BIL 301
2022 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 70 DAKİKADIR.
- Toplam da 110 puanlık soru olmakta iken alınabilecek maksimum not 100'dür: 10 puan bonus olarak düşünülmüştür.
- Cevaplarınızı okunaklı bir şekilde ayrılan kısımlara yazınız.
- Gerektiğinde diğer boş kısımlarıda kullanabilirsiniz.
- Hatalı soru tespitinde soru iptal edilecektir.

İsim ve Soyisim : _____

Medeniyet ID : _____

İmza : _____

Soru	Puan	Alınan skor
1	5	
2	5	
3	5	
4	5	
5	5	
6	5	
7	15	
8	5	
9	5	
10	5	
11	5	
12	5	
13	5	
14	5	
15	30	
Toplam:	110	

5

1. (read-copy-update) RCU ile ilgili aşağıda söylenenlerden hangisi **yanlıştır?** *(Eng:Which of the following statements about (read-copy-update) RCU technique is **wrong?**)*

   A. Writer–writer çekişmeleri normal senkronizasyon (lock gibi) yöntemleri kullanılarak yapılması. (*Eng: write conflicts are managed(synhronized) with regular techniques (e.g. locks)*)

   B. Data structure'a write yapıldığı zaman, takip eden readerların data struture'a erişimi kaldırılır (mesela pointerlar silinir) *(eng: When writing to the data structure, it does not allow subsequent readers to gain a reference to it by, for example, removing pointers).*

   C. Writer daha önceki readerlara dokunmayarak onların critical sectionlarını bitirmesini bekler. (*Eng: Writer waits for all previous readers to complete their RCU read-side critical sections*).

   D. Writer beklemenin sonunda (yani tüm readerlar bitince), yaptığı update'i yansıtır. (*Eng: At this point, there cannot be any readers who hold references to the data structure, so it now may safely be reclaimed*)

   E. Hiçbirisi (Yani hepsi doğru) *Eng: None of the above (all statements are correct.)*

   Solution: Bu soruda mantik hatasi oldugu icin soru iptal edilmistir. Aslinda anlasiliyor ama yapacak bir sey yok.

5

2. 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?*

A. 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.)*

B. 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`

C. 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.)*

5 3.

I Kernel içinde kullanılmak için interrupt disable/enable mekanizmasıyla yazılabilirler, user space kütüphanelerde interruptlarla implementasyonu sistem kontrolü yönünden sorun olacağı için olmaz (*Eng: In kernel, they can be implemented by disabling/enabling interrupts, but in the user space libraries this method leads to serious problems.*)

II 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.*).

Herhangi bir lock üzerinde acquire/release implementasyonu ile ilgili söylenen yukarıdakilerden hangileri doğrudur?

A. Sadece (only) I

B. Sadece (only) II

C. I, II

D. Hiçbirisi (None)

5 4. Diskin belirli kısmının loga ayrılarak, metadata yazma işlemlerinin öncelikle loga sonra diske yazılmasıyla, crach/reboottan sonra yapılan işlemlerin tekrarlarak sistemin recover edilmesinde oluşabilecek tutarsızlıkların giderilmesi tekniğinin ismi nedir? (*Eng: What is the name of the technique that works as reserving a portion of the disk for a log and writing any metadata operation first to the log and then to the disk in order to remove inconsistencies in the file system that may occur after a crash/recovery?*)

A. caching

B. journaling

C. c-scan

D. FTL

5 5. Dosya sistemleri ilgili hangisi söylenebilir? (*Eng: Which of the followings can be said related to the file systems?*)

A. Fat sisteminde sabit büyüklükte tablo kullanılır ve tabloda bir sonraki bloğun indeksi tutulur. (*Eng: In FAT file system, a table with fixed size is used to store the next block indices: each row shows the index for its next block.*)

B. Birbiriyle ilşkili objelerin aynı silindir grubunda toplanması dosya erişim işlemlerinin hızı artırılabilir. (*Eng: By gruping related references in the same disk, the disk access time can be decreased (the file ops can be faster).*)

C. Bir dosyadaki herhangi bir bloğa daha önceki bloklara uğramadan erişmeye random access denir. (*Eng: Accessing a block without going through the previous blocks is called random access.*)

D. FAT sistemi random accesslerde, blokların birbirine disk üzerinde linklendiği, linked dosya sistemine göre daha hızlıdır. (*Eng: Random accesses are in general faster in FAT system than linked file system where the file blocks are linked in the disk.*)

E. Hepsi (*eng: all of the above*)

5

6. 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

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

$$7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1.$$

Demand (istek) Paging'ın **3 framele** yapıldığını varsayarsak, aşağıdaki frame değiştirme algoritmalarında kaç tane page fault oluşur? (*Eng: Assuming demand paging with **three frames**, how many page faults would occur for the following replacement algorithms?*)

