

BIL 301
2022 Güz Dönemi
Bütünleme Soruları

- **Fotoğraflı üniversite kimliği** olmayan bir öğrenci bu sınava giremez. Üzerinde kimliği olmayanların fotoğrafları çekilip isimleri not alınacaktır (kimliklerinin sonradan teyidi için).
- Her türlü kağıt, kitap, not ve hesap makinesi, telefon, saat vb. her türlü elektronik aygıt kullanımı **kopya** olarak nitelendirilecektir.
- Sınav süresince her türlü kalem, silgi vb. paylaşımalar yine **kopya** olarak nitelendirilecektir. Silgisi olmayanlar cevapların üzerini karalayıp farklı bir boş alanı kullanabilirler.
- **TOPLAM SÜRE 70 DAKİKADIR.**
- Toplam da 102.5 puanlık soru olmakta iken alınabilecek maksimum not 100'dür: 2.5 puan bonus olarak düşünlülmüştür.
- Cevaplarınızı okunaklı bir şekilde ayrılan kısımlara yazınız.
- Gerektiğinde diğer boş kısımlarında kullanabilirsiniz.
- Hatalı soru tespitinde soru iptal edilecektir. Kalan puan maksimum alınabilecek not olacaktır.

İsim ve Soyisim : _____

Medeniyet ID : _____

İmza : _____

Soru	Puan	Alınan skor
1	7½	
2	5	
3	5	
4	5	
5	5	
6	5	
7	5	
8	5	
9	15	
10	15	
11	15	
12	15	
Toplam:	102½	

1. (read-copy-update) RCU ile ilgili aşağıdakilerin doğru veya yanlışlığını T/F koyarak belirtiniz. (Eng: For the following statements about (read-copy-update) RCU technique, fill in the blanks with false/true to indicate whether each statement is true or false.)

I False Writer-reader çekişmeleri normal senkronizasyon (lock gibi) yöntemleri kullanılarak yapılır. (Eng: write-read conflicts are managed(synchronized) with regular techniques (e.g. locks))

II True Writer-writer çekişmeleri normal senkronizasyon (lock gibi) yöntemleri kullanılarak yapılır. (Eng: write conflicts are managed(synchronized) with regular techniques (e.g. locks))

III False Reader-reader çekişmeleri normal senkronizasyon (lock gibi) yöntemleri kullanılarak yapılır. (Eng: reader-reader conflicts are managed(synchronized) with regular techniques (e.g. locks))

2. Aşağıdaki statementları, boşlukları doldurarak, açıklamalarla eşleştiriniz (match the following statements with the explanations by filling the blanks.).

1. `asm volatile("mfence" ::: "memory");`

2. `asm volatile("::: "memory");`

(a) 2 Compiler'in StoreLoad instructionları yeniden sıralamasına (memory ordering yapmasına) izin vermez. (Eng: This statement prevents compiler from reordering store followed by a load (prevents Compiler reordering))

(b) 1 CPU'nun StoreLoad instructionları yeniden sıralamasına (memory ordering yapmasına) izin vermez. (Eng: This statement prevents CPU from reordering store followed by a load (prevents CPU reordering))

3. User spaced thread library (posix gibi) gerçekleştirilirken interrupt disable/enable yöntemi kullanılması doğru olur mu? Security ve multiple CPU ortamlarını düşünerek, evet/hayır nedenini açıklayınız. Doğru açıklaması olmayan cevaplar değerlendirilmeyecek. (Eng: Would it be alright to implement a lock (synchronization) mechanism in user space by using interrupt enable/disable mechanism? Explain your answers by considering security and multiple CPU. The answers without correct explanations will not be graded.).

4. FAT dosya sisteminde random accesslerin, linked dosya sistemine göre genelde neden daha hızlı olduğunu açıklayınız. (Eng: Explain why random accesses are in general faster in FAT system than linked file system where the file blocks are linked in the disk.)

5

5. Dosya sisteminin recovery edilmesinde kullanılan journaling yöntemini açıklayınız. (*Explain how journal technique is used in the file system recovery mechanism.*)

5

6. Hiyerarşisel page table kullanmanın avantaj ve dezavantajını yazınız. (Eng: Write the advantage and disadvantage of using a hierarchical page table.)

