



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

Adı – Soyadı – Numarası:

Soru 1: İşletim sisteminin temel fonksiyonlarını birer cümle ile listeleyiniz.

İşletim sistemi, bilgisayar donanımı ile kullanıcı arasında bir köprü görevi görerek çeşitli temel fonksiyonları yerine getirir. **Süreç yönetimi**, çalışan programların yürütülmesini, çizelgelenmesini ve senkronizasyonunu sağlar. **Bellek yönetimi**, RAM'in verimli kullanılmasını sağlayarak süreçlerin bellek gereksinimlerini düzenler. **Dosya sistemi yönetimi**, verilerin düzenlenmesi, depolanması ve erişimini kontrol eder. **Giriş/çıkış (I/O) yönetimi**, klavye, fare, disk ve yazıcı gibi donanımlarla etkileşimi sağlar.

Çoklu görev (multitasking) yönetimi ile aynı anda birden fazla sürecin çalışmasını düzenler. **Kullanıcı arayüzü** sağlama, grafiksel (GUI) veya komut satırı (CLI) üzerinden kullanıcı ile etkileşim kurmayı mümkün kılar. **Güvenlik ve erişim kontrolü**, yetkisiz erişimleri engelleyerek sistem güvenliğini sağlar. **Hata yönetimi**, sistemde oluşan hataları tespit ederek düzeltme mekanizmaları sunar.

Soru 2: İşlemci ile ana bellek arasında veri erişimini hızlandırmak için neler yapılabilir?

İşlemci ile ana bellek arasındaki veri erişimini hızlandırmak için önbellek (cache memory) kullanılır; L1, L2 ve L3 önbellekler sık kullanılan verileri saklayarak erişim süresini azaltır. Çok kanallı bellek (dual/quad-channel) yapılandırması, bellek modülleri arasındaki veri iletimini artırarak bant genişliğini genişletir. Önceden getirme (prefetching) mekanizmaları, işlemcinin ihtiyaç duyabileceği verileri tahmin ederek gecikmeyi azaltır. DDR4 veya DDR5 gibi yüksek hızlı RAM kullanımı, veri aktarım hızlarını artırarak performansı iyileştirir. DMA (Direct Memory Access) kullanımı, işlemciyi meşgul etmeden doğrudan veri transferi yapılmasını sağlayarak veri erişimini hızlandırır.

Soru 3: Kabuğun (shell) temel işlevi nedir?

Kabuk (shell), işletim sistemi ile kullanıcı arasında bir ara yüz sağlayarak komutları yorumlama, program çalıştırma, dosya yönetimi ve süreç kontrolü gibi temel işlevleri yerine getirir. Kullanıcıdan gelen komutları alıp işletim sisteminin çekirdeğine (kernel) ileterek yürütülmesini sağlar. Bash, Zsh, PowerShell gibi farklı kabuk türleri, kullanıcıların komut satırı aracılığıyla işletim sistemini yönetmesine olanak tanır.

Soru 4: Sistem çağrısının (system call) temel amacı nedir?

Sistem çağrısı (system call), kullanıcı seviyesindeki programların işletim sistemi çekirdeği (kernel) ile iletişim kurmasını sağlayan bir mekanizmadır. Temel amacı, uygulamaların dosya işlemleri, bellek yönetimi, süreç kontrolü ve donanım erişimi gibi düşük seviyeli işlemleri güvenli bir şekilde gerçekleştirmesidir. Kullanıcı programları doğrudan donanım veya çekirdek seviyesindeki kaynaklara erişemez, bu yüzden read, write, open, close, fork, exec, exit gibi sistem çağrıları aracılığıyla işletim sisteminden hizmet talep eder.

Soru 5: İşletim sistemleri bağlamında soyutlama nedir?

