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

Adı – Soyadı – Numarası:

Aşağıdaki sorulardan 10 tanesini seçip cevaplayınız.

Soru 1: İşletim sistemi ve temel fonksiyonları nelerdir? İşletim sisteminin tanımını yapınız ve kullanıcı ile donanım arasındaki rolünü açıklayınız.

İşletim sistemi (OS), bilgisayar donanımı ile kullanıcı arasında aracı görevi gören, sistem kaynaklarını (CPU, bellek, disk) yöneten ve uygulamaların çalışmasını sağlayan bir yazılımdır.

Temel Fonksiyonlar:

Kaynak Yönetimi: CPU, bellek ve giriş/çıkış (I/O) cihazlarını süreçler arasında paylaştırır.
Süreç Yönetimi: Süreçlerin oluşturulması, çalıştırılması ve sonlandırılmasını koordine eder.
Dosya Yönetimi: Dosya oluşturma, okuma, yazma ve silme işlemlerini sağlar.
Kullanıcı Arayüzü: Komut satırı veya grafik arayüz ile kullanıcı etkileşimini kolaylaştırır.
Güvenlik: Yetkisiz erişimi engeller ve veri bütünlüğünü korur.

Kullanıcı ve Donanım Arasındaki Rolü:

OS, donanımın karmaşıklığını soyutlar. Örneğin, bir kullanıcı dosyayı kaydetmek istediğinde, OS disk sürücüsüne erişim, veri yazımı ve dosya sistemi güncellemesi gibi işlemleri otomatik olarak gerçekleştirir. Kullanıcı yalnızca "kaydet" komutunu verir, donanım detaylarıyla uğraşmaz.

Soru 2: İşletim sisteminde çekirdek (kernel) ve rolü nedir? Çekirdeğin işletim sistemindeki işlevlerini ve neden önemli olduğunu açıklayınız.

Çekirdek, işletim sisteminin merkezi bileşenidir ve donanımla doğrudan iletişim kurarak kaynakları yönetir.

İşlevleri:

Süreç Kontrolü: Süreçlerin oluşturulması, zamanlanması ve senkronizasyonu.
Bellek Yönetimi: Sanal bellek tahsisi ve fiziksel bellek organizasyonu.
Cihaz Yönetimi: I/O cihazlarıyla iletişim (ör. klavye, yazıcı).
Kesme İşleme: Donanım olaylarını (ör. fare tıklaması) algılar ve yanıtlar.
Sistem Çağrıları: Uygulamaların donanıma erişimini sağlar.

Önemi:

Çekirdek, işletim sisteminin temel hizmetlerini sunar ve donanıma güvenli erişim sağlar. Onsuz, uygulamalar donanımla doğrudan iletişim kuramaz, bu da kaos ve güvenlik açıklarına yol açar. Çekirdek, sistemin performansını ve kararlılığını doğrudan etkiler.

Soru 3: Monolitik çekirdek ile mikro çekirdek arasındaki farklar nelerdir? Bu iki çekirdek türünün avantajlarını ve dezavantajlarını karşılaştırınız.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 05.05.2025

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

Monolitik Çekirdek: Tüm işletim sistemi hizmetleri (süreç yönetimi, dosya sistemi, cihaz sürücüleri) tek bir adres alanında çalışır. Avantaj: Yüksek performans, hızlı yürütme. Dezavantaj: Hata yönetimi zor, bakım karmaşık.

Mikro Çekirdek: Yalnızca temel hizmetler (süreç yönetimi, bellek yönetimi, iletişim) çekirdekte yer alır; diğer hizmetler kullanıcı modunda ayrı süreçler olarak çalışır. Avantaj: Daha güvenli, kolay genişletilebilir. Dezavantaj: IPC nedeniyle performans kaybı.

Soru 4: Tamponlama (buffering) bir bilgisayar sisteminin performansını nasıl artırır? Tamponlamanın nasıl çalıştığını ve sistem performansına etkisini açıklayınız.

Tamponlama, farklı hızlarda çalışan cihazlar (ör. disk ve CPU) arasında veri aktarımını düzenlemek için geçici bir bellek alanı (tampon) kullanılmasıdır.

