Güz 2010 - Bil461
İşletim Sistemleri Finali
21 Aralık Salı

Öğrenci No:

(Ad ve Soyad Yazmayınız)

Bu sınav 10 sorudan oluşmaktadır.

Tüm işlemlerinizi gösteriniz.

Cevaplarınızı soruların altındaki boş alanlara yazınız.

Sınav süresince kendi aranızda ve sınav gözetmeni ile konuşmanız yasaktır.

Hatalı veya açık olmadığını düşündüğünüz sorulara ilişkin notlarınızı ve/veya varsayimlarinizi çözümünüzde belirtiniz.

1 12	2 6	3 2	4 0	5 1	6 3
7 8	8 2	9 4	10 2		/110

Soru 1. (4+3+3+3+4+4 puan) Aşağıdaki tanım sorularına kısa ve özlü vererek cevaplayınız.

a) Dış parçalanma (external fragmentation) nedir? İç parçalanma (internal fragmentation) nedir?

İç parçalanma - Atanan depolama alanının kullanılmayan kısmıdır. Örneğin, 1KB'lik bir sayfa verilen bir işin sadece 100B'ı kullanması.
Dış parçalanma - Atanmış (bitişik)depolama alanları arasındaki kalan kullanılamayan boşluklardır. Bu boşlukların kullanılabilmesi için birbirlerine bitiştirilmesi gereklidir.

b) Belady anomalisi (Belady's Anomaly) kavramı neyi ifade etmektedir?

Bir iş için ayrılan sayfa sayısı arttıkça sayfa hatası sayısının genelde düşmesi beklenir. Bu durumun geçerli olmadığı durumları ifade eder.

c) Hyperthreading nedir?

Az sayıda iş parçacığının donanım desteğiyle her bir vuruşta dönüşümlü olarak aynı anda çalıştırılmasıdır.
Bir çok kişi bu kavramı multithreading ile eş anlamlı olarak düşünmüş.

d) Jounaling tabanli dosyalama sistemi nedir?

Kısa süreli dosya değişiklikerin diske direk yansıtılması yerine öncelikle disk üstünde bir loga ardışık olarak yazan ve dosya güncellemerinin daha sonra yapılmasını sağlayan dosyalama sistemi türüdür.

e) *Nth chance clock* algoritmasının temel clock algoritmasına (N=1) göre bir avantaj ve bir dezavantajını belirtiniz.
Daha eski bir sayfayı belirleme imkanı, boş sayfa bulmak bazen daha uzun vakit alabilir.

f) Bir işe (process) ait iş-parçacıklarının (thread) kullandıkları sayfa tabloları aynı mıdır yoksa farklı mıdır? İşlemci bu iş-parçacıklarından birinden diğerine *context switch* ederken TLB girdilerini sıfırlamalı mıdır?

Bir işe ait iş parçacıkları aynı adres alanında çalıştıklarından aynı sayfa tablosunu paylaşmak zorundadırlar. TLB sayfa tablosu öğelerine ilişkin bir önbellek olduğu için context switch sonrasında TLB'nin sıfırlanmasına gerek yoktur.

Soru 2. **(4+5+3 puan)** Aşağıdaki verilen sorularda şıklarda verilen seçenekleri değerlendirmeniz gerekmektedir. Açıklama yapmanız beklenMİyor.
a) Bir sistemin *thrashing* durumunda olduğu tespit edildikten sonra aşağıdaki işlemlerin yapılması önerilmektedir. Hangileri problemin giderilmesinde etkili olur?

1. İşlemci bağımlı (cpu bound) işlerin sayısının artırılması
2. Daha hızlı bir işlemcinin takılması
3. Eş zamanlı olarak çalışan program sayısının azaltılması.
4. Daha büyük bir bellek takılması
5. (Bellek) Sayfa büyüklüğünün düşürülmesi.
6. Verinin daha hızlı okunmasın olanak sağlayan yeni bir diskin takılması

1,2, 5 veya 6'nın herhangi biri seçildiyse 0 puan

b) Aşağıda verilenlerin hangi kategoriye konabileceğini belirtiniz. Kategoriler: interrupt (i), exception (e) veya hiçbiri (h). Parantez içerisinde belirtiniz.

1. Zamanlayıcı - timer ( )
2. Bölütleme hatası - segmentation fault ( )
3. Klavye girdisi ( )
4. Sıfır ile bölme ( )
5. İşlem cağrısı - procedure call ( )
6. Sistem çağrısı - system call ( )

(a) Timer – (I)
(b) Segmentation violation – (E)
(c) Keyboard input – (I)
(d) Divide by zero – (E)
(e) Procedure call – (N)
(f) System call – (E)

c) Aşağıda verilen aygıtları erişim zamanlarına göre en hızlıdan, yavaşa doğru sıralayınız.
Place the following memory storage devices in order from fastest access time to slowest access time.

