Cevap

- Cevap: C
- Spooling (Simultaneous Peripheral Operations Online), bir bilgi işlem terimidir ve genellikle veri giriş/çıkış işlemlerini optimize etmek için kullanılır. Verileri geçici bir kuyruğa (spool) depolayarak işleme hızını dengeler. Özellikle, yavaş bir aygıtın hızlı bir şekilde işlenebilmesini sağlar. Örneğin, bir yazıcının yavaş işleme hızı, spooling sayesinde yazdırma işlemlerinin hızlı bir şekilde sırayla gerçekleştirilmesini sağlar. Birden çok işlemin aynı anda gerçekleştirilmesine olanak tanır. Bu, bir cihazın meşgul olması durumunda diğer işlemlerin sıraya alınmasını ve beklemesini sağlar. Genellikle yazıcı çıktısını yönetmek için kullanılır, ancak aynı zamanda diğer veri giriş/çıkış işlemlerini de optimize etmek için kullanılabilir.



Soru

- Hangi disk çizelgeleme algoritması, talepleri aldığı sıraya göre hizmet verir?
- A) SCAN
- B) C-SCAN
- C) FCFS (İlk Gelen, İlk Hizmet)
- D) SSTF (En Kısa Arama Süresi İlk)



Cevap

- Cevap: C
- FCFS (First-Come, First-Served), algoritması, talepleri aldığı sıraya göre hizmet verir. İlk gelen talep, ilk hizmet edilen talep olur. Bu algoritma, talepleri aldıkları sıraya göre işlediği için, tüm talepleri eşit şekilde ele alır. Bu, talepler arasında adil bir dağılım sağlar. Özellikle, taleplerin gelme sırası, disk üzerindeki veri dağılımına göre optimize edilmemişse, bu algoritma verimli olmayabilir.



Soru

- G/Ç işlemleri bağlamında, "pipelining" terimi ne anlama gelir?
- A) Birden fazla G/Ç isteğini tek bir veri akışına birleştirme süreci.
- B) G/Ç işleminin birden fazla aşamasının eşzamanlı olarak yürütülmesi.
- C) Verinin, bir ağ boru hattı ağı kullanılarak giriş ve çıkış aygıtları arasında yönlendirilmesi.
- D) G/Ç görevlerinin paralel olarak yürütülmesi için daha küçük alt görevlere bölünmesi.



Cevap

- Cevap: B
- Pipelining, bir G/Ç işleminin farklı aşamalarının aynı anda yürütülmesini sağlar. Örneğin, bir veri transferi işlemi, veri okuma, işleme ve yazma aşamalarından oluşabilir. Pipelining sayesinde, bu aşamalar eşzamanlı olarak yürütülür ve böylece işlem tamamlanma süresi azalır. Pipelining, işlemci veya G/Ç aygıtının tam kapasitesini kullanarak verimliliği artırır. G/Ç işlemlerinin paralel olarak işlenmesini sağlar. Bu, işlemi daha hızlı tamamlamak için birden fazla kaynağın kullanılmasını mümkün kılar.



Soru

- Programlanmış G/Ç'yi, kesme-tabanlı G/Ç'den ayıran nedir?
- A) Programlanmış G/Ç, CPU'nun sürekli olarak G/Ç aygıtını sorgulamasını gerektirir, kesme-tabanlı G/Ç ise CPU'nun G/Ç işlemi tamamlanana kadar diğer görevleri gerçekleştirmesine izin verir.
- B) Programlanmış G/Ç, veri transferi için DMA kullanırken, kesme-tabanlı G/Ç doğrudan CPU müdahalesine dayanır.
- C) Programlanmış G/Ç, kesme-tabanlı G/Ç'ye göre daha hızlı ancak daha az verimlidir.
- D) Programlanmış G/Ç, yalnızca düşük hızlı G/Ç aygıtları için uygunken, kesme-tabanlı G/Ç yüksek hızlı aygıtlar için kullanılır.



Cevap

- Cevap: A
- Programlanmış G/Ç (Programmed I/O): Bu yöntemde, CPU sürekli olarak G/Ç aygıtını sorgular ve G/Ç işlemi tamamlanana kadar diğer görevleri gerçekleştiremez. CPU, G/Ç işlemi için aygıtı komut gönderir ve ardından aygıtın hazır olduğunu kontrol etmek için tekrar tekrar sorgular.
- Kesme-tabanlı G/Ç (Interrupt-driven I/O): Bu yöntemde, CPU, G/Ç aygıtı işini tamamladığında kesmeler aracılığıyla bilgilendirilir. CPU, G/Ç işlemi tamamlanana kadar diğer görevleri gerçekleştirebilir. Kesme-tabanlı G/Ç, CPU'nun boşa harcanan zamanını azaltır ve sistem verimliliğini artırır.