Soyutlama (abstraction), işletim sistemlerinin donanımın karmaşıklığını gizleyerek kullanıcıya ve yazılımlara daha basit ve tutarlı bir arayüz sunma işlemidir. İşletim sistemleri, donanım kaynaklarını doğrudan yönetmek yerine dosya sistemleri, süreç yönetimi, bellek yönetimi ve giriş/çıkış işlemleri gibi soyut kavramlar sunarak programcıların ve kullanıcıların düşük seviyeli ayrıntılarla uğraşmasını önler. Örneğin, bir depolama aygıtı blok cihazı olarak soyutlanır ve kullanıcılar fiziksel disk yapısını bilmeden dosyalarla işlem yapabilir. Soyutlama, taşınabilirlik, güvenlik ve modülerlik gibi avantajlar sağlayarak yazılım geliştirmeyi kolaylaştırır ve farklı donanımlarla uyumlu çalışmayı mümkün kılar.

Soru 6: Eşzamanlılık nedir?



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

Eşzamanlılık (concurrency), bir bilgisayar sisteminde birden fazla işlemin (process) veya iş parçacığının (thread) aynı zaman aralığında yürütülmesi durumudur. Gerçek donanımda, tek bir işlemci varsa, işletim sistemi zaman paylaşımı (time-sharing) çoklu görev (multitasking) mekanizmalarıyla bu işlemleri kısa süreli aralıklarla sırayla çalıştırır, böylece eşzamanlı çalışıyormuş gibi görünmesini sağlar. Çok çekirdekli sistemlerde ise gerçek eşzamanlılık mümkün olup, farklı çekirdekler aynı anda farklı işlemleri yürütebilir. Eşzamanlılık, verimliliği artırmak, yanıt süresini iyileştirmek ve sistem kaynaklarını daha etkin kullanmak için işletim sistemlerinde ve paralel programlamada kritik bir rol oynar. Ancak, yarış koşulları (race conditions), karşılıklı kilitlenme (deadlock) ve senkronizasyon problemleri gibi zorlukları da beraberinde getirir.

Soru 7: Linux'ta proc dizinin amacı nedir? Dizin içerisinde hangi bilgiler bulunur?

/proc dizini, Linux işletim sisteminde sanal bir dosya sistemi (procfs) olup, çekirdek ve çalışan süreçler hakkında sistem bilgileri sağlayan bir arabirimdir. /proc dizini, sistem yöneticileri ve geliştiriciler için gerçek zamanlı sistem izlemesi ve hata ayıklama gibi işlemlerde kritik bir rol oynar.

/proc/[PID]: Belirtilen süreç kimliğine (PID) sahip işlemin bilgilerini içerir.
/proc/[PID]/status: Sürecin durum bilgileri (bellek kullanımı, kullanıcı kimliği vb.).
/proc/[PID]/fd/: Sürecin açık dosya tanımlayıcıları (file descriptors).
/proc/cpuinfo: İşlemci hakkında bilgiler (çekirdek sayısı, hız, model vb.).
/proc/meminfo: Sistem belleği kullanımı ve tahsisi hakkında bilgiler.
/proc/uptime: Sistemin ne kadar süredir çalıştığını gösterir.
/proc/loadavg: Sistem yük ortalamalarını gösterir.
/proc/modules: Yüklü çekirdek modüllerini listeler.
/proc/sys/: Çekirdek parametrelerini ayarlamak için kullanılan yapılandırma dosyalarını içerir.

Soru 8: Von Neumann mimarisinin en önemli özelliği nedir?

Von Neumann mimarisinin en önemli özelliği, program ve verilerin aynı bellekte saklanması ilkesidir. Bu mimaride hem program komutları hem de veriler, tek bir ortak bellek biriminde depolanır ve işlemci, bu belleği hem komutları almak hem de verileri işlemek için kullanır. Bu yaklaşım, işlemciyi daha basit ve esnek hale getirirken, bellek ile işlemci arasındaki veri alışverişinin daha doğrudan yapılmasını sağlar. Von Neumann mimarisi, modern bilgisayarların temelini oluşturur.

Soru 9: Mikro çekirdek ve monolitik mimariyi karşılaştırınız.

