



**Adı – Soyadı – Numarası:**

1. Aşağıdaki çizelgeleme algoritmalarından hangisi açlıkla (starvation) sonuçlanabilir?

- a) Sıralı çizelgeleme (*round robin*)
- b) Öncelik çizelgelemesi (*priority scheduling*)**
- c) İlk Gelen, İlk Hizmet Alır (*first come first served*)
- d) Önce En Kısa İş (*shortest job first*)

2. Aşağıdaki çizelgeleme algoritmalarından hangisi konvoy etkisine yol açabilir?

- a) Sıralı çizelgeleme (*round robin*)
- b) Öncelik çizelgelemesi (*priority scheduling*)
- c) İlk Gelen, İlk Hizmet Alır (*first come first served*)**
- d) Önce En Kısa İş (*shortest job first*)

3. Aşağıdakilerden hangisi işletim sisteminin birincil işlevlerinden değildir?

- a) Bellek yönetimi
- b) Girdi/çıkı yönetimi
- c) Kullanıcı arayüzü tasarımı**
- d) Süreç yönetimi

4. Aşağıdakilerden hangisi sistem çağrısı (system call) türü değildir?

- a) Dosya işlemleri
- b) Bellek tahsisi
- c) G/Ç aygıtı işlemleri
- d) Veri şifreleme**

5. Aşağıdakilerden hangisi bir sürecin durumlarından değildir?

- a) Çalışıyor (running)
- b) Hazır (ready)
- c) Bloke (blocked)
- d) Aktif (active)**

6. Aşağıdakilerden hangisi bir süreç çizelgeleme algoritması türü değildir?

- a) Sıralı çizelgeleme (Round-robin)
- b) Öncelik çizelgeleme (Priority scheduling)
- c) İlk gelen ilk hizmet alır (FCFS)
- d) Eşzamanlı çizelgeleme (Concurrent scheduling)**

7. Aşağıdakilerden hangisi bir süreç içinde iş parçacığı kullanmanın yararlarından biri değildir?

- a) Paralel yürütme yoluyla geliştirilmiş performans
- b) İşlemci kaynaklarının daha iyi kullanılması
- c) İş parçacıkları arası kolay iletişim ve koordinasyon
- d) Dış süreçlere bağlılığın azalması ile artan güvenilirlik**

8. Aşağıdakilerden hangisi çizelgeleme algoritmalarını değerlendirmek için kullanılan yaygın bir ölçü değildir?

- a) Ortalama bekleme süresi (waiting time)
- b) Ortalama geri dönüş süresi (turnaround time)
- c) İşlemci kullanımı (utilization)
- d) Disk G/Ç verimi (Disk I/O throughput)**

9. Aşağıdakilerden hangisi eşzamanlı yürütme ile ilgili yaygın bir sorun değildir?

- a) Kilitlenmeler
- b) Açlık (Starvation)
- c) Canlı kilit (Livelocks)
- d) Tekeller (Monopolies)**

10. Aşağıdakilerden hangisi işletim sistemlerinde yaygın olarak kullanılan senkronizasyon mekanizmalarından biri değildir?

- a) Semaforlar (semaphore)
- b) Muteksler
- c) Durum değişkenleri (Condition variables)
- d) Paylaşımlı bellek (shared memory)**

11. Aşağıdakilerden hangisi mikro çekirdek (micro kernel) mimarisi kullanmanın avantajlarından biri değildir?

- a) Modülerlik ve geliştirme kolaylığı
- b) Daha verimli süreçler arası iletişim (IPC)
- c) Daha iyi güvenlik ve güvenilirlik
- d) Daha verimli bellek yönetimi**

12. Aşağıdakilerden hangisi monolitik (monolithic) çekirdeğin özelliklerinden biri değildir?

- a) Tüm çekirdek işlevleri tek bir adres alanında uygulanır
- b) Çekirdek kodu belleğe tek bir ikili dosya olarak yüklenir
- c) Çekirdek modülerdir ve kolayca genişletilebilir**
- d) Çekirdek, tüm sistem hizmetlerini doğrudan uygulamalara sağlar.