Çalışma Mekanizması:

Veri, yavaş bir cihazdan (ör. sabit disk) hızlı bir cihaza (ör. RAM) aktarılırken tamponda birikir. CPU, bu verilere tampon üzerinden erişir, böylece disk erişim gecikmeleri azalır.

Performans Etkisi:

G/Ç Verimliliği: CPU, disk verisini beklemeden tampondan veri alır, işlemci boşta kalmaz.
Hızlı Yanıt: Kullanıcıya daha hızlı veri erişimi sunulur.
Kaynak Optimizasyonu: Farklı cihazların hız farkını telafi eder, sistem verimliliğini artırır.
Örneğin, bir video akışı oynatılırken, veri tamponda birikir ve oynatma kesintisiz devam eder.

Soru 5: İşletim sisteminde "iş" (job) ile "süreç" (process) arasındaki fark nedir? Bu iki kavramı tanımlayınız ve aralarındaki farkları belirtiniz.

İş (Job): Kullanıcının çalıştırmak istediği bir görev veya programdır (ör. bir betik dosyası çalıştırma). Genellikle toplu işlem (batch processing) bağlamında kullanılır.

Süreç (Process): Çalışan bir programın aktif örneğidir. İşletim sistemi tarafından yönetilir ve kendine ait bellek alanı, CPU zamanı ve kaynakları vardır.

Farklar:

Kapsam: İş, daha geniş bir kavramdır; bir iş birden fazla süreci içerebilir.
Yönetim: İş, kullanıcı düzeyinde tanımlanır; süreç, OS tarafından yönetilir.
Örnek: Bir derleyici çalıştırmak bir iştir; derleyicinin çalışması bir süreçtir.

Soru 6: Bir süreç kontrol bloğunda (process control block - PCB) hangi bileşenler bulunur? PCB'nin içeriğini kısaca açıklayınız.

PCB, bir sürecin durumunu ve özelliklerini saklayan veri yapısıdır. İşletim sistemi, süreç yönetimi için PCB'yi kullanır. PCB, süreçler arası geçiş (context switch) sırasında kritik rol oynar.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 05.05.2025

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

Bileşenler:

Süreç Kimliği (PID): Süreci karakteristik olarak benzersiz şekilde tanımlar.
Süreç Durumu: Yeni, hazır, çalışıyor, bekliyor, sonlanmış.
Program Sayacı: Bir sonraki çalıştırılacak komutun adresi.
CPU Kayıtları: Sürecin geçici verileri (ör. genel amaçlı kayıtlar).
Bellek Bilgisi: Sürecin kullandığı bellek adresleri ve sınırları.
I/O Durumu: Açık dosyalar, kullanılan I/O cihazları.
Zamanlama Bilgisi: Öncelik, CPU kullanım süresi.

Soru 7: Unix'te süreçler fork() sistem çağrısı ile nasıl oluşturulur? fork() çağrısının çalışma mekanizmasını ve süreç oluşturma sürecini tarif ediniz.

Unix'te fork() sistem çağrısı, mevcut süreci (ata/parent) kopyalayarak yeni bir süreç (çocuk/child) oluşturur.

Çalışma Mekanizması:

Ata süreç fork() çağrısını yapar.
İşletim sistemi, ata sürecin bellek görüntüsünü, dosya tanıtıcılarını ve durumunu kopyalar.
Çocuk süreç, ayrı bir PCB alır ancak bellek alanı başlangıçta ata ile paylaşılır (copy-on-write ile).

fork() dönüş değeri:
Ata süreçte: Çocuk sürecin PID'si (0'dan farklı).
Çocuk süreçte: 0.
Çocuk süreç genellikle exec() ile yeni bir program yükler.
Örnek: Bir terminalde çalışan bir program, yeni bir kopya oluşturmak için fork() kullanır.

Soru 8: Bir süreç, engellenmiş (blocked) durumdan hazır (ready) duruma nasıl geçer? Süreç durumlarını ve bu geçişin nasıl gerçekleştiğini açıklayınız.

Süreç Durumları:

Yeni (New): Süreç oluşturuluyor.
Hazır (Ready): CPU'da çalışmak için bekliyor.
Çalışıyor (Running): CPU'yu kullanıyor.
Engellenmiş (Blocked/Waiting): I/O veya bir olay bekliyor.
Sonlanmış (Terminated): Süreç tamamlandı.

