

BIL 301  
2023 Güz Dönemi  
Vize 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 50 DAKİKADIR.**
- Toplam da 150 puanlık soru olmakta iken alınabilecek maksimum not 100'dür: 50 puan bonus olarak düşünülmüştür.
- Cevaplarınızı okunaklı bir şekilde ayrılan kısımlara yazınız.
- Gerektiğinde diğer boş kısımlarda kullanabilirsiniz.

İsim ve Soyisim : \_\_\_\_\_

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

İmza : \_\_\_\_\_

Question	Points	Score
1	6	
2	6	
3	6	
4	6	
5	6	
6	6	
7	6	
8	6	
9	6	
10	6	
11	6	
12	6	
13	6	
14	6	
15	66	
Total:	150	

1. [6 pts] Linuxte, kullanıcı programları tarafından yapılabilecek ayrıcalıklı işleme bir tane örnek veriniz.Sadece ismini yazmanız yeterli. (Give an example privileged operation that can be needed by a user program.)?

1. \_\_\_\_\_ **input/output, creation of a new process, etc.** \_\_\_\_\_

2. [6 pts] Kullanıcı programları ayrıcalıklı işlem yapabilmeleri için hangi mekanizmayı kullanarak kernel tarafından sunulan servis alabilirler.(Which mechanism user programs use to get privileged operations serviced by the kernel. )?

2. \_\_\_\_\_ **system calls** \_\_\_\_\_

3. [6 pts] Sistemin user moddan kernel moda transition yapmasını sağlayan, ve kullanıcı programlarının ayrıcalıklı işlem yapabilmeleri için gönderdikleri instruction requestlere ne denir.(What is the name of the instruction mechanism that cause the system to transition from user mode to kernel mode in order to answer the request sent by the user application. )?

A. Hardware interrupt

B. Trap

4. [6 pts] Threadlerin kullanıcı alanında yapılması neden daha hızlı olabilir? (Why implementation of a thread library in user space would be faster?)

**Solution:** Kernel uzayında gerçekleştirilmesi durumunda, threadlerle alakalı system callların context switch gerektirmeleri (System calls requires context switch when threads are implemented in kernel space...)

5. [6 pts] System calllarında kullanıcı programından system calla parametreler nasıl geçirilir? (How are the parameters passed from user program to a system call?)

A. *copy\_to\_user* kullanılarak (using *copy\_to\_user*)

B. *File* kullanılarak (using File)

C. Küçük büyülükler için registerlar üzerinde, büyükler için stack kullanılarak (for msall number of variables, by using registers, for larger numbers by using stack)

D. *mmap* ile kernella paylaşımı memory oluşturarak (creating shared memory with kernel by using mmap)

6. [6 pts] (politika) policy ve mekanizma (mechanism) için 1er tane örnek veriniz.

(a) **Policy**

(a) \_\_\_\_\_ **CPU timer'in belirli bir grup process için ne kadar işletilmesi gerektiği** \_\_\_\_\_

(b) **Mechanism**

(b) \_\_\_\_\_ **CPUya koruma sağlayan timer oluşturulması** \_\_\_\_\_

7. [6 pts] Askeri bir ortamda, farklı bölümlerin olduğu ve her bir bölüm kendine ait donanımları ve diğer kısımların donanımlarına erişimde yetki ve kısıtlamalarının olduğu bir ortamda, sadece güvenliği göz önüne alırsanız hangi yapıda bir kernel dizayn edersiniz, neden? (Consider a military-like environment where there are departments and each department has their own hardware and devices. And each department can access others' resources by following strict regulations and permissions. Considering only security, which structure would you use to design an OS for this environment. Why? ) Doğru cevap açıklamanıza bağlı, açıklamasız cevaplara puan verilmeyecektir.

- Monolithic kernel
- Layered Kernel

- Microkernel

**Sebebi (Reason):**

7. \_\_\_\_\_ ...

**Solution:** microkernel, yada layered kernel sebebi güvenlikte her ikisinde de moduller tamamen ayristir-ilarak layerlar arasında checkler artırlarak ve yetkiler tanımlanarak security artırılabilir

8. **[6 pts]** Bir process çalışırken timer interruptla çalışması kesildikten sonra hangi durumda olur? (What would be the status of a running process after it is preempted because of the timer interrupt?)

- A. New
- B. Running
- C. Waiting
- D. Ready**
- E. Terminated

9. **[6 pts]** Bir zombie processin durumu aşağıdakilerden hangisidir?(What is the status of a zombie process)?

