

BIL 301 2024 Güz Dönemi Vize Soruları

- **Fotoğraflı üniversite kimliği** olmayan bir öğrenci bu sınava giremez. Üzerinde kimliği olmayanların fotoğrafları çekilib isimleri not alınacaktır (kimliklerinin sonradan teyidi için).
- Her türlü kağıt, kitap, not ve hesap makinesi, telefon, saat vb. her türlü elektronik aygit 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 90 DAKİKADIR.**
- Toplam da 104 puanlık soru olmakta iken alınabilecek maksimum not 100'dür: 4 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	3	
2	3	
3	3	
4	5	
5	4	
6	3	
7	3	
8	3	
9	3	
10	3	
11	3	
12	8	
13	3	
14	3	
15	3	
16	36	
17	15	
Total:	104	

1. [3 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**
2. [3 pts] Threadlerin kullanıcı alanında yapılması (M:1 veya M:N thread eşleşmesi ile) neden daha hızlı olabilir? (Why implementation of a thread library in user space (M:1 threading or M:N threading) would be faster?)

2. _____

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

3. [3 pts] Threadlerin kullanıcı uzayında (M:1 veya M:N thread eşleşmesi ile) yapılması durumunda ortaya çıkan kısıtlamalardan 1 tanesini yazınız? (Write one of the limitations/complications of fully-user-level thread library(M:1 threading or M:N threading))?

3. _____

Solution: User does not know which kthread is blocked, available CPUs, system calls blocks, deadlocks..

4. [5 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 _____

5. Aşağıdaki boşluklara en uygun kelimeleri yazınız.

- (a) [2 pts] linker Birden fazla object file'ı ve static kütüphane fonksiyonlarını birleştirerek binary dosya oluşturur: mesela, `gcc -o main main.o -lm` ile `main` binary (executable file) dosyasının oluşturulması
- (b) [2 pts] loader `./main` ile belirlenen programı yeni oluşturulan processin memory address space'ine yükleyerek çalışmaya hazır hale getirir
6. [3 pts] Askeri bir ortamda, farklı bölümlerin olduğu ve her bir bölümün kendine ait donanımları ve diğer kimselere 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):

6. _____ ...

Solution: microkernel, yada layered kernel sebebi güvenlikte her ikisinde de modüller tamamen ayarlılarak layerlar arasında checkler artırılarak ve yetkiler tanımlanarak security artırılabilir

- [3 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
- [3 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
- [3 pts] 4 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 4 cores?)

9. _____ 4

- [3 pts] Aşağıdaki boşlukların her birine "make", "make install", ve "make menuconfig" den en uygun olanını yazınız. (Her yanlış -2 puan.)
 (a) **make install** Build edilen kernel'i mevcut sisteme yüklememizi sağlar.
 (b) **make** Kernel source dosyasındaki(mevcut directoryde) bulunan makefile içerisinde verilen komutları çalıştırarak kerneli compile ve build eder.
 (c) **make menuconfig** linux kaynak kodunu compile etmeden önce gerekli konfigurasyonları yapmamızı sağlar.
- [3 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
- [8 pts] Aşağıdaki CPU scheduling algoritmalarından **starvation** sebep olur (her yanlış seçim bir doğruya götürmektedir). (Which of the following algorithms causes **starvation**?)
 Earliest deadline first (preemptive)
 Rate monotonic scheduling (preemptive)
 Deadline monotonic scheduling (preemptive)
 First come first serve
 Shortest job first (non-preemptive)
 Shortest remaining time first (preemptive SJF)
 Round robin
 Priority
- [3 pts] Round robin CPU scheduling algoritmasında q (quantum) değeri küçültüldüğünde context switch sayısı muhakkak (yani her zaman) artar. (Number of context switches **increases** every time we decrease the value of q in round robin algorithm.
 A. True B. False

14. [3 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. [3 pts] Aşağıda SJF(shortest job first) algoritmasında bir processinin daha önceki görülen (a_i) ve tahmin edilen (τ_i) CPU burst zamanları verilmiştir. τ_2 değeri nedir? (Below is the observed(a_i) and estimated (τ_i) CPU burst times of a process in SJF algorithm. What is the value of τ_2 ?)

i	0(baslangic)	1	2	3
a_i		6	4	6
τ_i	10	8	<u>6</u>	?