Monolitik mimari, tüm işletim sistemi işlevlerini tek bir büyük çekirdek altında toplar, yani dosya sistemi, bellek yönetimi, süreç yönetimi gibi işlevler aynı çekirdek içinde yer alır. Bu, daha hızlı iletişim ve işlemci kaynaklarının verimli kullanılması sağlar, ancak büyük ve karmaşık bir yapıya yol açar, ayrıca hata durumlarında çekirdek çökebilir. Öte yandan, mikro çekirdek mimarisi yalnızca temel işlevleri (örneğin, iletişim, çizelgeleme, temel donanım erişimi) çekirdek içinde tutar ve diğer sistem işlevleri kullanıcı alanında çalışan modüller olarak dışarıya taşınır. Bu yapı daha güvenli ve esnektir, ancak mikro çekirdekler arasındaki iletişim, monolitik mimariye göre daha yavaştır. Mikro çekirdekler daha modüler ve hata izole edilebilirken, monolitik çekirdekler daha hızlı ve daha az karmaşıktır.

Soru 10: Kesme tabanlı (preemptive) ile kesme tabanlı olmayan (non-preemptive) çizelgeleme algoritmaları arasındaki fark nedir?

Kesme tabanlı (preemptive) çizelgeleme, bir işlem çalışırken başka bir işlem daha öncelikli hale geldiğinde, işlemciyi zorla keser (preempt) ve yeni işlemi çalıştırmaya başlar. Bu yaklaşım, daha adil ve yanıt süresi kısa olan sistemler sağlar, çünkü yüksek öncelikli işlemler hemen yürütülür.



Örnekler arasında Round Robin, Shortest Job Next (SJN) gibi algoritmalar yer alır.

Kesme tabanlı olmayan (non-preemptive) çizelgeleme ise, bir işlem başladığında, işlem tamamlanana kadar kesilmez ve işlem sırasındaki diğer işler beklemek zorundadır. Bu tür sistemler genellikle daha az yönetim karmaşası içerir ancak yanıt süreleri daha uzun olabilir, çünkü bir işlem tamamlanmadan diğer işlemler çalıştırılmaz. First Come First Serve (FCFS) ve Priority Scheduling algoritmaları bu türe örnek gösterilebilir.

Soru 11: "Round Robin" zamanlama algoritmasında, zaman dilimi (time quantum) çok küçük seçilirse ne olur?

a) Sistem verimliliği artar ve süreçler daha hızlı tamamlanır.

b) Bağlam değiştirme (context switch) masrafı artar ve sistem verimliliği azalır.

c) Süreçler arasında adalet sağlanamaz.

d) Kilitlenme (Deadlock) olasılığı artar.

Round Robin zamanlama algoritmasında, zaman dilimi (time quantum) çok küçük seçilirse, her süreç çok kısa sürelerle çalıştırılır ve bağlam değiştirme (context switch) işlemi sıklaşır. Bağlam değiştirme, işlemciyi bir süreçten diğerine geçirmeyi içerdiğinden, her geçişte zaman ve kaynak tüketimi oluşur.

Soru 12: Aşağıdaki senkronizasyon mekanizmalarından hangisi, meşgul bekleme (busy waiting) yöntemine dayanır?

a) Semafor işaret fişegi (Semaphore)

b) Gözleyici (Monitor)

c) Döngüsel Kilit (Spinlock)

d) Koşul Değişkeni (Condition Variable)

Döngüsel Kilit (Spinlock), bir süreç kaynakları kullanılamaz olduğunda sürekli olarak kilidin serbest kalıp kalmadığını kontrol eden ve bu kontrolü sürekli olarak yapan bir senkronizasyon mekanizmasıdır. Bu nedenle, meşgul bekleme (busy waiting) yöntemiyle çalışır; yani işlem kilidi

beklerken aktif olarak işlemciyi kullanmaya devam eder.

