

## BIL 301 2024 Güz Dönemi Final 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. paylaşımlar yine **kopya** olarak nitelendirilecektir. Silgisi olmayanlar cevapların üzerini karalayıp farklı bir boş alanı kullanabilirler.
- TOPLAM SÜRE 40 DAKİKADIR.
- Toplam da 101 puanlık soru olmakta iken alınabilecek maksimum not 100'dür: 1 puan bonus olarak düşünülmüştür.
- Hatalı soru tespitinde soru iptal edilecektir.

İsim ve Soyisim : \_\_\_\_\_

Medeniyet ID : \_\_\_\_\_

İmza : \_\_\_\_\_

Soru	Puan	Alınan skor
1	10	
2	4	
3	4	
4	4	
5	26	
6	4	
7	4	
8	4	
9	6	
10	6	
11	4	
12	4	
13	4	
14	12	
15	16	
<b>Toplam:</b>	<b>112</b>	

6

1. Bir bilgisayar sisteminde **Alice, Bob ve Cyndy** isiminde üç kullanıcı ve bunlara ait **alicerc, bobrc, ve cyndyrc** dosyaları bulunmaktadır. Alice'e ait alicerc dosyasını Bob ve Cyndy okuyabilmektedir. Cyndy ayrıca Bob'a ait bobrc üzerinde hem okuma hem de yazma işlemi yapabilmektedir. Ancak Alice, bobrc üzerinde sadece okuma yapabilmektedir. Yine Cyndy'ye ait cyndyrc üzerinde sadece Cyndy okuma ve yazma işlemi yapabilmektedir. Son olara her dosyanın sahibi kendi dosyasını execute edebilmektedir. Bu sistem için erişim kontrol matrisi (access control matrix) çiziniz. İzinler için o, r, w, e kullanınız. *Eng: Consider a computer system with three users: Alice, Bob, and Cyndy. Alice owns the file alicerc, and Bob and Cyndy can read it. Cyndy can read and write the file bobrc, which Bob owns, but Alice can only read it. Only Cyndy can read and write the file cyndyrc, which she owns. Assume that the owner of each of these files can execute it.*

**Her hatalı o, e, r, veya w -1 puan.**

Solution:		alicerc	bobrc	cyndyrc
	Alice	o,e	r	
	Bob	r	o,e	
	Cyndy	r	r,w	o,e,r,w

2

- (a) Access control list, access control matrisi sütun sütun bölünerek (is created by dividing access control matrix column-wise);

2

- (b) Capability list, satır satır bölünerek oluşturulur(is created by dividing it row-wise).

4

2. Hangisi **policy** hangisi ise **mechanism** (mekanizma)dır (*Eng: which one is policy and which one is mechanism*)? **Yanlış doğruyu götürmektedir.**

- (a) Policy Dosyalara erişimde bir kullanıcının dosyaya hangi şartlarda erişebileceği kuralı(*Eng: The rules that define under what conditions, a file can be accessed by a user.*)
- (b) Mechanism Access control matrix ile dosya erişim kural ve izinlerinin kontrol edilmesi (*Eng: Using an access control matrix to determine/check a file access permissions/rules.*)

4

3. Hangisi **DAC (discretionary access control)** hangisi ise **MAC (mandatory access control)** access (erişim) kontrolüdür. (*Eng: which one is an example of DAC (discretionary access control) and which one is an example of MAC (mandatory access control)*)? **Yanlış doğruyu götürmektedir.**

- (a) MAC Kullanıcıların dosyalara erişimlerinin sistem (yani admin) tarafından kontrol edilmesi. (*eng: All permissions for the users' access to files determined by the system (admin).*)
- (b) DAC A kullanıcısının yeni kullanıcı B nin kendi dosyaları üzerindeki erişim izinlerini belirlemesi (*Eng: User-A determines the access type of a new user-B on user-A's files.*)

4

4. Aşağıdakilerden hangisi bufferoverflow ataklarına ve hatalarına karşı koruma mekanizmalarından birisi **değildir**? (*Eng: Which of the following does not describe a bufferoverflow protection mechanism?*)

- A. **Speculative execution (spekülatif çalıştırma)**
- B. Canaries (kanaryalar)
- C. Address Space Layout Randomization (adres uzayının rastlaştırılması)
- D. Non-executable stack

26

5. Aşağıdaki cümleleri doğru/yanlış olarak işaretleyiniz. *Eng: Mark the statements below true or false.* **Bu kısımda 1 yanlış 1 doğruyu götürmektedir. Eng: 1 wrong answer cancels 1 correct one.**