16. Aşağıda processlerin geliş zamanları ve beklenen çalışma (burst) zamanları verilmiştir. İstenilen herbir 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	7
P_2	0	6
P_3	0	5
P_4	1	3
P_5	1	1

(a) FCFS algoritması kullanıldığında,

- i) [3 pts] Time vs Process states

- ii) [1 pts] Gantt chart

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

- iii) [1 pts] Process waiting times

- iv) 1 pts AWT: _____

- Round robin ($q = 7$) kullanıldığında

ii) [1 pts] Gantt chart

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

iii) [1 pts] Process waiting times

(c) Shortest job first (SJF) algoritması kullanıldığında

j) [3 pts] Time vs Process states

ii) [1 pts] Gantt chart

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

iii) [1 pts] Process waiting times

iv) [1 pts] AWT: _____

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

i) [3 pts] Time vs Process states

ii) [1 pts] Gantt chart

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

iii) [1 pts] Process waiting times

iv) 1 pts AWT: _____

Process	Arrival(varış) Time	Burst time	Priority (1-highest, 5-lowest (1-en yüksek, 5-en düşük))
P_1	0	7	5
P_2	1	6	4
P_3	2	5	3
P_4	3	3	2
P_5	4	1	1

(e) Non-Preemptive priority scheduling algoritması

i) [3 pts] Time vs Process states

ii) [1 pts] Gantt chart

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

iii) [1 pts] Process waiting times

[REDACTED]

iv) [1 pts] AWT: _____

(f) Preemptive priority scheduling algoritması

i) [3 pts] Time vs Process states

ii) [1 pts] Gantt chart

Process Waiting time

P₁ ...

$$P_2 \quad \dots$$

$$P_3 \dots$$

$P_4 \dots$

$P_5 \dots$

iv) 1 pts AWT: _____

Solution: Bu soruda round robin ($q=7$) ile FCFS aynı çözüm sahipler..

FCFS process vs Time Graph	
Process ID	
P5	:W :P5
P4	:W :P4 :P4
P3	W :W :W :W :W :W :W :W :W :W :W :W :W :W :W :W :P3 :P3 :P3 :P3
P2	W :W :W :W :W :W :P2 :P2 :P2 :P2 :P2 :P2 :P2 :P2 :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

Gantt Chart	
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	7	7	0
P2	0	6	13	7
P3	0	5	18	13
P4	1	3	21	17
P5	1	1	22	20
Average	0.4	4.4	16.2	11.4

Round robin q = 7 için FCFS ile aynı sonucu veriyor.

WT = FT - (AT+BT)

SJF process vs Time Graph

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

Gantt Chart

P3	P3	P3	P3	P3	P5	P4	P4	P4	P2	P2	P2	P2	P2	P2	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

	AT	BT	FT	WT
P1	0	7	22	15
P2	0	6	15	9
P3	0	5	5	0
P4	1	3	9	5
P5	1	1	6	4
Average	0.4	4.4	11.4	6.6

WT = FT - (AT+BT)

SRT process vs Time Graph

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

Gantt Chart

P3	P5	P4	P4	P4	P3	P3	P3	P2	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

	AT	BT	FT	WT
P1	0	7	22	15
P2	0	6	15	9
P3	0	5	9	4
P4	1	3	5	1
P5	1	1	2	0
Average	0.4	4.4	10.6	5.8

WT = FT - (AT+BT)

NONPREEMPTIVE PRIORITY process vs Time Graph																						
Process ID	P5																					
P5					W	W	W	W	P5													
P4				W	W	W	W	W	P4	P4	P4											
P3		W	W	W	W	W	W	W	P3	P3	P3	P3										
P2	W	W	W	W	W	W	W	W	W	W	W	W	W	P2	P2	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