Blocked'dan Ready'ye Geçiş:

Süreç, bir I/O işlemi (ör. dosya okuma) veya olay (ör. kullanıcı girişi) beklerken blocked durumuna geçer.
Beklenen olay gerçekleştiğinde (ör. I/O tamamlandığında), işletim sistemi süreci ready kuyruğuna taşır.
Süreç, CPU boşaldığında running durumuna geçer.
Örnek: Bir süreç, klavye girişi beklerken blocked olur; kullanıcı bir tuşa bastığında ready olur.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 05.05.2025

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

Soru 9: Ata süreç sonlandığında çocuk süreçlere ne olur? Ata sürecin sonlanmasının çocuk süreçler üzerindeki etkisini tartışınız.

Unix sistemlerinde, ata süreç sonlandığında çocuk süreçler "yetim" (orphan) olur.

Etkiler:

Çocuk süreçler, init süreci (PID 1) tarafından evlat edinilir.
Çocuk süreçler çalışmaya devam edebilir, ancak ata süreçten bağımsızlık kazanır.
Bazı sistemlerde (ör. Windows), ata süreç sonlandığında çocuk süreçler de sonlandırılabilir.
Örnek: Bir terminal kapanırsa (ata süreç), arka planda çalışan bir program (çocuk süreç) init tarafından devralınır.

Soru 10: İşletim sistemlerinde farklı iş parçacığı (thread) modelleri nelerdir? Thread modellerini (ör. Kullanıcı seviyesi, çekirdek seviyesi) açıklayınız.

İş parçacıkları, süreç içinde çalışan hafif yürütme birimleridir.

Modeller:

Kullanıcı Seviyeli Thread'ler: Uygulama tarafından yönetilir, işletim sistemi thread'leri görmez. Avantaj: Hızlı, düşük maliyetli context switch. Dezavantaj: Bir thread engellenirse tüm süreç engellenir.

Çekirdek Seviyeli Thread'ler: İşletim sistemi tarafından yönetilir, her thread ayrı bir varlık olarak görülür. Avantaj: Bir thread engellense bile diğerleri çalışabilir. Dezavantaj: Context switch pahalıdır.

Hibrit Model: Kullanıcı ve çekirdek seviyeli kombinasyonu; esneklik sağlar.

Soru 11: İş parçacıkları (thread) ile kullanıldığında fork() sistem çağrısı nasıl davranır? İş parçacığı içeren bir süreçte fork() çağrısının davranışını tarif ediniz.

Unix'te, çok thread'li bir süreç fork() çağrısı yaptığında: Yalnızca fork()'u çağıran thread kopyalanır; diğer thread'ler çocuk süreçte yer almaz. Çocuk süreç, ata sürecin bellek görüntüsünü alır, ancak yalnızca tek bir thread ile başlar. Bu durum, paylaşılan kaynaklarda (ör. kilitler) sorunlara yol açabilir, çünkü diğer thread'lerin durumu kopyalanmaz.
Örnek: Bir web sunucusu, yeni bir istemci isteği için fork() çağırdığında, yalnızca istemciyi işleyen thread kopyalanır.

Soru 12: İş parçacığı havuzu (thread pool) nedir ve neden kullanılır? Thread havuzunun amacını ve avantajlarını açıklayınız.

Thread havuzu, önceden oluşturulmuş ve yeniden kullanılabilir thread'lerin bir koleksiyonudur.

Amaç: Sürekli thread oluşturma ve yok etme maliyetini azaltmak. Görevleri hızlı bir şekilde işlemek için hazır thread'ler sağlamak.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 05.05.2025

Avantajlar:

Performans artışı: Thread oluşturma süresi ortadan kalkar.
Kaynak tasarrufu: Bellek ve CPU kullanımı optimize edilir.
Ölçeklenebilirlik: Çoklu görevleri paralel olarak yönetir.
Örnek: Web sunucuları (ör. Apache), istemci isteklerini işlemek için thread havuzu kullanır.

Soru 13: Kritik bölüm (critical section) nedir ve senkronizasyon için neden önemlidir? Kritik bölüm kavramını tanımlayınız ve senkronizasyondaki rolünü açıklayınız.

Kritik bölüm, birden fazla süreç veya thread'in paylaşılan bir kaynağa (ör. dosya, bellek) aynı anda erişimini engellemek için korunan kod parçasıdır.

