

1-) Proses çizelgeleyicisi (scheduler), CPU'da işlenecek prosesi bekleyen prosesler arasından seçer?

- a-) Hazır kuyruğunda b-) Giriş/Çıkış kuyruğunda  
c-) Aygıt kuyruğunda d-) Proses kuyruğunda  
e-) İş kuyruğunda

2-) Bir proseste yerel değişkenlerin saklandığı kısım hangisidir?

- a-) Veri Bölümü b-) Bellek Kümesi-Yığıt (heap)  
c-) Yığın (stack) d-) Program kodu  
e-) Program sayacı

3-) Aşağıdakilerden hangisi prosesin bölümlerinden değildir?

- a) Yığın (stack) b) Yığıt (heap) c) Data d) Kod e) Id

4-) Aşağıdakilerden hangisi proses kontrol bloğunda yer almaz?

- a-) Proses durumu b-) Program sayacı  
c-) CPU kaydedicileri d-) Proses kodu  
e-) Proses numarası

5-) Aşağıdakilerden hangisi sıradan tüneller (pipe) hakkında doğru değildir?

- a-) Ebeveyn ve çocuk proses ilişkisi olmalı  
b-) İletişim tek yönlüdür  
c-) Genelde, parent proses pipe oluşturur  
d-) Ebeveyn bir uçtan yazarken çocuk proses diğer uçtan okur  
e-) Ortak bir bellek alanına veri yazılır

6-) İş parçacıkları ile alakalı aşağıdakilerden hangisi yanlıştır.

- a-) Çoğu modern uygulamalar çoklu iş parçacıklıdır  
b-) Proses, çalıştırılacağı zaman hard diske yüklenmelidir.  
c-) İş parçacığı oluşturma proses oluşturmaya göre daha az kaynak ve zaman tüketir.  
d-) Bir iş parçacığının kendisine ait bir ID'si, program sayacı, vardır.  
e-) İş parçacıkları prosesin kaynaklarını kullanır.

7-) 7. ve 8. Sorular için kod parçası:

- ```
1. int deger=5;
2. int main()
3. {
4.   int pid=8, deger2=9;
5.   pid=fork();
6.   if (pid==0) some giracal
7.   {
8.     printf("pid:%d",pid);
9.     printf("deger:%d",deger);
10.  } else
11.  {
12.    printf("deger2:%d",deger2);
13.  }
14. }
```

Yukarıdaki programın ekran çıktısı sırasıyla (pid, deger, deger2) ne olabilir?

- a-) 0,0,9 b-) 8,5,9 c-) 2154,0,9 d-) 2154,0,5 e-) 0,5,9

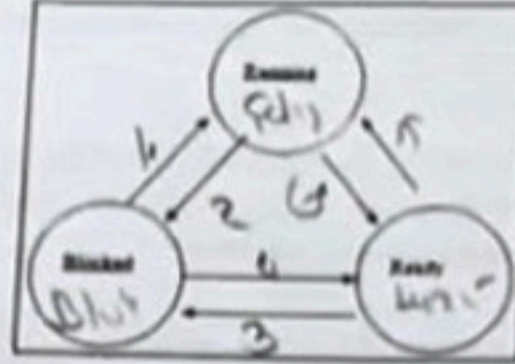
8-) Yukarıda verilen kod parçası ile alakalı hangisi söylenemez?

- a-) Yeni bir proses oluşmuştur  
b-) Hangi prosesin önce çalışacağı belli değildir  
c-) Çocuk prosesin ebeveyn prosesini beklemesi için 8. Satıra wait(NULL) komutu yazılabilir  
d-) fork() sistem çağrısının geri dönüş değeri proseslere göre değişir  
e-) Ebeveyn ve çocuk prosesler getpid() fonksiyonuyla kendi ID değerlerini öğrenebilirler

- ```
9-)
1. void foo(int *a)
2. {
3.   printf("%d\n", *a); pthread_exit(0);
4. }
5. int main()
6. {
7.   pthread_t tid[N]; pthread_attr_t attr; int i;
8.   pthread_attr_init(&attr);
9.   for(i=0; i<N; i++){
10.    pthread_create(&tid[i], &attr, (void *)&foo, &i);
11.  }
12.  for(i=0; i<N; i++)
13.    pthread_join(tid[i], NULL);
14.  pthread_exit(0); }
```

Yukarıdaki programın ekran çıktısı 0, 1, 2, 3, 4 olması için nasıl bir düzeltme yapılmalıdır.

- a-) pthread\_join() 9. Satırdaki for'un içine alınmalı  
b-) foo fonksiyonundaki pthread\_exit() komutu kaldırılmalı  
c-) 14. Satırdaki pthread\_exit() komutu kaldırılmalı  
d-) 14. Satırdaki pthread\_exit() 9. Satırdaki for'un içine alınmalı  
e-) pthread\_create döngünün dışında 1 kere çağırılmalı  
10-) Bir proses ne zaman "Çalışıyor" durumundan "Bekleme" durumuna geçebilir?  
I-) Proses sonlandığında II-) Bir kesme geldiğinde  
III-) Bellekten veri alınırken IV-) Giriş/Çıkış işlemi gerektiğinde  
V-) Diske veri yazacağı zaman  
a-) III-V b-) I-II c-) IV d-) III-IV e-) III-IV-V



Proseslerin yukarıdaki şekilde görüldüğü gibi sadece 3 durumda olabildiğini düşünün: Running, Ready ve Blocked. Şekilden de görüldüğü gibi 6 çeşit durum geçişi mümkündür (1'den 6 ya kadar etiketlenmiş). Mümkün olmayan geçiş(ler) aşağıdakilerden hangisidir?