5

7. **DAC (discretionary access control)** ve **MAC (mandatory access control)** access (erişim) kontrollerine birer örnek veriniz. (eng: *Explain DAC and MAC by giving an example for each.*)

(a) DAC:

(b) MAC:

5

8. Aygıtlara erişimde polling ve interrupt mekanizmasının hangi durumlarda kullanılması uygundur. (Eng: *when accessing devices, explain under which conditions using the polling and under which other conditions using the interrupt mechanism are more appropriate.*)

(a) interrupts:

(b) polling:

9. Aşağıda stack based bufferoverflow ataklarına ve hatalarına karşı koruma mekanizmaları verilmiştir
kısaca açıklayınız. (Eng: Briefly explain the following prevention mechanism against stack based
bufferoverflow?)

5 (a) Canaries (kanaryalar)

5 (b) Address Space Layout Randomization (adres uzayının rastlaştırılması)

5 (c) Non-executable stack

10. Aşağıdaki page (sayfa) referans metni göz önüne alarak (Eng: Consider the following page reference string) :

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3

Demand (istek) Paging'ın 4 framele yapıldığını varsayırsak, aşağıdaki frame değiştirme algoritmalarında kaç tane page fault olur? (Eng: Assuming demand paging with **three frames**, how many page faults would occur for the following replacement algorithms?)

5 (a) Optimal replacement algorithm

- A. 4
- B. 5
- C. 6**
- D. 7

5 (b) LRU (yakın geçmişte en az kullanılan, eng: the least recently used) algorithm

- A. 4
- B. 5
- C. 6**
- D. 7

5 (c) FIFO algorithm

- A. 4
- B. 5
- C. 6
- D. 7**

11.

```
int A[100][100]; /*sizeof(int) --> 2-byte*/
a.
for (int j = 0; j < 100; j++)
    for (int i = 0; i < 100; i++)
        A[i][j] = 0;
b.
for (int i = 0; i < 100; i++)
    for (int j = 0; j < 100; j++)
        A[i][j] = 0;
```

Yukarıdaki kodda tanımlanan A matrisi a. ve b. deki şekilde referans edilmektedir (döngü indekslerinin sırası farklı). Eğer büyülüğu 200 byte olan üç tane page frame varsa ve bu framelerden birisi programın kodunu tutuyorsa, (diğer iki frame boş ve a. ve b. ayrı ayrı çalıştırılıyor) (*Eng: Consider the above code, if we use 3 frames and assume frame-1 holds the code and the other two frames are empty:*)

- 5
- (a) LRU algoritması kullanıldığımda kaç tane page fault oluşur? (*Eng: with LRU algorithm how many page faults are expected while executing a. and b. seperately?*)

5

 - (b) FIFO kullanıldığımda kaç tane page fault oluşur? (*Eng: with FIFO algorithm how many page faults are expected while executing a. and b. seperately?*)

5

 - (c) Optimal algoritmada kaç pagefault oluşur? (*Eng: with optimal algorithm how many page faults are expected while executing a. and b. seperately?*)

12. Aşağıda processlerin geliş zamanları ve beklenen çalışma (burst) zamanları verilmiştir:

Process	Arrival(varış)	Time(zaman aynı olunca yukarıdaki önce gelmiş oluyor: eng: if it is the same, the upper process comes before the lower)	Burst time
P_1	0		5
P_2	1		4
P_3	2		3
P_4	3		2
P_5	4		1

5

- (a) FCFS algoritması kullanıldığında, **non-preemptive** sistemde processlerin CPUDA çalışmaları **Gantt chart çizerek gösteriniz.** (eng: When FCFS algorithm used in non-preemptive system, show the CPU usage of the processes by drawing a **Gantt chart**.)
- 5
- (b) Shortest job first (SJF) algoritması kullanıldığında, **non-preemptive** sistemde processlerin CPUDA çalışmaları **Gantt chart çizerek gösteriniz.** Zamanlar eşit olunca öncelik yukarıdaki process'e veriliyor. (eng: When SJF algorithm used in non-preemptive system, show the CPU usage of the processes by drawing a **Gantt chart**. When the times are the same, the upper process (the lower index) has the priority.)
- 5
- (c) Shortest remaining time first (SRT, preemptive SJF) algoritması kullanıldığında, processlerin CPUDA çalışmaları **Gantt chart çizerek gösteriniz.** Zamanlar eşit olunca öncelik yukarıdaki process'e veriliyor. eng: When SJF algorithm used in non-preemptive system, show the CPU usage of the processes by drawing a **Gantt chart**. When the times are the same, the upper process (the lower index) has the priority.