I. Ana bellek (RAM)
II. TLB
III. manyetik disk
IV. Yazmaçlar (registers)
V. Tape

IV, II, I, III, V

Soru 3. (4+3 puan) Bir buruğun çalıştırılmasının ortalamada 20ns sürdüğü bir bilgisayar sisteminde, ortalamada, her 1 milyon buyrukta bir sayfa hatası (page fault) oluşduğu gözlemlenmiştir. Bu sistemde bir sayfa hatasının giderilerek, buyruğun tekrar çalıştırılması için 40ms gerektiğini varsayarak, aşağıdaki soruları cevplayınız. (1ms = 10^6 ns)

a) Bu şartlar altında efektif buyruk çalıştırma süresi ne kadardır?

b) Bu sistemin performansını iyileştirmek için iki seçeneğiniz var. Ya işlemci hızını iki katında çıkaracaksınız (bir buyruğun çalıştırılması 10ns'ye düşecek), yada bellek boyutunu iki katına çıkaracaksınız (her 2 milyon buyrukta bir sayfa hatası oluşacak). Hangisini tercih edersiniz, neden?

a) Efektif buyruk çalışma zamanı = 0.000001*(20ns + 40000000ns) + 0.999999*20ns = 60ns

b) Ek bellek takarak sayfa hatası saysını düşürmek, işlemci hızını artırmaktan daha etkili olacaktır (0.0000005*(20ns + 40000000ns) + 0.9999995*20ns = 40ns).

İşlemci hızını iki katına çıkarılması = 0.000001*(10ns + 40000000ns) + 0.999999*10ns = 50ns

Dolayısıyla, belleği iki katına çıkarmak daha avantajlıdır.

*efektif zaman = hit rate * hit time + miss rate * miss time --- Bu ilişki kullanılmadan hesap yapıldıysa 0 puan.*

Miss time'da (20 ns + ...) 20 ns unutulduysa -1 puan.

Küçük matematiksel hatalar 1 puan; Büyük matematiksel hatlaar 3 puan.

Soru 4. (5 puan) LRU (least recently used) algoritmasinin bellege sayfa ekleme/cikarma ve dosyalama sistemi onbelleklerinde (buffer cache) kullanılmak istendiğinden bahsetmiştik. Buffer cache'in yönetiminde LRU algoritması kullanılırken sayfa değiştirme için LRU algoritması yerine ona yakınsayan bir algoritma kullanılmasının sebebi nedir?

LRU algoritmasının sayfalama için kullanımı pahalı bir işlemdir. Çünkü işlemcinin yaptığı her bir bellek erişiminden sonra sayfaların eskiden yeniye göre tekrar sıralanması gerekir. Bu sıralama gereksinim işlemci hızının ciddi bir şekilde düşmesine sebeb olacaktır. Buffer cache'de ise sadece açılan dosyaların inodeları ve contenti tutulmaktadır. Bir program için açılan dosya sayısı oldukça sınırlıdır ve işlemci vaktinin büyük kısmını kod, yığıt, yığın ve veri lerin bulunduğu sanal bellekte harcar. Yani, buffer cache de erişilen sayfa sayısı çok daha kısıtlıdır. Bu nedenle, buffer cache'de LRU kullanılması kabul edilebilir bir yük getirmektedir.

Hiç bir karşılaştırma yapmadan LRU algoritmasının masraflı olduğunu belirtmek (açıklamaya bağlı olmak şartıya) en fazla 2 puan.

Soru 5. (5 puan) *Unix* tabanlı bir işletim sisteminde */Users/taha/Documents/sinav.txt* dosyasına ilişkin *i-node*'u belleğe getirmek için diske kaç kere erişilmelidir? Belirtiniz. (*/Users/taha/Documents root* dizinine erişim yolunu, sinav.txt ise dosya ismini göstermektedir.) Sistemin yeni başlatıldığını dolayısıyla sadece *root* dizinine ait *i-node*'un bellekte olduğunu varsayın.

1. *root'a ait dizin içeriği (directory)*
2. *Users dizinine ait i-node*
3. *Users dizinine ait dizin içeriği*
4. *taha dizinine ait i-node*
5. *taha dizinine ait dizin içeriği*
6. *Documents'e ait i-node*
7. *Documents'e ait dizin içeriği (directory)*
8. *sinav.txt'nin i-node'u (directory)*

Toplamda 8 disk erişimi. Sadece i-node'lar sayıldıysa 2 puan.