- A. New
- B. Running
- C. Waiting
- D. Ready
- E. Terminated**

10. **[6 pts]** İki CPU corelu ve her biri tek threadli bir sistemde running statüde maksimum kaç tane process olabilir? (How many process' status can be running in a system with 2 cores?)

10. \_\_\_\_\_ **2**

11. **[6 pts]** Aşağıdaki boşlukların herbirine "make", "make install", ve "make menuconfig" den en uygun olanını yazınız. (Her yanlış -2 puan.)

- (a) make Kernel source dosyasındaki(mevcut directoryde) bulunan makefile içerisinde verilen komutları çalıştırarak kerneli compile ve build eder.
- (b) make menuconfig linux kaynak kodunu compile etmeden önce gerekli konfigurasyonları yapmamızı sağlar.
- (c) make install Build edilen kernel'ı mevcut sisteme yüklememizi sağlar.

12. **[6 pts]** Aşağıdaki processlerden hangisinin gönüllü context switch yapması beklenir? (Which of the followings can context switch voluntarily?)

- A. I/O bound process**
- B. CPU bound process

13. **[6 pts]** Aşağıdaki CPU scheduling algoritmalarından **starvation'a** sebep **olmaz** (her yanlış seçim -2 puan, **doğru seçim 3 puanıdır**). (Which of the following algorithms does **not** cause **starvation**?)

- Earliest deadline first (preemptive)**
- Rate monotonic scheduling (preemptive)
- First come first serve**
- Shortest job first (non-preemptive)

- Shortest remaining time first (preemptive SJF)

✓ Round robin

- Priority

14. [6 pts] Round robin CPU scheduling algoritmasında  $q$  (quantum) değeri çok büyük olursa algoritma **FCFS** algoritmasına indirgenmiş, çok küçükte olursa **context switchlerin** sayısı artırılmış olur.(if  $q$  is too great RR becomes equivalent to... else if  $q$  is too small, then we increase the number of...)

15. Aşağıda processlerin geliş zamanları ve beklenen çalışma (burst) zamanları verilmiştir. İstenilen her bir algoritma için (Below process arrival times and burst times are given. For each algorithm in the choices.)

- processlerin durumlarını zamana göre grafikleyerek ve CPU Gantt chart çiziniz. (draw time vs process status and CPU Gantt charts)
  - AWT (ortalama bekleme zamanı) değerlerini hesaplayınız. (Compute average waiting time)
  - Hatalı sonuçlara ilgili kısımdan 0 puan verilecektir. Kısmi notlandırma yapılmayacaktır.

Process	Arrival(varış) Time	Burst time
$P_1$	0	8
$P_2$	0	6
$P_3$	1	4
$P_4$	1	3
$P_5$	5	1

- (a) FCFS algoritması kullanıldığında,

- i) [5 pts] Time vs Process states

- ii) [2 pts] Gantt chart

Process	Waiting time
$P_1$	...
$P_2$	...
$P_3$	...
$P_4$	...
$P_5$	...

- iii) [2 pts] Process waiting times

- iv) 2 pts AWT: \_\_\_\_\_

- Round robin ( $q = 8$ ) kullanıldığında

ii) [2 pts] Gantt chart

Page 1 of 1

### Process Waiting time

$P_1$	...
$P_2$	...
$P_3$	...
$P_4$	...
$P_5$	...

iii) [2 pts] Process waiting times

(c) Shortest job first (SJF) algoritması kullanıldığında,

i) [5 pts] Time vs Process states

ii) [2 pts] Gantt chart

### Process Waiting time

$P_1$	...
$P_2$	...
$P_3$	...
$P_4$	...
$P_5$	...

iv) [2 pts] AWT:

(d) Shortest remaining time first (SRT, preemptive SJF) algoritması kullanıldığında,

j) [5 pts] Time vs Process states

ii) [2 pts] Gantt chart

\_\_\_\_\_

### Process Waiting time

$P_1$	...
$P_2$	...
$P_3$	...
$P_4$	...
$P_5$	...

iii) [2 pts] Process waiting times

iv) [2 pts] AWT:

Process	Arrival(varış) Time	Burst time	Priority (1-highest, 5-lowest (1-en yüksek, 5-en düşük))
$P_1$	0	8	5
$P_2$	0	6	4
$P_3$	1	4	3
$P_4$	1	3	2
$P_5$	5	1	1

(e) Non-Preemptive priority scheduling algoritması

i) **5 pts** Time vs Process states

ii) [2 pts] Gantt chart

Page 1 of 1 pages

### Process Waiting time

$P_1$	...
$P_2$	...
$P_3$	...
$P_4$	...
$P_5$	...

iv) [2 pts] AWT: \_\_\_\_\_

(f) Preemptive priority scheduling algoritması

i) [5 pts] Time vs Process states

ii) [2 pts] Gantt chart

### Process Waiting time

$P_1$	...
$P_2$	...
$P_3$	...
$P_4$	...
$P_5$	...

iii) [2 pts] Process waiting times

iv) [2 pts] AWT: \_\_\_\_\_

**Solution:** Bu soruda round robin ( $q=8$ ) ile FCFS priority(nonpreemptive) ile SJF priority(preemptive) ile SRT ayni cozume sahipler..