Senkronizasyondaki Rolü:

Veri tutarlılığını sağlar; örneğin, iki thread'in aynı değişkeni aynı anda değiştirmesi engellenir.
Yarış koşullarını (race conditions) önler.
Kilitler (locks) veya semaphore'lar kullanılarak uygulanır.
Örnek: Bir banka hesabına para yatırma işlemi, kritik bölümde yalnızca bir thread tarafından yapılır.

Soru 14: Mutex ile semaphore arasındaki farklar nelerdir? Bu iki senkronizasyon mekanizmasının farklarını ve kullanım alanlarını karşılaştırınız.

Mutex: İkili kilit (locked/unlocked); yalnızca bir thread'in kritik bölüme erişmesini sağlar.
Semaphore: Sayısal bir sayaç; birden fazla thread'in sınırlı sayıda kaynağa erişmesine izin verir.
Karşılaştırma:

Mutex	Semaphore
İkili (0 veya 1).	Sayısal (0, 1, 2, …).
Kritik bölüm koruma.	Kaynak paylaşımı kontrolü.
Dosya yazımı kilitleme.	5 bağlantıya izin veren sunucu.

Kullanım Alanları: Mutex: Tekil kaynak erişimi (ör. veritabanı kaydı güncelleme). Semaphore: Çoklu kaynak yönetimi (ör. bağlantı havuzu).

Soru 15: Yemek yiyen filozoflar problemi nedir ve senkronizasyonla nasıl ilişkili olduğunu belirtiniz.

Beş filozof, yuvarlak bir masada oturur ve yemek yemek için iki çatala ihtiyaç duyar. Her filozofun sağında ve solunda birer çatal vardır, ancak çatallar paylaşılır.

Senkronizasyonla İlişkisi:

Problem, paylaşılan kaynaklara (çatallar) erişimde yarış koşullarını ve deadlock'u gösterir.
Çözüm, senkronizasyon mekanizmaları (ör. semaphore, mutex) kullanılarak sağlanır.
Örneğin, her çatal bir semaphore ile korunabilir; filozoflar, iki çatalı da almadan yemek yiyemez.
Örnek Çözüm: Filozoflardan biri önce sol, diğeri sağ çatalı alırsa deadlock önlenebilir.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 05.05.2025

Soru 16: Süreç çizelgeleyici (process scheduler) nedir ve iyi bir çizelgeleyicinin özellikleri nelerdir?

Süreç çizelgeleyicisi, CPU'yu hangi sürecin kullanacağına karar veren işletim sistemi bileşenidir.

İyi Bir Çizelgeleyicinin Özellikleri:

Adalet: Her sürece eşit CPU süresi verir.
Verimlilik: CPU'yu maksimum düzeyde kullanır, boşta kalmaz.
Düşük Gecikme: Süreçlerin yanıt süresi kısa olur.
Esneklik: Farklı süreç türlerine (etkileşimli, toplu) uygun davranır.
Ölçeklenebilirlik: Çok sayıda süreçle başa çıkabilir.
Örnek: Round-robin çizelgeleme, adaleti sağlamak için her sürece sabit bir zaman dilimi atar.

Soru 17: Bir çizelgeleme algoritması seçerken hangi ilkeler göz önünde bulundurulmalıdır? Bilinen algoritmaları listeleyiniz.

Seçim İlkeleri:

CPU Kullanımı: CPU'nun boşta kalma süresi minimum olmalı.
Yanıt Süresi: Etkileşimli süreçler için hızlı yanıt.
Bekleme Süresi: Süreçlerin kuyrukta bekleme süresi kısa olmalı.
Adalet: Tüm süreçlere eşit fırsat.
Sistem Türü: Gerçek zamanlı, toplu veya etkileşimli sistemlere uygunluk.

Bilinen Algoritmalar:

First-Come-First-Served (FCFS): İlk gelen önce çalışır.
Shortest Job First (SJF): En kısa süreli iş önce çalışır.
Priority Scheduling: Öncelik sırasına göre çalışır.
Round-Robin (RR): Her sürece sabit zaman dilimi atanır.
Multilevel Queue: Süreçler farklı öncelik kuyruklarına ayrılır.
Multilevel Feedback Queue: Süreçler, davranışlarına göre kuyruklar arasında taşınır.