- a-) 1 ve 3 b-) 2 ve 3 c-) Sadece 4 d-) 2 ve 5 e-) Sadece 6

12-) Aşağıdaki kod bloğu çalışır mı? Çalıştığı takdirde ekran çıktısı nasıl nedir?

```
class Multi extends Thread{
  public void run()
  {
    System.out.println("thread is running...");
  }
  public static void main(String args[]) throws
  InterruptedException{
    Multi t1=new Multi();
    t1.start();
    t1.wait(1000);
    t1.start();
  }
}
```

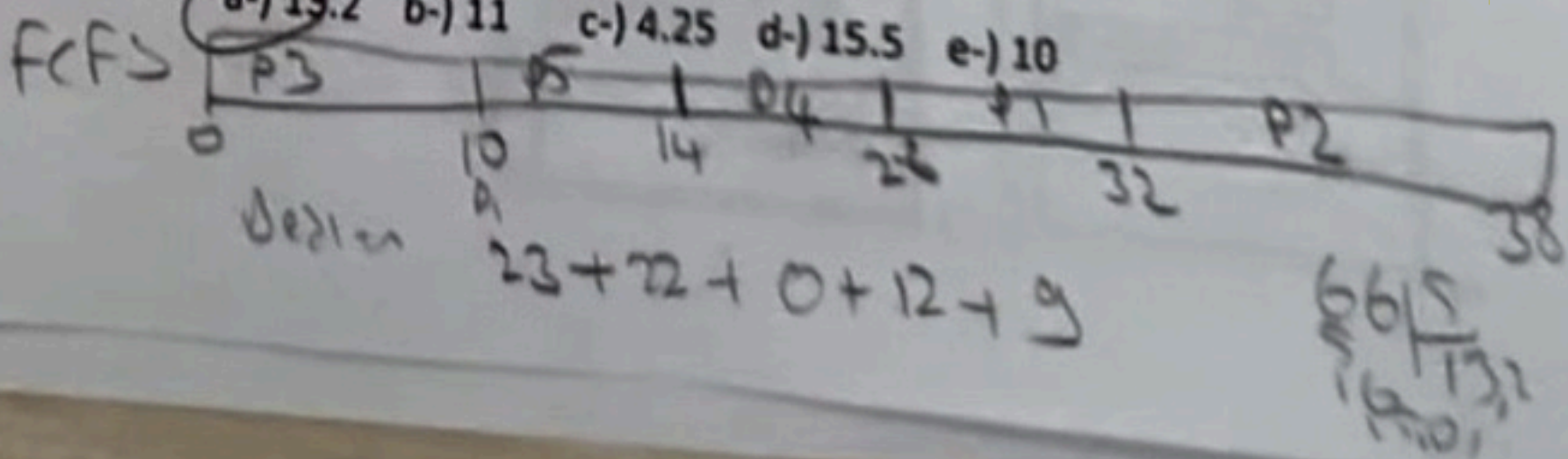
- a) Kod bloğu çalışmaz b) thread is running  
c) thread is running  
1000 thread is running  
d) thread is running  
thread is running  
e) thread is running 1000

13-19. soruları aşağıdaki tabloya göre yapınız.

	Varış Zamanı	CPU Patlama Zamanı	Öncelik
P1	5. ms	4 ms	2
P2	10. ms	6 ms	1
P3	0. ms	10 ms	3
P4	2. ms	14 ms	2
P5	1. ms	4 ms	0

13-) Yukarıda verilen Tablo verilerine göre, eğer ilk Gelen ilk Çalışır (FCFS) Çizelgeleme Algoritması çalıştırılırsa ortalama bekleme zamanı kaç ms olur?

- a-) 13.2 b-) 11 c-) 4.25 d-) 15.5 e-) 10





14-) Yukarıda verilen Tablo verilerine göre, eğer en kısa kalan zaman önce (Kesintili) Çizelgeleme Algoritması çalıştırılırsa prosesinin çalışma aralığı hangisidir?

- a-) 0-1 ve 5-14  
b-) 0-10 ve 25-27  
c-) 0-1, 9-10 ve 16-24  
d-) 2-12  
e-) 16-22 ve 25-27

15-) Yukarıda verilen Tablo verilerine göre, öncelikli (Kesintili-preemptive) planlama (çizelgeleme) algoritması çalıştırılırsa ortalama cevap süresi kaç ms olur?

- a-) 2.8 b-) 13.25 c-) 2 d-) 2.5 e-) 14.75

16-) Yukarıda verilen Tablo verilerine göre, Çevrimsel Sıralı (Round Robin-RR) planlama (çizelgeleme) algoritması çalıştırılırsa ve quantum süresi 7 ms olarak verilirse ortalama tamamlanma süresi kaç ms olur?