Soru 6. (3+3+3+4 puan) 44 bit adresleme yapabilen bir bilgisayar sisteminde sayfa büyüklüğü 64KB olarak belirlenmiştir. Bu durumda:

a) Sanal adres alanı kaç sayfadan oluşmaktadır?
44 bit adres - 16 bit offset = 28 bit = 2^28= 256M sayfa

b) Bu sistemde (P)resent, (D)irty, (A)accessed, ve 2 tane de erişim kontrol (permission) bitinin tanımlı olduğunu biliyorsak, bir sayfa tablosu öğesinin boyutu en az kaç byte olabilir?
28 + 5 = 33 bit --> 5 Byte.

c) Tek seviyeli bir sayfa tablosu kullanıldığı varsayılarsa, bir işe (process) ait sayfa tablosu bellekde minimum kaç sayfa yer kaplar?
2^28 * 5B = 1.25 * 1 GB / 64 KB = 1.25 * 2^14= 20K Sayfa

d) Bu tür bir sistemde neden çok seviyeli bir sayfa tablosunun, tek seviyeli sayfa tablosuna tercih edilmesi gerektiğini belirtiniz.
Sayfa tablosunun kullanilmayan kisimlari diske atabilmek, ve böylelikte tablonun bellekde yer kaplamasına engel olmak için.

Soru 7. (4+4+4 puan) 3 sayfalık belleği olan bir sistemde aşağıda verilen sanal sayfa dizisine erişim yapılmıştır.

012412553002

Sayfa değiştirme yöntemi a) FIFO b) LRU ve c) optımal olarak seçilirse toplamda kaç sayfa hatası oluşur? (Başlangıçta belleğin bütünüyle boş olduğunu varsayın.)

FIFO	0	1	2	4	1	2	5	5	3	0	0	2
1	0			4						0		
2		1					5					2
3			2						3			

LRU	0	1	2	4	1	2	5	5	3	0	0	2
1	0			4			5					2
2		1							3			
3			2							0		

MIN	0	1	2	4	1	2	5	5	3	0	0	2
1	0			4			5			0		
2		1							3			
3			2									

Sayfa hatası sayısı, 8, 8 ve 7'dir.

Soru 8. (5+3 puan) Ortalama arama zamanı (seek time) 0.4ms olan ve veri transfer kapasitesi 100 MB/sn olan bir disk 15,000 rev/min (dönüş/dakika) hızla dönmektedir. Çok büyük bir veritabanının yazılı olduğu bu disk üstünde bir çok iş-parçacağı birbiriden bağımsız olarak işlem yapmaktadır, ve her bir işlem disk üstündeki rastegele seçilmiş 1 KB'lık blokların yazılarak güncellemesi şeklinde gerçekleşmektedir.

a) Verilen parametreler için bu diskin 1sn'de cevap verebileceği güncelleme talebi sayısını bulunuz.

Rastegele arama olduğu için toplam seek + rotation + transfer zamanı hesaplanmalı (Bu ilişki görülmediyse bu şıkdan 0 puan.), rotation hesabı 2 puan, 1KB transfer hesabı 1 puan

*.4ms (write seek) + 2ms (yarım tur atmak için gerekli zaman 60sn/(2*15K)) + .01ms (1 KB transfer zamanı) = 2.4ms. --> maks. throughput 416 ışlem/saniye (1 puan)*

b) Eğer bu disk üstünde çalışan dosyalama sisteminin log-tabanlı yada journaled olduğunu varsayarsanız, cevabınız nasıl değişir?

Bu durumda peşpeşe sektörler yazılacağında bir tek transfer zamanı göz önüne alınmalı 100 MB/sec (2 puan) --> 100,000 guncelleme saniye (1 puan).

Soru 9. (5+5+5 puan) Derste bir dosyalama sistemi değerlendirilirken kullanım amaçlarına uygun performans kriterlerinin göz önünde bulundurulması gerektiğini belirtmiştik. Bu yöntemlerden en temel 3'ü aşağıda belirtilmiştir.

1. Kesintisiz atama (continious allocatıon)
2. Linklenmiş liste tabanlı atama (linked list allocation) - FAT32/16 gibi yöntemler
3. İndeksli atama (indexed allocation)

Aşağıda verilen her durum için yukarıda verilen dosyalama yöntemlerini en uygundan, az uyguna doğru sıralayınız ve sıralamanızın sebeplerini kısaca açıklayınız.
göre sıralayınız.

a) Dosyalama sisteminde aranan en önemli kriter çok büyük boyutlu (örneğin, 1-10GB boyutlu) dosyalara dizisel (sequential) erişim performansıdır, video vb.