Gantt Chart																						
	P1	P1	P1	P1	P1	P1	P1	P5	P4	P4	P4	P3	P3	P3	P3	P2	P2	P2	P2	P2	P2	
	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	5	0	7	7	0
P2	4	1	6	22	15
P3	3	2	5	16	9
P4	2	3	3	11	5
P5	1	4	1	8	3
Average		2	4.4	12.8	6.4

WT = FT - (AT+BT)

PREEMPTIVE PRIORITY process vs Time Graph

Process ID																						
P5					P5																	
P4				P4	W	P4	P4															
P3			P3	W	W	W	W	P3	P3	P3	P3											
P2		P2	W	W	W	W	W	W	W	W	W	P2	P2	P2	P2	P2	P2					
P1	P1	W	W	W	W	W	W	W	W	W	W	W	W	W	W	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	P2	P3	P4	P5	P4	P4	P3	P3	P3	P3	P2	P2	P2	P2	P2	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

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

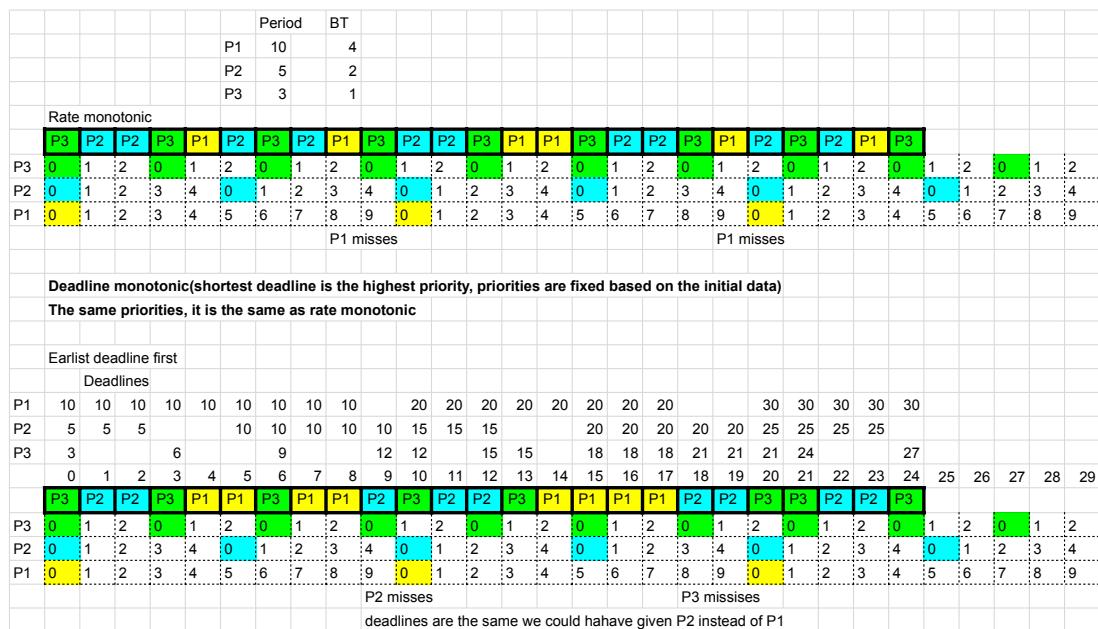
17. [15 pts] Aşağıdaki processler için 0-25 zaman aralığında Gantt chartları çiziniz.

Process	Period (tüm processler için başlangıç 0)	Burst time	Periodtaki Deadline
P_1	10	4	10(Periodun sonu)
P_2	5	2	5(Periodun sonu)
P_3	3	1	3(Periodun sonu)

(a) Preemptive rate monotonic algoritması

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Solution: rate monotonic priorityler ratelerle orantılı: periodu kısa olan high priority
Yani oncelik sirasina gore P_3, P_2, P_1 oluyor. deadline monotonic ve rate monotonic cozumleri aynı. Earliest deadline cizerek farkli secimlere gore farkli Gantt chart cikabilir. Mantikli bir sey yapanlara tam puan verilecek ve kucuk hatalardan puan kirilmayacak.



(b) Preemptive deadline monotonic algoritması

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

(c) **Preemptive** earliest deadline first algoritması

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--