PAMUKKALE ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
CENG 305 – İşletim Sistemleri
EĞİTİM ÖĞRETİM YILI GÜZ DÖNEMİ DÖNEM SONU SINAVI
Süre:85dk

Öğrenci No: **Ad:** Emre

Soyad: Şen

İmza:

PAÜ CENG 305 İşletim Sistemleri Dönem Sonu Sınavı

!4 veya 5'inci sorulardan sadece birini cevaplayınız, cevaplamadığınızı puan tablosundan çarpı işareti ile işaretleyiniz!

Soru	1 (25p)	2 (25p)	3 (25p)	4 (25p)	5(25p)	Toplam
Puan						

1.

a) Aşağıdaki ifadeleri doğru yanlış olarak nitelendiriniz ((D) ya da (Y) olarak.).

Page size büyüklüğündeki azalma daha düşük boyutlu page tablosu gereksinimi oluşturur. (Y) yanlış

Page size büyüklüğündeki azalma daha çok TLB miss oluşumuna neden olur. (D) Doğru

Thread yaratmak process yaratmaktan daha az maliyetlidir. (D) Doğru

Threadler arası context switch processler arası context switchten maliyetlidir. (Y) Yanlış

Pratikte *optimal page replacement* algoritması en iyi seçimdir. (Y) Yanlış

İşletim sistemi aynı kaynağa erişmek isteyen processleri denetlemekten sorumlu değildir. (Y) Yanlış

System call'ları işlemcinin öncelik (privelege) modunu değiştirmemektedir. (Y) Yanlış

Hyperthreading ile I/O için bekleyen threadler yerine başka threadler çalıştırılarak eş zamanlı çalışan thread sayısı fazla gösterilebilmektedir. (D) Doğru

b) Page tablosu memoryde depolanan bir sayfalama sistemimiz olduğunu düşünelim. Memorye erişme 200 nanosaniye sürüyorsa memoryde bulunan bir sayfaya erişmek ne kadar sürer?

200 + 200 = 400
(tablosu erişim, bulunan için)

Eğer sisteme TLB eklediğimizi varsayarsak ve %75 olasılıkla sayfalar TLB de bulunuyorsa memorydeki bir sayfaya erişmenin efektif süresi nedir? (TLB'ye erişmek 0ns alıyor.)

0.75 (0 + 200) + 0.25 (0 + 200 + 200) = 250
(TLB erişim, Frame erişim; TLB erişim, bulunduğu Page erişim, istediğimiz Frame erişim)

2. Sistemde 10 disk drive'ı ve 4 adet processin olduğunu varsayalım. t_0 anında sistemin durumu aşağıdaki şekildedir. Bu durumda sistem safe durumda mıdır? Öyleyse bunu kanıtlayan bir sequence gösteriniz. Değilse de nedenini açıklayınız.

Processes	Max	Allocation

Proses	max	Allocating
P_0	6	1
P_1	4	3
P_2	1	0
P_3	8	5

Max - Allocations = Need

Need	
P_0 5	X
P_1 1	V
P_2 1	V
P_3 3	X

Available = 1

$\leftarrow$ Available $\geq$ Need

allocation'dan eklenir

Güvenlidir

1 tane per bulsak

kilit sekerlidir

P_1 çalışır Available = 1+3 = 4

P_2 çalışır Available = 4+0 = 4

P_3 çalışır Available = 4+5 = 9

P_0 çalışır Available = 9+1 = 10

$P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_0$

3. ***(Virtual Memory)***

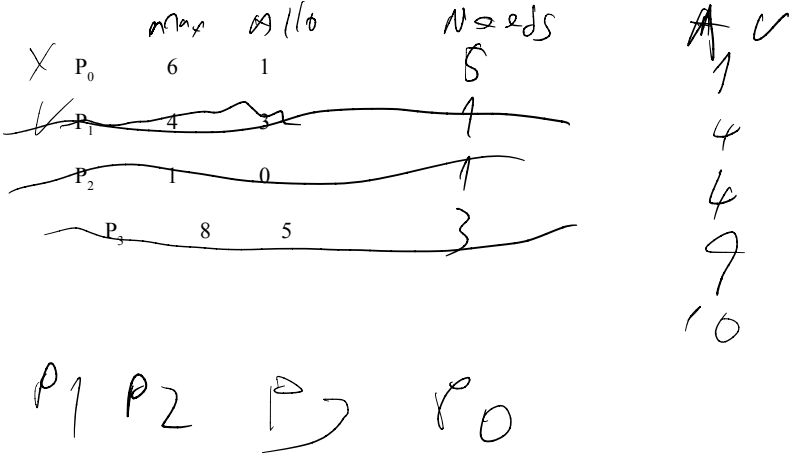
a. FIFO page replacement algoritmasını aşağıdaki referans string'ine uyguladığımızı düşünelim.

1 2 3 4 1 2 5 1 2 3 4 5

Virtual memorydeki frame sayısı üçten dörte yükseldiğinde **page fault** sayısı artar mı, azalır mı ya da aynı mı kalır?