Soru 13: Bir işletim sisteminde, iki süreç arasında süreçler arası iletişim (inter-process communication) için paylaşılan bellek (shared memory) kullanıldığında, hangi durum senkronizasyon gerektirmez?

a) Bir süreç yazarken diğer süreç okuma yapıyorsa

b) Her iki süreç de aynı anda yazma yapıyorsa

c) Her iki süreç de okuma yapıyorsa

d) Bir süreç belleği temizlerken diğer süreç erişmeye çalışıyorsa

Paylaşımlı bellek (shared memory) kullanıldığında, iki süreç arasındaki senkronizasyon, verilerin tutarlılığını sağlamak için gereklidir. Ancak, her iki süreç de sadece okuma yapıyorsa, birbirlerinin verilerine zarar verme riski yoktur, çünkü okuma işlemleri birbirini etkilemez.

Soru 14: Öncelik tersine dönmesi (Priority Inversion) sorunu nedir ve hangi senaryoda ortaya çıkar?

a) Düşük öncelikli bir iş parçasının, yüksek öncelikli bir iş parçasını engellemesi; paylaşılan bir kaynak kilitlendiğinde

b) Yüksek öncelikli bir sürecin CPU'yu tamamen ele geçirmesi; zamanlama algoritması başarısız olduğunda

c) Bir iş parçasının sonsuz döngüye girmesi; bellek yetersizliğinde

d) İki sürecin aynı anda aynı dosyaya yazmaya çalışması; dosya sistemi hatasında

Öncelik tersine dönmesi (Priority Inversion), düşük öncelikli bir iş parçasının, yüksek öncelikli bir iş parçasını engellemesi durumudur. Bu problem, genellikle paylaşılan bir kaynak (örneğin, kilit) üzerinde senkronizasyon yaparken ortaya çıkar. Düşük öncelikli bir iş parçası bir kaynağa kilit koyarsa ve daha sonra yüksek öncelikli bir iş parçası bu kaynağı beklerse, yüksek öncelikli iş parçası beklemek zorunda kalır. Bu durumda, bekleyen yüksek öncelikli iş parçasının işlemciye erişimi, düşük öncelikli iş parçası tarafından



engellenmiş olur ve bu, öncelik tersine dönmesi
problemine yol açar.

Soru 15: Bir süreç diskten okuma yapmak
istediğinde geçiş yapacağı durum hangisidir?

- a) Hazır (ready)
- b) Engellenmiş (blocked)**
- c) Çalışıyor (running)
- d) Sonlandırıldı (terminated)

Bir süreç, diskten okuma yapmak istediğinde, disk işlemi zaman alacağı için engellenmiş (blocked) duruma geçer. Bu durumda süreç, diskten veri okuma işlemi tamamlanana kadar işlemciyi bırakır ve bekleme durumuna geçer. Disk işlemi tamamlandığında, süreç hazır (ready) duruma geçer ve işlemciyi almak için beklemeye başlar. Hazır (ready) durumu, süreç işlemciyi beklerken bulunduğu durumdur. Çalışıyor (running) durumu, süreç işlemciyi kullanırken bulunduğu durumdur. Sonlandırıldı (terminated) durumu, süreç tamamlandığında ve sona erdiğinde kullanılan durumdur.

Soru 16: İşletim sistemlerinde süreç denetim bloğunun (PCB) temel amacı nedir?

- a) Bir sürecin yürütülebilir kodunu saklamak
- b) Sistem kaynaklarının süreçlere tahsisini yönetmek
- c) Bir sürecin durumu ve özellikleri hakkındaki bilgileri korumak**
- d) Birden çok süreç arasında süreç dışı iletişimi sağlamak

Süreç denetim bloğu (PCB), işletim sisteminin her bir süreçle ilgili bilgileri sakladığı bir veri yapısıdır. PCB, sürecin kimliği (PID), durumu (çalışıyor, bekliyor, vb.), program sayacı, kayıtlar, belleğe dair bilgiler ve diğer önemli durumu hakkında verileri içerir. Bu bilgiler, işletim sistemine süreçlerin yönetilmesi, planlanması ve geçişleri sırasında gerekli verileri sağlar.