## NONPREEMPTIVE PRIORITY process vs Time Graph

Process ID																						
P5																						
P4		W	W	W	W	W	W	P4	P4	P4												
P3		W	W	W	W	W	W	W	W	W	P3	P3	P3	P3								
P2	P2	P2	P2	P2	P2	P2																
P1	W	W	W	W	W	W	W	W	W	W	W	W	W	P1	P1	P1	P1	P1	P1	P1	P1	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

## Gantt Chart

P2	P2	P2	P2	P2	P5	P4	P4	P3	P3	P3	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

**Prior** AT BT FT WT

P1	<b>5</b>	0	8	22	<b>14</b>
P2	<b>4</b>	0	6	6	<b>0</b>
P3	<b>3</b>	1	4	14	<b>9</b>
P4	<b>2</b>	1	3	10	<b>6</b>
P5	<b>1</b>	5	1	7	<b>1</b>
Average		1.4	4.4	11.8	<b>6</b>

$$WT = FT - (AT + BT)$$

**PREEMPTIVE PRIORITY process vs Time Graph**

Process ID					P5																	
P5																						
P4		P4	P4	P4																		
P3		W	W	W	P3	W	P3	P3	P3													
P2	P2	W	W	W	W	W	W	W	W	P2	P2	P2	P2	P2								
P1	W	W	W	W	W	W	W	W	W	W	W	W	W	W	P1	P1	P1	P1	P1	P1	P1	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Gantt Chart

P2	P4	P4	P4	P3	P5	P3	P3	P2	P2	P2	P2	P1	P1	P1	P1	P1	P1	P1	P1	P1	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

**Prior AT BT FT WT**

P1	<b>5</b>	0	8	22	<b>14</b>
P2	<b>4</b>	0	6	14	<b>8</b>
P3	<b>3</b>	1	4	9	<b>4</b>
P4	<b>2</b>	1	3	4	<b>0</b>
P5	<b>1</b>	5	1	6	<b>0</b>
Average		1.4	4.4	11	<b>5.2</b>

WT = FT - (AT+BT)

## FCFS process vs Time Graph

Process ID																						
P5																						
P4																						
P3																						
P2																						
P1	P1	P1	P1	P1	P1	P1	P1	P1														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

## Gantt Chart

P1	P1	P1	P1	P1	P1	P1	P1	P2	P2	P2	P2	P2	P2	P3	P3	P3	P4	P4	P4	P5	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

AT BT FT WT

P1	0	8	8	<b>0</b>
P2	0	6	14	<b>8</b>
P3	1	4	18	<b>13</b>
P4	1	3	21	<b>17</b>
P5	5	1	22	<b>16</b>
Average	1.4	4.4	16.6	<b>10.8</b>

$$WT = FT - (AT + BT)$$

RR23-q8 process vs Time Graph																						
Process ID																						
P5																						
P4																						
P3																						
P2																						
P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

  

Gantt Chart																						
	P1	P1	P1	P1	P1	P1	P1	P1	P2	P2	P2	P2	P2	P2	P3	P3	P3	P3	P4	P4	P5	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

  

	AT	BT	FT	WT
P1	0	8	8	0
P2	0	6	14	8
P3	1	4	18	13
P4	1	3	21	17
P5	5	1	22	16
Average	1.4	4.4	16.6	10.8

  

WT = FT - (AT+BT)



SRT process vs Time Graph	
Process ID	
P5	P5
P4	P4 P4
P3	W W W P3 W P3 P3
P2	P2 W W W W W W W P2 P2 P2 P2
P1	W W W W W W W W W W W W P1 P1 P1 P1 P1 P1 P1
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

  

Gantt Chart																					
P2	P4	P4	P4	P3	P5	P3	P3	P3	P2	P2	P2	P2	P2	P1	P1	P1	P1	P1	P1	P1	P1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

  

	AT	BT	FT	WT
P1	0	8	22	14
P2	0	6	14	8
P3	1	4	9	4
P4	1	3	4	0
P5	5	1	6	0
Average	1.4	4.4	11	5.2

WT = FT - (AT+BT)