Soru

- Hangi G/Ç tekniđi, birden fazla G/Ç işleminin aynı anda başlatılmasına ve bağımsız olarak tamamlanmasına izin verir?
- A) Dağıtık G/Ç (Scatter-gather I/O)
- B) Eşzamansız G/Ç (Asynchronous I/O)
- C) Senkron G/Ç (Synchronous I/O)
- D) Tamponlu G/Ç (Buffered I/O)



Cevap

- Cevap: B
- Eşzamansız G/Ç, birden fazla G/Ç işlemini aynı anda başlatılmasına izin verir. Bu, farklı G/Ç işlemlerinin aynı anda sistemde çalışmasını sağlar. Her bir G/Ç işleminin bağımsız olarak tamamlanmasına izin verir. Yani, bir işlem tamamlandığında, diğer işlemler bu durumdan etkilenmez ve kendi süreçlerine devam eder. Bu teknik, sistemde verimliliği ve paralel işlem kapasitesini artırır.



Soru

- Disk çizelgeleme algoritmalarından hangisi, disk kafasının mevcut konumuna yakın taleplere öncelik verir, ardından iz takip numaralarının artış yönünde ilerler ve son izine ulaştıktan sonra yönünü tersine çevirir?
- A) LOOK
- B) C-SCAN
- C) SCAN
- D) SSTF



Cevap

- Cevap: C
- SCAN algoritması, disk üzerindeki veri taleplerini belirli bir yönde tarar ve taleplere hizmet verir. Mevcut konuma yakın olan talepler öncelikli olarak işlenir. SCAN, iz takip numaralarının artış yönünde ilerler. Bu, disk kafasının mevcut konumundan daha uzak taleplere öncelik verildiğinde bile, mevcut konumdan sonraki taleplere hizmet edilmesini sağlar. Tara ve geri dönme işlemi, taleplerin tamamen taranması ve hizmet edilmesi için tüm disk yüzeyini kapsar.



Soru

- Eşzamanlı G/Ç'yi, eşzamanlı olmayan G/Ç'den ayıran nedir?
- A) Eşzamanlı G/Ç, CPU'nun G/Ç tamamlanana kadar diğer görevleri gerçekleştirmesine izin verirken, eşzamanlı olmayan G/Ç, CPU'yu G/Ç işlemi tamamlanana kadar bekletir.
- B) Eşzamanlı G/Ç, G/Ç işlemlerinin hemen tamamlanmasını garanti ederken, eşzamanlı olmayan G/Ç'nin tamamlanma durumunu belirlemek için açıkça sorgulanmasını gerektirir.
- C) Eşzamanlı G/Ç, CPU ve G/Ç aygıtı arasında doğrudan iletişim içerirken, eşzamanlı olmayan G/Ç iletişim için kesmelerden yararlanır.
- D) Eşzamanlı G/Ç, veriyi doğrudan bellekten G/Ç aygıtlarına aktarırken, eşzamanlı olmayan G/Ç veriyi aktarmadan önce bellekte tamponlar.



Cevap

- Cevap: A
- Eşzamanlı G/Ç (Synchronous I/O): Bu yöntemde, CPU, bir G/Ç işlemi tamamlanana kadar diğer görevleri gerçekleştirebilir. Yani, CPU, G/Ç işlemi tamamlanana kadar beklemek zorunda değildir.
- Eşzamanlı Olmayan G/Ç (Asynchronous I/O): Bu yöntemde, CPU, G/Ç işlemi tamamlanana kadar diğer görevleri gerçekleştiremez. CPU, G/Ç işlemi tamamlanana kadar beklemek zorundadır ve diğer işlemlere devam edemez.



Soru

- Aşağıdakilerden hangisi tamponlu G/Ç'nin bir özelliğidir?
- A) Veriyi doğrudan bellekten G/Ç aygıtlarına aktarır.
- B) Her veri aktarım işlemi için CPU'dan açık müdahale gerektirir.
- C) G/Ç aktarımları sırasında veriyi geçici olarak saklamak için tampon kullanır.
- D) Doğrudan G/Ç tekniklerine göre daha yavaştır.



Cevap

- Cevap: C
- Tamponlu G/Ç, veriyi geçici olarak saklamak için bir tampon kullanır. Örneğin, bir dosyanın diskten okunması sırasında, okunan veri önce bir tampona yazılır ve ardından işlenir veya kullanılır. Tamponlu G/Ç, verinin doğrudan bellekten G/Ç aygıtlarına aktarılmasını sağlar. Tampon, G/Ç aygıtları arasındaki hız farklarını dengelemek ve veri işleme sürecini iyileştirmek için kullanılır.



SON