- (a) Disk üzerindeki dosya bloklarının linkler kullanılarak bağlanması ve takip edilmesiyle kullanılan allocation metoduyla ilgili hangileri söylenebilir *Eng: what can be said about the linked allocation method used for disk files:*

- (a) True external fragmentation olmaz. *Eng: no external fragmentation*
- (b) False rastgele erişim zamanı hızlıdır. *Eng: Fast random access*
- (c) True dosya için yeni bloklar eklenmesi çıkarılması kolaydır. *Eng: easy to modify for dynamical growth.*
- (d) True FAT ta linkler tablede saklanmaktadır (*in FAT, links saved in table*).
- (e) True İndeks allocation metodunda indeksler(pointerlar) arrayde tutulduğu için rastgele erişim hızlıdır. (*random accesses are fast in indexed allocation method*)
- (f) False Unix UFS de **büyük dosyalar** için **tek seviye**; küçük dosyalar için çok seviyeli indeks array (tablo) tutulur. *Eng: UFS keeps mult-level indexed blocks for small files(data) and single level for large files.*
- (g) True Clock algoritması LRU algoritmasını implement eden yaklaşık bir algoritmadır. *Eng: Clock algorithm is an approximation to LRU algorithm.*
- (h) False Memory idaresinde paging kullanılması hem internal hem de external fragmentationı ortadan kaldırır. *Eng: No internal and external fragmentation occur in memory management based on paging.*
- (i) False MMU page fault verdiğinde, OS processi sonlandırır ve processe ait tüm memory pageleri free eder. *Eng: After a page fault from MMU, OS terminates the process and frees all memory used by the process.*
- (j) False TLB flash diske erişimi hızlandırmada kullanılan bir yöntemdir. *TLB is method used to speedup access times to a flash disk.*
- (k) True Context switchlerde TLB flush edilir. *Eng: In context switches, TLB is flushed.*
- (l) True Hiyerarşik paging kullanılması çalışılan page table boyutunu küçültürken, TLBde olmayan yeni adres çevrimlerinde daha fazla memory erişimi gerektirir. *While hierarchical page tables makes the considered table size smaller; for new addresses not in TLB, it requires more memory accesses.*
- (m) False Multi core ortamlarda, çok fazla writer ve sadece bir kaç reader'ın olduğu durumlarda read-copy-update(RCU) senkronizasyon yöntemi kullanılması data erişimini hızlandırır. (*in multi core env, when there many writers but only a few readers, using RCU makes data accesses faster*)

4 6. FAT16 da her block(cluster) 64 KB ise dosya sisteminin maksimum büyüklüğü yaklaşık ne olur? *Eng: What would be the max size of file system when each block (cluster) size 64KB in FAT16 system.*  
 $64 \times 2^{16} = 65,536 \times 64KB \approx 4GB$

4 7. Hatalarda datanın dosya sisteminin kurtarılmasıyla alakalı olanları işaretleyiniz. *Eng: which of the followings are related to crash recovery of a FS?* **her yanlış seçim bir doğruyu götürmektedir.**

- ☒ **fschk**
- ☐ cache coherence
- ☐ time of check-time of use attack
- ☐ speculative execution
- ☒ **soft updates**
- ☒ **journaling**
- ☒ **LFS**

4 8. Atomic işlemlerde memory order kısıtlaması yapmak için kullanılan **memory\_order\_release**, **memory\_order\_acquire**, **memory\_order\_relaxed** kullanılarak aşağıdaki kod dizayn edilmiştir.

```
struct message msg_buf;
_Atomic(_Bool) msg_ready;
void send(struct message *m) {
```

```

msg_buf = *m;
/*0*/
atomic_thread_fence(memory_order_release);
/*1*/
atomic_store_explicit(&msg_ready, 1, memory_order_relaxed);
/*2*/
}
struct message *recv(void) {
    _Bool ready = atomic_load_explicit(&msg_ready, memory_order_relaxed);
    /*3*/
    if (!ready) return NULL;
    atomic_thread_fence(memory_order_acquire);
    /*4*/
    return &msg_buf;
}

```

Aşağıdakilerden hangisi **yanlıştır**? *Eng: Which of the followings is **wrong** related to above code?*  
**Her yanlış seçim bir doğruyu götürmektedir.**

- ☐ 3 noktasında bulunan herhangi bir okuma/yazmanın sırası kendi aralarında değişebilir. *(Eng: any read/write operations in region 3 can be reordered. They are still in the same region.)*
- ✓ **3 veya 4 noktasında bulunan herhangi bir okuma/yazma 4 veya 3 bölgesine taşınabilir.** *(Eng: any read/write operations in region 3 and 4 can be reordered to the region 4 or 3.)*
- ☐ 1 noktasında bulunan herhangi bir okuma/yazmanın sırası kendi aralarında değişebilir. *(Eng: any read/write operations in region 1 can be reordered. They are still in the same region.)*
- ✓ **0 noktasında bulunan okuma/yazma işleminin sırası 1 noktasında bulunan yazma işleminden sonraya alınabilir.** *(Eng: any read/write operation in region 0 can be reordered after a write operation in region 1.)*

**Solution:** acquire - no reads or writes in the current thread can be reordered before this load.

release - no reads or writes in the current thread can be reordered after this store.