- a-) 25.25 b-) 22.25 c-) 30.75 d-) 12.25 e-) 14.75

17-) Yukarıda verilen Tablo verilerine göre, Çevrimsel Sıralı (Round Robin-RR) Çizelgeleme Algoritması çalıştırılırsa ve quantum süresi 7 ms olarak verilirse ortalama çıkış (throughput) ne olur?

- a-) 4 b-) 0.132 c-) 1.22 d-) 3.25 e-) 2.571

18-) Yukarıda verilen Tablo verilerine göre, Eğer 60. ms'de sonunda CPU kullanımı değeri yaklaşık (%-yüzde) kaç olurdu?

- a-) %100 b-) %88.33 c-) %86.66 d-) %93.33 e-) %71

19-) Yukarıda verilen Tablo verilerine göre, "İş bitirme zamanı"na göre hangi planlama (çizelgeleme) algoritması en iyisidir?

- a-) Çevrimsel Sıralı (Round Robin-RR)  
b-) Öncelikli (Kesintili)  
c-) En kısa kalan zaman önce (Kesintili)  
d-) En kısa iş önce (Kesintisiz)  
e-) İlk Gelen İlk Çalışır

20-) Aşağıda verilen CPU Patlama/çalışma süreleri ve tahmin sürelerine göre  $\alpha$  değeri nedir?

Gerçek CPU Patlama değeri (ti)  
Tahmini değer (tj)

- a) 1/2 b) 1/4 c) 1/3 d) 1/8 e) 1

21-) Aşağıda verilen CPU Patlama zamanlarına göre en uygun kuantum zamanının (q) 8 ms olması için ? yazan yerlere aşağıdakilerden hangisi gelemez?

- a) 6-8 b) 8-10 c) 13-14 d) 15-22 e) 100-120

22-) Amdahl Kuralına göre bir programın %60'ı paralel talimatlara sahiptir. Programı 2 kat hızlı çalıştırmak istersek en az kaç çekirdek gereklidir.

- a) 1 b) 2 c) 6 d) 4 e) 5

TEXT (kod)	A
data	B
heap (yığın)	C
stack (yığın)	D

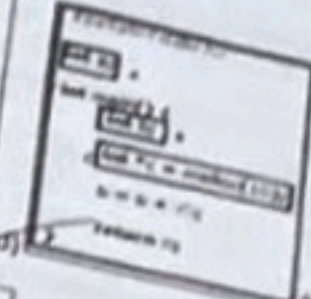
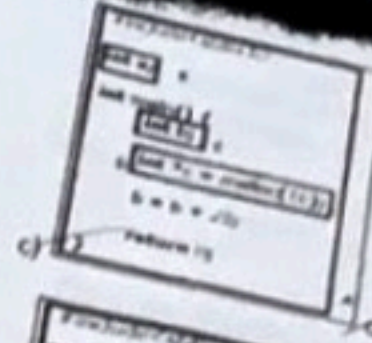
Yukardaki şekilde gösterilen bir prosesin bellekteki bölgelerine göre aşağıdakilerden hangisi doğrudur?

```
#include <stdio.h>
int a;
int main() {
    int b;
    c;
    int *c = malloc(10);
    b = b + 25;
    return 0;
}
```

```
#include <stdio.h>
int a;
int main() {
    int b;
    c;
    int *c = malloc(10);
    b = b + 25;
    return 0;
}
```

a)

b)



24-) Bir prosesin çalışması için 40 CPU (burst) zamanı gereksin, bu proses çok-seviyeli hazır kuyruğu (multi-level queue) olan bir sistemde çalışsın. Yeni bir proses çalışmaya ilk kuyruktan başlayıp, bir quantum süresi çalıştıktan sonra bir alt seviye kuyruğa düşürülsün (demotion). Her bir kuyruk Round-Robin yöntemi kullansın. Başlangıç kuyruğunun quantum süresi 2 birim zaman ve her bir alt düzeyde quantum süresi bir önceki göre 5 birim zaman artsın. Buna göre bu prosesin çalışması hangi seviye kuyruğa tamamlanır? (Hazır Kuyruk Seviyeleri: 0, 1, 2, 3, 4, 5 - şekildedir)

25-) "MyThread t=new MyThread();" şeklinde oluşturulan bir threadi başlatmak için kullanılan metod aşağıdakilerden hangisidir?

- a) t.wait(); b) t.sleep(); c) public void run() d) t.start(); e) t.join();

Süre 80 dakikadır. Başarılar

P3	P5	P1	P3	P2	P3	
1	5	9	10	16	24	

P3	P5	P1	P4	P2		P4	P3
1	5	9	10	16	24		

Beilen	P1	P2	P3	P4	P5	
0	0	28	13	0	4/5 = 0.25	

Carp	0	0	0	70	4/5	
------	---	---	---	----	-----	--

10 - 2. birim  
7. birim  
1. kuyruk  
2. kuyruk

- 0 - 2  
1 - 7  
2 - 12  
3 - 17  
4 -