1 (kesintisiz), 2 (linklenmiş), 3 (indeksli)
Dizisel erişim yapılacağından 1 doğal olarak en iyi seçenektir. 2 ve 3'ün her ikiside bir sonraki bloğun belirlenmesi için ek vakte ihtiyaç duyacaktır. Ancak, indeksli atama belirli sıklıkla indeks bloklarının okunmasını (indeks blokları veri bloklarına işaret eden işaretçileri tutan bloklardır) gerektirdiğinden, linklenmiş liste tabanlı atamaya göre daha yavaş performe edecektir.
Sadece 1'e ilişkin sebep verilmişşse 2 puan

kesintisiz başa konmadıysa 0 puan.

b) Dosyalama sisteminde aranan en önemli kriter çok büyük boyutlu (örneğin, 1-10GB boyutlu) dosyalara rastgele (random) erişim performansıdır, mailbox vb.

1 (kesintisiz), 3 (indeksli), 2 (linklenmiş)

Rastgele erişim içinde en iyi seçenek kesintisiz atamadır. Aranan bloğun yeri en baştaki bloğa göre tek seferde tespit edilebilir ve blok yüklenebilir. 2 ve 3'de ise öncelikle bloğun yeri belirlenmek zorunda. 2 için bu bellek için FAT içinde pointer takip etmeyi gerektirecek, indeksli atama için ise indeks bloklarının diskten yüklenmesi gerekecek. Hangisi daha çok vakit alır?
Çok büyük bir dosyadan bahsettiğimiz durumlarda indeksli atamada çok seviyeli indeks bloklarının yüklenmesi gerekecektir. Unix tabanlı sistemlerde bu 3 adet indeks bloğunun yüklenmesini gerektirecek. Yani yaklaşık olarak 30ms. Linki atamada ise, blok boyu 1-4K arasında varsayılırsa yaklaşık 1M bellek eriişimi gerektirecektir. Bu da yaklaşik olarak
50ns*10^6=10^4-5ms kadar zaman gerektirecektir.

linked list başa konduysa veya kesintisiz sona konduysa 0 puan

c) Dosyalama sisteminde aranan en önemli kriter disk kapasitesinin en yüksek utilizasyonudur. (Yani, diske en çok dosya veya en büyük boyuttaki dosyaların sığdırabileceği durum.)

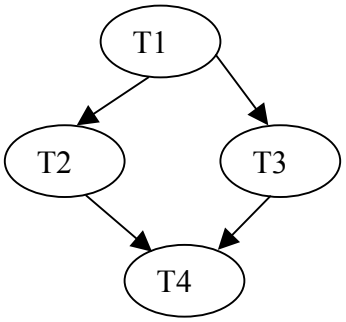
2 (linklenmiş), 3 (indeksli), 1 (kesintisiz)

Dış parçalanmaya neden olacağından bu durumda en kötü seçenek kesintisiz atama olacaktır. Ancak, küçük dosyalar ve büyük blok boyutları için 2 ve 3'den de iyi olabilir. 3 için ise tutulması gereken metadata (indeks blokları) 2'ye göre daha fazla olacağından, 3 daha avantajlıdır. 3'de tutulması gereken tek metadata bir sonraki bloğun adresi olduğu için disk üstünde daha az yer tutar. Dış parçalanmadan bahsedilmişşse 2 puan.

kesintisiz başa konduysa 0 puan

Açıklama doğru değilse veya hiç bir açıklama yapılmadan verilen cevaplar 0 puan.

Soru 10. (12 puan) Dört iş-parçacığı (T1, T2, T3, T4) arasındaki ilişkiyi gösteren grafik aşağıda verilmiştir. Bu grafikte bir iş parçacığından, Tx, diğerine, Ty, yönelmiş bir ok, Ty'nin başlaması için Tx'in bitmesi gerektiğini göstermektedir. İş-parçacıklarının herhangi bir sırayla çalıştırılmaya balayabileceğini varsayın. Semafor primitifini kullanarak bu grafikte verilen ilişkinin gerçeklenmesini sağlayınız. Semaforların ilk değerlerini ve her bir semafor üzerinde yapılan işlemleri aşağıda verilen kutular içerisinde gösteriniz.




```
/*Tanimlamalar ve ilk degerleme*/
semaphore semA=0;
semaphore semB=0;
```

void T1(void)
{
 	
/*Hesaplama */
 	
....
 	
/*Hesaplama sonu*/
semA.V();
semA.V();
}
	void T2(void)
	{
	
	semA.P();
	/*Hesaplama */
	
	....
	
	/*Hesaplama sonu*/
	
	semB.V();
	}
		void T3(void)
		{
		
		semA.P();
		/*Hesaplama*/
		
		....
		
		/*Hesaplama sonu*/
		
		semB.V();
		}
			void T4(void)
			{
			
			semA.P();
			semA.P();
			/*Hesaplama */
			
			....
			
			/*Hesaplama sonu*/
			
			}

Eğer 3 semafor ile çözüldüyse tam puan, fazladan semafor kullanımı için puan kırılmıştır.