The rest can be reordered!!! source: [https://en.cppreference.com/w/cpp/atomic/memory\\_order](https://en.cppreference.com/w/cpp/atomic/memory_order)

6

9. Herhangi bir lock üzerinde acquire/release implementasyonu ile ilgili söylenen yukarıdakilerden hangileri doğrudur? *Which of the followings can be said about the implementation of acquire/release semantics for locks?* **Her yanlış seçim bir doğruyu götürmektedir.**

- ✓ **Kernel içinde kullanılmak için interrupt disable/enable mekanizmasıyla yazılabilirler,**
- ☐ Kernel içinde kullanılmak için interrupt disable/enable mekanizmasıyla yazılırken lock değişkeninin sadece değeri değiştirilirken interrupt disable/enable yapılması mutual exclusion prensibinin bozulmasına sebep olur. *In kernel implementation, only disabling interrupts when changing the value of a lock variable leads to violation of mutual exclusion principal.*
- ✓ **interrupt disable/enable mekanizmasıyla lock implementasyonu user space kütüphanelerde sistem kontrolü yönünden sorun olacağı için kullanılmaz** *(Eng: in the user space libraries interrupt disable/enable mechanisms for locks can lead to serious problems and so should not be used.)*

- ✓ User space kütüphanelerde atomic read-write-update işlemleri ve futex system call kullanılarak implementasyonu gerçekleştirilebilir (*Eng: They can be implemented in user space libraries by using atomic read-write-update operations with futex system call.*).

6 10. Aygıtın servis edilmesinde kullanılan polling ve interrupt yöntemleriyle ilgili söylenenlerden hangileri doğrudur? (*eng: Which of the above is true about interrupt and polling methods used to service a device?*) **Yanlış doğruyu götürmektedir.**

- ✓ Interrupt kullanıldığında interrupt handler aygıtı servis ettiği için context switch yapılmaktadır. Polling de aygıtı bizzat CPU servis eder. *Eng: When interrupts used, the interrupt handler services/works with the device; so a context switch is done. In polling, CPU services the device.*
- ✓ Pollingde CPU düzenli kısa aralıklarla aygıtın servis edilip edilmediğini belirlemek için statusünü kontrol ettiği için bir çok CPU cycle'i boşa harcar. *Eng: in polling, because CPU waits and checks the status of a device at regular short intervals to see if it needs to be serviced, it wastes many of the CPU cycles.*
- Polling'in pek fazla servis edilmeyi bekleyen aygıt bulunamadığında veya geliş zamanı belirsiz küçük data transferlerinde kullanılması daha verimlidir. *Eng: polling becomes more efficient when CPU rarely finds a ready-device to be serviced or when there are infrequent data transfers.*
- ✓ interrupt yöntemi pek fazla servis edilmeyi bekleyen aygıt bulunamadığında veya geliş zamanı belirsiz küçük data transferlerinde kullanılması daha verimlidir. *Eng: interrupt driven method becomes more efficient when CPU rarely finds a ready-device to be serviced or when there are infrequent data transfers.*

4 11. Hangisi thrashing sebeplerinden birisi **değildir**? *Which one is NOT a reason for thrashing?*

- A. Sistemde çok fazla process olması *Eng: too many process in the system*
- B. Processin anlık çalışma setinin fiziksel memoriye sığmaması. *Eng: Process hot/working memory does NOT fit physical mem.*
- C. Processin locality prensibine uymayacak şekilde yazılması. *Eng: Process access pattern does not have locality.*
- D. Belady anomaly

4 12. Hangisi demand pagingde memory erişimlerinin verimliliğini artırır? (*Eng: Which of the above increase memory access time efficiency in demand paging?*) **Her yanlış seçim bir doğruyu götürmektedir.**

- ✓ Memory adreslerinin çok geniş ve seyrek olduğu yani çok büyük memory adres uzayının sadece belirli kısımlarının kullanıldığı durumlarda multilevel page table kullanmak (*Eng: Using multilevel page table when the adress space is huge and sparse*)
- ✓ TLB (Translation Lookaside Buffer) kullanma
- FTL ile flash wearing azaltma.
- Cache coherence