5

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

A. 9

B. 14

C. 18

D. 20

5

(b) FIFO algorithm

A. 13

B. 17

C. 20

D. 30

5

(c) Optimal algorithm

A. 9

B. 13

C. 16

D. 20

5

8.

```
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 büyüklüğü 200 byte olan üç tane page frame varsa ve bu framelerden birisi programın kodunu tutuyorsa, LRU algoritması kullanıldığında tahmini kaç tane page fault oluşur?

(diğer iki frame boş ve a. ve b. ayrı ayrı çalıştırılıyor) (*Eng: Consider the above code, if we use 3 frames with LRU algorithm how many page faults are expected while executing a. and b. seperately? Assume frame-1 holds the code and the other two frames are empty.*)

A. a:5000, b:50

B. a:50, b:50

C. a:5000, b:5000

D. a:5000, b: 50

5 9.

I Memory adreslerinin çok geniş ve seyrek olduğu yani çok büyük memory adres uzayının sadece belirli kısımlarının kullanıldığı durumlarda multilevel page table kullanmak (*Eng: Using multi-level page table when the adress space is huge and sparse*)

II TLB (Translation Lookaside Buffer) kullanma

Yukarıdakilerden hangisi demand pagingde memory erişimlerinin verimliliğini artırır? (*Eng: Which of the above increase memory access time efficiency in demand paging?*)

A. Yanlız(*only*) I

B. Yanlız(*only*) II

C. I ve II

D. Hiçbirisi (*None of the above*)

5 10. Bir processin çok CPUlu bir ortamda çalışma zamanı incelendiğinde, zamanın %90'nının I/O işlemlerine ve disk erişimlerine harcandığı 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?*)

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

B. Disk erişimlerine harcanan zamanın 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.*)

C. Eğer thrashing vars, processe ayrilan 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.*).

D. Eğer thrashing varsa thread sayısının azaltılması performansı artırabilir. (*Eng: If there is thrashing, decreasing number of threads may increase the performance.*)

E. Hepsi (*All of the above*)

5 11. I 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.*)

II 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.*)

Yukarıdakilerden hangisi **policy** hangisi ise **mechanism** (mekanizma)dır (*Eng: In the above, which one is policy and which one is mechanism*)?

A. I: policy, II: policy

B. I: policy, II: mechanism

C. I: mechanism, II:policy

D. I: mechanism, II: mechanism

5 12.

I Kullanıcıların dosyalara erişimlerinin sistem (yani admin) tarafından kontrol edilmesi. (*eng: All permissons for the users' access to files determined by the system (admin).*)

II 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.*)

Yukaridakilerden hangisi **DAC (discretionary access control)** hangisi ise **MAC (mandatory access control)** access (erişim) kontrolüdür. (*Eng: In the above, which one is an example of DAC (discretionary access control) and which one is an example of MAC (mandatory access control)*)?

A. I: DAC, II: DAC

B. I: DAC, II: MAC

C. I: MAC, II: DAC

D. I: MAC, II: MAC

5 13. Memory misslerde üretilen page faultlar aşağıdakilerden hangisidir? (*eng: What is a page fault occuring during memory misses?*)

A. Exception

B. Interrupt

C. System call

D. API

5 14.

I Interrupt kullanıldığında interrupt handler aygıtı servis ettiği için context switch yapılmaktadır.Polling de ise aygiti bizzat CPU servis eder. *Eng: When interrupts used, the interrupt handler services/-works with the device. In polling, CPU services the device.*

II Pollingde CPU düzenli kısa aralıklarla aygıtın servis edilip edilmediğini belirlemek için statüsünü 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.*

III Polling pek fazla servis edilmeyi bekleyen aygıt bulunamadığında, interruptlar ise devamlı aygıtların CPU'yu interrupt ettiğinde **verimli olmazlar**.*Eng: While polling becomes inefficient when CPU rarely finds a ready-device to be serviced; interrupt becomes inefficient if devices frequently interrupt the CPU.*

Yukarıda 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?)*

A. I, II, III

B. Sadece(only) I, II

C. Sadece(only) II, III

D. Sadece(only) I

E. Sadece(only) II

15. 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	3
P_2	1	1
P_3	1	1
P_4	2	2
P_5	2	1

10

(a) FCFS algoritması kullanıldığında, **non-preemptive** sistemde processlerin CPUda çalışma sıraları nasıldır? (*eng: When FCFS algorithm used in non-preemptive system, what is the order of the processes' CPU usage?*)

A. P1, P2, P3, P4, P5

B. P5, P4, P3, P2, P1

C. P1, P2, P3, P1, P5,P4, P1

D. P1, P2, P3, P5, P4, P1

10

(b) Shortest job first (SJF) algoritması kullanıldığında, **non-preemptive** sistemde processlerin CPUda çalışma sıraları nasıldır? Zamanlar eşit olunca öncelik yukarıdaki processe veriliyor. (*eng: When SJF algorithm used in non-preemptive system, what is the order of the processes' CPU usage? When the times are the same, the upper process (the lower index) has the priority.*)

A. P1, P2, P3, P4, P5

B. P2, P3, P5, P4, P1

C. P1, P2, P3, P5, P4

D. P1, P2, P3, P5, P1, P4

E. P1, P2, P3, P5, P4, P1

10

(c) Shortest remaining time first (SRT, preemptive SJF) algoritması kullanıldığında, processlerin CPUda çalışma sıraları nasıldır? Zamanlar eşit olunca öncelik yukarıdaki processe veriliyor. *eng: When SJF algorithm used in preemptive system, what is the order of the processes' CPU usage? When the times are the same, the upper process (the lower index) has the priority.*

A. P1, P2, P3, P4, P5

B. P2, P3, P5, P4, P1

C. P1, P2, P3, P5, P4

D. P1, P2, P3, P5, P1, P4

E. P1, P2, P3, P5, P4, P1