13. Aşağıdakilerden hangisi süreçler arası iletişim (IPC) mekanizması türlerinden biri değildir?

- a) Paylaşımlı bellek (shared memory)
- b) İleti geçişi (Message passing)
- c) Boru hattı (pipe)
- d) Kesmeler (interrupts)**



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

14. Bir süreç ile bir iş parçacığı arasındaki temel fark nedir?

- a) Bir sürecin kendi bellek alanı vardır, iş parçacıkları aynı bellek alanını paylaşır.
- b) Bir süreç birden fazla işlemci üzerinde çalışabilirken, bir iş parçacığı yalnızca tek bir işlemci üzerinde çalışabilir.
- c) Bir süreç birden çok iş parçacığına sahip olabilirken, bir iş parçacığı yalnızca tek bir sürece ait olabilir.
- d) Süreçler doğrudan donanıma erişebilirken, iş parçacıkları erişemez.

15. Muteksler ve semaforlar arasındaki fark nedir?

- a) Muteksler senkronizasyon için, semaforlar ise karşılıklı dışlama için kullanılır.
- b) Muteksler ikili kilitlerdir (binary lock), semaforlar ise tamsayı sayaçlarıdır (integer counters).
- c) Muteksler, aynı anda yalnızca bir sürecin bir kaynağa erişmesine izin verirken, semaforlar birden çok sürecin aynı anda bir kaynağa erişmesine izin verebilir.
- d) Mutex'ler sistem yükü açısından semaforlardan daha verimlidir.

16. Önleyici (pre-emptive) bir zamanlama algoritmasında, bağlam anahtarlama (context switch) ne zaman gerçekleşir?

- a) Bir süreç gönüllü olarak işlemci kullanımından vazgeçtiğinde
- b) Bir süreç G/Ç beklerken bloke edildiğinde
- c) Daha yüksek öncelikli bir süreç çalışmaya hazır hale geldiğinde
- d) Daha düşük öncelikli bir süreç çalışmaya hazır hale geldiğinde

17. Süreç çizelgeleme ile ilgili olarak aşağıdakilerden hangisi doğrudur?

- a) Süreçlere bellek tahsis etmekten sorumludur.
- b) Bir sonraki adımda hangi sürecin işlemciyi kullanacağını belirler.
- c) Süreçler arası iletişimi yönetir.
- d) G/Ç işlemlerini yönetmek için kullanılır.

18. Sistem çağrısının (system call) temel amacı nedir?

Sistem çağrısının (system call) temel amacı, kullanıcı uygulamaları ile işletim sistemi arasındaki iletişimi sağlamaktır. Kullanıcı seviyesindeki uygulamalar, doğrudan donanım kaynaklarına veya çekirdek seviyesindeki işletim sistemi işlevlerine erişim sağlayamazlar. Bu erişim, güvenlik ve kararlılık

açısından sınırlandırılmıştır. Bu nedenle, bir kullanıcı uygulaması, belirli bir donanım kaynağına (örneğin, dosya sistemi, bellek, ağ, vb.) veya çekirdek işlevine erişmek istediğinde, bir sistem çağrısı yapar. Örneğin, bir dosyayı okumak veya yazmak için read ve write sistem çağrıları kullanılır. Örneğin, yeni bir süreç oluşturmak (fork), mevcut bir süreci sonlandırmak (exit) veya süreci bekletmek (wait) gibi işlemler için sistem çağrıları kullanılır. Örneğin, bellek tahsisi için malloc ve free fonksiyonları dolaylı olarak sistem çağrılarını kullanır.

19. İşletim sistemlerinde süreç denetim bloğunun (PCB) temel amacı nedir?