4 13. Bir processin çok CPUlu bir ortamda çalışma zamanı incelendiğinde, zamanın %90'nının I/O işlemlerine ve disk erişimlerine harcadığı tespit edilmiştir. Aşağıdakilerden hangisi söylenebilir? (*Eng: Investigation of a process execution time in multi-CPU environment reveals that 90% of the time is spent on I/O operations and disk accesses. Which of the following statements can be said about this system and the process?*) **Yanlış doğruyu götürmektedir.**

- ✓ I/O işlemlerinde DMA kontrolör kullanılması sistemdeki concurrencyi artıracığından sistem performansını artırabilir (*Eng: Using DMA controller to do I/O tasks may increase system concurrency and so the system performance.* )

- ✓ Disk erişimlerine harcanan zaman çok fazla olduğu için thrashing olabilir. (Eng: Since the ratio of time spent on disk accesses is large, there may be thrashing happening in the system.)
- ✓ Eğer thrashing varsa, processe ayrılan frame sayısının büyük oranda artırılması ile disk erişimlerine harcanan zaman azaltılabilir (Eng: If there is thrashing, increasing number of frames by a large factor may decrease the time spent for disk accesses.).
- ✓ Eğer thrashing varsa thread sayısının azaltılması performansı artırabilir. (Eng: If there is thrashing, decreasing number of threads may increase the performance.)

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

7, 1, 1, 7, 6, 4, 0, 1, 6, 3, 4, 0, 3, 2, 4, 7, 4, 0, 1, 1, 4

**Demand (istek) Paging**'ın **5 framele** yapıldığını varsayarsak, aşağıdaki frame değiştirme algoritmalarında kaç tane page fault oluşur? (Eng: Assuming demand paging with **5 frames**, how many page faults would occur for the following replacement algorithms? )

4

(a) LRU (yakın geçmişte en eski kullanılan, eng: the least recently used) algorithm

**Solution:** 9 tane, 4 puan doğru cevap; cevap yanlış, yapılanlar algoritmaya göre mantikli ise 2 puan.

	7	1	1	7	6	4	0	1	6	3	4	0	3	2	4	7	4	0	1	1	4
LRU																					
7										3				3		3			1		
1										1				2		2			2		
6										6				6		7			7		
4										4				4		4			4		
0										0				0		0			0		
5 pf										1pf				1pf		1pf			1pf		

4

(b) FIFO algorithm

**Solution:** 10 tane, 4 puan doğru cevap; cevap yanlış, yapılanlar algoritmaya göre mantikli ise 2 puan.

	7	1	1	7	6	4	0	1	6	3	4	0	3	2	4	7	4	0	1	1	4
FIFO																					
7										3				3		3			3		3
1										1				2		2			2		2
6										6				6		7			7		7
4										4				4		4			1		1
0										0				0		0			0		4
5 pf										1pf				1pf		1pf			1pf		1pf

4

(c) Optimal algorithm

**Solution:** 7 tane. 4 puan doğru cevap; cevap yanlış, yapılanlar algoritmaya göre mantikli ise 2 puan.

	7	1	1	7	6	4	0	1	6	3	4	0	3	2	4	7	4	0	1	1	4
Optimum (gelecekte en sonraya kalan)																					
7										7				7							
1										1				1							
6										3				2							
4										4				4							
0										0				0							
5 pf										1pf				1pf							

15. Aşağıda diske erişim için gelen requestler verilmiştir. Disk başının (disk head) **başlangıçta 63'te** olduğunu düşünürsek, verilen algoritmalarda erişim için sıralama nasıl olur? (Eng: Consider the following disk I/o requests. If the head is at 63, what would be the access order with the given algorithms? ) :

60, 71, 147, 17, 106, 80, 148, 36, 73, 102

4

- (a) FCFS

**60, 71, 147, 17, 106, 80, 148, 36, 73, 102**

4

- (b) SPTF (shortest positioning time first)

**60, 71, 73, 80,102,106, 147,148,36,17**

4

- (c) Elevator scheduling (SCAN) )

**60, 36, 17, 71, 73, 80,102,106, 147,148**

**Solution:** Similar to SPTF, but next in the same direction. It does not SCAN empty disk: switches directions if there is not further request. **Once yakini sonrakilerde aynı yone bakıyor. Eger basa veya sona goturduyseniz puan kirilmayacak. Yine baslangicta hatali yonden puan kirilmayacak**

4

- (d) C-SCAN (head at 63, diskin sonuna doğru tek yönlü)

**71, 73, 80,102,106, 147,148,17,36, 60**