1	1	1	4	4	4	5	5	5
	2	2	2	1	1	1	3	3
		3	3	3	2	2	2	4

b. Aşağıda dört frame'lik kapasitesi bulunan bir main memory'nin herbir frame'inde bulunan page'lerin id'si, belleğe yüklenme anı, CPU tarafından en son kullanıldığı an ve son 6 mslik periyodda kullanıp kullanılmadığını gösteren R bitini görüyoruz.



3. ***(Virtual Memory)***

a. FIFO page replacement algoritmasını aşağıdaki referans string'ine uyguladığımızı düşünelim.

1 2 3 4 1 2 5 1 2 3 4 5

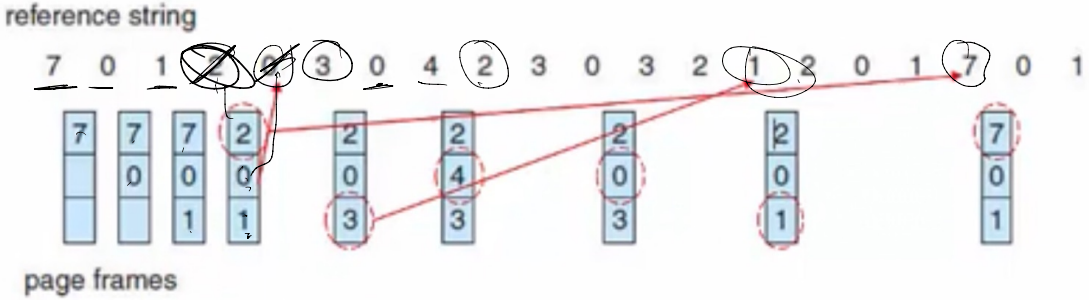
Virtual memorydeki frame sayısı üçten dörte yükseldiğinde **page fault** sayısı artar mı, azalır mı ya da aynı mı kalır?

b. Aşağıda dört frame'lik kapasitesi bulunan bir main memory'nin herbir frame'inde bulunan page'lerin id'si, belleğe yüklenme anı, CPU tarafından en son kullanıldığı an ve son 6 mslik periyodda kullanıp kullanılmadığını gösteren R bitini görüyoruz.

Mevcut sayfalar üzerine yerleştirme

Optimal page replacement algoritması

- Bu algoritmada, **en uzun süre kullanılmayacak sayfa ile yer değiştirme yapılır**.
- Aşağıdaki hafıza erişim serisinde **9 page fault** ile yer değiştirme yapılmaktadır.



Page	Yüklenme Anı (ms)	En son Kullanılma anı (ms)	R
0	126	280	1
1	230	265	0
2	140	270	0
3	110	285	1

Buna göre sisteme memoryde bulunmayan yeni bir page yüklemek istediğimizde;

FIFO, LRU ve second chance algoritması kullanıldığında bellekten atılacak page hangisi olacaktır? Sırasıyla belirtiniz.

4. ***(Process Syncronization)*** 10 tane maymunun halattan yapılmış dar bir köprüden geçtiği bir problem üzerinde düşünmeniz isteniyor. Köprünün batısında muz ağaçları, köprünün doğu tarafında ise gölge yapma kapasitesi yüksek ağaçlar bulunuyor. Maymunlar ilk başta doğuda bulunuyorlar ve muz yemek için köprü üzerinden batıya geçiyor, orada muz yedikten sonra gölgede biraz uyumak için köprü üzerinden doğu tarafa geçip her biri farklı sürelerde (rastgele) uyuyor. Uyanınca maymunlar tekrar muz yiyor ve bu süreç bu şekilde tekrarlanıyor. Bu

sistemde maymunlar karşı tarafa geçerken çeşitli problemler yaşıyor: 1) iki tane maymun köprüde farklı yöne geçecek şekilde karşılaşırsa *deadlock* oluşuyor. (Yani maymunlar geriye dönmeyi ya da kenardan geçmeyi gerçekleştiremiyor.) 2) Halattan yapılmış köprü maksimumda 5 tane maymunu taşıyabiliyor, daha fazlasında köprü kopuyor. Bu iki problemin oluşmayacağı maymunların barış içinde uyuyup muz yiyebileceği bir *process* senkronizasyon algoritmasını semaforlar kullanarak tasarlayınız.

5. **(Multithreaded)** Bu soruda girilen bir sayının mükemmel bir sayı olup olmadığını bulmak için bir multi-threaded program yazmanız istenmektedir (C#, Java ya da C kullanabilirsiniz.) . Mükemmel bir sayı kendisi hariç çarpanlarının toplamına eşit olan sayıdır. Örneğin 6 ve 28 birer mükemmel sayılardır. Dışardan girilen bir sayı (N) ve yine dışarıdan girilen thread sayısına (T) göre bu girilen sayının (N) mükemmel olup olmadığını girilen T sayısı kadar thread yaratarak bulan multi-threaded bir uygulama yazınız.

Başarılar.