İşletim sistemlerinde süreç denetim bloğunun (Process Control Block, PCB) temel amacı, bir sürece ait tüm önemli bilgileri saklamak ve yönetmektir. PCB, işletim sisteminin bir süreci izlemek ve kontrol etmek için ihtiyaç duyduğu verileri içeren bir veri yapısıdır. PCB, bir sürecin mevcut durumunu (örneğin, çalışıyor, bekliyor, durduruldu vb.) saklar. PCB, her sürece atanmış tekil bir kimlik numarasını içerir. Bu kimlik, süreci diğer süreçlerden ayırt etmek için kullanılır. PCB, sürecin çalışması sırasında kullandığı CPU yazmaçlarının (program sayacı, yığın işaretçisi, genel amaçlı yazmaçlar vb.) mevcut değerlerini saklar. PCB, sürecin sahip olduğu açık dosyalar, kullanılan G/Ç cihazları ve bu cihazların durum bilgilerini içerir.

20. Bağlam anahtarlama (context switch) ve amacı nedir?

Bağlam anahtarlama (context switch), bir işletim sisteminin CPU'da çalışmakta olan bir süreçten, başka bir sürece geçiş yaparken gerçekleştirdiği işlemler bütünüdür. Bu süreç, aktif bir süreçten (process) başka bir sürece geçiş yapılmasını sağlar ve çoklu görev (multitasking) gerçekleştirilmesine olanak tanır. Sistem kaynaklarının (özellikle CPU'nun) verimli kullanımını ve çoklu süreçlerin, aynı anda çalışıyormuş gibi görünmesini sağlar. CPU'nun farklı süreçler arasında etkin bir şekilde paylaşılmasını ve sistem kaynaklarının optimum düzeyde kullanılmasını sağlar.

21. İşletim sistemlerinde kesmenin (interrupt) temel amacı nedir?

İşletim sistemlerinde kesmenin (interrupt) temel amacı, donanım veya yazılım tarafından meydana



gelen önemli olaylara hızlı bir şekilde yanıt vermek ve bu olayları işlemek için normal program akışını geçici olarak durdurmaktadır. Kesme, bir işlemcinin, belirli bir olay gerçekleştiğinde, mevcut işlemini durdurup önceden tanımlanmış bir kesme işleyicisi (interrupt handler) adı verilen özel bir rutin çalıştırmasına neden olan bir sinyaldir. Kesintiler, donanım aygıtları (örneğin, klavye, fare, ağ kartı) ve işletim sistemi arasında etkili bir iletişim kanalı sağlar. Kesintiler, farklı öncelik seviyelerinde olabilir.

22. Sistem performansı açısından mikroçekirdek mimarisinin potansiyel dezavantajı nedir?

Mikroçekirdek (microkernel) mimarisinin dezavantajı, sistem performansı açısından ek yük (overhead) getirmesidir. Mikroçekirdek mimarisinde, çekirdek minimal tutulur ve yalnızca temel işlevleri (süreç yönetimi, bellek yönetimi, IPC - süreçler arası iletişim gibi) içerir. Diğer işletim sistemi hizmetleri (dosya sistemi, ağ yığınları, sürücüler vb.) kullanıcı alanında çalışır. Her IPC çağrısı, kullanıcı modu ile çekirdek modu arasında geçiş gerektirir. Bu geçişler zaman alır ve ek yük getirir. Mikroçekirdek tasarımı, modüler yapı nedeniyle daha fazla bellek ve kaynak kullanımına yol açabilir. Mikroçekirdek yapısı, küçük ve basit sistemlerde avantajlı olabilirken, çok sayıda hizmetin ve modülün bulunduğu büyük sistemlerde yönetim ve koordinasyon sorunları yaratabilir.

23. *proc* dizini altında neler bulunur?

Linux ve Unix-benzeri işletim sistemlerinde, */proc* dizini bir sanal dosya sistemidir ve çekirdek ile ilgili bilgi ve konfigürasyonları kullanıcıya sunar. Bu dizin altında bulunan dosya ve alt dizinler, sistemin çalışma durumu hakkında anlık veriler sağlar. */proc* dizini statik değil, dinamik olarak oluşturulmuş dosya ve dizinler içerir. Her aktif süreç için, süreç kimliği (PID) ile adlandırılmış bir alt dizin bulunur. Örneğin, PID 1234 olan bir süreç için */proc/1234* dizini vardır.